J. Cryptology (2006) 19: 211–239
DOI: 10.1007/s00145-005-0357-6

Journal of
CRYPTOLOGY

An Improved Real-Quadratic-Field-Based Key Exchange Procedure*

Michael J. Jacobson, Jr., Renate Scheidler, and Hugh C. Williams
University of Calgary,
2500 University Drive NW,
Calgary, Alberta, Canada T2N 1N4
jacobs@cpsc.ucalgary.ca
{rscheidl,williams}@math.ucalgary.ca

Communicated by Johannes Buchmann

Received 16 October 2003 and revised 25 October 2004
Online publication 27 May 2005

Abstract. To date, the only non-group structure that has been suitably employed as the key space for Diffie–Hellman-type cryptographic key exchange is the infrastructure of a real quadratic (number or function) field. We present an implementation of a Diffie–Hellman-type protocol based on real quadratic number field arithmetic that provides a significant improvement in performance over previous versions of this scheme. This dramatic speed-up is achieved by replacing the ordinary multiplication and reduction procedures for reduced ideals by a new version of the NUCOMP algorithm due to Shanks.

Key words. Cryptographic key exchange, Real quadratic field, Reduced principal ideal, NUCOMP.

1. Introduction

The first Diffie–Hellman-type key exchange protocol that is not based on a group structure was introduced by Buchmann and Williams in 1989 [2]. It uses as its underlying key space the set of reduced principal ideals of a real quadratic number field, which is not a group, but supports instead a so-called *infrastructure*. An initial implementation of this scheme was given in [12], and the method was extended to real quadratic function fields in [13]. Recently, the required precision in the number field scheme was significantly reduced [5], [7] and subsequently shown to depend only on the size of the exponents and not the underlying field itself [7].

While key exchange in real quadratic fields does not perform as efficiently as Diffie–Hellman protocols in other more conventional settings, there are nevertheless good reasons for further exploring and improving this scheme. The underlying discrete logarithm

* All three authors are funded by NSERC of Canada.

problem (DLP) is quite different from its well-known cousins, the DLP for finite fields and the DLP for elliptic curves. It reduces to the problem of finding the distance (or a generator) of a reduced principal ideal, which is provably at least as difficult as integer factorization [14]. Currently, the best known algorithm for solving the DLP in real quadratic fields is of higher complexity than finite field DLP methods, though still subexponential [20]. Its asymptotic complexity is essentially the same as that of the DLP in imaginary quadratic fields.

The implementation in [7] revealed upon profiling that for very large discriminants (2048 bits), the algorithm for ideal reduction took up the lion's share of computing time—up to 97% compared with only 3% for ideal multiplication. Previous numerical experiments [8] showed that the arithmetic of reduced ideals in both real and imaginary quadratic fields can be improved significantly by replacing the ordinary ideal multiplication and reduction procedure by Shanks' NUCOMP algorithm [15], [19]. NUCOMP reduces the magnitude of the intermediate operands from $O(\Delta)$ to $O(\Delta^{1/2})$ in most cases, with a (rarely occurring) worst case of $O(\Delta^{3/4})$; here, Δ is the discriminant of the underlying quadratic field. An additional benefit of NUCOMP is that the relatively expensive reduction operation is almost completely replaced by the more efficient extended Euclidean algorithm; thus, applying NUCOMP should also directly improve our observed bottleneck, the reduction step.

The idea of incorporating NUCOMP into the key exchange protocol of [7] with the purpose of effecting a dramatic speed-up was the primary motivating factor for the work in this paper. However, the algorithms provided here have applications well beyond this scope, for example in the context of regulator and class number computation as well as principal ideal testing and solving norm equations.

The article is organized as follows. We begin with an overview of continued fractions and ideals in Section 2 and an account of NUCOMP in the more illustrative language of ideals in real quadratic fields in Section 3; in [15] and [19] the algorithm is described in terms of binary quadratic forms. This new description provides more information about the algorithm and allows for a more efficient implementation. Next, in Section 4, we introduce a way of essentially approximating unknown ideals by known ideals, thus avoiding the need for working with distances. This is accomplished by utilizing a more flexible version of the idea of (f, p) representations first introduced in [7]; this notion has proved useful in other contexts that require calculations involving ideals, such as the ones mentioned above. We go on to describe fast arithmetic for such (f, p) representations in Section 5, including an adaptation of NUCOMP to produce from two such representations (approximating two possibly unknown ideals) a very good approximation to the product of the two (unknown) ideals. Our algorithms are proven correct in Section 6, where we also provide asymptotic complexity estimates.

In Sections 7 and 8 we illustrate how to use our arithmetic in the context of cryptographic key exchange in real quadratic fields. This resulted in a dramatic speed-up of the protocol without increasing the precision requirements given in [7]; in fact, our parameters were deliberately chosen so that the lower bound on the precision given in [7] remains correct. In addition, we describe how the number of bits that need to be transmitted may be reduced. Although this does not change the run-time of the protocol, it does significantly reduce the communication bandwidth required. We point out that

our scheme can also be adapted to produce ElGamal-type signatures and to transmit cryptographic keys non-interactively. As our new protocol differs slightly from that of [7], some new security considerations are introduced; these are discussed in Section 9. Finally, we discuss our implementation of the improved key exchange procedure and give numerical results in Section 10.

We expect that the reader is familiar with Diffie–Hellman-type key exchange protocols (see for example [3] and [9]). We also assume some knowledge of continued fractions and their relationship to ideals in real quadratic number fields; for details and proofs, consult [21], [17], and [18].

2. Continued Fractions and Ideals

Let D be a positive squarefree integer. As usual, we write the simple continued fraction expansion of a quadratic irrational $\varphi_0 = (P_0 + \sqrt{D})/Q_0$ (P_0, $Q_0 \in \mathbb{Z}$ with Q_0 dividing $P_0^2 - D$) as

$$\varphi_0 = [q_0, q_1, q_2, \ldots, q_{n-1}, \ldots] = [q_0, q_1, q_2, \ldots, q_{n-1}, \varphi_n] \tag{2.1}$$

for any $n \in \mathbb{N}$, where $q_j = \lfloor \varphi_j \rfloor$ and $\varphi_j = (P_j + \sqrt{D})/Q_j$ for $j \in \mathbb{N}$. Here,

$$q_j = \left\lfloor \frac{P_j + \sqrt{D}}{Q_j} \right\rfloor, \qquad P_{j+1} = q_j Q_j - P_j, \qquad Q_{j+1} = \frac{D - P_{j+1}^2}{Q_j} \tag{2.2}$$

for $j \in \mathbb{N}$. If

$$\begin{array}{lll} A_{-2} = 0, & A_{-1} = 1, & A_j = q_j A_{j-1} + A_{j-2}, \\ B_{-2} = 1, & B_{-1} = 0, & B_j = q_j B_{j-1} + B_{j-2}, \end{array} \tag{2.3}$$

then $[q_0, q_1, \ldots, q_j] = A_j/B_j$. Let $j \in \mathbb{N}$ and set

$$\Psi_j = \prod_{i=1}^{j-1} \psi_i \qquad \text{with} \qquad \psi_i = \frac{P_i + \sqrt{D}}{Q_{i-1}} = (-\overline{\varphi}_i)^{-1} \qquad (1 \le i \le j-1), \tag{2.4}$$

where $\overline{\varphi}_i = (P_i - \sqrt{D})/Q_i$ is the conjugate of φ_i. Since $\varphi_i > q_i \ge 1$ for all $i \in \mathbb{N}$, we see that $|\overline{\Psi}_j| \le 1$. We have

$$\Psi_j = A_{j-2} - \overline{\varphi}_0 B_{j-2}. \tag{2.5}$$

If we define Ψ_0 and Ψ_1 ($= 1$) by using (2.5), then it is easy to deduce that $\Psi_{j+2} = q_j \Psi_{j+1} + \Psi_j$ ($j \ge 0$) and

$$\Psi_j \overline{\Psi}_j = (-1)^{j-1} \frac{Q_{j-1}}{Q_0}, \tag{2.6}$$

where $Q_{-1} = (D - P_0^2)/Q_0$.

Let $\mathbb{K} = \mathbb{Q}(\sqrt{D})$ be a real quadratic number field with discriminant $\Delta = (4/\sigma^2)D$ and maximal order $\mathcal{O} = \mathbb{Z}[\omega]$ where $\omega = (\sigma - 1 + \sqrt{D})/\sigma$, $\sigma = 2$ if $D \equiv 1 \pmod 4$ and $\sigma = 1$ otherwise. Every non-zero ideal $\mathfrak{a}$ in $\mathcal{O}$ is a $\mathbb{Z}$-module of the form $\mathfrak{a} = \mathbb{Z}\, SQ/\sigma \oplus \mathbb{Z}\, S(P + \sqrt{D})/\sigma$ with $S, Q, P \in \mathbb{Z}$, $S, Q > 0$, σ dividing Q, and σQ

dividing $D - P^2$. Here, S and Q are unique and P is unique modulo Q. The first basis element is the norm of $\mathfrak{a}$, i.e. $N(\mathfrak{a}) = SQ/\sigma$. For brevity, write $\mathfrak{a} = (S)(Q, P)$ or simply $\mathfrak{a} = (Q, P)$ if $\mathfrak{a}$ is primitive, i.e. $S = 1$. A primitive ideal $\mathfrak{a} = (Q, P)$ in $\mathcal{O}$ is reduced if there exists no non-zero $\alpha \in \mathfrak{a}$ with $|\alpha|, |\overline{\alpha}| < Q/\sigma$.

If $\mathfrak{b}_1 = (Q_0, P_0)$, then (2.2) produces a sequence of pairwise equivalent ideals $\mathfrak{b}_j = (Q_{j-1}, P_{j-1})$ ($j \in \mathbb{N}$) where

$$\mathfrak{b}_j = (\psi_{j-1})\mathfrak{b}_{j-1} = (\Psi_j)\mathfrak{b}_1; \tag{2.7}$$

here, ψ_{j-1} and Ψ_j are given by (2.4) and for any $\alpha \in \mathbb{K}$, (α) denotes the principal fractional ideal generated by α in $\mathcal{O}$. If $\mathfrak{b}_1$ is non-reduced, then $\mathfrak{b}_j$ is reduced as soon as $(P_{j-1} - \sqrt{D})/Q_{j-1} < 0$, or, equivalently, $Q_{j-1} > 0$ and $P_{j-1} < \sqrt{D}$; this happens after no more than $O(\log(Q_0/\sqrt{D}))$ iterations of (2.2). If $\mathfrak{b}_1$ is reduced, then $\mathfrak{b}_j$ is reduced for all $j \in \mathbb{N}$, in which case $0 < P_j < \sqrt{D}$, $0 < Q_j < 2\sqrt{D}$, $N(\mathfrak{b}_j) < \sqrt{\Delta}$, and

$$1 + \frac{1}{\sqrt{\Delta}} < \psi_j < \sqrt{\Delta}, \quad \text{and} \quad \psi_j \psi_{j+1} > 2 \tag{2.8}$$

for all $j \in \mathbb{N}$. The continued fraction expansion (2.1) of φ_0 is always periodic with some period $l \in \mathbb{N}$. If $\mathfrak{b}_1$ is reduced, then the portion following q_0 is indeed purely periodic; that is, l is the minimal positive integer n such that $q_{n+j} = q_j$ for all $j \in \mathbb{N}$. Also, it follows that the sequence $\mathfrak{b}_j$, $1 \le j \le l$, represents all the reduced ideals equivalent to $\mathfrak{b}_1$; in particular, $\mathfrak{b}_{l+1} = \mathfrak{b}_1$ and hence $\Psi_{l+1} = \varepsilon$ is the fundamental unit of $\mathbb{K}$. We write $\rho(\mathfrak{b}_j) = \mathfrak{b}_{j+1}$ and $\rho^{-1}(\mathfrak{b}_j) = \mathfrak{b}_{j-1}$ (with $\rho^{-1}(\mathfrak{b}_1) = \mathfrak{b}_l$). Also $\rho^n(\mathfrak{b}_j) = \rho(\rho^{n-1}(\mathfrak{b}_j)) = \rho^{-1}(\rho^{n+1}(\mathfrak{b}_j))$ for $n \in \mathbb{Z}$.

3. NUCOMP

The infrastructure of the set of reduced principal ideals assures us that for any two reduced principal ideals $\mathfrak{b}'$ and $\mathfrak{b}''$, we can find a reduced principal ideal $\mathfrak{r}$ "close" to their product $\mathfrak{b}'\mathfrak{b}''$; that is, there exists an explicitly computable relative generator $\theta \in \mathbb{K}$ with $\mathfrak{r} = (\theta)\mathfrak{b}'\mathfrak{b}''$ and $|\theta|$ small. The conventional way of computing $\mathfrak{r}$ (and θ) is first to multiply $\mathfrak{b}'$ and $\mathfrak{b}''$ and then reduce the (primitive part of the) product ideal $\mathfrak{b}'\mathfrak{b}''$ using the continued fraction algorithm (2.2). The problem with this method is that ideal multiplication generally doubles the size of the coefficients (from approximately $\sqrt{D}$ to about D), and reduction shrinks them back to their original size $\approx \sqrt{D}$.

NUCOMP avoids this problem by applying a type of reduction—essentially the Euclidean algorithm—to the two input ideals right from the start. The ideal generated by NUCOMP is "almost" reduced; extensive numerical computations indicate that in practice, it is at most two continued fraction steps away from being reduced. Also, the generator of the ideal produced by NUCOMP relative to the product ideal is bounded in absolute value by 1 (see Lemma 6.1). The key advantage of NUCOMP over the traditional multiplication and reduction technique is the fact that throughout the algorithm, all intermediate operands remain of size $O(\sqrt{D})$; in very rare cases, namely when both reduced input ideals have extremely small norm, the operands may grow as large as $O(D^{3/4})$.

More explicitly, let $\mathfrak{b}' = (Q', P')$ and $\mathfrak{b}'' = (Q'', P'')$ be two reduced ideals in $\mathcal{O}$ and let $\mathfrak{c} = (Q, P)$ be given by $\mathfrak{b}'\mathfrak{b}'' = (S)\mathfrak{c}$, with $\mathfrak{c}$ a primitive ideal and $S \in \mathbb{N}$. To find S, Q, and P, we let $S = \gcd(Q'/\sigma, Q''/\sigma, (P' + P'')/\sigma)$ and write $\sigma S = VQ' + WQ'' + Y(P' + P'')$. Then $Q = Q'Q''/\sigma S^2$ and $P = P'' + UQ''/\sigma S$ where $U \equiv W(P' - P'') + YR'' \pmod{Q'/S}$, $0 \le U < Q'/S$, with $R'' = (D - P''^2)/Q''$. We thus expect Q and P to be of magnitude D, i.e. twice as large as P', Q', P'', Q''. Applying the continued fraction algorithm (2.2) to $\mathfrak{c}$ (i.e. to the quadratic irrational $(P + \sqrt{D})/Q$), we obtain $\mathfrak{r}$.

