J. Cryptology (1994) 7: 171–199

Journal of
CRYPTOLOGY

A Key-Exchange Protocol Using Real Quadratic Fields

Renate Scheidler
Department of Mathematical Sciences, University of Delaware, Newark, DE 19716, U.S.A.

Johannes A. Buchmann
FB-14 Informatik, Universität des Saarlandes, 66041 Saarbrücken, Germany

Hugh C. Williams
Department of Computer Science, University of Manitoba, Winnipeg, Manitoba, Canada R3T 2N2

Communicated by Andrew M. Odlyzko

Received 28 January 1992 and revised 16 July 1993

Abstract. In 1976 Diffie and Hellman first introduced their well-known key-exchange protocol which is based on exponentiation in the multiplicative group $GF(p)^*$ of integers relatively prime to a large prime p (see [8]). Since then, this scheme has been extended to numerous other finite groups. Recently, Buchmann and Williams [2] introduced a version of the Diffie–Hellman protocol which uses the infrastructure of a real quadratic field. Theirs is the first such system not to require an underlying group structure, but rather a structure which is "almost" like that of a group. We give here a more detailed description of this scheme as well as state the required algorithms and considerations for their implementation.

Key words. Key exchange, Discrete logarithm, Real quadratic field.

1. Introduction

The idea underlying the Diffie–Hellman key-exchange protocol [8] and all its extensions (see [17], [16], [20], [14], [1], and others) is as follows. Two communication partners Alice and Bob agree on a large finite multiplicative group G and an element $g \in G$ (G and g can be made public). Alice secretly selects some positive integer $a < |G|$, computes $x = g^a$, and transmits x to Bob. Similarly, Bob secretly chooses a positive integer $b < |G|$ and sends $y = g^b$ to Alice. Alice computes $k = y^a = g^{ba}$ and Bob computes $k = x^b = g^{ab}$. Then k can be used as the common key. A cryptanalyst tapping the communication line knows G, g, x, and y, and attempts to find k. One way to achieve this is to find $a = \log_g x$ or $b = \log_g y$, i.e., to solve the *discrete logarithm problem* (DLP) in G. Hence G should be chosen such that DLP in G is hard.

In [2] Buchmann and Williams sketched the first Diffie–Hellman protocol which does not require a group structure. Their scheme is instead based on the infrastructure of a real quadratic field. This approach not only introduces a new cryptographic idea, but is also quite unexpected in that it extends the scheme beyond the use of multiplication in a *group*—a concept which seemed essential to any Diffie–Hellman-like protocol—and employs, more generally, arithmetic in a set which is not a group. The fastest-known algorithm for solving the DLP corresponding to this scheme is subexponential if we assume certain Extended Riemann Hypotheses.

Unfortunately, there is a price to pay for this new idea. The key-generation algorithm is more complicated and computationally more involved than that of the standard Diffie–Hellman protocol. Furthermore, the scheme requires more bandwidth and an additional round of communication, although in the second round, the two partners transmit at most one bit each. Thus the system, while employing an interesting mathematical concept, seems to loose some of its practicality.

In this paper we give a more detailed description of the protocol [2]. Section 2 presents the underlying mathematical concepts and how they are used in the scheme. In Sections 3 and 4 we give the required algorithms, analyse their complexity, and consider implementation issues and error bounds. Section 5 shows how the two communication partners can agree on a unique key. The overall protocol is presented in Section 6. The paper concludes with a discussion of the scheme's security in Section 7 and some numerical examples and aspects of optimization in Section 8.

2. Real Quadratic Fields

2.1. *Reduced Ideals*

For a more detailed introduction to this material we refer the reader to [7] or [9] and [22]. Let D be a positive squarefree integer. $\mathbf{K} = \mathbf{Q} + \mathbf{Q}\sqrt{D}$ is the *real quadratic number field* generated by $\sqrt{D}$ over the rationals $\mathbf{Q}$. Let

$$\sigma = \begin{cases} 1 & \text{if } D \equiv 2, 3 \pmod 4, \\ 2 & \text{if } D \equiv 1 \pmod 4, \end{cases}$$

and let $\omega = (\sigma - 1 + \sqrt{D})/\sigma \in \mathbf{K}$. It can be shown that $\mathbf{O} = \mathbf{Z} \oplus \mathbf{Z}\omega$ is the *maximal real quadratic order* in $\mathbf{K}$, where $\mathbf{Z}$ denotes the set of rational integers.

For any $\alpha = x + y\sqrt{D} \in \mathbf{K}$ $(x, y \in \mathbf{Q})$, denote by $\alpha' = x - y\sqrt{D}$ its *algebraic conjugate*. The *norm* of α is defined as $\mathrm{N}(\alpha) = \alpha\alpha' = x^2 - y^2 D$.

A *unit* in $\mathbf{K}$ is a divisor (in $\mathbf{O}$) of 1, or, equivalently, an element in $\mathbf{O}$ of norm 1. A unique unit $\eta > 1$ in $\mathbf{K}$ exists such that every unit in $\mathbf{K}$ can be written as $\pm\eta^n$ for some $n \in \mathbf{Z}$. η is called the *fundamental unit* of $\mathbf{K}$. Denote by $R = \log \eta$ the *regulator* of $\mathbf{K}$.

A subset $\mathbf{a}$ of $\mathbf{O}$ is called an (*integral*) *ideal* in $\mathbf{O}$ if both $\mathbf{a} + \mathbf{a}$ and $\mathbf{O} \cdot \mathbf{a}$ are subsets of $\mathbf{a}$. It can be shown that every ideal $\mathbf{a}$ has a representation

$$\mathbf{a} = [a, b + c\omega] = \mathbf{Z}a \oplus \mathbf{Z}(b + c\omega),$$

where $a, b, c \in \mathbf{Z}$, $a, b > 0$, $c|b$, and $c|a$. The integers c and a are unique and a is the least positive rational integer in $\mathbf{a}$, denoted by $L(\mathbf{a})$. We also have $ac|N(b + c\omega)$.

A *principal ideal* $\mathbf{a}$ of $\mathbf{O}$ is an ideal of the form $\mathbf{a} = \alpha\mathbf{O}$ where $\alpha \in \mathbf{O}$. We say that α *generates* the ideal $\mathbf{a}$ and write $\mathbf{a} = (\alpha)$. Denote by $\mathbf{P}$ the set of principal ideals in $\mathbf{O}$. Clearly, $\mathbf{P}$ is a semigroup under multiplication, if we define the product of two principal ideals (α), (β) to be $(\alpha\beta)$.

We define a pair of ideals $\mathbf{a}$, $\mathbf{b}$ in $\mathbf{O}$ to be *equivalent* (written $\mathbf{a} \sim \mathbf{b}$) if $\gamma \in \mathbf{K} - \{0\}$ exists such that $\mathbf{a} = \gamma\mathbf{b}$, or, equivalently, if $\alpha, \beta \in \mathbf{O} - \{0\}$ exist such that $\alpha\mathbf{a} = \beta\mathbf{b}$. It is easy to see that $\sim$ is in fact a proper equivalence relation. Furthermore, $\mathbf{P}$ is exactly the equivalence class of the *unit* ideal $\mathbf{O} = (1)$ under $\sim$.

If $\mathbf{a}$ is any ideal, then we call a number $\mu \in \mathbf{a}$ a *minimum* in $\mathbf{a}$ if $\mu > 0$ and no $\alpha \in \mathbf{a} - \{0\}$ exists such that $|\alpha| < \mu$ and $|\alpha'| < |\mu'|$. Clearly, 1 is a minimum in $\mathbf{O}$. It can be shown that the set $\{\log \mu | \mu \text{ is a minimum in } \mathbf{O}\}$ is discrete in the real numbers $\mathbf{R}$. Furthermore, there is an iterative procedure which enables us to generate a sequence of minima in $\mathbf{O}$ such that $1 = \mu_1 < \mu_2 < \mu_3 < \cdots$ (details of this method are given in Section 3). Then we have $\mu_{l+1} = \eta$ for some $l \in \mathbf{Z}_+$, and in fact $\mu_{j+ml} = \mu_j\eta^m$ for all $j \in \mathbf{Z}_+$, $m \in \mathbf{Z}$ such that $j + ml \geq 1$. If D is chosen appropriately (see Section 7.2), then l might be as large as $O(\sqrt{D} \log \log D)$.

An ideal $\mathbf{a} = [L(\mathbf{a}), b + c\omega]$ is said to be *primitive* if $c = 1$. We define $\mathbf{a}$ to be *reduced* if $\mathbf{a}$ is primitive and $L(\mathbf{a})$ is a minimum in $\mathbf{a}$. Clearly, $\mathbf{O} = [1, \omega]$ is reduced. Denote by $\mathfrak{R}$ the set of all reduced principal ideals in $\mathbf{O}$. It can be shown that $\mathbf{r} \in \mathfrak{R}$ if and only if $\mathbf{r}$ is generated by a minimum in $\mathbf{O}$, i.e., $\mathbf{r} = \mathbf{r}_j = (\mu_j)$ for some $j \in \mathbf{Z}_+$. Thus the ordered sequence $(\mu_j)_{j \in \mathbf{Z}_+}$ of minima in $\mathbf{O}$ gives rise to an ordered sequence $\mathbf{r}_1 = (1), \mathbf{r}_2, \mathbf{r}_3, \ldots$ of reduced principal ideals Since $\mu_{j+ml} = \mu_j\eta^m$, it follows that $\mathbf{r}_{j+ml} = \mathbf{r}_j$ for all $j \in \mathbf{Z}_+$, $m \in \mathbf{Z}$ such that $j + ml \geq 1$. Hence the sequence $(\mathbf{r}_j)_{j \in \mathbf{Z}_+}$ is purely periodic with period length l, and the set $\mathfrak{R}$ is finite and of cardinality l. If we set $\mathbf{M} = \{\mu_1 = 1, \mu_2, \ldots, \mu_l\}$, i.e., $\mathbf{M}$ consists of all the minima $\mu \in \mathbf{O}$ such that $1 \leq \mu < \eta$, then we can write $\mathfrak{R} = \{\mathbf{r}_i = (\mu_i) | \mu_i \in \mathbf{M}\} = \{\mathbf{r}_1 = (1), \mathbf{r}_2, \ldots, \mathbf{r}_l\}$.

2.2. *Distances*

With each reduced ideal $\mathbf{r}_j = (\mu_j)$ where μ_j is a minimum in $\mathbf{O}$, we can associate a *distance*

$$\delta_j = \log \mu_j.$$

Then δ_j is a strictly monotonically increasing function, i.e., $\delta_{j+1} > \delta_j$. Furthermore, we can define the distance[1] between a reduced ideal $\mathbf{r}_j$ and a real number x as

$$\delta(\mathbf{r}_j, x) = \delta_j - x.$$

For each $x \in \mathbf{R}_+$, there is a unique $j \in \mathbf{Z}_+$ such that $\delta_j \leq x < \delta_{j+1}$. If $\mathbf{r}_j = (\mu_j)$ where $\delta_j = \log \mu_j$, then we call $\mathbf{r}_j$ the ideal *closest to the left* of x and denote it by $\mathbf{r}_-(x)$. Similarly, if $\mathbf{r}_{j+1} = (\mu_{j+1})$ where $\delta_{j+1} = \log \mu_{j+1}$, then $\mathbf{r}_{j+1} = \mathbf{r}_+(x)$ is the ideal *closest to the right* of x.

We are now ready to present the idea for our protocol.

---

[1] This definition differs from the one given in [2] only in its sign.

2.3. *Outline of the Protocol*

Our goal is to establish a protocol similar to the one developed by Diffie and Hellman [8]. However, as the underlying set we use $\mathfrak{R}$, the set of reduced principal ideals in $\mathbf{K}$. The idea is as follows.

Two communication partners Alice and Bob publicly agree on a real quadratic field $\mathbf{K}$ with large D. Alice secretly chooses a positive integer a and computes a reduced ideal $\mathbf{a} \in \{\mathbf{r}_-(a), \mathbf{r}_+(a)\}$ and an approximation $\hat{\delta}(\mathbf{a}, a)$ to its distance $\delta(\mathbf{a}, a)$ from a. She sends both the ideal and its approximate distance from a to Bob. Similarly, Bob secretly chooses $b \in \mathbf{Z}_+$ and determines a reduced ideal $\mathbf{b} \in \{\mathbf{r}_-(b), \mathbf{r}_+(b)\}$ and an approximation $\hat{\delta}(\mathbf{b}, b)$ of $\delta(\mathbf{b}, b)$. He transmits both $\mathbf{b}$ and $\hat{\delta}(\mathbf{b}, b)$ to Alice. From $\mathbf{b}$, $\hat{\delta}(\mathbf{b}, b)$, and a, Alice computes a reduced ideal $\mathbf{k}_A \in \{\mathbf{r}_-(ab), \mathbf{r}_+(ab)\}$. Likewise, Bob determines from $\mathbf{a}$, $\hat{\delta}(\mathbf{a}, a)$, and b a reduced ideal $\mathbf{k}_B \in \{\mathbf{r}_-(ab), \mathbf{r}_+(ab)\}$. Since our distances are irrational numbers and the two communication partners might use different rational approximations in their respective computations, $\mathbf{k}_A$ and $\mathbf{k}_B$ need not be the same ideal. Furthermore, Alice and Bob do not know whether they computed the same ideal, i.e., whether $\mathbf{k}_A = \mathbf{k}_B$. However, the exchange of at most two more bits of information will enable them to agree on a common key ideal $\mathbf{k}$.

This scheme introduces two problems:

1. Given a number a, how to find an ideal $\mathbf{a} \in \{\mathbf{r}_-(a), \mathbf{r}_+(a)\}$. More generally, given a real number a, an ideal $\mathbf{b} \in \{\mathbf{r}_-(b), \mathbf{r}_+(b)\}$, and $\hat{\delta}(\mathbf{b}, b)$, how to find $\mathbf{k}_A \in \{\mathbf{r}_-(ab), \mathbf{r}_+(ab)\}$.
2. How do the communication partners detect whether or not $\mathbf{k}_A = \mathbf{k}_B$ and, in case $\mathbf{k}_A \neq \mathbf{k}_B$, how do they agree on a common key ideal $\mathbf{k}$?

In order to solve problem 1, we need to be able to do arithmetic in $\mathfrak{R}$. The required algorithms are introduced in the following two sections. The ambiguity problem of the key ideal is solved in Section 5.

3. Arithmetic in $\mathfrak{R}$

3.1. *Ideals and Continued Fractions*

We introduce a *basis representation* for ideals which allows us to perform integer arithmetic on primitive principal ideals. Let $\mathbf{a} = [a, b + \omega]$ be a primitive principal ideal. If we set $Q = a\sigma$, $P = b\sigma + \sigma - 1$, then $\mathbf{a}$ can be written as

$$\mathbf{a} = \left[\frac{Q}{\sigma}, \frac{P + \sqrt{D}}{\sigma}\right] = \mathbf{Z}\frac{Q}{\sigma} \oplus \mathbf{Z}\frac{P + \sqrt{D}}{\sigma},$$

where $P, Q \in \mathbf{Z}$, $\sigma | Q$, and $\sigma Q | D - P^2$ (recall that $\sigma = 1$ or 2). So every primitive ideal can be associated with a pair (P, Q) of rational integer *coefficients*. For the unit ideal $\mathbf{O}$, we have $P = \sigma - 1$, $Q = \sigma$.

