Alcom-FT Technical Report Series
ALCOMFT-TR-03-39

Efficient algorithms for gcd and cubic residuosity in the ring of Eisenstein integers *

Ivan Bjerre Damgård   Gudmund Skovbjerg Frandsen

BRICS†
Department of Computer Science
University of Aarhus
Ny Munkegade
DK-8000 Aarhus C, Denmark

{ivan,gudmund}@brics.dk

version - May 11, 2003

Abstract

We present simple and efficient algorithms for computing gcd and cubic residuosity in the ring of Eisenstein integers, $\mathbf{Z}[\zeta]$, i.e. the integers extended with ζ, a complex primitive third root of unity. The algorithms are similar and may be seen as generalisations of the binary integer gcd and derived Jacobi symbol algorithms. Our algorithms take time $O(n^2)$ for n bit input. This is an improvement from the known results based on the Euclidean algorithm, and taking time $O(n \cdot M(n))$, where $M(n)$ denotes the complexity of multiplying n bit integers. The new algorithms have applications in practical primality tests and the implementation of cryptographic protocols.

1 Introduction

The Eisenstein integers, $\mathbf{Z}[\zeta] = \{a + b\zeta \mid a, b \in \mathbf{Z}\}$, is the ring of integers extended with a complex primitive third root of unity, i.e. ζ is root of x^2+x+1. Since the ring $\mathbf{Z}[\zeta]$ is a unique factorisation domain, a greatest common divisor (gcd) of two numbers is well-defined (up to multiplication by a unit). The gcd of two numbers may be found using the classic Euclidean algorithm, since $\mathbf{Z}[\zeta]$ is an Euclidean domain, i.e. there is a norm $N(\cdot) : \mathbf{Z}[\zeta] \setminus \{0\} \mapsto \mathbf{N}$ such that for $a, b \in \mathbf{Z}[\zeta] \setminus \{0\}$ there is $q, r \in \mathbf{Z}[\zeta]$ such that $a = qb + r$ with $r = 0$ or $N(r) < N(b)$.

When a gcd algorithm is directly based on the Euclidean property, it requires a subroutine for division with remainder. For integers there is a very efficient alternative in the form of the binary gcd, that only requires addition/subtraction and division by two [12]. A corresponding Jacobi symbol algorithm has been analysed as well [11].

*Partially supported by the IST Programme of the EU under contract number IST-1999-14186 (ALCOM-FT).

†Basic Research in Computer Science,
Centre of the Danish National Research Foundation.

It turns out that there are natural generalisations of these binary algorithms over the integers to algorithms over the Eisenstein integers for computing the gcd and the cubic residuosity symbol. The role of 2 is taken by the number $1 - \zeta$, which is a prime of norm 3 in $\mathbf{Z}[\zeta]$.

We present and analyse these new algorithms. It turns out that they both have bit complexity $O(n^2)$, which is an improvement over the so far best known algorithms by Scheidler and Williams [8], Williams [16], Williams and Holte [17]. Their algorithms have complexity $O(nM(n))$, where $M(n)$ is the complexity of integer multiplication and the best upper bound on $M(n)$ is $O(n \log n \log\log n)$ [10].

1.1 Related work

The asymptotically fastest algorithm for integer gcd takes time $O(n \log n \log\log n)$ and is due to Schönhage [9]. There is a derived algorithm for the Jacobi symbol of complexity $O(n(\log n)^2 \log\log n)$. For practical input sizes the most efficient algorithms seems to be variants of the binary gcd and derived Jacobi symbol algorithms [11, 7].

If ω_n is a complex primitive nth root of unity, say $\omega_n = e^{2\pi/n}$ then the ring $\mathbf{Z}[\omega_n]$ is known to be norm-Euclidean for only finitely many n and the smallest unresolved case is $n = 17$ [6, 4].

Weilert have generalised both the "binary" and the asymptotically fast gcd algorithms to $\mathbf{Z}[\omega_4] = \mathbf{Z}[i]$, the ring of Gaussian integers [13, 14]. In the latter case Weilert has also described a derived algorithm for computing the quartic residue symbol [15], and in all cases the complexity is identical to the complexity of the corresponding algorithm over $\mathbf{Z}$.