The idea of NUCOMP is as follows. Compute S and U as above; note that $S, U = O(\sqrt{D})$. Now, instead of computing P, Q and then applying the continued fraction algorithm to $\varphi = (P + \sqrt{D})/Q$, we essentially replace the (unknown) quantity φ by its good rational approximation US/Q'. Then the first few partial quotients in the simple continued fraction expansions of φ and US/Q' are identical. We compute the continued fraction expansion of US/Q' using the Euclidean algorithm, until a remainder not exceeding $D^{1/4}$ is obtained; this choice of cut-off point is due to Atkin (see [19]) and is explained in more detail below. At this point, it is possible to "recover" a pair of integers $\tilde{Q}$, $\tilde{P}$ such that $\mathfrak{b} = (\tilde{Q}, \tilde{P})$ is a primitive ideal equivalent to $\mathfrak{b}'\mathfrak{b}''$, and generally $\tilde{Q}, \tilde{P} = O(\sqrt{D})$ ($O(D^{3/4})$ if Q' and Q'' are very small). As mentioned above, $\mathfrak{b}$ tends to be only a few continued fraction steps away from being reduced.

We now describe the process of finding $\tilde{Q}$ and $\tilde{P}$. Suppose $(Q'/S)/U = [q_0, q_1, \ldots, q_m]$ with $m \in \mathbb{N}$ minimal. The partial quotients q_j ($0 \le j \le m$) can easily be computed using the Euclidean algorithm; more precisely, we have

$$b_{-1} = \frac{Q'}{S}, \qquad b_0 = U, \qquad b_j = b_{j-2} - q_{j-1}b_{j-1} \qquad (1 \le j \le m+1),$$

with $b_m = \gcd(Q'/S, U)$ and $b_{m+1} = 0$. Then $q_j = \lfloor b_{j-1}/b_j \rfloor$ for $0 \le j \le m$. If A_j and B_j are as in (2.3), then

$$b_{-1} = A_{j-2}b_j + A_{j-1}b_{j-1} \qquad (0 \le j \le m+1). \tag{3.1}$$

If we put $q'_0 = 0$ and $q'_j = q_{j-1}$ for $1 \le j \le m+1$, then we have

$$\frac{P + \sqrt{D}}{Q} = [0, q_0, q_1, q_2, \ldots] = [q'_0, q'_1, \ldots, q'_{j-1}, \varphi_j]$$

for $1 \le j \le m$, where $\varphi_j = (P'_j + \sqrt{D})/Q'_j$ with P'_j, $Q'_j \in \mathbb{Z}$ and Q'_j divides $P'^2_j - D$. Note that $[q'_0, q'_1, \ldots, q'_i] = A'_i/B'_i$, where $A'_i = B_{i-1}$ and $B'_i = A_{i-1}$ for $-1 \le i \le j-1$. If we set $\mathfrak{b}_{j+1} = (Q'_j, P'_j)$, then $\mathfrak{b}_{j+1}$ is a primitive ideal in $\mathcal{O}$ with $\mathfrak{b}_{j+1} = (\Psi_{j+1}/S)\mathfrak{b}'\mathfrak{b}''$, where $\Psi_{j+1} = B_{j-2} - A_{j-2}(P - \sqrt{D})/Q$ by (2.5). Using the identity $b_{-1}B_i - b_0A_i = (-1)^i b_{i+1}$ for $0 \le i \le m$, it is easy to verify that

$$\Psi_{j+1} = \frac{G_{j-2} + A_{j-2}\sqrt{D}}{Q}, \qquad \text{where} \quad G_{j-2} = (-1)^j b_{j-1}\frac{Q''}{\sigma S} - A_{j-2}P''. \tag{3.2}$$

Set

$$a_{-1} = 0, \qquad a_0 = -1, \qquad a_j = a_{j-2} - q_{j-1}a_{j-1} \qquad (1 \le j \le m+1),$$

so $a_j = (-1)^{j-1}A_{j-1}$ for $-1 \le j \le m+1$. If we put

$$c_{-1} = \frac{Q''}{\sigma S}, \qquad c_0 = \frac{P - P'}{b_{-1}}, \qquad c_j = c_{j-2} - q_{j-1}c_{j-1},$$
$$d_{-1} = P' + P'', \qquad d_0 = \frac{1}{b_{-1}}\left((P' + P'')b_0 - \sigma S R''\right), \qquad d_j = d_{j-2} - q_{j-1}d_{j-1},$$

for $j \in \mathbb{N}$, then a simple induction proof shows that for $-1 \le j \le m+1$,

$$\begin{aligned} c_j &= \frac{1}{b_{-1}}\left(b_j\frac{Q''}{\sigma S} + a_j(P' - P'')\right), \\ d_j &= \frac{1}{b_{-1}}(b_j(P' + P'') + a_j\sigma S R''), \end{aligned} \tag{3.3}$$

where we recall that $R'' = (D - P''^2)/Q''$. Using (2.6), (3.2), and the identity $Q = Q'Q''/\sigma S^2$, we have

$$\begin{aligned} (-1)^j Q'_j &= Q\Psi_{j+1}\overline{\Psi}_{j+1} = \frac{G_{j-2}^2 - A_{j-2}^2 D}{Q} \\ &= \frac{b_{j-1}}{Q'/S}\left(b_{j-1}\frac{Q''}{\sigma S} + a_{j-1}(P' - P'')\right) \\ &\quad -\frac{a_{j-1}}{Q'/S}\left(b_{j-1}(P' + P'') + a_{j-1}\sigma S R''\right), \end{aligned}$$

and hence from (3.3),

$$Q'_j = (-1)^j(b_{j-1}c_{j-1} - a_{j-1}d_{j-1}) \tag{3.4}$$

for $0 \le j \le m$. Similarly, we use (2.4), (2.6), and (3.2) to write

$$\begin{aligned} (-1)^j(P'_j + \sqrt{D}) &= (-1)^j Q'_j\varphi_j = (-)^{j+1}Q'_j\overline{\Psi}_j/\overline{\Psi}_{j+1} = -Q\overline{\Psi}_j\Psi_{j+1} \\ &= -(G_{j-3} - A_{j-3}\sqrt{D})(G_{j-2} + A_{j-2}\sqrt{D})/Q, \end{aligned}$$

so $P'_j = (-1)^j(A_{j-3}A_{j-2}D - G_{j-3}G_{j-2})/Q$ for $0 \le j \le m$. Using the identity $Q = Q'Q''/\sigma S^2$, the expression for G_{j-2} and G_{j-3} given in (3.2), as well as (3.1) and (3.3), we can now show that

$$P'_j = (-1)^j(b_{j-2}c_{j-1} - a_{j-1}d_{j-2}) + P'' \tag{3.5}$$

for $0 \le j \le m$.

Suppose that $b_{-1} > D^{1/4}$. Since the sequence of remainders $(b_j)_{-1\le j\le m+1}$ strictly decreases to zero, there exists an index $n \in \{1, 2, \ldots, m+2\}$ such that $b_{n-1} < D^{1/4} < b_{n-2}$. Since $n \le m+2$ and $m = O(\log(Q'/S))$ by Lamé's theorem, it follows that $n = O(\log D)$. Since by (3.1), $0 < b_{n-2}A_{n-2} \le b_{-1} < 2\sqrt{D}$ and $b_{n-2} > D^{1/4}$, we have $|a_{n-2}| < |a_{n-1}| = A_{n-2} < 2D^{1/4}$. We usually expect q_{n-1} to be small, so $b_{n-2} < (q_{n-1}+1)b_{n-1}$ will not be much larger than $D^{1/4}$. Furthermore, $D^{1/4} < b_{-1} < 2\sqrt{D}$, and, usually, b_{-1} tends to be of magnitude $\sqrt{D}$, so we expect from (3.3) that $c_{n-1}, d_{n-2}, d_{n-1} = O(D^{1/4})$ ($O(\sqrt{D})$ if b_{-1} is close to the lower bound). It follows from (3.4) and (3.5) that Q'_n, $P'_n = O(\sqrt{D})$ most of the time and $O(D^{3/4})$ in rare cases.

Generally, we obtain our desired reduced ideal after applying only very few continued fraction steps to the ideal $\mathfrak{b}_{n+1} = (Q'_n, P'_n)$.

4. (f, p) Representations

In [7] we introduced a technique for representing ideals called (f, p) representations. For a fixed non-negative integer p, such a representation of a primitive ideal $\mathfrak{a}$ was a pair $(\mathfrak{b}, d)$ such that $\mathfrak{b}$ was an ideal equivalent to $\mathfrak{a}$, $d \in \mathbb{N}$, and if $\theta \in \mathbb{K}$ with $\mathfrak{b} = (\theta)\mathfrak{a}$, then $|2^p\theta/d - 1| < 2^{-p} f$ for some $f \in \mathbb{R}$ with $1 \leq f < 2^p$. While this representation was reasonably convenient for the relatively simple algorithms developed in [7], the proofs of their correctness were somewhat complicated. Simply put, this was because we had no control over the size of d in this kind of representation. As the algorithms used here are rather more intricate than those employed in [7], we decided to improve our definition of an (f, p) representation as follows.

Definition 4.1. Let $p \in \mathbb{N}$, $f \in \mathbb{R}$ with $1 \leq f < 2^p$, and let $\mathfrak{a}$ be an ideal in $\mathcal{O}$. An (f, p) *representation* of $\mathfrak{a}$ is a triple $(\mathfrak{b}, d, k)$ where

— $\mathfrak{b}$ is an ideal equivalent to $\mathfrak{a}$, $d \in \mathbb{N}$ with $2^p < d \leq 2^{p+1}$, $k \in \mathbb{Z}$;
— there exists $\theta \in \mathbb{K}$ with $\mathfrak{b} = (\theta)\mathfrak{a}$ and $\left|2^{p-k}\theta/d - 1\right| < f/2^p$.

An (f, p) representation $(\mathfrak{b}, d, k)$ of $\mathfrak{a}$ is *reduced* if $\mathfrak{b}$ is a reduced ideal, and is *near reduced* if it is reduced and the following additional conditions hold:

— $k < 0$;
— if $\mathfrak{b} = (\theta)\mathfrak{a}$ with $\left|2^{p-k}\theta/d - 1\right| < f/2^p$, and if $\mathfrak{b} = \mathfrak{b}_j$, i.e. $\rho(\mathfrak{b}) = \mathfrak{b}_{j+1} = (\psi)\mathfrak{b}$ with $\psi = \psi_j$ given by (2.7), then there exist integers d', k' with $2^p < d' \leq 2^{p+1}$ and $k' \geq 0$ such that $\left|2^{p-k'}\theta\psi/d' - 1\right| < f/2^p$.

Informally speaking, d is a $p+1$ bit integer, $k \approx \log_2 \theta$, $2^{k-p}d$ is an approximation of the (unknown) relative generator θ of $\mathfrak{b}$ with respect to the (unknown) ideal $\mathfrak{a}$ to accuracy $2^{-p} f$, and p is the precision of the approximation. This is a somewhat more flexible notion of an (f, p) representation than the one used in [7] because we can fix the size of d by introducing the new parameter k. Note that the last condition in the definition of near reduced implies that $(\rho(\mathfrak{b}), d', k')$ is itself a reduced (f, p) representation of $\mathfrak{a}$.

It is not hard to see that for any ideal $\mathfrak{a}$ in $\mathcal{O}$ and any $p \in \mathbb{N}$, $(\mathfrak{a}, 2^{p+1}, -1)$ is a $(1, p)$ representation, and hence an (f, p) representation for any $f \in [1, 2^p)$, of $\mathfrak{a}$. Furthermore, if $(\mathfrak{b}, d, k)$ is a near reduced (f, p) representation of some ideal $\mathfrak{a}$ and f is not too large, then the parameters θ and k are small; more exactly, we have the following lemma.

Lemma 4.1. *Let* $(\mathfrak{b}, d, k)$ *be a near reduced* (f, p) *representation of some ideal* $\mathfrak{a}$ *with* $p > 4$ *and* $f < 2^{p-4}$. *If* θ, ψ, *and* k *are as in Definition* 4.1, *then*

$$\frac{15}{16\psi} < \theta < \frac{17}{16} \quad \textit{and} \quad 0 > k > -\log_2\left(\frac{34}{15}\psi\right) \ .$$

Proof. By Definition 4.1 and (2.8), there exist integers d', k' with $2^p < d' \le 2^{p+1}$ and $k' \ge 0$ with

$$1 - \frac{f}{2^p} < \frac{2^{p-k}\theta}{d} < 1 + \frac{f}{2^p}, \qquad 1 - \frac{f}{2^p} < \frac{2^{p-k'}\theta\psi}{d'} < 1 + \frac{f}{2^p}.$$

Since $k < 0$ and $d \le 2^{p+1}$, we have $2^{k-p}d < 1$ and hence

$$\theta < 2^{k-p}d\left(1 + \frac{f}{2^p}\right) < 1 + \frac{2^{p-4}}{2^p} = \frac{17}{16}.$$

Similarly, since $k' \ge 0$ and $d' > 2^p$, we obtain $2^{k'-p}d' > 1$, implying

$$\theta\psi > 2^{k'-p}d'\left(1 - \frac{f}{2^p}\right) > 1 - \frac{2^{p-4}}{2^p} = \frac{15}{16},$$

and hence $\theta > 15/16\psi$. Finally, $k < 0$ by definition of near reduced, and

$$2^{-k} < \frac{d}{2^p\theta}\left(1 + \frac{f}{2^p}\right) < \frac{2^{p+1}}{2^p} \cdot \frac{16}{15}\psi \cdot \frac{17}{16} = \frac{34}{15}\psi. \qquad \square$$

Corollary 4.1. *Under the conditions of Lemma* 4.1, *we have*

$$\frac{15}{16\sqrt{\Delta}} < \theta < \frac{17}{16} \quad \textit{and} \quad 0 > k > -\log_2\left(\frac{34}{15}\sqrt{\Delta}\right).$$

Furthermore, if $\mathfrak{a}$ *is a principal ideal and* $b \in \mathbb{R}$ *with* $b \ge 1$, *then*

$$\theta > \frac{15}{16b} \quad \textit{and} \quad k > -\log_2\left(\frac{34}{15}b\right)$$

with probability approximately $1 - \log_2(1 + b^{-1})$.

Proof. The first set of inequalities follows from $1 < \psi < \sqrt{\Delta}$, which holds by (2.8). For the second set of inequalities, set $\mathfrak{b}_1 = \mathcal{O}$ and $\mathfrak{b} = \mathfrak{b}_j$ for some $j \in \mathbb{Z}$ with $1 \le j \le l$ where l is the period of the continued fraction expansion of ω. Then $\psi = \psi_j$ from (2.7).

Since $\mathfrak{b}_1 = \mathcal{O}$, we have $\varphi_0 = \omega$. The symmetry properties of the continued fraction expansion (2.1) of ω imply $\psi_n = \varphi_{l+1-n}$, so ψ_n is a complete quotient in this expansion for all $n \in \mathbb{N}$. By the Gauss–Kuz'min law, the probability that $\psi_n \ge b$ for any $b \ge 1$ is approximately $\log_2(1 + b^{-1})$. Hence, $\psi \le b$ with probability $1 - \log_2(1 + b^{-1})$, in which case $\theta > 15/16\psi \ge 15/16b$ and $-k < \log_2(34\psi/15) \le \log_2(34b/15)$. $\square$

For example, setting $b = \frac{30}{17} \approx 1.76$, we see that $\theta > \frac{17}{32}$ and $k = -1$ about 64% of the time.