Let

$$\mathbf{a} = \left[\frac{Q}{\sigma}, \frac{P + \sqrt{D}}{\sigma}\right] \in \mathbf{P}$$

be a primitive ideal. If we set $Q_0 = Q$, $P_0 = P$, $\varphi_0 = (P_0 + \sqrt{D})/Q_0$, and expand φ_0 into a continued fraction as described in Algorithm 1(a) below, then we obtain a sequence of primitive principal ideals $\mathbf{a}_2, \mathbf{a}_3, \ldots$, where

$$\mathbf{a}_j = \left[\frac{Q_{j-1}}{\sigma}, \frac{P_{j-1} + \sqrt{D}}{\sigma}\right] \qquad (j \in \mathbf{Z}_+).$$

For each $j \geq 2$, we call $\mathbf{a}_{j+1}$ the *right neighbour* and $\mathbf{a}_{j-1}$ the *left neighbour* of $\mathbf{a}_j$. If $\mathbf{a}_1 = \mathbf{a}$ is reduced, then $\mathbf{a}_1 = \mathbf{r}_k$ for some $k \in \mathbf{Z}_+$, $\mathbf{a}_j = \mathbf{r}_{k+j-1}$ is reduced for all $j \geq 1$, and the sequence $(\mathbf{a}_j)_{1 \leq j \leq l}$ will generate all the ideals in $\mathfrak{R}$. In this case we can compute reduced principal ideals by starting at any $\mathbf{r}_i$ $(i \geq k)$ and generating $\mathbf{r}_{i+1}$, $\mathbf{r}_{i+2}, \ldots$, or $\mathbf{r}_{i-1}, \mathbf{r}_{i-2}, \ldots, \mathbf{r}_1$ (for the latter sequence, we require $i \geq k+1$). In the case where $\mathbf{a}_1$ is not reduced, this method will yield a reduced ideal after $O(\log D)$ iterations. Hence, the continued fraction algorithm allows us to step through $\mathfrak{R}$ in either direction and quickly find, for any primitive principal ideal, an equivalent reduced one. The algorithm is given in [22] and operates as follows. Let $d = \lfloor\sqrt{D}\rfloor$.

Algorithm 1 (Continued Fraction Algorithm).

(a) *Input*: Any primitive ideal

$$\mathbf{a} = \left[\frac{Q}{\sigma}, \frac{P + \sqrt{D}}{\sigma}\right] \in \mathbf{P}.$$

Output: A sequence of primitive ideals $\mathbf{a}_1, \mathbf{a}_2, \ldots$ in $\mathbf{P}$, where

$$\mathbf{a}_j = \left[\frac{\mathbf{Q}_{j-1}}{\sigma}, \frac{P_{j-1} + \sqrt{D}}{\sigma}\right] \qquad (j \in \mathbf{Z}_+).$$

Algorithm: Set

$$P_0 = P, \qquad Q_0 = Q,$$

$$q_{j-1} = \left\lfloor \frac{P_{j-1} + \sqrt{D}}{Q_{j-1}} \right\rfloor, \quad P_j = q_{j-1}Q_{j-1} - P_{j-1}, \quad Q_j = \frac{D - P_j^2}{Q_{j-1}} \quad (j = 1, 2, \ldots).$$

Set

$$\mathbf{a}_j = \left[\frac{Q_{j-1}}{\sigma}, \frac{P_{j-1} + \sqrt{D}}{\sigma}\right].$$

Then $\mathbf{a}_{j+1} = \psi_j' \mathbf{a}_j$ where $\psi_j = (\sqrt{D} - P_j)/Q_{j-1}$ $(j \in \mathbf{Z}_+)$.

(b) *Input*: Any

$$\mathbf{r}_i \in \mathfrak{R}, \qquad \mathbf{r}_i = \left[\frac{Q_{i-1}}{\sigma}, \frac{P_{i-1} + \sqrt{D}}{\sigma}\right] \qquad (i \in \mathbf{Z}_+).$$

Output: The sequence of ideals $\mathbf{r}_{i-1}, \mathbf{r}_{i-2}, \ldots, \mathbf{r}_1 = \mathbf{O}$, where

$$\mathbf{r}_j = \left[\frac{Q_{j-1}}{\sigma}, \frac{P_{j-1} + \sqrt{D}}{\sigma}\right] \qquad (1 \leq j \leq i).$$

Algorithm:

$$Q_j = \frac{D - P_{j+1}^2}{Q_{j+1}}, \quad q_j = \left\lfloor \frac{P_{j+1} + d}{Q_j} \right\rfloor, \quad P_j = q_j Q_j - P_{j+1} \quad (j = i-2, i-1, \ldots, 1).$$

Set

$$\mathbf{r}_j = \left[\frac{Q_{j-1}}{\sigma}, \frac{P_{j-1} + \sqrt{D}}{\sigma} \right].$$

Then $\mathbf{r}_j = \varphi_j' \mathbf{r}_{j+1}$ where

$$\varphi_j = \frac{1}{\psi_j} = \frac{P_j + \sqrt{D}}{Q_j} \qquad (1 \le j \le i).$$

Theorem 3.1.

(a) *Let*

$$\mathbf{a} = \left[\frac{Q}{\sigma}, \frac{P + \sqrt{D}}{\sigma} \right]$$

be a primitive principal ideal and let $\mathbf{a}_1, \mathbf{a}_2, \ldots$ *be the sequence computed by Algorithm* 1(a). *If* $0 < Q_0 < \sqrt{D}$, *then, for all* $j \in \mathbf{Z}_+$:

(i) $\mathbf{a}_j \in \mathfrak{R}$; *if* $k \in \mathbf{Z}_+$ *is such that* $\mathbf{a}_1 = \mathbf{r}_k$, *then* $\mathbf{a}_j = \mathbf{r}_{k+j-1}$.

(ii) $q_j \ge 1, 0 < P_j < \sqrt{D}, \sigma \le Q_j < 2\sqrt{D}$.

(iii) $-1 < 1/\psi_j' = (P_j - \sqrt{D})/Q_j < 0$.

(iv) $q_j = \lfloor (P_{j+1} + d)/Q_j \rfloor$, *so the expressions for* q_j *in Algorithms* 1(a) *and* (b) *are equivalent.*

(b) *Let* $\mathbf{r}_i = [Q_{i-1}/\sigma, (P_{i-1} + \sqrt{D})/\sigma] \in \mathfrak{R}$ $(i \in \mathbf{Z}^{\ge 2})$ *and let* $\mathbf{r}_{i-1}, \mathbf{r}_{i-2}, \ldots$ *be the sequence computed by Algorithm* 1(b). *If* $0 < Q_{i-2} < \sqrt{D}$, *then, for all* $1 \le j \le i-1$:

(i) $\mathbf{r}_j \in \mathfrak{R}, \delta_j = \delta_{j+1} + \log|\varphi_j'| = \delta_{j+1} - \log|\psi_j'|$.

(ii) $q_{j-1} \ge 1, 0 < P_{j-1} < \sqrt{D}, \sigma \le Q_{j-1} < 2\sqrt{D}$.

(iii) $-1 < \varphi_{j-1}' = (P_{j-1} - \sqrt{D})/Q_{j-1} < 0$.

Proof. For part (a) see [22]. The second part follows analogously. □

The next lemma indicates how far we advance in distance in one step and in two consecutive steps of Algorithm 1(a).

Lemma 3.2. *Let* $\mathbf{a}_1$ *be as in Theorem* 3.1. *Then, for all* $j \in \mathbf{Z}_+$:

(a) $q_j < |\psi_{j+1}'| < q_j + 1$.

(b) $\psi_{j+1}' \psi_j' > 2$.

(c) $1 + 1/\sqrt{\Delta} < |\psi_j'| < \sqrt{\Delta}$, *where* $\Delta = (4/\sigma^2)D$ *is the* discriminant *of* $\mathbf{K}$.

By the Gauss–Kuz'min law of almost all continued fractions (see, for example, p. 92f. of [11]), a partial quotient q occurs with probability $\log_2(1 + 1/((q+1)^2 - 1))$. Hence, we expect q_j to be small in most cases, for example, $q_j = 1$ in 41.5%, $q_j \le 10$ in 87.4% of all cases.

Let $\mathbf{a}_1$ be a primitive principal ideal and let $\mathbf{a}_m$ ($m \in \mathbf{Z}_+$) be generated from $\mathbf{a}_1$ using Algorithm 1(a). Then $\mathbf{a}_m = \theta'_m \mathbf{a}_1$ where $\theta_1 = 1$, $\theta_m = \prod_{k=1}^{m-1} \psi_k$. It follows that, for any fixed $i \in \mathbf{Z}_+$ and $j \geq i$, we have $\mathbf{a}_j = (\theta'_j/\theta'_i)\mathbf{a}_i$. If we set $\zeta_m = 1/\theta_m$, i.e., $\zeta_1 = 1$, $\zeta_m = \prod_{k=1}^{m-1} \varphi_k$ ($m \geq 1$), then, for $1 \leq j \leq i$, we have $\mathbf{a}_j = (\zeta'_j/\zeta'_i)\mathbf{a}_i$.

In the special case where $\mathbf{a}_1 = \mathbf{r}_1 = (1)$, i.e., $\mathbf{a}_m = \mathbf{r}_m$, we have $(\theta'_m) = \mathbf{a}_m = \mathbf{r}_m = (\mu_m)$, and in fact $\mu_m = |\theta'_m|$. The following lemma summarizes some properties and gives a simple recurrence relation for θ_j, ζ_j.

Lemma 3.3. *For all* $m \geq 1$:

(a) $\theta_{m+2} = -q_m\theta_{m+1} + \theta_m$, $\zeta_{m+2} = q_m\zeta_{m+1} + \zeta_m$.
(b) $|\theta'_{m+1}| > |\theta'_m| \geq 1$, $|\zeta'_{m+1}| < |\zeta'_m| \leq 1$.
(c) $\operatorname{sgn}(\theta'_m) = (-1)^{m-1}$, $\operatorname{sgn}(\zeta'_m) = (-1)^{m-1}$.

Proof. We have $\theta_{m+2} = \psi_{m+1}\psi_m\theta_m$. An easy calculation shows $\psi_{m+1}\psi_m = -q_m\psi_m + 1$, whence follows the recurrence relation for θ_{m+2}. Similarly, we show $\varphi_{m+1}\varphi_m = q_m\varphi_m + 1$, which yields the recurrence relation for ζ_{m+2}. Now $|\theta'_{m+1}| = |\psi'_m||\theta'_m|$ and, by Theorem 3.1(a)(iii), we have $1 > 1/|\psi'_m|$, hence $|\theta'_{m+1}| > |\theta'_m|$. We show $\operatorname{sgn}(\theta'_m) = (-1)^{m-1}$ by induction on m. We have $\theta'_1 = 1 > 0$, $\theta'_2 = \psi'_1 = -(P_1 + \sqrt{D})/Q_0 < 0$ by Theorem 3.1, and for $m \geq 0$, using the induction hypothesis for $m+1$ and m, $\theta'_{m+2} = -q_m(-1)^m|\theta'_{m+1}| + (-1)^{m-1}|\theta'_m| = (-1)^m(q_m|\theta'_{m+1}| + |\theta'_m|)$. The rest of the lemma follows from the identity $\zeta_m = 1/\theta_m$. □

In order to find, for any $x \in \mathbf{R}$, a reduced ideal $\mathbf{r}_l \in \{\mathbf{r}_-(x), \mathbf{r}_+(x)\}$, we could use Algorithm 1(a), starting at $\mathbf{r}_1 = (1)$, to generate a sequence of reduced ideals $\mathbf{r}_2$, $\mathbf{r}_3$, ... with distances $\delta_2 = \log|\theta'_2|$, $\delta_3 = \log|\theta'_3|$, ..., until we obtain $\mathbf{r}_l$ such that $\delta_l \leq x < \delta_{l+1}$. However, since $|\psi'_l| < 2\sqrt{D}$ for $l \geq 1$ by Lemma 3.2(c), each step advances us $O(\log D)$ in distance, hence this will require exponential computation time if x is polynomial in D. We need to move through $\mathfrak{R}$ at a much more rapid pace. To achieve this, we make use of Shanks's *infrastructure* idea [19].

3.2. *Multiply & Reduce*

We impose an operation $*$ ("Multiply & Reduce") in $\mathfrak{R}$ as follows. If $\mathbf{r}_i$, $\mathbf{r}_j$ are reduced ideals with respective distances δ_i, δ_j, then $\mathbf{r}_i * \mathbf{r}_j$ is a reduced ideal $\mathbf{r}_m$ such that $\delta_m \approx \delta_i + \delta_j$, i.e., $m \approx i + j$. Now if we want to find a reduced ideal $\mathbf{r}_l$ such that $\delta_l \leq x < \delta_{l+1}$, where x is polynomial in D, we start with a reduced ideal $\mathbf{r}_i$ with small $\delta_i = O(\log D)$. $\mathbf{r}_i$ can be obtained using Algorithm 1(a) on $\mathbf{r}_1 = (1)$. We then compute $\mathbf{r}_j = \mathbf{r}_i * \mathbf{r}_i * \cdots * \mathbf{r}_i$ where the number of terms is $n \approx x/\delta_i$. Then $\delta_j \approx n\delta_i \approx x$, and it can be shown that a few applications of Algorithm 1, starting at $\mathbf{r}_j$, will yield $\mathbf{r}_l$. If we adopt a standard exponentiation technique as described, for example, on p. 442 of [12], we can compute $\mathbf{r}_j$ using $O(\log n) = O(\log D)$ applications of $*$, hence this method is much faster than the single-step method, provided the operation $*$ of two ideals can be done in time $O(\log D)$ and the computation of $\mathbf{r}_l$ from $\mathbf{r}_j$ requires at most $O(\log D)$ iterations of Algorithm 1.

In order to define $*$ formally, consider ideal multiplication as given in Section 2.1. Let $\mathbf{r}_i, \mathbf{r}_j \in \mathfrak{R}$. If we set $\mathbf{c} = \mathbf{r}_i\mathbf{r}_j$, then $\mathbf{c} = (\gamma)$ where $\log \gamma = \delta_i + \delta_j$, hence, $\mathbf{c}$ would give us exactly our required distance. Unfortunately, $\mathbf{c}$ need not be reduced. However, by using the reduction technique described in [22], we can compute a fixed reduced ideal $\mathbf{r}_m$ which we define to be $\mathbf{r}_i * \mathbf{r}_j$ such that $\delta_m = \delta_i + \delta_j + \varepsilon$ where $|\varepsilon| = O(\log D)$, so $|\varepsilon|$ is usually very small relative to δ_i, δ_j. $\mathbf{r}_m$ can be generated as follows. If we set $\mathbf{a}_1 = \mathbf{c}$ and apply the continued fraction algorithm as given in Algorithm 1(a) to the product ideal $\mathbf{c} = \mathbf{a}_1$ $O(\log D)$ times, then we obtain an ideal $\mathbf{a}_k$ which is reduced, i.e., $\mathbf{a}_k = \mathbf{r}_m$ for some $m \in \mathbf{Z}_+$. Since ideal multiplication requires time $O(\log D)$, $\mathbf{r}_i * \mathbf{r}_j$ can in fact be computed in time $O(\log D)$. The algorithms for ideal multiplication and reduction are given below.