Williams [16], Williams and Holte [17] both describe algorithms for computing gcd and cubic residue symbols in $\mathbf{Z}[\omega_3]$, the Eisenstein integers. Scheidler and Williams describe algorithms for computing gcd and nth power residue symbol in $\mathbf{Z}[\omega_n]$ for $n = 3, 5, 7$ [8]. Their algorithms all have complexity $O(nM(n))$ for $M(n)$ being the complexity of integer multiplication.

Weilert suggests that his binary (i.e. $(1+i)$-ary) gcd algorithm for the Gaussian integers may generalise to other norm-Euclidean rings of algebraic integers [13]. Our gcd algorithm for the Eisenstein integers was obtained independently, but it may nevertheless be seen as a confirmation of this suggestion in a specific case. It is an open problem whether the "binary" approach to gcd computation may be further generalised to $\mathbf{Z}[\omega_5]$.

Weilert gives an algorithm for the quartic residue symbol that is derived from the asymptotically fast gcd algorithm over $\mathbf{Z}[i]$. For practical purposes, however, it would be more interesting to have a version derived from the "binary" approach. In the last section of this paper, we sketch how one can obtain such an algorithm.

1.2 Applications

Our algorithms may be used for the efficient computation of cubic residuosity in other rings than $\mathbf{Z}[\zeta]$ when using an appropriate homomorphism. As an example, consider the finite field $GF(p)$ for prime $p \equiv 1 \bmod 3$. A number $z \in \{1, \ldots, p-1\}$ is a cubic residue precisely when $z^{(p-1)/3} \equiv 1 \bmod p$, implying that (non)residuosity may be decided by a (slow) modular exponentiation. However, it is possible to decide cubic residuosity much faster provided we make some preprocessing depending only on p. The preprocessing consists in factoring p over $\mathbf{Z}[\zeta]$, i.e. finding a prime $\pi \in \mathbf{Z}[\zeta]$ such that $p = \pi\bar{\pi}$. A suitable π may be found as $\pi = \gcd(p, r - \zeta)$, where $r \in \mathbf{Z}$ is constructed as a solution to the quadratic equation $x^2 + x + 1 = 0 \bmod p$. Following this preprocessing cubic residuosity of any z is decided using that $z^{(p-1)/3} \equiv 1 \bmod p$ if and only if $[z/\pi] = 1$, where $[\cdot/\cdot]$ denotes the cubic residuosity symbol.

When the order of the multiplicative group in question is unknown, modular exponentiation cannot be used, but it may still be possible to identify some nonresidues by computing residue symbols. In particular, the primality test of Damgård and Frandsen [2] uses our algorithms for finding cubic nonresidues in a more general ring.

Computation of gcd and cubic residuosity is also used for the implementation of cryptosystems by Scheidler and Williams [8], and by Williams [16].

2 Preliminary facts about $\mathbf{Z}[\zeta]$

$\mathbf{Z}[\zeta]$ is the ring of integers extended with a primitive third root of unity ζ (complex root of $z^2 + z + 1$). We will be using the following definitions and facts (see f.x. [3]).

Define the two conjugate mappings $\sigma_i : \mathbf{Z}[\zeta] \mapsto \mathbf{Z}[\zeta]$ by $\sigma_i(\zeta) = \zeta^i$ for $i = 1, 2$. The rational integer $N(\alpha) = \sigma_1(\alpha)\sigma_2(\alpha) \geq 0$ is called the norm of $\alpha \in \mathbf{Z}[\zeta]$, and $N(a + b\zeta) = a^2 + b^2 - ab$. (Note that $\sigma_2(\cdot)$ and $N(\cdot)$ coincides with complex conjugation and complex norm, respectively).

A unit in $\mathbf{Z}[\zeta]$ is an element of norm 1. There are 6 units in $\mathbf{Z}[\zeta]$: $\pm 1, \pm\zeta, \pm\zeta^2$. Two elements $\alpha, \beta \in \mathbf{Z}[\zeta]$ are said to be associates if there exists a unit ϵ such that $\alpha = \epsilon\beta$.

A prime π in $\mathbf{Z}[\zeta]$ is a non-unit such that for any $\alpha, \beta \in \mathbf{Z}[\zeta]$, if $\pi|\alpha\beta$, then $\pi|\alpha$ or $\pi|\beta$.

