8-9-2014

Fake Real Quadratic Orders

Richard Michael Oh
University of South Carolina - Columbia

Fake Real Quadratic Orders

by

Richard Michael Oh

Bachelor of Science
Emory University 2010

Submitted in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy in

Mathematics

College of Arts and Sciences

University of South Carolina

2014

Accepted by:

Frank Thorne, Major Professor

Michael Filaseta, Committee Member

Matthew Ballard, Committee Member

Ralph Howard, Committee Member

Duncan Buell, External Examiner

Lacy Ford, Vice Provost and Dean of Graduate Studies

Dedication

To my Love, Melissa.

Acknowledgments

I would like to show a deep appreciation and thanks to my advisor, mentor, and friend, Frank Thorne. Without his encouragement, genius intellect, and support, I would not have been able to complete this.

I would also like to extend additional gratitude to Duncan Buell. His expertise in binary quadratic forms gave me the insight to tackle the problems in a different light. His support was invaluable.

I give my sincere thanks to the Department of Mathematics and the Graduate School of the University of South Carolina at Columbia, who together made it possible to accomplish this landmark.

Many thanks to my parents, Thomas and Yung Oh, for their support throughout my years in graduate school. Without them, my life would have been much more complicated.

Many thanks to Netflix, Hulu, iHeart Radio, Team Fortress 2, Titanfall, and all other technological vices that have allowed me to get refreshed, refocused, and motivated which have allowed me to finish this milestone.

Thanks to Shahein Tajmir for the late night banter and help with my computer issues. I would like to mention Bethany Wentzky for her comic relief through busy and trying times.

Finally, my goals, dreams, and accomplishments would never have come to fruition without the support of Melissa Oh, my love, best friend, and life partner.

Abstract

The study of fake real quadratic orders is fascinating as their class group structure is similiar to real quadratic fields. Statistical data strongly agree with the heuristics of Cohen and Lenstra of real quadratics with class number one. We will investigate why this holds true as well as explore other analogues to open conjecture on real quadratic fields, such as the Ankeny-Artin-Chowla Conjecture, and present various results that mark the similiarities between real quadratic fields and fake real quadratic orders. Fake real quadratics are defined by inverting an ideal above any prime p which is split in $\mathcal{O}_K$ where $K = \mathbb{Q}(\sqrt{D})$ is an imaginary quadratic field.

Table of Contents

Chapter 1

Introduction to Fake Real Quadratic Orders

The study of quadratic fields over the centuries produced extensive results as it is the most basic of number fields over the rationals. These results are numerous, while cubic fields and number fields with higher degrees lagged in progress as they present their own challenges. However natural and tangible quadratic fields appear to be, there is still quite a lot that is unknown. For example, it is expected there are infinitely many real quadratic fields with class number 1. Although numerical data strongly suggests this should be true, there has been little to no progress on establishing this result. The Ankeny-Artin-Chowla conjecture is another open result regarding the fundamental unit of real quadratic fields. It states the fundamental unit of a real quadratic field of the form $(a + b\sqrt{D})/2$ is given, then $D \nmid b$. Extensive data have been generated in attempts to provide convincing data or discover a counterexample. Despite the fact no counterexamples have been found, it is believed by many that the conjecture is false due to the "log log" argument which will be discussed further in Section 5.1. In other words, we still have not searched far enough to find a counterexample. On the other hand, similar results for imaginary quadratic fields have been proven since they are slightly better understood. For example, it has been shown there are finitely many imaginary quadratic fields with class number 1.

Henri Cohen suggested by inverting an ideal above any prime p which is split in $\mathcal{O}_K$, where $\mathcal{O}_K$ is the ring of integers for an imaginary quadratic field, it is a Dedekind domain (which we will prove in the next section), thus allowing us to study its class group structure. We will call this algebraic structure a *fake* real quadratic order.

Cohen observed the class group structure of fake real quadratic orders have a similar structure to class groups of real quadratic fields. We investigate further the properties of the class group, the unit group, their asymptotics, and analogues to open questions of real quadratic fields. We will demonstrate these fake real quadratic orders have many similiarities to real quadratic fields, many of them evident while some less so.

1.1 Construction of Fake Real Quadratic Orders

Fix a prime number p. We take any fundamental discriminant $D < 0$ such that D is a quadratic residue modulo p. Let $K = \mathbb{Q}(\sqrt{D})$ and let $\mathcal{O}_K$ denote its ring of integers. Since $\left(\frac{D}{p}\right) = 1$, we have that $(p) = \mathfrak{p}\overline{\mathfrak{p}}$. We call the ring $\mathcal{O}_K[\mathfrak{p}^{-1}]$ a **fake real quadratic order**, and let $\mathcal{O}_{K,p}$ be the shorthand for it. The elements of $\mathcal{O}_{K,p}$ are elements $\gamma \in \mathbb{Q}(\sqrt{D})$ such that $v_{\mathfrak{q}}(\gamma) \geq 0$ for all prime ideals $\mathfrak{q}$ of $\mathcal{O}_K$, for $\mathfrak{q} \neq \mathfrak{p}$.

We note that the Galois conjugation induces a natural canonical isomorphism between $\mathcal{O}_K[\mathfrak{p}^{-1}]$ and $\mathcal{O}_K[\overline{\mathfrak{p}}^{-1}]$. Therefore, we are able to simply write $\mathcal{O}_{K,p}$. Now, $\mathcal{O}_{K,p}$ is a ring, but more specifically, it is a $\mathcal{O}_K$-algebra. We will now show that $\mathcal{O}_{K,p}$ is a Dedekind domain, where Henri Cohen laid out a detailed sketch of the proof and the author filled in the details.

Proposition 1.1. *(Cohen) The ring $\mathcal{O}_{K,p}$ is a Dedekind domain.*

Proof. We can write $\mathcal{O}_{K,p} = \bigcup_{k \geq 0} \mathfrak{p}^{-k}$. Therefore, if $\gamma \in \mathfrak{p}^{-k}$, then there is a x such that $x\mathfrak{p}^k \subset \mathcal{O}_K$, where $v_{\mathfrak{q}}(x) \geq 0$ for all prime ideals $\mathfrak{q} \neq \mathfrak{p}$ of $\mathcal{O}_K$. On the other hand, if $v_{\mathfrak{q}}(x) \geq 0$ for all prime ideals $\mathfrak{q} \neq \mathfrak{p}$, then we are able to write that $x\mathcal{O}_K = \mathfrak{p}^m \cdot \mathfrak{a}$ for some $m \in \mathbb{Z}$ and some integral ideal $\mathfrak{a}$ of $\mathcal{O}_K$, where $\mathfrak{a}$ is coprime to $\mathfrak{p}$. Then we have that $x \in \mathfrak{p}^m$. Another way to stating this is $\mathfrak{p}\mathcal{O}_{K,p} = \mathcal{O}_{K,p}$.

Now, consider the following map:

$$\varphi : \mathcal{O}_K \to \mathcal{O}_{K,p}$$

$$\mathfrak{a} \mapsto \mathfrak{a}\mathcal{O}_{K,p},$$

where $\mathfrak{a}$ is from the set of integral ideals of $\mathcal{O}_K$ coprime to $\mathfrak{p}$ mapping to the set of integral ideals of $\mathcal{O}_{K,p}$. We claim that this map is well-defined and is a bijection. We can see clearly that $\mathfrak{a}\mathcal{O}_{K,p}$ is an ideal of $\mathcal{O}_{K,p}$. Now, if $\mathfrak{a}\mathcal{O}_{K,p} = \mathfrak{b}\mathcal{O}_{K,p}$, then

$$\mathfrak{a}\mathfrak{p}^{-1} \subset \mathfrak{b}\mathcal{O}_{K,p}.$$

Thus,

$$\mathfrak{a} \subset \mathfrak{b}\mathfrak{p}\mathcal{O}_{K,p} = \mathfrak{b}\mathcal{O}_{K,p},$$

which gives us $v_\mathfrak{q}(\mathfrak{b}) \leq v_\mathfrak{q}(\mathfrak{a})$ for all prime ideals $\mathfrak{q} \neq \mathfrak{p}$. By a symmetric argument, we get that $v_\mathfrak{q}(\mathfrak{a}) \leq v_\mathfrak{q}(\mathfrak{b})$ for all prime ideals $\mathfrak{q} \neq \mathfrak{p}$. Therefore, $\mathfrak{a} = \mathfrak{b}$, since both $\mathfrak{a}$ and $\mathfrak{b}$ are coprime to $\mathfrak{p}$, showing us that φ is injective.

Now, let I be an ideal of $\mathcal{O}_{K,p}$. Define $\mathfrak{a} = I \cap \mathcal{O}_K$, which is an ideal of $\mathcal{O}_K$. Trivally, we have that

$$\mathfrak{a}\mathcal{O}_{K,p} \subset I\mathcal{O}_{K,p} \cap \mathcal{O}_{K,p} = I.$$

On the other hand, if $x \in I$, there is a $k \geq 0$ such that $x \in \mathfrak{p}^{-k}$, which means $x\mathfrak{p}^k \subset \mathcal{O}_K$, and $x\mathfrak{p}^k \subset I$, since $x \in I$. Therefore,

$$x \in \mathfrak{a}\mathfrak{p}^{-k} \subset \mathfrak{a}\mathcal{O}_{K,p},$$

giving us that $I = \mathfrak{a}\mathcal{O}_{K,p}$. So, if $\mathfrak{a} = \mathfrak{p}^m\mathfrak{a}'$, where $\mathfrak{a}'$ is coprime to $\mathfrak{p}$, then

$$\mathfrak{a}\mathcal{O}_{K,p} = \mathfrak{a}'\mathcal{O}_{K,p}.$$

This gives us that φ is surjective, thus proving our claim. Since this is a bijection, the generators of $\mathfrak{a}\mathcal{O}_{K,p}$ are the same as $\mathfrak{a}$, thus all ideals of $\mathfrak{a}\mathcal{O}_{K,p}$ are finitely generated.

Next, we will show every nonprime ideal is maximal. We claim the mapping

$$\mathfrak{a}\mathcal{O}_{K,p} \cap \mathcal{O}_K = \mathfrak{a}$$

is a bijection, where $\mathfrak{a}$ is an ideal of $\mathcal{O}_K$ that is coprime to $\mathfrak{p}$. It is clear that this map is injective since

$$\mathfrak{a}\mathcal{O}_{K,p} \cap \mathcal{O}_K = \mathfrak{a},$$

by the above argument.

To see surjectivity, consider $x \in \mathcal{O}_{K,p}$. If $x \in \mathcal{O}_K$, then it is the image of itself. If $x \notin \mathcal{O}_K$, then we choose the smallest $k > 0$ such that $x\mathfrak{p}^k \subset \mathcal{O}_K$. Since $\mathfrak{a}$ is coprime to $\mathfrak{p}^k$, then there exists $a \in \mathfrak{a}$ and $u \in \mathfrak{p}^k$ such that $a + u = 1$, so that $x = ax + ux$, where $ax \in \mathfrak{a}\mathcal{O}_{K,p}$ and $ux \in \mathcal{O}_K$. Therefore, x is the image of ux modulo $\mathfrak{a}\mathcal{O}_{K,p}$.

From this isomorphism, we see that $\mathfrak{a}\mathcal{O}_{K,p}$ is a prime ideal if and only if $\mathfrak{a}$ is a prime ideal different from $\mathfrak{p}$ of $\mathcal{O}_K$. Similarly, we can say that $\mathfrak{a}\mathcal{O}_{K,p}$ is a maximal ideal if and only if $\mathfrak{a}$ is a maximal ideal of $\mathcal{O}_K$, again not being $\mathfrak{p}$. This means that the prime ideals of $\mathcal{O}_{K,p}$ are either 0 or $\mathfrak{q}\mathcal{O}_{K,p}$ for all prime ideals $\mathfrak{q} \neq \mathfrak{p}$ of $\mathcal{O}_K$. Thus, every nonzero prime ideal is maximal.

Finally, we need to show that $\mathcal{O}_{K,p}$ is integrally closed. Consider the following monic polynomial of degree n

$$P(X) = X^n + a_{n-1}X^{n-1} + \ldots + a_1X + a_0,$$

where $a_i \in \mathcal{O}_{K,p}$, and there is a $x \in K$ such that $P(x) = 0$. If $\mathfrak{q} \neq \mathfrak{p}$ is a prime ideal of $\mathcal{O}_K$, then

$$v_\mathfrak{q}(a_i \cdot x^i) \geq i \cdot v_\mathfrak{q}(x),$$

and

$$v_\mathfrak{q}(x^n) = n \cdot v_\mathfrak{q}(x).$$

So, if $v_\mathfrak{q}(x) < 0$, then

$$v_\mathfrak{q}(a_i \cdot x^i) > n \cdot v_\mathfrak{q}(x),$$

which means that the $\mathfrak{q}$-valuation for the lead coefficient will not be able to cancel for the rest of the coefficients giving us $P(x) \neq 0$. This is a contradiction.

Therefore, it must be $v_\mathfrak{q}(x) \geq 0$, giving us $x \in \mathcal{O}_{K,p}$, which shows $\mathcal{O}_{K,p}$ is integrally closed. Since all the conditions have been satisifed, this shows that $\mathcal{O}_{K,p}$ is a Dedekind domain. $\square$

1.2 The Unit and Class Groups of Fake Real Quadratics

Since $\mathcal{O}_{K,p}$ is a Dedekind domain, we can study its unit and class groups. In this section, we build up the structure of both the unit and class group and explicity define what elements in these respective groups look like. All the results in this sections are attributed to Henri Cohen. Note that this establishes an analogy with unit and class groups of real quadratic fields.

Proposition 1.2. *(Cohen) Let $U_{K,p}$ denote the unit group of $\mathcal{O}_{K,p}$. Then $U_{K,p} = \mu_K \times \varepsilon^{\mathbb{Z}}$, where μ_K is the group of roots of unity in K, and the fundamental unit ε is the generator of the principal ideal $\mathfrak{p}^{o(\mathfrak{p})}$, where $o(\mathfrak{p})$ is the order of the ideal $\mathfrak{p}$ in the ideal class group Cl_K.*

Proof. We have that $\gamma \in U_{K,p}$ if and only if $v_{\mathfrak{q}}(\gamma) \geq 0$ and $v_{\mathfrak{q}}(1/\gamma) \geq 0$ for all prime ideals $\mathfrak{q} \neq \mathfrak{p}$. Therefore, $\gamma \in U_{K,p}$ if and only if $v_{\mathfrak{q}}(\gamma) = 0$ for all $\mathfrak{q} \neq \mathfrak{p}$. In order for $v_{\mathfrak{q}}(\gamma) = 0$, it must be that $\gamma\mathcal{O}_K = \mathfrak{p}^k$ for some $k \in \mathbb{Z}$. By the definition of $o(\mathfrak{p})$, the order of the ideal $\mathfrak{p}$ in the ideal class group Cl_K, this is only possible if and only if $o(\mathfrak{p})|k$.

This means that $\gamma\mathcal{O}_K = (\varepsilon\mathcal{O}_K)^{k/o(\mathfrak{p})}$. In other words, $\gamma = \eta\varepsilon^{k/o(\mathfrak{p})}$, for η being some unit in $\mathcal{O}_K$. Since the only group of units in $\mathcal{O}_K$ is μ_K, the roots of unity, since K is an imaginary quadratic field, we get the statement that $\gamma \in U_{K,p}$ if and only if $\gamma = \eta\varepsilon^{k/o(\mathfrak{p})}$. Therefore, $U_{K,p} = \mu_K \times \varepsilon^{\mathbb{Z}}$. □

Next, we explore what the ideal class group of $\mathcal{O}_{K,p}$ looks like and define the class number.

Proposition 1.3. *(Cohen) The class group $\mathrm{Cl}_{K,p}$ of $\mathcal{O}_{K,p}$ is canonicallly isomorphic to $\mathrm{Cl}_K / \langle \mathfrak{p} \rangle$, where Cl_K is the ideal class group of K and $\langle \mathfrak{p} \rangle$ is the cyclic group generated by the ideal $\mathfrak{p}$. Thus, $h_{K,p} = |\mathrm{Cl}_{K,p}| = h_K/o(\mathfrak{p})$.*

Proof. Consider the map

$$\varphi : \mathrm{Cl}_K \to \mathrm{Cl}_{K,p}$$
$$[\mathfrak{a}] \mapsto [\mathfrak{a}\mathcal{O}_{K,p}],$$

where $[\mathfrak{a}]$ is an ideal class in Cl_K and $[\mathfrak{a}\mathcal{O}_{K,p}]$ is an ideal class of $\mathrm{Cl}_{K,p}$. This map is a well-defined group homomorphism. We have from the proof of Proposition 1.1 that ideals of $\mathcal{O}_{K,p}$ have the form $\mathfrak{a}\mathcal{O}_{K,p}$, it is clear to see that this map is surjective.