Algorithm 2 (Ideal Multiplication).
Input:

$$\mathbf{r}_i = \left[\frac{Q_{i-1}}{\sigma}, \frac{P_{i-1} + \sqrt{D}}{\sigma}\right], \qquad \mathbf{r}_j = \left[\frac{Q_{j-1}}{\sigma}, \frac{P_{j-1} + \sqrt{D}}{\sigma}\right] \in \mathfrak{R} \qquad (i, j \in \mathbf{Z}_+).$$

Output: $\mathbf{c} \in \mathbf{P}$ primitive, $U \in \mathbf{Z}_+$ such that $\mathbf{r}_i\mathbf{r}_j = (U)\mathbf{c}$.

Algorithm:

1. Solve

$$\frac{Q_{i-1}}{\sigma}x_1 + \frac{Q_{j-1}}{\sigma}y_1 = Y = \gcd\left(\frac{Q_{i-1}}{\sigma}, \frac{Q_{j-1}}{\sigma}\right) \qquad \text{for } x_1, y_1, Y \in \mathbf{Z}.$$

2. Solve

$$\frac{P_{i-1} + P_{j-1}}{\sigma}x_2 + Yy_2 = U = \gcd\left(\frac{P_{i-1} + P_{i-1}}{\sigma}, Y\right) \qquad \text{for } x_2, y_2, U \in \mathbf{Z}.$$

3. Set $Q = Q_{i-1}Q_{j-1}/\sigma U^2$.
4. Set

$$X \equiv y_2x_1(P_{j-1} - P_{i-1}) + x_2\frac{D - P_{i-1}^2}{Q_{i-1}} \quad \left(\mathrm{mod}\,\frac{Q_{j-1}}{U}\right).$$

5. Set $P = P_{i-1} + (XQ_{i-1}/\sigma U) \pmod{Q}$. (If $U = Y$, then set $x_2 = 0$, $y_2 = 1$.)
6. Set

$$\mathbf{c} = \left[\frac{Q}{\sigma}, \frac{P + \sqrt{D}}{\sigma}\right].$$

This algorithm, which is basically Shanks' modification to Gauss' composition algorithm for quadratic forms, is mentioned in [21]. The factor U is extracted to ensure that the product ideal $\mathbf{c}$ is primitive.

Theorem 3.4. *If* $\mathbf{r}_i$, $\mathbf{r}_j$ *are such that* $0 < Q_{i-1}, Q_{j-1} < 2\sqrt{D}$, $-1 < (P_{i-1} - \sqrt{D})/Q_{i-1}, (P_{j-1} - \sqrt{D})/Q_{i-1} < 0$, *then Algorithm* 2 *performs* $O(\log D)$ *arithmetic operations on numbers requiring* $O(\log D)$ *bits of storage.*

Proof. $Q_{i-1}, Q_{j-1}, P_{i-1}, P_{j-1} = O(\sqrt{D})$, hence all numbers throughout the algorithm are bounded by $O(D)$. Our algorithm performs a fixed number of arithmetic operations plus two applications of the Extended Euclidean Algorithm (EEA) to solve the diophantine equations. The number of arithmetic operations performed by the EEA is logarithmic in its largest input number. □

Algorithm 3 (Reduction).
Input: $\mathbf{a}_1 \in \mathbf{P}$ where

$$\mathbf{a}_1 = \mathbf{c} = \frac{1}{U}\mathbf{r}_i\mathbf{r}_j = \left[\frac{Q}{\sigma}, \frac{P+\sqrt{D}}{\sigma}\right]$$

is computed by Algorithm 2.
Output:

$$\mathbf{a}_k = \theta_k'\mathbf{a}_1 \in \mathfrak{R},\ \theta_k = (-1)^{k-1}\frac{G_{k-2} - B_{k-2}\sqrt{D}}{Q}$$

such that $B_{k-2}, G_{k-2} \in \mathbf{Z}_{\geq 0}$ and $k \geq 2$.

Algorithm:

1. Set $Q_0' = Q$, $P_0' = P$, $B_{-2} = 1$, $B_{-1} = 0$.
2. Repeat, starting at $j = 1$:
   Compute q_{j-1}', Q_j', P_j' as in Algorithm 1(a).
   Set $B_{j-1} = q_{j-1}'B_{j-2} + B_{j-3}$.
   until $\sigma \leq Q_j' \leq d$.
3. Compute one more quadruple $(q_{j-1}', Q_j', P_j', B_{j-1})$ as in step 2.
4. Set

$$k = j+1, \qquad \mathbf{a}_k = \left[\frac{Q_{k-1}'}{\sigma}, \frac{P_{k-1}'+\sqrt{D}}{\sigma}\right], \qquad \theta_k = (-1)^{k-1}\frac{G_{k-2} - B_{k-2}\sqrt{D}}{Q_0'},$$

where $G_{k-2} = P_{k-1}'B_{k-2} + Q_{k-1}'B_{k-3}$.

Algorithm 3 is discussed in [22]. As soon as Q_j' is obtained such that $\sigma \leq Q_j' \leq d$, the ideal

$$\mathbf{a}_k = \left[\frac{Q_{k+1}'}{\sigma}, \frac{P_{k-1}'+\sqrt{D}}{\sigma}\right]$$

is reduced, so $\mathbf{a}_k = \mathbf{r}_m$ for some $m \in \mathbf{Z}_+$. The extra iteration in step 3 of the algorithm is to ensure that the bounds $0 < P_{k-1}' < \sqrt{D}$, $0 < Q_{k-1}' < 2\sqrt{D}$ of Theorem 3.1 are satisfied. (Note that we write P_{j-1}', Q_{j-1}' instead of P_{j-1}, Q_{j-1} to indicate that these are the coefficients of an ideal $\mathbf{a}_j$ which is not reduced for $j < k-1$. This notation is not to be confused with the notation $\alpha' \in \mathbf{K}$ which denotes the conjugate of α.)

Lemma 3.5. *Let* $\mathbf{c}$, B_i $(-2 \leq i \leq k-2)$, G_{k-2}, θ_k, *and* $\mathbf{a}_k$ *be as in Algorithm* 3. *Then* $0 \leq B_i < B_{i+1} < Q_0'/\sqrt{D}$ $(-2 \leq i \leq k-4)$, $B_{k-2} < (q_{k-2}'+1)(Q_0'/\sqrt{D})$, *and* $G_{k-2} < 3\sqrt{D}B_{k-2}$.

Proof. $0 \le B_i < B_{i+1}$ is clear from the recursion since all $q_i' \ge 1$ for $i \ge 1$ by Theorem 3.1. From Theorem 4.2 in [21], we get $B_{k-3} < Q_0'/\sqrt{D}$. The bound for B_{k-2} follows from the recursion. The inequality for G_{k-2} can be obtained using the bounds on P_{k-1}', Q_{k-1}' in Theorem 3.1 (it is at this point that we need the extra iteration in step 3 of Algorithm 3). □

Theorem 3.6. *If* $\mathbf{c} = (1/U)\mathbf{r}_i\mathbf{r}_j$, *where the coefficients of* $\mathbf{r}_i$ *and* $\mathbf{r}_j$ *satisfy the bounds of Theorem* 3.4, *then Algorithm* 3 *performs* $O(\log D)$ *arithmetic operations on numbers of* $O(\log D)$ *bits.*

Proof. From Algorithm 2, we have $Q_0' = O(D)$, $P_0' = O(D)$ where

$$\mathbf{c} = a_1 = \left[\frac{Q_0'}{\sigma}, \frac{P_0' + \sqrt{D}}{\sigma}\right].$$

If

$$\mathbf{a}_k = \left[\frac{Q_{k-1}'}{\sigma}, \frac{P_{k-1}' + \sqrt{D}}{\sigma}\right],$$

then it follows from Theorem 4.1 and Corollary 4.1.1 in [21] that $|P_i'| < \sqrt{D} + Q_0'$, $|Q_i'| \le Q_0'$ for $0 \le i \le k-2$. Theorem 3.1 yields $P_{k-1}', Q_{k-1}' = O(\sqrt{D})$. From Lemma 3.5, we obtain $B_i = O(\sqrt{D})\,(-2 \le i \le k-3)$ and $B_{k-2} < (q_{k-2}' + 1)B_{k-3} = O(D^{3/2})$, since $q_{k-2}' \le (P_{k-2}' + \sqrt{D})/Q_{k-2}' = O(D)$, and $G = O(D^2)$. (Indeed, by the Gauss–Kuz'min law, q_{k-2}' will be small most of the time, so generally we have $|P_{k-2}'| = O(\sqrt{D})$, $B_{k-2} = O(\sqrt{D})$, $G_{k-2} = O(D)$). Hence all numbers in the algorithm are bounded by a fixed power of D. By Corollary 4.2.1 of [22] the maximum number of iterations is $O(\log|Q_0'|/\sqrt{D}) = O(\log D)$. □

Theorem 3.7. *Let* $\mathbf{c} = (1/U)\mathbf{r}_i\mathbf{r}_j$, *where* $\mathbf{r}_i$, $\mathbf{r}_j$ *are as in Theorem* 3.4, *and let* $\mathbf{a}_k$ *be the reduced ideal computed from* $\mathbf{c} = \mathbf{a}_1$ *using Algorithm* 3. *If* $\mathbf{a}_k = \mathbf{r}_m$, *then* $\delta_m = \delta_i + \delta_j + \varepsilon$ *where* $|\varepsilon| = \log D + O(1)$.

Proof. We have, from Algorithm 2, $Q_0' = Q_{i-1}Q_{j-1}/\sigma U^2$, so

$$\frac{|\theta_k'|}{U} = \frac{G_{k-2} + B_{k-2}\sqrt{D}}{Q_{i-1}Q_{j-1}}\sigma U.$$

Since step 3 of Algorithm 3 ensures that $k \ge 2$, we have $B_{k-2} \ge 1$, so

$$\frac{|\theta_k'|}{U} > \frac{G_{k-2} + B_{k-2}\sqrt{D}}{4D} \ge \frac{1}{4\sqrt{D}}$$

and, from Lemma 3.5,

$$\frac{|\theta_k'|}{U} \le |\theta_k'| < \frac{3B_{k-2}\sqrt{D} + B_{k-2}\sqrt{D}}{Q_0'} = \frac{4B_{k-2}\sqrt{D}}{Q_0'} < 4(q_{k-2}' + 1) = O(D).$$

Set $\varepsilon = \log(|\theta_k'|/U)$, then $|\varepsilon| = \log D + O(1)$. If $\mathbf{r}_i = (\mu_i)$, $\mathbf{r}_j = (\mu_j)$, and if $\mathbf{a}_k = \mathbf{r}_m = (\mu_m)$, then $\mathbf{r}_m = \theta_k'\mathbf{c} = (\theta_k'/U)\mathbf{r}_i\mathbf{r}_j$, so $\mu_m = (|\theta_k'|/U)\mu_i\mu_j$, hence $\delta_m = \log \mu_m = \log(|\theta_k'|\mu_i\mu_j/U) = \varepsilon + \delta_i + \delta_j$. □

Let $x, y \in \mathbf{R}$. Our next goal is to compute efficiently from ideals $\mathfrak{r}_i \in \{\mathfrak{r}_-(x), \mathfrak{r}_+(x)\}$ and $\mathfrak{r}_j \in \{\mathfrak{r}_-(y), \mathfrak{r}_+(y)\}$ an ideal $\mathfrak{r}_n \in \{\mathfrak{r}_-(x+y), \mathfrak{r}_+(x+y)\}$. To achieve this, we first compute $\mathfrak{c} = (1/U)\mathfrak{r}_i\mathfrak{r}_j$ and a reduced ideal $\mathfrak{r}_m$ such that $\delta_m = \delta_i + \delta_j + \varepsilon$, $|\varepsilon| = \log D + O(1)$ in time $O(\log D)$, using Algorithms 2 and 3. Then $\delta(\mathfrak{r}_m, x+y) = \delta_m - x - y = \delta(\mathfrak{r}_i, x) + \delta(\mathfrak{r}_j, y) + \varepsilon$, and $\mathfrak{r}_m$ need not yet be our correct ideal $\mathfrak{r}_n$. However, $\mathfrak{r}_n$ can be computed from $\mathfrak{r}_m$ using Algorithm 1. It remains to be shown that this requires $O(\log D)$ iterations of Algorithm 1. We prove this result and discuss the details of the algorithm in the next section.

4. Computing Closest Ideals

4.1. *Preliminaries for the Implementation*

Since we wish to avoid evaluating logarithms in our implementation, we use *expoential* distances. If $\mathfrak{r}_j = (\mu_j)$ is any reduced ideal with distance δ_j, then define its exponential distance as simply $e^{\delta_j} = \mu_j$. Similarly, if $x \in \mathbf{R}_+$, we define

$$\lambda(\mathfrak{r}_j, x) = e^{\delta(\mathfrak{r}_j, x)} = \mu_j e^{-x}.$$

Since distances are generally irrational numbers, we need to use rational approximations in our algorithms. More specifically, we approximate a distance $\lambda(\mathfrak{r}, x) \in \mathbf{R}$ by $\hat{\lambda}(\mathfrak{r}, x) \in \mathbf{Q}$ with a fixed *precision* of p bits, i.e., we write

$$\hat{\lambda}(\mathfrak{r}, x) = \frac{M(\mathfrak{r}, x)}{2^p}, \qquad \text{where} \quad M(\mathfrak{r}, x) \in \mathbf{Z}_+.$$

We define the *relative error*:

$$\rho(\mathfrak{r}, x) = \frac{\hat{\lambda}(\mathfrak{r}, x)}{\lambda(\mathfrak{r}, x)}.$$

We denote by $\hat{\mathfrak{r}}(x)$ the ideal actually computed by our algorithm, so we always have $\hat{\mathfrak{r}}(x) \in \{\mathfrak{r}_-(x), \mathfrak{r}_+(x)\}$. Let $\lambda(x) = \lambda(\hat{\mathfrak{r}}(x), x)$, $\hat{\lambda}(x) = \hat{\lambda}(\hat{\mathfrak{r}}(x), x) = M(x)/2^p$, and $\rho(x) = \hat{\lambda}(x)/\lambda(x)$.

The following lemma is an immediate consequence of Lemma 3.2(a) and (c).

Lemma 4.1. *Let $x \in \mathbf{R}$ and let $j \in \mathbf{Z}_+$ be such that $\mathfrak{r}_-(x) = \mathfrak{r}_j$, $\mathfrak{r}_+(x) = \mathfrak{r}_{j+1}$.*

(a) $1/(q_{j-1}+1) < \lambda(\mathfrak{r}_-(x), x) \le 1 < \lambda(\mathfrak{r}_+(x), x) < q_{j-1} + 1$.

(b) $\lambda(\mathfrak{r}_-(x), x) > 1/\sqrt{\Delta}$, $\lambda(\mathfrak{r}_+(x), x) < \sqrt{\Delta}$.