$1 - \zeta$ is a prime in $\mathbf{Z}[\zeta]$ and $N(1 - \zeta) = 3$. A primary number has the form $1 + 3\beta$ for some $\beta \in \mathbf{Z}[\zeta]$. If $\alpha \in \mathbf{Z}[\zeta]$ is not divisible by $1 - \zeta$ then α is associated to a primary number. (The definition of *primary* seems to vary in that some authors require the alternate forms $\pm 1 + 3\beta$ [5] and $-1 + 3\beta$ [3], but our definition is more convenient in the present context).

A simple computation reveals that the norm of a primary number has residue 1 modulo 3, and since the norm is a multiplicative homomorpism it follows that every $\alpha \in \mathbf{Z}[\zeta]$ that is not divisible by $1 - \zeta$ has $N(\alpha) \equiv 1 (\bmod 3)$.

3 Computing gcd in $\mathbf{Z}[\zeta]$

It turns out that the well-known binary integer gcd algorithm has a natural generalisation to a gcd algorithm for the Eisenstein integers. The generalised algorithm is best understood by relating it to the binary algorithm in a nonstandard version. The authors are not aware of any description of the latter in the literature (for the standard version see f.x. [1]).

A slightly nonstandard version of the binary gcd is the following. Every integer can be represented as $(-1)^i \cdot 2^j \cdot (4m + 1)$, where $i \in \{0, 1\}$, $j \geq 0$ and $m \in \mathbf{Z}$. Without loss of generality, we may therefore assume that the numbers in question are of the form (4m+1). One iteration consists in replacing the numerically larger of the two numbers by their difference. If it is nonzero then the dividing 2-power (at least 2^2) may be removed without changing the gcd. If necessary the resulting odd number is multiplied with -1 to get a number of the form $4m + 1$ and we are ready for the next iteration. It is fairly obvious that the product of the numeric values of the two numbers decreases by a factor at least 2 in each step until the gcd is found, and hence the gcd of two numbers a, b can be computed in time $(\log^2 |ab|)$.

To make the analogue, we recall that any element of $\mathbf{Z}[\zeta]$ that is not divisible by $1-\zeta$ is associated to a (unique) primary number, i.e. a number of the form $1+3\alpha$. This implies that any element in $\mathbf{Z}[\zeta] \setminus \{0\}$ has a (unique) representation on the form $(-\zeta)^i \cdot (1-\zeta)^j \cdot (1+3\alpha)$ where $0 \leq i < 6$, $0 \leq j$ and $\alpha \in \mathbf{Z}[\zeta]$. In addition, the difference of two primary numbers is divisible by $(1-\zeta)^2$, since $3 = -\zeta^2(1-\zeta)^2$. Now a gcd algorithm for the Eisenstein integers may be formulated as an analogue to the binary integer gcd algorithm. We may assume without loss of generality that the two input numbers are primary. Replace the (normwise) larger of the two numbers with their difference. If it is nonzero, we may divide out any powers of $(1-\zeta)$ that divide the difference (at least $(1-\zeta)^2$) and convert the remaining factor to primary form by multiplying with a unit. We have again two primary numbers and the process may be continued. In each step we are required to identify the (normwise) larger of two numbers. Unfortunately it would be too costly to compute the relevant norm, but it suffices to choose the large number based on an approximation that we can afford to compute. By a slightly nontrivial argument one may prove that the product of the norms of the two numbers decreases by a factor at least 2 in each step until the gcd is found, and hence the gcd of two numbers α, β can be computed in time $O(\log^2 N(\alpha\beta))$.

Algorithm 1 describes the details including a start-up to bring the two numbers on primary form.

Algorithm 1 Compute gcd in $\mathbf{Z}[\zeta]$

Require: $\alpha, \beta \in \mathbf{Z}[\zeta] \setminus \{0\}$
Ensure: $g = \gcd(\alpha, \beta)$

Let primary $\gamma, \delta \in \mathbf{Z}[\zeta]$ be defined by $\alpha = (-\zeta)^{i_1} \cdot (1-\zeta)^{j_1} \cdot \gamma$ and $\beta = (-\zeta)^{i_2} \cdot (1-\zeta)^{j_2} \cdot \delta$.
$g \leftarrow (1-\zeta)^{\min\{j_1, j_2\}}$
Replace α, β with γ, δ.
while $\alpha \neq \beta$ **do**
  LOOP INVARIANT: α, β are primary
  Let primary γ be defined by $\alpha - \beta = (-\zeta)^i \cdot (1-\zeta)^j \cdot \gamma$
  Replace "approximately" larger of α, β with γ.
end while
$g \leftarrow g \cdot \alpha$

