IV. ALGORITHM.

1. OUTLINE OF THE ALGORITHM.

(1.1) Remark. This section contains a rough outline of the primality testing algorithm that is described in the following sections in detail. We will also comment upon the differences with its predecessor, the Cohen-Lenstra version of the Adleman-Pomerance-Rumely primality proving algorithm, which we will refer to as *the old algorithm*, (cf. [2], [29], [30]).

The detailed description in the next five sections is interspersed with comments (in small print), which do not form part of the algorthm. They are meant to elucidate the steps of the algorithm, and provide references to the other chapters. The following five steps correspond to these five sections.

(1.2) Preparation of tables. This pre-calculation step is done once and for all. The integers t_0 and s_0 are chosen; the auxiliary numbers t and s, to be chosen in step (1.4), will be divisors of t_0 and s_0 respectively. The choice of s_0 gives an upper bound for the size of the integers that can be dealt with by the Jacobi sum test alone: n may not exceed s_0^3 (cf. II.9). Furthermore, all tables that will be needed, and that can be created irrespective of the arithmetic properties of n, are generated. They include: a list of primes, a list of data for the extension rings, a list of Jacobi sums and a table of exponents to express certain quotients of Gauss sums as products of Jacobi sums.

(1.3) Initializations. This is the preliminary step for testing n. It is checked whether n is not obviously composite, by subjecting it to a compositeness test and some trial divisions by small primes; during the latter at the same time factors of $n-1$ and $n+1$ are found. Here also known factors for n^w-1 for small values of w, for instance found before by someone else, can be read into the program.

(1.4) Optimization. Integers s, t, u, v and w have to be chosen; roughly speaking these have the following meaning. The completely factored part of n^w-1 is denoted by v; it will consist of the factors found in the previous steps and new factors found here by trial division of n^w-1 for small w. The integer s will be a product $\prod q$ of primes q with the property that $q-1 \mid t$; also we want s to be large, in particular we want $s \cdot v > n^{\mu}$, with $\mu = \frac{1}{2}$ or $\mu = \frac{1}{3}$. Therefore t can best be built up from powers of small primes; we require $t \mid t_0$. The integer u is the total degree of the ring extension in which the Jacobi sum test will take place.

An effort is made in this step, to determine the auxiliary integers in such a way that the total running time of the algorithm will be minimal. The main contributions to the running time are the time spent on looking for more factors of $n^w - 1$, the time needed to construct the necessary roots of unity in the extension rings, the time necessary for doing the Jacobi sum tests, which is influenced by how well they combine, and the time necessary to do the final trial divisions.

(1.5) Lucas-Lehmer and Jacobi sum tests. A Jacobi sum test has to be performed for every pair (p^k, q) consisting of a prime power p^k dividing $q-1$ and a prime factor q of s. Basically, this comes down to raising a product of Jacobi sums to a power which is of the same magnitude as n and checking that the result equals some p^k-th root of unity; all this is done in a ring extension of $\mathbf{Z}/n\mathbf{Z}$ of degree equal to the order of n in $\left(\mathbf{Z}/p^k\mathbf{Z}\right)^*$, which divides u. Several of these tests will be combined to one test in common ring extension of $\mathbf{Z}/n\mathbf{Z}$, of which the degree is equal to the maximum of, and divisible by each of, the degrees of the individual tests in the combination. Performing the Lucas-Lehmer test comes down to finding v-th roots of unity in the proper w-th degree ring extension of $\mathbf{Z}/n\mathbf{Z}$.

(1.6) Final trial divisions. If the preceding tests have been performed successfully, then every divisor of n must be a power of n modulo $s \cdot v$. Thus the final step consists of finding out whether there exist integers r dividing n in the residue class $r_i \equiv n^i \bmod (s \cdot v)$, with $1 \leq i < t \cdot w$.

(1.7) Remarks. There are three important differences between this algorithm and the old algorithm, which have far-reaching consequences, especially in step (1.4) above.

In the first place there is the combination of the Jacobi sum test with the Lucas-Lehmer type tests. Every factor found in $n^w - 1$ contributes to v above. This makes it possible to decrease s by the same factor. Of course one has to balance the cost of finding extra factors and performing the Lucas-Lehmer tests against the expected gain of having less Jacobi sum tests to perform.

Secondly, an observation made in [30] is used, namely that the Jacobi sum tests can be done in a ring extension of $\mathbf{Z}/n\mathbf{Z}$ of degree equal to the order of n in $\left(\mathbf{Z}/p^k\mathbf{Z}\right)^*$ instead of the ring $\mathbf{Z}[\zeta_{p^k}]/n\mathbf{Z}[\zeta_{p^k}]$, which is of degree $\phi(p^k)$. Usually the degree used is much smaller than $\phi(p^k)$; it may be equal to 1, in which case this observation was also used in the implementation of the old algorithm.

As an aside it should be remarked that in fact the fixed extension $\mathbf{Z}[\zeta_{p^k}]/n\mathbf{Z}[\zeta_{p^k}]$ has been replaced by a list of extensions of $\mathbf{Z}/n\mathbf{Z}$, all having a non-zero probability of at most $\frac{1}{2}$ of not being suitable for a given n. That makes it more cumbersome to find efficient multiplication algorithms in each of the extension rings, than in the old algorithm.

Thirdly, advantageous use is made of the possibility to combine several Jacobi sum tests in rings of degree u_i into one large test in a ring of degree $\mathrm{lcm}(u_i)$ (compare II.(8.6)). Of course this only makes sense if the resulting amount of work is less than it would have been without combining. The Jacobi sum tests consist roughly of n-th powerings of elements that are represented as tensor products over $\mathbf{Z}/n\mathbf{Z}$ modulo polynomials of prime power degree $l^e \parallel u_i$. Assuming that the time to perform one multiplication is quadratic in the number of coordinates, combining a set of tests in rings of degree u_i into one test only makes sense if $\sum u_i^2 > (\mathrm{lcm}(u_i))^2$, where the sum is taken over different values of u_i; tests in rings having the same degree can be combined without any extra costs. Using that the u_i are built up from small prime powers, this is easily seen to be true only in case $\max\{u_i\} = \mathrm{lcm}(u_i)$ (see also III.2); that is, if all u_i divide the maximal u in the combination. It is part of the optimization step to determine the optimal combination of tests for given s, t and u. There is an easy, efficient procedure for doing that. On the other hand it is hard to find out for what choice of s and t the optimal choice will be minimal (for more on this, see Sections III.3 and III.4).

All this makes the optimization step much more complicated than in the old algorithm; but especially for large n the investment made pays off tremendously (see Chapter VI).

It should be remarked that the first and second of the differences pointed out above, and the use of $s \cdot v \geq \sqrt[3]{n}$ instead of $s \cdot v \geq \sqrt[2]{n}$, are the main contributions to the improvement of the primality test over the old Jacobi sum test.

(1.8) Conventional notation. In the next sections we will use the phrase "$a = b \bmod c$" when a is defined to be the unique integer in $\{0, 1, \ldots, c-1\}$ congruent to b modulo c.

2. PREPARATION OF TABLES.

Perform steps (2.1) through (2.5).

In this section a description is given of the preparation of the tables needed in the primality test described in Sections 3 through 6. It should be emphasized that the work done here is done once and for all, and is independent of properties, other than the size, of the integers n that are to be tested. The choice of the parameters determines the size of the integers that can be tested. With the choices made below, every n of up to 6000 digits (and some larger n) can be dealt with; the optimization routines work best however for integers n up to about 1500 decimal digits.

(2.1) Creation of a file to generate prime numbers.

Select a positive integer B_0, and create a file containing the differences between the consecutive primes up to B_0.

To find small divisors of a large number (e.g. in steps (3.2) and (4.6)), one can use a table of prime numbers up to a certain bound B_0. This table is made here. The bound B_0 could for instance be 10^6.

(2.2) Selection of t_0.

Perform steps (a) through (h).

In the Jacobi sum test one needs auxiliary integers t and s with the property that t is small, while s is large and built up from primes q such that $q-1 \mid t$ (cf. II.6). In this step a number t_0 and sets $\mathcal{P}_0$ and $\mathcal{Q}_0$ are chosen. The value of t, to be selected for a specific n during the optimization step, will divide t_0 and is built up from powers of small primes in a subset $\mathcal{P}$ of $\mathcal{P}_0$, and s will be built up from primes in a subset $\mathcal{Q}$ of $\mathcal{Q}_0$.

(a) Select a positive integer t_0 with $t_0 \equiv 0 \bmod 4$.

The integer t_0 should be "rich" in the sense that it is small but $q-1$ divides t_0 for many primes q. For a table of nice values see II.6. In II.(6.17) it is explained why preferably $t_0 \equiv 0 \bmod 4$. A possible choice is $t_0 = 2^5 \cdot 3^3 \cdot 5^2 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 = 6983776800$.

(b) Let $\mathcal{P}_0 = \{p : p \text{ prime}, p \mid t_0\}$, and let k_p be given by $\prod_{p\in\mathcal{P}_0} p^{k_p} = t_0$.

(c) Calculate $\phi(p^k) = (p-1)p^{k-1}$ for $p \in \mathcal{P}_0$ and $1 \le k \le k_p$.

(d) Put $\lambda(t_0) = \operatorname{lcm}\{2^k,\ (p-1)p^{k_p-1} :\ \ p \in \mathcal{P}_0, p\ \text{ odd}\}$ with $k = \max\{1, k_2 - 2\}$, let $\mathcal{L}_0 = \{l : l \text{ prime}, l \mid \lambda(t_0)\}$, and let e_l be given by $\prod_{l\in\mathcal{L}_0} l^{e_l} = \lambda(t_0)$.

The integer $\lambda(t_0)$ is the exponent of $(\mathbf{Z}/t_0\mathbf{Z})^*$, i.e., the maximal order of the elements in the group $(\mathbf{Z}/t_0\mathbf{Z})^*$. For the choice of t_0 suggested above, one gets $\lambda(t_0) = \operatorname{lcm}\{8, 18, 20, 6, 10, 12, 16, 18\} = 2^4 \cdot 3^2 \cdot 5$.

(e) Determine $\mathcal{Q}_0 = \{q : q \text{ prime}, q-1 \mid t_0\}$.

For the above choice of t_0 one gets $\#\mathcal{Q}_0 = 618$, and $\prod_{q\in\mathcal{Q}_0} q$ just exceeds $5.3\cdot 10^{3000}$. This product determines the maximal size of integers that can be taken care of by the Jacobi sum test alone; one can deal with numbers of size (= logarithm) up to twice (or even thrice, see II.9) the size of this product times $2t_0$.

(f) For all $q \in \mathcal{Q}_0$ and $p \in \mathcal{P}_0$ determine $o_p(q-1)$, the number of times that p occurs in the prime factorization of $q-1$.

(g) For every divisor t of t_0 such that $t \equiv 0 \bmod 4$, calculate

$$i(t) = \sum_{p \in \mathcal{P}_0} o'_p(t) \prod_{\substack{p'>p \\ p' \in \mathcal{P}_0}} (k_{p'} + 1) \quad \text{and} \quad e'_{i(t)} = \prod_{\substack{q-1|t \\ q \text{ prime}}} q \ ;$$

here we define $o'_p(t)$ by $o'_2(t) = o_2(t) - 2$ and $o'_p(t) = o_p(t)$ for other primes p, with $o_p(t)$ the number of times that p occurs in the prime factorization of t.

In the optimization step, one uses a divisor t of t_0 for which the primality test can be finished; i.e., for which $e(t)$ is large enough. Note that $e(t)$ as in II.(6.12) is the product of $2t$ and the value $e'_{i(t)}$ given here. In order to be able to retrieve these values quickly for all divisors t of a given t', they are indexed by the number $i(t)$, running from 0 to one less than the total number of divisors congruent to 0 mod 4. The numbers k_p are defined in (2.2)(b).

(h) Tabulate t_0, $\mathcal{P}_0$, k_p for $p \in \mathcal{P}_0$, $\phi(p^k)$ for $p \in \mathcal{P}_0$ and $1 \le k \le k_p$, $\lambda(t_0)$, $\mathcal{L}_0$, e_l for $l \in \mathcal{L}_0$, $\mathcal{Q}_0$, and $o_p(q-1)$ for $q \in \mathcal{Q}_0$, $p \in \mathcal{P}_0$. For $i = 0, \ldots, \#\{t : \ t \mid t_0, \ 4 \mid t\} - 1$, also tabulate the pair $(t, \ \log e'_{i(t)})$, where t is such that $i(t) = i$ (cf. (2.2)(g)).

(2.3) Generation of Galois extensions.

Perform steps (a) through (c) for all prime powers $u = l^e$ dividing $\lambda(t_0)$, with $e > 0$.

In the Jacobi sum test one will compute in certain extension rings of $\mathbf{Z}/n\mathbf{Z}$; as explained in II.(4.11) these rings can be constructed from rings of integers of cyclic subfields (of prime power degrees dividing $\lambda(t_0)$) of cyclotomic fields. Here some preliminary steps are performed without knowledge of the integer n; they will facilitate arithmetic in the extension rings later on. Since a given cyclic field of degree l^e has for random n a probability of $l-1$ out of l to provide a "good" ring (useful in the test of n), one pre-calculates a list of such extensions for every l^e.

(a) Put $\mathcal{M}(u)$ equal to the empty set.

The set $\mathcal{M}(u)$ will in the end contain the conductors m of the field extensions of degree u from which the ring extensions will be constructed.

(b) Select a constant C.

If $u = 2$ perform steps (b1), (b2), (b3), and (b10) for every $m \in \{4, 8\}$ and steps (b1), (b2), (b4), (b5) and (b10) for every $m = k \cdot u + 1$ for which m is prime and $m \le C \cdot l^{e_l}$.

If $u = 2^e$ with $e > 1$, perform steps (b1), and (b5) through (b10) for every $m = k \cdot u + 1$ for which m is prime and $m \le C \cdot l^{e_l}$, as well as for $m = 2^{e+2}$.

If $u = l^e$ with l odd and $e \ge 1$, perform steps (b1), and (b5) through (b10) for every $m = k \cdot u + 1$ for which m is prime and $m \le C \cdot l^{e_l}$, as well as for $m = l^{e+1}$.

As was pointed out in II.4, every prime conductor m for which $u \mid m-1$ provides a cyclic field of degree u; moreover, for $u=2$ there are 3 quadratic fields inside $\mathbf{Q}(\zeta_8)$ that can be used, corresponding to conductors 4, 8 and 8. From II.(4.10) it follows that if the field of conductor 8 generated by $\zeta_8 - \zeta_8^{-1}$ can be used in the test for n, then at least one of the other fields might also be used. Therefore the field of conductor 8 with generator $\zeta_8 + \zeta_8^{-1}$ and the field of conductor 4 with generator ζ_4 suffice.

For $u = 2^e$ with $e > 1$ and $u = l^e$ with l odd there are additional useful extensions of conductor 2^{e+2} and l^{e+1} respectively, cf II.(4.6). The constant C gives an arbitrary bound on the size – and thus on the number – of conductors in the tables. One could for instance use $C = 10$.

(b1) Replace the set $\mathcal{M}(u)$ by $\mathcal{M}(u) \cup \{m\}$. Put $r = 1$ if m is prime, and put $r = 0$ if m is not prime.

(b2) Put $D = 1$. If m is prime put

$$S = \begin{pmatrix} -1 & 1 \\ -1 & 0 \end{pmatrix} \text{ and } S^* = \begin{pmatrix} 0 & -1 \\ 1 & -1 \end{pmatrix}.$$

If $m \in \{4, 8\}$ put S and S^* both equal to the 2×2 unit matrix.

In general, the matrices S and S^* will be the transition matrices between the bases consisting of powers of the generator η of the ring (see II.(4.11)) and that consisting of the elements $\varsigma_{g,i,u}$ for $i = 0, \dots, u-1$. For $u = 2$ and m is prime we have that $\varsigma_{g,0,u} = \sigma_g^0(\eta_u) = \eta_u = \eta$ and $\varsigma_{g,1,u} = \sigma_g^1(\eta_u) = \sigma_g(\eta)$; for $u = 2$ and $m \in \{4, 8\}$ we have that $\varsigma_{g,0,u} = 1$ and $\varsigma_{g,1,u} = \eta_u = \eta$. The isomorphism σ_g is defined by $\sigma_g(\zeta_m) = \zeta_m^g$, for some element g of order $\lambda(m)$ modulo m (cf. (2.2)(d)). For $u \neq 2$ the elements $\varsigma_{g,i,u}$ for $i = 0, \dots, u-1$ will be defined below. The number $D \in \mathbf{Z}_{>0}$ will be the denominator of S^{-1}, i.e., the smallest integer such that $S^* = D \cdot S^{-1}$ is an integral matrix. The matrix S will express an element which is represented in the basis $\{\eta^0, \eta^1, \dots, \eta^{u-1}\}$ in terms of the elements $\varsigma_{g,i,u}$ for $i = 0, \dots, u-1$. The matrix $D^{-1} \cdot S^*$ performs the inverse operation.

(b3) If $m = 4$ put $g = 3$ and $f = X^2 + 1$; if $m = 8$ put $g = 5$ and $f = X^2 - 2$.

In the sequel $g \in (\mathbf{Z}/m\mathbf{Z})^*$ will be such that the restriction of $\zeta_m \mapsto \zeta_m^g$ generates the Galois group of a u-th degree cyclic subextension of $\mathbf{Q}(\zeta_m)$, where m is the conductor. Furthermore, f will be the minimal polynomial for η, the generator of the cyclic field, which equals here ζ_4, respectively $\zeta_8 + \zeta_8^{-1}$, see II.(4.10).

(b4) Put $f = X^2 + X + (4 \cdot \lfloor \frac{m+1}{4} \rfloor - m) \cdot \lfloor \frac{m+1}{4} \rfloor$.

This is the minimal polynomial $X^2 + X + (1 \mp m)/4$ for the generator

$$\eta = \sum_{i=0}^{(m-3)/2} \zeta_m^{g^{2i}}$$

of our ring in the quadratic subfield $\mathbf{Q}(\sqrt{\pm m})$ of $\mathbf{Q}(\zeta_m)$, where g is a primitive root modulo m. The sign under the square root equals the Legendre symbol $(\frac{-4}{m})$, so the square root is $\sqrt{-m}$ for $m \equiv 3 \bmod 4$ and $\sqrt{m}$ for $m \equiv 1 \bmod 4$.

(b5) If m is odd, find a primitive root g modulo m, for instance by trying all $g \geq 2$ with $\gcd(m, g) = 1$ in succession. If $m = 2^{e+2}$, put $g = 5$.

It would be more efficient to determine a primitive root g modulo m for each m only once, but since this table is made only once, and since the time to determine a primitive root is relatively small in comparison to the time needed to generate other elements in the table, this improvement will not be described.

(b6) Let $b_{i,j} = 0$ for $0 \le i < u$ and $0 \le j < m$. Put $j = 1$. If m is odd, for $i = 0, 1, \ldots, \phi(m) - 1$ in succession replace $b_{i \bmod u, j}$ by 1 and j by $j \cdot g \bmod m$. If $m = 2^{e+2}$, for $i = 0, 1, \ldots, 2^e - 1$ in succession replace $b_{i \bmod u, j}$ and $b_{i \bmod u, m-j}$ by 1 and j by $j \cdot g \bmod m$. Define $\eta = \sum_{j=0}^{m-1} b_{0,j} \cdot \zeta_m^j$ and, for $i = 0, 1, \ldots, u-1$, its conjugates $\sigma_g^i(\eta) = \sum_{j=0}^{m-1} b_{i,j} \cdot \zeta_m^j$. The $\sigma_g^i(\eta)$ will be represented by the vectors $(b_{i,j})_{j=0}^{m-1}$.