We have already seen that the map $\mathfrak{a} \mapsto \mathfrak{a}\mathcal{O}_{K,p}$ is an injective map for all ideals coprime to $\mathfrak{p}$. Thus, if $\mathfrak{a}\mathcal{O}_{K,p} = x\mathcal{O}_{K,p}$ for some $x \in \mathcal{O}_K$. The $\mathfrak{a} = x\mathfrak{p}^k$ for some $k \in \mathbb{Z}$. This means that the kernel of the map φ is the cyclic group generated by the ideal $\mathfrak{a}$ that minimizes k. Thus, by the Fundamental Isomorphism Theorem,

$$\mathcal{O}_K/\langle\mathfrak{p}\rangle \cong \mathcal{O}_{K,p}. \tag{1.1}$$

The fact that $h_{K,p} = h_K/o(\mathfrak{p})$ follows immediately from this isomorphism. □

We would like a specific representation of elements inside of $\mathcal{O}_{K,p}$. The following lemma gives us representations of these elements. For simplicity, we will assume that $p \neq 2$ from here on. Since $\left(\frac{D}{p}\right) = 1$, there exists $a \in \mathbb{Z}$ such that $D \equiv a^2 \pmod{p}$. Fixing one of the two possible values of a modulo p is equivalent to choosing between $\mathfrak{p}$ and $\overline{\mathfrak{p}}$. Therefore, without loss of generality we will let

$$\mathfrak{p} = p\mathbb{Z} + (a + \sqrt{D})\mathbb{Z}.$$

Lemma 1.4. *(Cohen) Let $\alpha = x + y\sqrt{D} \in \mathcal{O}_K$, with $x, y \in (\frac{1}{2})\mathbb{Z}$ and $\gcd(x, y, p) = 1$. Then for $k > 0$, we have that $\alpha \in \mathfrak{p}^k$ if and only if $\mathfrak{p}^k | x^2 - D \cdot y^2$ and $x \equiv ay \pmod{p}$.*

Proof. To see sufficiency, let $\alpha \in \mathfrak{p}^k$. Then $N(\mathfrak{p})^k | N(\alpha)$. Since $\alpha = x + y\sqrt{D}$, we have $N(\alpha) = x^2 - D \cdot y^2$. Therefore,

$$p^k | x^2 - D \cdot y^2.$$

If $k > 0$, then $\alpha \in \mathfrak{p}$, so

$$\begin{aligned} \alpha &= pu + (a + \sqrt{D})v \\ &= (pu + av) + v \cdot \sqrt{D}. \end{aligned}$$

So, $x = pu + av$ and $y = v$. Thus,

$$x = av + pu \equiv av = ay \pmod{p}.$$

Thus, $x \equiv ay \pmod{p}$.

For necessity, assume that $p^k | x^2 - D \cdot y^2$ and $x \equiv ay \pmod{p}$. Since $k > 0$ so $p | N(\alpha)$. Then either $\alpha \in \mathfrak{p}$ or $\alpha \in \overline{\mathfrak{p}}$. But if

$$\alpha \in \overline{\mathfrak{p}} = p\mathbb{Z} + (a - \sqrt{D})\mathbb{Z},$$

then

$$\alpha = pu + (a - \sqrt{D})v,$$

for $u, v \in \mathbb{Z}$. This gives us $x = pu + av$ and $y = -v$. So

$$x = pu + av \equiv av = -ay \pmod{p}.$$

Therefore $2x \equiv D \pmod{p}$. Since we assume that $p \neq 2$, this gives us $p | x$. Now, using the fact that $\left(\frac{D}{p}\right) = 1$ and $p | (x^2 - Dy^2)$ implies that $p | y$, but this contradicts our assumption the $\gcd(x, y, p) = 1$.

Therefore, $\alpha \in \mathfrak{p}$ and $v_{\overline{p}}(\alpha) = 0$, so $v_{\mathfrak{p}}(N(\alpha)) = v_{\mathfrak{p}}(\alpha)$ giving us the fact that $\alpha \in \mathfrak{p}^k$. □

This implies the following corollary regarding the representation of elements in $\mathcal{O}_{K,p}$.

Corollary 1.5. *(Cohen) The elements of $\mathcal{O}_{K,p}$ can be written in a unique way. Specifically if $\gamma \in \mathcal{O}_{K,p}$, then*

$$\gamma = \frac{x + y \cdot \sqrt{D}}{p^k},$$

where $k \in \mathbb{Z}$, $x, y \in (\frac{1}{2})\mathbb{Z}$, $\gcd(x, y, p) = 1$, *and either* $k \leq 0$ *(i.e.* $\gamma \in \mathcal{O}_K$*) or* $k > 0$, *which must mean that* $p^k | x^2 - D \cdot y^2$ *and* $x \equiv -ay \pmod{p}$.

Proof. Since $\mathfrak{p}^k \gamma \subseteq \mathcal{O}_K$ for some $k \in \mathbb{Z}$, we have that $\mathfrak{p}^k \gamma \in \mathcal{O}_K$. Therefore,

$$\gamma = \frac{x + y\sqrt{D}}{p^k}.$$

Since we can always divide out the largest power of p from $\gcd(x, y)$, we can further assume that $\gcd(x, y, p) = 1$. Therefore, this representation will be unique.

On the other hand, if $k \leq 0$, then $\gamma \in \mathcal{O}_K \subset \mathcal{O}_{K,p}$, so there are no other conditions. If $k > 0$, then using the condition that $\gamma \in \mathcal{O}_{K,p}$, this is equivalent to

$$v_{\overline{\mathfrak{p}}}(x + y\sqrt{D}) \geq k.$$

By Lemma 1.4 applied to $\overline{\mathfrak{p}}$, this statement is equivalent to $p^k | x^2 - D \cdot y^2$ and $x \equiv -ay \pmod{p}$. □

We see here that the unit and class groups of $\mathcal{O}_{K,p}$ in this situation is analogous to the real quadratic field case. With this construction, we see that fake real quadratic orders have unit groups of rank one. Also, with the class group having parallels with real quadratics, we seek to find an asymptotic formula involving the class number as the discriminant varies. Therefore, it would be interesting to test out open conjectures that exist regarding real quadratics in this setting. We will discuss two open conjectures and their analogues to fake real quadratics in Section 4 and 5.

1.3 Chebotarev Density Theorem

Before going into the class number formula and giving some asymptotics, we need to discuss the Chebotarev Density Theorem. The Chebotarev density theorem describes the statistical behavior of splitting of primes in a given Galois extension K of Q of rational numbers. Let K be a quadratic field, and Cl_K be the class group of K. Let H be the Hilbert class field of the quadratic field K, which is the maximal unramified abelian extension of K. Let $n_H = [H : \mathbb{Q}]$. By Class Field Theory, $\mathrm{Gal}(H/K) \simeq \mathrm{Cl}_K$. Let C be an element of $\mathrm{Gal}(H/K)$. Let π_C count the number of primes in $\mathbb{Q}$ that split in K such that $N_{K/\mathbb{Q}}(\mathfrak{p}) < X$ and $\mathfrak{p}$ is represented by the element C in the class group Cl_K.

$$\pi_C(X) = \#\{\mathfrak{p} : \mathfrak{p} \text{ prime, } \mathfrak{p} \text{ is represented by } C, \text{ and } N_{K/\mathbb{Q}}(\mathfrak{p}) \leq X\}.$$

The Chebotarev density theorem asserts that

$$\pi_C(X) \sim \frac{1}{|\,\mathrm{Cl}_K\,|} \cdot \mathrm{Li}(X) = \frac{1}{h(D)} \cdot \mathrm{Li}(X), \quad \text{as } X \to \infty$$

where $\mathrm{Li}(X)$ is the familiar logarithmic integral

$$\mathrm{Li}(X) = \int_2^X \frac{dt}{\log t} \sim \frac{X}{\log X}, \quad \text{as } X \to \infty.$$

In 1977, Lagarias and Odlyzko [17] gave two versions of the Chebotarev density theorem, both having effective and computable error bounds with respect to X, $h(D)$, D, and $1/|\,\mathrm{Cl}_K\,|$. The unconditional bound is given by the following theorem.

Theorem 1.6. *(Lagarias, Odlyzko) There are effective computable constants c_1 and c_2 such that if*

$$x \geq \exp(10 \cdot h(D) \log^2(|D|)),$$

then

$$\left|\pi_C(X) - \frac{1}{|\,\mathrm{Cl}_K\,|} \cdot \mathrm{Li}(X)\right| \leq \frac{1}{|\,\mathrm{Cl}(D)|} \cdot \mathrm{Li}(X^{\beta_0}) + c_1 \cdot X \cdot \exp(-c_2 \cdot h(D)^{-1/2} \cdot (\log X)^{1/2}), \tag{1.2}$$

where β_0 is a Siegel zero. Note that if $h(D) = 1$, there is no zero in this range.

In order to make sure that Theorem 1.6 has an error term that only depends on X, $h(D)$, D, and $1/|\operatorname{Cl}_K|$, we need a uniform bound for β_0. We give the following theorem for completeness of the unconditional effective bound.

Theorem 1.7. *(Lagarias, Odlyzko) Let the notation be the same as the previous theorem. If the Hilbert class field H of the number field K is normal over $\mathbb{Q}$, then $m_H = 4$. Let $m_H = 16$ if there is a sequence of fields*

$$\mathbb{Q} = K_0 \subset K_1 \subset \ldots \subset K_r = H,$$

which each field being normal over the preceding one. Otherwise, let $m_H = 4 \cdot h(D)!$. Then there is a computable and effective constant c_3 such that

$$\beta_0 < \max[1 - (m_H \cdot \log|D|)^{-1}, 1 - (c_3 \cdot |D|^{1/h(D)})^{-1}].$$

Therefore, we have an unconditional, effective error bound to the Chebotarev density theorem. We also have a version of Chebotarev density theorem that assumes GRH.

Theorem 1.8. *(Lagarias, Odlyzko) Assuming the Generalized Riemann Hypothesis holds true, there is an effective, computable positive absolute constant c_4 such that for every $x > 2$,*

$$\left|\pi_C(K) - \frac{1}{|\operatorname{Cl}_K|} \cdot \operatorname{Li}(X)\right| \leq c_4 \left(\frac{1}{|\operatorname{Cl}_k|} \cdot X^{1/2} \cdot \log(|D| \cdot X^{h(D)}) + \log|D|\right). \quad (1.3)$$

As the study of fake real quadratic orders require the parameters p and D, the distribution of primes is important. Thus the above two formulas are significant in our studies as we desire to study the behavior of the class number as p varies with D.

1.4 Class Number Formula

One of the common themes in number theory is special values of L-functions encode arithmetic information. The Dirichlet class number formula is a famous example of using L-functions to compute the class number of a number field. For imaginary quadratic fields, this is

$$h(D) = \frac{\omega \cdot \sqrt{|D|} \cdot L(1, \chi_D)}{2\pi}, \tag{1.4}$$

where ω denote the automorphs of binary quadratic forms of discriminant D or the number of units in $\mathbb{Q}(\sqrt{D})$. Specifically,

$$\omega = \begin{cases} 2 & : \quad D < -4 \\ 4 & : \quad D = -4 \\ 6 & : \quad D = -3. \end{cases} \tag{1.5}$$

For real quadratic fields, the Dirichlet class number formula is

$$h(D) = \frac{\sqrt{D} \cdot L(1, \chi_D)}{\log \varepsilon}, \tag{1.6}$$

where ε is the fundamental unit for $\mathbb{Q}(\sqrt{D})$.

In this section, we will prove the analogue to the Dirichlet class number formula for fake real quadratic orders. This will allow us to investigate asymptotics of class numbers for fake real quadratics and compare similarities to real quadratic fields. In this section, we will denote $\mathcal{O}_{K,p}$ as $\mathcal{O}_{D,p}$, where $K = \mathbb{Q}(\sqrt{D})$. This will allow us to define asymptotics in relation to D.

Theorem 1.9. *(O.) The class number formula for fake real quadratic orders $\mathcal{O}_{D,p}$ is given by*

$$h(D, p) = \frac{\omega \cdot \sqrt{|D|} \cdot L(1, \chi_D)}{2\pi \cdot \log_p N(\varepsilon_{D,p})},$$

where ω is the same as defined in (4.3) *and $\varepsilon_{D,p}$ is the fundamental unit of $\mathcal{O}_{D,p}^{\times}$.*

Proof. The class number formula for imaginary quadratic fields is given by

$$h(D) = \frac{\omega \cdot \sqrt{|D|} \cdot L(1, \chi_D)}{2\pi}.$$

By Proposition 1.3, we know that the class group is canonically isomorphic to $\mathrm{Cl}_D / \langle \mathfrak{p} \rangle$ and $h(D, p) = h(D)/o(\mathfrak{p})$. Therefore,

$$h(D, p) = \frac{h(D)}{o(\mathfrak{p})} = \frac{\omega \cdot \sqrt{|D|} \cdot L(1, \chi_D)}{2\pi} \cdot \frac{1}{o(\mathfrak{p})}.$$

We use the classical technique of multiplying by 1, which in this case we will multiply by $\frac{\log p}{\log p}$ which gives us

$$\frac{\omega \cdot \sqrt{|D|} \cdot L(1, \chi_D)}{2\pi} \cdot \frac{1}{o(\mathfrak{p})} \cdot \frac{\log p}{\log p} = \frac{\omega \cdot \sqrt{|D|} \cdot L(1, \chi_D)}{2\pi} \cdot \frac{\log p}{\log p^{o(\mathfrak{p})}}.$$

Since $\varepsilon_{D,p}$, the fundamental unit of $\mathcal{O}_{D,p}$, is a generator for the principal ideal $\mathfrak{p}^{o(\mathfrak{p})}$, we know $\log_p \mathfrak{p}^{o(\mathfrak{p})} = N(\varepsilon_{D,p})$. Therefore,

$$\frac{\omega \cdot \sqrt{|D|} \cdot L(1, \chi_D) \cdot \log p}{2\pi \cdot \log N(\varepsilon_{D,p})} = \frac{\omega \cdot \sqrt{|D|} \cdot L(1, \chi_D)}{2\pi \cdot \log_p N(\varepsilon_{D,p})}.$$

This gives us the desired class number formula for fake real quadratics. □

We would like to investigate some asymptotics on $h(D, p)$. We need the following lemma before we get to our theorem.

Lemma 1.10. *Let $D < 0$ be a fundamental discriminant. Let $\mathrm{Cl}(D)$ denote the class group of $\mathbb{Q}(\sqrt{D})$. Then*

$$\sum_{C \in [Cl(D)]} \frac{1}{\mathrm{ord}(C)} \leq \frac{h(D) + 1}{2}.$$

where $\mathrm{ord}(C)$ is the order of class C in the ideal class group $\mathrm{Cl}(D)$.

Proof. There are only finitely many class groups of order 1 for negative fundamental discriminants. Therefore, the worst possible situation is the ideal class group is isomorphic to the product of cyclic groups of order 2. In other words, all elements

have order 2 except the identity. Therefore, given a fundamental discriminant of an imaginary quadratic field, if the class group is isomorphic to the product of cyclic groups of order 2, then the number of elements of order 2 is $h(D)-1$. Therefore, the sum of $\frac{1}{\mathrm{ord}(C)}$ over all elements of order 2 is

$$(h(D)-1)\cdot\frac{1}{2}.$$

By accounting for the identity element, we have that the upper bound over all elements in $\mathrm{Cl}(D)$ is

$$(h(D)-1)\cdot\frac{1}{2}+1=\frac{h(D)+1}{2}.$$

□

It is possible to improve this bound given here by looking at the bound on 2-torsion elements from Cox on Gauss genus theory [9] and bound on 3-torsion elements from Ellenberg and Venkatesh's *Reflection Principles* [12].

Theorem 1.11. *(O.) Let*

$$\sum_{-X<D<0}\sum_{\substack{(p)=\mathfrak{p}\overline{\mathfrak{p}}\\ N_{K/\mathbb{Q}}(\mathfrak{p})<Y}} h(D,p)\leq \mathrm{Li}(Y)\cdot\left(\frac{2\pi}{21\zeta(3)}\cdot X^{3/2}+\frac{X}{\pi^2}\right)+O\left(\frac{X^{1/2}\cdot Y}{\log Y}\right).$$

where the first summation ranges over all $-X<D<0$ fundamental discriminants such that $\left(\frac{D}{p}\right)=1$ and second summation is over all primes that split in K over $\mathbb{Q}$ such that $N_{K/\mathbb{Q}}(\mathfrak{p})<Y$, where both X and Y are large.