Theorem 1 *Algorithm 1 takes time $O(\log^2 N(\alpha\beta))$ to compute the gcd of α, β, or formulated alternatively, the algorithm has bit complexity $O(n^2)$.*

Proof. Let us assume that a number $\alpha = a + b\zeta \in \mathbf{Z}[\zeta]$ is represented by the integer pair (a, b). Observe that since $N(\alpha) = a^2 + b^2 - ab$, we have that $\log|a| + \log|b| \leq \log N(\alpha) \leq 2(\log|a| + \log|b|)$ for $a, b \neq 0$, i.e. the logarithm of the norm is proportional to the number of bits in the representation of a number.

We may do addition, subtraction on general numbers and multiplication by units in linear time. Since $(1-\zeta)^{-1} = (2+\zeta)/3$, division by (and check for divisibility by) $(1-\zeta)$ may also be done in linear time.

Clearly, the startup part of the algorithm that brings the two numbers on primary form can be done in time $O(\log^2 N(\alpha\beta))$. Hence, we need only worry about the while loop.

We want to prove that the norm of the numbers decrease for each iteration. The challenge is to see that forming the number $\alpha - \beta$ does not increase the norm too much. In fact

$N(\alpha-\beta) \le 4 \cdot \max\{N(\alpha), N(\beta)\}$. This follows trivially from the fact that the norm is non-negative combined with the equation $N(\alpha+\beta) + N(\alpha-\beta) = 2(N(\alpha)+N(\beta))$ that may be proven by an elementary computation. Hence, for the γ computed in the loop of the algorithm, we get $N(\gamma) = 3^{-j}N(\alpha-\beta) \le 3^{-2}4 \cdot \max\{N(\alpha), N(\beta)\}$. In each iteration, γ ideally replaces the one of α and β with the larger norm. However, we can not afford to actually compute the norms to find out which one is the larger. Fortunately, by Lemma 2, it is possible in linear time to compute an approximate norm that may be slightly smaller than the exact norm, namely up to a factor 9/8. When γ replaces the one of α and β with the larger approximate norm, we know that $N(\alpha\beta)$ decreases by a factor at least $9/4 \cdot 8/9 = 2$ in each iteration, i.e. the total number of iterations is $O(\log N(\alpha\beta))$.

Each loop iteration takes time $O(\log N(\alpha\beta))$ except possibly for finding the exponent of $(1-\zeta)$ that divides $\alpha - \beta$. Assume that $(1-\zeta)^{t_i}$ is the maximal power of $(1-\zeta)$ that divides $\alpha-\beta$ in the ith iteration. Then the combined time complexity of all loop iterations is $O((\sum_i t_i)\log N(\alpha\beta))$. We also know that the norm decreases by a factor at least $3^{t_i-2}\cdot 2$ in the ith iteration, i.e. $\prod_i(3^{t_i-2}\cdot 2) \le N(\alpha\beta)$. Since there is only $O(\log N(\alpha\beta))$ iterations it follows that $\prod_i 3^{t_i} \le (9/2)^{O(\log N(\alpha\beta))}N(\alpha\beta)$ and hence $\sum_i t_i = O(\log N(\alpha\beta))$. ∎

Lemma 2 *Given $\alpha = a + b\zeta \in \mathbf{Z}[\zeta]$ it is possible to compute an approximate norm $\tilde{N}(\alpha)$ such that*

$$\frac{8}{9}N(\alpha) \le \tilde{N}(\alpha) \le N(\alpha)$$

in linear time, i.e. in time $O(\log N(\alpha))$.