For our implementation details, we need to define a number of constants together with their properties. Let $B \in \mathbf{Z}_{\ge 2}$ be an upper bound on the secretly chosen "exponents" a, b such that B is polynomial bounded in D. Set

$$d^* = \lceil 2^p \sqrt{D} \rceil, \qquad \chi = 1 + \frac{1}{2^{p-1}}, \qquad g = 1 + \frac{1}{47d},$$

$$\gamma = \lceil g^{-1} 2^p \rceil, \qquad K = \left(\frac{\chi^2}{1 - \gamma^{-1}}\right)^2, \qquad A = g^{1/16B^2}.$$

Also recall that $d = \lfloor D \rfloor$ and $\Delta = (4/\sigma^2)D$ is the discriminant of $\mathbf{K}$. Four our computation, we require

$$2^p \geq 3072dB^2,$$

i.e., our precision is polynomial in D. For example, if a, b are bounded by $d = \lfloor \sqrt{D} \rfloor$, we must carry $O(D^{3/2})$ bits of precision; a bound of $\lfloor \sqrt[4]{D} \rfloor$ on a and b requires a precision of $O(D)$ bits, etc. Furthermore, γ will be a lower bound for all our approximate distances $M(\mathbf{a}, x)$ throughout our protocol. The following inequalities hold.

Lemma 4.2.

(a) $\gamma > 1$.
(b) $K > 1$.
(c) $\chi^2(1 + g/2^p) < \sqrt{K} < A^3$.
(d) $g^7 < 1 + 1/\sqrt{\Delta}$.
(e) $\chi + 2^{-p} < 1 + 1/2^{p-2} < A < g$.
(f) $(1 + 2^{-p})g^6/(1 - g^3 2^{-p}) < 1 + 1/\sqrt{\Delta}$.

Proof. We only prove the inequality $1 + 1/2^{p-2} < A$ as it explains our lower bound for p. We have

$$\left(1 + \frac{1}{2^{p-2}}\right)^{16B^2} = \left(1 + \frac{1}{2^{p-2}}\right)^{2^{p-2}(16B^2/2^{p-2})} < \exp\left(\frac{16B^2}{2^{p-2}}\right).$$

Since $\log(1 + 1/x) > 1/(1 + x)$ for $x > 1$;

$$\log(g) \geq \frac{1}{47d + 1} \geq \frac{1}{48d} \geq \frac{16B^2}{2^{p-2}}.$$

So

$$A^{16B^2} = g \geq \exp\left(\frac{16B^2}{2^{p-2}}\right) > \left(1 + \frac{1}{2^{p-2}}\right)^{16B^2}. \qquad \square$$

4.2. *The Algorithms*

Our first algorithm in this section takes two input ideals $\mathfrak{r}(x) \in \{\mathbf{r}_-(x), \mathbf{r}_+(x)\}$ and $\mathfrak{r}(y) \in \{\mathbf{r}_-(y), \mathbf{r}_+(y)\}$ $(x, y \in \mathbf{R})$ and computes $\mathfrak{r}(x + y) \in \{\mathbf{r}_-(x + y), \mathbf{r}_+(x + y)\}$. The second algorithm computes, from a positive integer m and $\mathfrak{r}(x) \in \{\mathbf{r}_-(x), \mathbf{r}_+(x)\}$, an ideal $\mathfrak{r}(mx) \in \{\mathbf{r}_-(mx), \mathbf{r}_+(mx)\}$.

Algorithm 4.
Input: $\mathfrak{r}(x), \mathfrak{r}(y) \in \mathfrak{R}$, $M(x)$, $M(y)$ $(x, y \in \mathbf{R})$ such that:

(i) $M(x), M(y) \geq \gamma$.
(ii) $\mathfrak{r}(x) \in \{\mathbf{r}_-(x), \mathbf{r}_+(x)\}$, $\mathfrak{r}(y) \in \{\mathbf{r}_-(y), \mathbf{r}_+(y)\}$.
(iii) $g^{-1} \leq \rho(x)\rho(y) \leq g$.

Output: $\mathfrak{r}(x + y) \in \mathfrak{R}$ such that $\mathfrak{r}(x + y) \in \{\mathbf{r}_-(x + y), \mathbf{r}_+(x + y)\}$, $M(x + y) \geq \gamma$.

Algorithm: First use Algorithm 2 to compute $U \in \mathbf{Z}_+$,

$$\mathbf{c} = \left[\frac{Q_0'}{\sigma}, \frac{P_0' + \sqrt{D}}{\sigma}\right] \in \mathbf{P}$$

such that $\mathbf{c} = (1/U)\hat{\mathbf{f}}(x)\hat{\mathbf{f}}(y) = \mathbf{a}_1$. Then compute

$$\mathbf{a}_k = \left[\frac{Q_{k-1}'}{\sigma}, \frac{P_{k-1}' + \sqrt{D}}{\sigma}\right] \in \mathfrak{R}, \qquad \theta_k = (-1)^{k-1}\frac{G_{k-2} - B_{k-2}\sqrt{D}}{Q_0'}$$

such that $\mathbf{a}_k = \theta_k'\mathbf{c}$ and $G_{k-2}, B_{k-2} \geq 0$, using Algorithm 3. Then $\mathbf{a}_k = (\theta_k'/U)\hat{\mathbf{f}}(x)\hat{\mathbf{f}}(y)$ and

$$\mathbf{a}_k = \mathbf{r}_m = \left[\frac{Q_{m-1}}{\sigma}, \frac{P_{m-1} + \sqrt{D}}{\sigma}\right] \qquad \text{for some} \quad m \in \mathbf{Z}_+.$$

Set

$$T = \left\lceil \frac{2^{2p}G_{k-2}2^p + B_{k-2}d^*}{Q} \right\rceil, \qquad \hat{\theta} = \frac{T}{2^{3p}}, \qquad L = g\frac{\hat{\theta}}{U}\hat{\lambda}(x)\hat{\lambda}(y).$$

Case 1: $1 \leq L \leq g^3$. Then set

$$\hat{\mathbf{f}}(x + y) = \mathbf{r}_m, \qquad \mathrm{M}(x + y) = \left\lceil \frac{\hat{\theta}}{U}2^p\hat{\lambda}(x)\hat{\lambda}(y) \right\rceil = \left\lceil \frac{L2^p}{g} \right\rceil.$$

Case 2: $L < 1$. Compute q_{m-1}, P_m, Q_m using Algorithm 1(a). Set

$$T_m = 2^p, \qquad T_{m+1} = \left\lceil \frac{P_m 2^p + d^*}{Q_{m-1}} \right\rceil.$$

From $q_{m-1}, P_m, Q_m, T_m, T_{m+1}$, compute $q_{j-2}, P_{j-1}Q_{j-1}$, using Algorithm 1(a) and T_j $(j \geq m + 2)$ where

$$T_j = q_{j-2}T_{j-1} + T_{j-2}$$

until $j = n$ such that $T_n \geq 2^p/L > T_{n-1}$. Set

$$\hat{\mathbf{f}}(x + y) = \mathbf{r}_n, \qquad \mathrm{M}(x + y) = \left\lceil \frac{T_n\hat{\theta}}{U}\hat{\lambda}(x)\hat{\lambda}(y) \right\rceil = \left\lceil \frac{T_n L}{g} \right\rceil.$$

Case 3: $L > g^3$. Set

$$T_m = 2^p - 1, \qquad T_{m-1} = \left\lceil \frac{(P_{m-1}2^p + d^*)2^{p-1}}{Q_{m-2}(2^{p-1} + 1)} \right\rceil = \left\lceil \frac{P_{m-1}2^p + d^*}{\chi Q_{m-2}} \right\rceil.$$

From $Q_{m-1}P_{m-1}, T_m, T_{m-1}$, compute Q_j, q_j, P_j, using Algorithm 1(b) and T_j $(j \leq m - 2)$ where

$$T_j = q_j T_{j+1} + T_{j+2}$$

until $j = n$ such that $T_{n-1} > L2^p \geq T_n$. Set

$$\hat{\mathbf{f}}(x + y) = \mathbf{r}_n, \qquad \mathrm{M}(x + y) = \left\lceil \frac{\hat{\theta}2^{2p}}{UT_n}\hat{\lambda}(x)\hat{\lambda}(y) \right\rceil = \left\lceil \frac{L2^{2p}}{gT_n} \right\rceil.$$

Proof of correctness of Algorithm 4. As mentioned before, we need to step through $\mathfrak{R}$ in the appropriate direction, starting at $\mathbf{r}_m$, to obtain $\mathbf{r}_n \in \{\mathbf{r}_-(x+y), \mathbf{r}_+(x+y)\}$. To see that $\mathbf{r}_n$ is indeed the correct ideal, it suffices to show that $\lambda(\mathbf{r}_{n-1}, x+y) < 1 < \lambda(\mathbf{r}_{n+1}, x+y)$.

Since $\lambda(\mathbf{a}_k, x+y) = (|\theta_k'|/U)\lambda(x)\lambda(y)$, we see that $\hat{\theta}$ and L are approximations for $|\theta_k'|$ and $\lambda(\mathbf{a}_k, x+y)$, respectively. For simplicity, let $\theta = |\theta_k'|$, $G = G_{k-2}$, $B = B_{k-2}$, and $Q = Q_0'$. Then

$$\theta \le \hat{\theta} < \chi\theta.$$

Proof.

$$2^{3p}\theta = \frac{2^{3p}G + 2^{3p}B\sqrt{D}}{Q} \le 2^{2p}\frac{G^{2p} + Bd^*}{Q} \le T < 2^p\frac{G^{2p} + Bd^*}{Q} + 1$$
$$< 2^p\frac{G2^p + B(2^p\sqrt{D} + 1)}{Q} + 1 = 2^{3p}\theta + 2^{2p}\frac{B}{Q} + 1.$$

Now $2^{3p}\chi = 2^{3p} + 2^{2p+1}$ and $2^{2p} \ge 3072^2 d^2 B^4 \ge (3072/2)^2(2d)^2 \ge (3072/2)^2 D$, since $D = (\sqrt{D})^2 < (d+1)^2 \le (2d)^2$. Furthermore, step 3 of Algorithm 3 guarantees that $k \ge 2$, so $B \ge B_0 \ge 1$ and $G \ge P_{k-1}B \ge 1$. Therefore

$$2^{2p+1}\frac{B}{Q} \ge \frac{2^{2p+1}}{Q} > \frac{2^{2p+1}}{4D} \ge \frac{1}{2}\left(\frac{3072}{2}\right)^2 > 1$$

and trivially

$$2^{2p+1}\frac{B\sqrt{D}}{Q} > 2^{2p}\frac{B}{Q}.$$

It follows that

$$T < 2^{3p}\theta + 2^{2p}\frac{B}{Q} + 1 < 2^{3p}\theta + 2^{2p+1}\frac{B\sqrt{D}}{Q} + 2^{2p+1}\frac{G}{Q} = 2^{3p}\theta + 2^{2p+1}\theta = 3^p\chi\theta.$$

We also have

$$\frac{L}{g^3} < \lambda(\mathbf{a}_k, x+y) \le L.$$

Proof.

$$\lambda(\mathbf{a}_k, x+y) = \frac{\theta}{U}\lambda(x)\lambda(y) = \frac{\theta\hat{\lambda}(x)\hat{\lambda}(y)}{U\rho(x)\rho(y)} = \frac{\theta L}{\hat{\theta}g\rho(x)\rho(y)},$$

so, since $\chi < g$, by Lemma 4.2(e),

$$\frac{L}{g^3} < \frac{L}{g^2\chi} < \frac{L}{g\chi\rho(x)\rho(y)} < \lambda(\mathbf{a}_k, x+y) \le \frac{L}{g\rho(x)\rho(y)} \le L.$$

The bounds on $\lambda(\mathbf{a}_k, x+y)$ show that our three cases, $1 \le L \le g^3$, $L < 1$, $L > g^3$, correspond to $\lambda(\mathbf{r}_m, x+y) \approx 1$, $\lambda(\mathbf{r}_m, x+y) < 1$, and $\lambda(\mathbf{r}_m, x+y) > 1$, respectively. For each case we need to show $\mathrm{M}(\mathbf{r}_n, x+y) \ge \gamma$ and $\lambda(\mathbf{r}_{n-1}, x+y) < 1 < \lambda(\mathbf{r}_{n+1}, x+y)$.

Case 1. In this case we have $1/g^3 < \lambda(\mathbf{r}_m, x + y) \leq g^3$, so, by Lemmas 3.2(c) and 4.2(d),

$$\lambda(\mathbf{r}_{m-1}, x + y) = \frac{\lambda(\mathbf{r}_m, x + y)}{|\psi'_{m-1}|} < \frac{g^3}{1 + 1/\sqrt{\Delta}} < \frac{1}{g^4} < 1$$

and

$$\lambda(\mathbf{r}_{m+1}, x + y) = |\psi'_m| \lambda(\mathbf{r}_m, x + y) > \frac{1 + 1/\sqrt{\Delta}}{g^3} > g^4 > 1.$$

Furthermore, $\mathrm{M}(x + y) \geq \lceil 2^p/g \rceil = \gamma$.

Case 2. Here $\lambda(\mathbf{r}_m, x + y) < 1$, so we need to compute right neighbours of $\mathbf{r}_m$ in order to increase the distance and move closer to $x + y$. Note that if $\mathbf{r}_j = (\theta'_j/\theta'_m)\mathbf{r}_m$, then $T_j/2^p$ is an approximation for $|\theta'_j/\theta'_m| = \prod_{i=m}^{j-1} |\psi'_i|$ $(j \geq m)$ and this expression increases as j increases. Hence, if $T_j \approx 2^p/L$, then

$$1 \approx \frac{T_j}{2^p} L \approx \left|\frac{\theta'_j}{\theta'_m}\right| \lambda(\mathbf{r}_m, x + y) = \lambda(\mathbf{r}_j, x + y).$$

We have

$$\left|\frac{\theta'_j}{\theta'_m}\right| \leq \frac{T_j}{2^p} < \chi \left|\frac{\theta'_j}{\theta'_m}\right| \qquad (j \geq m).$$

Proof by induction on j. The case $j = m$ is $2^p = T_m < \chi 2^p$. For $j = m + 1$, we have

$$\begin{aligned} 2^p \left|\frac{\theta'_{m+1}}{\theta'_m}\right| &\leq 2^p |\psi'_m| < \frac{P_m 2^p + d^*}{Q_{m-1}} \leq T_{m+1} < \frac{P_m 2^p + d^*}{Q_{m-1}} + 1 \\ &< \frac{P_m 2^p + \sqrt{D} 2^p + 1}{Q_{m-1}} + 1 = 2^p |\psi'_m| + \frac{1}{Q_{m-1}} + 1 < 2^p |\psi'_m| + 2 \\ &= \chi 2^p |\psi'_m| = \chi 2^p \left|\frac{\theta'_{m+1}}{\theta'_m}\right|. \end{aligned}$$

From Lemma 3.3(a) and (c), it follows that

$$\left|\frac{\theta'_{j+2}}{\theta'_m}\right| = q_j \left|\frac{\theta'_{j+1}}{\theta'_m}\right| + \left|\frac{\theta'_j}{\theta'_m}\right|,$$

hence, since all $q_j \geq 1$ for $j \geq m$, the rest follows trivially from the linear recursion for T_j.