Proof. Since $h(D,p)=h(D)/o(\mathfrak{p})=h(D)/\operatorname{ord}_{\mathrm{Cl}(D)}(\mathfrak{p})$,

$$\begin{aligned}\sum_{-X<D<0}\sum_{\substack{(p)=\mathfrak{p}\overline{\mathfrak{p}}\\ N_{K/\mathbb{Q}}(\mathfrak{p})<Y}} h(D,p) &= \sum_{-X<D<0} h(D)\sum_{\substack{(p)=\mathfrak{p}\overline{\mathfrak{p}}\\ N_{K/\mathbb{Q}}(\mathfrak{p})<Y}}\frac{1}{o(\mathfrak{p})}\\ &= \sum_{-X<D<0}\cdot\sum_{C\in[\mathrm{Cl}(D)]}\frac{1}{\mathrm{ord}_{\mathrm{Cl}(D)}(C)}\cdot\sum_{\substack{N_{K/\mathbb{Q}}(\mathfrak{p})<Y\\ \mathfrak{p}\text{ rep by }C}}1.\end{aligned}$$

By the Chebotarev Density Theorem, we know the primes that split in K are evenly represented in the class group. More specifically,

$$\sum_{\substack{N_{K/\mathbb{Q}}(\mathfrak{p})<Y \\ \mathfrak{p} \text{ rep by } C}} 1 \sim \frac{1}{h(D)} \cdot \mathrm{Li}(Y).$$

Therefore, using this fact and Lemma 1.10, we get

$$\sum_{-X<D<0} h(D) \sum_{C \in [\mathrm{Cl}(D)]} \frac{1}{h(D)} \cdot \mathrm{Li}(Y) \leq \mathrm{Li}(Y) \sum_{-X<D<0} \frac{h(D)+1}{2}.$$

This simplifies to

$$\frac{\mathrm{Li}(Y)}{2} \cdot \sum_{-X<D<0} h(D) + \frac{\mathrm{Li}(Y)}{2} \cdot \sum_{-X<D<0} 1.$$

The average value for the class number for imaginary quadratic fields, given by Chamizo and Ubis [5], is

$$\sum_{-X<D<0} h(D) \sim \frac{4\pi}{21\zeta(3)} X^{3/2} - \frac{2}{\pi^2} X.$$

Furthermore, the number of negative fundamental discriminants, given by Bhargava and Pomerance [21], is

$$\frac{3X}{\pi^2} + o(X^{1/2}).$$

Using the above two bounds, we get

$$\begin{aligned}
&\frac{\mathrm{Li}(Y)}{2} \cdot \sum_{-X<D<0} h(D) + \frac{\mathrm{Li}(Y)}{2} \cdot \sum_{-X<D<0} 1 \\
=&\frac{\mathrm{Li}(Y)}{2} \cdot \left(\frac{4\pi}{21\zeta(3)} X^{3/2} - \frac{2}{\pi^2} X \right) + \frac{\mathrm{Li}(Y)}{2} \cdot \left(\frac{3X}{\pi^2} + O(X^{1/2}) \right) \\
=&\mathrm{Li}(Y) \cdot \left(\frac{2\pi}{21\zeta(3)} X^{3/2} - \frac{X}{\pi^2} \right) + O\left(\frac{X^{1/2} \cdot Y}{\log Y} \right).
\end{aligned}$$

□

Chapter 2

Fundamental Unit for Fake Real Quadratic Orders

The fundamental unit is a generator modulo the roots of unity for the unit group of the ring of integers of a number field. For real quadratic fields $\mathbb{Q}(\sqrt{D})$ where D is a fundamental discriminant, then the fundamental unit is

$$\varepsilon = \frac{a + b\sqrt{D}}{2}$$

where the pair (a, b) is the smallest solution to the Pell equation

$$x^2 - Dy^2 = \pm 4.$$

The classical method to compute the fundamental unit for real quadratic fields involves the computation of the continued fractions expansion for $\sqrt{D}$. For fake real quadratic orders, there is no analogous method has yet to be found. Therefore, in the subsequent sections, we will discuss two methods to compute the fundamental unit for fake real quadratic orders. The algorithms are implemented in SAGE [24] and the code in their entirety can be found in the Appendices.

2.1 Method 1: Brute Force

The first method to compute the fundamental unit requires the knowledge of the class number $h(D)$ for the given imaginary quadratic field and pre-computation of the factors of the class number $h(D) = h = \prod_{i=0}^{m} p_i^{e_i}$, where k is the number of distinct prime divisors.

Algorithm 2.1. *Inputs: p prime, $D < 0$ fundamental discriminant, $\left(\frac{D}{p}\right) = 1$.*
Output: $\varepsilon_{K,p}$, fundmental unit for $\mathcal{O}_{K,p}$, where $K = \mathbb{Q}(\sqrt{D})$.
Assume: $h(D) = h$, the class number for K has been pre-computed, and the factorization $h = \prod_{i=0}^{m} p_i^{e_i}$ has been found.

1. *Factor $(p) = \mathfrak{p}\overline{\mathfrak{p}}$.*
2. *List all the possible divisors in ascending order of h. Let $L = \{d_i\}_{i=1}^{\tau(h)}$, where d_i is a divisor of h.*
3. *Compute $\mathfrak{p}^{d_i}$, starting with $i = 0$*
4. *If $\mathfrak{p}^{d_i}$ is principal, then d_i is the order of $\mathfrak{p}$. Return $\mathfrak{p}^{d_i}$ as the fundamental unit. If it is not principal, then $i = i + 1$, and return to Step 3.*

Since we assume for this algorithm the value $h(D)$ has been computed and factored already, this algorithm has low complexity. In practice, the computing of the class number and factoring it is non-trivial and is a different computational problem. Hence, this algorithm is only practical when $h(D)$ is small or when $|D|$ is small. We note that this method is not the most efficient method to compute the order for $\mathfrak{p}$. It would be quicker to look at $L = \{h/p_i^k\}_{k=1}^{e^i}$ to get the order. If the class number is unknown, the method discussed in the following section may be a better alternative.

2.2 Method 2: Binary Quadratic Forms

The following method employs the use of binary quadratic forms. We start with a brief summary of binary quadratic forms before discussing the algorithm, but the author recommends Buell's Binary Quadratic Forms: Classical Theory and Modern Computations [4] for further reading on this topic. Let $Q(x, y) = Ax^2 + Bxy + Cy^2$ be a binary quadratic form with discriminant $D = B^2 - 4AC$. We use the shorthand $Q = (A, B, C)$ for the binary quadratic form $Q(x, y) = Ax^2 + Bxy + Cy^2$. A principal

ideal has a binary quadratic form that is $\mathrm{SL}_2(\mathbb{Z})$ equivalent to a principal binary quadratic form.

Definition 2.2. Let D be a discriminant. Then the *principal binary quadratic form* of discriminant D is

$$\begin{cases} \left(1, 0, -\dfrac{D}{4}\right) & : \quad D \equiv 0 \pmod 4 \\ \left(1, 1, -\dfrac{D-1}{4}\right) & : \quad D \equiv 1 \pmod 4. \end{cases} \tag{2.1}$$

Let $M = \begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix} \in \mathrm{SL}_2(\mathbb{Z})$, where $\alpha\delta - \beta\gamma = +1$. Let $Q = (A, B, C)$ be a binary quadratic form with discriminant D. We say M acts on Q on the right by

$$Q \circ M = Q(\alpha X + \beta Y, \gamma X + \delta Y).$$

Another way to state is letting Q be represented by the matrix

$$\begin{bmatrix} A & \frac{B}{2} \\ \frac{B}{2} & C \end{bmatrix}.$$

Then the matrix M acts on Q by $M^{-1}QM = S$, where S is an imaginary binary quadratic form with discriminant D. We say Q and Q' are equivalent forms when there is $M \in \mathrm{SL}_2(\mathbb{Z})$ with $Q \circ M = Q'$.

Before we outline the steps to compute the fundamental unit, we define *powering up* a binary quadratic form as repeatedly using Gauss's composition on a form Q, which is well-defined up to equivalence. For example, we say Q powered up 3 times as $Q^3 = Q \circ Q \circ Q$.

Algorithm 2.3. *Inputs: p prime, D fundamental discriminant.*
Output: $\varepsilon_{K,p}$, fundmental unit for $\mathcal{O}_{K,p}$, where $K = \mathbb{Q}(\sqrt{D})$.

1. *Compute $\left(\frac{D}{p}\right)$. If $\left(\frac{D}{p}\right) \neq 1$, then stop. Otherwise, continue to Step 2.*

2. *Since* $\left(\frac{D}{p}\right) = 1$*, then there exists a binary quadratic form* $Q = (p, b, c)$ *such that* $b^2 - 4pc = D$*. [9] In order to find this form, we first compute the* b *coefficient, which is the solution* $x^2 \equiv D \pmod 4$ *such that*

$$c = \frac{b^2 - D}{4p} \in \mathbb{Z}.$$

3. *Starting with* $i = 1$*, compute* Q^i*.*

4. *Check if* Q^i *is equivalent to the principal form of discriminant* D*. If not, then* $i = i + 1$ *and return to Step 3. If* Q^i *is principal, then let* $m = i$ *and go to the next step.*

5. *Let* $Q^m = (p^m, b', c')$ *which has been verified in Step 4, is equivalent to the principal form* $P = (a, b, c)$*. Compute the solution* (x, y) *such that*

$$Ax^2 + Bxy + Cy^2 = p^m.$$

6. *Complete the square to retrieve the fundamental unit. In other words,*

$$(x + y\sqrt{D})(x - y\sqrt{D}) = p^m.$$

This algorithm could be made more efficient during Steps 3 and 4. Rather than powering up with increments of a single power in each step, we can employ Shanks' baby-step-giant-step approach [22] to reduce the number of multiplications. Since the reduced binary quadratic forms of a given negative discriminant form a group, Shanks' baby-step-giant-step approach uses a meet-in-the-middle method to find the principal form. The time-memory trade-off can be performed with $O(\sqrt{n})$ group multiplications rather than the brute force's $O(n)$ group multiplications, where n is the size of the group. The complexity of the algorithm is discussed in further depth in the following section.

2.3 Algorithm Complexity

Algorithm complexity quantifies the amount of time it takes for an algorithm to run as a function of the input. The time complexity is given in big O notation, which mean the complexity is described asymptotically, i.e. as the input size goes to infinity. For example, an algorithm runs on linear time has $O(n)$ where n is the input, and $O(2^n)$ is called exponential time. An algorithm is said to be **polynomial time** if the running time has an upper bound with a polynomial expression, i.e. $O(n^k)$. We note that the Euclidean algorithm for computing the greatest common divisor of two integers runs in polynomial time. This is polynomial with respect to the binary representation of integer n and m. The inputs n and m have binary representation of roughly $\log n + \log m$. The Euclidean algorithm runs in $O((\log n + \log m)^2)$, which is polynomial with respect to $\log n + \log m$.

We will discuss the algorithm complexity of recovering the fundamental unit through the use of binary quadratic forms. To see the SAGE program, see Appendix A. Let $D < 0$ be a fundamental discriminant and p be a given fixed prime. The first step is to verify $\left(\frac{D}{p}\right) = 1$, which is equivalent to the complexity of the Euclidean algorithm.

The next step is to find the binary quadratic form of discriminant D such that $a = p$, or in other words, $Q = (p, b, c)$ where $b^2 - 4pc = D$. This step is costliest since it requires calculating a modular square root, or in other words find b such that

$$b^2 \equiv D \pmod{p}. \tag{2.2}$$

The complexity of finding such a modular square root is discussed in Section 2.4. In section 2.4, we will explain that by assuming GRH, we will have polynomial time.

In step 3, we power up Q by repeatedly using Gauss's composition law. The complexity for composing two forms is equivalent to using the Euclidean algorithm twice. It costs one Euclidean for getting the form, while it costs another Euclidean for

reducing the form and checking to see if the power is principal. Therefore, this step can be done in polynomial time with respect to $|D|$. We would like to iterate the use of Shanks' little-step-giant-step algorithm can reduce the number of multiplications necessary to reach the principal form by generating the powers of binary quadratic form Q in a more efficient, organized manner.

Assume that the order of the binary quadratic form Q in the ideal class group Cl_K is k. Then after step 3, we have the binary quadratic form $Q^k = (p^k, B, C)$ which reduces to the principal form (A', B', C'). Using an extended Euclidean algorithm computation, we get the representation of p^k by the principal form (i.e. find (x, y) such that $A'x^2 + B'xy + C'y^2 = p^k$.) We complete the square and factor to recover the fundamental unit.

In the worst case, our algorithm will cost us 1 expensive computation of the square root modulo p and $2\sqrt{|D|}$ Euclidean algorithm computations. If we assume GRH, our computation of the modular square root can be done in polynomial time, giving us the complexity to recover the fundamental unit in polynomial time with respect to $\sqrt{|D|}$. In the following section, we will discuss the expensive calculation of the modular square root and show examples of how it can be used.

2.4 Computation of Modular Square Roots

As stated in the previous section, the most computationally intensive step of computing the fundamental unit of a fake real quadratic order is calculating a modular square root, or given integers n and m, find a solution for the following equation:

$$x^2 \equiv n \pmod{m}.$$

Since the n value is our fundamental discriminant D and $m = p$ is prime, we are trying to find solutions to $x^2 \equiv D \pmod{p}$. One method to find such a square root modulo a prime is to use the Tonelli-Shanks algorithm [23]. The algorithm solves

the following problem: Given an odd prime p and an integer D such that $\left(\frac{D}{p}\right) = 1$, recover a solution to

$$x^2 \equiv D \pmod{p}.$$

Algorithm 2.4.

Step 1: Let M be an integer greater than or equal to 0 such that $2^M || (p-1)$, and let

$$p - 1 = R \cdot 2^M,$$

where $R \equiv 1 \pmod{2}$. If $M = 1$, then $p \equiv 3 \pmod{4}$. In this case, we are able to compute directly the solutions to $x^2 \equiv D \pmod{p}$ by directly calculating

$$R = \pm D^{(p+1)/4} \mod p.$$

Step 2: In this step, we need a quadratic nonresidue element modulo p. In other words, we want to compute some z such that $\left(\frac{z}{p}\right) = -1$. Set

$$C \equiv z^R \pmod{p}.$$

Step 3: Set

$$\begin{aligned} T &\equiv D^{(R+1)/2} \pmod{p}, \\ P &\equiv D^R \pmod{p}, \\ U &= M. \end{aligned}$$

Step 4: Iterate over the following loop:

1. *If $P \equiv 1 \pmod{p}$, return the value T. T is the square root of D modulo p and our desired solution.*

2. *Otherwise, find the smallest j where $0 < j < U$ such that*

$$p^{2^j} \equiv 1 \pmod{p}.$$

Note this step can be accomplished efficiently by repeated squaring and reduction modulo p.

3. Let $B \equiv C^{2^{M-j-1}} \pmod{p}$*, and compute the following:*

$$\begin{aligned} T &\equiv T \cdot B \pmod{p}, \\ P &\equiv B^2 \pmod{p}, \\ C &\equiv B^2 \pmod{p}, \\ U &= j. \end{aligned}$$

Return to Step 1.

Once we have recovered the value T*, the second solution to our modular square root is* $p - T$*.*

The complexity of the algorithm is

$$2d + 2k + \frac{M(M-1)}{4} + \frac{1}{2^{M-1}} - 9$$

modular multiplications where d is the number of digits in the binary representation of p and k is the number of ones in the binary representation of p, and M is the same as above, the largest nonzero exponent of 2 that divides $p - 1$ exactly. This is the average complexity of Tonelli-Shanks over all possible inputs involving both quadratic residues and nonresidues.

In our algorithm, we need a quadratic nonresidue. One method is to randomly select a value y and check whether it is a nonresidue. This should take on average about 2 computations of the Legendre symbol, which costs about 2 Euclidean algorithm computations each [25]. The reason why we expect on average 2 of these computations to find our nonresidue is the probability that a randomly chosen y is a quadratic residue is

$$\frac{\frac{p+1}{2}}{p} = \frac{1 + \frac{1}{p}}{2},$$

which is less than 1 but greater than $\frac{1}{2}$. Therefore on average, selecting 2 elements from $\{0, 1, \ldots, p-1\}$ should result in finding a quadratic nonresidue. By assuming GRH, recovering a nonresidue can be done in polynomial time [1].

In conclusion, Tonelli-Shanks is a great choice for solving $x^2 \equiv D \pmod{p}$ when M from $p-1 = R \cdot 2^M$ is small relative to the number of digits in the binary expansion of p. Alternatively for large M, Cipolla's algorithm is proven to be more efficient in [25] if and only if

$$M(M-1) > 8M + 20.$$

2.5 Example of Tonelli-Shanks algorithm

We will show two separate examples illustrating the computation of the Tonelli-Shanks algorithm for computing a square root modulo a prime.

Example 1

We will solve $x^2 \equiv -23 \pmod{13}$. We satisfy the conditions to use Tonelli-Shanks since 13 is an odd prime and $\left(\frac{-23}{13}\right) = 1$ since

$$(-23)^{(13-1)/2} \equiv 1 \pmod{13}.$$

Step 1:

We have $M = 2$ since $p - 1 = 13 - 1 = 12 = 3 \cdot 2^2$, so $R = 3$.

Step 2:

We have $z = 2$ is a quadratic nonresidue modulo 13 as $\left(\frac{2}{13}\right) = -1$. Set

$$C \equiv z^R \equiv 2^3 \equiv 8 \pmod{13}.$$

Step 3:

Let

$$\begin{aligned} T &\equiv D^{(R+1)/2} \equiv (-23)^{(3+1)/2} \equiv 9 \pmod{13}, \\ P &\equiv D^R \equiv (-23)^3 \equiv 1 \pmod{13}, \\ U &= M = 2. \end{aligned}$$

Step 4:

Since $P \equiv 1 \pmod{13}$, we stop our loop and return $T = 9$.

The other solution to $x^2 \equiv -23 \pmod{13}$ is $p - T = 13 - 9 = 4$. This example does not illustrate properly the use of the loop in Step 4, so now we will attempt to solve $x^2 \equiv 10 \pmod{13}$.

Example 2

This is a valid use of Tonelli-Shanks since 13 is an odd prime and $\left(\frac{10}{13}\right) = 1$ since

$$10^{(13-1)/2} \equiv 1 \pmod{13}.$$

Step 1:

Since $p - 1 = 13 - 1 = 12 = 3 \cdot 2^2$, our $R = 3$ and $M = 2$ like from above.