Proof. Note that

$$N(a+b\zeta) = \frac{(a-b)^2 + a^2 + b^2}{2}.$$

Given $\epsilon > 0$, we let $\tilde{d}$ denote some approximation to integer d satisfying that $(1-\epsilon)|d| \le \tilde{d} \le |d|$. Note that

$$(1-\epsilon)^2 N(a+b\zeta) \le \frac{\widetilde{(a-b)}^2 + \tilde{a}^2 + \tilde{b}^2}{2} \le N(a+b\zeta)$$

Since we may compute $a-b$ in linear time it suffices to compute $\tilde{\ }$-approximations and square them in linear time for some $\epsilon < 1/18$. Given d in the usual binary representation, we take $\tilde{d}$ to be $|d|$ with all but the 6 most significant bits replaced with zeroes, in which case

$$(1-\frac{1}{32})|d| \le \tilde{d} \le |d|$$

and we can compute $\tilde{d}^2$ from d in linear time. ∎

4 Computing cubic residuosity in $\mathbf{Z}[\zeta]$

Just as the usual integer gcd algorithms may be used for constructing algorithms for the Jacobi symbol, so can our earlier strategy for computing the gcd in $\mathbf{Z}[\zeta]$ be used as the basis for an algorithm for computing the cubic residuosity symbol.

We start by recalling the definition of the cubic residuosity symbol.

$$[\cdot/\cdot] : \mathbf{Z}[\zeta] \times (\mathbf{Z}[\zeta] - (1-\zeta)\mathbf{Z}[\zeta]) \mapsto \{0, 1, \zeta, \zeta^{-1}\}$$

is defined as follows:

- For prime $\pi \in \mathbf{Z}[\zeta]$ where π is not associated to $1-\zeta$:

$$[\alpha/\pi] = (\alpha^{\frac{N(\pi)-1}{3}}) \bmod \pi$$

- For number $\beta = \prod_{i=1}^{t} \pi_i^{m_i} \in \mathbf{Z}[\zeta]$ where β is not divisible by $1-\zeta$:

$$[\alpha/\beta] = \prod_{i=1}^{t} [\alpha/\pi_i]^{m_i}$$

Note that these rules imply $[\alpha/\epsilon] = 1$ for a unit ϵ and $[\alpha/\beta] = 0$ when $\gcd(\alpha, \beta) \neq 1$. In addition, we will need the following laws satisfied by the cubic residuosity symbol (recall that β is primary when it has the form $\beta = 1 + 3\gamma$ for $\gamma \in \mathbf{Z}[\zeta]$) [5]:

- Modularity:
$$[\alpha/\beta] = [\alpha'/\beta], \quad \text{when } \alpha \equiv \alpha' (\bmod \beta).$$

- Multiplicativity:
$$[\alpha\alpha'/\beta] = [\alpha/\beta] \cdot [\alpha'/\beta].$$

- The cubic reciprocity law:
$$[\alpha/\beta] = [\beta/\alpha], \quad \text{when } \alpha \text{ and } \beta \text{ are both primary.}$$

- The complementary laws (for primary $\beta = 1 + 3(m + n\zeta)$, where $m, n \in \mathbf{Z}$)
$$\begin{aligned} [1-\zeta/\beta] &= \zeta^m, \\ [\zeta/\beta] &= \zeta^{-(m+n)}, \\ [-1/\beta] &= 1. \end{aligned}$$

The cubic residuosity algorithm will follow the gcd algorithm closely. In each iteration we will assume the two numbers α, β to be primary with $\tilde{N}(\alpha) \geq \tilde{N}(\beta)$. We write their difference on the form $\alpha - \beta = (-\zeta)^i (1-\zeta)^j \gamma$, for primary $\gamma = 1 + 3(m + n\zeta)$. By the above laws, $[\alpha/\beta] = \zeta^{mj-(m+n)i}[\gamma/\beta]$. If $\tilde{N}(\alpha) < \tilde{N}(\beta)$, we use the reciprocity law to swap γ and β before being ready to a new iteration. The algorithm stops, when the two primary numbers are identical. If the identical value (the gcd) is not 1 then the residuosity symbol evaluates to 0.

Algorithm 2 describes the entire procedure including a start-up to ensure that the numbers are primary.

