ELLIPOTIC CURVES AND PRIMALITY PROVING

A. O. L. ATKIN AND F. MORAIN

Dedicated to the memory of D. H. Lehmer

ABSTRACT. The aim of this paper is to describe the theory and implementation of the Elliptic Curve Primality Proving algorithm.

Problema, numeros primos a compositis dignoscendi, hosque in factores suos primos resolvendi, ad gravissima ac utilissima totius arithmeticae pertinere, et geometrarum tum veterum tum recentiorum industiam ac sagacitatem occupavisse, tam notum est, ut de hac re copióse loqui superfluum foret.

C. F. Gauss [38, Art. 329]

1. INTRODUCTION

Primality testing is one of the most flourishing fields in computational number theory. Dating back to Gauss, the interest has recently risen with modern cryptology [16]. For quite a long time, it has been known that one could quickly recognize most composite numbers using Fermat’s little theorem. For cryptographical purposes, this idea was extended and it has yielded some fast probabilistic compositeness algorithms (for this, we refer to [52], the introduction of [28], and [9]). On the other hand, testing an arbitrary number for primality depended on integer factorization. For this era, see [18, 92, 95]. The reader interested in large or curious primes is referred to [80] as well as [68].

The year 1979 saw the appearance of the first general-purpose primality testing algorithm, designed by Adleman, Pomerance, and Rumely [3]. The running time of the algorithm was proved to be $O((\log N)^c \log \log \log N)$ for some effective $c > 0$. This algorithm was simplified and made practical by H. W. Lenstra and H. Cohen [28] and then successfully implemented by H. Cohen and A. K. Lenstra [27]. Motivated by our results with elliptic curves (see below), the algorithm was recently optimized by Bosma and Van der Hulst [15] (see also [60]). However, it is not possible to check the results of this algorithm independently without rewriting and rerunning the entire program; by contrast, our algorithm gives a “certificate” which enables a second programmer to verify our proof in a time much shorter than the original time.
In 1985, H. W. Lenstra (Jr.) introduced the use of elliptic curves in factorization. There was then hope to find a similar use for primality testing. This was first done by Goldwasser and Kilian [40] using the architecture of the DOWNRUN algorithm of [97] together with a theoretical algorithm due to Schoof [83]. They found that this algorithm recognizes primes in expected random polynomial time, at least assuming some very plausible conjectures in analytic number theory. Almost simultaneously, the first author [4] designed a practical algorithm based on the same ideas, but using results from the theory of elliptic curves over finite fields (see also [25] and [14] for a first insight). From a practical point of view, this algorithm is faster and yields a proof that the computation is correct in the form of a list of numbers by means of which one can easily check the primality properties (see §10). In another direction, the theory of *elliptic pseudoprime* tests and *elliptic pseudoprimes* was introduced [41, 61, 8].

Shortly afterwards, Adleman and Huang announced [2] that they designed a primality testing algorithm using curves of genus two whose expected running time is also polynomial, but without any unproven hypothesis. As for now, it seems that this algorithm has not been implemented.

The purpose of this paper is to describe the test due to the first author (which is known as the Elliptic Curve Primality Proving—ECPP—algorithm), together with the implementations made by the authors (other implementations include that of D. Bernardi for the class number one case and more recently that of Kaltofen, Valente and Yui [49] and that of Vardi (personal communication, August 1989) for the MATHEMATICA system).

Since there are considerable differences of detail between the implementations of the two authors, we have decided for the sake of clarity to present the algorithm solely as implemented by the second author. We make a few historical remarks in §8.1.

The plan of the paper is as follows. In §2, we recall some well-known properties of quadratic forms and fields necessary for presenting §3, which deals with the theory of Hilbert class fields of imaginary quadratic fields via modular forms. At this point, we introduce Weber’s functions as well as Dedekind’s \( \eta \). In §4, we present the relevant theory of elliptic curves in a manner similar to that of [57]: This unified approach is well suited for our purpose, which goes from classical elliptic curves over \( \mathbb{C} \) to curves over a finite field. Section 5 is concerned with primality testing using elliptic curves as used by Goldwasser and Kilian on the one hand, and the first author in his designing the ECPP algorithm on the other. A path towards analyzing ECPP is made in §6: We present some heuristic arguments concerning the ability for a number to be good with respect to ECPP as well as the probability of failure of a weak version of ECPP. In §7, we develop an efficient algorithm for constructing the Hilbert class field of an imaginary quadratic field by means of the functions introduced in §3. At this point, we introduce the concept of *Weber polynomials* and we detail a fast algorithm to compute the factorization of Weber polynomials over their genus field. In §8, we detail the computational routines we use in the implementation: Section 9 contains some typical running times for numbers of less than 300 digits and also some running times for larger numbers, most of all taken from [19] or discovered by the authors. Section 10 is briefly concerned with the second problem mentioned above, namely that of the *actual* proof we get by ECPP.
Notation. Throughout the paper, \( N \) will denote a probable prime, which means that \( N \) was not declared composite by any of the probabilistic primality testing algorithms which were used.

Historical note. The basic algorithm was designed and implemented by the first author in 1986. In 1987, the second author implemented a version of the algorithm based on a paper of Cohen [26]. In May 1989, the two authors met and merged part of their ideas to come up with the present paper, a longer version of which is available as [6].

2. SOME PROPERTIES OF QUADRATIC FORMS AND FIELDS

Our aim is to recall basic properties of quadratic forms and fields that are necessary for the following sections. We introduce first quadratic forms that are easy to compute with, and then quadratic fields that are well suited for explaining the theory. These are two sides of the same object.

2.1. Quadratic forms. The following results are well known and can be found in [35, 30]. Let \(-D\) be a fundamental discriminant, i.e., \( D \) is a positive integer which is not divisible by any square of an odd prime and which satisfies \( D \equiv 3 \mod 4 \) or \( D \equiv 4, 8 \mod 16 \). We can factor \(-D\) as \( q_1^* \cdots q_i^* \), where

\[
q_i^* = \begin{cases} \frac{(-1)^{q_i^* - 1}}{2} q & \text{if } q_i \text{ is an odd prime and } D = 3 \mod 4, 8 \mod 16, \\ \pm 4 & \text{otherwise.} \end{cases}
\]

In the sequel, the \( q_i \)'s are supposed to be ordered as follows: if \( D \equiv 0 \mod 4 \), then \( q_1 = 4 \) or \( 8 \). Then the \( q_i \)'s with \( q_i^* = q \) are listed in increasing order and finally the \( q_i \)'s with \( q_i^* = -q \), also in increasing order. We put \( l = \#\{i, q_i^* = q_i\} \). It is easy to see that

\[
t - l - 1 \equiv 0 \mod 2.
\]

A quadratic form of discriminant \(-D\) is a 3-tuple of integers \((a, b, c)\) such that \( b^2 - 4ac = -D \). There is a correspondence between the set of quadratic forms and the set of \( 2 \times 2 \) matrices with half integer coefficients. With \( Q = (a, b, c) \), we associate the \( 2 \times 2 \) matrix

\[
M(Q) = \begin{pmatrix} a & b/2 \\ b/2 & c \end{pmatrix}.
\]

Two forms \( Q \) and \( Q' \) of the same discriminant are said to be equivalent (or \( Q \sim Q' \)) if there exists \( N \) in \( \text{SL}_2(\mathbb{Z}) \) (i.e., a \( 2 \times 2 \) integer matrix with determinant 1) such that

\[
M(Q') = N^{-1} M(Q) N.
\]

This clearly defines an equivalence relation on quadratic forms. It can be shown that

**Proposition 2.1.** Each equivalence class contains exactly one form \((a, b, c)\) with \( a, b, c \) relatively prime and satisfying \(|b| \leq a \leq c\) and \((|b| = a \text{ or } a = c \Rightarrow b > 0)\). Such a form is called reduced.

There is an algorithm that computes a reduced form equivalent to a given form: we refer to the literature for this [85].

The set of primitive reduced quadratic forms of discriminant \(-D\), denoted by \( \mathcal{H}(-D) \), is finite (for \(|b| \leq \sqrt{D}/3\) if \((a, b, c)\) is reduced). Moreover, it is possible to define an operation on classes that gives to \( \mathcal{H}(-D) \) the structure
of an Abelian group. This operation is called the composition of classes and is ordinarily written multiplicatively. For the actual computation, we refer to [85]. The order of \( \mathcal{O}(D) \) is denoted by \( h(D) \). The neutral element \( F_D \) is called the principal form. It is equal to \((1, 0, D/4)\) or \((1, 1, (D + 1)/4)\) according as \(D \equiv 0\) or \(3 \pmod{4}\).

Let \( C = (a, b, c) \) be an element of \( \mathcal{O}(D) \). For \((x, y)\) in \(\mathbb{Z}^2\), put \(C(x, y) = ax^2 + bxy + cy^2\) and assume that \(a\) is prime to \(D\) (otherwise consider \(c\) instead of \(a\), since \(a\) and \(c\) cannot both have a common factor with \(D\)). Let \(p\) be a rational prime. The equation \(p = C(x, y)\) has a solution in \((x, y)\) only if the following conditions are satisfied:

\[
\left( \frac{-D}{p} \right) = +1 \quad \text{and} \quad \left( \frac{p}{q_i} \right) = \left( \frac{a}{q_i} \right), \quad 1 < i \leq t.
\]

(Hint: write \(4ap = (2ax + by)^2 + Dy^2\).)

Put \(\chi_i(a) = \chi_i(C) = (a/q_i)\) for all \(i\). This defines a map from \(\mathcal{O}(D)\) to \(Z_t = \{±1\}^t\) by

\[
\Xi: \mathcal{O}(D) \to Z_t,
\]

\[
C \mapsto (\chi_1(C), \ldots, \chi_t(C)).
\]

The following theorem was proven by Gauss:

Theorem 2.1. The map \(\Xi\) is onto: If we start from \(\varepsilon = (\varepsilon_1, \ldots, \varepsilon_t)\) satisfying \(\Pi_i \varepsilon_i = +1\), we can find a \(C\) such that \(\Xi(C) = \varepsilon\). Moreover, \(\Xi\) is a homomorphism. The associated cosets are called the genera and they inherit the group law. Each coset has cardinality \(e = h/g\), where \(g = 2^{l-1}\).

We define the principal genus as \(G_0 = \Xi^{-1}(+1, \ldots, +1)\). For each genus \(G_i\), we can find \(C_i\) in \(\mathcal{O}(D)\) such that \(G_i = C_iG_0\). Thus, the product of the genera \(G_i = C_iG_0\) and \(G_j = C_jG_0\) is \(G_k\) with \(C_k = C_i \cdot C_j\).

A prime \(p\) which is representable by a form of \(G_i\) is said to belong to \(G_i\) (this is denoted by \(p \in G_i(D)\)).

2.2. Quadratic fields. Consider now \(K = \mathbb{Q}(\sqrt{-D})\). The extension \(K/\mathbb{Q}\) is Abelian of degree 2, of Galois group \(\{1, \tau\}\), where \(\tau\) denotes complex conjugation. The ring of integers of \(K\) is \(\mathcal{O}_K = \mathbb{Z}[\omega]\), where

\[
\omega = \begin{cases} 
\sqrt{-D}/4 & \text{if } D \equiv 0 \mod 4, \\
1 + \sqrt{-D}/2 & \text{otherwise.}
\end{cases}
\]

The conjugate of an element \(\alpha = x + y\omega\) is \(\alpha' = \tau(\alpha) = x + y\tau(\omega)\). The trace (resp. the norm) of \(\alpha\) is \(T_K(\alpha) = \alpha + \tau(\alpha)\) (resp. \(N_K(\alpha) = \alpha\tau(\alpha)\)). If \(\alpha\) is an element of \(K\), its associates are the \(v\alpha\), where \(v\) is any unit of \(K\) (that is, \(N_K(v) = 1\)). The number of units is denoted by \(w(-D)\) and is equal to 6, 4, or 2 according to \(D\) equal to 3, 4, or \(> 4\).

The decomposition of the ideal \((p)\) in \(K\) is given by the following theorem:

Proposition 2.2. If \((-D/p) = +1\), the ideal \((p)\) splits as the product of two distinct ideals in \(K\). If \((-D/p) = 0\), \((p)\) ramifies, and if \((-D/p) = -1\), it is inert.

We conclude this section with
Proposition 2.3. The equation \( p = N_k(n) \) has a solution in \( \mathfrak{O}_K \) if and only if \( (p) \) splits as the product of two principal ideals in \( K \). This is equivalent to saying that \( p \) is represented by the principal form of discriminant \( -D \). In other words: \( 4p = A^2 + DB^2 \) with \( A \) and \( B \) in \( \mathbb{Z} \).

If \( p \) is representable by the principal form of discriminant \( -D \), we shall say that "\( p \) is a norm in \( \mathbb{Q}(\sqrt{-D}) \)" or simply "\( p \) is a norm" when the context is clear. Conversely, we shall say that "\( -D \) is good for \( p \)" if \( p \) is a norm. Thus, in general, \( (-D/p) = 1 \), that \( p \) splits in \( \mathbb{Q}(\sqrt{-D}) \), and even that \( p \) is representable by a form of the principal genus, are all necessary conditions for \( p \) to be a norm.

2.3. Genus field. The genus field of \( K \) is \( K_{\mathfrak{c}} = \mathbb{Q}(\sqrt{q_1^*}, \ldots, \sqrt{q_t^*}) \), the \( q_i \) being described above. The field \( K_{\mathfrak{c}} \) is the maximal unramified extension of \( K \) that is Abelian over \( \mathbb{Q} \). The Galois group of \( K_{\mathfrak{c}}/\mathbb{Q} \) is isomorphic to \( (\mathbb{Z}/2\mathbb{Z})^t \).

We recall that the Artin symbol associated with the quadratic form \( C \) (in fact with the genus \( G \) containing \( C \)) is (see [29]):

\[
\mathcal{A}_G = \left( \frac{K_{\mathfrak{c}}/K}{p} \right) \simeq (\chi_1(G), \ldots, \chi_t(G)),
\]

with \( \chi_i(G) = (q_i^*/p) \), where \( (p) = pp' \) is any prime number represented by a form of \( G \) and \( p \) the ideal above \( p \) in \( K \).

3. Modular forms

3.1. The modular group and the modular invariant \( j \). We follow [84]. The modular group is defined to be \( \Gamma = \text{SL}_2(\mathbb{Z})/\{\pm 1\} \). An element \( g = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \) of \( \Gamma \) acts on \( H = \{z \in \mathbb{C}, \text{Im}(z) > 0\} \) by

\[
gz = \frac{az + b}{cz + d},
\]

It is known that \( \Gamma \) is generated by \( S \) and \( T \) where

\[
S = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, \quad T = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}.
\]

A modular form of weight \( 2k \) (\( k \) any integer) is a function meromorphic everywhere on \( H \) and at infinity, satisfying

\[
\forall \left( \begin{array}{cc} a & b \\ c & d \end{array} \right) \in \text{SL}_2(\mathbb{Z}), \forall z \in H, \quad f(z) = (cz + d)^{-2k} f \left( \frac{az + b}{cz + d} \right).
\]

If the form is holomorphic everywhere (which implies \( k > 0 \) for nonconstant forms), we say that the form is regular.

Let \( L(1, \omega) = \mathbb{Z} + \omega \mathbb{Z} \) be a lattice in \( \mathbb{C} \) (\( \omega \in H \)). Put

\[
G_{2k}(L) = \sum_{(m,n) \neq (0,0)} \frac{1}{(m\omega + n)^{2k}},
\]

for \( k > 1 \). If \( k > 1 \), then \( G_{2k}(L) \) is a regular modular form of weight \( 2k \).