The element $\eta = \eta_u$ generates a cyclic subfield of degree u inside $\mathbf{Q}(\zeta_m)$ by II.(4.8). The conjugates $\sigma_g^i(\eta)$ are also computed, where σ_g acts via $\sigma_g(\zeta_m) = \zeta_m^g$. Notice that $\sigma_g^0(\eta) = \eta$; also notice that the representation given is not unique (since the powers $\zeta_m^0, \zeta_m^1, \ldots, \zeta_m^{m-1}$ are dependent) and that $b_{i,j} = 0$ for $0 \le i < u$, $0 \le j < m$ with $\gcd(j, m) \ne 1$.

(b7) Determine the polynomial $f = \prod_{i=0}^{u-1}(X - \sigma_g^i(\eta)) \in \mathbf{Z}[X]$ by calculating sufficiently close approximations $c_i \in \mathbf{C}$ to $\sum_{j=1}^{m-1} b_{i,j} \cdot \zeta_m^j$ for $0 \le i < u$, with $\zeta_m = e^{2\pi\sqrt{-1}/m}$, and by rounding the coefficients of the polynomial $\prod_{i=0}^{u-1}(X - c_i)$ to the nearest integers.

Here f is again the minimal polynomial of η over $\mathbf{Q}$; writing $f = \sum f_i X^i$, one should notice that $f_u = 1$ and $f_{u-1} = r$, with r as in (b1).

Since all coefficients $|b_i| \le 1$ the value c_i approximates $\sum_{j=1}^{m-1} b_{i,j} \cdot \zeta_m^j$ within $m \cdot \epsilon$, where ϵ is the absolute error made in the calculation of $\zeta_m = e^{2\pi\sqrt{-1}/m}$. Since $|\zeta_m| \le 1$, this is at most equal to the machine-precision. For the machines we used the machine-precision was at most 2^{-24}.

(b8) If m is prime, define $\varsigma_{g,k,u}$ for $0 \le k < u$ by

$$\varsigma_{g,k,u} = \sigma_g^k(\eta_u) = \sigma_g^k(\eta).$$

If m is not prime, define $\varsigma_{g,k,u}$ for $0 \le k < u$ by $\varsigma_{g,0,u} = 1$, and

$$\varsigma_{g,k,u} = \sigma_g^{(k - l^{i-1})}(\eta_{l^i}),$$

with i such that $1 \le i \le e$ and $l^{i-1} \le k < l^i$. Compute a $u \times u$ dimensional integer matrix S by performing steps (b8a) through (b8c).

The matrix S is to convert an element expressed in the basis of $\eta^0, \ldots, \eta^{u-1}$ to its representation in the basis $\varsigma_{g,0,u}, \varsigma_{g,1,u}, \ldots, \varsigma_{g,u-1,u}$. For any element $x \in \mathbf{Z}[\eta]$ which is represented as a u-dimensional column vector over $\mathbf{Z}$, such that $x = \sum_{i=0}^{u-1} x_i \cdot \eta^i$, the u-dimensional column vector $y = S \cdot x$ represents the same element with respect to the basis $\varsigma_{g,0,u}, \varsigma_{g,1,u}, \ldots, \varsigma_{g,u-1,u}$:

$$\sum_{i=0}^{u-1} y_i \cdot \varsigma_{g,i,u} = \sum_{i=0}^{u-1} x_i \cdot \eta^i.$$

The element η_{l^i} generates the cyclic subfield of degree l^i inside $\mathbf{Q}(\zeta_m)$ for $i = 1, \ldots, e$, see II.(4.7). In case m is not prime, the basis $\varsigma_{g,0,u}, \varsigma_{g,1,u}, \ldots, \varsigma_{g,u-1,u}$ consists of the basis of the cyclic subfield of degree u/l with generator $\eta_{u/l}$ extended with the conjugates $\sigma_g^k(\eta)$, for $0 \le k < u - u/l$; the basis of the cyclic subfield of degree u/l equals 1 if $e = 1$ and equals $\varsigma_{g,0,u/l}, \varsigma_{g,1,u/l}, \dots, \varsigma_{g,u/l-1,u/l}$ if $e > 1$.

(b8a) Determine $d_{i,j}$ such that $\eta^i = \sum_{j=0}^{m-1} d_{i,j} \cdot \zeta_m^j$ for $i = 2, 3, \dots, u-1$, by computing the consecutive powers of the polynomial $\sum_{j=0}^{m-1} b_{i,j} \cdot T^j$ modulo the polynomial $T^m - 1$.

The powers of η are expressed as linear combinations of $\zeta_m^0, \zeta_m^1, \dots, \zeta_m^{m-1}$. Note again that this is not a unique representation.

(b8b) Perform step (b8b1) if m is prime and perform step (b8b2) for $k = 1, \dots, e$ in succession if m is not a prime.

(b8b1) First put $h = 1$ and for $j = 0, 1, \dots, u-1$ in succession first put $s_{j,i} = d_{i,h} - d_{i,0}$ for $0 \le i < u$ and next replace h by $h \cdot g \bmod m$.

(b8b2) Put $h = 1$ and for $j = l^{k-1}, \dots, l^k - 1$ in succession first put $h' = (hm/l^k) \bmod m$, next put $s_{j,i} = d_{i,h'}$ for $0 \le i < u$ and finally replace h by $h \cdot g \bmod m$. For $j = l^k, \dots, l^k + l^{k-1} - 1$ in succession first put $h' = (hm/l^k) \bmod m$, next put $s_{(j-zl^{k-1}),i} = s_{(j-zl^{k-1}),i} - d_{i,h'}$ for $z = 1, \dots, l-1$ and $0 \le i < u$ and finally replace h by $h \cdot g \bmod m$.

(b8c) Put $s_{0,i} = d_{i,0}$ for $0 \le i < u$. Let S be the matrix having the $(s_{j,i})_{j=0}^{u-1}$ as columns, for $0 \le i < u$.

(b9) Compute the $u \times u$ dimensional integer matrix S^* and $D \in \mathbf{Z}_{>0}$ such that $D^{-1} \cdot S^* = S^{-1}$ and D is minimal. This can efficiently be done using a variant of the Gaussian elimination method. See V.6 for more details.

(b10) Tabulate u, $m_u = m$, $r_{u,m} = r$, $g_{u,m} = g$, $f_{u,m} = f$, $S_{u,m} = S$, $S^*_{u,m} = S^*$, and $D_{u,m} = D$.

(c) Tabulate $\mathcal{M}(u)$.

(2.4) Replacing Gauss sums by Jacobi sums.

Perform steps (a) through (f).

In this step a method is described to determine a set of prime numbers $\mathcal{J}$, and for every prime $\pi \in \mathcal{J}$ one pair of positive integers (a, b) with $a + b = \pi$, as well as a set of expressions $e = e_{\pi,p^k,i}$ of the form $e = \sum z_j \sigma_j$ with $z_j \in \mathbf{Z}_{\ge 0}$ and j ranging over $(\mathbf{Z}/p^k\mathbf{Z})^*$, for every $\pi \in \mathcal{J}$, every $p^k \mid t_0$ and $1 \le i \le p^k$ with $p \nmid i$. One should think of the prime $\pi \in \mathcal{J}$ and the pair (a, b) as a representation of the Jacobi sum $J_\pi = J(\chi^a, \chi^b) = \tau(\chi^a)\tau(\chi^b)/\tau(\chi^\pi)$, and the exponents $e_{\pi,p^k,i}$ will be chosen in such a way that for every character χ of order p^k

$$\frac{\tau(\chi)^i}{\tau(\chi^i)} = \prod_{\pi \in \mathcal{J}} J(\chi^a, \chi^b)^{e_{\pi,p^k,i}};$$

here $J(\chi^a,\chi^b)^{z_j\sigma_j} = \sigma_j J(\chi^a,\chi^b)^{z_j} = J(\chi^{aj},\chi^{bj})^{z_j}$. Note that $\sigma_1 = 1$ and $\sigma_j\sigma_l = \sigma_{jl \bmod p^k}$. It turns out that for this purpose it suffices to consider only Jacobi sums $J(\chi^a,\chi^b)$ with $a+b$ prime.

Since the calculation of Jacobi sums (see (2.5)) is cumbersome for large conductor and the memory needed to store them expands fast for growing conductor, an effort is made here to find both a small set $\mathcal{J}$ and for given p^k a small subset of $\mathcal{J}$ such that for every π outside this subset $e_{\pi,p^k,i} = 0$ for every i. For more information, see II.(8.7) – (8.13) and V.(2.1).

(a) Let $\Pi = \max\{\pi : \pi \text{ prime, there exists a prime power } p^k \mid t_0 \text{ such that } p^k > \pi\}$. Put $\mathcal{J} = \emptyset$, put $\rho_{p^k,i} = \rho'_{p^k,i} = 0$ and put $\alpha_{p^k,i,a} = 0$ for every $p^k \mid t_0$, every $0 \le i \le p^k$, and $0 \le a \le \lfloor \frac{\Pi}{2} \rfloor$. Perform step (a1) successively for every prime power p^k dividing t_0.

The primes π in $\mathcal{J}$ will be at most equal to Π. While constructing the set $\mathcal{J}$ below, one will sometimes be forced to add primes π to $\mathcal{J}$, while some choice in splitting π into $\pi = a+b$ remains; the set $\mathcal{A}_\pi$ will for $\pi \in \mathcal{J}$ consist of the "breaking points" that are still allowed, i.e. those values for $0 < a \le \lfloor \frac{\pi}{2} \rfloor$ that one is free to choose from in (a,b). In the end choices are made such that $\#\mathcal{A}_\pi = 1$.

The exponents will be built up recursively, using the values for ρ and ρ', indicating which prime from $\mathcal{J}$ is used in the final step of the construction of $\tau(\chi)^i/\tau(\chi^i)$, and the values for α, indicating the operation that we have to perform on it. So far ρ and α were initialized only.

(a1) For $i = 2,\dots,p^k-1$ with $p \nmid i$ in succession, do the following. Put $\kappa_{p^k,i} = 1$. If there exist primes π dividing i such that $\pi \in \mathcal{J}$ or $\pi < i$, let π be minimal among these; put $\rho_{p^k,i} = \pi$ and put $\alpha_{p^k,i,0} = \sigma_{i/\pi}$.

We are trying to find an expression for $\tau(\chi)^{i-\sigma_i} = \tau(\chi)^i/\tau(\chi^i)$ that holds for any character of the present order p^k ; generating the proper denominator is the main problem. In the present case we can use the identity: $\sigma_{i/\pi} J(\chi^a,\chi^b) = \tau(\chi^{ai/\pi})\tau(\chi^{bi/\pi})/\tau(\chi^i)$, if $a+b=\pi$, which is particularly useful because both $ai/\pi < i$ and $bi/\pi < i$; this means that the numerator can be cancelled using previous instances of this step yielding $\tau(\chi^{ai/\pi})$ and $\tau(\chi^{bi/\pi})$ in the denominator. Therefore we let ρ point to the prime π and α to $\sigma_{i/\pi}$; since the above relation holds independently of the value of a, we use $\alpha_{p^k,i,0}$. The use of κ will become clear in (c).

If there exist $\pi \in \mathcal{J}$ such that for every $a \in A_\pi$ there exists j with $1 \le j \le a$ having the property that either

$$(*) \qquad a \mid j\cdot p^k - i \quad \text{and} \quad 0 < d < i, \quad \text{for some} \quad d \equiv (\pi - a)\left(\frac{j\cdot p^k - i}{a}\right) \bmod p^k$$

or

$$(**) \qquad \pi - a \mid j\cdot p^k - i \quad \text{and} \quad 0 < d < i, \quad \text{for some} \quad d \equiv a\left(\frac{j\cdot p^k - i}{\pi - a}\right) \bmod p^k,$$

then let π be the smallest of these; for this π put $\rho'_{p^k,i} = \pi$, choose j such that $(*)$ or $(**)$ holds and for every $a \in \mathcal{A}_\pi$ replace $\alpha_{p^k,i,a}$ by $\sigma_{(j\cdot p^k - i)/a}$ if $(*)$ holds and else by $\sigma_{(j\cdot p^k - i)/(\pi - a)}$.

In this case we can utilize the rule $J(\chi^a, \chi^{\pi-a})^\sigma = \tau(\chi^c)\tau(\chi^d)/\tau(\chi^i)$ for certain $c, d < i$ and the indicated value for σ, as was pointed out in II.(8.13). For step (c) we have to distinguish between the use of II.(8.13) and the rule applied in the previous step, and for that purpose we introduce ρ'. Since for the chosen value of π the "breaking point" has not been established yet and σ depends on it, we have to record α now for every value for a that is still available.

If there exist $\pi \in \mathcal{J}$ with the property that there exists at least one $a \in \mathcal{A}_\pi$ for which there is $1 \le j \le a$ such that either $(*)$ or $(**)$ above holds, let π be the smallest of these; if $\rho_{p^k,i} = \rho'_{p^k,i} = 0$ remove all a from $\mathcal{A}_\pi$ for which neither $(*)$ nor $(**)$ can be met for any j. Also in this case, if either $\rho_{p^k,i} = \rho'_{p^k,i} = 0$ or $\pi < \rho'_{p^k,i}$, replace $\rho'_{p^k,i}$ by π, choose j such that $(*)$ or $(**)$ holds, and replace $\alpha_{p^k,i,a}$ for $a \in \mathcal{A}_\pi$ by $\sigma_{(j\cdot p^k-i)/a}$ if $(*)$ holds and by $\sigma_{(j\cdot p^k-i)/(\pi-a)}$ if $(**)$ holds.

Here the same rule is used as in the previous case, but now it does not work for every a any more. To use the rule we thus have to make restrictions concerning the "breaking points" a; we are only willing to do so in case we found nothing before (for the present i) or in case we can use a smaller π than before. The latter is often advantageous, since smaller π are more likely to be needed later on anyhow.

If now still $\rho_{p^k,i} = \rho'_{p^k,i} = 0$ (in which case i must be prime), put i in $\mathcal{J}$, put $1, \dots, \lfloor \frac{i}{2} \rfloor$ in $\mathcal{A}_\pi$, put $\rho_{p^k,i} = \pi$ and put $\alpha_{p^k,i,a} = 1$ for every $a \in \mathcal{A}_\pi$.

If we have not yet been able to succeed for the current i so far, we will have to introduce a new prime. That is done here; every "breaking point" is allowed.

(b) For every π with $\#\mathcal{A}_\pi > 1$, let a be the smallest of the elements of $\mathcal{A}_\pi$, and replace $\mathcal{A}_\pi$ by $\{a\}$.

For every prime π for which there is a choice left, we choose the "breaking point" a in $\pi = a + b$.

(c) Put $e_{\pi,p^k,1} = 0$ for every $\pi \in \mathcal{J}$ and every $p^k \mid t_0$. For every prime power $p^k \mid t_0$ do the following for $i = 2, \dots, p^k - 1$ (with $p \nmid i$) in succession to find the exponents $e_{\pi,p^k,i}$ for $i < p^k$. Put $e_{\pi,p^k,i} = 0$ for every $\pi \in \mathcal{J}$. If $\rho_{p^k,i} \ne 0$ and either $\rho_{p^k,i} \le \rho'_{p^k,i}$ or $\rho'_{p^k,i} = 0$, then perform step (c1); in all other cases replace $\alpha_{p^k,i,0}$ by $\alpha_{p^k,i,a}$ with $a \in \mathcal{A}_\pi$ and perform step (c2).

Using the values for ρ and α, we will now assemble the exponents $e_{\pi,p^k,i}$ recursively.

(c1) If $\rho = \rho_{p^k,i} \in \mathcal{J}$, let a, b be such that $a + b = \rho$ and $a \in \mathcal{A}_\rho$. In this case put $e_{\rho,p^k,\rho} = \sigma_{i/\rho}$ and next replace $e_{\pi,p^k,i}$ by $e_{\pi,p^k,i} + \sigma_{i/\rho} \cdot (e_{\pi,p^k,a} + e_{\pi,p^k,b}) + \rho \cdot e_{\pi,p^k,i/\rho}$ for every $\pi \in \mathcal{J}$. If $\rho = \rho_{p^k,i} \notin \mathcal{J}$, put $e_{\pi,p^k,i} = \sigma_{i/\rho} \cdot e_{\pi,p^k,\rho} + \rho \cdot e_{\pi,p^k,i/\rho}$ for every $\pi \in \mathcal{J}$.

(c2) Let $\rho' = \rho'_{p^k,i}$. Replace $e_{\rho',p^k,i}$ by $\alpha_{p^k,i} \cdot e_{\rho',p^k,\rho'}$. If $\alpha_{p^k,i} = \sigma_{(j\cdot p^k-i)/a}$ (that is, if $(*)$ held above) put $d = (\pi - a) \cdot (j \cdot p^k - i)/a \bmod p^k$ and else (if $(**)$ was true above) put $d = a \cdot (j \cdot p^k - i)/(\pi - a) \bmod p^k$. Put $c = i - d$ and replace $e_{\pi,p^k,i}$

by $e_{\pi,p^k,i} + e_{\pi,p^k,c} + e_{\pi,p^k,d}$ for every $\pi \in \mathcal{J}$. Finally, if $p = 2$, replace $\kappa_{p^k,i}$ by $\chi^d(-1)\kappa_{p^k,i}\kappa_{p^k,c}\kappa_{p^k,d}$.

We use $\kappa_{p^k,i}$, for $i < p^k$, to keep track of sign changes, due to an appeal to II.(8.11). A factor -1 can only arise if $p = 2$.

(d) For every prime power $p^k \mid t_0$ put $e_{\pi,p^k,p^k} = e_{\pi,p^k,p^k-1}$ for every $\pi \in \mathcal{J}$. Put $\kappa_{p^k,p^k} = \chi(-1)q\kappa_{p^k,p^k-1}$.

The relation $\tau(\chi)^{p^k} = \chi(-1)q \cdot \prod_{\pi\in\mathcal{J}} J^{e_{\pi,p^k,p^k-1}}$ of II.(8.8) is used. In this particular case we multiply κ by the non-Gauss-sum factor $\chi(-1)q$.

(e) Let $\mathcal{J}_{p^k} \subset \mathcal{J}$ consist of those $\pi \in \mathcal{J}$ such that $e_{\pi,p^k,i} \neq 0$ for some $i \le p^k$.

The set $\mathcal{J}_{p^k}$ indicates which Jacobi sums we need for any character of order p^k to express $\tau(\chi)^{i-\sigma_i}$ for every $i \le p^k$.

(f) Tabulate $\mathcal{J}$, and also for every prime $\pi \in \mathcal{J}$ the pair (a, b) with $a \in \mathcal{A}_\pi$ and $a + b = \pi$, the subsets $\mathcal{J}_{p^k}$ for every $p^k \mid t_0$, and for every $\pi \in \mathcal{J}$ the exponents $e_{\pi,p^k,i}$ for every $p^k \mid t_0$ and every $0 < i \le p^k$; also tabulate $\kappa_{p^k,i}$, for all pairs p^k, i.

(2.5) Calculation of Jacobi sums.

Create a direct access file. Perform steps (a) through (d) for all odd $q \in \mathcal{Q}_0$.

A direct access file is a file which can be read and written in random access order, i.e., without sequentially reading the complete file to read one entry from or write one entry in the file (as in a sequential file). Each entry in the direct access file has the same fixed size. Using a direct access file is beneficial here, since only a few Jacobi sums will be needed for the primality test of a particular n.

For every pair p^k, q consisting of a prime $q \in \mathcal{Q}_0$ (as determined in (2.2)(e)) and a prime power p^k such that $p^k \parallel q - 1$, in this step the Jacobi sums $J(\chi^a, \chi^b)$ with $a + b = \pi$ as in (2.4)(f) are computed, for $\pi \in \mathcal{J}_{p^k}$, for a character of conductor q and order p^k; notice that $p \in \mathcal{P}_0$ (cf. (2.2)(b)). This is done by using the definition, cf. II.(8.1) :

$$J(\chi^a, \chi^b) = \sum_{x=0}^{q-1} \chi^a(x)\chi^b(1-x) \quad \text{that can be expressed as} \quad \sum_{i=0}^{\phi(p^k)-1} c_{\pi,q,p,i} \cdot \zeta_{p^k}^i\,;$$

so we represent the Jacobi sum $J(\chi^a, \chi^b)$ by the vector $(c_{\pi,q,p,i})_{0\le i<\phi(p^k)}$.