We clearly have $\mathrm{M}(x + y) \geq \lceil 2^p/g \rceil = \gamma$. Now

$$\lambda(\mathbf{r}_j, x + y) = \left|\frac{\theta'_j}{\theta'_m}\right| \lambda(\mathbf{r}_m, x + y) = \frac{|\theta'_j| 2^p}{|\theta'_m| T_j} \frac{\theta}{\hat{\theta}} \frac{T_j L}{2^p} \frac{1}{g\rho(x)\rho(y)} \qquad \text{for all} \quad j \geq m,$$

so

$$\lambda(\mathbf{r}_{n-1}, x + y) = \frac{|\theta'_{n-1}| 2^p}{|\theta'_m| T_{n-1}} \frac{\theta}{\hat{\theta}} \frac{T_{n-1}}{L 2^p} \frac{1}{g\rho(x)\rho(y)} < 1 \cdot 1 \cdot 1 \cdot \frac{1}{g} \cdot g = 1$$

and, as in Case 1,

$$\lambda(\mathbf{r}_{n+1}, x+y) > \left(1+\frac{1}{\sqrt{\Delta}}\right)\frac{|\theta_n'|2^p}{|\theta_m'|T_n}\frac{\theta}{\hat{\theta}}\frac{T_nL}{2^p}\frac{1}{g\rho(x)\rho(y)} > g^7\frac{1}{\chi}\frac{1}{\chi}\cdot 1\cdot\frac{1}{g^2} = \frac{g^5}{\chi^2} > g^3 > 1,$$

hence $\hat{\mathbf{r}}(x+y) \in \{\mathbf{r}_-(x+y), \mathbf{r}_+(x+y)\}$.

Case 3. In this final case $\lambda(\mathbf{r}_m, x+y) > 1$, so we need to compute left neighbours of $\mathbf{r}_m$ in order to decrease the distance and move closer to $x+y$. Here, if $\mathbf{r}_j = (\zeta_m'/\zeta_j')\mathbf{r}_m$, then $T_j/2^p$ is an approximation for

$$\left|\frac{\zeta_j'}{\zeta_m'}\right| = \left|\frac{\theta_m'}{\theta_j'}\right| = \prod_{i=j}^{m-1}|\psi_i'| \qquad (j \le m),$$

and this expression increases as j decreases. Hence, if $T_n \approx L2^p$, then

$$1 \approx \frac{2^pL}{T_n} \approx \left|\frac{\zeta_m'}{\zeta_n'}\right|L \approx \left|\frac{\zeta_m'}{\zeta_n'}\right|\lambda(\mathbf{r}_m, x+y) = \lambda(\mathbf{r}_n, x+y).$$

We show, by induction on j,

$$\frac{1}{\chi}\left|\frac{\zeta_j'}{\zeta_m'}\right| < \frac{T_j}{2^p} < \left|\frac{\zeta_j'}{\zeta_m'}\right| \qquad (j \le m).$$

Proof. $\chi(2^p-1) = (1+1/2^{p-1})(2^p-1) = 2^p+1-1/2^{p-1} > 2^p$, so $2^p/\chi < 2^p-1 = T_m < 2^p$.

$$\frac{2^p}{\chi}\left|\frac{\zeta_{m-1}'}{\zeta_m'}\right| = \frac{2^p}{\chi}|\psi_{m-1}'| \le \frac{d^*-P_{m-1}2^p}{\chi Q_{m-2}} \le T_{m-1} < \frac{P_{m-1}2^p+d^*}{\chi Q_{m-2}}+1$$
$$< \frac{P_{m-1}2^p+\sqrt{D}2^p+1}{\chi Q_{m-2}}+1 = \frac{2^p}{\chi}|\psi_{m-1}'|+\frac{1}{\chi Q_{m-2}}+1.$$

Now $\chi 2^p = 2^p+2$, so $2^p/\chi = 2^p - 2/\chi$ and

$$\begin{aligned}
\frac{2^p}{\chi}|\psi_{m-1}'|+\frac{1}{\chi Q_{m-2}}+1 &< 2^p|\psi_{m-1}'|+1-\frac{1}{\chi}\left(2|\psi_{m-1}'|-\frac{1}{Q_{m-2}}\right)\\
&= 2^p|\psi_{m-1}'|+1-\frac{1}{\chi}\frac{2P_{m-1}+2\sqrt{D}-1}{Q_{m-2}}\\
&< 2^p|\psi_{m-1}'|+1-\frac{1}{\chi}\frac{2+Q_{m-2}-1}{Q_{m-2}}\\
&= 2^p|\psi_{m-1}'|+1-\frac{1}{\chi}\left(1+\frac{1}{Q_{m-2}}\right)\\
&< 2^p|\psi_{m-1}'|+1-\frac{1}{\chi}\left(1+\frac{1}{2\sqrt{D}}\right)\\
&< 2^p|\psi_{m-1}'|+1-\frac{1}{\chi}\left(1+\frac{1}{2^{p-1}}\right) = 2^p|\psi_{m-1}'|\\
&= 2^p\left|\frac{\zeta_{m-1}'}{\zeta_m'}\right|,
\end{aligned}$$

using $\sqrt{D} < 2^{p-2}$. Now from Lemma 3.3(a) and (c), we obtain $|\zeta'_{j+2}| = -q_j|\zeta'_{j+1}| + |\zeta'_j|$, hence

$$\left|\frac{\zeta'_j}{\zeta'_m}\right| = q_j\left|\frac{\zeta'_{j+1}}{\zeta'_m}\right| + \left|\frac{\zeta'_{j+2}}{\zeta'_m}\right| \qquad (j \le m-2),$$

so T_j and $|\zeta'_j/\zeta'_m|$ satisfy the same recursion. Hence the above inequalities follow since all $q_j \ge 1$ for $j \le m-2$ by Theorem 3.1.

Now, as in Case 1, we have $M(x+y) \ge \lceil 2^p/g \rceil = \gamma$ and

$$\lambda(\mathbf{r}_j, x+y) = \left|\frac{\zeta'_m}{\zeta'_j}\right|\lambda(\mathbf{r}_m, x+y) = \frac{|\zeta'_m|\,T_j}{|\zeta'_j|\,2^p}\frac{\theta}{\hat{\theta}}\frac{L2^p}{T_j}\frac{1}{g\rho(x)\rho(y)} \qquad \text{for all} \quad j \le m,$$

so

$$\lambda(\mathbf{r}_{n-1}, x+y) = \frac{|\zeta'_m|\,T_{n-1}}{|\zeta'_{n-1}|\,2^p}\frac{\theta}{\hat{\theta}}\frac{L2^p}{T_{n-1}}\frac{1}{g\rho(x)\rho(y)} < 1\cdot 1\cdot 1\cdot\frac{1}{g}\cdot g = 1$$

and

$$\lambda(\mathbf{r}_{n+1}, x+y) > \left(1+\frac{1}{\sqrt{\Delta}}\right)\frac{|\zeta'_m|\,T_n}{|\zeta'_n|\,2^p}\frac{\theta}{\hat{\theta}}\frac{L2^p}{T_n}\frac{1}{g\rho(x)\rho(y)} > g^7\frac{1}{\chi}\frac{1}{\chi}\cdot 1\cdot\frac{1}{g^2} = \frac{g^5}{\chi^2} > g^3 > 1,$$

hence $\hat{\mathbf{r}}(x+y) \in \{\mathbf{r}_-(x+y), \mathbf{r}_+(x+y)\}$. □

Theorem 4.3. *If* $\hat{\mathbf{r}}(x), \hat{\mathbf{r}}(y) \in \mathfrak{R}$ *are such that the bounds of Theorem* 3.4 *and the conditions of Algorithm* 4 *hold, then Algorithm* 4 *performs* $O(\log D)$ *arithmetic operations on inputs requiring* $p + \frac{1}{2}\log D + o(1)$ *bits of storage in almost all cases. In particular,* $M(x+y) = O(2^p)$ *almost always.*

Proof. By Theorem 3.4, computing $\mathbf{c}$ takes $O(\log D)$ arithmetic operations on numbers bounded by $O(D)$. By Theorem 3.6, the same is true for the computation of $\mathbf{r}_m$. By Theorem 3.1, in obtaining $\hat{\mathbf{r}}(x+y)$ from $\mathbf{r}_m$, all coefficients computed by the neighbouring algorithm are bounded by $O(\sqrt{D})$. So we only need to prove that $\mathbf{r}_n = \hat{\mathbf{r}}(x+y)$ can be obtained from $\mathbf{r}_m$ in $O(\log D)$ iterations (i.e., $|n-m| = O(\log D)$) and that the maximum value of T_j is bounded by $O(2^p\sqrt{D})$ in almost all cases. From the proof of Theorem 3.7, we have $1/4\sqrt{D} < \theta/U < 4(q+1)$ where q is as in Theorem 3.7. By Lemma 4.1(a), $\lambda(x) < q'+1$, $\lambda(y) < q''+1$ for some partial quotients q', q'' generated by the continued fraction algorithm. Therefore

$$L < g\chi\frac{\theta}{U}\rho(x)\rho(y)\lambda(x)\lambda(y) < 4\chi g^2(q+1)(q'+1)(q''+1)$$

and

$$L \ge g\frac{\theta\gamma^2}{U2^{2p}} > g\frac{1}{4\sqrt{D}}\frac{1}{g^2} = \frac{1}{4g\sqrt{D}}.$$

Distinguish between the same three cases as in Algorithm 4.

Case 1. $M(x+y) = \lceil L2^p/g \rceil \le \lceil 2^p g^2 \rceil$. There is nothing else to prove.

Case 2. From Lemma 3.2(b), $4g\sqrt{D} > 1/L > T_{n-1}/2^p \ge |\theta'_{n-1}/\theta'_m| = \prod_{i=m}^{n-2} |\psi'_i| \ge 2^{(n-m-1)/2}$ or $2^{n-m} < 32g^2 D$. So $n - m = O(\log D)$. Now $T_{n-1} < 2^p/L < 2^{p+2} g\sqrt{D}$, $T_n < (q_{n-2}+1)T_{n-1} < (q_{n-2}+1)g2^{p+2}\sqrt{D}$, and q_{n-2} is almost always small by the Gauss–Kuz'min law. Finally,

$$M(x+y) \le \left\lceil (q_{n-2}+1)\frac{T_{n-1}L}{g} \right\rceil \le \left\lceil (q_{n-2}+1)\frac{2^p}{g} \right\rceil.$$

Case 3. Here

$$4\chi g^2(q+1)(q'+1)(q''+1) > L \ge \frac{T_n}{2^p} = \left|\frac{\zeta'_n}{\zeta'_m}\right| = \left|\frac{\theta'_m}{\theta'_n}\right| = \prod_{i=n}^{m-1} |\psi'_i| > 2^{(m-n)/2},$$

so $2^{m-n} < (4\chi g^2(q+1)(q'+1)(q''+1))^2 = O(1)$ in almost all cases and $O(D^4)$ in the worst case. Hence $m - n = O(\log D)$ and $m - n = O(1)$ almost always. Now $T_{n-1} < (q_{n+1}+1)T_n \le (q_{n+1}+1)L2^p \le 4\chi g^2(q+1)(q'+1)(q''+1)(q_{n+1}+1)2^p$, so again $T_{n-1} = O(2^p)$ in most cases, and T_{n-1} is the largest value computed in the recursion. Finally,

$$\frac{1}{T_n} < \frac{q_{n+1}+1}{T_{n-1}} < \frac{q_{n+1}+1}{L2^p},$$

so $M(x+y) \le \lceil (q_{n+1}+1)(2^p/g) \rceil$. □

Algorithm 5.
Input: $\hat{\mathfrak{r}}(x) \in \mathfrak{R}$ for $x \in \mathbf{R}$, $M(x)$, $m \in \mathbf{Z}_+$.
Output: $\hat{\mathfrak{r}}(mx)$, $M(mx)$.

Algorithm:

1. Obtain the binary decomposition $m = \sum_{i=0}^{r} b_i 2^{r-i}$ of m, $b_i \in \{0, 1\}$, $b_0 = 1$.
2. Set $\hat{\mathfrak{r}}(z_0) = \hat{\mathfrak{r}}(x)$.
3. For $i = 1$ to r do
   (a) Compute $\hat{\mathfrak{r}}(2z_{i-1})$, $M(2z_{i-1})$ using Algorithm 4.
   Set $\hat{\mathfrak{r}}(z_k) = \hat{\mathfrak{r}}(2z_{i-1})$, $M(z_i) = M(2z_{i-1})$.
   (b) If $b_i = 1$, then compute $\hat{\mathfrak{r}}(z_i + x)$, $M(z_i + x)$ using Algorithm 4.
   Set $\hat{\mathfrak{r}}(z_i) = \hat{\mathfrak{r}}(z_i + x)$, $M(z_i) = M(z_i + x)$.
4. Set $\hat{\mathfrak{r}}(mx) = \hat{\mathfrak{r}}(z_r)$, $M(mx) = M(z_r)$.

This algorithm uses a standard exponentiation technique (see, e.g., [12]). From Algorithm 4, it is clear that if $g^{-1} \le \rho(z_{i-1})^2 \le g$ after step 3(a) and $g^{-1} \le \rho(2z_{i-1})\rho(x) \le g$ after step 3(b) in each iteration of the algorithm, we have $M(z_i) \ge \gamma$ and $\hat{\mathfrak{r}}(z_i) \in \{\mathfrak{r}_+(z_i), \mathfrak{r}_-(z_i)\}$ $(1 \le i \le r)$.

Theorem 4.4. *Let $m \in \mathbf{Z}_+$, $x \in \mathbf{R}$, and let $\hat{\mathfrak{r}}(x)$ satisfy the bounds of Theorem* 3.4. *Furthermore, assume that conditions* (i)–(iii) *of Algorithm* 4 *are satisfied for each application of Algorithm* 4 *in step* 3 *of Algorithm* 5. *If m is polynomially bounded in D, then Algorithm* 5 *performs* $O((\log D)^2)$ *arithmetic operations on inputs of* $O(\log D)$ *bits.*

Proof. By Theorem 4.3, since $p = O(\log D)$, all numbers have input size $O(\log D)$. Step 1 of Algorithm 5 takes $O(r) = O(\log m)$ operations and this is $O(\log D)$ if m is polynomially bounded in D. Steps 2 and 4 take $O(1)$ operations. For each iteration, steps 3a and 3b each perform $O(\log D)$ operations. So the number of operations needed for step 3 is $O(r \log D) = O((\log D)^2)$. □

Now that all the required algorithms for our protocol are known, there remain two more problems to be solved:

1. Both communication partners need to start with an initial ideal such that conditions (i)–(iii) of Algorithm 4 are satisfied throughout the protocol, i.e., for each iteration of Algorithm 4.
2. Algorithm 5 computes one of two possible ideals. The two partners need not obtain the same ideal from Algorithm 5 and must be able to agree on a common unique key.