We put \( g_2(L) = 60G_4 \), \( g_3(L) = 140G_6 \), and \( \Delta = g_2^3 - 27g_3^2 \); these are regular modular forms of weight 4, 6, and 12, respectively. The modular invariant is then \( j = 12^3g_2^3/\Delta \). We have
Proposition 3.1. The function $j$ is a modular function (i.e., a modular form of weight 0), is holomorphic in $\mathbb{H}$, and has a simple pole at infinity. The function $j$ is a complex analytic isomorphism from $\mathbb{H}/\Gamma$ to $\mathbb{C}$.

One can show that the $q$-expansion of $j$ is (cf. [84])

\begin{equation}
    j(q) = \frac{1}{q} + 744 + \sum_{n \geq 1} c_n q^n,
\end{equation}

where the $c_n$ are positive integers. For a survey of the arithmetical and numerical properties of the $c_n$, see, for instance, [84, 63].

3.2. Complex multiplication for lattices. Let $L = L(1, \omega)$ be a lattice in $\mathbb{C}$. Put $M(L) = \{\alpha \in \mathbb{C}, \alpha L \subset L\}$. It is clear that $\mathbb{Z} \subset M(L)$. When $M(L)$ is greater than $\mathbb{Z}$, we say that $L$ has complex multiplication. It can be shown [53, Chapter 1] that if $L$ has complex multiplication, then $\omega$ belongs to a complex quadratic field $K = \mathbb{Q}(\sqrt{-D})$. Then $M(L)$ is an order of $K$, that is, a ring which is a free submodule of rank 2 over $\mathbb{Z}$ of $\mathcal{O}_K$, the ring of integers of $K$.

3.3. Class field theory of imaginary quadratic fields. Class field theory is one of the most remarkable achievements of mathematics. One of its motivating problems was the construction of the maximal unramified Abelian extension of an imaginary quadratic field (for a modern presentation of the classical approach, see [13]). An algebraic treatment was given by Deuring [34]. The theory was generalized in [87]. In the present paper, we only need to use a comparatively small part of the theory, which we specify below.

Let $-D$ be a fundamental discriminant and $K = \mathbb{Q}(\sqrt{-D})$. The Hilbert Class Field of $K$ is the maximal unramified Abelian extension of $K$ and is denoted by $K_H$ (see [34]). We have (see [13, 91]):

Theorem 3.1. The field $K_H$ can be obtained by adjoining to $K$ any value $j_r = j(\omega_r)$, where $\omega_r$ is the complex number associated with $C_r$, i.e., $\omega_r = \omega(C_r) = (-b_r + i\sqrt{D})/(2a_r)$ with $C_r = (a_r, b_r, c_r)$ in $\mathcal{H}(-D)$. The minimal polynomial of the $j_r$'s is denoted by $H_D(X)$. It follows that $K_H$ is precisely the splitting field of $H_D(X)$.

The Galois group $\Sigma_H$ of $K_H/K$ is isomorphic to $\mathcal{H}(-D)$. If $C$ is an element of $\mathcal{H}(-D)$, the corresponding element $\sigma_C$ of $\Sigma_H$ acts on $j(C')$ by

\begin{equation}
    \sigma_C(j(C')) = j(C^{-1} \cdot C').
\end{equation}

We also require the following (see [31, 33]):

Theorem 3.2. A rational prime $p$ is a norm in $K$ if and only if $(p)$ splits completely in $K_H$. This is equivalent to saying that $H_D(X) \pmod{p}$ has only simple roots and they are all in $\mathbb{Z}/p\mathbb{Z}$. Moreover, we have that

\[ 4p = A^2 + DB^2 \]

has a solution in rational integers $(A, B)$ if and only if $H_D(X)$ splits completely modulo $p$.

The last statement follows from Proposition 2.3.
3.4. Dedekind's and Weber's functions. Let \( z \) be any complex number and put \( q = \exp(2i\pi z) \). Dedekind's \( \eta \) function is defined by [91, §24, p. 85]

\[
\eta(z) = q^{1/24} \prod_{m \geq 1} (1 - q^m).
\]

We can expand \( \eta \) as [91, §34, p. 112]

\[
\eta(z) = q^{1/24} \left( 1 + \sum_{n \geq 1} (-1)^n (q^{n^2 + 1} + q^{n^2 + 1/2}) \right).
\]

The function \( \eta \) is a modular form of weight 1/2 with a complicated multiplier function.

If we let \( \zeta_n \) stand for \( \exp(2i\pi/n) \), the Weber functions are [91, §34, p. 114]

\[
f(z) = \frac{\zeta_{48}^{-1} \eta((z + 1)/2)}{\eta(z)},
\]

\[
f_1(z) = \frac{\eta(z/2)}{\eta(z)},
\]

\[
f_2(z) = \sqrt{2} \frac{\eta(2z)}{\eta(z)},
\]

and [91, §54, p. 179]

\[
\gamma_2 = \frac{f^{24} - 16}{f^8},
\]

\[
\gamma_3 = \frac{(f^{24} + 8)(f_1^8 - f_2^8)}{f^8}.
\]

We can reconstruct the modular invariant \( j \) through [91, §54, p. 179]:

\[
j(z) = \frac{(f^{24} - 16)^3}{f^{24}} = \frac{(f_1^{24} + 16)^3}{f_1^{24}} = \frac{(f_2^{24} + 16)^3}{f_2^{24}} = \gamma_2^3 = \gamma_3^2 + 1728.
\]

We also note the following transformation formulas [91, §34, p. 113]. First,\n
\[
\eta(z + 1) = \zeta_{24} \eta(z), \quad \eta(-1/z) = \sqrt{z/i} \eta(z),
\]

from which

\[
f(z + 1) = \zeta_{48}^{-1} f_1(z), \quad f_1(z + 1) = \zeta_{48}^{-1} f(z), \quad f_2(z + 1) = \zeta_{24} f_2(z)
\]

and

\[
f(-1/z) = f(z), \quad f_1(-1/z) = f_2(z), \quad f_2(-1/z) = f_1(z).
\]

4. Elliptic curves

4.1. Definition. We follow Lenstra [57]. Let \( k \) be a field of characteristic 0 or prime to 6. Let \( \mathbb{P}^2(k) \) be the projective plane over \( k \). The equivalence class containing \( (x, y, z) \) is denoted by \( (x : y : z) \).

An elliptic curve is a pair \( E = (a, b) \) (which we sometimes write as \( E(a, b) \)) of elements of \( k \) such that \( \Delta_E = -16(4a^3 + 27b^2) \neq 0 \). This quantity is called the discriminant of the curve \( E \). We also define the invariant of the curve \( j(E) = 2834283a^3/(4a^3 + 27b^2) \).
The set of points of $E$ over $k$ is:

$$E(k) = \{(x : y : z) \in \mathbb{P}^2(k), y^2z = x^3 + axz^2 + bz^3\}.$$ 

There is exactly one point of $E(k)$ with $z = 0$, namely $(0 : 1 : 0)$, called the point at infinity, denoted by $O_E$. The set $E(k)$ can be made an Abelian group with an operation denoted by $+$ using the tangent-and-chord method. Suppose temporarily that $k = \mathbb{R}$. Then $E(\mathbb{R})$ is a projective curve that we can look at. In order to add two points $M_1$ and $M_2$ resulting in $M_3$, we draw the line $M_1M_2$ (or the tangent if $M_1 = M_2$). This line intersects $E$ in a third point, $P$, whose reflection in the $x$-axis yields the sum $M_3 = M_1 + M_2$. The symmetric of a point $M = (x : y : z)$ is $-M = (x : -y : z)$, and the neutral element is the point at infinity. From a practical point of view, the coordinates of a nontrivial $M_3$ are

$$\begin{align*}
x_3 &= \lambda^2 - x_1 - x_2, \\
y_3 &= \lambda(x_1 - x_3) - y_1,
\end{align*}$$

where

$$\lambda = \begin{cases} (y_2 - y_1)(x_2 - x_1)^{-1} & \text{if } x_2 \neq x_1, \\
(3x_1^2 + a)(2y_1)^{-1} & \text{otherwise.}
\end{cases}$$

We can compute $kP$ using the binary method [52] (see also [27]) or addition-subtraction chains [71].

![Figure 1. An elliptic curve over \( \mathbb{R} \)](image)
4.2. $k = \mathbb{Z}/p\mathbb{Z}$. Let $p$ be a prime number greater than 3. Let $E$ be an elliptic curve defined over $\mathbb{Z}/p\mathbb{Z}$. We do not intend to explain Deuring’s work concerning its reduction modulo $p$, but the interested reader may consult [53, Chapter 13] and the references given there. It can be shown that $E$ can be described as the reduction modulo $p$ of an elliptic curve $E(C)$ with complex multiplication by an order of a quadratic field $K = \mathbb{Q}(\sqrt{-D})$.

From a practical point of view, one can construct a curve which has complex multiplication by the ring of integers of $\mathbb{Q}(\sqrt{-D})$ in the following way. Suppose that $p$ is a norm in $K$: $(p) = pp' = (\pi)(\pi')$ and $p$ splits completely in $K_H$ as $p = \mathfrak{P}_1 \cdots \mathfrak{P}_h$ (with $h = h(-D)$). The polynomial $H_D(X)$ splits completely modulo $p$. Let $j$ be any root of $H_D(X)$ modulo $p$ and $E$ an elliptic curve of invariant $j$. Then $#E(\mathbb{Z}/p\mathbb{Z}) = p+1-\sum_{K} = p+1-A$ with $|A| < 2\sqrt{p}$ (this theorem was originally proved by Hasse) and $E$ has complex multiplication by the ring $\mathfrak{K}^K$.

Concerning the structure of $E(\mathbb{Z}/p\mathbb{Z})$ as an Abelian group, we have [22]:

**Theorem 4.1.** The group $E(\mathbb{Z}/p\mathbb{Z})$ is either cyclic or the product of two cyclic groups of order $m_1$ and $m_2$ that satisfy

$$m_1|m_2, \quad m_1|\gcd(m, p-1),$$

where $m = #E(\mathbb{Z}/p\mathbb{Z})$.

5. **Primality testing**

5.1. Traditional primality testing and the DOWNRUN process. Before the advent of the Jacobi sums algorithm, the main method for primality testing was to use some known factors of $N^t - 1$, $t = 1, 2, 3, 4, 6$, involving either some converse of Fermat’s theorem, or Lucas sequences or a generalization thereof.

The simplest way to prove that an odd number $N$ is prime is to prove that the group $(\mathbb{Z}/N\mathbb{Z})^\times$ is cyclic (and that $N$ is not a prime power). For this, we need only to exhibit a generator of this group. This yields the following theorem. (This is not the optimal theorem, but we cite it for the sake of simplicity.)

**Theorem 5.1.** If there exists an $a$ prime to $N$ such that $a^{N-1} \equiv 1 \pmod{N}$ but $a^{(N-1)/q} \not\equiv 1 \pmod{N}$ for every prime divisor $q$ of $N-1$, then $N$ is prime.

Of course, we need to factor $N-1$. Starting with a number $N_0$, a favorable situation occurs whenever we can completely factor $N_0 - 1$ or we find that $N_0 - 1$ has a large factor $N_1$ which is probably prime: such a number $N_0$ we call probably factored. The problem is then reduced to proving that $N_1$ is prime. Also, we can use some factors of $N_1 - 1$ to help us in our job.

This idea forms the DOWNRUN process of [97]: Build a decreasing sequence of probable primes $N_0 > N_1 > \cdots > N_k$ such that the primality of $N_{i+1}$ implies that of $N_i$ (see [52, pp. 376–377]). Indeed, this is a factor and conquer method. The problem is that for each $N_i$, there is only a limited number of candidates that we can try to factor. We will see that this difficulty is overcome when using elliptic curves.

5.2. **The Goldwasser-Kilian algorithm.** From [40], we have:

**Theorem 5.2.** Let $N$ be an integer prime to 6, $E$ an elliptic curve over $\mathbb{Z}/N\mathbb{Z}$, together with a point $P$ on $E$ and $m$ and $s$ two integers with $s \mid m$. For
each prime divisor $q$ of $s$, we put $(m/q)P = (x_q : y_q : z_q)$. We assume that $mP = O_E$ and $\gcd(z_q, N) = 1$ for all $q$. Then, if $p$ is a prime divisor of $N$, one has $#E(\mathbb{Z}/p\mathbb{Z}) \equiv 0 \mod s$.

We have also:

**Corollary 5.1.** With the same conditions, if $s > (\sqrt[3]{N} + 1)^2$, then $N$ is prime.

Combining this theorem with Schoof’s algorithm that computes $#E(\mathbb{Z}/p\mathbb{Z})$ in time $O((\log p)^{8+e})$ (see [56]), we obtain the Goldwasser-Kilian algorithm.

**procedure** GK($N$)

1. choose an elliptic curve $E$ over $\mathbb{Z}/N\mathbb{Z}$, for which the number of points $m$ (computed with Schoof’s algorithm) satisfies $m = 2q$, with $q$ a probable prime;
2. if $(E, m)$ satisfies the conditions of the theorem with $s = m$, then $N$ is prime, otherwise it is composite;
3. the primality of $q$ is proved in the same way;
4. end.

We see that we have solved one of the problems arising in the ordinary DOWRNE: this time, we have many numbers which we can try to factor.

The problem with GK is that School’s algorithm seems almost impossible to implement (however, see [5]). We will use instead the properties of elliptic curves over finite fields related to complex multiplication.

5.3. The ECPP algorithm. In algorithm GK, we begin by searching for a curve and then compute its number of points. Here, we do exactly the contrary. We get:

**procedure** ECPP($N$);

(*$N$ is a probable prime *)

1. set $i := 0, N_0 := N$;
2. while $N_i > N_{\text{small}}$
   1. find a fundamental discriminant $-D_i$ which is good for $N_i$; in other words, $N_i = \pi_i\pi'_i$ in $K = \mathbb{Q}(\sqrt{-D_i})$ (see §2);
   2. if one of the $w(-D_i)$ numbers $m_1, \ldots, m_w$ ($m_r = N_K(v_r\pi_i - 1)$, where $v_r$ is a unit in $K$) is probably factored, go to step 2.3 else go to 2.1;
   3. store $\{i, N_i, D_i, v_r\pi_i, m_r, F_i\}$, where $m_r = F_iN_{i+1}$. Here, $F_i$ is a completely factored integer and $N_{i+1}$ a probable prime; set $i := i + 1$ and go to step 2.1;
   4. compute a root $j$ of $H_{D_i}(X) \equiv 0 \mod N_i$;
   5. compute an equation of the curve $E_i$ of invariant $j$ and whose cardinality modulo $N_i$ is $m_i$;
   6. find a point $P_i$ on the curve $E_i$;
   7. check the conditions of the theorem with $s = N_{i+1}$ and $m = m_i$: in other words, check that $Q_i = F_iP_i \neq O_{E_i}$ but $sQ_i = O_{E_i}$;
3. end.

Finding $m_r$ which is probably factored will be referred to as “finding a suitable $m$.”
6. Analysis

6.1. Theoretical results. The running time of GK is analyzed in the following theorems [40, 56].

Theorem 6.1. Suppose that there exist two positive constants \( c_1 \) and \( c_2 \) such that the number of primes in the interval \([x; x + \sqrt{x}](x \geq 2)\) is greater than \( c_1 \sqrt{x}(\log x)^{-c_2} \). Then GK proves the primality of \( N \) in expected time \( O((\log N)^{10+c_2}) \).