The term "near" is further motivated by the fact that for f sufficiently small, two near reduced (f, p) representations of the same ideal cannot be far away from one another, as the following two lemmas illustrate:

Lemma 4.2. *Let* $(\mathfrak{b}, d, k)$ *and* $(\mathfrak{c}, e, h)$ *be two near reduced* (f, p) *representations of some principal ideal* $\mathfrak{a}$ *with* $p > 4$ *and* $f < 2^{p-4}$. *Then* $\mathfrak{b} \in \{\rho^{-2}(\mathfrak{c}), \rho^{-1}(\mathfrak{c}), \mathfrak{c}, \rho(\mathfrak{c}), \rho^2(\mathfrak{c})\}$.

Proof. Let l be the period of ω, so $\rho^l(\mathfrak{b}) = \mathfrak{b}$. Write $\mathfrak{c} = \rho^i(\mathfrak{b})$ with $i \in \mathbb{Z}$. If $0 \le i \le 2$ or $l-2 \le i \le l-1$, then the claim holds, so assume $3 \le i \le l-3$.

Set $\mathfrak{b}_1 = \mathfrak{b}$. Then with the notation of (2.7), we have $\mathfrak{c} = \mathfrak{b}_{i+1}$, $\rho(\mathfrak{b}) = (\psi_1)\mathfrak{b}$, and $\rho(\mathfrak{c}) = \mathfrak{b}_{i+2} = (\psi_{i+1})\mathfrak{b}_{i+1} = (\psi_{i+1})\mathfrak{c}$. By Lemma 4.1, there exist $\theta, \varphi \in \mathbb{K}$ with $\mathfrak{b} = (\theta)\mathfrak{a}$, $\mathfrak{c} = (\varphi)\mathfrak{a}$, $15/16\psi_1 < \theta < \frac{17}{16}$, and $15/16\psi_{i+1} < \varphi < \frac{17}{16}$. Since $\mathfrak{c} = (\Psi_{i+1}\theta/\varphi)\mathfrak{c}$, there exists $j \in \mathbb{Z}$ with $\Psi_{i+1}\theta/\varphi = \varepsilon^j$, where $\varepsilon = \Psi_{l+1}$ is the fundamental unit of $\mathbb{K}$. Since $3 \le i \le l-3$, (2.8) implies

$$\varepsilon^j = \frac{\theta\psi_1}{\varphi}(\psi_2\psi_3\cdots\psi_i) > \frac{15}{16}\cdot\frac{16}{17}\cdot 2 > 1$$

and

$$\varepsilon^j = \frac{\theta}{\psi_{i+1}\varphi}\Psi_{i+2} < \frac{17}{16}\cdot\frac{16}{15}\Psi_{l-1} = \frac{17}{15}\cdot\frac{\varepsilon}{\psi_{l-1}\psi_l} < \frac{17}{15}\cdot\frac{\varepsilon}{2} < \varepsilon,$$

which is impossible. □

Using similar techniques and Definition 4.1, it is not hard to establish a third lemma:

Lemma 4.3. *Let* $(\mathfrak{b}, d, k)$ *and* $(\mathfrak{c}, e, h)$ *be two near reduced* (f, p) *representations of some principal ideal* $\mathfrak{a}$ *with* $p > 4$ *and* $f < 2^{p-4}$. *Then* $\mathfrak{b} = \rho^i(\mathfrak{c})$ *with* $|i| \le 2$ *by Lemma* 4.2, *and we have reduced* (f, p) *representations* $(\rho(\mathfrak{b}), d', k')$ *and* $(\rho(\mathfrak{c}), e', h')$ *of* $\mathfrak{a}$ *as per Definition* 4.1. *Set* $a = 2^{k-h}d/e$ *and* $a' = 2^{k'-h}d'/e$.

1. *If* $h - k' \le -2$, *then* $0 \le i \le 2$. *If* $k - h' \le -2$, *then* $-2 \le i \le 0$.
2. *If* $a > \frac{8}{7}$, *then* $i = 1$ *or* 2. *If* $a < \frac{7}{8}$, *then* $i = -1$ *or* -2.
   If $a > \frac{7}{12}$, *then* $-1 \le i \le 2$. *If* $a < \frac{12}{7}$, *then* $-2 \le i \le 1$.
3. *Suppose that* $\frac{7}{12} < a < \frac{12}{7}$.
   If $a' > \frac{8}{7}$ *or* $a' < \frac{7}{8}$, *then* $i = 0$ *or* 1. *If* $a' < \frac{12}{7}$, *then* $i = -1$ *or* 0.

We note that throughout our protocol, we will have $p > 4$ and $f < 2^{p-4}$.

5. Arithmetic of (f, p) Representations

We now describe how to perform arithmetic on (f, p) representations. For $a \in \mathbb{R}$, we denote by $\lfloor a \rceil$ the nearest integer to a; that is, $\lfloor a \rceil = \lfloor a + \frac{1}{2} \rfloor$ and $-\frac{1}{2} \le a - \lfloor a \rceil < \frac{1}{2}$.

Theorem 5.1 (Products of (f, p) Representations). *Let* $(\mathfrak{b}', d', k')$ *be an* (f', p) *representation of an ideal* $\mathfrak{a}'$ *and let* $(\mathfrak{b}'', d'', k'')$ *be an* (f'', p) *representation of an ideal* $\mathfrak{a}''$. *Put* $d^* = \lfloor 2^{-p}d'd'' \rceil$, $d^{**} = \lfloor 2^{-(p+1)}d'd'' \rceil$, $k^* = k' + k''$, *and*

$$(d, k) = \begin{cases} (d^*, k^*) & \text{if} \quad d^* \le 2^{p+1}, \\ (d^{**}, k^* + 1) & \text{if} \quad d^{**} \ge 2^p + 1, \\ (2^{p+1}, k^*) & \text{otherwise.} \end{cases}$$

Then $(\mathfrak{b}'\mathfrak{b}'', d, k)$ *is an* (f, p) *representation of the product ideal* $\mathfrak{a}'\mathfrak{a}''$ *where* $f = f^* + \frac{1}{2} + 2^{-(p+1)} f^*$ *and* $f^* = f' + f'' + 2^{-p} f' f''$.

Proof. Using the inequality $-\frac{1}{2} \le a - \lfloor a \rceil < \frac{1}{2}$ for $a = d^*, d^{**}$, we easily obtain $d^* \ge 2^p + \frac{3}{2} + 2^{-p}$ and $d^{**} \le 2^{p+1} + \frac{1}{2}$, so $2^p < d \le 2^{p+1}$ in all three cases of the theorem.

Furthermore, we obtain in the case where $d^* > 2^{p+1}$ that $2d^{**} > 2^{-p} d'd'' - 1 \ge d^* - \frac{3}{2} \ge 2^{p+1} - \frac{1}{2}$, which implies $d^{**} \ge 2^p$, and $d^{**} = 2^p$ in the "otherwise" case. Similarly, if $d^{**} < 2^p + 1$, then $d^* \le 2 \cdot 2^{-(p+1)} d'd'' + \frac{1}{2} < 2(d^{**} + \frac{1}{2}) + \frac{1}{2} \le 2^{p+1} + \frac{3}{2}$, so $d^* \le 2^{p+1} + 1$, and $d^* = 2^{p+1} + 1$ in the "otherwise" case. This also shows that all three cases are mutually exclusive and exhaustive.

Let $\mathfrak{b}' = (\theta')\mathfrak{a}'$ and $\mathfrak{b}'' = (\theta'')\mathfrak{a}''$ with $\theta', \theta'' \in \mathbb{K}^*$. We need to show that

$$1 - \frac{f}{2^p} < \frac{2^{p-k}\theta'\theta''}{d} < 1 + \frac{f}{2^p}. \tag{5.1}$$

By assumption, $|2^{p-k'}\theta'/d' - 1| < 2^{-p} f'$ and $|2^{p-k''}\theta''/d'' - 1| < 2^{-p} f''$. Hence

$$\left(1 - \frac{f'}{2^p}\right)\left(1 - \frac{f''}{2^p}\right) < \frac{2^{2p-k'-k''}\theta'\theta''}{d'd''} < \left(1 + \frac{f'}{2^p}\right)\left(1 + \frac{f''}{2^p}\right). \tag{5.2}$$

Now it is easy to see that $(1 + 2^{-p} f')(1 + 2^{-p} f'') = 1 + 2^{-p} f^*$ and $(1 - 2^{-p} f')(1 - 2^{-p} f'') = 1 - 2^{-p} f^* + 2^{2p-1} f' f'' > 1 - 2^{-p} f^*$. Therefore from (5.2),

$$1 - \frac{f^*}{2^p} < \frac{2^{2p-k'-k''}\theta'\theta''}{d'd''} < 1 + \frac{f^*}{2^p}. \tag{5.3}$$

Suppose $d = d^*$, then $d - \frac{1}{2} \le 2^{-p} d'd'' < d + \frac{1}{2}$. Similarly, if $d = d^{**}$, then $d - \frac{1}{2} \le 2^{-(p+1)} d'd'' < d + \frac{1}{2}$. Multiplying by $2^{k'+k''-k}/d$, we obtain in both cases

$$1 - \frac{1}{2d} \le \frac{2^{k'+k''-k-p} d'd''}{d} < 1 + \frac{1}{2d},$$

and since $d > 2^p$, we have

$$1 - \frac{1}{2^{p+1}} < \frac{2^{k'+k''-k-p} d'd''}{d} < 1 + \frac{1}{2^{p+1}}. \tag{5.4}$$

Finally, if $d = 2^{p+1}$, then

$$1 + \frac{1}{2^{p+2}} = \frac{1}{2^{p+1}}\left(d^* - \frac{1}{2}\right) \le \frac{1}{d} \cdot \frac{d'd''}{2^p} = \frac{1}{2^p} \cdot \frac{d'd'}{2^{p+1}} < \frac{1}{2^p}\left(d^{**} + \frac{1}{2}\right) = 1 + \frac{1}{2^{p+1}},$$

so (5.4) holds in this last case of the theorem as well. Multiplying (5.3) by (5.4) gives

$$\left(1 - \frac{f^*}{2^p}\right)\left(1 - \frac{1}{2^{p+1}}\right) < \frac{2^{p-k}\theta'\theta''}{d} < \left(1 + \frac{f^*}{2^p}\right)\left(1 + \frac{1}{2^{p+1}}\right). \tag{5.5}$$

However, $(1 + 2^{-p} f^*)(1 + 2^{-(p+1)}) = 1 + 2^{-p} f$ and $(1 - 2^{-p} f^*)(1 - 2^{-(p+1)}) = 1 - 2^{-p} f + 2^{-2p} f^* > 1 - 2^{-p} f$, whence (5.1) follows. □

We now present five algorithms for computing with (f, p) representations. The correctness of these algorithms will be proved and asymptotic complexity estimates given in Section 6. The first algorithm removes principal ideal factors from (f, p) representations; that is, given $(\mathfrak{b}, d, k)$ such that $((\mu)\mathfrak{b}, d, k)$ is an (f, p) representation of some ideal (with $\mu \in \mathbb{K}^*$, $\mu > 1$), it outputs an $(f + \frac{9}{8}, p)$ representation $(\mathfrak{b}, d', k')$ of the same ideal.

Algorithm *DIV*

Input: $((\mathfrak{b}, d, k), T, C, s, p)$ where $((\mu)\mathfrak{b}, d, k)$ is an (f, p) representation of some ideal $\mathfrak{a}$ with $\mu = |(A + B\sqrt{D})/C| \geq 1$ ($A, B, C \in \mathbb{Z}$ with $C \neq 0$), $T = 2^s A + B\lfloor 2^s \sqrt{D} \rfloor$, and $s \in \mathbb{Z}^{\geq 0}$ with $2^s |C| > 2^{p+4}|B|$.
Output: An $(f + \frac{9}{8}, p)$ representation $(\mathfrak{b}, d', k')$ of $\mathfrak{a}$.
Algorithm:

1. Set $e = \lfloor 2^{p+3-s}|T/C| \rceil$.
2. Find $t \in \mathbb{Z}^{\geq 0}$ with $2^{t-1} \leq e/8d < 2^t$.
3. Set $d' = \lceil 2^{p+3+t} d/e \rceil$ and $k' = k - t$.

The next algorithm is essentially NUCOMP and subsequent ideal reduction, performed on (f, p) representations.

Algorithm *NUCOMP*

Input: $((\mathfrak{b}', d', k'), (\mathfrak{b}'', d'', k''), p)$ where $(\mathfrak{b}', d', k')$ is a reduced (f', p) representation of an ideal $\mathfrak{a}'$ and $(\mathfrak{b}'', d'', k'')$ is a reduced (f'', p) representation of an ideal $\mathfrak{a}''$. Here, $\mathfrak{b}' = (Q', P')$ and $\mathfrak{b}'' = (Q'', P'')$ with $Q' \geq Q'' > 0$.
Output: A reduced (f, p) representation $(\mathfrak{b}, d, k)$ of $\mathfrak{a}'\mathfrak{a}''$ where $\mathfrak{b} = (Q, P)$, $Q > 0$, $P < \sqrt{D}$, $(P + \sqrt{D})/Q > 1$, $k \leq k' + k'' + 1$, and $f = f^* + \frac{13}{8} + 2^{-(p+1)} f^*$ with $f^* = f' + f'' + 2^{-p} f' f''$.
Algorithm:

1. Compute $G = \gcd(Q'/\sigma, Q''/\sigma)$ and solve $(Q''/\sigma)X \equiv G \pmod{Q'/\sigma}$ for $X \in \mathbb{Z}$, $0 \leq X < Q'/\sigma$.
2. Compute $S = \gcd((P' + P'')/\sigma, G)$ and solve $Y(P' + P'')/\sigma + ZG = S$ for $Y, Z \in \mathbb{Z}$.
3. Put $R'' = (D - P''^2)/Q''$ and $U \equiv XZ(P' - P'') + YR'' \pmod{Q'/S}$, $0 \leq U < Q'/S$.
4. Put $b_{-1} = Q'/S$, $b_0 = U$, $A_{-2} = 0$, $A_{-1} = 1$, $i = 0$.
   Put $e^* = \lfloor 2^{-p} d' d'' \rceil$, $e^{**} = \lfloor 2^{-(p+1)} d' d'' \rceil$, $h^* = k' + k''$,
   $$(e, h) = \begin{cases} (e^*, h^*) & \text{if} \quad e^* \leq 2^{p+1}, \\ (e^{**}, h^* + 1) & \text{if} \quad e^{**} \leq 2^p + 1, \\ (2^{p+1}, h^*) & \text{otherwise.} \end{cases}$$
5. If $b_{-1} < D^{1/4}$ then put $j = 0$ and
   $$Q'_j = \frac{Q'Q''}{\sigma S^2}, \qquad P'_j \equiv P'' + \frac{UQ''}{\sigma S} \pmod{Q'_j}, \qquad T_{j-1} = Q'_j, \qquad s = 0.$$
   Go to 12.