Step 2:

We will use the same quadratic nonreside $z = 2$ and set

$$C \equiv z^R \equiv 2^3 \equiv 8 \pmod{13}.$$

Step 3:

Set

$$\begin{aligned} T &\equiv D^{(R+1)/2} \equiv 10^{(3+1)/2} \equiv -4 \pmod{13}, \\ P &\equiv D^R \equiv 10^3 \equiv -1 \pmod{13}, \\ U &= M = 2. \end{aligned}$$

Since $T \not\equiv 1 \pmod{13}$, we take j from $0 < j < 2$. We start with $j = 1$. Let

- Let

$$B \equiv C^{2^{M-j-1}} \equiv 8^{2^{2-1-1}} \equiv 8 \pmod{13},$$

  which means

$$B^2 \equiv 8^2 \equiv -1 \pmod{13}.$$

- Set

$$\begin{aligned} T &\equiv T \cdot B \equiv -4 \cdot 8 \equiv 7 \pmod{13}, \\ P &\equiv B^2 \equiv (-1)^2 \equiv 1 \pmod{13}, \\ U &= 1. \end{aligned}$$

We restart the loop. Since $P \equiv 1 \pmod{13}$, we return $T = 7$. The other solution is $p - T = 13 - 7 = 6$.

Chapter 3

Mean Number of Three Torsion Elements

In this chapter, we explore the number of three torsion elements in the class group of fake real quadratic orders.

Definition 3.1. Let G be a finite Abelian group and let $\gamma \in G$. Then γ is a 3-torsion element if and only if $\gamma^3 = 1_G$, where 1_G is the identity element of G. Another way to say this is $o(\gamma)|3$ (i.e. $o(\gamma) = 1$ or 3), where $o(\gamma)$ is the order of γ in the group G.

The classical theorems of Davenport and Heilbronn [10] state the asymptotic formulas for the total number of 3-torsion elements in the ideal class group of the quadratic fields with a bounded discriminant.

Theorem 3.2. *(Davenport, Heilbronn) Let D denote the discriminant of a quadratic field and let $\mathrm{Cl}_3(D)$ denote the 3-torsion subgroup of the ideal class group $\mathrm{Cl}(D)$ of $\mathbb{Q}(\sqrt{D})$. Then*

$$\sum_{0<D<X} |\mathrm{Cl}_3(D)| = \frac{4}{3} \cdot \sum_{0<D<X} 1 + o(X); \tag{3.1}$$

$$\sum_{-X<D<0} |\mathrm{Cl}_3(D)| = 2 \cdot \sum_{-X<D<0} 1 + o(X). \tag{3.2}$$

3.1 3-Torsion Elements in Fake Real Quadratic Orders

Since we have shown that the class group of fake real quadratics share similar class group structures as real quadratic fields, it would be safe to assume the average number of 3-torsion elements in fake real quadratic class groups would be $\frac{4}{3}$. The following theorem shows this fact is true, and the fields share this same property

when averaged simultaneously over D and p. That is, when D and p are sufficiently large, the desired result occurs.

Theorem 3.3. *(O.) Let $\mathcal{O}_{D,p}$ denote the fake real quadratic order derived from the imaginary quadratic field $\mathbb{Q}(\sqrt{D})$ and prime p where D is a quadratic residue modulo p. Let $\mathrm{Cl}_3(D,p)$ be the 3-torsion subgroup of the ideal class group $\mathrm{Cl}_{D,p}$ of $\mathcal{O}_{D,p}$. Then*

$$\sum_{-X<D<0} \sum_{\substack{0<p<Y \\ (p)=\mathfrak{p}\bar{\mathfrak{p}}}} |\mathrm{Cl}_3(D,p)| = \frac{4}{3} \cdot \frac{1}{2} \cdot \mathrm{Li}(Y) \cdot \frac{3X}{\pi^2} + \textit{error}, \tag{3.3}$$

where the error bound assuming GRH is

$$O((XY)^{1/2} \cdot \log^2 X + XY^{1/2} \cdot \log Y). \tag{3.4}$$

Before we prove this, the following lemma is needed.

Lemma 3.4. *Let G be a finite Abelian group, and let $r_3(G)$ denote the 3-rank of G. Let $\langle a \rangle$ denote a cyclic subgroup generated by $a \in G$. Then if $a \in G$ is chosen at random with probability $\frac{1}{|G|}$, then the average value of $|\mathrm{Tor}_3(G/\langle a \rangle)|$ is $\dfrac{3^{r_3(G)}+2}{3}$, where $\mathrm{Tor}(A)$ represents the 3-torsion subgroup of group G.*

Proof. Let $r_3(G) = k$. By the fundamental theorem of finitely generated abelian groups, we have $G/\langle 1 \rangle \simeq G$, the group of order k. There are $3^k - 1$ elements $a \in G$ where $a \neq 1_G$ that will generate $G/\langle a \rangle$, which is a subgroup of 3^{k-1}. Therefore, since each element of G has a $\frac{1}{|G|}$ probability of being chosen, the average size of the 3-torsion subgroup will be

$$3^k \cdot \frac{1}{3^k} + 3^{k-1} \cdot \frac{3^k - 1}{3^k} = 1 + \frac{3^k - 1}{3} = \frac{3^k + 2}{3}.$$

□

By the Davenport-Heilbronn Theorem, the mean number of 3-torsion elements for imaginary quadratic fields is 2. Another way to state this is

$$\sum_{k=0}^{\infty} \mathrm{Prob}(k) \cdot 3^k = 2,$$

where $\mathrm{Prob}(k)$ represents the probability that $r_3(\mathrm{Cl}(D)) = k$, and $\mathrm{Cl}(D)$ denotes the ideal class group for an imaginary quadratic field for $-X < D < 0$. Since Lemma 3.4 states that the average size of the 3-torsion subgroup of a finite Abelian group modulo a random cyclic subgroup is $\frac{3^{r_3(G)}+2}{3}$, the mean number of 3-torsion elements for fake real qudratic fields is

$$\sum_{k=0}^{\infty} \mathrm{Prob}(k) \cdot \left(\frac{3^k+2}{3}\right).$$

Since

$$\sum_{k=0}^{\infty} \mathrm{Prob}(k) \cdot 3^k = 2,$$

we have

$$\sum_{k=0}^{\infty} \mathrm{Prob}(k) \cdot 3^k - 1 = 1,$$

which simplifies to

$$\sum_{k=0}^{\infty} \mathrm{Prob}(k) \cdot 3^k - \sum_{k=0}^{\infty} \mathrm{Prob}(k) = \sum_{k=0}^{\infty} \mathrm{Prob}(k)(3^k - 1) = 1,$$

since the sum of all probabilities equals 1. Now dividing both sides by 3 results in

$$\sum_{k=0}^{\infty} \mathrm{Prob}(k) \cdot \left(\frac{3^k-1}{3}\right) = \frac{1}{3}.$$

Therefore, we have

$$\begin{aligned}\sum_{k=0}^{\infty} \mathrm{Prob}(k) \cdot \left(\frac{3^k-1}{3}\right) + 1 &= \sum_{k=0}^{\infty} \mathrm{Prob}(k) \cdot \left(\frac{3^k-1}{3}\right) + \sum_{k=0}^{\infty} \mathrm{Prob}(k) \\ &= \sum_{k=0}^{\infty} \mathrm{Prob}(k) \cdot \left(\frac{3^k+2}{3}\right) \\ &= \frac{4}{3}. \end{aligned} \tag{3.5}$$

Now, we can prove our theorem on hand.

Proof. By (3.5), we are able to replace $|\mathrm{Cl}_3(D,p)|$ with $\frac{4}{3}$, resulting in

$$\sum_{-X<D<0} \sum_{\substack{0<p<Y \\ (p)=\mathfrak{p}\bar{\mathfrak{p}}}} |\mathrm{Cl}_3(D,p)| = \frac{4}{3} \sum_{-X<D<0} \sum_{\substack{0<p<Y \\ (p)=\mathfrak{p}\bar{\mathfrak{p}}}} 1.$$

Using the fact that half the primes in $\mathbb{Q}(\sqrt{D})$ split and the prime number theorem, we get

$$\frac{4}{3}\sum_{-X<D<0}\sum_{\substack{0<p<Y\\(p)=\mathfrak{p}\overline{\mathfrak{p}}}}1=\frac{4}{3}\cdot\frac{1}{2}\cdot\mathrm{Li}(Y)\cdot\sum_{-X<D<0}1.$$

By counting the number of negative fundamental discrimants taken from [21], our main term becomes

$$\frac{4}{3}\cdot\frac{1}{2}\cdot\mathrm{Li}(Y)\cdot\frac{3X}{\pi^2}.$$

Assuming GRH, our error bound is

$$\begin{aligned}
&\ll\sum_{-X<D<0}\left(\frac{Y^{1/2}\cdot\log(|D|+Y^{h(D)})}{h(D)}+\log|D|\right)\\
&=\sum_{-X<D<0}\left(\frac{Y^{1/2}\cdot\log|D|}{h(D)}+Y^{1/2}\cdot\log Y+\log|D|\right)\\
&\le c_3\cdot Y^{1/2}\cdot\sum_{-X<D<0}\frac{\log^2|D|}{\sqrt{|D|}}+Y^{1/2}\cdot\log Y\cdot\sum_{-X<D<0}1+\sum_{-X<D<0}\log|D|\\
&=(c_3\cdot X^{-1/2}\cdot Y^{1/2}\cdot\log^2 X+Y^{1/2}\log Y+\log X)\cdot\left(\frac{3X}{\pi^2}+O(X^{17/54+\varepsilon})\right)\\
&\ll(XY)^{1/2}\cdot\log^2 X+XY^{1/2}\cdot\log Y
\end{aligned}$$

The unconditional error bound from Chebotarev is

$$\begin{aligned}
&\ll\sum_{-X<D<0}\left(\frac{1}{h(D)}\cdot\frac{Y}{\log Y}+c_1\cdot Y\cdot\exp(-c_2\cdot h(D)^{-1/2}\cdot(\log Y)^{1/2})\right)\\
&\sim\sum_{-X<D<0}\left(\frac{1}{\sqrt{|D|}}\cdot\frac{Y}{\log Y}+c_1\cdot Y\cdot\exp(-c_2|D|^{-1/4}\cdot(\log Y)^{1/2})\right)\\
&\sim\frac{3\sqrt{X}Y}{\pi^2\cdot\log Y}+3c_1\cdot\frac{XY}{\pi^2}\cdot\exp(-c_2X^{-1/4}(\log Y)^{1/2})\\
&\ll\frac{X^{1/2}Y}{\log Y}+XY\cdot\exp(X^{-1/4}(\log Y)^{1/2}).
\end{aligned}$$

□

Chapter 4

Cohen-Lenstra Heuristics

4.1 Class Number Problem

One of the most prominent open conjectures of real quadratic fields started with the class number problems proposed by Gauss. In 1801, he discussed and proposed two conjectures in regards to imaginary quadratic fields and stated a third conjecture for real quadratic fields.

Conjecture 4.1. *(Gauss) Let $D < 0$, and let $h(D)$ denote the class number for $\mathbb{Q}(\sqrt{D})$. Then $h(D)$ tends to ∞ as D tends to $-\infty$.*

This result was proven by Heilbronn [14] in 1934. This beautiful statement does not shed light on how many fields have a particular class number since his proof proved an ineffective lower bound for $h(D)$. Gauss's second conjecture on imaginary quadratic fields attempts to answer the question of how many.

Conjecture 4.2. *(Gauss) Given n, there are finitely many imaginary quadratic fields with class number n.*

Heilbronn and Linfoot [15], in 1934, proved that there are at most 10 imaginary quadratic fields with class number 1, with 9 of those fields being known at that time and one unknown. Through recent developements, Watkins [27], in 2004, gave a exhaustive list of all imaginary quadratic fields for all class numbers up to 100.

Dealing with imaginary quadratic fields is different and is considered generally easier to deal with than real quadratic fields. We will go into slightly more depth on

why imaginary quadratic fields are easier to deal with than real quadratics in a later section. It is evident to see that it is true with the following open problem.

Conjecture 4.3. *(Gauss) There are infinitely many real quadratic fields with class number one.*

It is assumed that this conjecture is true, although there have been no great advances in establishing its validity. It is obvious to see the difference between real versus imaginary quadratic fields as we know there are exactly 9 fundamental discriminants of imaginary quadratic fields with class number one. Not much progress has been made in regards to this conjecture.

One of the reasons that this problem contrasts sharply is the class number formula of each respective type of quadratic field.

4.2 Class Number Formula for Real and Imaginary Quadratic Fields

Recall the class number formula for real quadratic fields is

$$h(D) = \frac{\sqrt{D} \cdot L(1, \chi_D)}{\log \varepsilon}. \tag{4.1}$$

For imaginary quadratic fields, the class number formula is

$$h(D) = \frac{\omega \cdot \sqrt{|D|} \cdot L(1, \chi_D)}{2\pi}, \tag{4.2}$$

where

$$\omega = \begin{cases} 2 & : \quad D < -4 \\ 4 & : \quad D = -4 \\ 6 & : \quad D = -3. \end{cases} \tag{4.3}$$

Inspecting the two contrasting formulas, we see that the analytic formula for class number for real quadratic fields has an extra factor of $\log \varepsilon$ in the formula. This extra factor makes it harder to control. Although it is well believed that there are infinitely many real quadratic fields with class number one, there has been very little

progress made on this. The heuristic argument of Cohen and Lenstra where they produced probabilistic arguments on which Abelian groups occur naturally is a major conjecture regarding real quadratic fields with class number one. These probabilities are weighted by the size of these automorphism groups.

Conjecture 4.4. *(Cohen, Lenstra) The proportion of real quadratic fields with prime discriminant* $D \equiv 1 \pmod 4$ *such that* $h(D) = 1$ *should exist and be equal to*

$$C = \prod_{k \geq 2} (1 - 2^{-k})^{-1} \zeta(k)^{-1} = 0.754458\ldots.$$

In general, the proportion of those class groups to a given Abelian group G *should be proportional to*

$$\frac{C}{|G| \cdot |\operatorname{Aut}(G)|},$$

where $\operatorname{Aut}(G)$ *is the group of automorphisms of* G*.*

4.3 Heuristics for Fake Real Quadratic Orders

Although the data strongly shows the similarity, Cohen makes a clear statement about the difficulty of producing an asymptotic since the data set is small, and the convergence to this limit is slow. The ability to differentiate between large powers of log versus small powers of x is quite difficult. Based on Cohen's data from an unpublished manuscript, he states that the asymptotic for the secondary term could be

$$0.754458\ldots + \frac{230}{\log^3(x)},$$

but it could also be

$$0.754458\ldots + \frac{0.505}{x^{1/5}}.$$

Regardless, the Cohen-Lenstra heuristics seem to hold true numerically. We give the following analogue for fake real quadratic orders.

Conjecture 4.5. *Let p be a fixed prime. The proportion of primes $D \equiv 1 \pmod 4$ such that $\left(\frac{D}{p}\right) = 1$ of which $h_{K,p} = 1$, where $K = \mathbb{Q}(\sqrt{D})$ is the imaginary quadratic field, exists and is equal to $C = 0.754458\ldots$.*

This conjecture is a natural implication since the class group of a real quadratic field is isomorphic to $G/\langle\sigma\rangle$, where G is an abelian group and σ is a randomly chosen element from G [8]. Fake real quadratic orders have the same structure. The class group has structure $\mathrm{Cl}_K/\langle\mathfrak{p}\rangle$, where Cl_K is the class group of an imaginary quadratic field and $\mathfrak{p}$ is a randomly chosen element of the class group which represents an element of the class group. Thus, this conjecture is implied from what Cohen and Lenstra originally conjectured regarding quadratic fields.

Here are the tables to the modest limit of 2^{26} of the proportion of fake real quadratic orders with $D \equiv 3 \pmod 4$ with class number $h_{K,p} = 1$. The convergence of the fake real quadratic is consistent with the real quadratic fields the proportion is the far right column of both tables.

Table 4.1 Ratio of Prime Discriminants $D \equiv 1 \pmod 4$ of Fake Real Quadratics with $h_{K,p} = 1$: Part 1

	2	3	5	7	11	Real
2^{14}	.833232	.820886	.826853	.827565	.823529	.810597
2^{15}	.807906	.805917	.812256	.803711	.806630	.808206
2^{16}	.806424	.794974	.808407	.794414	.796014	.795179
2^{17}	.795764	.794199	.791770	.790069	.794188	.793950
2^{18}	.789878	.787790	.788826	.786843	.784353	.789673
2^{19}	.782992	.784135	.783784	.782650	.778906	.784095
2^{20}	.780740	.780362	.779968	.778748	.776097	.777829
2^{21}	.776594	.776978	.775072	.772542	.771588	.774867
2^{22}	.773628	.772959	.772230	.771919	.770220	.772734
2^{23}	.770983	.771040	.770584	.770025	.769845	.770283
2^{24}	.768857	.768855	.768615	.768377	.767820	.768519
2^{25}	.767230	.766897	.766656	.766279	.765958	.766459
2^{26}	.765842	.765504	.765098	.764852	.763370	.764857

The figure following the table places the information found in the table in a visually appealing graph.