Theorem 6.2. There exist two positive constants \( c_3 \) and \( c_4 \) such that, for all \( k \geq 2 \), the proportion of prime numbers \( N \) of \( k \) bits for which the expected time of GK is bounded by \( c_3 (\log N)^{11} \) is at least \( 1 - c_4 2^{-k/\log k} \).

As for ECPP, we only have the heuristic analysis cited in [54]. These authors find that the running time of the algorithm is roughly \( O((\log N)^{6+\varepsilon}) \) for some \( \varepsilon > 0 \).

The remaining of this section is devoted to some practical considerations concerning ECPP.

6.2. What is a good discriminant? Let \( p \) be a prime number. Then \( p \) is a norm in \( \mathbb{Q}(\sqrt{-D}) \) if and only if \( p \) is represented by the principal form of \( \mathcal{P}(-D) \). As in §2, let \(-D = q_1^* \cdots q_r^*\), its class number is \( h = h(-D) \), and the number of genera is \( g = 2^{r-1} \). The prime \( p \) is represented by a form of \( G_0 \) if and only if \( \forall i, \chi_i(p) = +1 \) (see §2), which occurs with probability \( 1/2^r \). Given this, \( p \) is represented by \( F_D \) with conditional probability \( g/h \). We deduce

Proposition 6.1. A prime \( p \) is represented by \( F_D \) with probability \( 1/(2h) \).

A proof with less handwaving can be found in [33, Chapter 8].

6.3. Practical considerations: good and bad numbers. For practical purposes, we are only interested in fundamental discriminants \( D \) (\( D < 10^6 \)) with \( h(-D) \leq 50 \) (the parameters 10^6 and 50 are somewhat arbitrary, and represent the extreme limits of what we expect to need). They form a set \( \mathcal{D} \). We have (presumably) that \( \#\mathcal{D} = 10628 \). Let \( H \) and \( G \) be two integers. We write \( ND(H, G) \) for the number of \( D \) in \( \mathcal{D} \) for which \( h(-D) = H \) and \( g(-D) = G \). In Table 1 (next page), we indicate the values of \( ND(H, G) \) for \( H \leq 50 \) (they agree with those of [20]). From this, we can deduce the number of \( D \) such that \( h(-D) \leq 50 \) and with given value of \( H/G \). This quantity represents the degree of the final polynomial of which we want a root, and its inverse \((G/H)\) is just the probability that \( N \) is a norm in \( K \) (provided that \((-D/N) = +1\)). This yields Table 2 (next page).

Let \( S \) be a finite set of primes (here 4 and 8 are assumed to be distinct primes). We define \( N_p(S) \) to be the number of \( D \) in \( \mathcal{D} \) which are divisible by at least one prime of \( S \): This quantity is tabulated in Table 3 (see p. 41). From the above results, it is quite clear that bad numbers are those which are quadratic nonresidues modulo small primes, such as \( N \equiv -1 \mod 12 \), which kill off one third of our discriminants. As an example, it is interesting to compute the smallest prime which does not split in any of the quadratic fields with class number 1. This number is 3167 (the next one is 607823).
Table 1. Statistics on the discriminants $D$ with $h(-D) \leq 50$

<table>
<thead>
<tr>
<th>$H/G$</th>
<th>$D_{\text{min}}$</th>
<th>$D_{\text{max}}$</th>
<th>#</th>
<th>$H/G$</th>
<th>$D_{\text{min}}$</th>
<th>$D_{\text{max}}$</th>
<th>#</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
<td>27</td>
<td>1</td>
<td>983</td>
<td>103387</td>
<td>93</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>15</td>
<td>18</td>
<td>2</td>
<td>831</td>
<td>126043</td>
<td>174</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>23</td>
<td>16</td>
<td>4</td>
<td>935</td>
<td>106723</td>
<td>283</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>39</td>
<td>30</td>
<td>1</td>
<td>887</td>
<td>166147</td>
<td>83</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>84</td>
<td>24</td>
<td>2</td>
<td>671</td>
<td>134467</td>
<td>255</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>47</td>
<td>25</td>
<td>1</td>
<td>719</td>
<td>133387</td>
<td>73</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>87</td>
<td>51</td>
<td>2</td>
<td>791</td>
<td>164803</td>
<td>187</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>71</td>
<td>31</td>
<td>2</td>
<td>1239</td>
<td>136843</td>
<td>333</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>95</td>
<td>62</td>
<td>2</td>
<td>3080</td>
<td>89947</td>
<td>173</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>260</td>
<td>56</td>
<td>16</td>
<td>7140</td>
<td>40755</td>
<td>15</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>420</td>
<td>13</td>
<td>33</td>
<td>839</td>
<td>222643</td>
<td>101</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>199</td>
<td>34</td>
<td>2</td>
<td>1079</td>
<td>189883</td>
<td>219</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>119</td>
<td>87</td>
<td>1</td>
<td>1031</td>
<td>210907</td>
<td>103</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>167</td>
<td>41</td>
<td>2</td>
<td>959</td>
<td>217627</td>
<td>271</td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>327</td>
<td>88</td>
<td>36</td>
<td>1295</td>
<td>175123</td>
<td>397</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
<td>231</td>
<td>118</td>
<td>37</td>
<td>1487</td>
<td>158923</td>
<td>85</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>191</td>
<td>37</td>
<td>38</td>
<td>1199</td>
<td>289963</td>
<td>237</td>
</tr>
<tr>
<td>14</td>
<td>2</td>
<td>215</td>
<td>95</td>
<td>39</td>
<td>1439</td>
<td>253507</td>
<td>115</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>239</td>
<td>68</td>
<td>40</td>
<td>1271</td>
<td>260947</td>
<td>251</td>
</tr>
<tr>
<td>16</td>
<td>2</td>
<td>407</td>
<td>101</td>
<td>4</td>
<td>2255</td>
<td>250387</td>
<td>438</td>
</tr>
<tr>
<td>16</td>
<td>4</td>
<td>399</td>
<td>160</td>
<td>40</td>
<td>2415</td>
<td>148603</td>
<td>223</td>
</tr>
<tr>
<td>16</td>
<td>8</td>
<td>1140</td>
<td>60</td>
<td>41</td>
<td>1151</td>
<td>296587</td>
<td>109</td>
</tr>
<tr>
<td>16</td>
<td>16</td>
<td>5460</td>
<td>1</td>
<td>42</td>
<td>1959</td>
<td>280267</td>
<td>339</td>
</tr>
<tr>
<td>17</td>
<td>1</td>
<td>383</td>
<td>45</td>
<td>43</td>
<td>1847</td>
<td>300787</td>
<td>106</td>
</tr>
<tr>
<td>18</td>
<td>2</td>
<td>335</td>
<td>150</td>
<td>44</td>
<td>1391</td>
<td>319867</td>
<td>261</td>
</tr>
<tr>
<td>19</td>
<td>1</td>
<td>311</td>
<td>47</td>
<td>44</td>
<td>2135</td>
<td>319243</td>
<td>430</td>
</tr>
<tr>
<td>20</td>
<td>2</td>
<td>776</td>
<td>150</td>
<td>45</td>
<td>1319</td>
<td>308323</td>
<td>154</td>
</tr>
<tr>
<td>20</td>
<td>4</td>
<td>455</td>
<td>200</td>
<td>46</td>
<td>2615</td>
<td>308947</td>
<td>267</td>
</tr>
<tr>
<td>21</td>
<td>1</td>
<td>431</td>
<td>85</td>
<td>47</td>
<td>3023</td>
<td>375523</td>
<td>107</td>
</tr>
<tr>
<td>22</td>
<td>2</td>
<td>591</td>
<td>139</td>
<td>48</td>
<td>1751</td>
<td>333547</td>
<td>343</td>
</tr>
<tr>
<td>23</td>
<td>1</td>
<td>647</td>
<td>68</td>
<td>48</td>
<td>3615</td>
<td>335203</td>
<td>621</td>
</tr>
<tr>
<td>24</td>
<td>2</td>
<td>695</td>
<td>167</td>
<td>48</td>
<td>4935</td>
<td>275587</td>
<td>355</td>
</tr>
<tr>
<td>24</td>
<td>4</td>
<td>759</td>
<td>240</td>
<td>48</td>
<td>11220</td>
<td>94395</td>
<td>46</td>
</tr>
<tr>
<td>24</td>
<td>8</td>
<td>2184</td>
<td>104</td>
<td>49</td>
<td>1511</td>
<td>393187</td>
<td>132</td>
</tr>
<tr>
<td>25</td>
<td>1</td>
<td>479</td>
<td>95</td>
<td>50</td>
<td>1799</td>
<td>389467</td>
<td>345</td>
</tr>
<tr>
<td>26</td>
<td>2</td>
<td>551</td>
<td>190</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2. Number of discriminants $D$ with given $H/G$ for $H \leq 50$

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>65</td>
<td>6</td>
<td>683</td>
<td>11</td>
<td>610</td>
<td>16</td>
<td>187</td>
<td>21</td>
<td>424</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>161</td>
<td>7</td>
<td>409</td>
<td>12</td>
<td>788</td>
<td>17</td>
<td>264</td>
<td>22</td>
<td>261</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>335</td>
<td>8</td>
<td>434</td>
<td>13</td>
<td>227</td>
<td>18</td>
<td>271</td>
<td>23</td>
<td>335</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>395</td>
<td>9</td>
<td>581</td>
<td>14</td>
<td>174</td>
<td>19</td>
<td>284</td>
<td>24</td>
<td>343</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>535</td>
<td>10</td>
<td>588</td>
<td>15</td>
<td>323</td>
<td>20</td>
<td>251</td>
<td>25</td>
<td>440</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

6.4. A theoretical failure case. Corollary 5.1 cannot be applied when $s \leq (\sqrt{p} + 1)^2$. In particular, we cannot use it when the number of points, $m$, is a perfect square and $E(\mathbb{Z}/p\mathbb{Z})$ is isomorphic to $(\mathbb{Z}/M\mathbb{Z}) \times (\mathbb{Z}/M\mathbb{Z})$ with
ELLiptic CURves AND PRIMality PROVING

Table 3. Divisibility of the discriminants

<table>
<thead>
<tr>
<th>$S$</th>
<th>$N_p(S)$</th>
<th>$S$</th>
<th>$N_p(S)$</th>
<th>$S$</th>
<th>$N_p(S)$</th>
<th>$S$</th>
<th>$N_p(S)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>{3}</td>
<td>2495</td>
<td>{3,4}</td>
<td>3669</td>
<td>{3,5}</td>
<td>3825</td>
<td>{3,4,5}</td>
<td>4803</td>
</tr>
<tr>
<td>{4}</td>
<td>1540</td>
<td>{5}</td>
<td>1744</td>
<td>{4,5}</td>
<td>3020</td>
<td>{3,4,5,7,8}</td>
<td>6382</td>
</tr>
</tbody>
</table>

$m = M^2$. A necessary condition for that is

$$M | p - 1.$$  

We also have $\sqrt{p} - 1 \leq M \leq \sqrt{p} + 1$, by Hasse’s theorem. Put $\lfloor \sqrt{p} \rfloor = a$ and $p = a^2 + r$, with

$$0 < r < 2a + 1.$$  

Then

$$a \leq M \leq a + 1.$$  

Suppose first that $M = a$. Then (18) implies $a|a^2 + r - 1$, that is, $a|r - 1$. There are two cases. First, when $r = 1$, one has $p = a^2 + 1$ and $E$ has complex multiplication by $\mathbb{Q}(\sqrt{-D})$, with $-D = (m - p - 1)^2 - 4p = -4a^2$. When $r > 1$, (19) implies $r - 1 = a$ and thus $p = a^2 + a + 1$. It is then easy to see that $E$ has complex multiplication by $\mathbb{Q}(\sqrt{-3})$.

7. Precomputations

This rather lengthy section deals with the effective construction of the Hilbert Class Field of $K = \mathbb{Q}(\sqrt{-D})$. This will be done using $j$ and other modular functions, especially Weber’s class invariants. For this purpose, we introduce the following notation. Let $u$ be any complex function. We will denote by $H_D[u](X)$ the minimal polynomial of $u(\omega)$ over $\mathbb{Q}$ (remember that $\sigma_K = \mathbb{Z}[\omega]$, where $\omega$ has been defined in §2). When $u = j$, we will abbreviate $H_D[u]$ to $H_D$.

7.1. Hilbert polynomials. The determination of $j$ as an algebraic integer in $\mathbb{Q}(j)$ has been studied by many authors, including Weber [91], Greenhill [42], Watson [90], Berwick [11], and more recently Gross and Zagier [44] (see also [36]).

We first prefer a basic approach. The simplest way to compute $j$ is to compute $H_D(X)$ using floating-point numbers (see [50, 31, 51]). In order to recognize that we have the right polynomial, we use an easy corollary of the work of Gross and Zagier, that can be stated as follows.

Proposition 7.1. The norm of $j$ in $\mathbb{Q}(j)$, which is the same as $H_D(0)$, is the cube of an integer in $\mathbb{Z}$.

It is worth remarking at this point that we do not need to prove that our calculations with $j$ are correct. If in fact they are, they will lead to elliptic curves which have the properties we need for proving primality, but the primality proof depends only on our computations on those curves. Thus, we may find it convenient in the algorithm to work to limited floating-point accuracy and confirm our $j$-value without formal proof using observations like Proposition 7.1.
We want to evaluate \( j(z) \) as fast as possible. For this, we compute in sequence \( \eta(z) \), \( \eta(2z) \), \( f_2(z) \), and \( j(z) \). The heart of the computation being the evaluation of \( \eta(z) \), we now study the optimal choice of the parameters. Let us define

\[
\mathcal{N}(q) = \sum_{n \geq 1} (-1)^n \left( q^{n(3n-1)/2} + q^{n(3n+1)/2} \right),
\]

and

\[
\mathcal{N}_N(q) = \sum_{n=1}^{N} (-1)^n \left( q^{n(3n-1)/2} + q^{n(3n+1)/2} \right),
\]

where as usual \( q = \exp(2\pi i z) \). We want to compute the error made when computing \( \mathcal{N}_N(q) \) instead of \( \mathcal{N}(q) \). We put \( q = \rho \exp(i\theta) = \rho (\cos \theta + i \sin \theta) \).

The following proposition is easy to establish.

**Proposition 7.2.** There holds

\[
|\mathcal{N}(q) - \mathcal{N}_N(q)| \leq 6\rho^{3N^2/2}.
\]

We have to evaluate \( j \) for values of \( z \) of the form \( z = (-b + i\sqrt{-D})/2a \), where \( (a, b, (b^2 + D)/4a) \) is a primitive reduced form of discriminant \(-D\). When there is no ambiguity, we write \( j(a, b) \) for \( j((-b + i\sqrt{-D})/2a) \). We put \( q = \rho e^{i\theta} \), with \( \rho = e^{-\pi\sqrt{-D}/a} \) and \( \theta = -\pi b/a \). Since this form is reduced, \( a \leq \sqrt{-D}/3 \). We deduce that \( \rho \leq e^{-\pi/3} < 4.34 \times 10^{-3} \).

Now, we remark that if \( (a, b, c) \) is an ambiguous form, then \( j(a, b) \) is a real number. When \( (a, b, c) \) is nonambiguous, we get

\[
j(a, -b) = j(a, b)
\]

(conjugation in \( \mathbb{C} \)), which halves the computation. After we have computed the \( h \)-values of \( j \), we build \( H_D(X) \).