These two problems are solved in Sections 4.3 and 5, respectively.

4.3. *Error Analysis*

Theorem 4.5. *Let* x, y, $\hat{\mathfrak{r}}(x)$, $\hat{\mathfrak{r}}(y)$, $M(x)$, *and* $M(y)$ *be as in Algorithm* 4. *Then*

$$\rho(x)\rho(y) \leq \rho(x + y) < \sqrt{K}(x)\rho(y).$$

Proof. Assume $M(x)$, $M(y) \geq \gamma$, $\hat{\mathfrak{r}}(x) \in \{\mathbf{r}_-(x), \mathbf{r}_+(x)\}$, $\hat{\mathfrak{r}}(y) \in \{\mathbf{r}_-(y), \mathbf{r}_+(y)\}$, and $g^{-1} \leq \rho(x)\rho(y) \leq g$. We only prove the result for Case 3; the reasoning for Cases 1 and 2 is analogous.

$$\rho(x + y) = \frac{\hat{\lambda}(\mathbf{r}_n, x + y)}{\lambda(\mathbf{r}_n, x + y)} \geq \frac{(2^p/T_n)(\hat{\theta}/U)\hat{\lambda}(x)\hat{\lambda}(y)}{|\zeta'_m/\zeta'_n|(\theta/U)\lambda(x)\lambda(y)}$$

$$= \frac{\hat{\theta}}{\theta}\frac{2^p}{T_n}\left|\frac{\zeta'_n}{\zeta'_m}\right|\rho(x)\rho(y) > \rho(x)\rho(y),$$

$$\rho(x + y) < \frac{(2^p/T_n)(\hat{\theta}/U)\hat{\lambda}(x)\hat{\lambda}(y) + 2^{-p}}{|\zeta'_m/\zeta'_n|(\theta/U)\hat{\lambda}(x)\lambda(y)}$$

$$= \rho(x)\rho(y)\left(\frac{2^p}{T_n}\left|\frac{\zeta'_n}{\zeta'_m}\right|\frac{\hat{\theta}}{\theta} + \frac{1}{2^p|\zeta'_m/\zeta'_n|(\theta/U)\hat{\lambda}(x)\hat{\lambda}(y)}\right)$$

$$< \rho(x)\rho(y)\frac{\hat{\theta}}{\theta}\left(\chi + \frac{g}{L2^p}\left|\frac{\zeta'_n}{\zeta'_m}\right|\right) < \rho(x)\rho(y)\chi\left(\chi + \frac{g\chi T_n}{L2^{2p}}\right)$$

$$\leq \rho(x)\rho(y)\chi^2\left(1 + \frac{g}{2^p}\right)$$

$$< \sqrt{K}\rho(x)\rho(y)$$

by Lemma 4.2(c). □

Theorem 4.6. *Let* $m \in \mathbf{Z}$, $1 \le m \le B$, $x \in \mathbf{R}$, $\hat{\mathfrak{r}}(x) \in \{\mathfrak{r}_-(x), \mathfrak{r}_+(x)\}$, $\mathrm{M}(x) \ge \gamma$ *and let*

$$\hat{\mathfrak{r}}(x) = \left[\frac{Q}{\sigma}, \frac{P + \sqrt{D}}{\sigma}\right]$$

be such that $0 < Q < 2\sqrt{D}$, $-1 < (P + \sqrt{D})/Q < 0$. *If*

$$g^{-1/(2B-1)} \le \rho(x) \le \left(\frac{g}{K^{B-1}}\right)^{1/(2B-1)},$$

then $\mathrm{M}(mx) \ge \gamma$, $\hat{\mathfrak{r}}(mx) \in \{\mathfrak{r}_-(mx), \mathfrak{r}_+(mx)\}$, *and*

$$\min\{1, \rho(x)^{2m-1}\} \le \rho(mx) \le \max\{1, \rho(x)^{2m-1}\} K^{m-1}.$$

Proof. Let $U_0 = \max\{1, \rho(x)\} \ge 1$, $L_0 = \min\{1, \rho(x)\} \le 1$, and $2^r \le m < 2^{r+1}$. Define

$$U_{i+1} = KU_0U_i^2, \qquad L_{i+1} = L_0L_i^2 \qquad (0 \le i \le r-1).$$

Then $U_{i+1} \ge U_i \ge 1$ since $K > 1$, and $L_{i+1} \le L_i \le 1$. By induction on i it is easy to prove

$$U_i = U_0^{2^{i+1}-1} K^{2^i-1}, \qquad L_i = L_0^{2^{i+1}-1} \qquad (0 \le i \le r).$$

Then $U_r = U_0^{2^{r+1}-1} K^{2^r-1} \le U_0^{2m-1} K^{m-1} \le U_0^{2B-1} K^{B-1}$, $L_r = L_0^{2^{r+1}-1} \ge L_0^{2m-1} \ge L_0^{2B-1}$. If $U_0 = 1$, then $U_r \le K^{B-1} < A^{6(B-1)} = g^{3(B-1)/8B^2} < g$ by Lemma 4.2(c). If $U_0 = \rho(x)$, then $U_r \le \rho(x)^{2B-1} K^{B-1} \le (g/K^{B-1})K^{B-1} = g$. So in either case $U_i \le g$ for $0 \le i \le r$. Similarly, if $L_0 = 1$, then $L_r \ge 1 > g^{-1}$, and if $L_0 = \rho(x)$, then $L_r \ge \rho(x)^{2B-1} \ge g^{-1}$, so in either case $L_i \ge g^{-1}$ for $0 \le i \le r$.

We show $L_i \le \rho(z_i) \le U_i$ for $0 \le i \le r$. Prove this by induction on i: $L_0 \le \rho(x) \le U_0$ and $\rho(z_0) = \rho(x)$. Using Theorem 4.5, we obtain the following.

Case $b_{i+1} = 0$.

$$\rho(z_{i+1}) = \rho(2z_i) \ge \rho(z_i)^2 \ge L_i^2 \ge L_{i+1} \qquad \text{since} \quad L_0 \le 1.$$

$$\rho(z_{i+1}) < \sqrt{K}\rho(z_i)^2 \le \sqrt{K}U_i^2 \le U_{i+1} \qquad \text{since} \quad K \ge 1, \quad U_0 \ge 1.$$

Case $b_{i+1} = 1$.

$$\rho(z_{i+1}) = \rho(2z_i + x) \ge \rho(z_i)^2 \rho(z_0) \ge L_i^2 L_0 = L_{i+1}.$$

$$\rho(z_{i+1}) < \sqrt{K}\rho(2z_i)\rho(x) < K\rho(z_i)^2 \rho(z_0) \le KU_i^2 U_0 = U_{i+1}.$$

It follows that $L_0^{2m-1} \le L_r \le \rho(z_r) = \rho(mx) \le U_r \le U_0^{2m-1} K^{m-1}$.

Next we prove that $L_i \le \rho(z_{i-1})^2 \le U_i$ and (in case $b_i = 1$) $L_i \le \rho(2z_{i-1})\rho(x) \le U_i$ $(1 \le i \le r)$. Then it follows that $g^{-1} \le \rho(z_{i-1})^2 \le g$ after step 3(a) and $g^{-1} \le \rho(2z_{i-1})\rho(x) \le g$ after step 3(b) in each iteration of Algorithm 5, hence $\mathrm{M}(z_i) \ge \gamma$ and $\hat{\mathfrak{r}}(z_i) \in \{\mathfrak{r}_+(z_i), \mathfrak{r}_-(z_i)\}$ for $1 \le i \le r$, and from the rth iteration $\hat{\mathfrak{r}}(mx) \in \{\mathfrak{r}_-(mx), \mathfrak{r}_+(mx)\}$ and $\mathrm{M}(mx) \ge \gamma$. Again, we prove our claim by induction on i. For simplicity, we let $\rho(z_{-1}) = \rho(2z_{-1}) = 1$.

The case $i = 0$ is $L_0 \le 1 \le U_0$ and $L_0 \le \rho(x) \le U_0$. Now assume that our claim holds for i and prove it for $i + 1$. From our previous result $L_i \le \rho(z_i) \le U_i$.

$L_{i+1} = L_0 L_i^2 \le L_i^2 \le \rho(z_i)^2 \le U_i^2 \le K U_0 U_i^2 = U_{i+1}$. Now assume $b_{i+1} = 1$. From Theorem 4.5, $\rho(2z_i)\rho(x) \ge \rho(z_i)^2 \rho(x) \ge L_i^2 L_0 = L_{i+1}$ and $\rho(2z_i)\rho(x) \le \sqrt{K}\rho(z_i)^2\rho(x) \le \sqrt{K}U_i^2 U_0 \le U_{i+1}$. □

Theorem 4.7. *Let* $a, b \in \mathbf{Z}$, $1 \le a, b \le B$, $c \in \mathbf{R}$, *and let*

$$\hat{\mathbf{r}}(c) = \left[\frac{Q}{\sigma}, \frac{P + \sqrt{D}}{\sigma}\right]$$

be such that $\hat{\mathbf{r}}(c) \in \{\mathbf{r}_-(c), \mathbf{r}_+(c)\}$, $\mathrm{M}(c) \ge \gamma$, $A^{-1} \le \rho(c) \le A$, *and* $0 < Q < 2\sqrt{D}$, $-1 < (P - \sqrt{D})/Q < 0$. *Then* $\mathrm{M}(abc) \ge \gamma$, $\hat{\mathbf{r}}(abc) \in \{\mathbf{r}_-(abc), \mathbf{r}_+(abc)\}$, *and* $g^{-1} \le \rho(abc) \le g$ *where* $\hat{\mathbf{r}}(abc)$ *is obtained by applying Algorithm 5 to* $\hat{\mathbf{r}}(c)$ *and* b *to compute* $\hat{\mathbf{r}}(bc)$, *then applying Algorithm 5 to* $\hat{\mathbf{r}}(bc)$ *and* a.

Proof. We want to apply Theorem 4.6 first to $\hat{\mathbf{r}}(c)$ and b and then to $\hat{\mathbf{r}}(bc)$ and a. Hence we first need to show that $g^{-1/(2B-1)} \le \rho(c) \le (g/K^{B-1})^{1/(2B-1)}$. To prove these inequalities, observe that, by Lemma 4.2(c), $A^{2B-1}K^{B-1} < A^{2B-1}A^{6(B-1)} = A^{8B-7} = g^{(8B-7)/16B^2} < g$, so $A^{2B-1} < g/K^{B-1} < g$ and hence $g^{-1/(2B-1)} < A^{-1} \le \rho(c) \le A < (g/K^{B-1})^{1/(2B-1)}$. By Theorem 4.6, we have $\mathrm{M}(bc) \ge \gamma$, $\hat{\mathbf{r}}(bc) \in \{\mathbf{r}_-(bc), \mathbf{r}_+(bc)\}$, and $\min\{1, \rho(c)^{2b-1}\} \le \rho(bc) \le \max\{1, \rho(c)^{2b-1}\}K^{b-1}$. From Theorem 3.1, we know that $\hat{\mathbf{r}}(bc)$ satisfies $0 < Q < 2\sqrt{D}$, $-1 < (P - \sqrt{D})/Q < 0$, so only $g^{-1/(2B-1)} \le \rho(bc) \le (g/K^{B-1})^{1/(2B-1)}$ remains to be proven.

Assume first that $\rho(c) \ge 1$, then from our above result $\rho(bc) \ge 1 > g^{-1/(2B-1)}$. Using Lemma 4.2(c) again, we see that $g = A^{16B^2} > A^{4B^2}K^{2B^2} > A^{(2B-1)^2}K^{2B(B-1)} = (A^{2B-1}K^{B-1})^{2B-1}K^{B-1}$, so $\rho(bc) \le \rho(c)^{2b-1}K^{b-1} \le \rho(c)^{2B-1}K^{B-1} \le A^{2B-1}K^{B-1} < (g/K^{B-1})^{1/(2B-1)}$. Now consider the case $\rho(c) < 1$. Then $\rho(bc) \ge \rho(c)^{2b-1} \ge \rho(c)^{2B-1} \ge A^{-(2B-1)} = g^{-(2B-1)/16B^2} > g^{-((2B-1)/(4B-2)^2)} > g^{1/(2B-1)}$ and from $g > A^{12B^2} > K^{2B^2} > K^{2B(B-1)} = K^{(B-1)(2B-1)}K^{B-1}$, we obtain $\rho(bc) \le K^{b-1} \le K^{B-1} < (g/K^{B-1})^{1/(2B-1)}$. If follows from Theorem 4.6 that $\mathrm{M}(abc) \ge \gamma$, $\hat{\mathbf{r}}(abc) \in \{\mathbf{r}_-(abc), \mathbf{r}_+(abc)\}$, and $\min\{1, \rho(bc)^{2a-1}\} \le \rho(abc) \le \max\{1, \rho(bc)^{2a-1}\}K^{a-1}$.

We finally need to show that $g^{-1} \le \rho(abc) \le g$. If $\rho(bc) \ge 1$, we have $\rho(abc) \ge 1 > g^{-1}$ and $\rho(abc) \le \rho(bc)^{2a-1}K^{a-1} < \rho(bc)^{2B-1}K^{B-1} \le (g/K^{B-1})K^{B-1} = g$. In the case where $\rho(bc) < 1$, it follows that $\rho(abc) \ge \rho(bc)^{2a-1} \ge \rho(bc)^{2B-1} \ge g^{-1}$ and $\rho(abc) \le K^{a-1} \le K^{B-1} \le A^{6(B-1)} = g^{3(B-1)/8B^2} < g$. □

Lemma 4.8. *Let*

$$\mathbf{r} = (\mu) = \left[\frac{Q}{\sigma}, \frac{P + \sqrt{D}}{\sigma}\right] \in \mathfrak{R},$$

where $\mathbf{r}$ *is obtained from* $\mathbf{O}$ *by applying Algorithm* 1(a) *to* $\mathbf{O}$ *a few times (at least once). Set* $c = \log|\mu|$, $\hat{\mathbf{r}}(c) = \mathbf{r}$, $\mathrm{M}(c) = 2^p$. *Then* $\hat{\mathbf{r}}(c)$ *and* $\mathrm{M}(c)$ *satisfy the conditions of Theorem* 4.7.

Proof. By Theorem 3.1, $0 < Q < 2\sqrt{D}$, $-1 < (P - \sqrt{D})/Q < 0$. Since $\lambda(\mathbf{r}, c) = |\mu|e^{-c} = 1$, we have $\mathbf{r} \in \{\mathbf{r}_-(c), \mathbf{r}_+(c)\}$. Furthermore, $\mathrm{M}(c) = 2^p \ge \gamma$, so $\rho(c) = 1$. □

Now if both communication partners start on an initial ideal $\mathfrak{k}(c)$ generated as described in Lemma 4.8, they can obtain their respective key ideals $\mathfrak{k}(abc)$ such that $M(abc) \geq \gamma$, $\mathfrak{k}(abc) \in \{\mathfrak{r}_-(abc), \mathfrak{r}_+(abc)\}$, and $g^{-1} \leq \rho(abc) \leq g$, and the conditions of Algorithms 4 and 5 as well as those for Theorem 4.3–4.7 are satisfied throughout their entire computation. Given the above bounds on the relative error $\rho(abc)$, we show that the partners are able to agree on a common unique key.