Table 4.2 Ratio of Prime Discriminants $D \equiv 1 \pmod 4$ of Fake Real Quadratics with $h_{K,p} = 1$: Part 2

	13	17	19	23	29	Real
2^{14}	.816413	.798417	.818237	.795745	.819682	.810597
2^{15}	.805951	.795912	.794060	.802812	.810705	.808206
2^{16}	.804747	.796138	.793235	.795851	.799757	.795179
2^{17}	.792229	.789720	.787567	.791605	.790901	.793950
2^{18}	.783438	.789030	.782039	.787057	.786245	.789673
2^{19}	.779968	.783448	.779649	.782491	.778906	.784095
2^{20}	.777067	.778146	.776277	.778748	.776097	.777829
2^{21}	.773423	.773714	.773678	.772542	.771588	.774867
2^{22}	.771451	.770931	.771173	.769554	.770261	.772734
2^{23}	.769238	.769426	.768673	.767680	.767901	.770283
2^{24}	.767576	.767239	.766869	.765940	.766849	.768519
2^{25}	.765572	.765256	.765058	.764624	.764370	.766459
2^{26}	.764243	.763984	.764104	.763558	.763370	.764857

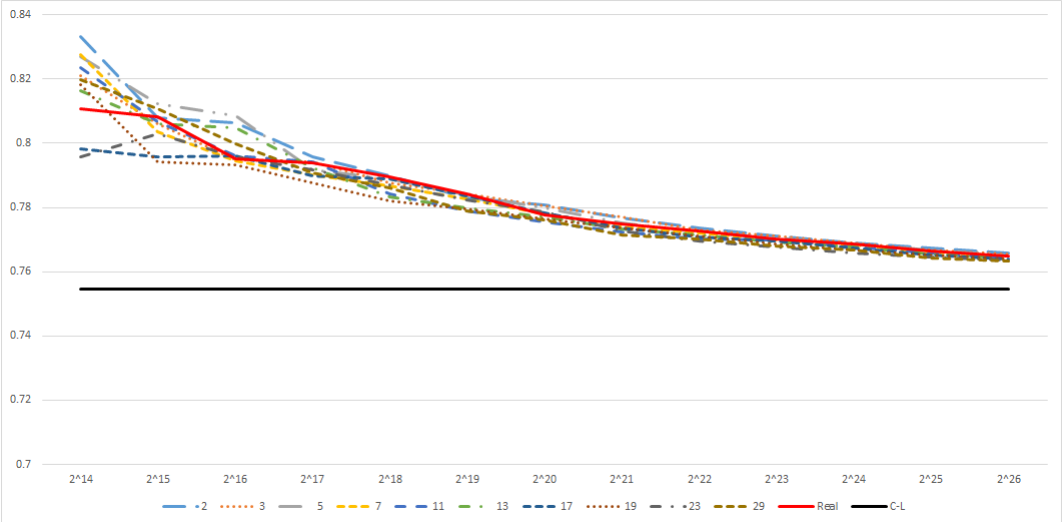

Figure 4.1 Graph of Ratio of Prime Discriminants $D \equiv 1 \pmod 4$ such that $h_{K,p} = 1$

Chapter 5

Ankeny-Artin-Chowla Conjecture

This section is credited to Henri Cohen. His work investigated the analogue to the Ankeny-Artin-Chowla conjecture for fake real quadratic orders. He computed the counterexamples, but the author computed the fundamental unit counterexamples and placed them in the Appendix.

5.1 Conjecture for Real Quadratic Fields

The Ankeny-Artin-Chowla Conjecture (AAC for short) is a divisibility result involving the fundmental unit with the discriminant of the field. Specifically, let $K = \mathbb{Q}(\sqrt{D})$ be a real quadratic field with prime discriminant D such that $D \equiv 1 \pmod 4$. Then the fundamental unit of K has the form

$$\varepsilon = \frac{a + b\sqrt{D}}{2}.$$

The AAC Conjecture asserts that $D \nmid b$. As previously stated in the introduction, this conjecture is still an open problem, although extensive numerical data has been generated over the years which confirms the truth for $D < X$. Table 5.1 gives the progression of verification since its proposal in 1952, and the most recent result given by van der Poorten, Williams, et al., verifying the conjecture to be true for $D < 2{\cdot}10^{11}$.

We must mention that although there are effective and "fast" methods to compute the fundamental unit for real quadratic fields, the computations can be extremely memory intensive. For example, for discriminant $D = 40,094,470,441$, the a and b values of the fundamental unit exceed 10^{330000} [26]. The work of van der Poorten,

Table 5.1 Verification of AAC Conjecture for all $D < X$

X	Investigator(s)	Date
2 000	Ankeny, Artin, Chowla $p \equiv 5 \pmod 8$ only	1952
100 000	Goldberg	1954
6 279 714	Beach, Williams, Zarnke	1971
100 028 010	Soleng	1986
1 000 000 000	Stephens, Williams	1988
200 000 000 000	van der Poorten, te Riele, Williams	2000

Williams, et al designed special equipment and algorithms to minimize the memory load in order to verify as far as they did. To most, the value of $X = 2 \cdot 10^{11}$ might seem like a very large number, so it might seem convincing that this conjecture may be true. Despite the lack of a counterexample, there are many who believe this conjecture is false. The "log log" argument is a popular explanation why the AAC conjecture may be false. One way to state this argument is we have nothing to show otherwise that $b \mod D$ behaves randomly. Thus,

$$\sum_{\substack{D<X \\ D\equiv 1 \pmod 4 \\ D \text{ prime}}} \frac{1}{D}, \tag{5.1}$$

should tell us how many counterexamples we should expect for a given X value. The integral from (5.1) is of the order of $\frac{1}{2}\log\log X$, where the $\frac{1}{2}$ accounts for the $D \equiv 1 \pmod 4$ condition. So even for the "large" value $X = 2 \cdot 10^{11}$, it is expected to have 1.39 counterexamples. Therefore, it is not unreasonable that there have been no counterexamples found even for this large of an X value.

5.2 Analogue Conjecture for fake real quadratic orders

Since we believe fake real quadratic orders behave the same as real quadratic fields, it would be interesting to study the analogue to the AAC conjecture and see if it holds up the same way. Perhaps not surprisingly, there are counterexamples. The following counterexamples are given by Cohen [7]. The simpliest of all of them is $p = 7$ and

$D = -3$. For $K = \mathbb{Q}(\sqrt{-3})$, the class number $h(-3) = 1$. Therefore, without loss of generality, $\mathfrak{p} = (2 + \sqrt{-3})$ and $\overline{\mathfrak{p}} = (2 - \sqrt{-3})$, where $(7) = \mathfrak{p}\overline{\mathfrak{p}}$. The fundamental unit for $\mathcal{O}_{\mathbb{Q}(\sqrt{-3}),7}$ is

$$\varepsilon = (2 + \sqrt{-3}) \left(\frac{1 + \sqrt{-3}}{2} \right)^n,$$

where $n \in \mathbb{Z}$. Letting $n = 1$, the fundamental unit

$$\varepsilon_{\mathbb{Q}(\sqrt{-3}),7} = \frac{-1 + 3\sqrt{-3}}{2}$$

violates the AAC conjecture since clearly $-3 \,|\, 3$. Although this is a counterexample, one might question the validity that this is a true counterexample for the analogue since the group of roots of unity for this field has order 6. To avoid this being a complication since real quadratics have unit group of rank 1, we can take a fake real quadratic that has a similar structure. Consider $p = 347$ with discriminant $D = -7$. Then the fundamental unit for this particular fake real is

$$\varepsilon_{\mathbb{Q}(\sqrt{-7}),347} = 2 + 7\sqrt{-7}.$$

It is quite obvious that this violates the AAC conjecture, but avoid the earlier complication since the unit group for $\mathcal{O}_{\mathbb{Q}(\sqrt{-7}),347}$ has rank 1.

It appears the $\log\log$ philosophy holds true for the fake real quadratic case. As D varies with fixed p, the probability of failure is $\frac{1}{D}$, which is expected since there is no reason why we believe D should not behave randomly. Cohen mentions that no counterexamples have been found for $p = 2, 3, 5, 11, 17$, and 29, although this has been only tested for a very small bound $D < 1072000$. Further analysis could be done for these primes, but it would not be unreasonable to speculate that perhaps these primes without counterexamples for very large D might inherently behave exactly as real quadratic fields. It would be interesting to study these and see if there is any substance to these unsupported claims.

To see more counterexamples, see Appendix D.

Chapter 6

Number Field Cryptography Application

Public key cryptography is one of the main techniques for securing the Internet. Although there are many symmetric key crypto-systems that are secure, they require both parties to have the same key. Communicating this secure key over the Internet is a difficult problem, which presents an issue on using symmetric key crypto-systems. Most of the current public key crypto-systems, such as RSA, are based on seemingly difficult computational problems in number theory such as factoring integers.

In 1988, Buchmann and Williams [2] presented a variant of the classical Diffie-Hellman key exchange protocol using the class groups of imaginary quadratic fields. Since then, there has been development in this field such as extending it to real quadratic fields. There are also methods now to devise cryptographic protocols using these fields [3].

In this chapter, we will give an overview of the key exchange protocol using the ideal class group of fake real quadratic orders. This protocol is identical to the one proposed by Buchmann and Williams proposed for imaginary quadratic fields with an exception. As stated in previous sections, fake real quadratic orders have similar class group structures to real quadratic fields. The algorithm discussed here gives another way to use Buchmann and Williams key exchange but given a restriction on how the ideals are chosen. This adds an additional bit of difficulty to the problem since the class group used has a structure similiar to real quadratic fields rather than imaginary quadratic ones. The security proof shows that using the correct procedure, the users should expect reasonable protection of the key and another attempt to a solution for

the key exchange problem. Therefore, we are able to accept the security proof given by them, but we will also discuss the limitations and the cautions regarding the use of this protocol with these fake reals.

6.1 Ideals of Imaginary Quadratic Fields

From Proposition 1.1, we saw that the map φ, defined by $\mathfrak{a} \mapsto \mathfrak{a}\mathcal{O}_{K,p}$ where $\mathfrak{a}$ is from the set of integral ideals of $\mathcal{O}_K$ mapping to ideals of $\mathcal{O}_{K,p}$, is an isomorphism. Therefore, we can look at the properties of $\mathfrak{a}$ as an ideal of $\mathcal{O}_K$ and develop the same properties for the ideals of $\mathcal{O}_{K,p}$.

Before we develop these properties about the ideals in $\mathcal{O}_K$, we need to define some of the terminology. This section is borrowed heavily from [2]. If $\alpha \in K$, let the trace of α is $\mathrm{Tr}(\alpha) = \alpha + \overline{\alpha}$, and the norm of α is $N(\alpha) = \alpha\overline{\alpha}$, so that $|\alpha|^2 = \alpha\overline{\alpha} = N(\alpha)$.

If $\alpha, \beta \in K$, let $[\alpha, \beta]$ denote the set $\alpha\mathbb{Z} + \beta\mathbb{Z}$. Since K is an imaginary quadratic field, $\mathcal{O}_K = [1, \omega]$, where

$$\omega = \frac{r - 1 + \sqrt{D}}{r},$$

where

$$r = \begin{cases} 1 & : \quad D \equiv 2, 3 \pmod 4 \\ 2 & : \quad D \equiv 1 \pmod 4 \end{cases}$$

Thus, $\alpha \in \mathcal{O}_K$ if and only if $\mathrm{Tr}(\alpha)$ and $N(\alpha)$ are both in $\mathbb{Z}$. The discriminant Δ of K is equal to $(\omega - \overline{\omega})^2 = 4D/r^2$.

Each integral principal ideal $\mathfrak{a}$ of $\mathcal{O}_K$ has the form

$$\mathfrak{a} = \mathbb{Z} + \mathbb{Z}\frac{b + \sqrt{D}}{2a}$$

where $a, b \in \mathbb{Z}$, $0 < a < \sqrt{D}$, and $0 \leq b < 2a$. For shorthand, we denote

$$\mathfrak{a} = [a, b + c\omega], \tag{6.1}$$

where $a, b, c \in \mathbb{Z}$, $a > 0$, and $c > 0$. We have $c|a$, $c|b$, and $ac|N(b + c\omega)$. In addition, if $\mathfrak{a} = [a, b + c\omega]$, where $a, b, c \in \mathbb{Z}$, $c|a$, $c|b$, and $ac|N(b + c\omega)$, then $\mathfrak{a}$ is an ideal of

$\mathcal{O}_K$. For a given ideal $\mathfrak{a}$, the value a is called the least positive rational integer in $\mathfrak{a}$, and we denote it by $L(\mathfrak{a})$. We say an ideal $\mathfrak{a}$ is *primitive* if it is not divisible by any other ideal except by the trivial ideal (1). If an ideal is primitive, then $c = 1$.

The following lemmas are attributed to Buchmann and Williams, and are given without proof. The proofs can be found in [2].

Lemma 6.1. *(Buchmann, Williams) If $\mathfrak{a}$ is any primitive ideal of $\mathcal{O}_K$, then there exists some $\alpha \in \mathfrak{a}$ such that $\mathfrak{a} = [L(\mathfrak{a}), \alpha]$ and $|\operatorname{Tr}(\alpha)| \leq L(\mathfrak{a})$.*

The following lemma tells us that the value of $|\operatorname{Tr}(\alpha)|$ is unique.

Lemma 6.2. *(Buchmann, Williams) If $\mathfrak{a}$ is an ideal of $\mathcal{O}_K$, $\alpha = [L(\mathfrak{a}), \alpha] = [L(\mathfrak{a}), \beta]$, and $|\operatorname{Tr}(\alpha)| \leq L(\mathfrak{a})$, $|\operatorname{Tr}(\beta)| \leq L(\mathfrak{a})$, then $|\operatorname{Tr}(\alpha)| = |\operatorname{Tr}(\beta)|$.*

We call $\mathfrak{a}$ a *reduced* ideal if $\mathfrak{a}$ is primitive and there does not exist a $\beta \in \mathfrak{a}$ such that $|\beta| < L(\mathfrak{a})$. The following theorems tell us when $\mathfrak{a}$ is a reduced ideal and some properties of reduced ideals.

Theorem 6.3. *(Buchmann, Williams) If $\mathfrak{a}$ is a primitive ideal of $\mathcal{O}_K$ and $\mathfrak{a} = [L(\mathfrak{a}), \alpha]$ with $|\operatorname{Tr}(\alpha)| \leq L(\mathfrak{a})$, then $\mathfrak{a}$ is a reduced ideal if and only if $|\alpha| \geq L(\mathfrak{a})$.*

Theorem 6.4. *(Buchmann, Williams) If $\mathfrak{a}$ is a reduced ideal of $\mathcal{O}_K$, then $L(\mathfrak{a}) < \sqrt{|\Delta|/3}$.*

Theorem 6.5. *(Buchmann, Williams) If $\mathfrak{a}$ is a primitive ideal of $\mathcal{O}_K$ and $L(\mathfrak{a}) < \sqrt{|\Delta|}/2$, then $\mathfrak{a}$ is a reduced ideal of $\mathcal{O}_K$.*

Lemma 6.6. *(Buchmann, Williams) If $\mathfrak{a} \sim \mathfrak{b}$, then there exists $\alpha \in \mathfrak{a}$ such that $(\alpha)\mathfrak{b} = (L(\mathfrak{b}))\mathfrak{a}$.*

Finally, the next theorem gives us that there are at most two reduced ideals in any given equivalence class of ideals.

Theorem 6.7. *(Buchmann, Williams) Let $\mathfrak{a}, \mathfrak{b}$ be primitive ideals of $\mathcal{O}_K$ where $\mathfrak{a} = [L(\mathfrak{a}), \alpha]$ and $\mathfrak{b} = [L(\mathfrak{b}), \beta]$ such that $|\operatorname{Tr}(\mathfrak{a})| \leq L(\mathfrak{a})$ and $|\operatorname{Tr}(\mathfrak{b})| \leq L(\mathfrak{b})$. If $\mathfrak{a} \sim \mathfrak{b}$, then $L(\mathfrak{a}) = L(\mathfrak{b})$ and $|\operatorname{Tr}(\alpha)| = |\operatorname{Tr}(\beta)|$.*

We will show in the section that each equivalence class of $\mathcal{O}_K$ contains a single reduced ideal. We will present Buchmann and Williams's algorithm for recovering this reduced ideal. This is equivalent to Gauss's reduction theory.

6.2 Ideal Reduction (Found in [2])

This section will provide two algorithms to reduce primitive ideals. We note that if $\mathfrak{a} = [L(\mathfrak{a}), \alpha]$ is a primitive ideal of $\mathcal{O}_K$, then the ideal $\mathfrak{b} = [N(\alpha)/L(\mathfrak{a}), -\overline{\alpha}]$ is also a primitive ideal. Therefore,

$$(\overline{\alpha})\mathfrak{a} = (L(\mathfrak{a}))\mathfrak{b}.$$

This means that $\mathfrak{a} \sim \mathfrak{b}$.

Algorithm 6.8. *1. Given a primitive ideal $\mathfrak{a} = \mathfrak{a}_1 = [L(\mathfrak{a}), \alpha]$ of $\mathcal{O}_K$. Define $Q_0 = rL(\mathfrak{a}) > 0$ and $P_0 = r\alpha - \sqrt{D} \in \mathbb{Z}$. Recall that the value r is the same as discussed in Section 6.1.*

2. Compute the following:

$$\begin{cases} q_i & = \operatorname{Ne}(P_i/Q_i), \\ P_{i+1} & = q_i Q_i = P_i, \\ Q_{i+1} & = \dfrac{P_{i+1}^2 - D}{Q_i}, \end{cases} \tag{6.2}$$

where $\operatorname{Ne}(\eta)$ is the nearest integer (i.e. $|\eta - \operatorname{Ne}(\eta)| \leq \frac{1}{2}$). This is unique unless η has fractional part exactly $\frac{1}{2}$.