By means of the \( q \)-expansion of \( j \), it is not hard to see that \( \log |j| \approx \pi \sqrt{-D}/a \). The number of decimal digits of \( j(q) \) is asymptotically \( \pi \sqrt{-D}/(a \log 10) \). We have to compute the coefficients of \( H_D(X) \) to within 0.5. The precision required is thus

\[
\text{Prec}(D) = \left( \frac{h}{[h/2]} \right) \frac{\pi \sqrt{-D}}{\log 10} \sum \frac{1}{a} + \nu_0,
\]

where the sum is taken over all primitive reduced forms of discriminant \(-D\), and \( \nu_0 \) a positive constant that takes care of the rounding error and the error made in our estimation of \( \log |j| \) (typically \( \nu_0 = 10 \)).

Suppose we want to compute \( j(a, b) \). Then, using (7.2), we compute \( \eta(kz) \) to the order

\[
\sqrt{S \times \frac{a}{k}},
\]

where

\[
S = \frac{2 \log 6 + \text{Prec}(D) \log 10}{3 \pi \sqrt{-D}}.
\]

We then form all products of the form \( X - j \), grouping terms of the type \( (X - j) \) and \( (X - \bar{j}) \), to get

\[
(X - j)(X - \bar{j}) = X^2 - (j + \bar{j})X + j\bar{j},
\]

which reduces computational errors.
We check the result with Proposition 7.1. If we find that \( HD(0) \) is the cube of an integer to within 0.5, we are confident that the computed polynomial is indeed the one we were looking for.

The coefficients of these polynomials become very large. For example, \( D = 23 \) already yields

\[
H_{23}(X) = X^3 + 3491750X^2 - 5151296875X + 233753.
\]

Thus, it may be desirable to use subsidiary functions on subgroups of \( \Gamma \).

**7.2. Weber polynomials.** Let \( u(z) \) denote any modular function: Weber calls \( u(\omega) \) a class invariant if \( u(\omega) \) is in \( K(j(\omega)) = K_H \) (\( \omega \) is the generator of \( \mathcal{O}_K \)). It turns out that there are a lot of alternative choices of class invariants other than \( j \).

The following results can be found in [91, §125-144] or in [12, 82].

**Theorem 7.1** [91, §125, p. 459]. Let \( z \) be a quadratic number defined by \( Az^2 + Bz + C = 0 \). If

\[
3 \mid B, \quad 3 \nmid A, \quad 3 \nmid B^2 - 4AC,
\]

we have

\[
Q(\gamma_2(z)) = Q(j(z)).
\]

Note that the conditions are redundant, since \( A \) and \( B \) cannot be both divisible by 3 (else \( D \equiv 0 \mod 3 \)). Moreover, a careful look at the proof of this in [91, §125] shows that we can replace the above conditions by \( A \equiv C \equiv 0 \mod 3 \) and \( B \not\equiv 0 \mod 3 \). From this, we deduce a very simple algorithm to compute the correct value of the conjugates of \( \gamma_2(\omega) \). We start from a form \( (a, b, c) \) associated with \( z_0 \) and we compute an equivalent form satisfying the above conditions, say \( (A, B, C) \) associated with \( z \). We use the following procedure.

**procedure** GAMMA2\((a, b, c)\)

1. if \( a \not\equiv 0 \mod 3 \), then choose \( k \) such that \( B \equiv b + 2ak \equiv 0 \mod 3 \); take \( k \equiv -b/(2a) \mod 3 \) and \((a, b + 2ak, c + bk + ak^2)\) satisfies one of the above conditions;
2. if \( a \equiv 0 \mod 3 \), but \( b \not\equiv 0 \mod 3 \), then find \( k \) such that \( C \equiv c + bk + ak^2 \equiv 0 \mod 3 \); a solution is given by \( k \equiv -c/b \mod 3 \);
3. compute \( \gamma_2(z) = e^{2i\pi k/3}\gamma_2(z_0) \). This is valid because of (14) and (11).

From a practical point of view, the computation of \( \gamma_2(z) \) is thus quite fast. It turns out that its coefficients are smaller than those of the original \( H_D(X) \). For example, for \( D = 23 \), we find

\[
H_{23}[\gamma_2](X) = X^3 + 155X^2 + 650X + 23375.
\]

When \( D \equiv 3 \mod 6 \), we have the following result.

**Theorem 7.2** [91, §134, p. 502]. If \( Az^2 + Bz + C = 0 \) with \( 2 \nmid A \), then

\[
Q(\sqrt{-D}\gamma_3(z)) = Q(j(z)).
\]

For instance, if \( D = 15 \), then

\[
H_D[\sqrt{-D}\gamma_3](X) = X^2 - 1575X - 218295.
\]
We can also use some power of the functions \( f, f_1, \) or \( f_2 \). We extract the following results from [51] (alternatively, see the references above). It is assumed from now on that \( D \not\equiv 0 \mod 3 \). With each value of \( D \mod 32 \), we have a canonical choice for \( u \). Hence, we write \( W_D(X) \) for the corresponding minimal polynomial.

<table>
<thead>
<tr>
<th>( D \mod 8 )</th>
<th>( u )</th>
<th>( W_D(0) )</th>
<th>( \deg(W_D) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 mod 4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( D/4 \equiv \pm 2 \mod 8 )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 mod 8</td>
<td>( f(\sqrt{-D}) )</td>
<td>(-2^h)</td>
<td>( 3h )</td>
</tr>
<tr>
<td>5 mod 8</td>
<td>( f(\sqrt{-D})^4 )</td>
<td>( \pm 2^h )</td>
<td>( h )</td>
</tr>
<tr>
<td>1 mod 8</td>
<td>( f(\sqrt{-D})^2 )</td>
<td>( (-1)^h )</td>
<td>( h )</td>
</tr>
</tbody>
</table>

Weber also gives conditions for more general \( z \) to satisfy the same properties. (One should also consult [88].) By extension, we will call \( \text{class invariant} \) any conjugate of \( u(\omega) \) for a suitable \( u \).

**Theorem 7.3.** Suppose \( Az^2 + 2Bz + C = 0 \) with \( 4B^2 - 4AC = -4D \), \( A \) and \( C \) odd, \( 3 \| B \), or equivalently, \( A \equiv C \equiv 0 \mod 3 \) and \( B \not\equiv 0 \mod 3 \). Then
1. in the case where \( D \equiv 1, 5, \pm 2 \mod 8 \): if \( B \equiv 2((2/A) - 1) \mod 8 \), then \( f(z)^2/\sqrt{2} \) (resp. \( f(z)^4, f_1(z)/\sqrt{2} \)) is a class invariant;
2. in the case where \( D \equiv 3, 7 \mod 8 \): if \( B \equiv 4((2/A) - 1) \mod 16 \), then \( f(z) \) (resp. \( f(z)/\sqrt{2} \)) is a class invariant.

We only sketch the proof in the case \( D \equiv 7 \mod 8 \). We combine the following results.

**Proposition 7.3** [91, §127, p. 467]. Let \( z \) be a root of \( Az^2 + 2Bz + C = 0 \), with \( -4D = 4(B^2 - AC) \). Assume that \( 3 \| B \) and that \( A \) and \( C \) are both odd and nondivisible by \( 3 \) (or \( A \equiv C \equiv 0 \mod 3 \) and \( B \not\equiv 0 \)). Then \( f_8(z) \) is a class invariant.

**Proposition 7.4** [91, §127, p. 472]. Assume the same conditions as above and also that

\[
(26) \quad C^2 + CB - 1 \equiv 0 \text{ or } 8 \mod 16.
\]

Then \( \sqrt{2}f^3(z) \) is a class invariant.

Note that (26) implies that \( B \) is divisible by \( 8 \). Suppose now that all the preceding conditions are satisfied for \( (A, 2B, C) \). Since \( AC \equiv D \equiv -1 \mod 8 \), we see that \( A \equiv -C \mod 8 \), and therefore \( (2/A) = (2/C) \). Assume first that \( (2/C) = 1 \). Then \( C \equiv \pm 1 \mod 8 \), and if \( B \equiv 0 \mod 16 \), then

\[
C^2 + CB - 1 \equiv 0 \mod 16.
\]

The case \( (2/C) = -1 \) is treated in the same way. We then write

\[
\frac{f(z)}{\sqrt{2}} = \frac{1}{4} \left( \frac{\sqrt{2}f^3}{f^8} \right).
\]
The other cases are dealt with using results from the same section of Weber's book.

We now briefly describe the algorithm needed to compute \( H_{4D}[f/\sqrt{2}](X) \) for \( D \equiv 7 \mod 8 \).

**procedure** WEBER7(a, 2b, c)

1. if \( a \) is even, replace \((a, 2b, c)\) with \((c, -2b, a)\);
2. put \( \xi(a) = 4((2/a) - 1) \mod 16 \);
3. if \( a \not\equiv 0 \mod 3 \), choose \( k \) such that \( B \equiv b + ak \equiv 0 \mod 3 \) and \( B \equiv \xi(a) \mod 16 \); then \((a, 2b + 2ak, c + bk + ak^2)\) satisfies one of the conditions of Theorem 7.3;
4. if \( a \equiv 0 \mod 3 \), but \( b \not\equiv 0 \mod 3 \), then find \( k \) such that \( C = c + 2bk + ak^2 \equiv 0 \mod 3 \) and \( b + ak \equiv \xi(a) \mod 16 \);
5. if \( z_0 \) (resp. \( z \)) is associated with \((a, 2b, c)\) (resp. \((A, 2B, C)\)), then \( z = z_0 - k \) and \( f(z) = \xi_{48}^{k}f(z_0) \) (resp. \( f(z) = \xi_{48}^{k}f_{1}(z_0) \)) if \( k \) is even (resp. odd), using (15).

As an example, we find

\[
H_{4 \times 23}[f/\sqrt{2}](X) = X^3 - X - 1.
\]

From this, it is easy to compute \( j(\omega) \) for \( \omega = (-1 + \sqrt{-23})/2 \) via \( f_2(\omega) = \sqrt{2}\xi_{48}/f(\sqrt{-23}) \) (see [91, §34, (19)]).

Other cases yield the same kind of algorithms.

### 7.2.1. Alternative class invariants.

The second author is indebted to J.-F. Mestre who explained the following [59]. Let \( s \) be a prime positive integer and \( X_0(s) \) be the modular curve [73]. It can be shown that (see, for example, [37] or [58]), when \( X_0(s) \) is of genus 0 (i.e., \( s = 2, 3, 5, 7, 13 \)), it can be parametrized by

\[
x_s(z) = \left( \frac{\eta(z)}{\eta(sz)} \right)^{24/(s-1)}.
\]

The modular invariant \( j \) is related to \( x_s \) via the following formulae:

\[
\\begin{align*}
j &= \frac{(x_2 + 16)^3}{x_2} = \frac{(x_3 + 27)(x_3 + 3)^3}{x_3} = \frac{(x_5^2 + 10x_5 + 5)^3}{x_5} \\
&= \frac{(x_7^2 + 13x_7 + 49)(x_7^2 + 5x_7 + 1)^3}{x_7} \\
&= \frac{(x_{13}^4 + 5x_{13} + 13)(x_{13}^4 + 7x_{13}^2 + 20x_{13}^2 + 19x_{13} + 1)^3}{x_{13}}.
\end{align*}
\]

**Theorem 7.4.** Let \( -D \) be a fundamental discriminant and \( s \in \{3, 5, 7, 13\} \) such that \( (-D/s) = 1 \). Let \( (s) = ss' \) in \( K = \mathbb{Q}(\sqrt{-D}) \). Let \( i \) be a reduced ideal. Suppose we have found a basis \((e_1, e_2)\) of \( i \) such that \( s \times i = (e_1, se_2) \). If we put \( \tau = -e_2/e_1 \), the number \( u_s(i) = x_s(\tau) + s^{12/(s-1)}x_s(\tau) \) is a class invariant.

For example, let \( D = 23 \) and \( s = 13 \). We have \( s = (13, 8 + \omega) \). There are three reduced ideals. In the following table, we give the values of these ideals, the values of the reduced ideals \( s \times i \) and the values of \( u_{13} \).
We remark that, as soon as \( i = (a, b + \omega) \) and \( s \times i = (u, v + \omega) \), then \((e_1, e_2)\) is precisely \((v + \omega, a)\), since \( u = as \).

Finally, we get
\[
\prod_{i}(X - u_{13}(i)) = X^3 + 16.99999999X^2 + 97.99999994X + 190.9999999,
\]
and the minimal polynomial of \( u_{13} \) is \( H_{23}[u_{13}](X) = X^3 + 17X^2 + 98X + 191 \).

It is easily seen that in this case the minimal polynomial of \( x_{13} \) is
\[
H_{23}[x_{13}](X) = \left( X^3 + \frac{17}{2}X^2 + \frac{59}{2}X + 37 \right)^2 + 23 \left( \frac{X^2}{2} + \frac{5X}{2} + 6 \right)^2,
\]
so that \( x_{13} \) and hence \( j \) could be found by solving a cubic equation modulo \( p \) (recall that \( \sqrt{-23 \mod p} \) will already have been found). The same situation arises in all cases given by Theorem 7.4.

With more effort one can also use values of \( s \) for which \( \Gamma_0(s) \) does not have genus 0.

7.2.2. Remarks. A naive approach to the computation of \( W_D \) is to use polynomial factorization, or the LLL algorithm [51].

One of the phases of ECPP is to factor the polynomials \( H_D \) over \( \mathbb{Z}/p\mathbb{Z} \). This can be expensive, since for a fixed large \( p \) the complexity of such computations is basically proportional to the square of the degree of the polynomial (see §8.6.1): This explains why we discard the case \( D \equiv 3 \mod 8 \), since in this case, we might work on polynomials of degree \( 3h \).

We shall see in the following section how this computation can be simplified by factoring these equations over the genus field of \( K \). In order to simplify the notation, we will refer to \( \mathbb{W}_D(X) \) as the defining polynomial of \( K_H \) corresponding to whichever \( H_D[ \cdot ] \) we can use. We call \( \mathbb{W}_D \) a Weber polynomial associated with \(-D\).

Let us end this subsection by summarizing the strategy for computing \( \mathbb{W}_D \), given \( D \).

procedure Weber(D)

1. if \( D \not\equiv 0 \mod 3 \) and \( D \not\equiv 3 \mod 8 \) then
   1. if \( D \equiv 7 \mod 8 \) then \( \mathbb{W}_D = H_D[f/\sqrt{2}] \);
   2. if \( D/4 \equiv \pm 2 \mod 8 \) then \( \mathbb{W}_D = H_D[f_1/\sqrt{2}] \);
   3. if \( D/4 \equiv 5 \mod 8 \) then \( \mathbb{W}_D = H_D[f^4] \);
   4. if \( D/4 \equiv 1 \mod 8 \) then \( \mathbb{W}_D = H_D[f^2/\sqrt{2}] \);
2. if there exists \( s \) in \{3, 5, 7, 13\} such that \((-D/s) = 1\) then \( \mathbb{W}_D = H_D[x_s] \);
3. if \( D \equiv 3 \mod 6 \) then \( \mathbb{W}_D = H_D[\sqrt{-D/3}] \);
4. otherwise take \( \mathbb{W}_D = H_D \).
7.3. Factoring the equations over the genus field. The aim of this subsection is to explain how it is possible to factor our \( W_D \)'s over \( K_\varphi \). We will show that \( W_D \) has exactly \( g \) factors, each of degree \( e = h/g \), with coefficients in \( K_\varphi \). This reduces the time needed to compute a root of \( W_D \mod p \) for large \( p \), since we have to find a root of degree \( e \) instead of \( h \).

\[
\begin{align*}
K_H & | \quad e = h/g \\
K_\varphi & | \quad g \\
K &
\end{align*}
\]

We first give some properties of composite quadratic fields, including the computation of an integral basis. Then, we set up an ordering on the genera of \( \mathcal{H}(-D) \) through the action of the Galois group of \( K_\varphi/K \). After proving the preceding results, we detail our algorithm and give some examples.