6. While $b_{i-1} > D^{1/4}$ and $b_i \neq 0$ do
   put $q_i = \lfloor b_{i-1}/b_i \rfloor$, $b_{i+1} = b_{i-1} - q_i b_i$, $A_i = q_i A_{i-1} + A_{i-2}$, $i \leftarrow i+1$.
7. Put $a_{i-1} = (-1)^{i-2} A_{i-2}$, $a_{i-2} = (-1)^{i-3} A_{i-3}$,
$$c_{i-1} = \frac{1}{b_{-1}} \left(b_{i-1} \frac{Q''}{\sigma S} + a_{i-1}(P' - P'') \right),$$
$$d_{i-1} = \frac{1}{b_{-1}} \left(b_{i-1}(P' + P'') + a_{i-1}\sigma S R'' \right),$$
$$d_{i-2} = \frac{1}{b_{-1}} \left(b_{i-2}(P' + P'') + a_{i-2}\sigma S R'' \right),$$
$$Q_i = (-1)^i \left(b_{i-1} c_{i-1} - a_{i-1} d_{i-1} \right),$$
$$P_i = (-1)^i (b_{i-2} c_{i-1} - a_{i-1} d_{i-2}) + P'',$$
$$G' = (-1)^{i-1} \left(b_{i-2} \frac{Q''}{\sigma S} - a_{i-2} P'' \right),$$
$$G = (-1)^i \left(b_{i-1} \frac{Q''}{\sigma S} - a_{i-1} P'' \right),$$
$$B' = A_{i-3}, \qquad B = A_{i-2}.$$
8. Put $Q'_0 = |Q_i|$, $t = \operatorname{sgn}(Q_i)$, $P'_0 = P_i$, $q'_0 = \lfloor (P'_0 + \sqrt{D})/Q'_0 \rfloor$ (store q'_0), $m = 1$, $P'_m = q'_{m-1} Q'_{m-1} - P'_{m-1}$, $Q'_m = (D - {P'_m}^2)/Q'_{m-1}$.
9. While $P'_m > \sqrt{D}$ or $Q'_m < 0$ do
   put $q'_m = \lfloor (P'_m + \sqrt{D})/Q'_m \rfloor$ (store q'_m), $m \leftarrow m+1$, $P'_m = q'_{m-1} Q'_{m-1} - P'_{m-1}$, $Q'_m = (D - {P'_m}^2)/Q'_{m-1}$.
10. If $P'_m + Q'_m > \sqrt{D}$ then
    put $q = \lceil (P'_{m-1} - \lfloor \sqrt{D} \rfloor)/Q'_{m-1} \rceil$, $P'_{m-1} \leftarrow P'_{m-1} - q Q'_{m-1}$, $j = m-1$,
    else put $j = m$.
11. Compute $s \in \mathbb{Z}^{\geq 0}$ such that $2^s Q'_j > 6 \cdot 2^{p+4} S D^{1/4}$.
    Put $T_{-1} = 2^s G - B \lfloor 2^s \sqrt{D} \rfloor$, $T_{-2} = t(2^s G' - B' \lfloor 2^s \sqrt{D} \rfloor)$.
    For $m = 0$ to $j-1$, put $T_m = q'_m T_{m-1} + T_{m-2}$.
12. Put $\mathfrak{b} = (Q'_j, P'_j)$. Execute $(\mathfrak{b}, d, k) = \mathrm{DIV}((\mathfrak{b}, e, h), ST_{j-1}, Q'_j, s, p)$.

The idea of this algorithm is as follows. Theorem 5.1 shows that $(\mathfrak{b}'\mathfrak{b}'', e, h)$ is an (f, p) representation of $\mathfrak{a}'\mathfrak{a}''$, with f as in Theorem 5.1 and e, h as in step 4. Steps 1–3, the first line of step 4, and steps 5–7 perform NUCOMP on the ideals $\mathfrak{b}'$ and $\mathfrak{b}''$, producing an "almost" reduced ideal. Steps 8 and 9 apply the continued fraction algorithm to reduce this ideal, and step 10 guarantees the required bounds $Q > 0$, $P < \sqrt{D}$, and $(P + \sqrt{D})/Q > 1$. After step 10, we have obtained a reduced ideal $\mathfrak{b}$ such that $(\mu)\mathfrak{b} = \mathfrak{b}'\mathfrak{b}''$ for some $\mu \in \mathbb{K}$ with $\mu > 1$. Using the recursion on the T_m (which is the same recursion as the one given for Ψ_{j+2} just before (2.6)), a suitable approximation of $2^s \mu$ is generated in step 11, and step 12 uses this approximation to call DIV and find the correct values for d and k.

Throughout our protocol, we wish to use only near reduced representations. Since the output of NUCOMP need not be near reduced, we next describe a routine NEAR that given a reduced (f, p) representation of some ideal finds a near reduced $(f + \frac{9}{8})$ representation of the same ideal.

Algorithm *NEAR*

Input: $((\mathfrak{b}, d, k), p)$ where $(\mathfrak{b}, d, k)$ is a reduced (f, p) representation of some ideal $\mathfrak{a}$ with $k < 0$ and $\mathfrak{b} = (Q, P)$ with $Q > 0$, $P < \sqrt{D}$, and $(P + \sqrt{D})/Q > 1$.
Output: A near reduced $(f + \frac{9}{8}, p)$ representation $(\mathfrak{c}, g, h)$ of $\mathfrak{a}$.
(Optional output: a reduced $(f + \frac{9}{8}, p)$ representation $(\rho(\mathfrak{c}), g', h')$ of $\mathfrak{a}$.)
Algorithm:

1. Find $s \in \mathbb{Z}^{\geq 0}$ with $2^s Q \geq \max\{2^{p+4}, 2^{2|k|+1}\}$.
   Put $P_0 = P$, $Q_0 = Q$, $T_{-2} = -2^s P_0 + \lfloor 2^s\sqrt{D}\rfloor$, $T_{-1} = 2^s Q_0$,
   $M = \lceil 2^{p+s-k} Q_0/d\rceil$, $r_{-2} = 2^s P_0 + \lfloor 2^s\sqrt{D}\rfloor$, $r_{-1} = 2^s Q_0$, $i = 1$.
2. While $T_{i-2} \leq M$ do
   put $q_{i-1} = \lfloor r_{i-3}/r_{i-2}\rfloor$ (store all the q_j),
   $r_{i-1} = r_{i-3} - q_{i-1} r_{i-2}$ (store r_{i-2}),
   $T_{i-1} = q_{i-1} T_{i-2} + T_{i-3}$, (store T_{i-2}), $\quad i \leftarrow i + 1$.
3. Put $G_{i-2} = 2^{-(s+1)}(T_{i-2} + (-1)^{i-1} r_{i-2})$,
   $$B_{i-2} = \frac{T_{i-2} + (-1)^i r_{i-2}}{2\lfloor 2^s\sqrt{D}\rfloor}, \qquad B_{i-3} = \frac{T_{i-3} + (-1)^{i-1} r_{i-3}}{2\lfloor 2^s\sqrt{D}\rfloor},$$
   $$Q_{i-1} = (-1)^{i-1}\frac{G_{i-2}^2 - D B_{i-2}^2}{Q_0}, \qquad P_{i-1} = \frac{G_{i-2} - Q_{i-1} B_{i-3}}{B_{i-2}}.$$
4. While $P_{i-1} + \sqrt{D} < Q_{i-1}$ or $Q_{i-1} < 0$ do
   put $Q_{i-2} = (D - P_{i-1}^2)/Q_{i-1}$, $P_{i-2} = q_{i-2} Q_{i-2} - P_{i-1}$,
   $T_{i-4} = T_{i-2} - q_{i-2} T_{i-3}$ (store T_{i-3}), $i \leftarrow i - 1$.
5. While $T_{i-2} \leq M$ do
   put $q_{i-1} = \lfloor (P_{i-1} + \sqrt{D})/Q_{i-1}\rfloor$, $P_i = q_{i-1} Q_{i-1} - P_{i-1}$, $Q_i = (D - P_i^2)/Q_{i-1}$,
   $T_{i-1} = q_{i-1} T_{i-2} + T_{i-3}$, $i \leftarrow i + 1$.
6. Put $e_{i-1} = \lceil 2^{p-s+3} T_{i-3}/Q_0\rceil$.
   If Q_{i-2} and P_{i-2} were not computed in step 4 then
   put $Q_{i-2} = (D - P_{i-1}^2)/Q_{i-1}$, $P_{i-2} = q_{i-2} Q_{i-2} - P_{i-1}$.
   If $d e_{i-1} \leq 2^{2p-k+3}$ then
   put $\mathfrak{c} = (Q_{i-2}, P_{i-2})$, $e = e_{i-1}$
   (put $\rho(\mathfrak{c}) = (Q_{i-1}, P_{i-1})$, $e' = \lceil 2^{p-s+3} T_{i-2}/Q_0\rceil$),
   else
   if Q_{i-3} and P_{i-3} were not computed in step 4 then
   put $Q_{i-3} = (D - P_{i-2}^2)/Q_{i-2}$, $P_{i-3} = q_{i-3} Q_{i-3} - P_{i-2}$,
   put $\mathfrak{c} = (Q_{i-3}, P_{i-3})$, $e = \lceil 2^{p-s+3} T_{i-4}/Q_0\rceil$
   (put $\rho(\mathfrak{c}) = (Q_{i-2}, P_{i-2})$, $e' = e_{i-1}$).
7. Find t with $2^t < ed/2^{2p+3} \leq 2^{t+1}$. Put $g = \lceil ed/2^{p+t+3}\rceil$, $h = k + t$.
   (Find t' with $2^{t'} < e'd/2^{2p+3} \leq 2^{t'+1}$. Put $g' = \lceil e'd/2^{p+t'+3}\rceil$, $h' = k + t'$.)

Once again, we sketch the idea behind this algorithm. If $\mathfrak{c} = (\theta)\mathfrak{b}$ with $\theta > 0$, then θ is determined by the continued fraction algorithm applied to the ideal $\mathfrak{b}$, yielding ideals $\mathfrak{b}_1 = \mathfrak{b}, \mathfrak{b}_2, \mathfrak{b}_3, \ldots$, where $\mathfrak{b}_i = (\Psi_i)\mathfrak{b}$ with Ψ_i given by (2.4). For $i \geq 0$, the quantity T_{i-2} is an approximation to $2^s\Psi_i$. However, the divisions in the continued fraction algorithm (2.2) can be costly, so we use a NUCOMP-like trick. Suppose $\mathfrak{b} = (Q, P)$. Instead

of applying continued fraction steps to the irrational $\alpha = (P + \sqrt{D})/Q$, we perform the Euclidean algorithm on the rational approximation $(2^s P + \lfloor 2^s \sqrt{D} \rfloor)/2^s Q$ of α in steps 1 and 2. This can be shown to yield the correct partial quotients, provided the coefficient B_{i-2} of $\sqrt{D}$ in Ψ_i satisfies $B_{i-2}^2 < 2^{s-1} Q$ (see Lemma 6.2 below). In step 3 we "pick up" the coefficients P_{i-1}, Q_{i-1} of $\mathfrak{b}_i$ and G_{i-2}, B_{i-2} of Ψ_i, but it is possible (though unlikely) that we have gone slightly too far in our rational continued fraction expansion. Therefore, we "back up" in step 4 to a point where the values of P_{i-1} and Q_{i-1} are correct and the corresponding ideal is reduced. If necessary, we continue with the continued fraction algorithm applied to P_{i-1} and Q_{i-1} in step 5. Finally, steps 6 and 7 produce our desired near reduced representation.

We now combine the previous two algorithms into one operation that on input of two near reduced representations outputs a near reduced representation of the product of the two ideals represented by the inputs.

Algorithm *NEAR-PRODUCT*

Input: $((\mathfrak{b}', d', k'), (\mathfrak{b}'', d'', k''), p)$ where $(\mathfrak{b}', d', k')$ is a near reduced (f', p) representation of an ideal $\mathfrak{a}'$ ($\mathfrak{b}' = (Q', P')$) and $(\mathfrak{b}'', d'', k'')$ is a near reduced (f'', p) representation of an ideal $\mathfrak{a}''$ ($\mathfrak{b}'' = (Q'', P'')$).

Output: A near reduced $(f^* + \frac{11}{4} + 2^{-(p+1)} f^*, p)$ representation $(\mathfrak{c}, g, h)$ of $\mathfrak{a}'\mathfrak{a}''$ with $f^* = f' + f'' + 2^{-p} f' f''$.
(Optional output: a reduced $(f^* + \frac{11}{4} + 2^{-(p+1)} f^*, p)$ representation $(\rho(\mathfrak{c}), g', h')$ of $\mathfrak{a}$.)

Algorithm:

1. If $Q' \geq Q''$ then
   $(\mathfrak{b}, d, k) = \text{NUCOMP}((\mathfrak{b}', d', k'), (\mathfrak{b}'', d'', k''), p)$
   else
   $(\mathfrak{b}, d, k) = \text{NUCOMP}((\mathfrak{b}'', d'', k''), (\mathfrak{b}', d', k'), p)$.
2. $(\mathfrak{c}, g, h) = \text{NEAR}((\mathfrak{b}, d, k), p)$.
   $(((\mathfrak{c}, g, h), (\rho(\mathfrak{c}), g', h')) = \text{NEAR}((\mathfrak{b}, d, k), p)$.)

We can now use the standard binary exponentiation technique to do "exponentiation" on near reduced (f, p) representations:

Algorithm *EXP*

Input: $((\mathfrak{b}_0, d_0, k_0), n, p)$ where $n \in \mathbb{N}$ and $(\mathfrak{b}_0, d_0, k_0)$ is a near reduced (f_0, p) representation of some ideal $\mathfrak{a}$.

Output: A near reduced (f, p) representation $(\mathfrak{b}, d, k)$ of $\mathfrak{a}^n$ for suitable $f \in [1, 2^p)$.

Algorithm:

1. Compute the binary representation of n, say $n = \sum_{i=0}^{l} b_i 2^{l-i}$
   ($b_0 = 1$, $b_i \in \{0, 1\}$ for $1 \leq i \leq l$, $l = \lfloor \log_2 n \rfloor$).
2. Set $(\mathfrak{b}, d, k) = (\mathfrak{b}_0, d_0, k_0)$.
3. For $i = 1$ to l do
   (a) $(\mathfrak{b}, d, k) = \text{NEAR-PRODUCT}((\mathfrak{b}, d, k), (\mathfrak{b}, d, k), p)$.
   (b) If $b_i = 1$ then
   $(\mathfrak{b}, d, k) = \text{NEAR-PRODUCT}((\mathfrak{b}, d, k), (\mathfrak{b}_0, d_0, k_0), p)$.

6. Proofs of Correctness

Theorem 6.1. *Algorithm DIV is correct.*

Proof. We need to prove $2^p < d' \le 2^{p+1}$ and

$$1 - \frac{f + \frac{9}{8}}{2^p} < \frac{2^{p-k'}\theta}{\mu d'} < 1 + \frac{f + \frac{9}{8}}{2^p}. \tag{6.1}$$

Since $2^s|C| > 2^{p+4}|B|$, we have

$$\left| \frac{A + B\sqrt{D}}{C} - \frac{T}{2^s C} \right| < \frac{|B|}{2^s|C|} < \frac{1}{2^{p+4}},$$

and hence $\operatorname{sgn}(T) = \operatorname{sgn}(A + B\sqrt{D})$, implying

$$\left| 2^{p+3}\mu - 2^{p-s+3}\frac{|T|}{|C|} \right| < \frac{1}{2}.$$

It follows that $|2^{p+3}\mu - e| < 1$, so $e > 2^{p+3}\mu - 1 \ge 2^{p+3} - 1$, i.e. $e \ge 2^{p+3}$. Therefore $t \ge 0$, and it is now not hard to show that $2^p < d' \le 2^{p+1}$.