3. We have

$$\mathfrak{a}_{i+1} = \left[\frac{Q_i}{r}, \frac{P_i + \sqrt{D}}{r}\right],$$

is a reduced ideal of $\mathcal{O}_K$ *when*

$$Q_{i+1} \geq Q_i.$$

Proof. The proof can be found in Buchmann and Williams paper in [2]. □

The computations in Algorithm 6.8 can be quite cumbersome, so the following modification to the above provides a method for easier computations. Let $T_0 = |P_0|$, $t_0 = P_0/T_0$, and $Q_{-1} = (P_0^2 - D)/Q_0$. Then we change Step 2 in the above algorithm as follows:

Algorithm 6.9.

$$\begin{aligned} s_i &= [T_i/Q_i] \\ R_i &= \textit{remainder from dividing } T_i \textit{ by } Q_i. \\ M_i &= Q_i - 2R_i. \end{aligned}$$

If $M_i \geq 0$*, then*

$$\begin{aligned} T_{i+1} &= R_i, \\ Q_{i+1} &= Q_{i-1} - s_i(R_i + T_i), \\ t_{i+1} &= -t_i, \end{aligned}$$

while if $M_i < 0$*, then*

$$\begin{aligned} T_{i+1} &= R_i + M_i, \\ Q_{i+1} &= Q_{i-1} - s_i(R_i + T_i) + M_i, \\ t_{i+1} &= t_i. \end{aligned}$$

In this version, P_i from (6.2) is $P_i = t_i T_i$ for all $i \geq 0$.

The following theorem gives us a upper bound on the number of steps necessary to reach a reduced ideal, ensuring that the algorithm gives us a reduced ideal every time.

Theorem 6.10. *If $\mathfrak{a}$ is given as in Algorithm 6.8, then we have $Q_{i+1} \geq Q_i$ for some $i \leq 2 + [\frac{1}{2}\log_2(3Q_0/5\sqrt{|D|})]$.*

With this theorem, we now have an algorithm that will give us a reduced ideal every time. In the next section, we will discuss the key-exchange protocol.

6.3 Key Exchange Protocol

We will now discuss the key exchange protocol. Suppose that Alice and Bob would like to exchange a key. Select a value of D such that $|D|$ is large, approximately at least 10^{200}. As discussed, select an ideal $\mathfrak{a}$ in $\mathcal{O}_{K,p}$ where $\mathfrak{a}$ is coprime to $\mathfrak{p}$. Both the value of D and the ideal $\mathfrak{a}$ can be made public.

1. Let Alice choose a random integer x. Alice computes $\mathfrak{a}^x \sim \mathfrak{b}$, where $\sim$ represents ideal equivalence discussed in Section 6.1. Alice sends $\mathfrak{b}$ to Bob.

2. Bob selects a random integer y and computes $\mathfrak{a}^y = \mathfrak{c}$. He sends $\mathfrak{c}$ to Alice.

3. Alice computes the reduced ideal $\widetilde{\mathfrak{f}_1} \sim \mathfrak{c}^x$.

4. Bob computes the reduced ideal $\widetilde{\mathfrak{f}_2} \sim \mathfrak{b}^y$.

Since

$$\widetilde{\mathfrak{f}_1} \sim \mathfrak{c}^x \sim (\mathfrak{a}^y)^x = (\mathfrak{a}^x)^y \sim \mathfrak{b}^y \sim \widetilde{\mathfrak{f}_2},$$

Alice and Bob both have the same reduced ideal of $\mathcal{O}_{K,p}$. Therefore, by Theorem 6.7, we have that $L(\widetilde{\mathfrak{f}_1}) = L(\widetilde{\mathfrak{f}_2})$. Now, if both $\widetilde{\mathfrak{f}_1} = [L(\widetilde{\mathfrak{f}_2}), \kappa_1]$ and $\widetilde{\mathfrak{f}_2} = [L(\widetilde{\mathfrak{f}_2}), \kappa_2]$, then $|\operatorname{Tr}(\kappa_1)| = |\operatorname{Tr}(\kappa_2)|$. Thus, Alice and Bob have the option to agree on whether $L(\widetilde{\mathfrak{f}_1}) = L(\widetilde{\mathfrak{f}_2})$ or $|\operatorname{Tr}(\kappa_1)| = |\operatorname{Tr}(\kappa_2)|$ to be the secret. Otherwise, they could agree on certain parts of the above two options could be chosen to be the secret key.

There is a cautionary tale here as since $L(\widetilde{\mathfrak{f}_1})|N(\kappa_1)$, $L(\widetilde{\mathfrak{f}_1})$ and $\operatorname{Tr}(\kappa_1)$ are not linearly independent, thus using parts of both could compromise the secret value of the key.

This key exchange system can be used as part of the public key crypto-system in a similar manner designed by El Gamal [11]. If Alice intends to send Bob a secure message, they would exchange keys as discussed above. Let $L(\widetilde{\mathfrak{f}})$ be the key that has been agreed upon between the two parties. Again, we emphasize that x is only known to Alice, and y is only known to Bob. Let $M = M_1 M_2 \dots M_k$ be the plaintext message Alice wishes to send to Bob, where M_i is a block corresponding to the bitlength of the key chosen. Then the cipher text sent from Alice to Bob is

$$(M_1 + L(\widetilde{\mathfrak{f}}), \mathfrak{b}),$$

where $\mathfrak{b} \sim \mathfrak{a}^x$, and M_1 is the first block of M and where the $M_1 < L(\widetilde{\mathfrak{f}})$. Each subsequent block of M can be sent in a similar way, although for additional security, Alice should select a new value of x for each block in Step 3 of the Key Exchange Protocol.

To decipher the message, Bob must determine the value of $L(\widetilde{\mathfrak{f}})$, but since they have already exchanged keys, he has $\mathfrak{b} \sim \mathfrak{a}^x$ and his secret value y. Thus, he recovers the key by computing

$$\mathfrak{b}^y \sim \mathfrak{a}^{xy} \sim \mathfrak{c}^x \sim \widetilde{\mathfrak{f}},$$

thus allowing Bob to recover M_1.

6.4 Security Proof

Security proofs for cryptosystems show that breaking the encryption via brute force is computationally equivalent to a difficult problem, such as factorization of a large number. We aim to show this version of the cryptosystem has similar difficulty as the discrete log problem, which is believed to be difficult. One of the biggest drawbacks to the described key exchange protocol is that the complexity is greater than El Gamal's system [11], which is an asymmetric encryption algorithm based on the Diffie-Hellman key exchange. In addition, the bandwidth required is much greater.

For example, in order to use a 100 digit key, Alice and Bob must communicate 200 digits of information. With the sacrifice of computational ease and increase of bandwidth, it is apparent the security of this system should be greater than [11] for increase in effort. Unfortunately, this crypto-system is not provably secure, but we will give evidence that this system has the foundations of a secure system.

As stated before, the number of equivalence classes of ideal in $\mathcal{O}_{K,p}$ is called the class number and denoted by $h_{K,p}$. If $\mathscr{C}_1$ and $\mathscr{C}_2$ are two of these equivalence classes, then define the product $\mathscr{C}_1\mathscr{C}_2$ of these classes by

$$\mathscr{C}_3 = \mathscr{C}_1\mathscr{C}_2 = \{\mathfrak{c} = \mathfrak{a}\mathfrak{b} | \mathfrak{a} \in \mathscr{C}_1, \mathfrak{b} \in \mathscr{C}_2\}.$$

Since the ideals of $\mathcal{O}_K$ form an Abelian group G of order h_K, we have that the ideals of $\mathcal{O}_{K,p}$ form an Abelian group with the structure of $G/\langle\mathfrak{p}\rangle$, which is consistent with Proposition 1.3, noting the fact that the identity of this group is the class of principal ideals.

From page 389 of [20], we know that

$$h_K \leq \frac{2}{\pi} \cdot |D|^{1/2} \left(1 + \log\left(\frac{2|D|^{1/2}}{\pi}\right)\right),$$

and for any $\varepsilon > 0$,

$$h_K > |D|^{1/2-\varepsilon}$$

for sufficiently large D. By assuming the Generalized Riemann Hypothesis (GRH), Littlewood [18] proved that

$$\frac{\pi(1+o(1))\sqrt{|D|}}{12e^{\gamma}\log\log|D|} < h_K < \frac{2(1+o(1))\sqrt{|D|}\log\log|D|}{\pi}.$$

Thus, we would expect the value of h_K to be approximately the same magnitude as $|D|^{1/2}$. Therefore, with the careful selection of prime p such that $|D|$ is large and $(p) = \mathfrak{p}\overline{\mathfrak{p}}$ such that $o(\mathfrak{p})$ is small, we have $h_{K,p}$ is large from Proposition 1.3. Therefore, the class group of ideals of $\mathcal{O}_{K,p}$ will be large enough.

If Eve was attempting to attack Alice and Bob's secure communication, she will have access to the value of D, p, and the ideals $\mathfrak{a}, \mathfrak{b}$, and $\mathfrak{c}$. She will win the game if she is able to recover either the secret value x or y. In order to do this is to figure out the value of $\widetilde{\mathfrak{f}_1}$ or $\widetilde{\mathfrak{f}_2}$. One approach is to solve the discrete log problem in $\mathrm{Cl}_K / \langle \mathfrak{p} \rangle$. In otherwords, given a reduced ideal $\mathfrak{a}$ and reduced ideal $\mathfrak{b}$ such that $\mathfrak{b} \sim \mathfrak{a}^x$, recover the value x. Trying to find the value of x is quite a difficult problem. The best known method to compute h_K is $O(|\Delta|^{1/4+\varepsilon})$ on average and $O(|\Delta|^{1/5+\varepsilon})$ at best [2].

Although the difficulty of this problem is not entirely known, Shanks proved in [22] that if there is an effective way to solve for x, then it will lead to an effective method to recover the factors of D. Therefore, this lets us believe that the difficulty of solving for x should be on a similar difficulty level as factoring a number. For more in depth analysis, see Section 4 of [2].

Chapter 7

Open Questions

7.1 Infrastructure

In this section, we describe a brief outline of the infrastructure of the class group of real quadratic fields followed by our desire of developing a similar notion for fake real quadratics. The author recommends for people interested in reading more about this topic are encouraged to view Jacobson and Scheidler's *Structure Inside the Class Group of a Real Quadratic Field* [16].

The infrastructure for real quadratic fields is the set of reduced principal ideals that forms a group-like structure. Specifically, let $\{\mathfrak{a}_i\}_{i=1}^n$ be the set of reduced ideals in the class group of $\mathbb{Q}(\sqrt{D})$, a real quadratic field. These ideals are equivalent giving us the following relationship:

$$\mathfrak{a}_{i+1} = (\psi_i)\mathfrak{a}_i.$$

The generators for the ideals $\mathfrak{a}_i$ and the values ψ_i are generated from intermediate steps computing the continued fractions expansion for $\sqrt{D}$. This fact is especially nice since otherwise finding these ideals would be a difficulty endeavor. Since the ideals $\mathfrak{a}_i$ are principal, we can write them as $\mathfrak{a}_i = (\theta_i)$, and we have the following relationship regarding the generators θ_i and ψ_i:

$$\theta_i = \prod_{k=1}^{i-1} \psi_k.$$

From this equality, we are able to say the *distance* between two ideals are

$$\delta(\mathfrak{a}_{i+1}, \mathfrak{a}_i) = \log(\theta_{i+1}/\theta_i) = \log \psi_i.$$

Although these ideals do not form a group, they have a cyclic "group-like" structure. It should not be surprising that the reduced ideals generated from the continued fractions expansion will start to repeat as quadratic irrationals have periodic expansions. The best way to see this cyclic group-like structure is to place the ideals on a circle, as seen in Figure 7.1 below.

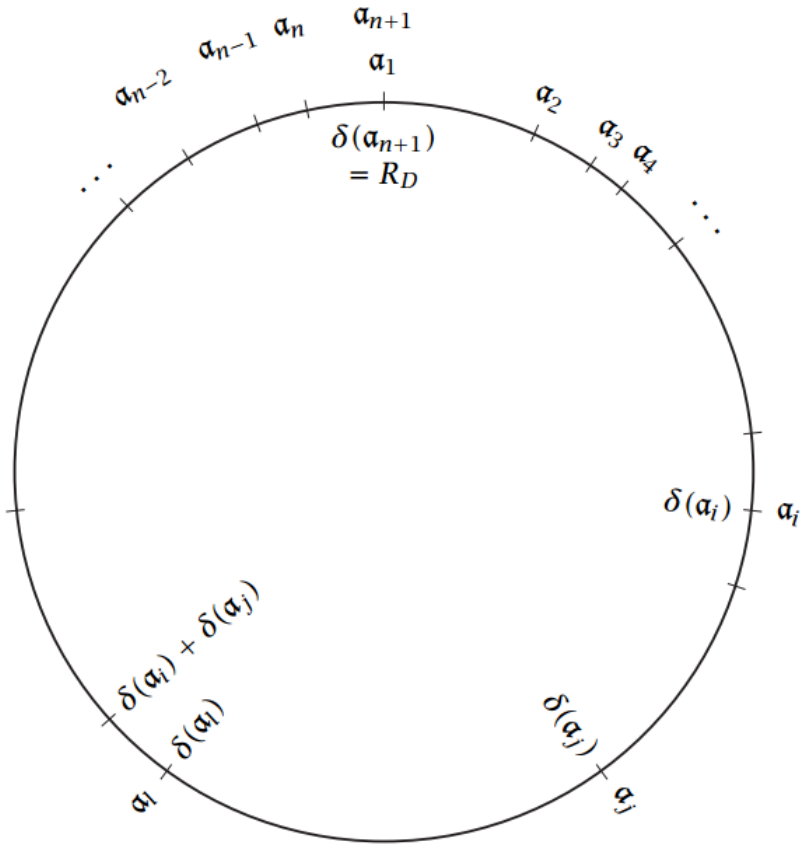

Figure 7.1 The Infrastructure of the Class Group of Real Quadratic Fields
Image Credit to Jacobson and Scheidler [16]

Although the distance between any two ideals are not necessarily equal, the amazing part is the circumference of this circle is the regulator $R_D = \log \varepsilon_D$ from Dirichlet's class number formula. The infrastructure is a powerful method to represent the class group structure of a real quadratic field as it is easy to generate the reduced ideals and to compute the regulator R_D using the continued fractions expansion of $\sqrt{D}$.

Since we believe the class group structure of fake real quadratics and real quadratic fields are similar, we suspect an analogous infrastructure argument could exist. Although a concerted effort was made to develop a comparable system, we fell short of our goal. The first difficulty encountered was to find the reduced ideals for fake real quadratics. For real quadratic fields, the reduced ideals could be found using continued fractions as discussed above. Another method to find these ideals employ the use of all binary quadratic forms of discriminant D and use Gauss's reduction theory. In contrast, the only tools to find these ideals in the fake real quadratic order was to perform an exhaustive search. This proved to be a laborious task. Furthermore, it was never clear on whether all the reduced ideals were found as the number of ideals in each cycle can widely vary. During this search, the question of whether an equivalent continued fractions expansion was raised, and this idea will be discussed in the next section.

Assuming that we had a complete set of reduced ideals, the next problem we encountered was the desired sum of distances between ideals to be equal to regulator $R_{D,p} = \log_p \varepsilon_{D,p}$. Unfortunately, the sum of distances computed between the reduced ideals never corresponded to the desired value $R_{D,p}$ in any discernible pattern. Considering this, it seemed apparent we needed a different approach to the development of our infrastructure.

Nevertheless, the author is convinced that an infrastructure for the class group of fake real quadratic orders should exist. It seems dubious for the class group structures to be this similar yet fail to have an infrastructure counterpart. In addition, there was no contrary evidence produced to convince the author otherwise.

7.2 Continued Fractions

As stated in the previous section, continued fractions expansion provide incredible amounts of information for real quadratic fields $\mathbb{Q}(\sqrt{D})$. For the infrastructure, it

yields the generators for the reduced principal ideals along with all the necessary parts to compute the regulator R_D. Even more interesting, the expansion can produce the fundamental unit ε_D for $\mathbb{Q}(\sqrt{D})$. Thus, it would be desired to have an equivalent expansion for fake real quadratic orders.

Both the infrastructure and the fundamental unit for $\mathbb{Q}(\sqrt{D})$ rely on the periodic nature of the continued fractions expansion for quadratic irrationals. Unfortunately, the continued fractions algorithm is only defined for real numbers. Since fake real quadratrics are complex algebraic structures, we must use another method to compute such an expansion.

The search for a continued fraction expansion counterpart led us to p-adics. The first major difficulty was to decide which process to use as many algorithms exist due to the variety of "floor" definitions for the p-adics. For real numbers, the floor of $n \in \mathbb{R}$ is the greatest integer part of n, denoted by $\lfloor n \rfloor$. In the p-adics, there is no clear equivalent definition for the floor. They were often some variation of the p-adic valuation of the number.