7.3.1. Some properties of composite quadratic fields. Let \( u_1, \ldots, u_n \) be \( n \) squarefree multiplicatively independent elements of \( \mathbb{Z} \). Suppose, moreover, that they are multiplicatively independent (i.e., \( u_1^{a_1} \times \cdots \times u_n^{a_n} = 1 \) is possible for some integers \( a_i \)'s if and only if the \( a_i \)'s are all zero). We put \( k_n = \mathbb{Q}(\sqrt{u_1}, \ldots, \sqrt{u_n}) \) and \( g = 2^n \). Following [23], we introduce the sequence \( \{A_i\}_{0 \leq i < g} \) defined by

\[
A_0 = 1, \\
A_j = \begin{cases} 
  u_{k+1} & \text{if } j = 2k, \\
  A_{2k-1}A_i/\gcd(A_{2k-1}, A_i)^2 & \text{if } j = 2k-1 + i \text{ and } 0 < i < 2k-1.
\end{cases}
\]

We also define \( \alpha_i = \sqrt{A_i} \). Then \( \{1, \alpha_1, \ldots, \alpha_{g-1}\} \) is a basis for \( k_n/\mathbb{Q} \).

**Proposition 7.5** [23]. The integers of \( k_n \) are necessarily of the form

\[
x = \frac{1}{2^n} \sum_{i=0}^{g-1} P_i \alpha_i,
\]

where the \( P_i \)'s are rational integers of the same parity, and all even if there is an \( i \) in \( \{0, \ldots, g-1\} \) such that \( A_i \equiv 1 \mod 4 \).

7.3.2. Computations in \( K_\varphi/K \). As in §2, we write \( -D = q_1^* \cdots q_l^* \), where \( q^* = (-1)^{(q-1)/2}q \) if \( q \) is an odd prime and \(-4 \) or \( \pm 8 \) otherwise. The \( q_i \)'s are supposed to be ordered as follows: if \( D \equiv 0 \mod 4 \), then \( q_1 = 4 \) or \( 8 \). Then the \( q_i \)'s with \( q_1^* = q \) are listed in increasing order and finally the \( q_i \)'s with \( q_1^* = -q \), also in increasing order. Then \( l \) is the number of positive \( q_i^* \)'s:

\[
-D = q_1 \cdots q_l(-q_{l+1}) \cdots (-q_l).
\]

The genus field \( K_\varphi = \mathbb{Q}(\sqrt{q_1^*}, \ldots, \sqrt{q_l^*}) \) can be described as

\[
K_\varphi = K(\sqrt{u_1}, \ldots, \sqrt{u_{l-1}}),
\]

where

\[
u_i = \begin{cases} 
  q_i & \text{for } 1 \leq i \leq l, \\
  q_iq_i & \text{for } 1 < i < l.
\end{cases}
\]
The Galois group of $K_g/K$ is $\Sigma_G = \langle \varphi_1, \ldots, \varphi_{t-1} \rangle$, where

$$\varphi_i(\sqrt{u_j}) = \begin{cases} -\sqrt{u_i} & \text{if } j = i, \\ \sqrt{u_j} & \text{if } j \neq i. \end{cases}$$

Hence, $\Sigma_G$ is isomorphic to $(\mathbb{Z}/2\mathbb{Z})^{t-1}$, and we can represent an element $\varphi$ of $\Sigma_G$ by a $(t-1)$-tuple of signs (i.e., elements of $\{\pm 1\}$). We decide to use the following ordering of the $\varphi_i$. If $i$ is an integer between 0 and $2^{t-1} - 1$, we can write $i = \sum_{s=0}^{t-1} \nu_s 2^s$ ($\nu_s \in \{0, 1\}$) and we take

$$\varphi_i = \varphi_{\nu_1} \circ \cdots \circ \varphi_{\nu_{t-1}}^{-1}.$$  

We represent $\varphi_i$ by $(e_1, \ldots, e_{t-1})$, where $e_s = 2\nu_s - 1$.

With this ordering, the $i$th conjugate of an integer $\theta$ of $K_g$ is $\theta(i) = \varphi_i(\theta)$. 

7.3.3. Ordering the genera. We show how to express the $\varphi_i$’s in terms of $\mathcal{A}_G$, as described in §2. Let us write $\varphi_i = (e_1, \ldots, e_{t-1})$ and $\mathcal{A}_G = (e_1, \ldots, e_t)$. What we have to solve is the system

$$\begin{cases} e_1 = e_1, \\ \vdots \\ e_l = e_l, \\ e_{l+1}e_l = e_{l+1}, \\ \vdots \\ e_{l-1}e_l = e_{l-1}. \end{cases}$$  

We compute

$$\prod_{i=1}^{t-1} e_i = \left(\prod_{i=1}^{t-1} e_i \right) e_{t-l-1}.$$  

With (1), we can simplify

$$\prod_{i=1}^{t-1} e_i = \prod_{i=1}^{t-1} e_i = e_t.$$  

The solution of the system (30) is thus

$$e_i = \begin{cases} e_1 & \text{for } 1 \leq i \leq l, \\ e_l & \text{if } i = t, \\ e_se_i & \text{for } l < i < t. \end{cases}$$  

We take the ordering on the genera to be that induced by the preceding process. Let us give an example. Suppose that $-D = -308 = (-7) \times (-11) \times (-4)$. We take $u_1 = (-1) \times (-7)$ and $u_2 = (-1) \times (-11)$. The $\varphi_i$’s and the associated genera are given below.

<table>
<thead>
<tr>
<th>$i$</th>
<th>$\varphi_i$</th>
<th>$G_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$(+, +)$</td>
<td>$(+, +, +)$</td>
</tr>
<tr>
<td>1</td>
<td>$(-, +)$</td>
<td>$(+, -, -)$</td>
</tr>
<tr>
<td>2</td>
<td>$(+, -)$</td>
<td>$(-, +, -)$</td>
</tr>
<tr>
<td>3</td>
<td>$(-, -)$</td>
<td>$(-, -, +)$</td>
</tr>
</tbody>
</table>
It should be noted that the genus associated with \( \varphi_0 \) is always \( G_0 \), the principal genus. Moreover, the ordering on the \( \varphi_i \)'s depends only on \( g \) and not on \( D \), whereas the correspondence with the genera depends on \( D \) and \( l \). With each pair \( (t, l) \) satisfying \( l \equiv t - 1 \mod 2 \), we associate the generic ordering defined by the above process. The example given above is the generic ordering \( (3, 0) \).

We end this subsection by introducing

\[
J_i = J(G_i) = \{ j(C), C \in G_i \} = \{ j_{i1}, \ldots, j_{ie} \}, \quad 0 \leq i < g,
\]

and

\[
\mathcal{W}_D^{(i)}(X) = P(J_i) = P(G_i) = \prod_{r=1}^{e} (X - j_{ir}).
\]

We remark that \( \mathcal{W}_D = \prod \mathcal{W}_D^{(i)} \) and that each \( \mathcal{W}_D^{(i)} \) has only real coefficients, since two conjugate \( j \)'s are in the same \( J \).

The following fundamental theorem is now an easy consequence of Galois theory.

**Theorem 7.5.** For all \( i \), \( \mathcal{W}_D^{(i)}(X) \) is in \( \mathbb{K}_g[X] \).

We also have

**Corollary 7.1.** For all \( i \), \( \varphi_i(\mathcal{W}_D^{(0)}) = \mathcal{W}_D^{(i)} \).

This motivates our choice of the ordering on the \( G \)'s, since otherwise we would have to justify that the \( \varphi \)'s permute the \( \mathcal{W}_i \)'s.

This result yields an algorithm for computing the expression of \( \mathcal{W}_D^{(0)} \) over \( \mathbb{K}_g \). We describe this algorithm in the next section.

**7.3.4. Description of the algorithm.** The preceding results make it clear that the critical parameters are \( h \) and \( g \); the algorithm does not depend explicitly on \( -D \). Our purpose is now to explain how we can compute the coefficients of \( \mathcal{W}_D^{(0)} \) and to exemplify the use of symbolic manipulation in the process.

We are looking for the coefficients of the polynomial \( \mathcal{W}_D^{(0)}(X) \), which is a factor of \( \mathcal{W}_D \) over \( \mathbb{K}_g \). We shall write \( \mathcal{W}_i \) for \( \mathcal{W}_D^{(i)} \) since there is no ambiguity. In fact, since the coefficients of \( \mathcal{W}_0 \) are real, we can work over \( k_{l-1} \) as defined above. The results are still valid by using the canonical isomorphism between the Galois groups of \( \mathbb{K}_g/\mathbb{K} \) and \( k_{l-1}/\mathbb{Q} \).

We write

\[
\mathcal{W}_0(X) = X^e + \sum_{r=0}^{e-1} \left( \sum_{s=0}^{g-1} a_{sr} \alpha_s \right) X^r,
\]

where all the \( a_{sr} \) are in \((1/g)\mathbb{Z}\) and the \( \alpha_s \)'s as in §7.3.1. We will find these coefficients by means of the resolution of a linear system. Let \( \alpha_s^{(i)} = \varphi_i(\alpha_s) \).

For any polynomial \( Q(X) \), let \([X^r]Q\) denote the coefficient of degree \( r \) of \( Q \). Then

\[
\sum_{s=0}^{g-1} a_{sr} \alpha_s = [X^r]\mathcal{W}_0.
\]

License or copyright restrictions may apply to redistribution; see http://www.ams.org/journal-terms-of-use
Suppose now that \( r \) is fixed, \( 0 \leq r \leq e - 1 \). If we apply \( \varphi_i \) to (34), we find

\[
\sum_{s=0}^{g-1} a_{sr} \alpha_s^{(i)} = [X'] \mathcal{W}_i.
\]

We do the same thing for \( i = 0, \ldots, g - 1 \), and we see that \( (a_{sr})_{0 \leq s < g} \) is the solution of the linear system

\[
\begin{aligned}
x_0 + x_1 \alpha_1^{(0)} + \cdots + x_{g-1} \alpha_{g-1}^{(0)} &= Y_0, \\
x_0 + x_1 \alpha_1^{(1)} + \cdots + x_{g-1} \alpha_{g-1}^{(1)} &= Y_1, \\
&\vdots \\
x_0 + x_1 \alpha_1^{(g-1)} + \cdots + x_{g-1} \alpha_{g-1}^{(g-1)} &= Y_{g-1},
\end{aligned}
\]

where we replace \( Y_i \) by \([X'] \mathcal{W}_i\). We call the preceding system the **generic system of order** \( g \), since it depends only on \( g \). We see that we have just to solve this system once for each different value of \( g \), computing all the \( a_{sr} \)'s by replacing the values of the \( \alpha \)'s by their corresponding floating-point approximations.

From a practical point of view, we compute an approximation to \( ga_{sr} \), take the nearest integer, and then divide out by the same \( g \). When we have computed our \( \mathcal{W}_0 \), we compute \( L \), the lcm of the denominators of the coefficients and we store the coefficients of \( L \mathcal{W}_0 \).

As an example, let us treat the case of \(-D = -308 = (-7) \times (-11) \times (-4)\).

The generic system of order 4 is

\[
\begin{aligned}
x_0 + x_1 \alpha_1^{(0)} + x_2 \alpha_2^{(0)} + x_3 \alpha_3^{(0)} &= Y_0, \\
x_0 + x_1 \alpha_1^{(1)} + x_2 \alpha_2^{(1)} + x_3 \alpha_3^{(1)} &= Y_1, \\
x_0 + x_1 \alpha_1^{(2)} + x_2 \alpha_2^{(2)} + x_3 \alpha_3^{(2)} &= Y_2, \\
x_0 + x_1 \alpha_1^{(3)} + x_2 \alpha_2^{(3)} + x_3 \alpha_3^{(3)} &= Y_3,
\end{aligned}
\]

where

\[
\begin{aligned}
\alpha_1^{(0)} &= \sqrt{u_1}, \\
\alpha_2^{(0)} &= \sqrt{u_2}, \\
\alpha_3^{(0)} &= \sqrt{u_1 u_2}.
\end{aligned}
\]

The generic ordering for \( D = 308 \) is \((3, 0)\) and was given in §7.3.3.

We want to get the expression of \( H_{308}[\gamma_2]^{(0)}(X) \) over \( K_{/\mathcal{W}} \). We have

\[
H_{308}[\gamma_2](X) = X^8 - 95835320X^7 - 923879753200X^6 + 12151678024000X^5 - 195287646706560000X^4 - 162741620553600000X^3 + 35433687468608000000X^2 + 136128371025152000000X - 12937041027046400000000.
\]

Suppose that we have built the sets of roots of \( H_{308}[\gamma_2] \) according to the genera. We have in this case

\[
\begin{aligned}
J_1 &= J(+, +, +) = \{880456353882407955305050.260304, 797.592915355\}, \\
J_2 &= J(+, -, -) = \{5648.96421088 \pm 8460.8161800511i\}, \\
J_3 &= J(-, +, -) = \{3456.226641, -938326357130.70446379\}, \\
J_4 &= J(-, -, +) = \{-47921735.6519096497 \pm 83004169.578235232i\}.
\end{aligned}
\]
Finally, we obtain
\[ H_{308}[\gamma_2]\text{\textsuperscript{0}}(X) = X^2 + (-23958830 - 9057440\alpha_1 - 7223840\alpha_2 - 2730910\alpha_3)X \\
+ 222228600 + 84022400\alpha_1 + 66972800\alpha_2 + 25321800\alpha_3. \]

Numerous additional checks on the accuracy of our calculations are available, using the supersingular equation. For example, \( j = 0 \) is the only supersingular value modulo 5, so that for \( (-D/5) = -1 \) all the roots of \( H_D \) must be zero modulo 5, as exemplified above.

8. IMPLEMENTATION DETAILS

8.1. Machines and languages. The algorithm as described in detail in this paper has been implemented by the second author on a SUN 3/60, using Le.Lisp and the arithmetic described in [45].

The first author implemented the main ideas in the spring of 1986, using an IBM 3081 and his procedure LMA4064V. Most of the general-purpose number-theoretic routines were already available and 95% efficient, using a combination of FORTRAN and ASSEMBLER. However, he did not at that time have his (subsequently written) arbitrary-precision complex floating-point routines, and was thus confined in the computation of the \( H_D \) to IBM quadruple precision and some casual ingenuity. With a list of only 119 discriminants he was compelled to factorize the numbers of points excessively at great cost for large inputs. However, the largest remaining prp343 in the Cunningham tables was done in 2.5 hours, and 250-digit numbers routinely in 3 to 8 minutes.

8.2. Strategies.

8.2.1. Architecture of the program. The first basic approach is the Factor and Prove Strategy (FPS), following the direct application of the procedure ECPP. In other words, as soon as we have found a probably factored number, we immediately verify the conditions of the corresponding theorem. This idea works fine with small numbers (less than \( 10^{300} \), say) since we are almost sure to find a good candidate among our list of \( D \)'s. However, for large \( N \)'s, our finite lists of \( D \)'s can be too short and sometimes we are forced to backtrack in our sequence of intermediate primes.