No use is made of direct formulae for $J(\chi^a, \chi^b)$ like for instance if $\mathrm{ord}(\chi) = 2$, because the calculations of all Jacobi sums for a fixed conductor q are performed in parallel. Omitting the calculation of a single Jacobi sum from this parallel computation hardly influences the total computing time. For more information, see II.(8.3) and V.(3.6).

(a) Find a primitive root g modulo q.

Among others, we will use this primitive root to make a choice for a character of order p^k and conductor q; we will choose $\chi(g) = \zeta_{p^k}$, where ζ_{p^k} is a primitive p^k-th root of unity.

(b) For all prime powers $p^k \parallel q-1$, put $c_{\pi,q,p,i} = 0$ for $0 \le i < p^k$ and for every $\pi \in \mathcal{J}_{p^k}$, as determined in step (2.4).

Let M_0 be the maximal number of integers less than q that can be stored in the available memory. Let $M = \min(M_0, q-2)$, and let $c(q)$ be the number of distinct prime divisors of $q-1$. If $M \geq \min\{q-2, 2q/((c(q)+2)\log_2 q)\}$, then perform step (b1), otherwise perform step (b2).

In (b1) and (b2) two methods are given for computing J. Comparing the number of operations required for each of these (see below), one arrives at the crossover point mentioned; in practice however, the crossover point between (b1) and (b2) will depend on the implementation and is therefore best determined empirically.

(b1) Perform step (b1a) for $0 \leq m \leq \lfloor (q-3)/M \rfloor$.

In the first method (for each of $\lfloor (q-3)/M \rfloor + 1$ blocks) a table is made of pairs $(x, f(x))$ such that $1 - g^x = g^{f(x)}$ for a block of length M out of the $q-2$ different powers of g modulo q. This requires about q^2/M multiplications modulo q.

(b1a) Put $F_i = 0$ for $i = m \cdot M + 2, m \cdot M + 3, \ldots, (m+1) \cdot M + 1$. For $x = 1, 2, \ldots, q-2$ in succession, do the following.

Compute $\gamma \equiv g^x \bmod q$ and $\delta \equiv 1 - g^x \bmod q$ with $0 \leq \gamma, \delta \leq q-1$. If $m \cdot M + 2 \leq \gamma \leq (m+1) \cdot M + 1$ and $F_\gamma \neq 0$, then increase $c_{\pi,q,p,ax+bF_\gamma \bmod p^k}$ by 1, for all prime powers $p^k \parallel q-1$ and all $\pi \in \mathcal{J}_{p^k}$; here $a + b = \pi$. Similarly, if $m \cdot M + 2 \leq \gamma \leq (m+1) \cdot M + 1$ and $F_\gamma = 0$ put $F_\gamma = x$. If $m \cdot M + 2 \leq \delta \leq (m+1) \cdot M + 1$ and $F_\delta \neq 0$, then increase $c_{\pi,q,p,aF_\delta+bx \bmod p^k}$ by 1, for all prime powers $p^k \parallel q-1$ and all $\pi \in \mathcal{J}_{p^k}$; here $a+b = \pi$. If $m \cdot M + 2 \leq \delta \leq (m+1) \cdot M + 1$ and $F_\delta = 0$ put $F_\delta = x$.

For a block of length at most M we find all pairs $(x, f(x))$ such that $g^x \bmod q$ or $g^{f(x)} \bmod q$ is in the current block and $g^x + g^{f(x)} \equiv 1 \bmod q$. Each pair $(x, f(x))$ gives a contribution $\zeta^{ax}\zeta^{bf(x)}$ to the Jacobi sum $J(\chi^a, \chi^b)$ for the choice of the character as in (a). Finding all pairs $(x, f(x))$ is done as follows. If $\gamma \equiv g^x \bmod q$ is in the block, we store x in F_γ, unless $F_\gamma \neq 0$; that can only happen if $\gamma = 1 - g^y$, for some y that we have dealt with before, in which case $y = F_\gamma$ and $f(y) = x$. Similarly, if $\delta \equiv 1 - g^x \bmod q$ is in the block, we store x in F_δ, unless $F_\delta \neq 0$; that can only happen if $\delta = g^y$, for some y that we have dealt with before, in which case $y = F_\delta$ and $f(y) = x$.

(b2) For all prime powers $p^k \parallel q-1$ put $g_{p,i} = g^{i(q-1)/p^k} \bmod q$ for $i = 0, 1, \ldots, p^k - 1$ in succession. Perform step (b2a) for $x = 1, 2, \ldots, q-2$ in succession.

For the second method one first computes the $c(q)$ powers $g^{(q-1)/p^k}$; this can be done in roughly $((c(q)/2) + 1)\log_2 q$ multiplications modulo q, if one does $\log_2 q$ squarings of g followed by assembling $g^{(q-1)/p^k}$ (requiring approximately $(\log_2 q)/2$ multiplications on the average) for each of the p^k. The same is later done for each of $q-2$ different values for $\bar{g}_x$. That adds up to $(q-1)((c(q)/2)+1)\log_2 q$ multiplications; at most q more are needed to find the p^k different powers $(g^{(q-1)/p^k})^i$ for all p^k.

(b2a) Put $\bar{g}_x = 1 - g^x \bmod q$. For all prime powers $p^k \parallel q-1$ put $\bar{g}_{p,x} = \bar{g}_x^{(q-1)/p^k} \bmod q$. Next for all prime powers $p^k \parallel q-1$, find i such that $g_{p,i} = \bar{g}_{p,x}$ and increase

$c_{\pi,q,p,ax+bi \bmod p^k}$, by 1 for every $\pi \in \mathcal{J}_{p^k}$, where $a + b = \pi$. Finding i can for instance be done using hashing.

A character χ of order p^k is chosen as in (a). The $g_{p,i}$ are the powers of $g_{p,1}$, a generator for the p-Sylow subgroup (of order p^k) inside $\left(\mathbf{Z}/q\mathbf{Z}\right)^*$. To evaluate $\chi(1-g^x)$, one raises $1-g^x$ in the power $\frac{q-1}{p^k}$ (to kill the non-p part) and finds from the list the $g_{p,i}$ to which the result is equal. The contribution to the Jacobi sum $J(\chi^a, \chi^b)$ is $\zeta^{ax}\zeta^{bi}$.

(c) For every prime power $p^k \parallel q-1$, every i such that $\phi(p^k) \le i < p^k$ and every $1 \le j < p$, decrease $c_{\pi,q,p,i-jp^{k-1}}$ by $c_{\pi,q,p,i}$ for all $\pi \in \mathcal{J}_{p^k}$.

Here we transform to a basis for $\mathbf{Z}[\zeta_{p^k}]$ by using the relation $\zeta^0 + \zeta^{p^{k-1}} + \ldots + \zeta^{(p-1)p^{k-1}} = 0$.

(d) Add the vector $(c_{\pi,q,p,i})_{0 \le i < \phi(p^k)}$ representing the Jacobi sum corresponding to π and (a,b) to the content of the direct access file for every $\pi \in \mathcal{J}_{p^k}$ and every $p^k \parallel q-1$.

3. INITIALIZATIONS.

Let $n > 1$ be an odd integer to be tested for primality. Suppose that files and tables are prepared according to Section 2.

(3.1) Initialization.

Put $G = \prod_{p \in \mathcal{P}_0 \cup \mathcal{Q}_0} p \bmod n$ (cf. (2.2)(b), (e)). If $G = 0$ the complete factorization of n can easily be derived, and the primality test is terminated.

As part of the primality proof for n, one has to verify that several integers are relatively prime to n. Instead of calculating the various gcd's, one calculates the product G modulo n of these integers, and one checks if the product equals 0 mod n. If $G \equiv 0 \bmod n$, a factor of n is easily derived, by taking the gcd of n and the last integer G has been multiplied with; the algorithm will be aborted then. If $G \equiv 0 \bmod n$ never occurs, $\gcd(G, n)$ will be computed once at the end of the algorithm (in step (6)).

(3.2) Trial division.

Put Ω equal to the empty set and perform steps (a) and (b).

In this step the numbers n, $n-1$, and $n+1$ are checked on divisibility by all small primes up to some bound B. In this way composite n with a small factor are easily detected, while the algorithm determines at the same time the small prime factors of $n^2 - 1$, which may be used in the Lucas-Lehmer part of the algorithm (cf. (3.5), (4.5), (5.3)). This step makes use of the table of prime numbers created in step (2.1). If no divisors of n are found, the sets $\mathcal{F}_1$ and $\mathcal{F}_2$ will contain all prime divisors up to B of $n-1$ and $n+1$ respectively; v_1 and v_2 will be equal to the product of all prime divisors up to B of $n-1$ and $n+1$ respectively, with their multiplicities. The value of B is highly dependent of $\log_2 n$, and should be determined empirically (cf. Chapter VI). Finally, Ω will contain all positive integers ω for which a factor of $n^{\omega} - 1$ is known, and w will be equal to $\mathrm{lcm}\{\omega : \omega \in \Omega\}$. Both Ω and w may be changed in steps (3.5) and (4.5).

(a) Set r_1 and r_2 equal to the largest odd factors of $n-1$ and $n+1$, respectively, and set $\mathcal{F}_1$ and $\mathcal{F}_2$ equal to $\{2\}$. Furthermore put $v_1 = 2^{o_2(n-1)}$ and $v_2 = 2^{o_2(n+1)}$. Select a trial division bound $1 \le B \le \min(B_0, \sqrt{n})$ (cf. (2.1)). Perform step (a1) for all odd primes $p \le B$, where the primes p are generated using the file created in (2.1).

(a1) Let n_p be the (smallest positive) remainder of the division of $n+1$ by p. If $n_p = 1$, then p is a divisor of n, and the primality test is terminated because n is composite. Otherwise, if $n_p = 0$, replace $\mathcal{F}_2$ by $\mathcal{F}_2 \cup \{p\}$, replace v_2 by $v_2 \cdot p^{o_p(r_2)}$ and r_2 by $r_2 / p^{o_p(r_2)}$. Finally, if $n_p = 2$, replace $\mathcal{F}_1$ by $\mathcal{F}_1 \cup \{p\}$, replace v_1 by $v_1 \cdot p^{o_p(r_1)}$ and r_1 by $r_1 / p^{o_p(r_1)}$.

(b) If $B = \lfloor \sqrt{n} \rfloor$, then n is proved to be prime and the test is terminated. Otherwise, put $v = v_1 \cdot v_2$, put $w = 2$ and put $\Omega = \{1, 2\}$.

(3.3) Compositeness test.

Let $n-1 = r \cdot 2^k$ with r odd and $k \geq 1$. Select a small positive integer C, and perform step (a) at most C times.

In this step a compositeness test is performed, see II.1. If some witness a to the compositeness of n is found, the number n is proved to be composite. Recall from II.1.3 that if n is a composite number, the probability that a random a in $\{2, 3, \ldots, n-1\}$ is a witness is at least $3/4$. The number of attempts C to find a witness, could for instance be taken equal to 4.

(a) Randomly select an integer a from $\{2, 3, \ldots, n-2\}$, and compute $a_r = a^r \bmod n$. If $a_r \not\equiv \pm 1 \bmod n$, then check by repeated squaring that there is an i in $\{1, 2, \ldots, k-1\}$ such that $a_r^{2^i} \equiv -1 \bmod n$; if such an i does not exist the primality test is terminated because n is composite.

(3.4) Preliminary calculations.

Perform steps (a), (b), and (c) for all primes p dividing t_0 (cf. (2.2)(a)).

In this step $u_{p,k} = \operatorname{ord}(n \bmod p^k)$ and $o^*_{p,k} = o_p(n^{u_{p,k}} - 1)$ for all prime powers p^k dividing t_0 are calculated. The $u_{p,k}$ will be the degrees of certain rings in which the calculations of Section 5 will be done. The number $o^*_{p,k}$ of factors p in $n^{u_{p,k}} - 1$ may be used in the Lucas-Lehmer step of the algorithm.

(a) For $k = 1, \ldots, k_p$ (cf. (2.2)(c)), perform step (a1).

(a1) If $p^k = 2$, then put $i = 1$. Otherwise, if $p^k \neq 2$, let $n_{p,k} = n \bmod p^k$, and find by repeated multiplication the minimal i in $\{1, 2, \ldots, \lambda(p^k)/2\}$ such that $n^i_{p,k} \equiv \pm 1 \bmod p^k$ (cf. (2.2)(d)). If $n^i_{p,k} \equiv 1 \bmod p^k$, then put $u_{p,k} = i$. Otherwise, if $n^i_{p,k} \equiv -1 \bmod p^k$, then put $u_{p,k} = 2i$.

(b) If p is odd, or $n \equiv 1 \bmod 4$, put $\bar{k} = 1$ and $\bar{u} = u_{p,\bar{k}} = u_{p,1}$. If $p = 2$ and $n \equiv 3 \bmod 4$, put $\bar{k} = 2$ and $\bar{u} = u_{p,\bar{k}} = 2$. Put $c = 0$ and calculate $o^*_{p,\bar{k}}$ by performing step (b1) only once and step (b2) as long as $c = 0$.

In this step $o^*_{p,k}$, the number of factors p in $n^{u_{p,k}} - 1$, is determined. Since $n^{u_{p,k}} \equiv 1 \bmod p^k$, it follows that the number of factors p in $n^{u_{p,k}} - 1$ is at least k. Also, $o^*_{p,k} = \max\{o^*_{p,\bar{k}}, k\}$ for $\bar{k} \leq k \leq k_p$, and $o^*_{p,k} = 1$ for $1 \leq k < \bar{k}$, so $o^*_{p,k}$ can be calculated from $o^*_{p,\bar{k}}$. To determine $o^*_{p,\bar{k}}$ one first divides $n^{u_{p,\bar{k}}} - 1 = n^{\bar{u}} - 1$ by $p^{\bar{k}}$. This is done by first writing $n^{\bar{u}} - 1$ in base n, i.e., $n^{\bar{u}} - 1 = \sum_{i=0}^{\bar{u}-1} b_i \cdot n^i$, and by next sequentially dividing all coefficients b_i by $p^{\bar{k}}$ while keeping track of a carry c. The b_i now satisfy $\sum_{i=0}^{\bar{u}-1} b_i \cdot n^i = (n^{\bar{u}} - 1)/p^{\bar{k}}$.

(b1) First put $o^*_{p,\bar{k}} = \bar{k}$. Next put $b_i = n - 1$ for $i = 0, 1, \ldots, \bar{u} - 1$. After this, for $i = \bar{u} - 1, \ldots, 0$ in succession first put $c' = (c \cdot n + b_i) \bmod p^{\bar{k}}$, next put $b_i = \lfloor (c \cdot n + b_i)/p^{\bar{k}} \rfloor$, and finally replace c by c'.

(b2) For $i = \bar{u}-1, \ldots, 0$ in succession first put $c' = (c \cdot n + b_i) \bmod p$ and $b_i = \lfloor (c \cdot n + b_i)/p \rfloor$ and next replace c by c'. If $c = 0$ replace $o^*_{p,\bar{k}}$ by $o^*_{p,\bar{k}} + 1$.

Using the same method as in step (b1) the number $\sum_{i=0}^{\bar{u}-1} b_i \cdot n^i$ is divided by p as long as c, the remainder of the division, is equal to zero. The resulting $o^*_{p,\bar{k}}$ is the number of factors p of $n^{\bar{u}} - 1$.

(c) Put $o^*_{p,k} = 1$ for $1 \le k < \bar{k}$, and put $o^*_{p,k} = \max\{o^*_{p,\bar{k}}, k\}$ for $\bar{k} \le k \le k_p$. Put $u_0 = \operatorname{lcm}\{u_{p,k} : p^k \parallel t_0\} = \operatorname{lcm}\{u_{p,k_p}\}$ (cf. (2.2)(c)).

(3.5) Utilization of known factors.

Perform steps (a) and (b) for every known prime factor f of $n^\omega - 1$ with $\omega \in \mathbf{Z}_{>0}$ and either $\omega \notin \Omega$ or $f \nmid v_\omega$.

In the algorithm any prime factor f of $n^\omega - 1$ that is known is useful, since such a factor may be used in the Lucas-Lehmer part of the algorithm (cf. (5.3)). Therefore any known factor f will be stored, together with its multiplicity and ω. In the optimization step (Section 4) it will be decided which of these factors will be used. The set Ω will contain all values ω for which a factor is known (cf. (3.2)), w will be equal to $\operatorname{lcm}\{\omega : \omega \in \Omega\}$ and v_ω denotes the completely factored part of $\Phi_\omega(n)$, for $\omega \ge 1$, with $\omega \in \Omega$, where Φ_ω denotes the ω-th cyclotomic polynomial; that is, v_ω consists of the known primitive prime factors of $n^\omega - 1$: those that are not already in $n^i - 1$ for $i \mid \omega$ and $i \ne \omega$. Furthermore, v will be the product of v_ω for $\omega \in \Omega$ and finally, $\mathcal{F}_\omega$ denotes the set of prime factors found in $n^\omega - 1$. Note that in (3.2)(a1) we have found values for v_1 and v_2. All values and sets may be changed in step (4.5).

(a) Let $k = o_f(n^\omega - 1)$, and $c = o_f(\omega)$. Put $\bar{k} = 1$ if f is odd or $n \equiv 1 \bmod 4$, and $\bar{k} = 2$ if f is 2 and $n \equiv 3 \bmod 4$. Put $o^*_{f,\bar{k}} = k - c + \bar{k} - 1$, $o^*_{f,i} = 1$ for $1 \le i < \bar{k}$ and $o^*_{f,i} = \max\{o^*_{f,\bar{k}}, i\}$ if $\bar{k} \le i \le k$. Calculate $\bar{\omega} = \min\{i : i \mid \omega,\ f \mid (n^i - 1)\}$.

Calculating $o^*_{f,i}$ from $o^*_{p,\bar{k}}$ as well as calculating $\bar{\omega}$ is done in the same way as in (3.4). When calculating the values $v_{\omega'}$, one only has to update those values $v_{\omega'}$, with $\omega' = \bar{\omega} \cdot f^i$, with $i \ge 0$, since the number of factors f in $n^{\omega'} - 1$ only changes for these values of ω'.

(b) For $i = 0, \ldots, c$ in succession, put $\omega' = \bar{\omega} \cdot f^i$ and perform step (b1) if $\omega' \notin \Omega$, and step (b2) if $\omega' \in \Omega$.

Although $f^{k-c+i} \mid n^{\omega'} - 1$, one should not multiply v by f^{k-c+i} because then factors f would be counted more than once in v. The number of factors f in $\Phi_{\bar{\omega} \cdot f^{j_1}}(n)$, not in $\Phi_{\bar{\omega} \cdot f^{j_2}}(n)$ for $0 \le j_2 \le j_1$ is equal to $j_1 - j_2$. Only these factors f may be used by the Lucas-Lehmer part of the algorithm. Therefore one has to keep track of factors f which are due to $\Phi_{\omega'}(n)$ itself, and factors f which are due to $\Phi_{\omega''}(n)$ for some divisor ω'' of ω'.

(b1) Replace Ω by $\Omega \cup \{\omega'\}$, put $\mathcal{F}_{\omega'}$ equal to $\{f\}$, replace w by $\operatorname{lcm}(w, \omega')$, put $v_{\omega'} = f^j$, replace v by $v \cdot f^j$, where $j = \bar{k}$ if $i = 0$ and $j = 1$ otherwise.

If $\omega' \notin \Omega$ the set $\mathcal{F}_{\omega'}$ and the $v_{\omega'}$ have to be initialized. Furthermore the set Ω, and the integers w and v are updated.

(b2) If $f \nmid v_{\omega'}$ then replace $\mathcal{F}_{\omega'}$ by $\mathcal{F}_{\omega'} \cup \{f\}$, replace $v_{\omega'}$ by $v_{\omega'} \cdot f^j$, replace v by $v \cdot f^j$, where $j = \bar{k}$ if $i = 0$ and $j = 1$ otherwise.