We were specifically looking for an expansion that was periodic. The only expansion we encountered with this possibility was Justin Miller's algorithm from his Ph.D. dissertation [19]. He proved that quadratic irrationals were periodic in $\mathbb{Q}_2$ and $\mathbb{Q}_3$ and observed that many quadratic irrationals were periodic in $\mathbb{Q}_p$ using his method. Unfortunately, his expansion did not provide a similar notion of reduced ideals or a fundamental unit for fake real quadratics. Recall from Chapter 2 that a fundamental unit for $\mathcal{O}_{D,p}$ should solve the Pell Equation

$$x^2 - Dy^2 = p^k,$$

where $k = o(\mathfrak{p})$. As with most other p-adic continued fractions expansion, the numerator and denominators of the convergents solved the Diophantine equation

$$x^2 - Dy^2 = zp^k.$$

It was observed that some of these expansion did solve the desired Pell equation, but there seemed to be no discernible pattern on why this was true. After a while, it seemed apparent that it was sheer coincidence that the expansion happen to solve the Diophantine equation with $z = 1$. We attempted to isolate and understand if it was possible to modify the algorithm to force $z = 1$ everytime. Unfortunately, we fell short of our goal here as well as there was no strong evidence explaining why $z = 1$. We hope to inspire the reader to pursue the development of an equivalent continued fractions expansion for fake real quadratic orders.

7.3 Class Number Problem

It would be remiss if we did not mention an analogue to Gauss's class number problem in terms of fake real quadratics.

Conjecture 7.1. *(O.) Let p be a fixed prime. Then there are infinitely many fundamental discriminants D such that $\left(\frac{D}{p}\right) = 1$ such that $h_{K,p} = 1$.*

It would be interesting to see if there was any logic argument why the element in the class group Cl_K representing the element p would not behave randomly. One of the largest hurdles for large $|D|$ would be the difficulty of computing the order of $\mathfrak{p}$ in the class group Cl_K. This knowledge is required to compute the class number $h_{K,p}$. If there was an efficient method to compute the order, then one might be better equipped to produce and study the class numbers generated for large $|D|$.

Bibliography

[1] Eric Bach, *Explicit bounds for primality testing and related problems*, Mathematics of Computation **55** (1990), no. 191, 355–380.

[2] J. Buchmann and H. C. Williams, *A key-exchange system based on imaginary quadratic fields*, J. Cryptol. **1** (1988), no. 2, 107–118.

[3] Johannes A. Buchmann and Hugh C. Williams, *A key exchange system based on real quadratic fields*, Proceedings on Advances in Cryptology (New York, NY, USA), CRYPTO '89, Springer-Verlag New York, Inc., 1989, pp. 335–343.

[4] D.A. Buell, *Binary quadratic forms: Classical theory and modern computations*, Springer, 1989.

[5] Fernando Chamizo and Adrián Ubis, *An average formula for the class number*, ACTA ARITHMETICA-WARSZAWA- **122** (2006), no. 1, 75.

[6] Jia Chaohua, *The distribution of square-free numbers*, Acta Sci Natur Univ Pekinensis **3** (1987), 21–27.

[7] Henri Cohen, *Fake real quadratic orders*, Unpublished Manuscript, 2013.

[8] Henri Cohen and Hendrik W Lenstra Jr, *Heuristics on class groups of number fields*, Number Theory Noordwijkerhout 1983, Springer, 1984, pp. 33–62.

[9] D.A. Cox, *Primes of the form $x^2 + ny^2$: Fermat, class field theory, and complex multiplication*, Pure and Applied Mathematics: A Wiley Series of Texts, Monographs and Tracts, Wiley, 2011.

[10] Harold Davenport and Hans Heilbronn, *On the density of discriminants of cubic fields. ii*, Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences (1971), 405–420.

[11] Taher El Gamal, *A public key cryptosystem and a signature scheme based on discrete logarithms*, Proceedings of CRYPTO 84 on Advances in Cryptology (New York, NY, USA), Springer-Verlag New York, Inc., 1985, pp. 10–18.

[12] Jordan S Ellenberg and Akshay Venkatesh, *Reflection principles and bounds for class group torsion*, International Mathematics Research Notices **2007** (2007), rnm002.

[13] Dorian Goldfeld, *GaussâĂŹ class number problem for imaginary quadratic fields*, Bulletin of the American Mathematical Society **13** (1985), no. 1, 23–37.

[14] H. Heilbronn, *On the class-number in imaginary quadratic fields*, Oxford University Press, 1934.

[15] H. Heilbronn and E.H. Linfoot, *On the imaginary quadratic corpora of class-number one*, Oxford University Press, 1934.

[16] MJ Jacobson Jr and R Scheidler, *Infrastructure: Structure inside the class group of a real quadratic field*, Notices of the AMS **61** (2014), no. 1.

[17] Jeffrey C Lagarias and Andrew M Odlyzko, *Effective versions of the chebotarev density theorem*, Algebraic number fields (1977), 409–464.

[18] J.E. Littlewood, *On the class number of the corpus* $p(\sqrt{-k})$, Proc. London Math. Soc. (1928), 358–372.

[19] Justin Miller, *On p-adic continued fractions and quadratic irrationals*, ProQuest, 2007.

[20] W. Narkiewicz, *Elementary and analytic theory of algebraic numbers*, Springer Monographs in Mathematics, Springer, 2004.

[21] Carl Pomerance and Manjul Bhargava, *Counting fields*, (2010).

[22] Daniel Shanks, *Class number, a theory of factorization, and genera*, Proceedings of Symposia in Pure Mathematics, vol. 20, AMS. Providence, RI, 1971, pp. 415–440.

[23] ———, *Five number-theoretic algorithms*, Proceedings of the Second Manitoba Conference on Numerical Mathematics, 1972, pp. 51–70.

[24] W. A. Stein et al., *Sage Mathematics Software (Version x.y.z)*, The Sage Development Team, 2014, `http://www.sagemath.org`.

[25] Gonzalo Tornaría, *Square roots modulo p*, LATIN 2002: Theoretical Informatics (Sergio Rajsbaum, ed.), Lecture Notes in Computer Science, vol. 2286, Springer Berlin Heidelberg, 2002, pp. 430–434 (English).

[26] A van der Poorten, H te Riele, and H Williams, *Computer verification of the ankeny–artin–chowla conjecture for all primes less than 100000000000*, Mathematics of computation **70** (2001), no. 235, 1311–1328.

[27] Mark Watkins, *Class numbers of imaginary quadratic fields*, Math. Comp. **7** (2004), 907–938.

Appendix A

Fundamental Unit Method 1 SAGE Code

The following code uses SAGE [24] to find the fudamental unit for a given p prime and fundamental discriminant $D < 0$. The program follows Method 1 uses the following steps:

1. Computes the class number $h(D)$ for $\mathbb{Q}(\sqrt{D})$.
2. Determines all divisors of $h(D)$.
3. Iterates through all the divisors of $h(D)$ starting from the smallest divisor and determines if p^k is represented by the binary quadratic form $x^2 - Dy^2$.

The smallest divisor k such that there exists a (x, y) satisfying the Pell equation

$$x^2 - Dy^2 = p^k$$

gives the fundamental unit $x + y\sqrt{D}$.

```
import csv
import numpy
import pylab as p
import matplotlib
import math


def class_number_frqf_precision(p, D, prec = 2):
   ‘‘‘Program searches through all possible divisors of the class
```

```
number
Determines whether there is a solution to the Pell Equation
x^2 - D*y^2 = p*k, where k is the order of p in the class group
\n
Usage: \n
D: D < 0 \n
p: fixed prime \n
\n
Outputs:
(x, y, k)
where x^2 - D*y^2 = p^k \n '''
# Checks to see if p is prime
if is_prime(p) == False:
    return "ERROR> " + str(p) + " is not a prime number."
# Checks to see if D is a negative value
elif D >= 0:
    return "ERROR> D should be < 0."
# Checks to see if D is a square modulo p
elif  legendre_symbol(D, p) != 1:
    return "ERROR> " + str(D) + " is not a quadratic residue
    modulo " + str(p) + "."
elif is_prime(-D) != True or (-D%4) != 3:
    return "ERROR> " + str(D) + " is not a prime or -D not
    \equiv 3 (mod 4)."
# Creation of the Number Field \Q(\sqrt(D)) and its class group
K = QuadraticField(D)
h = K.class_number()
```

```
# p O_K = \mathfrak{p} \bar{\mathfrak{p}}, let A = \mathfrak{p}
A = K.factor(p)[0][0]
# value holds the value for the order of \mathfrak{p} in Class
# Group of K
value = 0
# creation of PARI number field
bnf = A.number_field().pari_bnf()
# goes through every possible divisor of the class number
for k in h.divisors():
    # temp holds the possible ideal, checks whether to see if it
    # is principal
    temp = A**k
    # if temp is principal, then break, otherwise, check next
    # divisor
    pari_ideal_challenge = bnf.bnfisprincipal(temp.pari_hnf(), prec)
    if sum(list(pari_ideal_challenge)) == 0:
        value = k
        break
# Return D, p, h_{K,p}, h_K, ord(p) in Cl(K), fundamental
#  unit
return D, p, h/value, h, value, (A**value).gens_reduced()[0]
```

Appendix B

Fundamental Unit Method 2 SAGE Code

The following code uses SAGE [24] to determine the fundamental unit for the fake real quadratic order generated by prime p and the fundamental discriminant $D < 0$. The SAGE code listed here uses the method described in Section 2.2, where the order of the class group of the fake real quadratic is determined by binary quadratic forms and through matrix reduction delivers the fundamental unit.

The function `binary_quadratic_order(D,p)` uses the built in functionality of SAGE to determine the order of the prime fractional ideal $\mathfrak{p}$ where $(p) = \mathfrak{p}\overline{\mathfrak{p}}$ in Cl_K, the class group of the imaginary quadratic field $K = \mathbb{Q}(\sqrt{D})$.

The function `square_root_mod(n,p)` computes the square root of r modulo p, or solutions to

$$x^2 \equiv r \pmod{p}$$

using the Tonelli-Shanks algorithm discusses in Section 2.4.

The function `quadratic_nonresidue(p)` scans the set $\{0, 1, \ldots, p-1\}$, starting from 0 increments by 1, stopping when it has found a quadratic non-residue. This function is necessary for the Tonelli-Shanks algorithm. Returns a quadratic non-residue.

During the process of computing the order of the ideal $\mathfrak{p}$, we need to check whether or not each power of the given binary quadratic form is equivalent to the principal binary quadratic field. If the D is not either 0 or 1 $\pmod 4$, then the principal binary quadratic form has not been implemented.

The following function reduces a given binary quadratic form $Q = (a, b, c)$ with

discriminant D so that it satisifies $|b| \leq a < c$. It is programmed to do the following: Given a binary quadratic form $Q = (a, b, c)$, we look at it as a matrix

$$Q = \begin{bmatrix} a & b/2 \\ b/2 & c \end{bmatrix}.$$

Then find the matrix M such that

$$M^{-1}QM = \begin{bmatrix} A & B/2 \\ B/2 & C \end{bmatrix}$$

where (A, B, C) represents the reduced principal binary quadratic form with discriminant D.

The function `delta_value(b,c,D)` is used in conjuction with the previous function `BinaryQF_reduction_matrix(Q)`. The transformation matrix has the form

$$\begin{bmatrix} 0 & -1 \\ 1 & \delta \end{bmatrix}$$

where $\delta \in \mathbb{Z}$ is the value that satisfies

$$| - b + 2c\delta| \leq c,$$

where b and c are from the binary quadratic form (a, b, c) in which it is being reduced. This function will find the δ value necessary for the step of reduction that will reduce the binary quadratic form.

```
def binary_quadratic_order(D, p):
    '''This program uses binary quadratic forms
    with the built-in functionality of SAGE to
    determine the order of \mathfrak{p} in the
    class group Cl(K) where K = Q(\sqrt{D}).
    Also will return the fundamental unit for
    the fake real quadratic field O_{K,p} where
```

```
K is the imaginary quadratic field defined
the same as above.\n
\n
\n
D: the D value for Q(\sqrt{D}) \n
p: the prime that creates the fake real quadratic'''
# Check whether D is a square modulo p
if legendre_symbol(D,p) != 1:
    return "ERROR> " + str(D) + " is not a square modulo " + str(p)
# Compute the binary quadratic form with disc D
# with leading coefficient p.
# Quadratic Form (a,b,c) = (p,b,c)
# leading coefficient
a = p
# computing b and c value
square_root_values = square_root_mod(D, p)
c = (int(square_root_values[0])**2 - D)/(4*a)
if int(c) == c:
    b = int(square_root_values[0])
else:
    b = int(square_root_values[1])
    c = (b**2 - D)/(4*a)
# creation of the binary quadratic form of disc D
# with leading coefficient p
Q = BinaryQF([a,b,c])
# creation of principal binary quadratic form of
# disc D
```

```
    P = BinaryQF_principal(D)
    # Checks to make sure principal QF is sane
    if type(P) == str:
        return "ERROR> Not Implemented Yet."
    # variable to keep track of order
    order = 1
    # temp variable to hold the current binary
    # QF in question
    temp = Q
    # loop until Q is the principal form
    while temp.is_equivalent(P) == False:
        temp *= Q
        order += 1
    # compute the reduction matrix and fundamental unit
    data = BinaryQF_reduction_matrix(temp)
    # returns the fundamental unit, order of \mathfrak{p}
    # in class group K, binary quadratic form, and
    # reduction matrix
    return data[2], order, temp, data[1]

def square_root_mod(n, p):
    '''This program computes the square root of r modulo p
    and returns the two solution. Uses the Tonelli-Shanks
    algorithm.'''
    # checks that r is a square modulo p
    if legendre_symbol(n, p) != 1:
        return "ERROR> " + str(n) + " is not a square modulo " + str(p)
    # p-1 = Q*2^S
```

```
S = (p-1).trailing_zero_bits()
Q = int((p-1)/2**S)
# Retrieve a quadratic non-residue modulo p
z = quadratic_nonresidue(p)
# Define the following values for R, t, and M
R = mod(n**((Q+1)/2), p)
t = mod(n**Q, p)
M = S
# Set value of c
c = mod(z**Q, p)
# checks the first step of the loop
if t == 1:
    return R, p-R
# iterate over the following loop
while t != 1:
    # find lowest i, 0<i<M, such that t^(2^i) \equiv 1
    # mod p
    i = 1
    end_condition = False
    while end_condition == True:
        if mod(t**(2**i), p) != 1:
            i += 1
        else:
            end_condition = True
    b = mod(c**(2**(M-i-1)), p)
    # Set the following congruences
    R = mod(R*b, p)
```

```
        t = mod(t*(b**2), p)
        c = mod(b**2, p)
        M = i
    return R, p-R

def quadratic_nonresidue(p):
    '''This program finds a random element in {0,1,...,p-1}
    that is a non-residue and returns it.'''
    # Starts at the beginning of the set and searches for a
    # quadratic non-residue
    for i in xrange(p-1):
        # checks to see if a quadratic non-residue
        if legendre_symbol(i,p) == -1:
            # returns quadratic non-residue
            return i

def BinaryQF_principal(D):
    '''Returns the reduced principal binary quadratic
    forms with discriminant D.'''
    if D%4 == 0:
        return BinaryQF([1,0,-D/4])
    elif D%4 == 1:
        return BinaryQF([1,1,-(D-1)/4])
    else:
        return "ERROR> Not Implemented Yet."

def BinaryQF_reduction_matrix(Q):
    '''This program attempts to reduce a binary quadratic
    form Q = (a,b,c). While it reduces, it keeps track of
```

```
the reduction matrices used and produces the entire
matrix at the end.'''
# Converts from SAGE BinaryQF type to list
L = list(Q)
# Creation of Binary Quadratic Field
K = QuadraticField(Q.discriminant(), 'a')
# stores all the ideal necessary
reduction_BQF = []
reduction_BQF.append((-L[1]+K('a'))/(2*L[0]))
# list delta holds all delta values (matrix entry 2,2)
# used to transform matrix
delta = []
# Loops to see if L is reduced (without using built
# in functionality
# loop while binary quadratic form is not reduced
while (abs(L[1]) <= L[0] < L[2]) == False:
    # find the new b and new c value along with delta
    temp = delta_value(L[1], L[2], Q.discriminant())
    # append the compute delta value in temp into list
    delta.append(temp[0])
    # save the new binary quadratic form into list
    L[0] = L[2]
    L[1] = temp[1]
    L[2] = temp[2]
    # adds the new binary quadratic form to
    # reduction_BQF list
    reduction_BQF.append((-L[1]+K('a'))/(2*L[0]))
```

```
    # creation of the identity 2x2 matrix
    M = Matrix.identity(2)
    # for each d in delta value
    for d in delta:
        # create the transformation matrix for that
        # step of reduction
        temp = Matrix([[0,-1],[1,d]])
        # multiply it to the product of all these
        # transformation matrices
        M *= temp
    # Computes the fundamental unit
    fund_unit = 1
    for j in xrange(len(reduction_BQF)-1):
        fund_unit *= reduction_BQF[j]
    fund_unit = fund_unit**(-1)
    # return reduction transformation matrix, its inverse,
    # multiplied by -1,
    # the list of BQF involved with reduction
    return M, -M.inverse(), fund_unit

def delta_value(b, c, D):
    '''Compute the valid delta value used to transform
    matrix to reduce a binary quadratic form.'''
    # start at the smallest d (i.e. delta) value
    d = 0
    # starting value of the new b
    new_b = -b + 2*c*d
    # counter to decide which value of d to try
```

```
# allow alternation between plus or minus d
counter = 0
while (abs(new_b) <= c) == False:
    # when counter is even, increase the
    # magnitude of delta by +1
    # find the new b value and recheck if
    # it works
    if counter%2 == 0:
        d = abs(d) + 1
        counter += 1
        new_b = -b + 2*c*d
    # when counter is odd, check the
    # negative form of delta, find the
    # new b value and recheck if it works
    else:
        d = -abs(d)
        counter += 1
        new_b = -b + 2*c*d
# once a valid new b value has been found
# compute the new c value
new_c = (D - new_b**2)/(-4*c)
# return the delta value, new b, and new c
return d, new_b, new_c
```

Appendix C

C-L/Hooley/Class Number Scanner

We attempted to investigate an analogue to the Hooley conjecture which states that

$$\sum_{D<X} h(D) \sim X/8.$$

We fell short of our goal for showing an analogous statement, but we list here the code developed in the attempt to generate data and trendlines.