The preferred one is the Factor All Strategy (FAS) which first builds the sequence of intermediate primes and then proves all the theorems. This enables backtracking, as well as a more rational distributed algorithm (see [67]).

8.2.2. Philosophy. We constantly use some principles:

1. the tests with \( N \pm 1 \) are treated as a particular case of the elliptic curve test;
2. it is understood that, if a probable prime is later proved composite, then the program immediately returns to the preceding place in the DOWNRUN or exits if we were at the top. This of course involves the possibility of backtracking inside the program.

8.2.3. Computing \( W_D(X) \). In the proving part of our algorithm, we must compute \( W_D(X) \) in order to find a zero modulo \( p \). There are two strategies. The first one is to precompute a list of \( W_D(X) \) for a subset of \( \mathcal{D} \) and store them in a file. The other is to compute \( W_D(X) \) on the fly, as required by the
factoring part of the algorithm. It is clearly impossible to store all the $W_D$ for all $D$’s, and thus we mix the two ideas. We have computed $W_D$ for all $D$ with $h(-D) \leq 20$ and stored them. This makes about 1.5 Mbytes (on a SUN 3/60). If necessary, other polynomials can be computed and introduced in the program. The actual computation of $W_D$ is done by means of a MAPLE program. If one has the desired complex multiprecision arithmetic, one can of course merge the two programs.

We have computed all $W_D(X)$ for all (known) $D$ such that $h < 20$ and for $(h, g) \in \{(32, 16), (24, 8), (48, 16), (32, 8), (64, 16)\}$. This yields 4500 potential numbers of points for each probable prime in our DOWNRUN. These are made up of 2 for $\pm 1$, 6 for $-3$, 4 for $-4$ and 2 each for the remaining 2244 discriminants.

8.2.4. Ordering the data. We decide to use only the $D$ less than $10^6$ with $h(-D) \leq 50$. We remark that there are two parts in deciding whether $p$ is a norm in $\mathbb{Q}(\sqrt{-D})$ or not. The first is checking that $p \in G_0(-D)$: this is easy because we have only Jacobi symbols to compute. At this point, $p$ is represented by $F_D$ with probability $g(-D)/h(-D)$, but to be certain, we must find a square root of $-D \mod p$ and in effect reduce a quadratic form. So, we store our $D$’s in increasing order with respect to $(h/g, h, D)$. The most interesting discriminants are those with $h = g$, which are called idoneal numbers: Under the assumption of the Extended Riemann Hypothesis, there are 65 of them (see [35, 24]).

8.3. Logistics and tactics. Many of the routines we use are explained and codified in [27]. We mention here one or two additional points.

8.3.1. Multiprecision. It is obvious that we need the fastest algorithms possible, especially a good routine for finding gcd’s and multiplicative inversions. Also, the size of numbers we are currently tackling (more than twenty 32-bit words) makes it worth using Karatsuba’s algorithm. We refer to [52] for all this. We add below some remarks which may be well known, but not easily found in the literature.

We can use a special routine for squaring based on the following (trivial) observation. Let $m = \sum_{i=0}^{l-1} m_i B^i$ be an integer written in base $B$. Then

$$m^2 = \sum_{i=0}^{l-1} m_i^2 B^{2i} + 2 \sum_{i=1}^{l-2} m_i m_i' \sum_{j=i+1}^{l-1} m_j B^i.$$

This yields an algorithm for squaring that is asymptotically twice as fast as (ordinary standard) multiplication. In order to speed up things, it is necessary to program it directly in assembly in order to minimize overhead.

With this idea, we can replace multiplication by

$$ab = \frac{(a + b)^2 - (a - b)^2}{4} = \frac{(a + b)^2 - a^2 - b^2}{2},$$

both formulae being useful, the latter one in the case where we must multiply many $a$’s by the same $b$. Another application is given below.

8.3.2. Exponentiation over various rings. We use the exponentiation-by-blocks method as described in [27] for $\mathbb{Z}/p\mathbb{Z}$ ($= \text{GF}(p)$), $\text{GF}(p^2)$ ($N + 1$ primality
test), and for elliptic curves. The optimal value for the size of the block was
determined empirically. The value of $2^6$ seems to be the right one for almost
all values of $N$.

When using Berlekamp’s algorithm (as well as Girstmair’s ideas, see below),
we have to compute $P(z)^t \mod(p, f(z))$ for a fixed monic $f(z)$. Write
$f(z) = z^d + f_{d-1}z^{d-1} + \cdots + f_0$. We precompute

$$F^{(i)} = z^i \mod(p, f(z)) = F^0 + \cdots + F^0$$

$0 \leq i \leq 2d - 2$.

The basic operation we have to perform is the multiplication of $P(z) =
p_{d-1}z^{d-1} + p_{d-2}z^{d-2} + \cdots + p_0$ by $Q(z) = q_{d-1}z^{d-1} + q_{d-2}z^{d-2} + \cdots + q_0$. We have

$$P(z) \times Q(z) \mod(p, f(z)) = \sum_{i=0}^{d-1} \sum_{j=0}^{d-1} p_i q_j z^{i+j} = \sum_{i} \sum_{j} p_i q_j F^{(i+j)}.$$

The evaluation of $p_i q_j$ is then done using the multiply-by-squaring method
described above. If we precompute the $p_i^2$ and the $q_j^2$ (modulo $p$), we reduce
the cost of computing $P \times Q$ to

$$\sum_i T_{sq}(p_i) + \sum_j T_{sq}(q_j) + \sum_{i,j} T_{sq}(p_i + q_j),$$

which is basically $(d^2 + 2d)T_{sq}$ compared to $d^2T_x$. The gain is thus

$$\frac{d^2 + 2d T_{sq}}{d^2 T_x} \approx \frac{1}{2} \left(1 + \frac{2}{d}\right).$$

The most obvious gain is when we have to compute the square of a polynomial.
The cost of it is now $(d^2 + d)T_{sq}$.

8.4. Finding a good $D$. We have decided to consider the $N \pm 1$ test as a special
case corresponding to a fictitious $D = \pm 1$. These tests have been well studied,
and many tricks are known to speed them up. In particular, we prefer the
description of [27] since we can apply very easily the exponentiation-by-blocks
method when working directly over $\text{GF}(p^2)$, but not with Lucas sequences.
We make here the remark that we use a trick of [21] to reduce the number of
computations needed when one of our $N \pm 1$ has many factors (this is also
valid for the elliptic case).

8.4.1. Looking for a splitting $D$. In the general case, we are looking for a
fundamental discriminant $-D$ for which our probable prime $N$ is a norm. The
first thing we do is to check that $N \in G_0(-D)$. This is done by computing the
Jacobi symbols $\chi_i(N) = (q_i^\ast /N)$ in our notation. If all these symbols are equal
to $+1$, we proceed to the second phase, that is, computing a representation of
$N$ by $F_D$.

Though the computation of Jacobi symbols is very cheap, one can arrange
the $D$’s in such a way that if $N$ is a nonresidue modulo 3 (say), then we only
look at those $D$ which are not divisible by 3. In the same way, we can store
the values of $(N/q)$ for some small primes $q$ (typically $q < 100$) so as not to
recompute the same objects.
8.4.2. **Solving** \( p = N_K(\pi) \). We want to get the representation of a prime \( p \) as a norm in \( K = \mathbb{Q}(\sqrt{-D}) \). Equivalently, we must solve

\[
4p = A^2 + DB^2
\]

with \( A \) and \( B \) in \( \mathbb{Z} \). We can solve this problem using Shanks's algorithm [86] or lattice reduction [89]. These two algorithms are basically the same and solve the general case of representation of a prime number by a given quadratic form. In the case where we want to represent \( p \) by the principal form only, one can do slightly better, using the work of Cornacchia.

We first make some remarks. If \( D \equiv 0 \) mod 4, one puts \( D = 4d \), and we have to solve \( p = A^2 + dB^2 \). If \( D \equiv -1 \) mod 8, then

\[
4p = A^2 + (8d - 1)B^2 \Rightarrow A^2 - B^2 \equiv 4 \text{ mod } 8,
\]

which is possible if \( A^2 \equiv 0 \) mod 8 and \( B^2 \equiv 4 \) mod 8 and in particular \( A \) and \( B \) even. So we actually solve \( p = A'^2 + (8d - 1)B'^2 \). We can say nothing when \( D \equiv 3 \) mod 8.

Cornacchia's algorithm [32] finds the representation of \( p = u^2 + dv^2 \) whenever one exists, with \( (p, u, v) = 1 \). A proof of its validity can be found in [70]. It runs like this:

**procedure** CORNACCHIA(u, v, d, p);

(* solution of \( u^2 + dv^2 = p \*)

1. let \( x_0 \) be a solution of \( x^2 \equiv -d \text{ mod } p \) that satisfies \( p > x_0 > p/2 \);
2. develop \( p/x_0 \) as a continued fraction:
   
   \[
   p = q_0x_0 + x_1, \\
   x_0 = q_1x_1 + x_2, \\
   \dotsc \\
   x_r = q_{r+1}x_{r+1} + x_{r+2}
   \]

   and stop when \( x_r < p \leq x_{r-1}^2 \);
3. put
   
   \[
   u = x_r \quad \text{and} \quad v = \sqrt{\frac{p - x_r^2}{d}}.
   \]

4. if \( v \) is not an integer, \( p \) is not representable as \( u^2 + dv^2 \).

In the case \( D \equiv 3 \) mod 8, we can use the same algorithm, using for \( x_0 \) a solution of \( x^2 + x + \frac{D+1}{4} \text{ mod } p \).

8.4.3. **Extracting a square root modulo** \( N \). In using the above procedure, we have to compute square roots modulo \( N \). We can benefit from some previous computations as follows.

If we use Shanks's algorithm, we need a \( z \) such that \( z^{2k-1} \equiv -1 \text{ mod } N \), where \( N = 1 + 2^k \times b \), \( b \) odd. We can obtain \( z \) as a byproduct of the pseudoprimitivity test for \( N \) as follows: first, find \( a \) such that \((a/N) = -1\) (if after 50 trials, we have not found one, maybe \( N \) is a square). Then, compute \( z = a^b \): If \( z^{2k-1} \not\equiv -1 \text{ mod } N \), then \( N \) is composite. Otherwise, \( N \) is a probable prime, and we can use \( z \) in Shanks's algorithm. It should be noted that succeeding in computing a square root with this algorithm is almost a guarantee that we have a prime. In other words, if a composite number passed
the pseudoprimality test, it is very unlikely to pass this step (see [94] for the combination of square roots with primality tests).

In many cases, a number of square roots can be found very cheaply at this first stage. If \( N \equiv 1 \mod q \) for small odd \( q \), we can usually find the square root of \((-1/q)q\); while if \( N \equiv 1 \mod 8 \), we find both \( \sqrt{-1} \) and \( \sqrt{-2} \).

Also, it is possible to accumulate the square roots that we have to compute until we find a suitable \( D \). Hence, we can store \( \sqrt{q^2 \mod N} \) for some small primes \( q \). After that, we can use these values in the computation of \( j \mod N \) when \( \mathcal{O}_D \) splits over an intermediate quadratic field of discriminant \( q^* \). We also order our \( D \)'s in such a way that we compute less square roots modulo \( p \). For instance, we try \( D = 3, D = 4 \), then \( D = 15 \) by computing only \( r_5 = \sqrt{5} \mod p \), \( D = 20 \) by combining \( D = 4 \) and \( r_5 \), etc.

8.5. Factorizing the number of points.

8.5.1. Finding all factors of an integer \( m \) which are less than \( B \). It is well known that the general problem of getting all factors of an arbitrarily large number is very difficult (see [55]). However, the problem of getting small factors of a number \( m \) is a little better understood.

What we want is an algorithm that can find small factors of a number in a reasonable amount of time. Apart from trial division that is routinely used to find all factors less than \( 10^6 \), the two best candidates are Pollard’s \( \rho \) method [62] and the ECM method of Lenstra [57]. According to [17], it seems that the first one is worth using for finding factors less than \( 10^8 \), and the second for factors from \( 10^{10} \) to \( 10^{15} \) using various speedups [62, 7].

However, the very best value among these probabilistic factoring methods is given by Pollard’s \( p - 1 \), even though this can only be used once.

It should be noted that we do not store the intermediate factors found, only their product. This is motivated by the fact that we do not need to have the exact factorization of \( m \) (unless \( m \) is small). It can happen that a 20-digit factor of a 1000-digit number is not prime, but we are only interested in having a 980-digit probable prime.

We detail the choices we made in the following section.

8.5.2. Sieving with small primes. We begin by looking for small prime factors of \( m \). Let \( p_1, \ldots, p_k \) be all the prime numbers less than a given \( B \). We suppose they are stored in a file. We extend a method already used in [18, §7, Remark 1] and [27]. We first compute the quantities

\[
 r_i = (N + 1) \mod p_i \quad \text{for } i = 1, \ldots, k.
\]

Divisibility of \( N + 1 \) is then tested as follows. If \( r_i = 0 \), then \( N + 1 \) is divisible by \( p_i \), and if \( r_i = 2 \), then \( p_i \) divides \( N - 1 \). In the case of testing the divisibility of \( m_i = N + 1 \pm t \), with \( |t| \leq 2\sqrt{N} \), we first compute \( r = t \mod p_i \) and check whether \( r \equiv \pm r_i \mod p_i \). We replaced \( 2k \) divisions of numbers of size \( L \) with \( k \) divisions of integers of size \( L/2 \).

However, there is yet a further factor of 2 to be gained. With \( 4p = A^2 + DB^2 \) we have \( 4m = (A \pm 2)^2 + DB^2 \). For any sieving prime \( p_i > 2 \) with \( (-D/p_i) = -1 \), we have \( p_i | m \Rightarrow p_i | B^2 \) and \( p_i | (A \pm 2)^2 \), respectively. Thus, we first form \( \gcd(A + 2, B) \) and \( \gcd(A - 2, B) \) and remove the common factors from \( m \); subsequently, only primes \( p_i \) with \( (-D/p_i) = 1 \) are used in the sieve (and recognized from a lookup table modulo \( D \) or \( 4D \)).
8.5.3. **Pollard's \(\rho\).** From [17], it is reasonable to find all factors less than \(10^8\) with this method. Using the ideas of [62], we decide to make \(10^5\) iterations of this method. We accumulate the iterates of the function and do only two gcd's.

8.5.4. **ECM.** We use the algorithm as described in [17] with the parametrizations of [62, 7] for having curves with some prescribed small divisors. One of the major problems is the storage that is prohibitive when dealing with 1000-digit numbers. This explains why the second stage is not performed on numbers of size greater than \(10^{700}\).

8.5.5. **Pollard's \(p - 1\).** We note that this is reasonable when testing the Cunningham numbers which often have the property of being congruent to \(\pm 1\) modulo some large known prime integer. So we can spend a little time to see if we can get a factor (possibly large) of \(m\) this way.

8.5.6. **Further improvements.** The best discriminants are those for which \(h = 1\), because \(j\) is easy to compute and \(E\) is easy to find (see next section). Hence, we decide to use more factoring power on them. Say we multiply all factorization parameters by 1.2, maybe with all possible methods as well. However, in order for them to appear in the DOWNRUN, we must find a suitable number of points.