4. OPTIMIZATION.

In this section we give a description of the optimization step. It should be noted that steps (4.2)–(4.6) are not performed sequentially, but are used while executing step (4.1).

(4.1) Main optimization step.

Select a value for $\bar{T}$ and perform steps (a) through (d).

The value $\bar{T}$ is the time one is prepared to spend on proving the primality of the number n.

In this step one tries to choose the values of $\mathcal{P}$, $\mathcal{Q}$, s, t, u, v, and w in such a way that the time needed to perform the algorithm is (close to) minimal; if this time exceeds $\bar{T}$, the algorithm will be aborted, since one cannot prove the primality within a reasonable amount of time. One should keep in mind that, roughly speaking, $\text{lcm}(s, v)$ should exceed $\sqrt{n}$; here s forms the Jacobi sum contribution and v the Lucas-Lehmer contribution.

(a) Put $B_2 = 1$. Set $\mathcal{L}^+$ and $\mathcal{L}^-$ equal to the empty set. For all primes l dividing $\text{lcm}(u_0, w)$ put $a_l = 0$, and define $\hat{e}_l$ by $\text{lcm}(u_0, w) = \prod_{l \text{ prime}} l^{\hat{e}_l}$, (cf. (3.4) and (3.5) for the definitions of u_0 and w). Perform step (4.2) for every prime power $l^{\hat{e}_l}$. Perform steps (a1) and (a2).

The variable B_2 indicates up to which bound factors have been searched for in step (4.6). It will be changed in step (4.6).

The sets $\mathcal{L}^+$ and $\mathcal{L}^-$ will be used to indicate which extensions have been found. If at least one extension of degree l^e that can be used in the algorithm for testing n has been found, then l^e will be put in $\mathcal{L}^+$. If no suitable extension of degree l^e has been found, then l^e will be put in $\mathcal{L}^-$.

The variable a_l is used to indicate whether or not step (4.2) has been performed for any power of the prime l. If $a_l \neq 0$ then certain conditions in step (4.2) have been checked for some power of l. For more information, we refer to step (4.2).

(a1) For all $t \mid t_0$ put $\mathcal{Q}_t$ equal to $\{q \in \mathcal{Q}_0 : q - 1 \mid t\}$ and put $\mathcal{T}_t$ equal to $\{(q, p, h) : h = o_p(u_{p,o_p(q-1)})$ with $q \in \mathcal{Q}_t$ and prime $p \mid q - 1\}$.

The set $\mathcal{Q}_t$ contains the prime factors of $e'_{i(t)}$, (cf. (2.2)(g)). We get a contribution q to s, if we perform Jacobi sum tests for all triples (q, p, h) in $\mathcal{T}_t$, where q is the conductor and $p^{o_p(q-1)}$ is the order of the character involved. This test can be performed in an extension of degree $u_{p,o_p(q-1)} = p^h \cdot u_{p,1}$. These sets are calculated in advance, since in step (4.3) one needs to compare the costs of using t and (subsets of) $\mathcal{Q}_t$ and $\mathcal{T}_t$ quickly. For all prime powers p^k dividing t_0 the $u_{p,k}$ have been defined as $\text{ord}(n \bmod p^k)$ (cf. (3.4)).

(a2) For all $q \in \mathcal{Q}_0$ calculate $c_1(q) = (\sum_{p|q-1} u^2_{p,o_p(q-1)})/\log_2(q)$. Order the elements $q \in \mathcal{Q}_0$ in such a way that $c_1(q)$ is decreasing (cf. III.(3.17)).

Given a value for t, a quick and reliable method is needed to delete the most expensive q's from $\mathcal{Q}_t$ (and all corresponding triples (q, p, h) from $\mathcal{T}_t$). The ordering of q's according to c_1 assumes that the costs for using q's are independent; in fact this is not true, since Jacobi sum tests for different conductor may be combined. Therefore c_1 is not guaranteed to give the optimal ordering, but experiments have shown that it hardly differs from this. This ordering is used in step (4.3), to reject most values for t as being too expensive. Possibly better, but more expensive ways of ordering are used in step (4.3).

(b) Let v and w be as found in (3.5). Find initial values $\mathcal{P}'$, $\mathcal{Q}'$, $\mathcal{T}'$, s', t', u', v', w', μ', and an initial estimate C' for the running time of the steps of the algorithm described in Sections 5 and 6, by performing step (4.3) with $C^* = -1$, $B^* = \lceil \sqrt{n}/v_1 \rceil$, $z^* = 0$, and $\mu^* = \frac{1}{2}$. Next put $\mathcal{P}' = \mathcal{P}^*$, $\mathcal{Q}' = \mathcal{Q}^*$, $\mathcal{T}' = \mathcal{T}^*$, $s' = s^*$, $t' = t^*$, $u' = u^*$, $v' = v_1$, $w' = 1$, and $\mu' = \frac{1}{2}$. Finally put $C' = C^*$.

In this step initial values for $\mathcal{P}'$, $\mathcal{Q}'$, $\mathcal{T}'$, s', t', u', v', w', and μ' are chosen in such a way, that it hardly takes any time to find them, and that the time to perform the steps of Sections 5 and 6 with these values is close to minimal. These values will be improved in the next steps of (4.1). Throughout the rest of step (4.1), the $\mathcal{P}'$, $\mathcal{Q}'$, $\mathcal{T}'$, s', t', u', v', w', and μ' will denote the values that give the minimal cost C' found so far.

Initially, the value C^* is negative to indicate that step (4.3) has not yet been performed.

The flag z^* is used to indicate which kind of optimization should be performed in (4.3). For further comments on this subject we refer to that step. The value μ' indicates which final trial division will be used in Section 6. During the final trial division it will be checked whether there exist divisors of n in the residue classes $r \equiv n^i \bmod \operatorname{lcm}(s', v')$ for $i = 1, \ldots, \operatorname{lcm}(t', w')$, where $\operatorname{lcm}(s', v') > n^{\mu'}$.

(c) Put $T = \min(\bar{T}, C')$ and put $P = T - T_0$, where T_0 is the minimal time needed to perform the rest of step (c). If $P > 0$, repeat for $\mu = \frac{1}{2}$ as well as for $\mu = \frac{1}{3}$ steps (c1) through (c6) until either $P < 0$ or $T < 0$ in step (c4); as soon as this happens, we jump to step (d). Otherwise, if $P \le 0$, perform step (4.3) with $C^* = C'$, $B^* = \lceil n^{\mu}/v_1 \rceil$, $z^* = 2$, and $\mu^* = \mu$.

In this step a further effort is made to optimize C'. It may be that if more factors are found in step (4.3), the total running time (including the time needed to find these factors) is less than the minimal running time found up till now.

The time left to be spent is denoted by T; if T_0 exceeds T, then optimization only consists of performing step (4.3)(b) for the current value of t'. The value for T_0 should be determined empirically.

(c1) First put $\hat{x} = \lceil n^{\mu}/v_1 \rceil$, and perform step (4.3) with $C^* = C'$, $B^* = \hat{x}$, $z^* = 1$, and $\mu^* = \mu$. Next put $f'(\hat{x}) = C^*$, $f''(\hat{x}) = 0$, $f(\hat{x}) = f'(\hat{x}) + f''(\hat{x})$. If $f(\hat{x}) < C'$, then put $C' = f(\hat{x})$ and put $\mathcal{P}' = \mathcal{P}^*$, $\mathcal{Q}' = \mathcal{Q}^*$, $\mathcal{T}' = \mathcal{T}^*$, $s' = s^*$, $t' = t^*$, $u' = u^*$, $v' = v_1$, $w' = 1$, and $\mu' = \mu^*$.

In this step (and steps (c2) through (c6)) an attempt is made to find a value $\tilde{x}$, for which the sum $f(\tilde{x})$ of the cost $f'(\tilde{x})$ of performing a Jacobi sum test with $s \approx \tilde{x}$, and the cost $f''(\tilde{x})$ of performing a Lucas-Lehmer test with $v \approx n^{\mu}/\tilde{x}$, is minimal. This is done by varying the amount of time spent on looking for more factors for the Lucas-Lehmer part and estimating the consequences. As soon as $f(\tilde{x}) < C'$, it is expected to be profitable to search for more factors in order to change $\mathcal{P}'$, $\mathcal{Q}'$, $\mathcal{T}'$, s', t', u', v', w', μ', and C'.

(c2) Put $W^{\#} = \max\{\operatorname{ord}(n \bmod p^k) : p \text{ prime}, p^k \parallel \operatorname{lcm}(t', v')\}$. Let A_1 be an approximate solution for A in the equation

$$\frac{A}{\log A} = \frac{C'}{(\log_2 n + W\#)} + \frac{B_2}{\log B_2}.$$

Let $\check{x}$ be an approximate solution for x in the equation

$$x \cdot v' \cdot \left(\frac{A_1 + B_2}{2}\right)^{W^{\#}(\log\log A_1 - \log\log B_2)} = n^{\mu}.$$

Perform (4.3) with $C^* = C'$, $B^* = \check{x}$, $z^* = 0$, and $\mu^* = \mu$, and next put $f'(\check{x}) = C^*$.

If we spend time equal to C' in looking for additional Lucas-Lehmer factors, we will find all prime factors up to the expected bound A_1 of $n^i - 1$, where $1 \le i \le W^{\#}$. If all these factors are used in a Lucas-Lehmer test, then a Jacobi sum test with $s \approx \check{x}$ would be sufficient to complete the primality proof. The time necessary to perform the Jacobi sum test is about the value of f' calculated here. The $\check{x}$ serves as lower bound for values of x that will be taken into consideration.

(c3) Select a value for ϵ. The machine dependent functions c_d and c_n must be known (cf. (4.4)). Find an approximate minimum $\tilde{x}$ between $\check{x}$ and $\hat{x}$ to the function

$$f''(x) + \left(\frac{f'(\hat{x}) - f'(\check{x})}{\log_2 \hat{x} - \log_2 \check{x}}\right) \cdot (\log_2 x - \log_2 \check{x}) + f'(\check{x}).$$

The function $f''(x)$ is approximated by

$$C_3(w') + \sum_{\substack{\omega \le W^{\#} \\ \omega \nmid \mathrm{lcm}(u^*, w')}} \omega^2 \log_2 n \cdot c_n(\log_2(n)) +$$
$$\left(\frac{A_2}{\log A_2} - \frac{B_2}{\log B_2}\right) \cdot (W^{\#} + c_d(\log_2(n), \log_2(M))),$$

where A_2 is an approximate solution for A in

$$x \cdot v' \cdot \left(\frac{A + B_2}{2}\right)^{W^{\#}(\log\log A - \log\log B_2)} = n^{\mu},$$

M is the largest integer representable in single precision, and C_3 is a function defined in (4.4). Perform step (c3a) until $\check{x} = \hat{x}$, but at most 5 times. Put $\tilde{x} = \check{x}$.

We minimize the sum of the cost $f''(x)$ of searching for Lucas-Lehmer factors (which is decreasing with increasing x) and the cost $f'(x)$ of finishing the primality proof using a Jacobi sum test with $s \approx x$, under the assumption that the latter is approximately linear in $\log x$.

For fixed t, the cost $f'(x)$ of the Jacobi sum test grows almost linearly as a function of $\log s$; changing to another value of t however, introduces a discontinuity. As a result the function f' will be almost piecewise linear in $\log x$. Here the assumption is made that no discontinuities on the interval $(\check{x}, \hat{x})$ exist; in (c3a) this assumption is verified.

The approximation for $f''(x)$ consists of a contribution $C_3(w')$ for generating the proper roots of unity (as in (4.4)(c)) for the known Lucas-Lehmer factors, an approximation for the costs of generating roots of unity for the Lucas-Lehmer factors expected to be found, and finally the costs of searching for these factors up to the bound A_2.

Notice that in the approximation of $f''(x)$ above, the first two terms are independent of x, so minimization of f in fact only includes minimization of the sum of the third term of $f''(x)$ and the linear approximation of the costs of the Jacobi sum test.

The value ϵ is the maximal relative error that is tolerated in f'. A suitable value for

ϵ is for instance 0.5. The machine dependent constant c_0^* is given by the ratio of the costs of division and multiplication of single precision integers. Minimization is done using the methods described in [15].

(c3a) Calculate $f'(\tilde{x})$ by performing (4.3) with $C^* = C'$, $B^* = \tilde{x}$, $z^* = 0$, and $\mu^* = \mu$, and by putting $f'(\tilde{x}) = C^*$. If

$$\left| \frac{f'(\tilde{x}) - f'(\check{x})}{\log_2(\tilde{x}) - \log_2(\check{x})} - \frac{f'(\tilde{x}) - f'(\hat{x})}{\log_2(\tilde{x}) - \log_2(\hat{x})} \right| < \epsilon$$

then put $\check{x} = \tilde{x}$ and $\hat{x} = \tilde{x}$. Otherwise, if

$$\frac{f'(\tilde{x}) - f'(\check{x})}{\log_2(\tilde{x}) - \log_2(\check{x})} > \frac{f'(\tilde{x}) - f'(\hat{x})}{\log_2(\tilde{x}) - \log_2(\hat{x})}$$

then put $\hat{x} = \tilde{x}$ and $\tilde{x} = (\hat{x} + \check{x})/2$. Otherwise, if

$$\frac{f'(\tilde{x}) - f'(\check{x})}{\log_2(\tilde{x}) - \log_2(\check{x})} < \frac{f'(\tilde{x}) - f'(\hat{x})}{\log_2(\tilde{x}) - \log_2(\hat{x})}$$

then put $\check{x} = \tilde{x}$ and $\tilde{x} = (\hat{x} + \check{x})/2$.

If f' is approximately linear in $\log x$ on the interval $(\check{x}, \hat{x})$, then f assumes its minimum at the initial value of $\tilde{x}$; if f' contains a discontinuity on the interval $(\check{x}, \hat{x})$, then we apply a few bisection steps in order to approximate the abscissa of this discontinuity from below.

(c4) First approximate $f''(\tilde{x})$ by calculating

$$C_3(w') + c_3^* \cdot \sum_{\substack{\omega \le W^{\#} \\ \omega \nmid \mathrm{lcm}(u^*, w')}} (\omega \log_2 n)^3 + c_0^* \cdot \left(\frac{A_2}{\log A_2} - \frac{B_2}{\log B_2} \right) \cdot (W^{\#} + \log_2 n),$$

where A_2 is an approximate solution for A in

$$\tilde{x} \cdot v' \cdot \left(\frac{A + B_2}{2} \right)^{W^{\#}(\log\log A - \log\log B_2)} = n^{\mu},$$

and where c_0^*, c_3^* are machine dependent constants (cf. c3)). Next put $f(\tilde{x}) = f'(\tilde{x}) + f''(\tilde{x})$.

Put P equal to $C' - f(\tilde{x})$. If $P > 0$ then first perform step (4.6) with $W^{\#} = \max\{\mathrm{ord}(n \bmod p^k) : p \text{ prime}, p^k \parallel \mathrm{lcm}(t', v')\}$, and with $B^{\#} = A_2$.

Next replace T by

$$T - c_0^* \cdot \left(\frac{A_2}{\log A_2} - \frac{B_2}{\log B_2} \right) \cdot (\log_2 n + W^{\#}).$$

If either $P \le 0$ or $T \le 0$, proceed with step (d).

The expected profit P of performing steps (c2)–(c4) is determined by calculating $P(\mathcal{P}', \mathcal{Q}', s', t', u', v', w', \mu')$, which gives the difference between the running time of the rest of the algorithm (calculated in step (4.4)) with the present values of $\mathcal{P}'$, $\mathcal{Q}'$, s', t', u', v', w', μ', and the sum of the expected time necessary to find better values $\mathcal{P}^*$, $\mathcal{Q}^*$, s^*, t^*, u^*, v^*, w^*, μ^*, and the expected time C^* to run the rest of the algorithm with these. If the expected profit is positive, it is expected to be beneficial to search for additional factors.

If the cost to use additional factors is larger than the cost to use the optimal values found up till now, the complete interval where profit could be found has been examined, and the optimization is terminated.

(c5) Perform step (4.3) with $C^* = C'$, $B^* = \tilde{x}$, $z^* = 1$, and $\mu^* = \mu$, and next put $\mathcal{P}' = \mathcal{P}^*$, $\mathcal{Q}' = \mathcal{Q}^*$, $\mathcal{T}' = \mathcal{T}^*$, $s' = s^*$, $t' = t^*$, $u' = u^*$, $\mu' = \mu^*$, and $f'(\tilde{x}) = C^*$. Next put $C' = f'(\tilde{x}) + f''(\tilde{x})$.

Using the new factors found in the previous step, the values of $\mathcal{P}'$, $\mathcal{Q}'$, $\mathcal{T}'$ s', t', u', v', w', and μ' are updated, as well as C'.

(c6) For all primes l dividing $\operatorname{lcm}(u', w')$, let $\hat{e}_l$ be such that $\operatorname{lcm}(u', w') = \prod_{l \text{ prime}} l^{\hat{e}_l}$. Put $a_l = 0$ for all primes $l \mid \operatorname{lcm}(u', w')$ with $l \notin \mathcal{L}^- \cup \mathcal{L}^+$. Next perform step (4.2) for every $l^{\hat{e}_l} \notin \mathcal{L}^- \cup \mathcal{L}^+$.

For $l^{\hat{e}_l} \notin \mathcal{L}^- \cup \mathcal{L}^+$ one has to find extensions that can be used in Sections 5 and 6. This can only happen for those $l^{\hat{e}_l} \mid w'$ with $l^{\hat{e}_l} \nmid u_0$, (cf. (4.1)(a)).

If $l \notin \mathcal{L}^- \cup \mathcal{L}^+$ then step (4.2) has not yet been performed for any power of l; this is indicated by putting $a_l = 0$.

(d) If $C' \le \bar{T}$, then put $\mathcal{P} = \mathcal{P}'$, $\mathcal{Q} = \mathcal{Q}'$, $\mathcal{T} = \mathcal{T}'$, $s = s'$, $t = t'$, $u = u'$, $v = v'$, $w = w'$, and $\mu = \mu'$, and proceed with Section 5; if $C' > \bar{T}$ the algorithm is aborted.

If the running time C' for the rest of the algorithm with the present values of $\mathcal{P}'$, $\mathcal{Q}'$, s', t', u', v', w', and μ' is "acceptable", i.e., at most $\bar{T}$, then the algorithm will be finished with these values. Otherwise the algorithm is aborted since the primality of n cannot be proved within a reasonable amount of time.

(4.2) Finding good extensions.

A value for $l^{\hat{e}_l}$ must have been specified. Perform steps (a) through (c).

For all prime powers $l^e \mid l^{\hat{e}_l}$ all entries in the table made in (2.3) that are useful in the test of n are retrieved, and the conductors are put in $\mathcal{M}^+(l^e)$; if there is at least one, l^e is put in the set $\mathcal{L}^+$, and else it is put in the set $\mathcal{L}^-$. If $l^e \in \mathcal{L}^-$, an extension of degree l^e can only be used in Sections 5 and 6 if it is created in step (5.1).

(a) If $a_l = 0$, put $e_0 = 0$ and perform step (a1). Otherwise, let $e_0 < \hat{e}_l$ be the largest e for which $l^e \in \mathcal{L}^- \cup \mathcal{L}^+$ and perform step (a2).

Initially, the sets $\mathcal{L}^+$ and $\mathcal{L}^-$ are empty. Step (a1) is performed only once per l.

(a1) Put $\mathcal{M}^+(l^e)$ equal to the empty set for $1 \le e \le \hat{e}_l$. For every prime m such that $m \in \mathcal{M}(l)$ (cf. (2.3)) and $n^{(m-1)/l} \not\equiv 1 \bmod m$ replace $\mathcal{M}^+(l)$ by $\mathcal{M}^+(l) \cup \{m\}$. Next, for every $e = 2, 3, \ldots, \hat{e}_l$ in succession, replace $\mathcal{M}^+(l^e)$ by $\{m \in \mathcal{M}^+(l^{e-1}) : l^e \mid m - 1\}$. Finally, for $1 \le e \le \hat{e}_l$ put l^e in $\mathcal{L}^+$ if $\mathcal{M}^+(l^e)$ is non-empty.