Theorem 3 *Algorithm 2 takes time $O(\log^2 N(\alpha\beta))$ to compute $[\alpha/\beta]$, or formulated alternatively, the algorithm has bit complexity $O(n^2)$.*

Proof. The complexity analysis from the gcd algorithm carries over without essential changes. ■

Algorithm 2 Compute cubic residuosity in $\mathbf{Z}[\zeta]$

Require: $\alpha, \beta \in \mathbf{Z}[\zeta] \setminus \{0\}$, and β is not divisible by $(1-\zeta)$
Ensure: $c = [\alpha/\beta]$

Let primary $\gamma, \delta \in \mathbf{Z}[\zeta]$ be defined by $\alpha = (-\zeta)^{i_1} \cdot (1-\zeta)^{j_1} \cdot \gamma$ and $\beta = (-\zeta)^{i_2} \cdot \delta$.
Let $m, n \in \mathbf{Z}$ be defined by $\delta = 1 + 3m + 3n\zeta$.
$t \leftarrow mj_1 - (m+n)i_1 \bmod 3$
Replace α, β by γ, δ.
If $\tilde{N}(\alpha) < \tilde{N}(\beta)$ then interchange α, β.
while $\alpha \neq \beta$ **do**
  LOOP INVARIANT: α, β are primary and $\tilde{N}(\alpha) \geq \tilde{N}(\beta)$
  Let primary γ be defined by $\alpha - \beta = (-\zeta)^i \cdot (1-\zeta)^j \cdot \gamma$
  Let $m, n \in \mathbf{Z}$ be defined by $\beta = 1 + 3m + 3n\zeta$.
  $t \leftarrow t + mj - (m+n)i \bmod 3$
  Replace α with γ.
  If $\tilde{N}(\alpha) < \tilde{N}(\beta)$ then interchange α, β.
end while
If $\alpha \neq 1$ then $c \leftarrow 0$ else $c \leftarrow \zeta^t$

5 Computing gcd and quartic residuosity in the ring of Gaussian integers

We may construct fast algorithms for gcd and quartic residuosity in the ring of Gaussian integers, $\mathbf{Z}[i] = \{a + bi \mid a, b \in \mathbf{Z}\}$, in a completely analogous way to the algorithms over the Eisenstein integers. In the case of the gcd, this was essentially done by Weilert [13]. However, the case of the quartic residue symbol may be of independent interest since such an algorithm is likely to be more efficient for practical input values than the asymptically ultrafast algorithm [15].

Here is a sketch of the necessary facts (see [5]). There are 4 units in $\mathbf{Z}[i]$: $\pm 1, \pm i$. $1+i$ is a prime in $\mathbf{Z}[i]$ and $N(1+i) = 2$. A primary number has the form $1 + (2+2i)\beta$ for some $\beta \in \mathbf{Z}[i]$. If $\alpha \in \mathbf{Z}[i]$ is not divisible by $1+i$ then α is associated to a primary number.

In particular, any element in $\mathbf{Z}[i] \setminus \{0\}$ has a (unique) representation on the form $i^j \cdot (1+i)^k \cdot (1 + (2+2i)\alpha)$ where $0 \leq j < 4$, $0 \leq k$ and $\alpha \in \mathbf{Z}[i]$. In addition, the difference of two primary numbers is divisible by $(1+i)^3$, since $(2+2i) = -i(1+i)^3$. This is the basis for obtaining an algorithm for computing gcd over the Gaussian integers analogous to Algorithm 1. This new algorithm has also bit complexity $O(n^2)$ as one may prove when using that $N((1+i)^3) = 8$ and $N(\alpha - \beta) \leq 4 \cdot \max\{N(\alpha), N(\beta)\}$.

For computing quartic residuosity, we need more facts [5]. If π is a prime in $\mathbf{Z}[i]$ and π is not associated to $1+i$ then $N(\pi) \equiv 1(\bmod 4)$, and the quartic residue symbol $[\cdot/\cdot] : \mathbf{Z}[i] \times (\mathbf{Z}[i] - (1+i)\mathbf{Z}[i]) \mapsto \{0, 1, -1, i, -i\}$ is defined as follows:

- For prime $\pi \in \mathbf{Z}[i]$ where π is not associated to $1+i$:

$$[\alpha/\pi] = (\alpha^{\frac{N(\pi)-1}{4}}) \bmod \pi$$

- For number $\beta = \prod_{j=1}^{t} \pi_j^{m_j} \in \mathbf{Z}[i]$ where β is not divisible by $1+i$:

$$[\alpha/\beta] = \prod_{j=1}^{t} [\alpha/\pi_j]^{m_j}$$

The quartic residuosity symbol satisfies in addition

- Modularity:
$$[\alpha/\beta] = [\alpha'/\beta], \quad \text{when } \alpha \equiv \alpha' (\bmod \beta).$$