Suppose we test \(N_i\) and we get a candidate \(N'\) for \(N_{i+1}\). First of all, we can impose an upper bound on \(N'\). More precisely, we want to go down in our sequence of primes as fast as possible. Therefore, we decide to reject all possible \(N'\) such that \(N_i/N' < 10^{\min x}\), say. The exact value of \(\min x\) is best found by experiments. This results in many different strategies, which we do not discuss here.

We also try to have a next candidate that is as promising as possible. If we find an \(N'\) which is congruent to 1 modulo 3 or 4, we take it. On the contrary, when \(N' \equiv -1 \mod 24\), we prefer to try another one; the two strategies can be combined.

8.6. **Finding \(j(E) \mod p\) and a point on \(E(\mathbb{Z}/p\mathbb{Z})\).** The process is the following: first compute \(j(E)\), a root of \(H_D(X) \equiv 0 \mod p\), then find the equation of \(E\) and a point on \(E\). In fact, we compute a root of \(\mathcal{H}_D(X) \equiv 0 \mod p\) and we compute \(j\).

8.6.1. **Solving \(\mathcal{H}_D(X) \equiv 0 \mod p\).** The obvious approach to solving \(\mathcal{H}_D(X) \equiv 0 \mod p\) is to use Berlekamp's algorithm [10, 52]. The complexity of this algorithm is roughly

\[O((d^2 \log p) + d^3)(\log d)(\log p)^2),\]

if we use standard algorithms (with \(d = h(-D)/g(-D)\)). For small \(d\), it is possible to mimic the standard resolution over \(\mathbb{C}\) (see [96, 65]).

Alternatively, one can use the fact that the Galois group of \(\mathcal{H}_D(X)\) is very often a dihedral group. Then, with Girstmair's ideas [39], it is possible to solve the equation \(\mathcal{H}_D(X) \equiv 0 \mod p\) by radicals and use the resulting expressions modulo \((p, f(z))\), where \(f(z)\) is any factor of the \(h\)th cyclotomic polynomial modulo \(p\). For example, take \(p = 439 = 1^2 + 1 \times 6 + 6^2 \times (47 + 1)/4\), whose order modulo 5 is 2. A root of \(W_{47}(X) = X^3 - X^3 - 2X^2 - 2X - 1\) over \(\mathbb{C}\) is given...
by
\[ 5x_5 = \sum_{k=1}^{\frac{4}{5}} z^{(k)}(y^k), \]
with:

<table>
<thead>
<tr>
<th>(i)</th>
<th>(2z^{(i)})</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>((80\sqrt{-47} - 650)\zeta^4 + (15\sqrt{-47} - 975)\zeta^3 + (-975 - 15\sqrt{-47})\zeta^2 + (-650 - 80\sqrt{-47})\zeta)</td>
</tr>
<tr>
<td>2</td>
<td>((15\sqrt{-47} - 105)\zeta^4 + (5\sqrt{-47} - 185)\zeta^3 + (-185 - 5\sqrt{-47})\zeta^2 + (-105 - 15\sqrt{-47})\zeta)</td>
</tr>
<tr>
<td>3</td>
<td>((-2\zeta^4 - 8\zeta^3 - 8\zeta^2 - 2\zeta))</td>
</tr>
<tr>
<td>4</td>
<td>((\sqrt{-47} - 35)\zeta^4 + (-15 - 3\sqrt{-47})\zeta^3 + (3\sqrt{-47} - 15)\zeta^2 + (-35 - \sqrt{-47})\zeta)</td>
</tr>
</tbody>
</table>

where \(\zeta\) is a primitive 5th root of unity and \(y^5 = z_1\). We work over \((\mathbb{Z}/439\mathbb{Z})[z]/(z^2 + 70z + 1)\): The corresponding value of \(\zeta\) is simply \(z\). A square root of \(-47 \mod 439\) is \(294\). We extract a fifth root of \(y\) using an extension of the algorithm of \([1]\) as described in \([47]\) (see also \([46]\)). We find
\[ y^5 = 269z + 64 = (383z + 244)^5 \mod(439, z^2 + 70z + 1), \]
and \(x = 15\) is a root of \(W_{41} \mod 439\). The ideas are detailed at some length in \([64]\).

In general, the Abelian Galois group \(K_H/K_S\) is cyclic; when this is so and the order is composite, the usual resolution into a sequence of equations of prime degree (each with coefficients in the field defined by the previous equation) is highly effective in solving for the (known) root modulo \(p\). For example, with \(D = -199\) we find
\[ H_{4\times 199}[f/\sqrt{2}](X) = X^9 - 5X^8 + 3X^7 - 3X^6 - 3X^3 - X - 1, \]
whose roots are solved via
\[ Y^3 - 4Y^2 + Y - 1, \quad X^3 - (Y^2 - 3Y + 1)X^2 - X - Y. \]
Further examples can be found in \([66]\).

8.6.2. Finding the right equation for \(E\). We have to find an equation of the curve \(E(\mathbb{Z}/p\mathbb{Z})\) whose invariant is \(j\) (computed above) and whose Frobenius is \(\pi\) with \(p\) a norm in \(\mathbb{Q}(\sqrt{-D}): p = \pi\pi'.\) In the general case \(D > 4\), the equation of \(E\) is of the form
\[ y^2 = x^3 + 3kc^2x + 2kc^3, \]
where \(k = j/(1728 - j)\) with \(c\) any element of \(\mathbb{Z}/p\mathbb{Z}\).

We can restate the problem as follows. By Deuring's work, we have
\[ \Sigma_E(p) = \sum_{x=0}^{p-1} \left( \frac{x^3 + 3kc^2x + 2kc^3}{p} \right) = -\text{Tr}_K(\pi_c), \]
where \(\pi_c\) is the actual Frobenius of \(E\) as parametrized by \(c\). As a matter of fact, it is always possible to write
\[ \text{Tr}_K(\pi_c) = \varepsilon(D, \pi) \left( \frac{c}{p} \right) \text{Tr}_K(\pi), \]
where $e(D, \pi)(\varepsilon \in \{\pm 1\})$ is a function of $\pi$ and $D$. The equation we are looking for is thus characterized by $c$ such that
\[
\left( \frac{c}{p} \right) = e(D, \pi).
\]

The aim of this section is to explain how we find the value of $e(D, \pi)$ in some cases. Before that, we treat two special cases.

**The case $h(-D) = 1$.** The first two cases are $D = 3, 4$. They are treated at length in [48, Chapter 18, §§3-4] and involve quartic and sextic symbols. For the sake of self-containedness, we just give the algorithms used in each case. The validity of these come from [63]. Let us first consider $D = 3$.

**procedure** FINDE3($p$)

(* $p = n \varepsilon' \text{ with } n = A + Bp, A, B \text{ in } \mathbb{Z} \text{ and } p = (1 + \sqrt{-3})/2$*)

(* the equation of $E$ is $y^2 = x^3 + b$*)

1. let $\zeta = r + sp$ with $r, s \in \{\pm 1, 0\}$, $r \equiv 2(A - B) \mod 3$, and $s \equiv B \mod 3$; then $\zeta^6 = 1$ and $\zeta^4 \equiv 2 \mod 3$;
2. determine $\mathcal{B}$ in $\mathbb{Z}/p\mathbb{Z}$ such that $\mathcal{B} \equiv -\zeta \mod p$:
   1. solve $(A - B)v + Bu = s$ in rational integers $(u, v)$;
   2. put $\mathcal{B} = -r + Au - Bv$;
3. any $b$ such that $(4b)(p^{-1/2}) \equiv \mathcal{B} \mod p$ yields a curve $E: y^2 = x^3 + b$ such that $#E = N_K(p - 1)$.

For $D = 4$, we have

**procedure** FINDE4($p, \pi$)

(* $p = n \varepsilon' \text{ with } n = A + Bi, A, B \text{ in } \mathbb{Z} \text{ and } i^2 = -1$*)

(* exactly one of $A$ or $B$ is even *)

(* the equation of $E$ is $y^2 = x^3 + ax$*)

1. let $r$ and $s$ be two integers in $\{\pm 1, 0\}$ such that $rs = 0$ and $(r, s) = (0, (B - A) \mod 4)$ if $A$ is even, and $(r, s) = ((A - B) \mod 4, 0)$ if $B$ is even; then $\zeta^4 = 1$ and $\pi \equiv \zeta \mod (2 + 2i)$;
2. determine $\mathcal{A}$ in $\mathbb{Z}/p\mathbb{Z}$ such that $\mathcal{A} \equiv \zeta^{-1} \mod p$:
   1. solve $Av + Bu = s$ in rational integers $(u, v)$;
   2. then $\mathcal{A} = r + Au - Bv$;
3. any $a$ with $(-a)(p^{-1/2}) \equiv \mathcal{A} \mod p$ gives a curve $E: y^2 = x^3 + ax$ with $#E = N_K(p - 1)$.

When $D = 8$, we use a result from [78]. Write
\[
E_\vartheta: y^2 = x(x^2 - 4\vartheta x + 2\vartheta^2)
\]
with $\vartheta$ in $\mathbb{Z}$.

**Theorem 8.1.** Let $p = n \varepsilon' = A^2 + 2B^2 = 8\kappa + l$ ($l \in \{1, 3\}$) with $A$ and $B$ in $\mathbb{Z}$, $A$ odd. Then
\[
\Sigma_{E_\vartheta}(p) = -\left( \frac{\vartheta}{p} \right)(-1)^\kappa \left( \frac{-1}{A} \right) \text{Tr}_K(\pi).
\]

We can restate this in the form of (39). We have
\[
j(\sqrt{-2}) = 20^3 \quad \text{and} \quad k = -\frac{5^3}{2.7^2}.
\]
Letting \( c = 140/15 \), we find that \( E_\theta \) is isomorphic to (39). Hence, we deduce that

\[
\Sigma_E(p) = \Sigma_{E_\theta}(p) = - \left( \frac{15 \times 14 \times c}{p} \right) (-1)^{\kappa} \left( \frac{-1}{A} \right) \text{Tr}_K(\pi).
\]

We conclude that

\[
e(8, \pi) = \left( \frac{2 \cdot 3 \cdot 5 \cdot 7}{p} \right) \left( \frac{-1}{A} \right) (-1)^{\kappa}.
\]

For the remaining cases where \( h(-D) = 1 \) and \( D \) is odd, we refer to some work of Rajwade. In [79], he has designed a method to solve the problem in the case where \( D = 7 \) and later extended it to the cases \( D \in \{11, 19\} \) (see also the bibliography in [76]). He uses the action of the Frobenius of the curve on the \( \sqrt{D} \)-division points to deduce the actual value of \( \Sigma_E \). For \( D \) in \( \{43, 67, 163\} \), he quotes some unpublished results from Stark (see [74]).

All these results can be summarized by the following theorem:

**Theorem 8.2.** Suppose that \( D \) is odd and \( h(-D) = 1 \). Let \( j = j((-1+\sqrt{-D})/2) \) be the invariant of the curve having complex multiplication by the maximal order of the quadratic field \( K = \mathbb{Q}(\sqrt{-D}) \), defined over \( \mathbb{Q} \). Let \( u \) and \( v \) be defined by

\[
u = \left( \frac{2}{D} \right)v_0, \quad v_0 > 0.
\]

Let \( p \) be a norm for \( -D : p = \pi\pi' \). (In this case, this is merely the same as \( (-D/p) = +1 \).) Then

\[
e(D, \pi) = \left( \frac{3uv}{p} \right) \left( \frac{2\text{Tr}_K(\pi)}{D} \right).
\]

We now give the numerical values of \( u \) and \( v_0 \) for all \( D \).

<table>
<thead>
<tr>
<th>( D )</th>
<th>( u )</th>
<th>( v_0 )</th>
<th>( (2/D) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>(-3.5)</td>
<td>(3^3)</td>
<td>+</td>
</tr>
<tr>
<td>11</td>
<td>(-2^5)</td>
<td>(2^3.7)</td>
<td>-</td>
</tr>
<tr>
<td>19</td>
<td>(-2^5.3)</td>
<td>(2^3.3^3)</td>
<td>-</td>
</tr>
<tr>
<td>43</td>
<td>(-2^6.3.5)</td>
<td>(2^3.3^2.7)</td>
<td>-</td>
</tr>
<tr>
<td>67</td>
<td>(-2^5.3.5.11)</td>
<td>(2^3.3^3.7.31)</td>
<td>-</td>
</tr>
<tr>
<td>163</td>
<td>(-2^6.3.5.23.29)</td>
<td>(2^3.3^3.7.11.19.127)</td>
<td>-</td>
</tr>
</tbody>
</table>

**Examples.** Let \( p = 17401 = 101^2 + 2 \times 60^2 \). We choose \( \pi = 101 + 60\sqrt{-2} \) \( (m = p + 1 - 2 \times 101 = 17200) \). We find that \( \kappa = 2175 \) and

\[
\left( \frac{c}{p} \right) = \left( \frac{2 \cdot 3 \cdot 5 \cdot 7}{p} \right) (-1)^{\kappa} \left( \frac{-1}{101} \right) = (+1)(+1)(+1)(-1)(-1)(+1) = +1.
\]

Therefore, the curve \( E : y^2 \equiv x^3 + 15444x + 10296 \mod 17401 \) has 17200 points. Let \( p = 107 \) and \( D = 7 \). We find

\[
107 = 10^2 + 7 \times 1^2,
\]
so that \((107) = (\pi)(\pi')\) with \(\pi = 10 + \sqrt{-7}\). We choose \(m = N_K(\pi - 1) = 88\).

We compute
\[
\left( \frac{c}{107} \right) = \left( \frac{-3(3 \times 5)(3^3)}{107} \right) \left( \frac{20}{7} \right) = (-1) \left( \frac{15}{107} \right) \left( \frac{20}{7} \right) = (-1)(-1)(-1) = -1,
\]
and the curve \(y^2 \equiv x^3 + 60x + 80 \mod 107\) has 88 points.

The case \(h = 2\). Let \(D \equiv 3 \mod 4\) such that \(h(-D) = 2\). Write \(-D = (q_1)(-q_2)\). Then \(j\) lies in \(Q(\sqrt{q_1})\). Let
\[ j = a + b\sqrt{q_1} \quad \text{and} \quad \nu = \sqrt{(j - 1728)/(-D)}.\]

Then \(\nu\) is also in \(Q(\sqrt{q_1})\) (see [44]). We conjecture the following:

**Conjecture 8.1.** Suppose that \(-D = q_1(-q_2)\) as above and \(\nu = A + B\sqrt{q_1}\) in such a way that \(\text{sign}(B) = -\text{sign}(b)\). Then
\[ e(D, \pi) = \left( \frac{3\nu (\frac{j}{D})}{p} \right) \times \left( \frac{\text{Tr}_K(\pi)}{D} \right).\]

**Example.** Let \(-D = -403 = (13)(-31)\). We have
\[
\begin{align*}
    j &= -122640569461466569598976000 \\
    &+ 340143739727246741938176000\sqrt{13}, \\
    \nu &= 1233529551576 - \frac{4445554048000}{13}. 
\end{align*}
\]

**Remark.** The above results are also related to the concept of \(Q\)-curve as introduced by Gross in [43]. Some of the methods used by him would yield the same results, but using deep methods from algebraic geometry.