One checks whether the extensions that were precomputed in step (2.3) can be used for testing n; an extension of degree l^e and prime conductor m can be used if $n^{(m-1)/l} \not\equiv 1 \bmod m$ (cf. II.(4.8)). Note that this condition does not involve e.

(a2) Put $\mathcal{M}^+(l^e)$ equal to the empty set for $e_0+1 \le e \le \hat{e}_l$. For $e = e_0+1, e_0+2, \ldots, \hat{e}_l$ in succession, do the following. Replace $\mathcal{M}^+(l^e)$ by $\{m \in \mathcal{M}^+(l^{e-1}) : l^e \mid m-1\}$; if $\mathcal{M}^+(l^e)$ is non-empty put l^e in $\mathcal{L}^+$.

If the extension of degree l^{e_0} and prime conductor m can be used, then one can also use the extension of degree l^e and conductor m provided that $l^e \mid m-1$.

(b) If $l = 2$ perform step (b1), if l is odd perform step (b2).

In this step the case that the conductor is a power of the prime l is considered; one has to distinguish between the case that $l = 2$ and the case that l is odd.

(b1) If $a_2 = 0$ then perform step (b1a) through (b1c). Otherwise, if $a_2 > 0$, perform step (b1c).

If $a_2 = 0$ then step (4.2) has not yet been performed for any power of 2. Otherwise, if $a_2 > 0$ then step (4.2) has been performed for powers of 2, with a_2 as maximal value for $\hat{e}_l$. Finally, if $a_2 < 0$ then it has been checked that no extension of degree 2^e with prime power conductor can be used, since $n \not\equiv \pm 3 \bmod 8$.

(b1a) If $n \equiv \pm 3 \bmod 8$ put $a_2 = 1$, replace $\mathcal{M}^+(2)$ by $\mathcal{M}^+(2) \cup \{8\}$, and replace $\mathcal{L}^+$ by $\mathcal{L}^+ \cup \{2\}$. If $n \not\equiv \pm 3 \bmod 8$ put $a_2 = -1$.

In the case that $l = 2$ and $e \ge 1$, one can use an extension of degree 2^e with conductor $m = 2^{e+2}$ if $n \equiv \pm 3 \bmod 8$ holds.

One has to check this condition only once; if it holds one can add 2^{e+2} to $\mathcal{M}^+(2^e)$ for every e (cf. (2.4)(b3) and II.(4.10)).

(b1b) Check if $n \equiv 3 \bmod 4$. If this holds, replace $\mathcal{M}^+(2)$ by $\mathcal{M}^+(2) \cup \{4\}$, and replace $\mathcal{L}^+$ by $\mathcal{L}^+ \cup \{2\}$.

If $n \equiv 3 \bmod 4$ holds, one can use the extension $\mathbf{Q}(\zeta_4)$ with conductor $m = 4$ (cf. (2.4)(b3) and II.(4.10)).

(b1c) If $a_2 > 0$, then replace $\mathcal{M}^+(2^e)$ by $\mathcal{M}^+(2^e) \cup \{2^{e+2}\}$ and $\mathcal{L}^+$ by $\mathcal{L}^+ \cup \{2^e\}$ for $e = a_2+1, a_2+2, \ldots, \hat{e}_l$, and next put $a_2 = \hat{e}_l$.

(b2) If $a_l = 0$ then perform step (b2a). Otherwise, if $a_l > 0$ then perform step (b2b).

In the case that l is odd, one can use an extension of degree l^e with conductor $m = l^{e+1}$ if $n^{l-1} \not\equiv 1 \bmod l^2$ holds.

If $a_l = 0$ then step (4.2) has not yet been performed for any power of l. Otherwise, if $a_l > 0$ then step (4.2) has been performed for powers of l, with a_l as maximal value for $\hat{e}_l$. Finally, if $a_l < 0$ then it has been checked that no extension of degree l^e with conductor l^{e+1} can be used, since $n^{l-1} \equiv 1 \bmod l^2$.

(b2a) Check whether $n^{l-1} \not\equiv 1 \bmod l^2$; if this holds, replace $\mathcal{M}^+(l^e)$ by $\mathcal{M}^+(l^e) \cup \{l^{e+1}\}$ and $\mathcal{L}^+$ by $\mathcal{L}^+ \cup \{l^e\}$ for $e = 1, \ldots, \hat{e}_l$. Next put $a_l = \hat{e}_l$. Otherwise, if $n^{l-1} \equiv 1 \bmod l^2$, put $a_l = -1$.

One has to check this condition only once; if it holds one can add l^{e+1} to $\mathcal{M}^+(l^e)$ for every e.

(b2b) For $e = a_l+1, \ldots, \hat{e}_l$ replace $\mathcal{M}^+(l^e)$ by $\mathcal{M}^+(l^e) \cup \{l^{e+1}\}$, and next put $a_l = \hat{e}_l$.

If $a_l > 0$ then $n^{l-1} \not\equiv 1 \bmod l^2$, so one can add l^{e+1} to $\mathcal{M}^+(l^e)$.

(c) For every e such that $1 \le e \le \hat{e}_l$ and $\mathcal{M}^+(l^e)$ is empty, put l^e in $\mathcal{L}^-$ and do the following. Let e^* be the smallest e for which $\mathcal{M}^+(l^e)$ is empty. If $e^* = 1$ check whether n is an l^{th} power; if this is the case the primality test is terminated. If n is not an l^{th} power, for $e = e^*, \ldots, \hat{e}_l$ in succession find the smallest prime m such that both $l^e \mid m-1$ and $n^{(m-1)/l} \not\equiv 1 \bmod m$, and replace $\mathcal{M}^+(l^{e'})$ by $\mathcal{M}^+(l^{e'}) \cup \{m\}$ for $1 \le e' \le e$.

If n is an l-th power, then one will never succeed in finding a conductor with the correct properties. If this is not the case, then there exist prime conductors m with $n^{(m-1)/l} \not\equiv 1 \bmod m$. In practice, these m are not very hard to find.

(4.3) Finding good sets $\mathcal{P}^*$, $\mathcal{Q}^*$, and $\mathcal{T}^*$ and good values for s^*, t^*, and u^*. Values for B^*, C^*, z^*, and μ^* must have been specified. Select values for b_1 and b_2.

If $C^* < 0$ and $z^* = 0$ then perform step (a). If $z^* = 2$, then perform step (b) with $\tilde{t} = t'$. In all other cases perform step (b), first for $\tilde{t} = t'$ and next for all other divisors $\tilde{t}$ of t_0 satisfying $e'_{i(\tilde{t})} \ge B^*$ and $c_5(e'_{i(\tilde{t})}, \tilde{t}, \operatorname{ord}(n \bmod \tilde{t}), 1, 1, \mu^*) < C^*$ (cf. (4.4)(e)).

Given values for B^*, C^* (representing the minimal costs found up till now), and z^*, values for $\mathcal{P}^*$, $\mathcal{Q}^*$, $\mathcal{T}^*$, s^*, t^*, u^* are chosen in this step in such a way, that the costs C^* to perform Jacobi sum tests for these particular values is (close to) minimal and such that $s^* \ge B^*$. The flag z^* is used to indicate if the (expensive) step (b2) should be performed after performing step (b1). The values b_1 and b_2 are used in steps (a) and (b2). Possible values are $b_1 = 1.05$ and $b_2 = 1.05$. For further comments we refer to steps (a) and (b2).

(a) Put t^* equal to the value t in the table made in step (2.2)(g) for which $e'_{i(t^*)}$ is equal to $\min\{e'_{i(t)} : e'_{i(t)} > b_1 \cdot B^*\}$. Next put $\mathcal{P}^* = \{p : p \mid t^*\}$, $\mathcal{Q}^* = \mathcal{Q}_{t^*}$, $s^* = \prod_{q \in \mathcal{Q}^*} q$, $u^* = \operatorname{lcm}\{u_{p,k} : p^k \parallel t^*\}$ (cf. (2.4)), and $\mathcal{T}^* = \mathcal{T}_{t^*}$ (cf. (4.1)(a1)). Finally, calculate $C^* = C(\mathcal{P}^*, \mathcal{Q}^*, \mathcal{T}^*, s^*, t^*, u^*, v_1, 1, \mu^*)$ by performing step (4.4) with $\tilde{\mathcal{P}} = \mathcal{P}^*$, $\tilde{\mathcal{Q}} = \mathcal{Q}^*$, $\tilde{\mathcal{T}} = \mathcal{T}^*$, $\tilde{s} = s^*$, $\tilde{t} = t^*$, $\tilde{u} = u^*$, $\tilde{v} = v_1$, $\tilde{w} = 1$, and $\tilde{\mu} = \mu^*$.

If $C^* < 0$, a first estimate for the running time, C^*, and initial values for s^*, t^*, u^*, $\mathcal{P}^*$, $\mathcal{Q}^*$ will be found in this step. It is known that the time needed to perform the steps of Sections 5 and 6 with this particular choice for s^*, t^*, u^*, $\mathcal{P}^*$, and $\mathcal{Q}^*$ may happen to be not so very close to the minimum. Its main purpose is to find very quickly an upper bound for the running time needed to perform the steps of Sections 5 and 6. In this way we can speed up the other optimization steps of (4.3). This is the case because some parts of the steps of Sections 5 and 6 may be even too expensive for particular values of $\tilde{t}$. These values can then be rejected, without doing any expensive optimization steps.

For fairly small n however, the choices made in this step may be the final choices, since any choice which gives approximately the minimum will be fine.

Experiments have shown that values of $\tilde{t}$ with $e'_{i(\tilde{t})} \approx B^*$ do not give a very good approximation for the minimum time needed for the steps of Sections 5 and 6. Better values of $\tilde{t}$ have values of $e'_{i(\tilde{t})}$ which seem to be at least some offset times the specified value of B^*. The value b_1 is equal to the offset needed to get a better value for $\tilde{t}$.

(b) First put $\tilde{\mathcal{Q}} = \mathcal{Q}_{\tilde{t}}$, $\tilde{\mathcal{P}} = \{p \text{ prime } : p \mid \tilde{t}\}$, $\tilde{\mathcal{T}} = \mathcal{T}_{\tilde{t}}$ (cf. (3.4)), $\tilde{s} = \prod_{q \in \tilde{\mathcal{Q}}} q$, $\tilde{u} = \operatorname{ord}(n \bmod \tilde{t})$. If $z^* \geq 1$ put $C^{**} = C^*$ and perform steps (b1) and (b2). Otherwise, if $z^* = 0$, perform step (b1).

In this step one tries to find the best values for $\mathcal{P}^*$, $\mathcal{Q}^*$, $\mathcal{T}^*$, s^*, t^*, and u^*. This is done by checking for all values $\tilde{t} \mid t_0$ if a subset of $\mathcal{Q}_{\tilde{t}}$ containing the least expensive q's results in a better value for C^*. Since it is easy to estimate the time needed to perform the final trial division, this is used as criterion to reject too expensive values (cf. (6.1), (4.4)(e)).

(b1) Put $\hat{\mathcal{Q}} = \tilde{\mathcal{Q}}$, $\hat{\mathcal{P}} = \tilde{\mathcal{P}}$, $\hat{\mathcal{T}} = \tilde{\mathcal{T}}$, $\hat{s} = \tilde{s}$, $\hat{t} = \tilde{t}$, and $\hat{u} = \tilde{u}$, $\hat{\mu} = \tilde{\mu}$, and perform steps (b1a) and (b1b).

Given initial sets $\tilde{\mathcal{P}}$, $\tilde{\mathcal{Q}}$, and $\tilde{\mathcal{T}}$, and initial values for $\tilde{s}$, $\tilde{t}$, $\tilde{u}$, $\tilde{v}$, and $\tilde{w}$, one tries to find subsets $\hat{\mathcal{Q}}$ and $\hat{\mathcal{T}}$ of $\mathcal{Q}$ and $\mathcal{T}$ respectively, and $\hat{s} \mid s$, such that the running time to perform the Jacobi-sum tests and the final trial division for these values is (possibly) smaller than the running time for the present values, while $\hat{s}$ is still large enough.

Since one might perform two optimization steps, a copy has to be made of all values in order to start the second optimization step with the same values. In the first optimization step all $q \in \mathcal{Q}$ are weighted according to $c_1(q) = (\sum_{p \mid q-1} u^2_{p,o_p(q-1)})/\log_2(q)$ (cf. (4.1)(a2)). So the q with $c_1(q) = \max\{c_1(q)\}$ is regarded as being the most expensive q.

(b1a) For q_i, with $q_i \in \tilde{\mathcal{Q}}$, $i = 1, 2, \ldots$ in succession, if $s/q_i \geq B^*$, then replace $\tilde{\mathcal{Q}}$ by $\tilde{\mathcal{Q}} \backslash \{q_i\}$, $\tilde{\mathcal{T}}$ by $\tilde{\mathcal{T}} \backslash \{(q_i, p, h) : p \mid q_i - 1, h = o_p(u_{p,o_p(q_i-1)})\}$, and $\tilde{s}$ by $\tilde{s}/q_i$.

In this step the q_i's were ordered according to step (4.1)(a2).

(b1b) Calculate $\tilde{C} = C(\tilde{\mathcal{P}}, \tilde{\mathcal{Q}}, \tilde{\mathcal{T}}, \tilde{s}, \tilde{t}, \tilde{u}, 1, 1, \tilde{\mu})$ by performing step (4.4). If $\tilde{C} < C^*$ or $\tilde{C} = C^*$ and $s^* < \tilde{s}$, replace $\mathcal{Q}^*$, $\mathcal{P}^*$, $\mathcal{T}^*$, s^*, t^*, u^*, μ^*, and C^* by $\tilde{\mathcal{Q}}$, $\tilde{\mathcal{P}}$, $\tilde{\mathcal{T}}$, $\tilde{s}$, $\tilde{t}$, $\tilde{u}$, $\tilde{\mu}$, and $\tilde{C}$ respectively.

(b2) If $\tilde{C} < b_2 \cdot C^{**}$ or $C^{**} < 0$ then put $\tilde{\mathcal{Q}} = \hat{\mathcal{Q}}$, $\tilde{\mathcal{P}} = \hat{\mathcal{P}}$, $\tilde{\mathcal{T}} = \hat{\mathcal{T}}$, $\tilde{s} = \hat{s}$, $\tilde{t} = \hat{t}$, and $\tilde{u} = \hat{u}$, $\tilde{\mu} = \hat{\mu}$ and perform steps (b2a) and (b2b).

In the second optimization step all $q \in \hat{\mathcal{Q}}$ are weighted according to $c_2(q) = C_q/\log_2(q)$, were

$$C_q = C(\tilde{\mathcal{Q}}, \tilde{\mathcal{P}}, \tilde{\mathcal{T}}, \tilde{s}, \tilde{t}, \tilde{u}, 1, 1, \tilde{\mu}) - C(\tilde{\mathcal{Q}} \backslash \{q\}, \tilde{\mathcal{P}}, \tilde{\mathcal{T}} \backslash \{(q, \tilde{p}, \tilde{h}) \in \tilde{\mathcal{T}}\}, \tilde{s}/q, \tilde{t}, \tilde{u}, 1, 1, \tilde{\mu}).$$

That is, C_q is the additional cost of performing the necessary Jacobi sum test for conductor q, if all other tests in $\tilde{\mathcal{T}}$ have been done.

This optimization step is more expensive than the one described in step (b1), and is only performed when indicated by the flag z^*. If the time needed to perform the steps of Sections 5 and 6 with the present values for $\tilde{\mathcal{P}}$, $\tilde{\mathcal{Q}}$, $\tilde{\mathcal{T}}$, $\tilde{s}$, $\tilde{t}$, and $\tilde{u}$, found in step (b1), is close to the minimum up till now (within a margin determined by a blow-up factor b_2), or if $C^{**} < 0$ (indicating that no initial values of s^*, t^*, u^*, $\mathcal{P}^*$, $\mathcal{Q}^*$, and $\mathcal{T}^*$ have been found) one tries this possibly better (but more expensive) optimizing procedure to find the minimum.

(b2a) For all $q \in \tilde{\mathcal{Q}}$ let C_q be defined as the difference between $C(\tilde{\mathcal{Q}}, \tilde{\mathcal{P}}, \tilde{\mathcal{T}}, \tilde{s}, \tilde{t}, \tilde{u}, 1, 1, \tilde{\mu})$ and $C(\tilde{\mathcal{Q}} \backslash \{q\}, \tilde{\mathcal{P}}, \tilde{\mathcal{T}} \backslash \{(q, \tilde{p}, \tilde{h}) \in \tilde{\mathcal{T}}\}, \tilde{s}/q, \tilde{t}, \tilde{u}, 1, 1, \tilde{\mu})$. Perform steps (b2a1) through (b2a3) as long as there exists a $q \in \tilde{\mathcal{Q}}$ with $C_q > 0$ and $\tilde{s}/q \geq B^*$.

(b2a1) For all $q \in \tilde{\mathcal{Q}}$ calculate C_q and $c_2(q) = C_q/\log_2(q)$ by performing step (4.4).

(b2a2) Find $\bar{q}$ such that $c_2(\bar{q}) = \max\{c_2(q) : q \in \tilde{\mathcal{Q}}, C_q \neq 0, \operatorname{lcm}(\tilde{s}/q, \tilde{v}) \geq B^*\}$.

(b2a3) Replace $\tilde{\mathcal{Q}}$ by $\tilde{\mathcal{Q}}\backslash\{\bar{q}\}$, $\tilde{\mathcal{T}}$ by $\tilde{\mathcal{T}}\backslash\{(\bar{q}, p, h) : p \mid \bar{q}-1, h = o_p(u_{p,o_p(\bar{q}-1)})\}$, and $\tilde{s}$ by $\tilde{s}/\bar{q}$. Finally replace $\tilde{C}$ by $\tilde{C} - C_{\bar{q}}$.

(b2b) If $\tilde{C} < C^*$ or $\tilde{C} = C^*$ and $s < \tilde{s}$, replace $\mathcal{Q}^*$, $\mathcal{P}^*$, $\mathcal{T}^*$, s^*, t^*, u^*, μ^*, and C^* by $\tilde{\mathcal{Q}}$, $\tilde{\mathcal{P}}$, $\tilde{\mathcal{T}}$, $\tilde{s}$, $\tilde{t}$, $\tilde{u}$, $\tilde{\mu}$, and $\tilde{C}$ respectively.

(4.4) Running times.

Values for $\tilde{\mathcal{P}}, \tilde{\mathcal{Q}}, \tilde{\mathcal{T}}, \tilde{s}, \tilde{t}, \tilde{u}, \tilde{v}$, $\tilde{w}$, and $\tilde{\mu}$ must have been specified. The machine dependent functions c_m, c_d, c_n and c_f must be known. Perform steps (a) through (f).

On input $\tilde{\mathcal{P}}, \tilde{\mathcal{Q}}, \tilde{\mathcal{T}}, \tilde{s}, \tilde{t}, \tilde{u}, \tilde{v}, \tilde{w}$, in this step an estimate of the running time $C(\tilde{\mathcal{P}}, \tilde{\mathcal{Q}}, \tilde{\mathcal{T}}, \tilde{s}, \tilde{t}, \tilde{u}, \tilde{v}, \tilde{w})$ of the steps of Sections 5 and 6 is computed.

The function $c_m(a)$ denotes the time needed to perform a multiplication of two integers of size (binary logarithm) a. The function $c_d(a, b)$ denotes the time needed to perform a division of an integer of size a by an integer of size b. The function $c_n(a)$ denotes the time needed to perform a multiplication of two integers modulo n, where $\lceil \log_2(n) \rceil = a$ and finally the function $c_f(a)$ denotes the time needed to perform the final trial division for n with $\log_2(n) = a$ if $\mu = \frac{1}{3}$ for one residue class. Each function will be an approximation of the time needed to perform the operation averaged over several cases of n; it should be determined empirically and will be specified in Chapter VI for some machines.