5. Solving the Ambiguity Problem

Before resolving the ambiguity in the ideal computed in the protocol, we need a method for computing from $\mathfrak{k}(abc)$ not only its neighbouring ideals as done in Algorithm 1, but also their approximate distances.

Algorithm 6 (Neighbouring).
Input: $\mathfrak{r}_j \in \mathfrak{R}$, $M(\mathfrak{r}_j, x)$ $(x \in \mathbf{R}, j \geq 2)$.
Output: $\mathfrak{r}_{j+1}, \mathfrak{r}_{j-1} \in \mathfrak{R}$, $M(\mathfrak{r}_{j+1}, x)$, $M(\mathfrak{r}_{j-1}, x)$.

Algorithm: Compute $\mathfrak{r}_{j+1}, \mathfrak{r}_{j-1}$ using Algorithm 1. Compute rational approximations $\hat{\psi}_j$, $\hat{\varphi}_{j-1}$ for $|\psi_j'|$ and $|\varphi_{j-1}'|$, respectively, as follows. Define $s \in \mathbf{Z}^{\geq 0}$ by

$$s = \begin{cases} 0 & \text{if } P_{j-1} < d, \\ \lfloor \log_2(2d+1) \rfloor & \text{if } P_{j-1} = d. \end{cases}$$

Let $t = s + p$ and $\tilde{d} = \lceil 2^t \sqrt{D} \rceil$. Set

$$\hat{\psi}_j = \frac{2^p P_j + d^*}{2^p Q_{j-1}}, \qquad \hat{\varphi}_{j-1} = \frac{\tilde{d} - 2^t P_{j-1}}{2^t Q_{j-1}},$$

$$M(\mathfrak{r}_{j+1}, x) = \lceil \hat{\psi}_j M(\mathfrak{r}_j, x) \rceil, \qquad M(\mathfrak{r}_{j-1}, x) = \lceil \hat{\varphi}_{j-1} M(\mathfrak{r}_j, x) \rceil.$$

Lemma 5.1. *Let* $\mathfrak{r}_{j-1}, \mathfrak{r}_j, \mathfrak{r}_{j+1} \in \mathfrak{R}$, $x \in \mathbf{R}$. *Then*

$$\rho(\mathfrak{r}_j, x) \leq \rho(\mathfrak{r}_{j\pm 1}, x) \leq \frac{1 + 2^{-p}}{1 - M(\mathfrak{r}_{j\pm 1}, x)^{-1}} \rho(\mathfrak{r}_j, x).$$

Furthermore, if t *is as in Algorithm* 6, *then* $t \leq 2(p - \log_2 B) - 9$.

Proof.

$$\frac{\hat{\varphi}_{j-1}}{|\varphi_{j-1}'|} = \frac{2^{-t}\tilde{d} - P_{j-1}}{\sqrt{D} - P_{j-1}} \geq 1.$$

Similarly, $\hat{\psi}_j / |\psi_j'| \geq 1$.

$$\frac{\hat{\varphi}_{j-1}}{|\varphi_{j-1}'|} < \frac{(2^{-t} + \sqrt{D}) - P_{j-1}}{\sqrt{D} - P_{j-1}} < 1 + \frac{1}{2^t(\sqrt{D} - P_{j-1})}.$$

If $P_{j-1} < d$, then $\sqrt{D} - P_{j-1} > 1$, so $\hat{\varphi}_{j-1} / |\varphi_{j-1}'| < 1 + 2^{-t} = 1 + 2^{-p}$. If $P_{j-1} = d$,

then $2d + 1 \leq 2^s$, so

$$\sqrt{D} - P_{j-1} = \frac{D - P_{j-1}^2}{\sqrt{D} + P_{j-1}} = \frac{D - d^2}{\sqrt{D} + d} \geq \frac{1}{\sqrt{D} + d} > \frac{1}{2d + 1} \geq 2^{-s} \quad \text{and}$$

$$\frac{\hat{\varphi}_{j-1}}{|\varphi'_{j-1}|} \leq 1 + \frac{1}{2^{t-s}} = 1 + 2^{-p}.$$

Analogously, we prove $\hat{\psi}_j/|\psi'_j| < 1 + 2^{-p}$.

$$\rho(\mathbf{r}_{j+1}, x) = \frac{\hat{\lambda}(\mathbf{r}_{j+1}, x)}{\lambda(\mathbf{r}_{j+1}, x)} \geq \frac{\hat{\psi}_j \hat{\lambda}(\mathbf{r}_j, x)}{|\psi'_j| \lambda(\mathbf{r}_j, x)} \geq \rho(\mathbf{r}_j, x)$$

and

$$\rho(\mathbf{r}_{j+1}, x) < \frac{\hat{\psi}_j \hat{\lambda}(\mathbf{r}_j, x) + 2^{-p}}{|\psi'_j| \lambda(\mathbf{r}_j, x)} < (1 + 2^{-p})\rho(\mathbf{r}_j, x) + \frac{\rho(\mathbf{r}_{j+1}, x)}{2^p \hat{\lambda}(\mathbf{r}_{j+1}, x)},$$

so

$$\rho(\mathbf{r}_{j+1}, x)\left(1 - \frac{1}{\mathrm{M}(\mathbf{r}_{j+1}, x)}\right) < (1 + 2^{-p})\rho(\mathbf{r}_j, x)$$

and

$$\rho(\mathbf{r}_{j+1}, x) < \frac{1 + 2^{-p}}{1 - \mathrm{M}(\mathbf{r}_{j+1}, x)^{-1}} \rho(\mathbf{r}_j, x).$$

Similarly

$$\rho(\mathbf{r}_{j-1}, x) \geq \rho(\mathbf{r}_j, x) \quad \text{and} \quad \rho(\mathbf{r}_{j-1}, x) < \frac{1 + 2^{-p}}{1 - \mathrm{M}(\mathbf{r}_{j-1}, x)^{-1}} \rho(\mathbf{r}_j, x).$$

If $P_{j-1} < d$, then $s = 0$, so $t = p$. If $P_{j-1} = d$, then since $2^{s-1} < 2d + 1$, i.e., $2^{s-2} \leq d$, we have $2^{s-2} \leq d \leq 2^p/3072B^2 < 2^p/2048B^2$, hence $s < p - 2 \log B - 9$ and $t < 2(p - \log B) - 9$. □

Denote by $\mathbf{r}(x)$ the reduced ideal *closest* to x, i.e., $|\delta(\mathbf{r}(x), x)| < |\delta(\mathbf{r}, x)|$ for any reduced principal ideal $\mathbf{r} \neq \mathbf{r}(x)$. Let $\lambda_1(x) = \lambda(\mathbf{r}(x), x)$; analogously, we define $M_1(x)$, $\hat{\lambda}_1(x)$, $\rho_1(x)$. Clearly, $\mathbf{r}(x) \in \{\mathbf{r}_-(x), \mathbf{r}_+(x)\}$, so Algorithm 4 computes either $\mathbf{r}(x + y)$ or one of its neighbours; similarly, Algorithm 5 generates $\mathbf{r}(mx)$ or one of its neighbours.

Our protocol is such that Alice and Bob are either both able to determine $\mathbf{r}(abc)$ or, if this is impossible, they both obtain $\mathbf{r}_+(abc)$. The next two lemmas give the details for resolving the ambiguity problem.

Lemma 5.2. *Let* $x \in \mathbf{R}$ *and* $g^{-1} \leq \rho(x) \leq g$. *If* $g^{-1} \leq \hat{\lambda}(x) \leq g$, *then* $g^{-2} < \lambda_1(x) < g^2$.

Proof. For brevity omit the argument x. If $\delta = \delta(x) = \delta(\mathbf{r}(x), x)$, then, by defini-

tion, $|\delta_1| \le |\delta|$, which gives four cases, depending on the signs of δ_1 and δ:

1. $\lambda \ge \lambda_1 \ge 1$.
2. $\lambda \le 1/\lambda_1 \le 1$.
3. $\lambda \ge 1/\lambda_1 \ge 1$.
4. $\lambda \le \lambda_1 \le 1$.

Suppose first $\lambda_1 \le g^{-2}$, so $\lambda_1 < 1$. If $\lambda \le 1$, then, from Case 4, $\hat{\lambda} = \rho\lambda \le \rho\lambda_1 < gg^{-2} = g^{-1}$ which is a contradiction. If $\lambda \ge 1$, then, from Case 3, $\hat{\lambda} \ge \rho(1/\lambda_1) > g^{-1}g^2 = g$ which is again a contradiction. The case $\lambda_1 \ge g^2$ follows analogously. □

Lemma 5.3. *Let a, b, c, $\mathfrak{r}(c)$, and $\mathrm{M}(c)$ be as in Theorem 4.7. If $\lambda_1(abc) \ge g^2$ or $\lambda_1(abc) \le g^{-2}$, then $\hat{\mathfrak{r}}(abc) = \mathbf{r}_+(abc)$. If $g^{-2} < \lambda_1(abc) < g^2$, then $\mathbf{r}(abc)$ can be determined from $\hat{\mathfrak{r}}(abc)$.*

Proof. Again omit the argument abc for brevity. If $\lambda_1 \ge g^2$ or $\lambda_1 \le g^{-2}$, then (since $\hat{\lambda} \ge g^{-1}$), by Lemma 5.2, $\hat{\lambda} > g$. It follows that $\lambda = \hat{\lambda}/\rho > gg^{-1} = 1$, so $\mathfrak{r} = \mathbf{r}_+$. Now let $g^{-2} < \lambda_1 < g^2$. From Theorem 4.7, $g^{-1} \le \rho \le g$ and $\hat{\mathfrak{r}} \in \{\mathbf{r}_-, \mathbf{r}_+\}$, so $\mathbf{r} = \hat{\mathfrak{r}}$ or $\mathbf{r}$ is one of the neighbours of $\hat{\mathfrak{r}}$. Therefore, by Lemma 5.1,

$$\rho \le \rho_1 \le \frac{1 + 2^{-p}}{1 - M_1^{-1}}\rho.$$

Now $M_1 = \hat{\lambda}_1 2^p = \rho_1\lambda_1 2^p \ge \rho\lambda_1 2^p > g^{-1}g^{-2}2^p = g^{-3}2^p$, so $1 - M_1^{-1} > 1 - g^3 2^{-p}$ and

$$\frac{1 + 2^{-p}}{1 - M_1^{-1}} < \frac{1 + 2^{-p}}{1 - g^3 2^{-p}}.$$

Hence

$$g^{-3} < \rho\lambda_1 \le \rho_1\lambda_1 \le \frac{1 + 2^{-p}}{1 - g^3 2^{-p}}\rho\lambda_1 < \frac{1 + 2^{-p}}{1 - g^3 2^{-p}}g^3.$$

Since $\rho_1\lambda_1 = \hat{\lambda}_1$, we can determine an ideal $\mathbf{a}$ which is either $\hat{\mathfrak{r}}$ or a neighbour of $\hat{\mathfrak{r}}$ such that

$$g^{-3} \le \hat{\lambda}(\mathbf{a}, abc) < \frac{1 + 2^{-p}}{1 - g^3 2^{-p}}g^3 \quad \text{and} \quad \rho \le \rho(\mathbf{a}, abc) < \frac{1 + 2^{-p}}{1 - g^3 2^{-p}}\rho.$$

If $\mathbf{r} = \psi\mathbf{a}$ where $\psi \in \mathbf{K}$, then

$$|\psi| = \frac{\lambda_1}{\lambda(\mathbf{a}, abc)} = \frac{\lambda_1\rho(\mathbf{a}, abc)}{\hat{\lambda}(\mathbf{a}, abc)} > \frac{g^{-1}\rho}{((1 + 2^{-p})/(1 - g^3 2^{-p}))g^3} \ge \frac{1 - 2^{-p}g^3}{(1 + 2^{-p})g^6}$$
$$> \frac{1}{1 + 1/\sqrt{\Delta}}$$

and

$$|\psi| < gg^3\frac{1 + 2^{-p}}{1 - g^3 2^{-p}}\rho \le \frac{(1 + 2^{-p})g^6}{1 - 2^{-p}g^3} < 1 + \frac{1}{\sqrt{\Delta}},$$

using Lemma 4.2(f). By Lemma 3.2(c), it follows that $\mathbf{a} = \mathbf{r}$ is the ideal closest to abc. □

Now assume that either of the communication partners computes a final ideal $\hat{\mathfrak{r}}(abc)$ with distance $\mathrm{M}(abc)$. He/she then determines the ideal's two neighbours and their respective distances. If among these three ideals there is one, say $\mathbf{a}$, which satisfies

$$\frac{2^p}{g^3} < \mathrm{M}(\mathbf{a}, abc) < \frac{(1+2^p)g^3}{1+2^{-p}g^3},$$

then $\mathbf{b} = \mathbf{r}(abc)$ from the proof of the previous lemma. Otherwise, by the same lemma, we must have $\lambda_1(bac) \leq g^{-2}$ or $\lambda_1(bac) \geq g^2$ and hence $\hat{\mathfrak{r}}(abc) = \mathbf{r}_+(abc)$. With this final observation, we are able to present the entire protocol.

6. The Protocol

1. Both Alice and Bob agree on D and an ideal $\mathbf{r} \in \mathfrak{R}$ (obtained by applying the right neighbour algorithm to the ideal $\mathbf{O} = [1, (\sigma - 1 + \sqrt{D})/\sigma]$ one or more times). They compute $p = \lfloor \log_2(3072dB^2) \rfloor + 1$ and $M = \mathrm{M}(c) = 2^p$ according to Lemma 4.8 where $c = \log|\mu|$, $\mathbf{r} = (\mu)$, i.e., $\mathbf{r} = \hat{\mathfrak{r}}(c)$. D, $\mathbf{r}$, and M can be made public.
2. Alice secretly chooses $a \in \{1, \ldots, B\}$ and from $\hat{\mathfrak{r}}(c)$ computes

$$\hat{\mathfrak{r}}(ac) = \left[\frac{Q_A}{\sigma}, \frac{P_A + \sqrt{D}}{\sigma}\right], \qquad \mathrm{M}(ac) \geq \gamma$$

   using Algorithm 5. She sends the triple $(P_A, Q_A, \mathrm{M}(ac))$ to Bob.
3. Bob secretly chooses $b \in \{1, \ldots, B\}$ and from $\hat{\mathfrak{r}}(c)$ computes

$$\hat{\mathfrak{r}}(bc) = \left[\frac{Q_B}{\sigma}, \frac{P_B + \sqrt{D}}{\sigma}\right], \qquad \mathrm{M}(bc) \geq \gamma$$

   using Algorithm 5. He sends the triple $(P_B, \mathbf{Q}_B, \mathrm{M}(bc))$ to Alice.
4. From $\hat{\mathfrak{r}}(ac)$, $\mathrm{M}(ac)$, and b, Bob computes $\hat{\mathfrak{r}}(bac)$ and its two neighbours as well as their approximate distances (i.e., M values) using Algorithms 5 and 6. If he finds among these an ideal $\mathbf{a}$ such that

$$\frac{2^p}{g^3} < \mathrm{M}(\mathbf{a}, bac) < \frac{(1+2^p)g^3}{1-2^{-p}g^3},$$

   then $\mathbf{a} = \mathbf{r}(bac)$. In this case he sends "0" back to Alice. If he cannot find such an ideal, then he has computed $\mathbf{r}_+(bac)$. In this case he sends "1" to Alice.
5. From $\hat{\mathfrak{r}}(bc)$, $\mathrm{M}(bc)$, and a, Alice computes $\hat{\mathfrak{r}}(abc)$, $\mathrm{M}(abc)$ using Algorithm 5. If she received "0" from Bob, then she computes the neighbours of $\hat{\mathfrak{r}}(abc)$ and their approximate distances and attempts to compute $\mathbf{r}(abc)$. If successful, she sends "0" back to Bob. The common key is then $\mathbf{r}(abc)$. Otherwise the ideal $\hat{\mathfrak{r}}(abc)$ she computed is $\mathbf{r}_+(abc)$. In this case she sends "1" to Bob. If Alice received "1" from Bob, then he was unable to determine $\mathbf{r}(bac)$ in which case the ideal $\hat{\mathfrak{r}}(abc)$ computed by Alice is $\mathbf{r}_+(abc)$. This is then the key.