The code listed here uses SAGE [24] to perform a handful of tasks that provide data for fake real quadratic orders. The function `neg_disc_range` generates a list of negative fundamental discriminants in the range [-D, -2), unless a different value for the lower bound is specified other than -2.

The function `class_number_prec_scanner` take a given list of negative fundamental discriminants and primes, and for each valid combination of p and D (i.e. $\left(\frac{D}{p}\right) = 1$), it computes the class number for the fake real quadratic order $\mathcal{O}_{K,p}$ where $K = \mathbb{Q}(\sqrt{D})$. It will record the frequency of each instance of a class number for each prime and write it out to a file.

The function `class_number_one_scanner` operates similarly to the `class_number_prec_scanner`, except checks only to see if the class number for the fake real quadratic order is one. If the class number is one, then it will write it to the file.

For the function `fake_real_hooley`, it takes a list of primes and functional discriminants and for valid combinations of p and D, computes partial sums for

$$\sum_{D=-2}^{x} h(D,p).$$

This function aims to see if there is a visual analogue to the Hooley conjecture for fake real quadratic orders. It will plot the data for a specified number of partial sums on a graph.

The function `class_number_formula` is used to compute the class number using the class number fomrula rather than the brute force methods listed in Method 1 and Method 2. Unfortunately, since there is not an efficient way to compute the fundamental unit, it still requires some data to be computed using Method 1.

The function `class_number_formula_scanner` uses the function `class_number_formula` function. Given a list of primes and fundamental discriminants, the function will produce a file that contains all valid combinations of p and D and the class number for the fake real quadratic order that is produced by them. The goal is the data can be post-processed in order to identify additional patterns.

```
def neg_disc_range(D, lower_bound = -2):
    '''Returns a list that contains all fundamental negative
 discriminants in the range [-D, lower_bound). '''
    if D <= -lower_bound:
        print "ERROR> Upper bound must be a postive value greater
 than", -lower_bound
        return
    # Create list that will hold all negative fundamental discriminants
    disc_list = [0] * D
    # atomic counter to keep track of which entry of the disc_list to
 save valid values
    counter = 0
    # iterate through range [-D, lower_bound). If fundamental
discriminant, place into
    # disc_list[counter].
```

```
    for i in xrange(-D, lower_bound):
        if is_fundamental_discriminant(i) == True:
            disc_list[counter] = i
            counter += 1
    # Returns disc_list, with only valid entries, with smallest to
largest negative disc
    return disc_list[counter-1::-1]

def class_number_prec_scanner(nth_prime_end, upper_bound_disc,
filename_raw, nth_prime_start = 1, lower_bound_disc = -2, print_out
= False, class_number_upperbound = 10**4, prec = 2):
    # Creates the list of primes based on the inputs
    P = primes_first_n(nth_prime_end)[nth_prime_start:]
    # Creates the list of negative fundamental discriminants based on
 the inputs
    # D = neg_disc_range(upper_bound_disc, lower_bound_disc)
    D = xrange(-upper_bound_disc, lower_bound_disc)
    # Creation of the file, in write mode
    filename = str(filename_raw) + ".txt"
    open_file = open(filename, 'w')
    # Creation of the csv writer for the given filename
    spamwriter = csv.writer(open_file)
    spamwriter.writerow(["Primes number " + str(nth_prime_start) +
" to " + str(nth_prime_end) +  ", Discriminants in range " +
str(lower_bound_disc-1) + " through " + str(-upper_bound_disc)])
    # Scan through each prime, range through each disc, and print to
 file class number
```

```
    for p in P:
        # writes the current prime out to file
        spamwriter.writerow([str(p)])
        # array keeps track of the number of each class number for a
given prime
        class_number_array = numpy.zeros(class_number_upperbound)
        # counter keeps track of total number of frqf for a given
prime
        counter = 0
        # Goes through each negative disc in given range
        for d in D:
            # retrieves the class number h_{K,p}
            data = class_number_frqf_precision(p, d, prec = 2)
            # If there is an error is the prime/discriminant combo,
continue
            if type(data) == str:
                continue
            # Else, print out to screen or file, pending on command
options
            else:
                # counts each negative disc for a given prime
                counter += 1
                if print_out == True:
                    spamwriter.writerow(["disc: " + str(d) + "
, h_{K,p} = " + str(int(data[0])), data[1], data[2]])
                # For each class number, add one to the counter
                class_number_array[int(data[0])] += 1
```

```
        # For each non-zero entry, print to file
        for i in (class_number_array.nonzero())[0]:
            # Print to file each class number and its number of
occurences
            spamwriter.writerow(["Class number: " + str(i) + ",
Number of Occurences: " + str(int(class_number_array[i]))])
        spamwriter.writerow(["Total FRQF: " + str(counter)])
    # Close file
    open_file.close()
    return "Finished!"

def class_number_one_scanner(nth_prime_end, upper_bound_disc,
filename_raw, nth_prime_start = 1, lower_bound_disc = -2,
print_out = False, class_number_upperbound = 10**4):
    # Creates the list of primes based on the inputs
    P = primes_first_n(nth_prime_end)[nth_prime_start:]
    # Creates the list of negative fundamental discriminants based
 on the inputs
    D = neg_disc_range(upper_bound_disc, lower_bound_disc)
    # Creation of the file, in write mode
    filename = str(filename_raw) + ".txt"
    open_file = open(filename, 'w')
    # Creation of the csv writer for the given filename
    spamwriter = csv.writer(open_file)
    spamwriter.writerow(["Primes number " + str(nth_prime_start) +
" to " + str(nth_prime_end) +  ", Discriminants in range " +
str(lower_bound_disc-1) + " through " + str(-upper_bound_disc)])
```

```
    # Scan through each prime, range through each disc, and print to
file class number
    for p in P:
        # writes the current prime out to file
        spamwriter.writerow([str(p)])
        # creates two counters. one for class number one, and one
for total number of frqf
        counter_one = 0
        counter_frqf = 0
        # Goes through each negative disc in given range
        for d in D:
            # retrieves the class number h_{K,p}
            data = class_number_frqf(p, d)
            # If there is an error is the prime/discriminant combo,
continue
            if type(data) == str:
                continue
            # Else, print out to screen or file, pending on command
options
            else:
                # Counts each valid frqf
                counter_frqf += 1
                if print_out == True:
                    spamwriter.writerow(["disc: " + str(d) + ",
h_{K,p} = " + str(int(data[0]))])
                # If class number one, add to counter. Otherwise,
go to next disc
```

```
                if int(data[0]) == 1:
                    counter_one += 1
                else:
                    continue
        # Print to file each class number and its number of occurences
        spamwriter.writerow(["Total class number 1: " +
str(counter_one) + ", Total FRQF: " + str(counter_rfqf)])
    # Close file
    open_file.close()
    return "Finished!"

def class_number_prec_scanner(nth_prime_end, upper_bound_disc,
filename_raw, nth_prime_start = 1, lower_bound_disc = -2,
print_out = False, class_number_upperbound = 10**4, prec = 2):
    # Creates the list of primes based on the inputs
    P = primes_first_n(nth_prime_end)[nth_prime_start:]
    # Creates the list of negative fundamental discriminants based on
the inputs
    # D = neg_disc_range(upper_bound_disc, lower_bound_disc)
    D = xrange(-upper_bound_disc, lower_bound_disc)
    # Creation of the file, in write mode
    filename = str(filename_raw) + ".txt"
    open_file = open(filename, 'w')
    # Creation of the csv writer for the given filename
    spamwriter = csv.writer(open_file)
    spamwriter.writerow(["Primes number " + str(nth_prime_start) +
" to " + str(nth_prime_end) +  ", Discriminants in range "
```

```
    + str(lower_bound_disc-1) + " through " + str(-upper_bound_disc)])
    # Scan through each prime, range through each disc, and print to
file class number
    for p in P:
        # writes the current prime out to file
        spamwriter.writerow([str(p)])
        # array keeps track of the number of each class number for
a given prime
        class_number_array = numpy.zeros(class_number_upperbound)
        # counter keeps track of total number of frqf for a given
prime
        counter = 0
        # Goes through each negative disc in given range
        for d in D:
            # retrieves the class number h_{K,p}
            data = class_number_frqf_precision(p, d, prec = 2)
            # If there is an error is the prime/discriminant combo,
continue
            if type(data) == str:
                continue
            # Else, print out to screen or file, pending on command
options
            else:
                # counts each negative disc for a given prime
                counter += 1
                if print_out == True:
                    spamwriter.writerow(["disc: " + str(d) + ",
```

```
h_{K,p} = " + str(int(data[0])), data[1], data[2]])
                # For each class number, add one to the counter
                class_number_array[int(data[0])] += 1
        # For each non-zero entry, print to file
        for i in (class_number_array.nonzero())[0]:
            # Print to file each class number and its number of
occurences
            spamwriter.writerow(["Class number: " + str(i) + ",
Number of Occurences: " + str(int(class_number_array[i]))])
        spamwriter.writerow(["Total FRQF: " + str(counter)])
    # Close file
    open_file.close()
    return "Finished!"

def fake_real_hooley(nth_prime_end, upper_bound_disc, filename_raw,
nth_prime_start = 1, lower_bound_disc = -2, class_number_upperbound
= 10**4, prec = 2):
    # Creates the list of primes based on the inputs
    P = primes_first_n(nth_prime_end)[nth_prime_start:]
    # Creates the list of negative fundamental discriminants based
on the inputs
    # D = neg_disc_range(upper_bound_disc, lower_bound_disc)
    D = xrange(-upper_bound_disc, lower_bound_disc)
    # Creation of the file, in write mode
    filename = str(filename_raw) + ".txt"
    # Create the filename for the picture
    filename_pic = str(filename_raw) + "_hooley.png"
```

```
    # Open text file name is write mode
    open_file = open(filename, 'w')
    # Creation of the csv writer for the given filename
    spamwriter = csv.writer(open_file)
    spamwriter.writerow(["Primes number " + str(nth_prime_start)
+ " to " + str(nth_prime_end) +  ", Discriminants in range " +
str(lower_bound_disc-1) + " through " + str(-upper_bound_disc)])
    # Creation of the x-axis for the plot
    X = [i for i in xrange(upper_bound_disc)]
    # Creation of the primary plot
    fig = p.figure(figsize = (20, 15), dpi = 500)
    # Creation of the actual subplot on the graph
    ax = fig.add_subplot(1, 1, 1)
    # x-axis label
    ax.set_xlabel("Disc < " + str(upper_bound_disc))
    # y-axis label
    ax.set_ylabel("H(p, X)")
    # Set plot Title
    ax.set_title("Hooley Data for Fake Real")
    # Specify the colors for plot
    colors = ['b', 'g', 'r', 'c', 'm', 'y', 'k', 'w']
    # Specify the two markers
    markers = ['o', 'x']
    # Color counter
    color_counter = -1
    # marker counter
    marker_counter = 0
```

```
    # prime array to hold labels
    prime_label_array = [0] * len(P)
    for i in len(P):
        prime_label_array[i] = ['subplot_' + str(P[i]), str(P[i]),
'placeholder']
    # Holds the interval for each sample
    interval = int(len(D)*.05)
    # Scan through each prime, range through each disc, and print to
file class number
    for p in P:
        # if loop to advance the counter
        if color_counter == 8:
            color_counter = 0
            markers = 1
        else:
            color_counter += 1
        # array keeps track of the number of each class number for
a given prime
        class_number_array = numpy.zeros(class_number_upperbound)
        # counter keeps track of total number of frqf for a given
prime
        counter = 0
        # Goes through each negative disc in given range
        for j in len(D):
            # retrieves the class number h_{K,p}
            data = class_number_frqf_precision(p, D[j], prec = 2)
            # If there is an error is the prime/discriminant combo,
```

```
continue
            if type(data) == str:
                continue
            # Else, print out to screen or file, pending on command
options
            else:
                # counts each negative disc for a given prime
                counter += 1
                # For each class number, add one to the counter
                class_number_array[int(data[0])] += 1
            if j % interval == 0:
                sum = 0
                for i in (class_number_array.nonzero())[0]:
                    sum += i*class_number_array[i]
        # For each non-zero entry, print to file
        sum = 0
        for k in (class_number_array.nonzero())[0]:
            sum += k*class_number_array[k]
    # Close file
    open_file.close()
    return "Finished!"

def class_number_formula_data(p, D):
    # Produce data for seeing the class number is possible
    data = class_number_frqf_precision(p, D)
    if type(data) == str:
        return "ERROR> NOT A VALID p, D COMBINATION"
```

```
    log_abs_fund_unit = log(abs(data[5]))
    log_norm_fund_unit = log(norm(data[5]))
    return D, p, data[2], data[3], data[4], data[5],
RR(log_abs_fund_unit), RR(data[4]/log_abs_fund_unit),
RR(log_norm_fund_unit), RR(data[4]/log_norm_fund_unit)

def class_number_formula_scanner(D, number_of_primes):
    # Creation of filename
    filename = "disc" + str(D) + "_" + str(number_of_primes)
+ "primes.txt"
    # Open text file name is write mode
    open_file = open(filename, 'w')
    # Creation of the csv writer for the given filename
    spamwriter = csv.writer(open_file)
    spamwriter.writerow(["disc","prime", "h(disc,p)", "h(disc)",
"order", "fundamental unit", "log of abs value fund unit",
"order | by log abs fund unit", "log of norm fund unit",
"order | log norm fund unit"])
    # list of primes
    P = primes_first_n(number_of_primes)[1:]
    # range through all the possible primes
    for p in P:
        data = class_number_formula_data(p, D)
        if type(data) == str:
            continue
        else:
            spamwriter.writerow([data[0], data[1], data[2],
```

```
data[3], data[4], data[5], data[6], data[7], data[8], data[9]])
    # Close file
    open_file.close()
    return
```

Appendix D

Ankeny-Artin-Chowla Conjecture Additional Counterexamples

We list additional counterexamples to the AAC Conjecture. For each given p and D, we give the fundamental unit that violates the AAC conjecture for the fake real quadratic field $\mathcal{O}_{K,p}$, where $K = \mathbb{Q}(\sqrt{D})$. The counterexamples were found by Henri Cohen, and the fundamental units were generated by the SAGE program found in Appendix A.

$p = 31$, $D = -3$

$\varepsilon = 2 - 3\sqrt{-3}$

$p = 53$, $D = -738319$

$\varepsilon = 754097868001499134083892110353483885757836930922670268485722939807722934259895179307031589007257860127574031936702919153184117140630937117749951393427431337623376249273945672801840382176843642006903356482597486102891987110842990236091073194292791819031670688241298327529206429375936424857897195628833466760906937990938977865561898604394386827077293723924404513148623264318494549100612411431217264159491293851464142170485331162300461124249606463542251463693039757422358612638106737176276092193373420691429669238171856340781595706061510 7 − 270249065378176440719842774577289426748384222521987949667353922393731152996435798643965945567316847455254042214874992580284 17$

$$406132683159526267039912910152025181745421011615704474361685704021691474160280550136621314945796488612002493197880773450090609529112116010163867307164405266696606327443420089631986711132458624589661152734425879330384144477803251765716950473712523909380126085674827337337679110697570916246989900340705896129909502722989811764823103648138499279912058867783120914561441147183433866846152192562481975594914447854 \cdot \sqrt{-738319}$$

$p = 281,\ D = -1071739$

$$\varepsilon = (65604922183630814188030129877363936067286299153553367478206714548773873999651033016421742193658338976500042620771783298202269554843716465070278002582832996777219078140122420461271+1712305725005216325090687959213971113559543855641696688473794947423121614863499004534277472729212561278304548841459714698619851544718768553615791441284724566585912538782392485 \cdot \sqrt{-1071739})/2$$

$$p = 859,\ D = -3$$
$$\varepsilon = \frac{13 + 33 \cdot \sqrt{-3}}{2}$$

$p = 859,\ D = -23$

$\varepsilon = 25174 + 69 \cdot \sqrt{-23}$

$p = 859,\ D = -311$

$\varepsilon = 18623521659069151082475321 58 + 40982137779207120429432415 5 \cdot \sqrt{-311}$

$p = 859,\ D = -664019$

$$\varepsilon = 271586565716377666196986335096080570493168914916040981109466019099164496105003790497073622673924157452038256356492548775268231060960384424777569$$

$$0018868914185020494977098004120637171658059772646192371067346561951262601200891064941054633959899498078993114370122334195307491255675192035443342866759553041507229914236290079614440041873247012271038683884627359596614012425260410877607065415847120910164553728570522415331938925642519868600842878341780032-501709160306416570932654200312909150957892542465622665389653267970260009550646031766192364654852662551703430253040249829239450871739427419267286967111700431468941217165635561971661557034993366134460162098957119919756567713140668937576624372921905159412497405565542457383642693109995376044374077307074727256519828860606834860526854813486708456129403718303002743339077837706708946970091526721662718737429317616394007822587566559261977255170345651 \cdot \sqrt{-664019}$$