(a) Put

$$C_1 = \sum_{l^{\hat{e}_l} \in \mathcal{L}^-} c_1(n, l^{\hat{e}_l}, m_l),$$

with $c_1(n, l^{\hat{e}_l}, m_l) = (l^{3\hat{e}_l} + m_l^2 \cdot l^{\hat{e}_l}) \cdot c_n(\log_2(n))$, where m_l is the smallest $m \in \mathcal{M}^+(l^{\hat{e}_l})$.

This is the time needed to perform step (5.1), i.e., the time needed to compute additional Galois extensions. These extensions only have to be computed for prime powers $l^{\hat{e}_l} \in \mathcal{L}^-$, since for all $l^{\hat{e}_l} \in \mathcal{L}^+$, the extensions have been generated in advance in step (2.3). The time needed for this step is dominated by the time needed for the operations in step (5.1)(b8) and (5.1)(b9).

(b) Put

$$C_2 = \sum_{l^{\hat{e}_l} \| \operatorname{lcm}(\tilde{u}, \tilde{w})} c_2(n, l^{\hat{e}_l}),$$

where $c_2(n, l^{\hat{e}_l}) = (l^{\hat{e}_l})^3 \cdot c_n(\log_2(n))$.

This is the time needed to perform step (5.2), i.e., the time needed to select cyclic rings and to create the transition matrices. The time needed for this step is dominated by the time needed for the operations in step (5.2)(b4).

(c) Put $c = 0$ and for each $\omega \in \Omega$ with $\omega \mid \tilde{w}$ put T_ω equal to $\max\{\frac{p}{p-1} : p \mid v_\omega\}$. First perform step (c1) as long as there exists an $\omega \in \Omega$ with $\omega \mid \tilde{w}$ such that $T_\omega > 0$. Next put $C_3 = c$.

This is an estimate of the time needed to perform step (5.3), i.e., the time needed to calculate cyclotomic extensions. The time needed for this step is dominated by the time needed for the operations in steps (5.3)(d3) and (5.3)(d4). A more accurate estimate could be calculated by dynamic programming techniques.

(c1) Put $\bar{\omega} = \max\{\omega \in \Omega,\ \omega \mid \tilde{w} : T_\omega > 0\}$. Replace c by $c + T_{\bar{\omega}} \cdot (2\bar{\omega}^3 \cdot \log_2 v_\omega + \bar{\omega}^2 \cdot \log_2 n) \cdot c_n(\log_2(n))$, and replace for all $\omega \in \Omega$ with $\omega \mid \bar{\omega}$ the value T_ω by $T_\omega - T_{\bar{\omega}}$ (cf. V.(4.3)).

(d) Perform (4.5) to compute $c_4(\tilde{\mathcal{T}})$, and set $C_4 = c_4(\tilde{\mathcal{T}}) \cdot \log_2 n \cdot c_n(\log_2(n))$.

This is the time needed to perform step (5.4), i.e., the time needed to perform the Jacobi sum tests. The time needed for this step is dominated by the time needed for the operations in step (5.4)(d).

(e) Put $C_5 = c_5(\tilde{s}, \tilde{t}, \tilde{u}, \tilde{v}, \tilde{w}, \tilde{\mu})$, where

$$c_5(\tilde{s}, \tilde{t}, \tilde{u}, \tilde{v}, \tilde{w}, \tilde{\mu}) = t_f \cdot (1 + \sqrt{n}/s_f) \cdot (c_m(\log_2(s_f)) + c_d(2\log_2(s_f), \log_2(s_f))),$$

if $\tilde{\mu} = \frac{1}{2}$, and

$$c_5(\tilde{s}, \tilde{t}, \tilde{u}, \tilde{v}, \tilde{w}, \tilde{\mu}) = t_f/2 \cdot c_f(\log_2(n))$$

if $\tilde{\mu} = \frac{1}{3}$, with $t_f = \operatorname{lcm}(\tilde{t}, \tilde{u}, \tilde{w})$ and $s_f = \operatorname{lcm}(\tilde{v}, \tilde{s} \cdot \prod_{p \mid \tilde{t}} p^{o_p(n^{\tilde{u}}-1) - o_p(\tilde{s})})$.

This is the time needed to perform step (6.1), i.e., the time needed to perform the final trial divisions.

(f) Put $C(\tilde{\mathcal{P}}, \tilde{\mathcal{Q}}, \tilde{\mathcal{T}}, \tilde{s}, \tilde{t}, \tilde{u}, \tilde{v}, \tilde{w}, \tilde{\mu}) = C_1 + C_2 + C_3 + C_4 + C_5$.

(4.5) Finding an optimal set of combined Jacobi sum tests.

The set $\tilde{\mathcal{T}}$ must have been specified.

Let $\bar{\mathcal{Q}} = \{q : \exists p, h \text{ with } (q,p,h) \in \tilde{\mathcal{T}}\}$, $\bar{\mathcal{P}} = \{p : \exists q, h \text{ with } (q,p,h) \in \tilde{\mathcal{T}}\}$, and $\bar{\mathcal{H}}_p = \{h : \exists q \text{ with } (q,p,h) \in \tilde{\mathcal{T}}\}$ for all $p \in \bar{\mathcal{P}}$.

Assume that the set $\tilde{\mathcal{T}}$ of triples (q,p,h) is given, where each triple (q,p,h) represents a Jacobi sum test (cf. (5.4)) for a character χ with $\operatorname{cond}(\chi) = q$, $\operatorname{ord}(\chi) = p^{o_p(q-1)}$, and $h = o_p(\operatorname{ord}(n \bmod p^{o_p(q-1)}))$. In this step a method is described that determines how these tests can be combined in an optimal way. A combination consists of a subset $\mathcal{S} \subset \tilde{\mathcal{T}}$ of triples, $\mathcal{S} = \cup_i\{(q_i, p_i, h_i)\}$ such that $p_i \neq p_j$ for $(q_i, p_i, h_i), (q_j, p_j, h_j) \in \mathcal{S}$ with $i \neq j$ (cf. II.(8.9) and III.(3.12)). The optimal set $\mathbf{S}_{\tilde{\mathcal{T}}}$ of such combinations $\mathcal{S}$ is found, i.e., the set of combinations for which the cost of performing step (5.4) with this $\tilde{\mathcal{T}}$ is minimal. This cost will be calculated as well, and is denoted by $c_4(\tilde{\mathcal{T}})$.

This is done as follows. Each Jacobi sum test, represented by a triple (q,p,h), involves an exponentiation in an extension of degree $u_{p,1} \cdot p^h$, where the exponent is roughly of the same magnitude as n. The time needed to do such an exponentiation is proportional to $(u_{p,1} \cdot p^h)^2 \cdot \log_2^3(n)$. With hardly any extra work, one is able to perform a number of Jacobi sum tests at the same time, thereby saving much time. Two (or more) Jacobi sum tests can be done together in an extension of degree which is the least common multiple of the original degrees. If the degree for one test divides that for another test, both can be done in the extension of the largest degree, in about the same time as performing the Jacobi sum test in the largest extension. Thus one can perform the Jacobi sum tests in extensions of degrees dividing u together in (almost) the same time as that for the single test in degree u itself. Performing tests together can however only be done if the orders of the characters involved are relatively prime, i.e., the primes p of the triples representing these tests should be relatively prime. These observations suggests a greedy strategy: find the largest degree, and next find all degrees dividing this degree (and relatively prime orders), and perform the tests together. As shown in III.3 this will give (under reasonable assumptions, cf. III.(2.17) and III.(3.1)) an optimal solution.

(a) For all $p \in \bar{\mathcal{P}}$ and all $h \in \bar{\mathcal{H}}_p$, let $d_{p,h} = \#\{q : q \in \bar{\mathcal{Q}}, (q,p,h) \in \tilde{\mathcal{T}}\}$. Put $\mathbf{S}_{\tilde{\mathcal{T}}}$ equal to the empty set, and $c_4(\tilde{\mathcal{T}}) = 0$. Put $b = 0$. Perform steps (a1) and (a2) as long as $d_{p,h} \neq 0$ for some $p \in \bar{\mathcal{P}}$ and some $h \in \bar{\mathcal{H}}_p$.

Each Jacobi sum test is represented by a triple (q,p,h), and consists of an exponentiation in an extension of degree $u_{p,1} \cdot p^h$, where the exponent is roughly of the same magnitude as n (and with $u_{p,1}$ as in (3.4)(b)). Two such tests will be combined only if the degree in which one extension has to be performed divides the degree of the other extension. The value $d_{p,h}$ gives the number of Jacobi sum tests left for a character of order p^k (for some k) to be done in an extension of degree $u_{p,1} \cdot p^h$.

(a1) Let $\bar{u} = \max\{u_{p,1} \cdot p^h : d_{p,h} \neq 0,\ p \in \bar{\mathcal{P}},\ h \in \bar{\mathcal{H}}_p\}$. Find $\bar{p}$ and $\bar{h}$ such that $\bar{u} = u_{\bar{p},1} \cdot \bar{p}^{\bar{h}}$. If more than one choice is possible, any choice with $u_{\bar{p},1} \cdot \bar{p}^{\bar{h}} = \bar{u}$ will do; to increase the speed of this step one can take a choice with $d_{p,h}$ maximal. Put $\bar{d} = d_{\bar{p},\bar{h}}$. Let $\bar{\mathcal{Q}}' = \{q : (q,\bar{p},\bar{h}) \in \tilde{\mathcal{T}}\}$, and order the elements $q_i \in \bar{\mathcal{Q}}'$ such that $\{q_i\}_{i=1}^{\bar{d}}$ is increasing. Put $\mathcal{S}_{b+i} = \{(q_i,\bar{p},\bar{h}) : (q_i,\bar{p},\bar{h}) \in \tilde{\mathcal{T}},\ (q_i,\bar{p},\bar{h}) \notin \mathcal{S}_f,\ 0 \leq f < b+i\}$, for $1 \leq i \leq \bar{d}$. For all $p \in \bar{\mathcal{P}}$ for which $p \neq \bar{p}$, first put $j = 0$ and next perform step (a1a).

In this step one finds all triples $(q,\bar{p},\bar{h})$ such that the degree $u_{\bar{p},1} \cdot \bar{p}^{\bar{h}}$ in which the Jacobi sum test has to be performed is maximal. All of these $\bar{d}$ tests have to be put in different combinations $\mathcal{S}_{b+i}$ because the p's are the same.

(a1a) Find $\tilde{h} \in \bar{\mathcal{H}}_p$ such that $\tilde{h} = \max\{h : d_{p,h} \neq 0 \text{ and } u_{p,1} \cdot p^h \mid \bar{u}\}$. If such a $\tilde{h}$ can be found let $d = \min\{d_{p,\tilde{h}},\ \bar{d} - j\}$ and let $\bar{\mathcal{Q}}' = \{q : (q,p,\tilde{h}) \in \tilde{\mathcal{T}}\}$, and order the elements $q_i \in \bar{\mathcal{Q}}'$ such that $\{q_i\}_{i=1}^{d}$ is increasing. For $1 \leq i \leq d$, replace $\mathcal{S}_{b+j+i}$ by $\mathcal{S}_{b+j+i} \cup \{(q_i,p,\tilde{h}) : (q_i,p,\tilde{h}) \in \tilde{\mathcal{T}}\}$, replace $d_{p,\tilde{h}}$ by $d_{p,\tilde{h}} - d$ and j by $j + d$. Repeat this step if $j < \bar{d}$ and $\{(q,p,h) : (q,p,h) \in \tilde{\mathcal{T}},\ d_{p,h} \neq 0,\ h < \tilde{h} \text{ and } u_{p,1} \cdot p^h \mid \bar{u}\}$ is not empty.

In this step as many tests as possible with degrees dividing $u_{\bar{p},1} \cdot \bar{p}^{\bar{h}}$ are added to the combinations $\mathcal{S}_{b+i}$ created in step (a1).

(a2) Replace $\mathbf{S}_{\tilde{\mathcal{T}}}$ by $\mathbf{S}_{\tilde{\mathcal{T}}} \cup \{\mathcal{S}_{b+i}\}$ for $1 \leq i \leq \bar{d}$, and increase $c_4(\tilde{\mathcal{T}})$ by $\bar{d} \cdot \bar{u}^2$. Replace b by $b + \bar{d}$ and replace $\bar{d}$ by zero.

At this point, no triple (q,p,h) with degree dividing $u_{\bar{p},1} \cdot \bar{p}^{\bar{h}}$ can be found. Therefore the combinations $\mathcal{S}_{b+i}$ are added to $\mathbf{S}_{\tilde{\mathcal{T}}}$. The time needed to perform a combination of tests is dominated by the time to do the exponentiation, which takes place in an extension of the maximal degree $(u_{p,1}p^h)$, and takes about $(u_{p,1} \cdot p^h)^2 \cdot (\log_2 n)^3$ operations.

(4.6) Trial division for Lucas-Lehmer step.

The positive integers $W^\#$ and $B^\#$ must have been specified.

Set a new trial division bound $B_1 = \min(B_0, \sqrt{n}, B^\#)$ (cf. (2.1)). First perform steps (a) through (d), and next put $B_2 = B_1$.

In this step an attempt is made to increase the value of v (cf. (3.2) and (3.5)) by finding factors of $n^\omega - 1$ for small ω up to the bound B_1. The choice made in step (2.1) implies an upper bound for B_1. So far all prime factors up to B_2 of $n^\omega - 1$ have been found for $\omega \le W^\#$; initially we have $B_2 = 1$. By v_ω is denoted the product of the prime power factors of $n^\omega - 1$, not dividing $n^{\omega'} - 1$ for any $\omega' \mid \omega$ with $\omega' < \omega$. Furthermore, $o^*_{p,k} = o_p(n^{\mathrm{ord}(n \bmod p^k)} - 1)$ (cf. (3.4) and (3.5)).

(a) For all $\omega \le W^\#$ and for $i = 0, 1, \ldots, \omega - 1$ put $r_{\omega,i}$ equal to $n - 1$.

In this step one writes $n^\omega - 1$ in base n, i.e., $n^\omega - 1 = \sum_{i=0}^{\omega-1} r_{\omega,i} \cdot n^i$.

(b) Perform step (b1) for all $\omega \le W^\#$.

(b1) If $\omega \in \Omega$ then divide $\sum_{i=0}^{\omega-1} r_{\omega,i} \cdot n^i$, which is initially $n^\omega - 1$, by v_ω. This can be done as follows. Put $c = 0$, and for $i = \omega - 1, \ldots, 0$ in succession put $c' = (c \cdot n + r_{\omega,i}) \bmod v_\omega$, put $r_{\omega,i} = \lfloor (c \cdot n + r_{\omega,i})/v_\omega \rfloor$, and replace c by c'. If $\omega \notin \Omega$ then put $\mathcal{F}_\omega$ equal to the empty set.

If $\omega \in \Omega$, then factors of $n^\omega - 1$ have already been found. The product of all the factors of $n^\omega - 1$, which is v_ω, will be divided out of $\sum_{i=0}^{\omega-1} r_{\omega,i} \cdot n^i$ This is done in this step, and is basically the same as step (3.4)(b). For further comments, we refer to this step.

(c) If $2 \le B_2$ continue with step (d). Otherwise, if $B_2 < 2 \le B_1$, let $o_2(n-1) = a$, let $o_2(n+1) = b$, and do the following. If $2^a \not| v_1$ then replace v by $v \cdot 2^a$, v_1 by $v_1 \cdot 2^a$, and put $o^*_{2,1} = a$. If $2^b \not| v_2$ then replace v by $v \cdot 2^b$, v_2 by $v_2 \cdot 2^b$, and put $o^*_{2,2} = a + b$. Perform step (c1) for all $\omega = 2^k \le W^\#$, with $k \ge 2$.

(c1) If $\omega \notin \Omega$ then replace Ω by $\Omega \cup \{\omega\}$. If $2^{a+b+k-1} \not| v_\omega$ then divide $\sum_{i=0}^{\omega-1} r_{\omega,i} \cdot n^i$ by $2^{a+b+k-1}$ using the same method as in (b1), replace $\mathcal{F}_\omega$ by $\mathcal{F}_\omega \cup \{2\}$, w by $\mathrm{lcm}(w, \omega)$, v by $v \cdot 2$, and v_ω by $v_\omega \cdot 2$, and put $o^*_{2,o_2(\omega)} = a + b + k - 1$.

(d) Perform step (d1) for all primes p with $B_2 < p \le B_1$ (where the primes p are generated using the file created in (2.1)), but only as long as $v \le \sqrt{n}$.

In this step it is checked if $n^\omega - 1$ is divisible by p, for some $\omega \le W^\#$. If such an ω can be found, then $\sum_{i=0}^{\omega-1} r_{\omega,i} \cdot n^i$ is actually divided by p to find the multiplicity of p in $n^\omega - 1$.

(d1) Let $n_p = n \bmod p$. Calculate $n_p^\omega \bmod p$ for $\omega = 1, 2, \ldots$ in succession until $n_p^\omega \equiv 1 \bmod p$ or $\omega = W^\#$. If $n_p^\omega \equiv 1 \bmod p$ for some $\omega \le W^\#$, then perform steps (d1a) through (d1d).

If $n_p^\omega \equiv 1 \bmod p$, then p is a divisor of $n^\omega - 1$ and $o_p(n^\omega - 1)$ should be calculated.

(d1a) Put $e = 0$, $m = 0$, and perform (d1a1) until $m \neq 0$.

(d1a1) Put $e = e + 1$ and let $c = 0$. For $i = \omega - 1, \omega - 2, \ldots, 0$ in succession, first put $c' = (c \cdot n + r_{\omega,i}) \bmod p$ and $r_{\omega,i} = \lfloor (c \cdot n + r_{\omega,i})/p \rfloor$, and next replace m by $(m \cdot n_p + r_{\omega,i}) \bmod p$ and c by c'.

In this step $n^\omega - 1$ is divided $o_p(n^\omega - 1)$ times by p. This is done differently from other multi-divisions (such as in (3.4)(b)), since one not only keeps track of a carry c, but also of a variable m, which has the value $(n^\omega - 1)/p^{e+1} \bmod p$. So m is used to check whether $e = o_p(n^\omega - 1)$. After every call to step (d1a) the $r_{\omega,i}$ satisfy $\sum_{i=0}^{\omega-1} r_{\omega,i} \cdot n^i = (n^\omega - 1)/(v_\omega \cdot p^e)$ and m is equal to $n^\omega - 1 \bmod p^{e+1}$.

(d1b) If $\omega \notin \Omega$ then replace Ω by $\Omega \cup \{\omega\}$ and w by $\mathrm{lcm}(w, \omega)$. If $p \not| v_\omega$ then replace v_ω by $v_\omega \cdot p^e$ and v by $v \cdot p^e$, and put $o^*_{p,e} = e$.

(d1c) For all $\omega' = \omega \cdot p^k \leq W^\#$, $k \geq 1$, do the following. If $\omega' \notin \Omega$ then replace Ω by $\Omega \cup \{\omega'\}$ and w by $\mathrm{lcm}(w, \omega')$. If $p \not| v_{\omega'}$ then replace $v_{\omega'}$ by $v_{\omega'} \cdot p$ and put $o^*_{p,e+k} = e + k$.

(d1d) For all $\omega' \leq W^\#$, with $\omega \cdot p^k \mid \omega'$ and $k \geq 1$, divide $\sum_{i=0}^{\omega'-1} r_{\omega',i} \cdot n^i$ by p^{e+k} using the same method as in (b1).

5. LUCAS-LEHMER AND JACOBI SUM TESTS.

In this section the actual Lucas-Lehmer tests and Jacobi sum tests are described.

(5.1) Generation of additional Galois extensions.