Let $(\mu)\mathfrak{b} = (\theta)\mathfrak{a}$. Then

$$\left| \frac{2^{p-k}\theta}{d} - 1 \right| < \frac{f}{2^p}, \qquad \left| 2^{p+3}\frac{\mu}{e} - 1 \right| < \frac{1}{e} \le \frac{1}{2^{p+3}},$$

so

$$\frac{1 - 2^{-p}f}{1 + 2^{-(p+3)}} < 2^{-(k+3)}\,\frac{\theta e}{\mu d} < \frac{1 + 2^{-p}f}{1 - 2^{-(p+3)}}. \tag{6.2}$$

Now

$$1 + \frac{f}{2^p} < 1 + \frac{f}{2^p} + \frac{1}{2^p}\left(1 - \frac{f + \frac{9}{8}}{2^{p+3}}\right) = \left(1 + \frac{f + \frac{9}{8}}{2^p}\right)\left(1 - \frac{1}{2^{p+3}}\right).$$

Since $2^{p-k'}\theta/\mu d' \le 2^{-(k+3)}\theta e/\mu d$ by the definition of d', (6.2) now yields the right inequality of (6.1).

Now $2^p < d' < 2^{p+t+3}d/e + 1$ implies

$$1 - \frac{1}{2^p} < 1 - \frac{1}{d'} < \frac{2^{p+t+3}d}{ed'},$$

so

$$\frac{2^{p-k'}\theta}{\mu d'} = 2^{-(k+3)}\,\frac{\theta e}{\mu d}\,\frac{2^{p+t+3}d}{ed'} > 2^{-(k+3)}\,\frac{\theta e}{\mu d}\left(1 - \frac{1}{2^p}\right) > \frac{(1 - 2^{-p})(1 - 2^{-p}f)}{1 + 2^{-(p+3)}},$$

where the last inequality follows from (6.2). Furthermore,

$$\left(1 + \frac{1}{2^{p+3}}\right)\left(1 - \frac{f + \frac{9}{8}}{2^p}\right) = 1 - \frac{f+1}{2^p} - \frac{f + \frac{9}{8}}{2^{2p+3}} < \left(1 - \frac{1}{2^p}\right)\left(1 - \frac{f}{2^p}\right),$$

yielding the left inequality of (6.1). □

Before we formally prove Algorithm NUCOMP correct, we require an auxiliary lemma.

Lemma 6.1. *Assume that $D \geq 256$ and let b_i $(i \geq -1)$ be as in steps 4 and 6 of NUCOMP. Suppose $b_{n-1} < D^{1/4} < b_{n-2}$. If Ψ_{n+1} is given by (3.2), then $|\Psi_{n+1}| < S$.*

Proof. Since $\Psi_1 = 1$, this is clear for $n = 0$, so suppose $n \geq 1$. We note that from (3.2),

$$\Psi_{n+1} = \frac{1}{b_{-1}}\left((-1)^n b_{n-1} - \overline{\varphi''}\sigma S A_{n-2}\right), \tag{6.3}$$

where $\varphi'' = (P'' + \sqrt{D})/Q''$. Since $\mathfrak{b}'' = (Q'', P'')$ is reduced, $-1 < \overline{\varphi''} < 0$ by Theorem 4.2 of [18]. We have $\Psi_2 = (|\overline{\varphi''}|\sigma S - b_0)/b_{-1} < \sigma S/b_{-1} \leq S$ since σ divides b_{-1}, and $\Psi_2 > -b_0/b_{-1} > -1 \geq -S$. From (6.3), $\Psi_3 > 0$. Also,

$$\Psi_3 = q_1'\Psi_2 + \Psi_1 = q_0\Psi_2 + 1 < \left\lfloor \frac{b_{-1}}{b_0} \right\rfloor \frac{\sigma S - b_0}{b_{-1}} + 1 \leq \frac{\sigma S}{b_0} \leq S,$$

since σ divides b_0. Finally, note that for $n \geq 3$, we have $A_{n-2} \geq A_1 = q_0 q_1 + 1 \geq 2$, so with (3.1),

$$|\Psi_{n+1}| < \frac{b_{n-1} + \sigma S A_{n-2}}{b_{n-2}A_{n-2}} < \frac{1}{A_{n-2}} + \frac{\sigma S}{D^{1/4}} \leq \frac{1}{2} + \frac{S}{2} \leq S. \qquad \square$$

Theorem 6.2. *If $D \geq 256$, then Algorithm NUCOMP is correct and performs $O(\log D)$ integer operations.*

Proof. After step 4, $(\mathfrak{b}'\mathfrak{b}'', e, h)$ is an $(f^* + \frac{1}{2} + 2^{-(p+1)}f^*, p)$ representation of $\mathfrak{a}'\mathfrak{a}''$ by Theorem 5.1. Let $(S)\mathfrak{b}_1 = \mathfrak{b}'\mathfrak{b}''$, $\mathfrak{b}_1 = (Q_0, P_0)$ where $Q_0 = Q'Q''/\sigma S^2$.

Suppose first that $b_{-1} < D^{1/4}$. In this case, $Q = Q_0' = Q_0$ and $P = P_0' = P_0$. Then $0 < P < Q \leq b_{-1}^2 < \sqrt{D}$ and hence $(P + \sqrt{D})/Q > 1$. Since $Q < \sqrt{D}$, $\mathfrak{b}_1$ is reduced by Theorem 3.4 of [18]. So after step 12, $\mathfrak{b} = \mathfrak{b}_1$, and $(\mathfrak{b}, d, k)$ is a reduced (f, p) representation of $\mathfrak{a}'\mathfrak{a}''$.

Assume now that $b_{-1} > D^{1/4}$. After step 6, we have $b_{i-1} < D^{1/4} < b_{i-2}$. From (3.4), (3.5), and (3.2), it follows that at the end of step 7, we have an ideal $\mathfrak{b}_{i+1} = (Q_i, P_i)$ where $\mathfrak{b}_{i+1} = (\Psi_{i+1})\mathfrak{b}_1$ and $\Psi_{i+1} = (G + B\sqrt{D})/Q$, $\Psi_i = (G' + B'\sqrt{D})/Q$, with $G, B, G', B' \in \mathbb{Z}$.

Let $\mathfrak{c}_1 = (Q_0', P_0')$ and note that $\Psi_0' = t\psi_i^{-1}$. Then step 9 outputs an ideal $\mathfrak{c}_{m+1} = (Q_m', P_m') = (Q_{i+m}, P_{i+m})$ where m is minimal with $P_m' < \sqrt{D}$ and $Q_m' > 0$, so $\overline{\varphi_m'} < 0$ where $\varphi_m' = (P_m' + \sqrt{D})/Q_m'$. We claim that step 10 produces a reduced ideal $\mathfrak{c}_{j+1} = (\Psi_{j+1}')\mathfrak{c}_1 = (Q_j', P_j')$ with $Q_j' > 0$, $P_j' < \sqrt{D}$, $\varphi_j' > 1$, and $|\Psi_{j+1}'| \leq 1$. To see this, suppose first that $P_m' + Q_m' > \sqrt{D}$, so $j = m - 1$. Then $-1 < \overline{\varphi_m'} < 0$, so $\mathfrak{c}_m$ is reduced and $Q_{m-1}' > 0$ by Theorem 4.3 of [18]. Furthermore, $|\Psi_m'| \leq 1$ by Theorem 4.4 of [18].[1] The modification of P_{m-1}' in step 10 ensures $P_{m-1}' < \sqrt{D}$ and $\varphi_{m-1}' > 1$. If,

[1] In the proof of Theorem 4.4, it was assumed that $0 < P_0' < Q_0'$; however, we note that this was only needed when $m = 2$. Nonetheless in this case, $\overline{\varphi_1'} > 0$, so $\Psi_2' = \psi_1' < 0$ by (2.4). Furthermore, $\psi_1' = q_0' - \overline{\varphi_0'} > \varphi_0' - 1 - \overline{\varphi_0'} = 2\sqrt{D}/Q_0' - 1 > -1$, so $|\Psi_2'| < 1$.

on the other hand, $P'_m + Q'_m < \sqrt{D}$, so $j = m$, then $\overline{\varphi'_m} < -1$, so $\mathfrak{c}_{m+1}$ is reduced by Theorem 3.5 of [18] and $|\Psi'_{m+1}| = |\Psi'_m/\overline{\varphi'_m}| < 1$.

Now $|\Psi'_{j+1}| \leq 1$ and, by Lemma 6.1, $|\Psi_{i+1}| \leq S$, so if we put $\mu = S/\Psi$ with $\Psi = |\Psi_{i+1}\Psi'_{j+1}|$, then $\mu \geq 1$. Now, by (2.6),

$$|\Psi\overline{\Psi}| = |\Psi_{i+1}\overline{\Psi}_{i+1}\Psi'_{j+1}\overline{\Psi'}_{j+1}| = \frac{Q_i}{Q_0}\,\frac{Q'_j}{Q'_0} = \frac{Q'_j}{Q_0}$$

since $Q'_0 = Q_i$ by step 8. If we set $\Psi = (\tilde{G} + \tilde{B}\sqrt{D})/Q_0$ with $\tilde{G}$, $\tilde{B} \in \mathbb{Z}$, then

$$\mu = \frac{S\overline{\Psi}}{\Psi\overline{\Psi}} = \frac{S(\tilde{G} - \tilde{B}\sqrt{D})}{Q'_j}.$$

Furthermore, by (3.2),

$$\begin{aligned} |\tilde{B}|\sqrt{D} \;&\leq\; \frac{Q_0}{2}(|\Psi| + |\overline{\Psi}|) \leq \frac{Q_0}{2}(|\Psi_{i+1}| + |\overline{\Psi}_{i+1}|) \leq b_{i-1}\,\frac{Q''}{\sigma S} + (P'' + \sqrt{D})A_{i-2} \\ &<\; D^{1/4} \cdot 2\sqrt{D} + 2\sqrt{D} \cdot 2D^{1/4} = 6D^{3/4}. \end{aligned}$$

Hence $|\tilde{B}| < 6D^{1/4}$, implying that in step 11, $2^s Q'_j/S > 2^{p+4}|\tilde{B}|$.

Now $T_{j-1} = 2^s\tilde{G} - \tilde{B}\lfloor 2^s\sqrt{D}\rfloor$. Since $(\mu)\mathfrak{b} = \mathfrak{b}'\mathfrak{b}''$ with $\mathfrak{b} = \mathfrak{c}_{j+1}$, the output of step 12 produces a reduced (f, p) representation of $\mathfrak{a}'\mathfrak{a}''$. Also, since $k = h - t$, where $t \geq 0$ and $h \leq k' + k'' + 1$, we have $k \leq k' + k'' + 1$.

To determine the asymptotic complexity of NUCOMP, note that steps 1–3 require $O(\log D)$ integer operations. The loop in step 6 is executed at most $O(\log D)$ times since $b_{-1} \leq Q' < 2\sqrt{D}$. The same is true for the loop in step 9, since Q'_m is bounded by a polynomial in $\sqrt{D}$ and it takes no more than $O(\log(Q'_m/\sqrt{D}))$ steps to obtain a reduced ideal (which happens after step 10). Finally, since $\mathfrak{b} = (Q'_j, P'_j)$ is reduced, we have $j = O(m) = O(\log D)$. □

Once again, we require some preliminary results before we can prove the correctness of NEAR. The following lemma illustrates a NUCOMP-like strategy, namely replacing the continued fraction algorithm on a quadratic irrational by its much faster rational counterpart, the Euclidean algorithm.

Lemma 6.2. *Let $\varphi_0 = (P_0 + \sqrt{D})/Q_0$ be a quadratic irrational with $Q_0 > 0$, $P_0 < \sqrt{D}$, and $\varphi_0 > 1$. Set $\hat{\varphi}_0 = (2^s P_0 + \lfloor 2^s\sqrt{D}\rfloor)/2^s Q_0$ for some $s \in \mathbb{N}$. Let $\hat{\varphi}_0 = [\hat{q}_0, \hat{q}_1, \hat{q}_2, \ldots, \hat{q}_m]$ be the simple continued fraction expansion of $\hat{\varphi}_0$ (with $m \in \mathbb{N}$ minimal) and let $\varphi_0 = [q_0, q_1, q_2, \ldots, q_n, \varphi_{n+1}]$ $(n \leq m - 1)$ be the simple continued fraction expansion of φ_0. Set*

$$B_{-2} = 1, \qquad B_{-1} = 0, \qquad B_j = \hat{q}_j B_{j-1} + B_{j-2} \qquad (0 \leq j \leq m).$$

If $B_{n+1}^2 \leq 2^{s-1}Q_0$, then $\hat{q}_j = q_j$ for $j = 0, 1, \ldots, n$.

Proof. We have $0 \le \varphi_0 - \hat{\varphi}_0 < 2^{-s} < 1$, so $q_0 = \hat{q}_0$. Assume now that our claim holds for $j = n-1 \ge 0$ and suppose that $B_{n+1}^2 \le 2^{s-1}Q_0$. Since the B_j strictly increase for $0 \le j \le m$, we inductively get $\hat{q}_j = q_j$ for $0 \le j \le n-1$ and only need to show $\hat{q}_n = q_n$. Since $q_j \ge 1$ for $0 \le j \le n$ and the simple continued fraction expansion of any quadratic irrational is unique, it suffices to show that $\varphi_n > 1$.