6. If Bob sent "1," then the ideal $\hat{\mathfrak{r}}(bac) = \mathfrak{r}_+(bac)$ is the key. If Bob sent and received "0," then the ideal $\mathfrak{a}$ he computed in step 4 is the key. If Bob sent "0" and received "1," then Alice was unable to determine $\mathfrak{r}(abc)$. The key is then the ideal $\hat{\mathfrak{r}}(bac) = \mathfrak{r}_+(bac)$ initially computed by Bob. Note that if Bob sends "1," Alice need not reply. Altogether:

Bob	Alice	Key
Sends "0"	Sends "0"	$\mathfrak{r}(abc)$
Sends "0"	Sends "1"	$\mathfrak{r}_+(abc)$
Sends "1"	No reply	$\mathfrak{r}_+(abc)$

The actual key is the bit string given by the binary representation of the coefficients of the key ideal (or any substring thereof).

7. Security

7.1. *The Discrete Logarithm Problem in* $\mathfrak{R}$

The only known way of breaking our scheme (apart from exhaustive search) is to solve the discrete logarithm problem in $\mathfrak{R}$, given as follows: for any given reduced ideal $\mathfrak{r}_j$ ($j \leq l$), find its distance δ_j. If a cryptanalyst can solve any instance of the DLP, clearly he can break our scheme, since on intercepting $\hat{\mathfrak{r}}(ac) = \mathfrak{r}_j$ and $M(ac)$, he can compute $ac \approx \delta_j - \log(M(ac)/2^p) + kR$ for some $k \in \mathbf{Z}$ (similarly for b). Since R is usually larger than ac, k will tend to be quite small; thus an adversary can retrieve the key ideal in a few trials for k values.

Since $\delta_j = \log \mu_j$ for $\mathfrak{r}_j = (\mu_j) \in \mathfrak{R}$, the DLP in $\mathfrak{R}$ is equivalent to the problem of finding, for any reduced principal ideal $\mathfrak{r}_j$, a generator μ_j. It should be pointed out that a fast algorithm for solving the DLP can be used to find the regulator R of $\mathbf{K}$ quickly. Details of this method are given in [3] and [2]. By a result of Schoof [18], we know that if it is possible to find R quickly, then D can be factored quickly. Thus the DLP in $\mathbf{K} = \mathbf{Q}(\sqrt{D})$ is at least as difficult as factoring D.

An algorithm to solve the DLP is sketched in [3]. The first stage employs a method which can also be used to determine R and the structure of the class group of $\mathbf{K}$. Details and an implementation are given in [4]. As an example, we mention that the computation for $\Delta = 10^{40} + 1$ (a discriminant far too small to guarantee security in our scheme) took 8.3 hours on a SparcStation 2. Then the results computed in the first stage are used to solve the actual DLP by an index calculus technique. This second part of the algorithm is explained in the case of imaginary quadratic fields (i.e., $D < 0$) in [15]; an extension to the real quadratic case is outlined in [3]. The overall algorithm seems to be subexponential, with the precomputation in the first stage requiring most of the work. The complexity is $L(\Delta)^{\sqrt{2}+o(1)}$, assuming certain Extended Riemann Hypotheses (ERH), where $L(x) = \exp(\sqrt{\log x \log\log x})$. For large values of D, this method is totally impractical.

7.2. *Choice of D*

To prevent an exhaustive key-search attack, we need to ensure that the number l of reduced principal ideals in $\mathbf{K}$ is sufficiently large. Since $\eta = |\theta'_{l+1}| = \prod_{j=1}^{l} |\psi_i|$, where $\mathbf{r}_1 = (1)$ and $\psi_j = (\sqrt{D} - P_j)/Q_j$ $(1 \le j \le l)$, we have $R = \log \eta = \sum_{j=1}^{l} \log|\psi'_j| < (l/2) \log \Delta$ by Lemma 3.2(c), and therefore $l > 2R/\log \Delta$. Hence we require a lower bound on R.

The analytic class number formula yields $R = (L(1, \chi)/2h)\sqrt{\Delta}$, where $L(s, \chi) = \sum_{k \ge 1} (\chi(k)/k^s)$ is the Dirichlet L-function corresponding to the Kronecker symbol $\chi = (\Delta/\cdot)$ of $\mathbf{K}$ and h is the class number of $\mathbf{K}$. By a result of Littlewood [13], we have $L(1, \chi) > C/(\log \log \Delta)$ (assuming ERH), where $C = \pi^2/12e^{\gamma}(1 + o(1))$ and γ is Euler's constant. Hence

$$l > \frac{C}{h} \frac{\sqrt{\Delta}}{\log \log \Delta},$$

and we need to bound h from above.

If k is the odd part of the class number, i.e., $h = 2^m k$ where $2 \nmid k$, then we can use the heuristics of Cohen and Lenstra [5], [6] to show that the probability that $k > x$ is asymptotic to $1/2x$. This can be done by estimating $\sum_{n \le x, n \text{ odd}} (w(n)/n)$ (for notation see [5]) and using the Tauberian theorem mentioned in the proof of Lemma 5.2 of [6], followed by partial summation. Now it is known (see, for example, [7]) that h is odd if $D = p$, $D = 2p$, or $D = p_1 p_2$ where p is any odd prime and p_1, p_2 are primes congruent to 3 (mod 4). More cases of values of D for which the even part of the class number can be bounded are given in [10]. Thus by selecting such a D value which is large, we expect that it would be most unlikely that $l < \sqrt{D}/(10^{20} \log D \log \log D)$, say. This renders the likelihood of success of a search technique to be very slight indeed.

8. Remarks on the Implementation

The implementation was done in C language using multiprecise integer arithmetic. Unfortunately, the only machine available to us at the time was a DEC MicroVAX. Tests show that a more modern workstation (such as a DECStation 5000) yields computation times which are approximately 100 times faster than those achieved by the MicroVAX.

Examples show that among the three cases of Algorithm 4, Case 1 was never encountered, Case 3 occurred very rarely (at most once per application of Algorithm 5), and Case 2 occurred almost always. This is to be expected since Case 1 $(1 \le L \le g^3)$ permits only a very small range of L values and corresponds to a very unlikely event, namely, having found $\mathfrak{f}(x + y)$ immediately after the reduction step. Using the bounds from Theorem 4.4, we see that the range for L in Case 2 $(1/4g\sqrt{D} < L < 1)$ is much larger than the one for Case 3 $(1 < L < 4\chi g^2(q + 1)(q' + 1)(q'' + 1))$. In addition, as pointed out in the proof of Theorem 4.3, the number of iterations of Algorithm 1(b) in Case 3 was always very small. In fact, it never exceeded two, even for our largest discriminants which were around 200 digits.

In all our examples, we encountered the simple case of the protocol where Bob and Alice both compute $\mathbf{r}_+(abc)$ and only Bob needs to send his bit 1. Again, we expect this since the bounds given in step 4 of the protocol leave an extremely narrow range for $M(\mathbf{a}, bac)$ and force $M(\mathbf{a}, bac)$ to be very close to 2^p.

8.1. *Examples*

Among many examples, we ran our scheme on the two fields K_1, K_2 generated by the square root of $D_1 = 2^{107} - 1$ (a 33-digit prime) and $D_2 = 2^{607} - 1$ (a 183-digit prime), respectively. For K_1, we chose $B = \lfloor\sqrt{D}\rfloor$ as our bound on a and b, requiring $O(D^{3/2})$ bits or roughly 50 decimal digits of precision. The computation time for a 16-digit exponent was 3 minutes 21 seconds. For K_2, we used $B = \lfloor\sqrt{D}\rfloor$ (precision $O(D^{3/2})$ or 275 decimal digits) for a 91-digit exponent and $B = \lfloor\sqrt[4]{D}\rfloor$ (precision $O(D)$) for a 45-digit exponent. We consider the latter bound sufficiently secure. The first exponent took approximately 97 CPU minutes to compute; the second one used 41 minutes of computation time. Recall that, by our previous remark, the above parameters would give us computation times of approximately 2 seconds, 1 minute, and 25 seconds, respectively, on a modern workstation.

8.2. *Improvements*

Considering the above remarks about the frequency of occurrence of the three cases of Algorithm 4 and the number of iterations of Algorithm 1 in each case, we focused our efforts for speed-up on Case 2 of Algorithm 4, i.e., on speeding up Algorithm 1(a). The following version of the continued fraction algorithm due to Tenner (see [22]) is a significant improvement and can be used for both Algorithms 3 and 4. Let

$$\mathbf{a} = \left[\frac{Q}{\sigma}, \frac{P + \sqrt{D}}{\sigma}\right]$$

be a primitive principal ideal. Set

$$P_0 = P, \qquad Q_0 = Q, \qquad Q_{-1} = \frac{D - P_0^2}{Q_0}, \qquad t_0 = \begin{cases} 0 & \text{if } Q > 0, \\ 1 & \text{if } Q < 0, \end{cases}$$

$$P_0 + d + t_0 = q_0 Q_0 + r_0$$

(i.e., $q_0 = \lfloor (P_0 + d + t_0)/Q_0 \rfloor$ and $r_0 = P_0 + d + t_0 - q_0 Q_0$), and, for $j \geq 0$,

$$P_{j+1} = d + t_j - r_j, \qquad Q_{j+1} = Q_{j-1} - a_j(P_{j+1} - P_j),$$

$$t_{j+1} = \begin{cases} 0 & \text{if } Q_{j+1} > 0, \\ 1 & \text{if } Q_{j+1} < 0, \end{cases} \qquad P_{j+1} + d + t_{j+1} = q_{j+1} Q_{j+1} + r_{j+1}$$

(i.e., $q_{j+1} = \lfloor (P_{j+1} + d + t_{j+1})/Q_{j+1} \rfloor$ and $r_{j+1} = P_{j+1} + d + t_{j+1} - q_{j+1} Q_{j+1}$).

This algorithm is particularly useful if division with remainder is a single operation as was the case in our implementation, since q_{j+1} and r_{j+1} are computed in one step. It cuts down the number of divisions and multiplications by half (i.e., from two to one per step) and merely introduces one extra addition if the ideals are reduced.

References

[1] J. A. Buchmann and H. C. Williams, A key-exchange system based on imaginary quadratic fields, *J. Cryptology* **1** (1988), 107–118.

[2] J. A. Buchmann and H. C. Williams, A key-exchage system based on real quadratic fields, *CRYPTO '89 Proceedings*, Springer-Verlag, Berlin, 1990, pp. 335–343.

[3] J. A. Buchmann and H. C. Williams, Quadratic fields and cryptography, in *Number Theory and Cryptography* (J. H. Loxton, ed.), Cambridge University Press, Cambridge, 1990, pp. 9–25.

[4] H. Cohen, F. Diaz y Diaz, and M. Oliver, Calculs de nombres de classes et de régulateurs de corps quadratiques en temps sous-exponentiel, *Séminaire de Théorie des Nombres de Paris*, 1992, to appear.

[5] H. Cohen and H. W. Lenstra, Heuristics on class groups, in *Number Theory* (H. Jager, ed.) (Noordwijkerhout, 1983), Lecture Notes in Mathematics, vol. 1052, Springer-Verlag, New York, 1984, pp. 26–36.

[6] H. Cohen and H. W. Lenstra, Heuristics on class groups of number fields, in *Number Theory* (H. Jager, ed.) (Noordwijkerhout, 1983), Lecture Notes in Mathematics, vol. 1068, Springer-Verlag, New York, 1984, pp. 33–62.

[7] H. Cohn, *Advanced Number Theory*, Dover, New York, 1962.

[8] W. Diffie and M. E. Hellman, New directions in cryptography, *IEEE Trans. Inform. Theory* **22**(6) (1976), 644–654.

[9] L. K. Hua, *Introduction to Number Theory*, Springer-Verlag, New York, 1982.

[10] P. Kaplan, Sur le 2-groupe des classes d'idéaux des corps quadratiques, *J. Reine Angew. Math.* **283/284** (1976), 313–363.

[11] A. Y. Khinchin, *Continued Fractions*, University of Chicago Press, Chicago, 1964.

[12] D. E. Knuth, *The Art of Computer Programming*, vol. 2, Addison-Wesley, Reading, MA, 1981.

[13] J. E. Littlewood, On the class number of the corpus $P(\sqrt{-k})$, *Proc. London Math. Soc.* **27**, (1982), 358–372.

[14] K. S. McCurley, A key distribution scheme based on factoring, *J. Cryptology* **1** (1988), 95–105.

[15] K. S. McCurley, Cryptographic key distribution and computation in class groups, in *Number Theory and Applications* (Proc. NATO Advanced Study Institute on Number Theory and Applications, Banff, 1988) (R. A. Mollin, ed.), Kluwer, Boston, 1989, pp. 459–479.

[16] V. Miller, Use of elliptic curves in cryptography, *Proceedings of Crypto 85*, Springer-Verlag, New York, 1985, pp. 417–426.

[17] R. W. K. Odoni, V. Varadharajan, and P. W. Sanders, Public-key distribution in matrix rings, *Electron. Lett.* **20** (1984), 386–387.

[18] R. J. Schoof, Quadratic fields and factorization, in *Computational Methods in Number Theory* (H. W. Lenstra and R. Tijdeman, eds.), Math. Centrum Tracts, Number 155, Part II, Amsterdam, 1983, pp. 235–286.

[19] D. Shanks, The infrastructure of a real quadratic field and its applications, *Proc. 1972 Number Theory Conference*, Boulder, CO, 1972, pp. 217–224.

[20] Z. Shmuely, Composite Diffie–Hellman public-key generating systems are hard to break, Technical Report No. 356, Computer Science Department, Technion-Israel Institute of Technology, February 1985.

[21] A. J. Stephens and H. C. Williams, Some computational results on a problem concerning powerful numbers, *Math. Comp.* **50**(182) (1989), 619–632.

[22] H. C. Williams and M. C. Wunderlich, On the parallel generation of the residues for the continued fraction factoring algorithm, *Math. Comp.* **48**(177) (1987), 405–423.