Perform steps (a) and (b) for every prime power $u = l^e \in \mathcal{L}^-$.

As explained in step (2.3), one will have to compute in certain extension rings of $\mathbf{Z}/n\mathbf{Z}$, which can be constructed from rings of integers of cyclic subfields of cyclotomic extensions. For all prime power degrees l^e dividing $\lambda(t_0)$, a list of extensions is precalculated in step (2.3). As explained in Section 4, an extension can only be used, if the conductor m of the corresponding field extension satisfies the correct condition (cf. (4.2) and II.(4.6)). It may be that none of the extensions on the list for l^e satisfies this condition.

Secondly, for ω with $l^e \mid \omega$ and $l^e \not| \lambda(t_0)$, it may be decided during the optimization step in Section 4, that using a factor of $n^\omega - 1$ (and therefore using an extension of degree ω), would reduce the running time of the algorithm.

In both cases, the degree l^e was added to the set $\mathcal{L}^-$ during the optimization step. In this step one generates an additional Galois extension for every prime power $l^e \in \mathcal{L}^-$. This step is basically the performance of step (2.3)(b) for all prime powers l^e in $\mathcal{L}^-$. Explanation can be found there. Here we will only comment on the differences.

(a) Find the smallest $m \in \mathcal{M}^+(u)$ such that $m \mid s$ (cf. (4.2)); if none such m exists take the smallest $m \in \mathcal{M}^+(u)$.

(b) If $u = 2$ and $m \in \{4, 8\}$ perform steps (b1), (b2), (b3), and (b10). If $u = 2$ and $u \mid m - 1$, perform steps (b1), (b2), (b4), (b5) and (b10). If $u = 2^e$ with $e > 1$, or $u = l^e$ with l odd and $e \geq 1$ perform steps (b1), and (b5) through (b10).

(b1) Put $r = 1$ if m is prime, and put $r = 0$ if m is not prime.

(b2) Put $D = 1$. If m is prime put

$$S = \begin{pmatrix} -1 & 1 \\ -1 & 0 \end{pmatrix} \text{ and } S^* = \begin{pmatrix} 0 & -1 \\ 1 & -1 \end{pmatrix}.$$

If $m \in \{4, 8\}$ put S and S^* both equal to the 2×2 unit matrix.

(b3) If $m = 4$ put $g = 3$ and $f = X^2 + 1$; if $m = 8$ put $g = 5$ and $f = X^2 - 2$.

(b4) Put $f = X^2 + X + (4 \cdot \lfloor \frac{m+1}{4} \rfloor - m) \cdot \lfloor \frac{m+1}{4} \rfloor$.

(b5) If m is odd, find a primitive root g modulo m, for instance by trying all $g \geq 2$ with $\gcd(m, g) = 1$ in succession. If $m = 2^{e+2}$, put $g = 5$.

(b6) Let $b_{i,j} = 0$ for $0 \leq i < u$ and $0 \leq j < m$. Put $j = 1$. If m is odd, for $i = 0, 1, \ldots, \phi(m) - 1$ in succession replace $b_{i \bmod u, j}$ by 1 and j by $j \cdot g \bmod m$. If $m = 2^{e+2}$, for $i = 0, 1, \ldots, 2^e - 1$ in succession replace $b_{i \bmod u, j}$ and $b_{i \bmod u, m-j}$ by 1 and j by $j \cdot g \bmod m$. Define $\eta = \sum_{j=0}^{m-1} b_{0,j} \cdot \zeta_m^j$ and, for $i = 0, 1, \ldots, u - 1$, its conjugates $\sigma_g^i(\eta) = \sum_{j=0}^{m-1} b_{i,j} \cdot \zeta_m^j$. The $\sigma_g^i(\eta)$ will be represented by the vectors $(b_{i,j})_{j=0}^{m-1}$.

(b7) Determine the polynomial $f = \prod_{i=0}^{u-1}(X - \sigma_g^i(\eta)) \in \mathbf{Z}[X]$ by calculating sufficiently close approximations $c_i \in \mathbf{C}$ to $\sum_{j=1}^{m-1} b_{i,j} \cdot \zeta_m^j$ for $0 \le i < u$, with $\zeta_m = e^{2\pi\sqrt{-1}/m}$,and by rounding the coefficients of the polynomial $\prod_{i=0}^{u-1}(X - c_i)$ to the nearest integers.

(b8) If m is prime, define $\varsigma_{g,k,u}$ for $0 \le k < u$ by

$$\varsigma_{g,k,u} = \sigma_g^k(\eta_u) = \sigma_g^k(\eta).$$

If m is not prime, define $\varsigma_{g,k,u}$ for $0 \le k < u$ by $\varsigma_{g,0,u} = 1$, and

$$\varsigma_{g,k,u} = \sigma_g^{(k-l^{i-1})}(\eta_{l^i}),$$

with i such that $1 \le i \le e$ and $l^{i-1} \le k < l^i$. Compute a $u \times u$ dimensional integer matrix S by performing steps (b8a) through (b8c).

(b8a) Determine $d_{i,j}$ such that $\eta^i = \sum_{j=0}^{m-1} d_{i,j} \cdot \zeta_m^j$ for $i = 2, 3, \ldots, u-1$, by computing the consecutive powers of the polynomial $\sum_{j=0}^{m-1} b_{i,j} \cdot T^j$ modulo the polynomial $T^m - 1$.

(b8b) Perform step (b8b1) if m is prime and perform step (b8b2) for $k = 1, \ldots, e$ in succession if m is not a prime.

(b8b1) First put $h = 1$ and for $j = 0, 1, \ldots, u-1$ in succession first put $s_{j,i} = d_{i,h} - d_{i,0}$ for $0 \le i < u$ and next replace h by $h \cdot g \bmod m$.

(b8b2) Put $h = 1$ and for $j = l^{k-1}, \ldots, l^k - 1$ in succession first put $h' = (hm/l^k) \bmod m$, next put $s_{j,i} = d_{i,h'}$ for $0 \le i < u$ and finally replace h by $h \cdot g \bmod m$. For $j = l^k, \ldots, l^k + l^{k-1} - 1$ in succession first put $h' = (hm/l^k) \bmod m$, next put $s_{(j-zl^{k-1}),i} = s_{(j-zl^{k-1}),i} - d_{i,h'}$ for $z = 1, \ldots, l-1$ and $0 \le i < u$ and finally replace h by $h \cdot g \bmod m$.

(b8c) Put $s_{0,i} = d_{i,0}$ for $0 \le i < u$. Let S be the matrix having the $(s_{j,i})_{j=0}^{u-1}$ as columns, for $0 \le i < u$.

(b9) Compute the $u \times u$ dimensional integer matrix S^* and $D \in \mathbf{Z}_{>0}$ such that $D^{-1} \cdot S^* = S^{-1}$ and D is minimal. This can efficiently be done using a variant of the Gaussian elimination method. See V.6 for more details.

Since n is known, one can perform these operations in $\mathbf{Z}/n\mathbf{Z}$, thereby reducing the length of the integers involved.

(b10) Tabulate u, $m_u = m$, $r_{u,m} = r$, $g_{u,m} = g$, $f_{u,m} = f$, $S_{u,m} = S$, $S^*_{u,m} = S^*$, and $D_{u,m} = D$.

(5.2) Selection of cyclic rings and creation of transition matrices.

Let t and v be as determined in step (4.1), and let $\mathrm{lcm}\{u_{p,\hat{k}} : p \text{ prime}, p^{\hat{k}} \parallel \mathrm{lcm}(t, v)\} = \prod_{l \text{ prime}} l^{\hat{e}_l}$. Perform steps (a) and (b) for all prime powers $\hat{u} = l^{\hat{e}_l}$.

In steps (1.3) and (5.1) cyclic extensions $\mathbf{Z}[\eta_{\hat{u}}]/n\mathbf{Z}[\eta_{\hat{u}}]$ of degree $\hat{u}$ over $\mathbf{Z}/n\mathbf{Z}$ were found; here $\eta = \eta_{\hat{u}}$ is a zero of $f_{\hat{u},m}$ and equals $\sum \zeta_m^j$ where j ranges over the powers of $g^{\hat{u}}$ modulo m, where g is a primitive root modulo m. The matrix $S_{\hat{u},m}$ converts an element expressed on the basis $(\eta^0, \eta^1, \ldots, \eta^{\hat{u}-1})$ to its representation on the basis $(\varsigma_{g,0,\hat{u}}, \varsigma_{g,1,\hat{u}}, \ldots, \varsigma_{g,\hat{u}-1,\hat{u}})$. If m_l is prime then $\varsigma_{g,k,\hat{u}}$ is defined by

$$\varsigma_{g,k,\hat{u}} = \sigma_g^k(\eta_{\hat{u}}) = \sigma_g^k(\eta),$$

for $0 \le k < \hat{u}$. If m_l is not prime then $\varsigma_{g,k\hat{u}}$ is defined by $\varsigma_{g,0,\hat{u}} = 1$ and

$$\varsigma_{g,k,\hat{u}} = \sigma_g^{(k-l^{i-1})}(\eta_{l^i})$$

for $l^{i-1} \le k < l^i$ and $i = 1, \ldots, \hat{e}_l$. Here the map σ_g acts via $\sigma_g(\zeta_m) = \zeta_m^g$. Instead of the automorphism σ_g, one needs σ_n in steps (5.3) and (5.4), where σ_n acts via $\sigma_n(\zeta_m) = \zeta_m^n$. Therefore, if $n \not\equiv g \bmod \hat{u}$, the matrix $S_{\hat{u},m}$ and its inverse are transformed in such a way that they perform the transformations with respect to the basis $(\varsigma_{n,0,\hat{u}}, \varsigma_{n,1,\hat{u}}, \ldots, \varsigma_{n,\hat{u}-1,\hat{u}})$. Finally a matrix $S^{\#}_{\hat{u},m}$ is calculated, such that for any element $x \in \mathbf{Z}[\eta]/n\mathbf{Z}[\eta]$ which is represented as a $\hat{u}$-dimensional column vector over $\mathbf{Z}/n\mathbf{Z}$, the element $\sigma_n(x)$ equals $S^{\#}_{\hat{u},m} \cdot x$.

(a) Retrieve the smallest $m_{\hat{u}} \in \mathcal{M}^+(\hat{u})$ with $m_{\hat{u}} \mid s$ from the list made in (4.2). If no such $m_{\hat{u}}$ exists, take the smallest $m_{\hat{u}}$ from the list. Put $m_l = m_{\hat{u}}$.

The set $\mathcal{M}^+(\hat{u})$ as constructed in step (4.2) contains those conductors m that satisfy the correct condition (cf. (4.2) and II.(4.8)). In step (4.1) and step (5.1) the sets $\mathcal{M}^+(\hat{u})$ are constructed in such a way that they will not be empty. By choosing m one fixes the extension of degree l^e which will be used in steps (5.3) and (5.4).

(b) Put $m = m_l$ and replace G by $(G \cdot m) \bmod n$. If $G = 0$ then n is composite and the primality test terminates. For $1 \le e \le \hat{e}_l$ put $u = l^e$ and perform steps (b1) through (b4).

If a cyclic extensions of degree l^e, $1 \le e < \hat{e}_l$ is needed, the l^e-th degree subextension of the extension of degree $l^{\hat{e}_l}$ is used for this; here $l^{\hat{e}_l}$ is the largest l-th power degree that will be used. This means in particular that the same conductor m is taken for all extensions of degree l^e with $1 \le e \le \hat{e}_l$. By construction the extension of degree u_1 is now contained in the extension of degree u_2 whenever $u_1 \mid u_2$. Furthermore it has to be checked that n and m are relatively prime.

(b1) Retrieve $r_{u,m}$, $g_{u,m}$, $f_{u,m}$, $S_{u,m}$, $S^*_{u,m}$, and $D_{u,m}$ from the table made in (1.3) and (5.1).

(b2) Replace $S^*_{u,m}$ by $D^{-1} \cdot S^*_{u,m} \bmod n$; if $D^{-1} \bmod n$ does not exist, then n is composite and the primality test is terminated.

After reducing the matrix S^* modulo n and multiplying it by $D^{-1} \bmod n$, the matrix S^* converts an element expressed in terms of $(\varsigma_{g,0,u}, \varsigma_{g,1,u} \ldots, \varsigma_{g,u-1,u})$, to its representation in terms of η^j with $0 \le j < u$.

(b3) Find $i^* \in \{1, \ldots, m-2\}$ such that $g_{u,m}^{i^*} \equiv n \bmod m$ by trying $i^* = 1, 2, \ldots$ in succession. Perform steps (b3a) and (b3b) if $i^* \neq 1$.

If $i^* \neq 1$, the matrices $S_{u,m}$ and $S_{u,m}^*$ as calculated in steps (1.3) or (5.1) are different from the matrices needed in the algorithm. In this case the rows of $S_{u,m}$ and the columns of $S_{u,m}^*$ have to be changed. This is much faster then recalculating them by using for instance step (5.1).

(b3a) Introduce a matrix $S'_{u,m} = (s'_{i,j})_{i,j=0}^{u-1} \in \mathbf{Z}^{u \times u}$ and initially put $s'_{i,j} = 0$ for $0 \leq i < u$ and $0 \leq j < u$. Perform step (b3a1) if m_l is prime and for $k = 1, \ldots, e$ perform step (b3a2) if m_l is not a prime. Finally replace $S_{u,m}$ by $S'_{u,m}$.

(b3a1) Put the matrix $S'_{u,m}$ equal to the whose j-th row equals the $(i^* \cdot j \bmod u)$-th row of $S_{u,m}$, for $0 \leq j < u$.

The matrix $S_{u,m}$, as constructed in step (1.3) or (5.1) converts an element expressed in terms of η^j, $0 \leq j < u$, to its representation in terms of $(\varsigma_{g,0,u}, \varsigma_{g,1,u}, \ldots, \varsigma_{g,u-1,u}) = (\sigma_g^0(\eta), \sigma_g^1(\eta), \ldots, \sigma_g^{u-1}(\eta))$, where the map σ_g acts via $\sigma_g(\zeta_m) = \zeta_m^g$. The map σ_n, which will be used in the sequel, acts via $\sigma_n(\zeta_m) = \zeta_m^n = \sigma_g^{i^*}(\zeta_m)$. This implies that if the matrix $S_{u,m}$ is to convert an element expressed in terms of η^j, $0 \leq j < u$, to its representation in terms of $(\varsigma_{n,0,u}, \varsigma_{n,1,u}, \ldots, \varsigma_{n,u-1,u}) = (\sigma_n^0(\eta), \sigma_n^1(\eta), \ldots, \sigma_n^{u-1}(\eta))$, which is in fact $(\sigma_g^{0 \cdot i^*}(\eta), \sigma_g^{1 \cdot i^*}(\eta), \ldots, \sigma_g^{(u-1) \cdot i^*}(\eta))$, one has to replace the matrix $S_g = S_{u,m}$ by a matrix S_n. This is done by permuting the 0-th through the $(u-1)$-nd row of S_g.

(b3a2) For $j = l^{k-1}, \ldots, l^k - 1$ first put $j' = (i^* \cdot (j - l^{k-1}) \bmod l^k) + l^{k-1}$ and next if $j' < l^k$ put the j-th row of $S'_{u,m}$ equal to the j'-th row of $S_{u,m}$; if $j' \geq l^k$ then put the j-th row of $S'_{u,m}$ equal to -1 times the sum of the $(j' - z \cdot l^{k-1})$-th rows of $S_{u,m}$ with $z = 1, \ldots, l-1$.

The matrix $S_{u,m}$ converts an element expressed in terms of η^j, $0 \leq j < u$, to its representation in terms of $(\varsigma_{g,0,u}, \varsigma_{g,1,u}, \ldots, \varsigma_{g,u-1,u})$. By changing from g to $n \bmod m_l$ the elements $\varsigma_{g,l^{i-1},u}, \ldots, \varsigma_{g,l^i-1,u}$ are permuted such that $\varsigma_{n,j,u} = \varsigma_{g,j',u}$, where j' is defined above, for $j = l^{k-1}, \ldots, l^k - 1$ and $k = 1, \ldots, e$. During the permutation elements $\varsigma_{g,j',u}$ with $j' \geq l^k$ may be introduced in the basis. These can be represented by $-\sum_{z=1}^{l-1} \varsigma_{g,j'-z \cdot l^{k-1},u}$.

(b3b) Similarly as in step (b3a), the matrix $S_{u,m}^*$ will be replaced by a permuted version. Introduce a matrix $S'_{u,m} = (s'_{i,j})_{i,j=0}^{u-1} \in \mathbf{Z}^{u \times u}$ and initially put $s'_{i,j} = 0$ for $0 \leq i < u$ and $0 \leq j < u$. Perform step (b3b1) if m_l is prime and for $k = 1, \ldots, e$ perform step (b3b2) if m_l is not a prime. Finally replace $S_{u,m}^*$ by $S'_{u,m}$.

The matrix $S_{u,m}^*$, as constructed in step (5.2)(b2) converts an element expressed in terms of $(\varsigma_{g,0,u}, \varsigma_{g,1,u}, \ldots, \varsigma_{g,u-1,u})$, to its representation in terms of η^j, $0 \leq j < u$, where the map σ_g (used in the definition of $\varsigma_{g,k,u}$) acts via $\sigma_g(\zeta_m) = \zeta_m^g$. Since the matrix $S_{u,m}^*$ is to convert an element expressed in terms of $(\varsigma_{n,0,u}, \varsigma_{n,1,u}, \ldots, \varsigma_{n,u-1,u})$, to its representation in terms of η^j, $0 \leq j < u$, one has to replace the matrix $S_g^* = S_{u,m}^*$ by a matrix S_n^*. This is done in the same way as in step (b3a) by replacing the row operations by the corresponding column operations.

(b3b1) Put the matrix $S'_{u,m}$ equal to the whose j-th row equals the $(i^* \cdot j \bmod u)$-th row of $S_{u,m}$, for $0 \leq j < u$.

(b3b2) For $j = l^{k-1}, \ldots, l^k - 1$ first put $j' = (i^* \cdot (j - l^{k-1}) \bmod l^k) + l^{k-1}$ and next if $j' < l^k$ put the j-th column of $S'_{u,m}$ equal to the j'-th column of $S_{u,m}$; if $j' \geq l^k$ then put the j-th column of $S'_{u,m}$ equal to -1 times the sum of the $(j' - z \cdot l^{k-1})$-th columns of $S_{u,m}$ with $z = 1, \ldots, l-1$.

(b4) Let $S_{u,m} = (s_{i,j})_{i,j=0}^{u-1}$ and $S^*_{u,m} = (s^*_{i,j})_{i,j=0}^{u-1}$. Perform step (b4a) if m_l is prime and for $k = 1, \ldots, e$ perform step (b4b) if m_l is not prime. Finally put $S^{\#}_{u,m} = (s^{\#}_{i,j})_{i,j=0}^{u-1}$.

(b4a) Put $s^{\#}_{i,j} = \sum_{k=0}^{u-1} s^*_{i,k} \cdot s_{(k-1) \bmod u, j}$ for $0 \leq i, j < u$.