Define $A_{-2} = 0$, $A_{-1} = 1$, and $A_j = \hat{q}_j A_{j-1} + A_{j-2}$ for $0 \le j \le n$. Then $A_j/B_j = [\hat{q}_0, \hat{q}_1, \ldots, \hat{q}_j] = [q_0, q_1, \ldots, q_j]$ for $0 \le j \le n-1$. If we set $\lambda_j = (-1)^{j-1}(A_j Q_0 - B_j(P_0 + \sqrt{D}))$, then, by (2.5) and (2.4), $\lambda_j = (-1)^{j+1}\overline{\Psi}_{j+2}Q_0 = Q_0 \prod_{i=1}^{j+1} \varphi_i^{-1} > 0$ for $-1 \le j \le n-1$. Furthermore, by (2.4),

$$\varphi_j = -\frac{\overline{\Psi}_j}{\overline{\Psi}_{j+1}} = \frac{\lambda_{j-2}}{\lambda_{j-1}} \qquad (-1 \le j \le n). \tag{6.4}$$

Set $r_{-2} = 2^s P_0 + \lfloor 2^s\sqrt{D} \rfloor$, $r_{-1} = 2^s Q_0$, and $r_j = r_{j-2} - \hat{q}_j r_{j-1}$ for $0 \le j \le m$. An easy induction argument shows that $r_j = (-1)^{j-1}(A_j r_{-1} - B_j r_{-2})$ and hence $2^s\lambda_j = r_j + (-1)^j B_j(2^s\sqrt{D} - \lfloor 2^s\sqrt{D} \rfloor)$ for $-1 \le j \le n-1$. It follows that

$$r_j - B_j < 2^s\lambda_j < r_j + B_j \qquad (-1 \le j \le n-1). \tag{6.5}$$

Furthermore, another simple induction argument yields $r_{-1} = B_j r_{j+1} + B_{j+1} r_j$, so $r_{-1} < 2B_{j+1}r_j$ for $-1 \le j \le m-1$. In particular, for $j = n$,

$$2^s Q_0 = r_{-1} < 2B_{n+1}r_n \le \frac{2^s Q_0 r_n}{B_{n+1}} < \frac{2^s Q_0 r_n}{B_n},$$

so $B_n < r_n$ and hence

$$B_{n-2} + B_{n-1} \le B_n < r_n \le r_{n-2} - r_{n-1}. \tag{6.6}$$

From (6.5) and (6.6), $2^s\lambda_{n-2} > r_{n-2} - B_{n-2} > r_{n-1} + B_{n-1} > 2^s\lambda_{n-1}$, and hence $\varphi_n > 1$ by (6.4). □

Lemma 6.3. *With the notation of Lemma* 6.2, *let* $\varphi_j = (P_j + \sqrt{D})/Q_j$ *for* $j = 1, 2, \ldots, n$. *Then for* $0 \le j \le n$,

$$\begin{aligned} Q_j &= (-1)^j \frac{G_{j-1}^2 - DB_{j-1}^2}{Q_0}, \\ P_j &= (-1)^j \frac{DB_{j-1}B_{j-2} - G_{j-1}G_{j-2}}{Q_0} = \frac{G_{j-1} - Q_j B_{j-2}}{B_{j-1}}, \end{aligned}$$

where the B_i *are given as in Lemma* 6.2 *and* $G_i = A_i Q_0 - B_i P_0$ *for* $-2 \le i \le n$.

Proof. From (6.4),

$$\frac{P_j + \sqrt{D}}{Q_j} = \varphi_j = \frac{\lambda_{j-2}}{\lambda_{j-1}} = -\frac{G_{j-2} - B_{j-2}\sqrt{D}}{G_{j-1} - B_{j-1}\sqrt{D}} \qquad (0 \le j \le n+1).$$

By clearing denominators and comparing coefficients of 1 and $\sqrt{D}$, we obtain $G_{j-1} = P_j B_{j-1} + Q_j B_{j-2}$ and $DB_{j-1} = P_j G_{j-1} + Q_j G_{j-2}$. Using the identities $B_{j-1}G_{j-2} - B_{j-2}G_{j-1} = (B_{j-1}A_{j-2} - B_{j-2}A_{j-1})Q_0 = (-1)^{j-1}Q_0$, we can solve for P_j, Q_j and derive the desired result. □

Theorem 6.3. *If $D \geq 256$, then Algorithm NEAR is correct and performs $O(|k|)$ integer operations.*

Proof. Since $d \leq 2^{p+1}$ and $k < 0$, we have $M \leq 2^{s-k}Q_0$. Therefore, $T_{-1} \leq M$ and the loop in step 2 is entered. Since $P_0 < \sqrt{D}$, we have $T_{-2} \geq 0$ and hence $T_0 \geq q_0 T_{-1} \geq T_{-1} > 0$. Therefore, the sequence T_j ($j \geq 0$) in step 2 is increasing. Let $j \in \{-1, 0, \ldots, i-3\}$ where i (≥ 2) is determined by $T_{i-3} \leq M < T_{i-2}$, and let A_j, B_j, G_j be as in Lemmas 6.2 and 6.3, with $\hat{q}_j$ replaced by q_j as defined in step 2. Then an easy induction argument shows that $T_j = (-1)^{j-1}r_j + 2\lfloor 2^s\sqrt{D}\rfloor B_j$, so

$$B_j = \frac{T_j + (-1)^j r_j}{2\lfloor 2^s\sqrt{D}\rfloor}$$

and

$$\begin{aligned} G_j &= A_j Q_0 - B_j P_0 = 2^{-s}(A_j r_{-1} - B_j(r_{-2} - \lfloor 2^s\sqrt{D}\rfloor)) \\ &= 2^{-s}((-1)^{j-1}r_j + B_j\lfloor 2^s\sqrt{D}\rfloor) = 2^{-(s+1)}\left((-1)^{j-1}r_j + T_j\right), \end{aligned}$$

which are exactly the expressions given in step 3 for $j = i-2$ and $i-3$. Now

$$\begin{aligned} T_j &= A_j r_{-1} - B_j r_{-2} + 2\lfloor 2^s\sqrt{D}\rfloor B_j = 2^s A_j Q_0 + B_j(\lfloor 2^s\sqrt{D}\rfloor - 2^s P_0) \\ &\geq 2^s A_j Q_0 \geq 2^s B_j Q_0, \end{aligned}$$

since $P_0 < \sqrt{D}$ and $A_j/B_j = [q_0, q_1, \ldots, q_j] \geq q_0 \geq 1$. Therefore $2^s B_{i-3}Q_0 \leq T_{i-3} \leq M \leq 2^{s+|k|}Q_0$, and hence $B_{i-3}^2 \leq 2^{2|k|} \leq 2^{s-1}Q_0$ by the definition of s in step 1. By Lemma 6.2, $(P_0+\sqrt{D})/Q_0 = [q_0, q_1, \ldots, q_{i-4}, \varphi_{i-3}]$ where P_{i-4} and Q_{i-4} are given by Lemma 6.3 (with $j = i-4$). We use the formulae in Lemma 6.3 to define Q_{i-1} and P_{i-1} in step 3. However, by the time the loop in step 2 has terminated, we may have proceeded a few steps further in our rational continued fraction expansion than the point determined by the bound in Lemma 6.2. That is, the quantities P_{i-1} and Q_{i-1} in step 3 may not be correct; however, they will be correct (and the corresponding ideal will be reduced) if $P_{i-1}+\sqrt{D} > Q_{i-1} > 0$, because in that case, $(P_0+\sqrt{D})/Q_0 = [q_0, q_1, \ldots, q_{i-2}, \varphi_{i-1}]$ with $\varphi_{i-1} = (P_{i-1} + \sqrt{D})/Q_{i-1} > 1$. If these conditions do not hold, we "back up" at most three iterations in step 4 until we reach a loop index where the ideal coefficients are once again guaranteed to be correct and we have a reduced ideal, at which point we pick up the correct continued fraction expansion in step 5.

At the beginning of step 6, we once again have $T_{i-2} > M \geq T_{i-3}$ where $i \geq 2$. Now set $e_j = \lceil 2^{p-s+3}T_{j-2}/Q_0\rceil$ for $j \in \mathbb{N}$. For $j \geq 3$, we have $T_{j-2} \geq T_{j-3} + T_{-1}$ and hence

$$e_j \geq 2^{p-s+3}\frac{T_{j-3} + 2^s Q_0}{Q_0} = 2^{p-s+3}\frac{T_{j-3}}{Q_0} + 2^{p+3} > e_{j-1} - 1 + 2^{p+3},$$

so $e_j \geq e_{j-1} + 2^{p+3} \geq e_{j-1} + 2$. Since $M \geq 2^{p+s-k} Q_0/d$, we have $de_i > 2^{p-s+3} dM/Q_0 \geq 2^{2p-k+3}$.

Suppose $de_{i-1} > 2^{2p-k+3}$. Then $i \geq 3$ (since $de_1 = 2^{p+3} < 2^{2p-k+3}$) and

$$\begin{aligned} e_{i-2} &\leq e_{i-1} - 2 < 2^{p-s+3} \frac{T_{i-3}}{Q_0} - 1 \leq 2^{p-s+3} \frac{M}{Q_0} - 1 \\ &< \frac{2^{p-s+3}}{Q_0} \left(2^{p+s-k} \frac{Q_0}{d} + 1 \right) - 1 = \frac{2^{2p-k+3}}{d} + \frac{2^{p+3}}{2^s Q_0} - 1 \\ &\leq \frac{2^{2p-k+3}}{d} - \frac{1}{2} < \frac{2^{2p-k+3}}{d} \end{aligned}$$

since by step 1, $2^s Q_0 \geq 2^{p+4}$. It follows that $de_{i-2} \leq 2^{2p-k+3}$, so after step 6, $de \leq 2^{2p-k+3} < de'$, yielding $t' \geq -k \geq t + 1$ and hence $h < 0 \leq h'$ in step 7. Furthermore, the definition of g and g' in step 7 easily yields $2^p < g, g' \leq 2^{p+1}$.

Write $e = \lceil 2^{p-s+3} T/Q_0 \rceil$ where $T = 2^s G + B\lfloor 2^s \sqrt{D} \rfloor$ ($G \in \mathbb{Z}$, $B \in \mathbb{N}$). Then in step 6, $\mathfrak{c} = (\Psi)\mathfrak{b}$ with $\Psi = (G + B\sqrt{D})/Q_0$ and $|\overline{\Psi}| < 1 < \Psi$. Since $\mathfrak{c} = (\Psi\theta)\mathfrak{a}$, it remains to show that

$$\left| 2^{p-h} \frac{\Psi\theta}{g} - 1 \right| < \frac{f + \frac{9}{8}}{2^p}. \tag{6.7}$$

We first claim that

$$\left| 2^{p+3} \frac{\Psi}{e} - 1 \right| < \frac{1}{2^{p+3}}. \tag{6.8}$$

To prove (6.8), we observe that $0 < \Psi - 2^{-s} T/Q_0 < 2^{-s} B/Q_0 \leq 2^{-(p+4)} B$, so since $e - 1 < 2^{p-s+3} T/Q_0 \leq e$, it follows that $-1 < 2^{p+3}\Psi - e < B/2$. If $B \leq 2$, then $e > 2^{p+3}\Psi - B/2 > 2^{p+3} - 1$, so $e \geq 2^{p+3}$ and $|2^{p+3}\Psi/e - 1| < e^{-1} \leq 2^{-(p+3)}$. Suppose now that $B \geq 3$. Then since $Q_0 < 2\sqrt{D}$, we have $\Psi > 2B\sqrt{D}/Q_0 - |\overline{\Psi}| > B - 1$, so

$$e > 2^{p+3}(B-1) - \frac{B}{2} = 2^{p+2} B + B\left(2^{p+2} - \tfrac{1}{2}\right) - 2^{p+3} > 2^{p+2} B,$$

since $B \geq 3$ and $2^{p+2} > \frac{3}{2}$. Therefore $|2^{p+3}\Psi/e - 1| < B/2e < 2^{-(p+3)}$.

Using (6.8) and the inequality $|2^{p-k}\theta/d - 1| < 2^{-p} f$, we can use exactly the same reasoning as in the proof of Theorem 6.1 (with μ replaced by Ψ) to derive (6.7).

Now the total number of steps performed by the loops in steps 2 and 5 is $O(i)$ where $T_{i-3} < M \leq T_{i-2}$. As outlined above, the loop in step 4 is executed at most three times and thus plays no role in the asymptotic complexity of the algorithm. Since $T_{-2} \geq 0$, a simple induction argument shows that $T_j \geq \tau^j T_{-1}$ for $j \in \mathbb{N}$, where $\tau = (1 + \sqrt{5})/2$, so $\tau^{i-3} < M/T_{-1} \leq 2^{s-k} Q_0/2^s Q_0 = 2^{|k|}$. □

Theorem 6.4. *If $D \geq 256$, then Algorithm NEAR-PRODUCT is correct and performs $O(\log D)$ integer operations.*

Proof. After step 1, we have $k \leq k' + k'' + 1 < 0$ since $k', k'' < 0$. So the output of step 1 of NEAR-PRODUCT is a legitimate input to step 2.

By Theorems 6.2 and 6.3, the output of NEAR-PRODUCT is a near reduced $(\tilde{f}, p)$ representation of $\mathfrak{a}'\mathfrak{a}''$ where $\tilde{f} = f^* + \frac{13}{8} + 2^{-(p+1)} f^* + \frac{9}{8} = f^* + \frac{11}{4} + 2^{-(p+1)} f^*$.

Furthermore, $|k| \leq |k'| + |k''| + 1 = O(\log \Delta)$ by Corollary 4.1, so, by Theorems 6.2 and 6.3, the algorithm performs $O(\log D)$ integer operations. □

Theorem 6.5. *If $D \geq 256$, then Algorithm EXP is correct and performs $O(\log n \log D)$ integer operations.*

7. The Protocol

Suppose that $f', f'' < 2^{p-4}$ for the inputs of NEAR-PRODUCT. Then $f^* = f' + f'' + 2^{-p} f' f'' < 2^{p-1}$, so $2^{-(p+1)} f^* < \frac{1}{4}$. Hence NEAR-PRODUCT generates a near reduced (f, p) representation, where $f = f^* + 3$. This is the same bound on f as in Algorithm MR of [7] and shows that the precision analysis of Section 3 of [7] and the lower bound on p given in Lemma 4.1 of [7] remain valid here.

We give a slight improvement of the result of Lemma 4.1 in [7] that allows for the transmission of fewer bits in our proposed Diffie–Hellman protocol.

Lemma 7.1. *Let $(\mathfrak{b}, d, k)$ be a near reduced (f, p) representation of some ideal $\mathfrak{a}$. Let $r \in \mathbb{N}$ with $r < p$. Set $d' = 2^r \lceil 2^{-r} d \rceil$. Then $(\mathfrak{b}, d', k)$ is a near reduced $(f + 2^r, p)$ representation of $\mathfrak{a}$.*

The proof of this lemma is straightforward using Definition 4.1.

Corollary 7.1. *Let $\mathfrak{r}$ be any reduced principal ideal and let $p, a, b, B \in \mathbb{Z}$ with $B \geq 12$, $0 < a, b \leq B$, and $2^p \geq 50B^2 \max\{16, \log_2 B\}$. Set $r = \lfloor \log_2 B \rfloor$ and*

$$\begin{aligned} (\mathfrak{a}, d_a, k_a) &= \mathrm{EXP}((\mathfrak{r}, 2^{p+1}, -1), a, p), \\ (\mathfrak{k}, d, k) &= \mathrm{EXP}((\mathfrak{a}, 2^r \lceil 2^{-r} d_a \rceil, k_a), b, p). \end{aligned}$$

Then $(\mathfrak{k}, d, k)$ is a near reduced (f, p) representation of $\mathfrak{r}^{ab}$ with $f < 2^{p-4}$.

Proof. Set $h = \max\{16, \log_2 B\}$. Using the same reasoning as in Theorem 3.9 and Lemma 4.1 of [7], we deduce that $(\mathfrak{a}, d_a, k_a)$ is a near reduced (g, p) representation of $\mathfrak{r}^a$ with $g \leq 13.33B$. By Lemma 7.1, $(\mathfrak{a}, 2^r \lceil 2^{-r} d_a \rceil, k_a)$ is a near reduced $(g + 2^r, p)$ representation of $\mathfrak{r}^a$ where $g + 2^r \leq g + B \leq 14.33B$. Since $B \geq 12$, we have $0.8481B > 9.9$, so $h(3.43 \cdot 14.33B + 9.9)b < 50B^2 h \leq 2^p$. By Theorem 3.9 of [7], $(\mathfrak{k}, d, k)$ is a near reduced (f, p) representation of $\mathfrak{r}^{ab}$ where $hf < 2^p$, so $f < 2^p / h \leq 2^{p-4}$. □

Corollary 7.1 and Lemma 4.2 now immediately yield the following.