- Multiplicativity:
$$[\alpha\alpha'/\beta] = [\alpha/\beta] \cdot [\alpha'/\beta].$$

- The quartic reciprocity law:
$$[\alpha/\beta] = [\beta/\alpha] \cdot (-1)^{\frac{N(\alpha)-1}{4} \cdot \frac{N(\beta)-1}{4}}, \quad \text{when } \alpha \text{ and } \beta \text{ are both primary.}$$

- The complementary laws (for primary $\beta = 1 + (2+2i)(m+ni)$, where $m, n \in \mathbf{Z}$)
$$\begin{aligned} [1+i/\beta] &= i^{-n-(n+m)^2}, \\ [i/\beta] &= i^{n-m}. \end{aligned}$$

This is the basis for obtaining an algorithm for computing quartic residuosity analogous to Algorithm 2. This new algorithm has also bit complexity $O(n^2)$.

References

[1] Eric Bach and Jeffrey Shallit. *Algorithmic number theory. Vol. 1.* Foundations of Computing Series. MIT Press, Cambridge, MA, 1996. Efficient algorithms.

[2] Ivan B. Damgård and Gudmund Skovbjerg Frandsen. An extended quadratic Frobenius primality test with average and worst case error estimates. Research Series RS-03-9, BRICS, Department of Computer Science, University of Aarhus, February 2003. Extended abstract in these proceedings.

[3] Kenneth Ireland and Michael Rosen. *A classical introduction to modern number theory*, Vol. 84 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, second edition, 1990.

[4] Franz Lemmermeyer. The Euclidean algorithm in algebraic number fields. *Exposition. Math.* **13**(5) (1995), 385–416.

[5] Franz Lemmermeyer. *Reciprocity laws.* Springer Monographs in Mathematics. Springer-Verlag, Berlin, 2000. From Euler to Eisenstein.

[6] Hendrik W. Lenstra, Jr. Euclidean number fields. I. *Math. Intelligencer* **2**(1) (1979/80), 6–15.

[7] Shawna Meyer Eikenberry and Jonathan P. Sorenson. Efficient algorithms for computing the Jacobi symbol. *J. Symbolic Comput.* **26**(4) (1998), 509–523.

[8] Renate Scheidler and Hugh C. Williams. A public-key cryptosystem utilizing cyclotomic fields. *Des. Codes Cryptogr.* **6**(2) (1995), 117–131.

[9] A. Schönhage. Schnelle Berechnung von Kettenbruchentwicklungen. *Acta Informat.* **1** (1971), 139–144.

[10] A. Schönhage and V. Strassen. Schnelle Multiplikation grosser Zahlen. *Computing (Arch. Elektron. Rechnen)* **7** (1971), 281–292.

[11] Jeffrey Shallit and Jonathan Sorenson. A binary algorithm for the jacobi symbol. *ACM SIGSAM Bull.* **27**(1) (1993), 4–11.

[12] J. Stein. Computationals problems associated with Racah algebra. *J. Comput. Phys.* **1** (1967), 397–405.

[13] André Weilert. $(1+i)$-ary GCD computation in $\mathbf{Z}[i]$ is an analogue to the binary GCD algorithm. *J. Symbolic Comput.* **30**(5) (2000), 605–617.

[14] André Weilert. Asymptotically fast GCD computation in $\mathbf{Z}[\mathbf{i}]$. In *Algorithmic number theory (Leiden, 2000)*, Vol. 1838 of *Lecture Notes in Comput. Sci.*, pp. 595–613. Springer, Berlin, 2000.

[15] André Weilert. Fast computation of the biquadratic residue symbol. *J. Number Theory* **96**(1) (2002), 133–151.

[16] H. C. Williams. An M^3 public-key encryption scheme. In *Advances in cryptology—CRYPTO '85 (Santa Barbara, Calif., 1985)*, Vol. 218 of *Lecture Notes in Comput. Sci.*, pp. 358–368. Springer, Berlin, 1986.

[17] H. C. Williams and R. Holte. Computation of the solution of $x^3+Dy^3=1$. *Math. Comp.* **31**(139) (1977), 778–785.