8.6.3. **Finding \(P\) on \(E\).** Let \((a, b)\) be two elements of \(Z/pZ\). If \(x_0\) is any element of \(Z/pZ\), put
\[ \lambda = x_0^3 + ax_0 + b.\]
Then \(P = (\lambda x_0, \lambda^2)\) is on the curve
\[ Y^2 = X^3 + a\lambda^2 X + b\lambda^3.\]

We suppose that \(E : y^2 = x^3 + 3kc^2x + 2kc^3\). If we know something about \((c/p)\) (typically when \(h(-D) = 1\), then we choose \(x_0\) such that \((\lambda/p)\) agrees with \((c/p)\). Then, we have simultaneously \(E\), and \(P\) on \(E\). Otherwise, we choose \(x_0\) at random and test whether \(mP\) is on \(E\). If it is not, then we try the twist of \(E\). In the general case, the time needed to find the right curve is thus 1.5 times the time needed for the \(h = 1\) case. In all cases, we have no extraction of square roots modulo \(p\).

9. **Numerical results**

9.1. **Timings on random input numbers.** We follow the protocol given in [27]. That is, we obtain certain statistics on the behavior of our program for 20 numbers of \(w\) 32-bit words. The program is the DecStation 5000 version with
the FAS strategy. We list the time for the first phase (building the sequence) on the first line, the second one (proving) on the second line and the total time on the third. Times are in seconds. The set $\mathcal{D}$ consists of all $D$’s with $h(-D) \leq 20$.

For larger numbers, we use a distributed process with all $D$ with $h$ less than 51 and some others (see §8.2.3). The order of magnitude of the time needed is given in equivalent time for a DecStation 5000.

9.2. Some large primes. Both authors used their implementations to give primality proofs for the probable primes of the Cunningham Tables [19]. The first author did some with 212 to 343 digits (namely the cofactor of $2,1171^+$) [19, Update #5] and the second author completed the long-standing list (about 50 numbers with more than 200 digits). The second author verified the primality of the cofactor of $F_{11}$ (564 digits) [19, Update 2.2], and also $(2^{3539}+1)/3$ (1065 digits) with a distributed version of ECPP [67].

Table 4. Time for testing a number of $w$ words for primality

<table>
<thead>
<tr>
<th>$w$</th>
<th>min</th>
<th>max</th>
<th>mean</th>
<th>st. dev.</th>
<th>$w$</th>
<th>min</th>
<th>max</th>
<th>mean</th>
<th>st. dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.1</td>
<td>0.4</td>
<td>0.2</td>
<td>0.1</td>
<td>12</td>
<td>14.9</td>
<td>238.9</td>
<td>45.6</td>
<td>48.1</td>
</tr>
<tr>
<td></td>
<td>0.0</td>
<td>0.3</td>
<td>0.1</td>
<td>0.1</td>
<td></td>
<td>31.3</td>
<td>88.3</td>
<td>61.1</td>
<td>14.2</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>0.6</td>
<td>0.3</td>
<td>0.1</td>
<td></td>
<td>50.4</td>
<td>301.9</td>
<td>106.6</td>
<td>49.4</td>
</tr>
<tr>
<td>4</td>
<td>0.4</td>
<td>2.6</td>
<td>1.2</td>
<td>0.5</td>
<td>14</td>
<td>41.8</td>
<td>181.1</td>
<td>75.5</td>
<td>36.8</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>2.7</td>
<td>1.5</td>
<td>0.7</td>
<td></td>
<td>75.8</td>
<td>150.7</td>
<td>107.3</td>
<td>21.4</td>
</tr>
<tr>
<td></td>
<td>0.7</td>
<td>5.3</td>
<td>2.7</td>
<td>1.1</td>
<td></td>
<td>122.9</td>
<td>259.7</td>
<td>182.8</td>
<td>49.3</td>
</tr>
<tr>
<td>6</td>
<td>1.7</td>
<td>55.2</td>
<td>6.1</td>
<td>11.4</td>
<td>16</td>
<td>120.2</td>
<td>763.8</td>
<td>261.1</td>
<td>138.2</td>
</tr>
<tr>
<td></td>
<td>3.5</td>
<td>8.9</td>
<td>5.8</td>
<td>1.5</td>
<td></td>
<td>129.5</td>
<td>226.8</td>
<td>174.9</td>
<td>26.6</td>
</tr>
<tr>
<td></td>
<td>5.4</td>
<td>61.6</td>
<td>12.0</td>
<td>11.7</td>
<td></td>
<td>249.7</td>
<td>990.7</td>
<td>436.0</td>
<td>154.1</td>
</tr>
<tr>
<td>8</td>
<td>2.5</td>
<td>13.7</td>
<td>7.7</td>
<td>3.0</td>
<td>18</td>
<td>118.8</td>
<td>1699.1</td>
<td>657.3</td>
<td>420.2</td>
</tr>
<tr>
<td></td>
<td>7.3</td>
<td>24.4</td>
<td>14.9</td>
<td>4.8</td>
<td></td>
<td>123.4</td>
<td>339.5</td>
<td>214.7</td>
<td>53.6</td>
</tr>
<tr>
<td></td>
<td>10.2</td>
<td>34.8</td>
<td>22.6</td>
<td>7.5</td>
<td></td>
<td>242.1</td>
<td>1973.8</td>
<td>872.1</td>
<td>447.5</td>
</tr>
<tr>
<td>10</td>
<td>7.6</td>
<td>31.3</td>
<td>19.9</td>
<td>6.2</td>
<td>20</td>
<td>345.9</td>
<td>5656.8</td>
<td>1598.0</td>
<td>1392.2</td>
</tr>
<tr>
<td></td>
<td>20.6</td>
<td>54.6</td>
<td>35.3</td>
<td>9.2</td>
<td></td>
<td>175.5</td>
<td>362.3</td>
<td>255.3</td>
<td>52.3</td>
</tr>
<tr>
<td></td>
<td>28.2</td>
<td>85.0</td>
<td>55.2</td>
<td>14.2</td>
<td></td>
<td>561.0</td>
<td>5931.5</td>
<td>1853.3</td>
<td>1404.1</td>
</tr>
</tbody>
</table>

Table 5. Time for testing a $d$-digit number for primality

<table>
<thead>
<tr>
<th>$d$</th>
<th>DOWNRUN</th>
<th>proving</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>1 day</td>
<td>0.1 days</td>
</tr>
<tr>
<td>600</td>
<td>6 days</td>
<td>0.5 days</td>
</tr>
<tr>
<td>800</td>
<td>18 days</td>
<td>5 days</td>
</tr>
</tbody>
</table>

Aside from the Cunningham project, the second author found all primes of the form

$$N_2(n, r) = r \underbrace{1 \cdots 1}_{(n \text{ times})}$$

(introduced in [93]) for $r$ greater than 1 and all $n$ between 100 and 1000. We indicate below these values (note that all numbers with $n \leq 99$ were found by Williams).
In addition, some large probable primes were successfully tested. Among these were $S_{1493}$ (572 digits, three weeks on a SUN 3/60) and $S_{1901}$ (728 digits, one month), thus solving the problem mentioned at the end of [72].

Apart from these numbers with quite a lot of arithmetical properties, the second author is currently looking for large primes coming from the factorization of the numbers constructed from well-known constants such as $\pi$, $e$, and $\gamma$. To this date, the three largest proven primes found are the cofactor of $\gamma_{1137} = [10^{1137}\gamma] = 2 \times 47 \times 4231 \times 7789 \times p_{1128}$ (with the distributed implementation in equivalently about 1.75 years of CPU of a SUN 3/60); the cofactor of $e_{1230} = [10^{1230}e] = 36037 \times p_{1226}$ (1.83 years of CPU); the partition number $p(1840926)$ with 1505 digits [69]. Together with the 1008-digit cofactor of $M_{3359}$, these are five Titanic primes successfully tested by ECPP.

10. What Proof do we Get?

We now turn our attention to the following problem: How can we be sure that our program did not make any error during one month of CPU time? We cannot be certain that there was no bit-loss during this period. However, when the program finishes, we have built a sequence of intermediate primes and found an elliptic curve and its number of points and a point on it satisfying the requirements of a theorem. This we call a certificate of primality. We thus generalize previous work of Pratt [77] and Pomerance [75]. We arrange such a certificate in blocks of integers. Each block has the following structure:

$$
\begin{array}{c|c}
N_i & type \\
\hline
& PROOF \\
\end{array}
$$

where $N_i$ is the number to be tested, type giving the type of theorem used to show the primality of $N_i$: it is $-1$ (resp. $+1$) if the $N - 1$ test (resp. $N + 1$ test) was used, and the absolute value of the fundamental discriminant used in the cases of elliptic curves. The primality proof of $N_0$ ends with a 0. To each of the types corresponds a list of numbers used to complete the proof of $N_i$ being prime, whenever the following block is valid. We now describe the four possible lists:

In this way, an independent verifier can check the results. A cross verification of certificates was carried out between the second author and Kaltofen and Valente (personal communications via e-mail, October 1989). After some adjustments of format, they both agreed on the certification of a 222-digit prime, namely 2, 1958M (in the notations of [19]). They also checked the 1226-digit record.
11. Conclusion

We have described a primality proving algorithm using the theory of elliptic curves with complex multiplication over finite fields. This algorithm is supposed to have polynomial complexity and performs well in practice, since it is powerful enough to prove the primality of numbers from 100 to 1500 digits. It is now possible to test arbitrary integers up to 400 digits in a few days on a single SUN 3/60 workstation. Numbers with less than 800 digits can be done in about one week of real time, using a distributed process [67] on about 10 workstations.

<table>
<thead>
<tr>
<th>type (-1)</th>
<th>type (+1)</th>
<th>type (D)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(p_0)</td>
<td>(q_0)</td>
<td>(m = #E)</td>
</tr>
<tr>
<td>(\cdots)</td>
<td>(\cdots)</td>
<td>(r_0)</td>
</tr>
<tr>
<td>(p_k)</td>
<td>(q_l)</td>
<td>(\cdots)</td>
</tr>
<tr>
<td>(0)</td>
<td>(0)</td>
<td>(r_u)</td>
</tr>
<tr>
<td>(b_0)</td>
<td>(P_0, Q_0)</td>
<td>0</td>
</tr>
<tr>
<td>(\cdots)</td>
<td>(\cdots)</td>
<td>(a, b: coefficients of E)</td>
</tr>
<tr>
<td>(b_l)</td>
<td>(P_k, Q_k)</td>
<td>(f_1)</td>
</tr>
<tr>
<td>(\text{id.})</td>
<td></td>
<td>(x, y: coordinates of } P \text{ on } E)</td>
</tr>
<tr>
<td>(\text{Cf. Theorem 1 in [97]})</td>
<td></td>
<td>(f_u)</td>
</tr>
<tr>
<td>(\text{factors of } m)</td>
<td></td>
<td>(\text{factors of the order of } P)</td>
</tr>
</tbody>
</table>

**Figure 2. Format of the primality proof**

There remains much uncertainty as to the best strategy for applying the method to large probable prime inputs. We first eliminate some minor points which are not germane to the general problem.

The situation for 100 digits and less is quite atypical. There the downrun is dominated by \(D = \pm 1, D = -3\), and \(D = -4\); in particular, once \(D = -3\) is reached, one can usually stay with it to the end. Square roots are much cheaper relative to sieving than they are for large inputs, and optimization is desirable at all stages of the program.

Also (for all sizes of input) the reduction of quadratic forms takes negligible time, and the polynomials \(H_D\) can be computed very quickly at the time when they are needed.

Thus, the general operations which should be programmed optimally, and whose timings on a particular machine are relevant to the strategy are:

1. Sieving and subsequent factorization of the numbers of points,
2. Exponentiation modulo \(p\) (and equivalent square roots, pseudoprime tests),
3. Exponentiation on an elliptic curve modulo \(p\),
4. Solution of polynomial congruences modulo \(p\).

Usually, 4 can be reduced to finding a small number of square roots, but an occasional discriminant with large class number which is unfavorable for \(p\) can be very expensive. As to sieving, it is worth pointing out that it is much more effective here, relative to other factorization methods, than usual. Once \(-1\) and \(+1\) have been done (as discriminants), there is available a list of \((N+1) \mod q\). For any particular discriminant \(-D\), one only needs to use half the sieving
primes $q$, dividing numbers of size the square root of $N$, and applicable to
two possible numbers of points, a total improvement factor of 8 (16 or 24 for
$D = 4$ or 3). On the other hand, $(p - 1)$-factorization and ECM are no better
than usual (except that one can in a few cases use an elliptic curve with complex
multiplication to good effect in ECM).

A further remark is that the timings of these operations depend not only on
the machine, but on the trouble which the programmer has been prepared to
take. For example, some critics purport to “prove” that the Weierstrass normal
form is not the best one to use in 3 above, but they rely on an unproved (and
possibly unconscious) assumption that finding inverses is slow. The first author
is fortunate in having the use of a very fast gcd routine written by N. W. Rickert
[81], which alters his choice of algorithm in this and other cases. We will now
assume that all these operations have been optimized as far as they are going
to be, and that the timings for various typical numbers of decimal digits are
known.

We feel that the optimal strategy will probably have more backtracking fa-
cility than either of us uses at the moment. At a given point in the downrun,
one has basically to choose four parameters: the size of the sieve, the additional
factoring to be used, the minimum acceptable downrun, and how many discrim-
inants to try before modifying the parameters. There is no doubt that sieving
represents by far the best value for time spent, so that for inputs of 500 decimal
digits or more one should probably think in terms of a sieve with several passes
and recomputed lists of primes. We hope to implement some of these ideas and
report further in due course.

The second author has made his C program (INRIA/ecpp.V3.4.1.tar.Z)
available via anonymous ftp from the site ftp.inria.fr (or 128.93.1.26).

Acknowledgments

Many people are to be thanked for the help they gave to various stages of this
work. J. Cougnard read a preliminary version of [65] that is incorporated in this
article. J. McKay gave some advice concerning Galois theory and pointed to
[39]. V. Miller and J.-F. Mestre communicated their work on $X_0(s)$. Finally, H.
Cohen carefully read a preliminary version of this paper. We also acknowledge
with gratitude the numerous detailed suggestions of the referee.

Bibliography

1. L. Adleman, K. Manders, and G. L. Miller, On taking roots in finite fields, Proc. 18th Annual
175–178.


3. L. M. Adleman, C. Pomerance, and R. S. Rumely, On distinguishing prime numbers from


5. ———, The number of points on an elliptic curve modulo a prime, Preprint, 1991.

INRIA, June 1990.

7. ———, Finding suitable curves for the elliptic curve method of factorization, Math. Comp. 60
(1993), 399–405.


32. G. Cornacchia, *Su di un metodo per la risoluzione in numeri interi dell' equazione $\sum_{k=0}^{n} C_k x^{n-k} y^k = P$*, G. Mat. Battaglini 46 (1908), 33–90.


A. O. L. ATKIN AND F. MORAIN


69. ____*, *Prime values of partition numbers and the primality of \(p(1840926)\)*, preprint, August 1992.


82. R. Schertz, *Die singulären Werte der Weberschen Funktionen \(f, f_1, f_2, \gamma_2, \gamma_3\)*, J. Reine Angew. Math. 286-287 (1976), 46–74.


86. ____*, *Five number theoretic algorithms*, Proc. 2nd Manitoba Conf. on Numerical Math., 1972, pp. 51–70.


**Department of Mathematics, University of Illinois at Chicago, Box 4348, Chicago, Illinois 60680**

**Projet ALGO, Institut National de Recherche en Informatique et en Automatique, Domaine de Voluceau, B. P. 105, 78153 Le Chesnay Cedex France and Département de Mathématique, Université Claude Bernard, 69622 Villeurbanne Cedex, France**