Theorem 7.1. *Let $\mathfrak{r}, r, p, a, b, B$ be as in Corollary 7.1 and set*

$$\begin{aligned} (\mathfrak{a}, d_a, k_a) &= \mathrm{EXP}((\mathfrak{r}, 2^{p+1}, -1), a, p), \\ (\mathfrak{b}, d_b, k_b) &= \mathrm{EXP}((\mathfrak{r}, 2^{p+1}, -1), b, p), \\ (\mathfrak{k}, d, k) &= \mathrm{EXP}((\mathfrak{a}, 2^r \lceil 2^{-r} d_a \rceil, k_a), b, p), \\ (\mathfrak{m}, e, h) &= \mathrm{EXP}((\mathfrak{b}, 2^r \lceil 2^{-r} d_b \rceil, k_b), a, p). \end{aligned}$$

Then $(\mathfrak{k}, d, k)$ *and* $(\mathfrak{m}, e, h)$ *are near reduced* (f, p) *representations of* $\mathfrak{r}^{ab}$ *with* $f < 2^{p-4}$ *and* $\mathfrak{k} \in \{\rho^{-2}(\mathfrak{m}), \rho^{-1}(\mathfrak{m}), \mathfrak{m}, \rho(\mathfrak{m}), \rho^2(\mathfrak{m})\}$.

To exchange a common key (which in our context will be any desired portion of the coefficients P, Q of a reduced ideal $\mathfrak{k} = (Q, P)$), two parties Alice and Bob first publicly agree on a large squarefree positive integer D (≥ 256), a reduced principal ideal $\mathfrak{r}$ in the maximal order $\mathcal{O}$ of $\mathbb{K} = \mathbb{Q}(\sqrt{D})$, and a bound $B \in \mathbb{N}$ on the exponents. Good choices of D yielding a high level of security, for example forcing D to be non-square modulo several small primes, were suggested in [7]. Hamdy [4] gives recommendations for sizes of D providing specific levels of security. Both communicants also precompute $p = \lceil \log_2(50B^2 \max\{16, \log_2 B\}) \rceil$ and $r = \lfloor \log_2 B \rfloor$ (note that $p > 4$). They then execute the following protocol.

Protocol 7.1 (Cryptographic Key Exchange).

1. Alice
   (a) secretly generates $a \in \mathbb{N}$, $a \leq B$;
   (b) computes $(\mathfrak{a}, d_a, k_a) = \text{EXP}((\mathfrak{r}, 2^{p+1}, -1), a, p)$;
   (c) sends $(\mathfrak{a}, \lceil 2^{-r} d_a \rceil, k_a)$ to Bob.
2. Bob
   (a) secretly generates $b \in \mathbb{N}$, $b \leq B$;
   (b) computes $(\mathfrak{b}, d_b, k_b) = \text{EXP}((\mathfrak{r}, 2^{p+1}, -1), b, p)$;
   (c) sends $(\mathfrak{b}, \lceil 2^{-r} d_b \rceil, k_b)$ to Alice.
3. Alice computes $(\mathfrak{k}, d, k) = \text{EXP}((\mathfrak{b}, 2^r \lceil 2^{-r} d_b \rceil, k_b), a, p)$.
4. Bob computes $(\mathfrak{m}, e, h) = \text{EXP}((\mathfrak{a}, 2^r \lceil 2^{-r} d_a \rceil, k_a), b, p)$.

Note that Alice transmits roughly $\log_2 \Delta + \log_2 B \log_2 \log_2 B$ bits: the coefficients of the ideal $\mathfrak{a}$ are of approximate size $\log_2(\sqrt{\Delta})$, $|k_a|$ tends to be very small, and $2^{-r} d_a \approx 2^{p-r} \approx 50B \log_2 B$; similarly for Bob. This is an improvement of approximately $\log_2(B)$ bits over the protocol in [7].

Asymptotically, Protocol 7.1 requires the same number of integer operations as the key exchange protocol in Section 4 of [7]. Nevertheless, as evidenced by the data presented in Section 10, our new protocol is significantly faster. The main advantages of NUCOMP are that the sizes of operands involved are smaller than standard ideal multiplication and reduction and that the continued fraction expansion of quadratic irrationals is replaced by the computationally simpler extended Euclidean algorithm. Neither of these reduce the number of integer operations required, but both have a positive effect on performance in practice.

By Theorem 7.1, $\mathfrak{k} \in \{\rho^{-2}(\mathfrak{m}), \rho^{-1}(\mathfrak{m}), \mathfrak{m}, \rho(\mathfrak{m}), \rho^2(\mathfrak{m})\}$. We have $\mathfrak{k} = \mathfrak{m}$ essentially all the time; in fact, in our numerical experiments, which consisted of the 5000 key exchanges computed to generate the runtimes in Table 2 plus the numerous protocol runs conducted during the process of optimizing the code, this was always true. If Alice and Bob have doubts about whether they computed the same ideal, they can choose $D \equiv 3 \pmod 4$ and execute another small protocol that guarantees them a common key ideal. Before we describe this final procedure in the next section, we explain how a user can employ a technique similar to Protocol 7.1 to transmit a cryptographic key of her choice to another user.

Suppose Alice wishes to send a private key $K \in \{0, 1\}^n$ to Bob. In addition to the public parameters required for Protocol 7.1, they both agree on a public hash function h that maps reduced ideals (i.e. their $\mathbb{Z}$-bases (Q, P)) into the set $\{0, 1\}^n$ of n-bit strings. Using the same notation as above, Bob generates a secret exponent b and publicizes $(\mathfrak{b}, \lceil 2^{-r} d_b \rceil, k_b)$. Now Alice looks up Bob's public entry $(\mathfrak{b}, \lceil 2^{-r} d_b \rceil, k_b)$, generates a secret exponent a, and computes first $(\mathfrak{a}, d_a, k_a)$, then $(\mathfrak{k}, d, k)$, and finally $S = K \oplus h(\mathfrak{k})$. She sends S and $(\mathfrak{a}, \lceil 2^{-r} d_a \rceil, k_a)$ to Bob. Bob now computes $(\mathfrak{m}, e, h)$ and $K = S \oplus h(\mathfrak{m})$ to obtain the secret key K.[2] Alice could even have sent a ciphertext $C = E_K(M)$ (where M is some secret message) to Bob along with the rest of the information, which Bob could immediately decrypt to $M = D_K(C)$.

We finally point out that by using ideas of [1] and [11], Protocol 7.1 can also be adapted to exchange ElGamal-type signatures.

8. The Final Key Agreement Protocol

We now describe a short protocol that establishes a provably unique key. To this extent, we assume that $D \equiv 3 \pmod 4$. Suppose Bob computed $(\mathfrak{m}, e, h)$ and Alice computed $(\mathfrak{k}, d, k)$ with $\mathfrak{k} = (Q, P)$. From her last call of NEAR-PRODUCT (within EXP), she also knows $(\rho(\mathfrak{k}), d', k')$.

Protocol 8.1 (Final Key Agreement).

1. Alice
   (a) puts $q \equiv Q \pmod 4$, $0 \le q \le 3$;
   (b) puts $b_1 = 0$ if $d \le 3 \cdot 2^{p-1}$ and $b_1 = 1$ otherwise;
   puts $b_2 = 0$ if $d' \le 3 \cdot 2^{p-1}$ and $b_2 = 1$ otherwise;
   (c) puts
   $$(a_0, a_1, a_2) = \begin{cases} (1, 1, 0) & \text{if} \quad k \le -3; \\ (1, 0, b_1) & \text{if} \quad k = -2; \\ (0, b_1, b_2) & \text{if} \quad k = -1; \end{cases}$$
   (d) sends (a_0, a_1, a_2, q) to Bob.
2. Bob
   (a) determines an ideal $\mathfrak{l} = (\tilde{Q}, \tilde{P})$ according to Table 1.
   (b) Sets $\mathfrak{k}' = \mathfrak{l}$ if $q \equiv \tilde{Q} \pmod 4$ and $\mathfrak{k}' = \rho(\mathfrak{l})$ otherwise.

The last column of Table 1 gives the ideal $\mathfrak{l}$ which Bob must determine in step 2(a) of Protocol 8.1. Entries of $*$ indicate any possible value for the parameter in that column.

Theorem 8.1. $\mathfrak{k}' = \mathfrak{k}$; *that is, the ideal* $\mathfrak{k}'$ *computed by Bob in step* 2(b) *of Protocol* 8.1 *is the same as the ideal* $\mathfrak{k}$ *computed by Alice in step* 3 *of Protocol* 7.1.

[2] As stated above, $\mathfrak{m} = \mathfrak{k}$ basically all the time. If Alice and Bob are not confident that $\mathfrak{m} = \mathfrak{k}$, Alice can send along the five extra bits specified in Protocol 8.1. This enables Bob to deduce $\mathfrak{k}$.

Table 1. Ideal $\mathfrak{l}$ for the final key agreement protocol.

a_0	h	a_1	a_2	$\mathfrak{l}$	
1	< -1	$*$	$*$	$\mathfrak{m}$	
1	$= -1$	1	$*$	$\rho^{-2}(\mathfrak{m})$	
1	$= -1$	0	0	$\rho^{-2}(\mathfrak{m})$	
1	$= -1$	0	1	$\rho^{-2}(\mathfrak{m})$	if $e > 2^{p+3}/7$
1	$= -1$	0	1	$\rho^{-1}(\mathfrak{m})$	if $e \le 2^{p+3}/7$
0	< -2	0	1	$\rho(\mathfrak{m})$	
0	$= -2$	1	$*$	$\rho(\mathfrak{m})$	
0	$= -2$	0	$*$	$\mathfrak{m}$	if $e > 7 \cdot 2^{p-2}$
0	$= -2$	0	$*$	$\rho(\mathfrak{m})$	if $e \le 7 \cdot 2^{p-2}$
0	$= -1$	1	$*$	$\rho^{-1}(\mathfrak{m})$	if $e > 21 \cdot 2^{p-4}$
0	$= -1$	1	$*$	$\mathfrak{m}$	if $e \le 21 \cdot 2^{p-4}$
0	$= -1$	0	$*$	$\rho^{-2}(\mathfrak{m})$	if $e > 3 \cdot 2^{p+2}/7$
0	$= -1$	0	0	$\rho^{-1}(\mathfrak{m})$	if $e \le 3 \cdot 2^{p+2}/7$
0	$= -1$	0	1	$\rho^{-1}(\mathfrak{m})$	if $21 \cdot 2^{p-4} < e \le 3 \cdot 2^{p+2}/7$
0	$= -1$	0	1	$\mathfrak{m}$	if $e \le 21 \cdot 2^{p-4}$

Proof. By Theorem 7.1, we have $\mathfrak{k} = \rho^i(\mathfrak{m})$ for some integer i with $|i| \le 2$. As in Lemma 4.3, set $a = 2^{k-h}d/e$.

If $a < \frac{7}{8}$, then, by part 2 of Lemma 4.3, $\mathfrak{k} \in \{\rho^{-2}(\mathfrak{m}), \rho^{-1}(\mathfrak{m})\}$. Suppose first that $(a_0, a_1, a_2) = (1, 1, 0)$. Then Bob knows that $k \le -3$. If $h < -1$, then $h - k', k - h' \le -2$, so, by part 1 of Lemma 4.3, $\mathfrak{l} = \mathfrak{m} = \mathfrak{k}$. If $h = -1$, then since $d \le 2^{p+1}$ and $e > 2^p$, we have $a \le d/4e < \frac{1}{2} < \frac{7}{8}$, so $\mathfrak{k} \in \{\rho^{-2}(\mathfrak{m}), \rho^{-1}(\mathfrak{m})\} = \{\mathfrak{l}, \rho(\mathfrak{l})\}$.

Suppose now that $(a_0, a_1, a_2) = (1, 0, b_1)$, so $k = -2$. Since $k - h' \le -2$, $\mathfrak{k} \in \{\rho^{-2}(\mathfrak{m}), \rho^{-1}(\mathfrak{m}), \mathfrak{m}\}$ by part 1 of Lemma 4.3. If $h < -1$, then $\mathfrak{l} = \mathfrak{m} = \mathfrak{k}$ as before. If $h = -1$, then we consider two cases. If $b_1 = 0$, then $d \le 3 \cdot 2^{p-1}$, so with $e > 2^p$, $a = d/2e < 3 \cdot 2^{p-1}/2^{p+1} = \frac{3}{4} < \frac{7}{8}$ and again $\mathfrak{k} \in \{\mathfrak{l}, \rho(\mathfrak{l})\}$. Suppose $b_1 = 1$, so $d > 3 \cdot 2^{p-1}$. If $e > 2^{p+3}/7$, then $a < 7 \cdot 2^{p+1}/2^{p+4} = \frac{7}{8}$, so $\mathfrak{k} \in \{\mathfrak{l}, \rho(\mathfrak{l})\}$, otherwise $a > 7 \cdot 3 \cdot 2^{p-1}/2^{p+4} = \frac{21}{32} > \frac{7}{12}$, in which case $\mathfrak{k} \in \{\rho^{-1}(\mathfrak{m}), \mathfrak{m}\}$ by part 2 of Lemma 4.3, so again $\mathfrak{k} \in \{\mathfrak{l}, \rho(\mathfrak{l})\}$.

Continuing in a similar fashion, one can show that $\mathfrak{k} \in \{\mathfrak{l}, \rho(\mathfrak{l})\}$ for every possible set of values of (a_0, a_1, a_2). By Lemma 4.5 of [7], if $\mathfrak{l} = (\tilde{Q}, \tilde{P})$ and $\rho(\mathfrak{l}) = (\hat{Q}, \hat{P})$, then $\tilde{Q} \not\equiv \hat{Q} \pmod 4$, provided $D \equiv 3 \pmod 4$. Hence the quantity q enables Bob to distinguish $\mathfrak{l}$ from $\rho(\mathfrak{l})$ correctly, and it follows that $\mathfrak{k}' = \mathfrak{k}$. □

9. Security

Aspects of the security of the general idea underlying the real quadratic fields key exchange protocol have been discussed in some detail in [12] and [7]. However, the specifics of this particular implementation are somewhat different from those in [12] and even in [7]; thus, we provide some additional remarks concerning this matter here.

We note that for the fixed values p and r given in Section 7, a pair of unknown integers a, b, and a given ideal $\mathfrak{r}$, the objects roughly corresponding to a Diffie–Hellman triple here are

$$(\mathfrak{a}, \lceil 2^{-r} d_a \rceil, k_a), \quad (\mathfrak{b}, \lceil 2^{-r} d_b \rceil, k_b), \quad \text{and} \quad \mathfrak{k},$$

where

$$\begin{aligned} (\mathfrak{a}, d_a, k_a) &= \mathrm{EXP}((\mathfrak{r}, 2^{p+1}, -1), a, p), \\ (\mathfrak{b}, d_b, k_b) &= \mathrm{EXP}((\mathfrak{r}, 2^{p+1}, -1), b, p), \\ (\mathfrak{k}, d, k) &= \mathrm{EXP}((\mathfrak{a}, 2^r \lceil 2^{-r} d_a \rceil, k_a), b, p), \end{aligned}$$

because usually $\mathfrak{k} = \mathfrak{m}$. Certainly, an attacker can break this system if he can deduce a or b from the transmitted information.

The discrete logarithm problem in this context is the problem of finding a generator α (or a good approximation of $\log|\alpha|$) of a reduced principal ideal (see [12]). Since $(\mathfrak{a}, d_a, k_a)$ is a near reduced (f, p) representation of $\mathfrak{r}^a$, we have

$$\mathfrak{a} = (\theta_a)\mathfrak{r}^a \tag{9.1}$$

for some $\theta_a \in \mathbb{K}$. Suppose an attacker can find a generator α of $\mathfrak{a}$, or a good approximation of $\log|\alpha|$. Then he could solve (9.1) for a as follows. It is easy to deduce $\gamma \in \mathbb{K}$ with $\mathfrak{r} = (\gamma)$. Then, by (9.1), $|\alpha| = \varepsilon^m \theta_a \gamma^a$ where $m \in \mathbb{Z}$ and ε is the fundamental unit of $\mathbb{K}$, so

$$\log \alpha = mR + \log \theta_a + a \log \gamma,$$

where $R = \log \varepsilon$ is the regulator of $\mathbb{K}$. Now $\log \gamma$ is small by construction of $\mathfrak{r}$, $\log \theta_a$ is small by Corollary 4.1, and m is small by our choice of the upper bound B on a (see [12]). So it would not be hard to find a, once R is known. Of course the determination of R is just another instance of our discrete logarithm problem.

However, it is possible to view (9.1) from another point of view. The attacker knows $\lceil 2^{-r} d_a \rceil$ and k_a. Note that knowledge of k_a provides him with little information, since k_a is usually small by Corollary 4.1—as mentioned right after the proof of that corollary, we expect $k_a = -1$ in 64% of all cases, for example—so many possible values for θ_a could have the same k_a value. However, the adversary also knows that $|2^{p-k_a}\theta_a/d_a - 1| < 2^{-p} f$ for some $f < 14.33B$, so he can use $\hat{\psi} = 2^{p-r-k_a}/\lceil 2^{-r} d_a \rceil$ as an approximation to $\psi = \theta_a^{-1}$. If we set $\delta = |\hat{\psi}/\psi - 1|$, then it can be shown that $\delta < (1 + 2^{1-r}) \cdot 2^{-p} f < (1 + 2^{-15}) \cdot 2^{-p} f$ when $B \geq 2^{16}$, so $\delta < (1 + 2^{-15}) \cdot 14.33/(50B \log_2 B)$. We may therefore assume that the attacker knows a good rational approximation $\hat{\psi}$ to ψ as well as an upper bound $\delta < 0.29/(B \log_2 B)$ on the relative error of that approximation.

As usual, for two functions $f(n)$, $g(n)$ $(n \in \mathbb{N})$, we write $f(n) = \Theta(g(n))$ if $c\, g(n) \leq f(n) \leq d\, g(n)$ for positive constants c, d, and n sufficiently large. Let m/n be any convergent of the continued fraction expansion of $\sqrt{\Delta}$. Then for a given n, it is possible to formulate explicitly an infinitude of distinct pairs of rationals $(r(n), s(n))$ such that $r(n), s(n)\sqrt{\Delta} = \Theta(n)$, and if $\psi(n) = r(n) - s(n)\sqrt{\Delta}$, then $|\hat{\psi} - \psi(n)| = O(n^{-1})$, so knowing $\hat{\psi}$ and δ still leaves infinitely many possibilities for ψ.

On the other hand, if the coefficients of ψ were known to lie in a certain range that is not too large, then it might be possible to search for them successfully. If $\psi = r - s\sqrt{\Delta}$,

then since $r = (\psi + \overline{\psi})/2$ and $s\sqrt{\Delta} = (\psi - \overline{\psi})/2$, an interval containing r and $s\sqrt{\Delta}$ can be determined from bounds on ψ and $|\overline{\psi}|$. By (9.1), we have

$$N(\mathfrak{r})^a = \psi|\overline{\psi}|N(\mathfrak{a}). \tag{9.2}$$

Now $N(\mathfrak{r})$ is public information, $1 \leq N(\mathfrak{a}) < \sqrt{\Delta}$ since $\mathfrak{a}$ is reduced, and we have the bounds $\frac{16}{17} < \psi < 16\sqrt{\Delta}/15$ on ψ from Corollary 4.1. It follows from (9.2) that $|\overline{\psi}| = \Theta(N(\mathfrak{r})^a)$ as a grows, so r and $s\sqrt{\Delta}$ are exponentially large in a. Since a tends to be quite large (see Section 10), this does not allow for an efficient search for a.

10. Implementation

When implementing Algorithm NUCOMP, there are a number of optimizations one can utilize. First, steps 1–3 can be simplified by making use of the observation that $\gcd(Q'/\sigma, Q''/\sigma) = 1$ most of the time. If this condition is detected, the second step, including the relatively expensive extended GCD computation, need not be executed and step 3 can be simplified.

The ideal reduction steps (steps 8 and 9 of NUCOMP and step 5 of NEAR) can be replaced by the more efficient Tenner's algorithm [21]. In order to use Tenner's algorithm, we need the R values corresponding to each ideal (Q, P), where $R = (D - P^2)/Q$. These can be computed expeditiously in step 7 of NUCOMP using

$$R_i = (-1)^{i-2}\left(a_{i-2}d_{i-2} - b_{i-2}c_{i-2}\right).$$

The R value corresponding to a given ideal computed using Tenner's algorithm is simply the Q value of the previous ideal.

As in [8], the computations in step 7 of NUCOMP can be rearranged and simplified in order to reduce the number of integer multiplications. Using (3.1) and (3.3), it is easy to prove $b_{i-1}c_{i-2} - b_{i-2}c_{i-1} = (-1)^{i-1}(P' - P'')$ and $a_{i-1}d_{i-2} - a_{i-2}d_{i-1} = (-1)^{i-2}(P' + P'')$, implying

$$c_{i-2} = \frac{b_{i-2}c_{i-1} + (-1)^{i-1}(P' - P'')}{b_{i-1}}, \qquad d_{i-1} = \frac{(P' + P'') - A_{i-2}d_{i-2}}{A_{i-3}},$$

from which we can derive the following optimized version of step 7:

— Compute a_{i-1}, a_{i-2}, c_{i-1} as before.
— $X_1 = b_{i-2}c_{i-1}$, $c_{i-2} = \left(X_1 + (-1)^{i-1}(P' - P'')\right)/b_{i-1}$.
— Compute d_{i-2} as before.
— $X_2 = A_{i-2}d_{i-2}$, $d_{i-1} = \left((P' + P'') - X_2\right)/A_{i-3}$.
— Compute Q_i as before,
  $P_i = P'' + (-1)^{i-1}(X_2 - X_1)$, $R_i = |a_{i-2}d_{i-2} - b_{i-2}c_{i-2}|$.
— Compute G', G, B', B as before.

We can also improve these formulas in the frequently occurring case of squaring an ideal. The resulting algorithm was called NUDUPL by Shanks. The modifications to steps 1–3 and step 7 are straightforward to derive using the facts that $Q' = Q''$, $P' = P''$, and $b_j = \sigma c_j$.

Another improvement to our new protocol over that of [7] is that we use significantly smaller s values. In [7] we had to choose s such that $2^s > D^{3/4}$, whereas here we can choose s dynamically. In practice, rather than computing s as in step 11 of NUCOMP or step 1 of NEAR, we precompute a table of valid s values which can be indexed by the size of Q'_j/S in step 11 of NUCOMP and by the size of Q in step 1 of NEAR.

Hamdy [4] reports that replacing the extended GCD computations of NUCOMP with Lehmer's variant yields significant performance increases when applied to key exchange in imaginary quadratic fields. We have used Lehmer's variant for the extended GCD computations in both NUCOMP (step 6) and NEAR (step 2). This resulted in a savings of about 25% over the classical extended Euclidean algorithm.

The key exchange protocol described in Section 7, including the optimizations described above, was implemented using the GNU C++ compiler version 3.2 and the C++ computer algebra library NTL [16]. For comparison purposes, we also implemented the protocol from [7] using the same software. The following computations were performed on a Pentium IV 2.53 GHz computer running Linux.

In order to test the efficiency of our improved protocol, we computed numerous examples using discriminants of 795, 1384, 1732, 3460, and 5704 bits. According to Hamdy's estimates [4] on solving the discrete logarithm problem in imaginary quadratic fields, discriminants of these sizes offer 80, 112, 128, 192, and 256 bits of security for cryptographic protocols based on that problem. NIST [10] currently recommends these five levels of security for key establishment in U.S. government applications. These discriminant sizes offer the same level of security or better for real quadratic fields, as the best-known algorithm for solving the infrastructure discrete logarithm problem has the same asymptotic complexity as that in imaginary quadratic fields [20], and may indeed be slightly harder in practice [6].

For each discriminant size, we randomly selected 1000 discriminants and executed both our new protocol and that of [7] once for each discriminant. We used $\rho^5((1))$ for the initial ideal $\mathfrak{r}$ and $B = 2^{160}, 2^{224}, 2^{256}, 2^{384}$, and 2^{512} for the exponent bounds, respectively. The exponent bounds were chosen so that the difficulty of solving the infrastructure discrete logarithm problem using generic algorithms running in time $O(\sqrt{B})$ is approximately the same as that using a subexponential index-calculus algorithm such as [6] or [20].

Table 2 contains the average CPU time per communication partner for a single application of the protocol. The protocol from [7] is indicated by MULT and our new protocol by NUCOMP. We also give the ratio of the average time using standard ideal multipli-

Table 2. Average CPU times (in seconds) per key exchange per partner.

$\log_2 \Delta$	IMAG	MULT	NUCOMP	MULT/NUCOMP
795	0.04	0.38	0.13	2.8948
1384	0.11	1.05	0.30	3.4518
1732	0.15	1.63	0.43	3.7925
3460	0.50	6.34	1.45	4.3814
5704	1.32	17.97	3.86	4.6553

cation and reduction (i.e. the protocol from [7]) over the average time using NUCOMP. For comparison, the first column contains the corresponding runtimes for an imaginary quadratic field-based key exchange protocol using the same NUCOMP implementation (without distance computation).

Our new protocol using NUCOMP was significantly more efficient than that of [7] in all cases, from just under 3 times as fast for 795-bit discriminants to over 4.5 times as fast for 5704-bit discriminants. As observed in [8], as the discriminant size increases, the savings obtained by using NUCOMP become more dramatic. The data suggest that the improvement factor might be of order $\log \log D$—a more precise complexity analysis and investigation are currently under investigation.

Although our algorithm does not yet rival key exchange in imaginary quadratic fields in terms of performance, our new implementation using NUCOMP is clearly much more competitive than that of [7]. In addition, A. Stein has observed that one can improve the efficiency of key exchange in real quadratic function fields to the point that it is faster than the corresponding protocols in imaginary quadratic function fields, by essentially taking advantage of the fact that reduction steps in the infrastructure are much less expensive than ideal multiplications. Whether these ideas will bear fruit in the real quadratic field case is the topic of further research.

As in [7], both partners end up with the same ideal in all cases. Thus, the second round of the protocol is never executed in practice.

References

[1] I. Biehl, J. A. Buchmann, and C. Thiel. Cryptographic protocols based on discrete logarithms in real-quadratic fields. In *Advances in Cryptology – CRYPTO* '94, pp. 56–60. Lecture Notes in Computer Science, vol. 839. Springer, Berlin, 1994.

[2] J. A. Buchmann and H. C. Williams. A key-exchange system based on real quadratic fields (extended abstract). In *Advances in Cryptology—CRYPTO* '89, pp. 335–343. Lecture Notes in Computer Science, vol. 435. Springer, Berlin, 1990.

[3] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Trans. Inform. Theory* **22** (1976), 472–492.

[4] S. Hamdy, Über die Sicherheit und Effizienz kryptographischer Verfahren mit Klassengruppen imaginär-quadratischer Zahlkörper. Doctoral Dissertation, Technische Universität Darmstadt, Darmstadt, 2002.

[5] D. Hühnlein and S. Paulus. On the implementation of cryptosystems based on real quadratic number fields (extended abstract). In *Selected Areas in Cryptography*, pp. 288–302. Lecture Notes in Computer Science, vol. 2012. Springer, Berlin, 2001.

[6] M. J. Jacobson, Jr. Computing discrete logarithms in quadratic orders. *J. Cryptology* **13** (2000), 473–492.

[7] M. J. Jacobson, Jr., R. Scheidler, and H. C. Williams. The efficiency and security of a real quadratic field based key exchange protocol. In *Public-Key Cryptography and Computational Number Theory*, pp. 91–112. de Gruyter, New York, 2001.

[8] M. J. Jacobson., Jr., and A. J. van der Poorten. Computational aspects of NUCOMP. *Proceedings of ANTS-V*, pp. 120–133. Lecture Notes in Computer Sciences, vol. 2369. Springer, Berlin, 2002.

[9] A. Menezes, P. van Oorschot, and S. Vanstone. *Handbook of Applied Cryptography*. Series on Discrete Mathematics and its Applications. CRC Press, Boca Raton, FL, 1996. ISBN 0-8493-8523-7.

[10] NIST. Recommendation on Key Establishment Schemes. Special Publication 800-56. Draft 2.0, January 2003. See: `http://csrc.nist.gov/CryptoToolkit/kms/keyschemes-Jan03.pdf`.

[11] R. Scheidler. Cryptography in quadratic function fields. *Des. Codes Cryptogr.* **22** (2001), 239–264.

[12] R. Scheidler, J. A. Buchmann, and H. C. Williams. Implementation of a key-exchange protocol using real quadratic fields. *J. Cryptology* **7** (1994), 171–199.

[13] R. Scheidler, A. Stein, and H. C. Williams. Key-exchange in real quadratic congruence function fields. *Des. Codes Cryptogr.* **7** (1996), 153–174.
[14] R. Schoof. Quadratic fields and factorization. In *Computational Methods in Number Theory*, *Part II*, pp. 235–286. Math. Centre Tracts 155. Math. Centrum, Amsterdam, 1982.
[15] D. Shanks. On Gauss and composition I, II. In *Proc. NATO ASI on Number Theory and Applications*, pp. 163–204. Kluwer, Dordrecht, 1989.
[16] V. Shoup. NTL: A library for doing number theory. Software, 2001. See `http://www.shoup.net.ntl`.
[17] A. J. Stephens and H. C. Williams. Some computational results on a problem concerning powerful numbers. *Math. Comp.* **50** (1988), 619–632.
[18] H. J. J. te Riele, A. J. van der Poorten, and H. C. Williams. Computer verification of the Ankeny–Artin–Chowla conjecture for all primes less than 100,000,000,000. *Math. Comp.* **70** (2001), 1311–1328.
[19] A. van der Poorten. A note on NUCOMP. *Math. Comp.* **72** (2003), 1935–1946.
[20] U. Vollmer. Rigorously Analyzed Algorithms for the Discrete Logarithm Problem in Quadratic Number Fields. Doctoral Dissertation, Technische Universität Darmstadt, Darmstadt, 2003.
[21] H. C. Williams and M. C. Wunderlich. On the parallel generation of the residues for the continued fraction factoring algorithm. *Math. Comp.* **48** (1987), 405–423.