For any element $x \in \mathbf{Z}[\eta_u]/n\mathbf{Z}[\eta_u]$, which is represented as a u-dimensional column vector over $\mathbf{Z}/n\mathbf{Z}$, the element $\sigma_n(x)$ will equal $S^{\#}_{u,m} \cdot x$. In order to apply σ_n to an element in $\mathbf{Z}[\eta_u]/n\mathbf{Z}[\eta_u]$ in an easy way, the element will be transformed to its representation with respect to $(\varsigma_{n,0,u}, \varsigma_{n,1,u}, \ldots, \varsigma_{n,u-1,u})$; in this representation the mapping σ_n is simply a shift of the coordinates. By transforming the result back to the representation with respect to η^j, $0 \leq j < u$, one gets the final result. Instead of applying $S_{u,m}$, performing a shift and applying $S^*_{u,m}$, one computes the complete transformation in advance, by multiplying $S_{u,m}$ by a 'shifted' version of $S^*_{u,m}$ to get $S^{\#}_{u,m}$. This is done as follows. By multiplying the element $x = \sum_{j=0}^{u-1} x_j \eta^j$ by S, one gets its representation $(y_0, y_1, \ldots, y_{u-1})$ with respect to the basis $(\varsigma_{n,0,u}, \varsigma_{n,1,u}, \ldots, \varsigma_{n,u-1,u}) = (\sigma_n^0(\eta), \sigma_n^1(\eta), \ldots, \sigma_n^{u-1}(\eta))$. So $y_i = \sum_{j=0}^{u-1} s_{i,j} \cdot x_j$. Next σ_n is applied, giving a representation with respect to the basis $(\sigma_n^1(\eta), \sigma_n^2(\eta), \ldots, \sigma_n^{u-1}(\eta), \sigma_n^0(\eta)) = (\varsigma_{n,1,u}, \varsigma_{n,2,u}, \ldots, \varsigma_{n,u-1,u}, \varsigma_{n,0,u})$. So $\sigma_n(x) = y_{u-1} \cdot \varsigma_{n,0,u} + \sum_{j=0}^{u-2} y_j \cdot \varsigma_{n,j+1,u}$. Expressing this in terms of $(\varsigma_{n,0,u}, \varsigma_{n,1,u}, \ldots, \varsigma_{n,u-1,u})$, one gets $\sigma_n(x) = y_{u-1} \cdot \varsigma_{n,0,u} + \sum_{j=2}^{u-1} y_{j-1} \cdot \varsigma_{n,j,u}$. To represent this element on the basis η^j with $0 \leq j < u$, one has to multiply it by S^*, giving $(\sigma_n(x))_i = \sum_{j=0}^{u-1} s^*_{i,j} \cdot y_{(j-1) \bmod u}$. Substituting $y_i = \sum_{j=0}^{u-1} s_{i,j} \cdot x_j$, gives the result above.

(b4b) Put $s^{\#}_{i,j} = \sum_{k=1}^{e} \sum_{z=l^{k-1}+1}^{l^k-1} s^*_{i,z} \cdot s_{(z-1),j} - \sum_{k=1}^{e} \sum_{z=0}^{l-1} s^*_{i,l^{k-1}} \cdot s_{(l^k - z l^{k-1} - 1),j}$ for $0 \leq i, j < u$.

In this step essentially the same operations are performed as in step (b4a). The only difficult part is the representation of basis elements that are introduced in the basis. Basically, the rules described in step (b3b) are used to solve these problems.

(5.3) Calculation of cyclotomic extensions.

Let t and v be as determined in step (4.1) and introduce for all primes $p \mid \mathrm{lcm}(t, v)$ a boolean variable f_p and put f_p initially equal to *false*. Furthermore, let $\hat{k}_p$ be such that $p^{\hat{k}_p} \parallel \mathrm{lcm}(t, v)$ for all primes $p \mid \mathrm{lcm}(t, v)$. Perform steps (a) through (e) as long as there exists at least one $p \mid \mathrm{lcm}(t, v)$ with f_p equal to *false*.

In this step one generates for each prime-divisor p of $\mathrm{lcm}(t, v)$ a $p^{k(p)}$-th root of unity in an extension of degree u, where $k(p) = \max(1, o_p(t))$ and $u = \mathrm{ord}(n \bmod p^{\hat{k}_p})$. This is done by trying the $(n^u - 1)/p^{k(p)}$-th power of random elements. Every random choice has a probability $\frac{p-1}{p}$ of success if n is prime. Instead of generating a root of unity for each prime $p \mid \mathrm{lcm}(t, v)$ separately, one can try to generate roots of unity for a number of primes simultaneously. Let $\hat{u}$ be such that $\hat{u} = \max\{u_{p,\hat{k}_p} : p \mid \mathrm{lcm}(t, v)\}$ and let $\hat{p}$ be such that $u_{\hat{p},\hat{k}_{\hat{p}}} = \mathrm{ord}(n \bmod \hat{p}^{\hat{k}_{\hat{p}}}) = \hat{u}$. While generating a $\hat{p}^{k(\hat{p})}$-th root of unity, it is conceivable that one generates $p^{k(p)}$-th roots of unity with $u_{p,\hat{k}_p} \mid \hat{u}$ at the same time. First take the product r of all prime powers $p^{k(p)} \mid \mathrm{lcm}(t,v)$ for which $u_{p,\hat{k}_p} \mid \hat{u}$. Next take a random element to the power $(n^{\hat{u}}-1)/r$, and for each p raise the result to the power $r/p^{k(p)}$. This will be a $p^{k(p)}$-th root of unity if its $p^{k(p)-1}$-th power is not equal to 1. For each such p one has a probability $\frac{p-1}{p}$ of success to find a $p^{k(p)}$-th root of unity independent of finding any other root of unity. In fact, as explained in II.(4.15), finding a p-th root of unity in an extension of degree $\mathrm{ord}(n \bmod p^{\hat{k}_p})$ in this way is sufficient for the proof that a $p^{\hat{k}_p}$-th root of unity exists in this extension; the reason for actually constructing the $p^{k(p)}$-th root of unity for primes p dividing t is that these are needed in step (5.4). For primes p dividing v (and not t), one only constructs a p-th root unity in the extension of degree $u_{p,\hat{k}_p}$ to prove the existence of a $p^{\hat{k}_p}$-th root of unity. The variable f_p indicates whether a $p^{k(p)}$-th root of unity has been found.

(a) Put $\hat{u} = \max\{u_{p,\hat{k}_p} : p \mid \mathrm{lcm}(t,v), f_p = \mathit{false}\}$. Find $\hat{p} \mid \mathrm{lcm}(t,v)$ with $f_{\hat{p}} = \mathit{false}$ and $u_{\hat{p},\hat{k}_{\hat{p}}} = \hat{u}$. Put $r = \hat{p}$ if $\hat{p} \nmid t$ and put $r = \hat{p}^{o_{\hat{p}}(t)}$ if $\hat{p} \mid t$.

For primes p not dividing t, only a p-th root of unity is constructed, which is by construction sufficient for the proof of the existence of a $p^{\hat{k}_p}$-th root of unity.

(b) For all $p \neq \hat{p}$ with $p \mid \mathrm{lcm}(t,v)$, $f_p = \mathit{false}$, and $u_{p,\hat{k}_p} \mid \hat{u}$, replace r by $r \cdot p$ if $p \nmid t$, and replace r by $r \cdot p^{o_p(t)}$ if $p \mid t$.

The variable r will be equal to the product of all primes p with $\mathrm{ord}(n \bmod p^{\hat{k}_p}) \mid \hat{u}$. Finding $p^{k(p)}$-th roots of unity, with $k(p) = \max(1, o_p(t))$, will be attempted simultaneously for all these primes.

(c) Put $a = \lfloor (n-1)/r \rfloor$ and let $c = 0$. For $i = \hat{u}-1, \hat{u}-2, \ldots, 0$ in succession perform step (c1).

In this step $n^{\hat{u}}-1$ is divided by r. This was done by writing $n^{\hat{u}}-1$ in base n, i.e., $n^{\hat{u}}-1 = \sum_{i=0}^{\hat{u}-1} b_i \cdot n^i$ and next sequentially dividing all coefficient b_i by r and keeping track of a carry c. The b_i now satisfy $\sum_{i=0}^{\hat{u}-1} b_i \cdot n^i = (n^{\hat{u}}-1)/r$. In order to raise a random element to the power $(n^{\hat{u}}-1)/r$, the base-n representation of this exponent is used, since one can replace the single large exponentiation by a few smaller exponentiations with exponents b_i and a few applications of σ_n (which can be done by applying the matrices $S^{\#}$, computed in step (5.2)).

(c1) First put $b_i = \lfloor (c \cdot n + n - 1)/r \rfloor$, and next replace c by $(c \cdot n + n - 1) \bmod r$. Finally put $b_{1,i} = \lfloor b_i/r \rfloor$ and $b_{2,i} = b_i \bmod r$.

In this step each element b_i of the base-n representation of $n^{\hat{u}}-1$ is written as $b_{1,i} \cdot a + b_{2,1}$, where $a = \lfloor (n-1)/r \rfloor$. In this way one can replace each exponentiation with exponent b_i by two exponentiations with relatively small exponents $b_{1,i}$ and $b_{2,i}$ and by one common exponentiation with exponent a. For more information on this, see V.(4.3).

(d) For all prime powers $\bar{u} = l^{\bar{e}} \parallel \hat{u}$ let $\eta_{\bar{u}}$ be a zero of $f_{\bar{u},m_l}$, with m_l as in (5.2)(a) and $f_{\bar{u},m_l}$ retrieved from the tables as in (5.2)(b1). Perform steps (d1) through (d4).

In this step a random element α will be taken to the power $(n^{\hat{u}}-1)/r$. Furthermore it will be checked that $\alpha^n = \sigma_n(\alpha)$ which should be the case if n is prime (see II.(2.3)).

(d1) Let α be a non-zero random multivariate polynomial in all the $\eta_{\bar{u}}$'s over $\mathbf{Z}/n\mathbf{Z}$.

Although any non-zero choice is allowed in this step, one can take $\alpha = \sum_{\bar{u}} \eta_{\bar{u}}$ as a possible first choice.

(d2) Put $\beta = \alpha$ and perform step (d2a) for all prime powers $\bar{u} = l^{\bar{e}} \parallel \hat{u}$.

If n is prime, the mapping σ_n is equal to n-th powering (cf. II.(2.3)). In this step one calculates $\sigma_n(\alpha)$, where α is the random element chosen in step (d1).

(d2a) Write β as a $\bar{u}$-dimensional column vector, i.e., write β as a polynomial in $\eta_{\bar{u}}$. Replace β by $S^{\#}_{\bar{u},m_l} \cdot \beta$ (cf. (5.2)(b4)).

For any element $x \in \mathbf{Z}[\eta]/n\mathbf{Z}[\eta]$, which is written as a $\bar{u}$-dimensional column vector over $\mathbf{Z}/n\mathbf{Z}$, the element $\sigma_n(x)$ equals $S^{\#}_{\bar{u},m_l} \cdot x$. In this step one calculates $\sigma_n(\alpha)$, where α is the random element chosen in step (d1).

(d3) Check that $\alpha^n = \beta$. If equality does not hold, then n is composite and the primality test is terminated.

If σ_n does not give the same result as n-th powering, the number n cannot be prime.

(d4) Put $\beta = \alpha^a$, $i = \hat{u} - 1$ and $\gamma = \alpha^{b_i}$. For $i = \hat{u} - 2, \hat{u} - 3, \ldots, 0$ in succession replace γ by $\sigma_n(\gamma) \cdot \beta^{b_{1,i}} \alpha^{b_{2,i}}$, where $\sigma_n(\gamma)$ is computed using the matrices $S^{\#}$.

In this step one calculates $\gamma = \alpha^{(n^{\hat{u}}-1)/r} = \prod_{i=0}^{\hat{u}-1} \alpha^{n^i \cdot b_i} = \prod_{i=0}^{\hat{u}-1} \alpha^{n^i \cdot (b_{1,i} \cdot a + b_{2,i})}$ by means of a Horner-scheme. Instead of raising an element to the power n, one uses the image under σ_n which is now known to give the same result for powers of α, at least.

It is possible to combine the exponentiations in this step with the exponentiation in step (d3), to speed up this step of the algorithm. Basically the exponentiations are done in the way as described in [29, Remark (3.6)].

(e) For all primes $p \mid r$ perform steps (e1) and (e2).

In this step, the individual $p^{k(p)}$-th roots are extracted from the result of the huge exponentiation in step (d), where $k(p) = \max(1, o_p(t))$. In fact the result γ of step (d) is taken to the power r/p. If the result of the exponentiation is unequal to 1, then this attempt to find a $p^{k(p)}$-th root of unity has been successful. Otherwise another attempt to find a root of unity should be made.

(e1) If $p \nmid t$ perform step (e1a) and if $p \mid t$ perform steps (e1b) and (e1c).

(e1a) Put $\gamma_{p,1} = \gamma^{r/p}$.

(e1b) First put $k(p) = \max(1, o_p(t))$, put $\gamma^* = \gamma^{r/p^{k(p)}}$. Next put $\gamma_{p,k} = \gamma^*$. For all prime powers $\bar{u} = l^{\bar{e}} \parallel \hat{u}$ perform step (e1b1) if $\bar{u}$ does not divide $u_{p,k(p)}$.

The element γ^* is constructed as an element in an extension of degree $\hat{u}$ over $\mathbf{Z}/n\mathbf{Z}$. It can however also be represented as an element in an extension of degree $u_{p,k(p)}$ over $\mathbf{Z}/n\mathbf{Z}$. For each prime $l \mid \hat{u}$ with $o_l(u_{p,k(p)}) < o_l(\hat{u})$, let $u = l^e \parallel u_{p,k(p)}$. The element γ^* is written as a $\bar{u}$-dimensional column vector, i.e., as a polynomial in $\eta_{\bar{u}}$. By applying $S_{\bar{u},m_l}$ to this vector, one writes γ^* as a combination of $(\varsigma_{n,0,\bar{u}}), \varsigma_{n,1,\bar{u}}), \ldots, \varsigma_{n,u-1,\bar{u}}))$. Since γ^* is known to live in a subextension, this representation has a repetitive character if m_l (cf. (2.3)) is prime. If m_l is not prime, then only the first u coordinates of this representation are non-zero. By taking only the first u coefficients and applying S^*_{u,m_l} to these, one gets the representation of γ^* as a polynomial in η_u, both in the case that m_l is prime and in the case that m_l is not prime.

(e1b1) Let e be such that $l^e \parallel u_{p,k(p)}$, and let $u = l^e$. Represent γ^* as a $\bar{u}$-dimensional column vector, i.e., write γ^* as a polynomial in $\eta_{\bar{u}}$. Apply S^*_{u,m_l} to the vector consisting of the first u coordinates of the vector $S_{\bar{u},m_l} \cdot \gamma^*$, and replace γ^* by the result (cf. (5.2)(b3a) and (5.2)(b3b)). Put $\gamma_{p,k(p)} = \gamma^*$.

(e1c) For $i = k(p) - 1, k(p) - 2, \ldots, 1$ in succession first compute $\gamma_{p,i} = \gamma_{p,i+1}^p$ and next perform step (e1c1) for those prime powers $\bar{u} = l^{\bar{e}} \parallel u_{p,i+1}$ such that $\bar{u}$ does not divide $u_{p,i}$.

The elements $\gamma_{p,j}$, for $j = 1, 2, \ldots, k(p)$, will be the p^j-th roots of unity if n is prime, which will be used in step (5.4).

(e1c1) Let e be such that $l^e \parallel u_{p,i}$, and let $u = l^e$. Represent γ^* as a $\bar{u}$-dimensional column vector, i.e., write γ^* as a polynomial in $\eta_{\bar{u}}$. Apply S^*_{u,m_l} to the vector consisting of the first u coordinates of the vector $S_{\bar{u},m_l} \cdot \gamma^*$, and replace γ^* by the result (cf. (5.2)(b3a) and (5.2)(b3b)). Put $\gamma_{p,i} = \gamma^*$.

This step will only be performed for those prime powers $\bar{u}$ which are not divisors of $u_{p,i}$; in particular if $u_{p,i+1} = u_{p,i}$ this step is skipped. This is in fact the same transformation as performed in step (e1b). Explanation can be found there.

(e2) Put $\delta = \gamma_{p,1} - 1$. If $\delta \neq 0$, select any non-zero coefficients a of α and d of δ, and replace G by $(G \cdot a \cdot d) \bmod n$. If $G = 0$ a factor of n can easily be derived and the primality test is terminated. Otherwise, if $G \neq 0$, put f_p =*true*.

If δ is not equal to zero, the $\gamma_{p,k}$ are really p^k-th roots of unity.

(5.4) Jacobi sum tests.

Let $\mathbf{S}_{\mathcal{T}}$ be as constructed in step (4.1). Perform steps (a) through (d) for all $\mathcal{S} \in \mathbf{S}_{\mathcal{T}}$.

Each set $\mathcal{S} \in \mathbf{S}_{\mathcal{T}}$ contains a set of triples (q, p, h) for which a combined Jacobi sum test will be performed. Each triple (q, p, h) represents a Jacobi sum test for a character χ of order p^k, where $k = o_p(q-1)$, and conductor q in an extension of degree $u_{p,1} \cdot p^h$. The sets $\mathcal{S} \in \mathbf{S}_{\mathcal{T}}$ have been determined in step (4.1).

In this step one proves that

$$\prod_{(q,p,h) \in \mathcal{S}} \tau(\chi_{p,q})^n / \tau(\chi_{p,q}^n)$$

is a power of an r-th root of unity, where r is defined by

$$r = \prod_{(q,p,h) \in \mathcal{S}} p^{o_p(q-1)}$$

and $\chi_{p,q}$ is a character of order p^k and of conductor q. This is done by expressing $\tau(\chi_{p,q})^{n-\sigma_n}$ in terms of Jacobi sums, for all $(q, p, h) \in \mathcal{S}$ and showing that the product of all $\tau(\chi_{p,q})^{n-\sigma_n}$ is equal to a power of an r-th root of unity. The Jacobi sums, and their exponents have been calculated in advance in step (1.4).

(a) Let $u^* = \operatorname{lcm}\{u_{p,1} \cdot p^h : (q,p,h) \in \mathcal{S}\} = \prod_{l \text{ prime}} l^{\tilde{e}_l}$. For all prime powers $u = l^e$, with $1 \le e \le \tilde{e}_l$, let η_u denote a zero of f_{u,m_l}; here m_l is as in step (5.2)(a), and f_{u,m_l} is as in (5.2)(b1).

By construction, u^* will be one of the $u_{p,1} \cdot p^h$ for some $(q,p,h) \in \mathcal{S}$.

(b) Let $r = \prod_{(q,p,h)\in\mathcal{S}} p^{o_p(q-1)}$. Let $c = \lfloor n/r \rfloor$ and $n_r = n - r \cdot c$. Set α, β, and γ equal to 1; these should be regarded as multivariate polynomials in all the $\eta_{\tilde{u}}$'s over $\mathbf{Z}/n\mathbf{Z}$, for $\tilde{u} = l^{\tilde{e}_l} \parallel u^*$.

For each element $(q,p,h) \in \mathcal{S}$ a Jacobi test for a character χ of order p^k and conductor q in an extension of degree $u_{p,1} \cdot p^h$ should be performed. These tests involve a large exponentiation in an extension of degree $u_{p,1} \cdot p^h$. Since for all elements $(p,q,h) \in \mathcal{S}$ about the same exponentiation has to be carried out, the common part of the exponentiations is done simultaneously, to save time. The exponent of this common part is equal to c.

(c) Perform steps (c1) through (c5) for all triples (q,p,k) such that $(q,p,h) \in \mathcal{S}$ and $k = o_p(q-1)$.

In this step the elements needed for the final huge exponentiation are calculated. These elements are

$$\alpha = \prod_{p|r} \tau(\chi_{p,q})^{n_r - \sigma_{n_p}},$$

and

$$\beta = \prod_{p|r} \tau(\chi_{p,q})^{r},$$

respectively, where $n_p \equiv n_r \bmod p^k$ and $0 \le n_p < p^k$.

(c1) Let $n_p \equiv n_r \bmod p^k$ and $0 \le n_p < p^k$. For every $J \in \mathcal{J}_{p^k}$ retrieve e_{π,p^k,n_p} and e_{π,p^k,p^k}, which were tabulated in step (2.4)(f).

For those $J \in \mathcal{J}_{p^k}$ for which $e_{\pi,p^k,n_p} \neq 0$ or $e_{\pi,p^k,p^k} \neq 0$ retrieve $J \in \mathbf{Z}[\zeta_{p^k}]$ from the direct access file created in (2.4) and transform these J to $\mathbf{Z}[\zeta_{p^k}]/n\mathbf{Z}[\zeta_{p^k}]$ by taking their coefficients modulo n.

Compute

$$J^* = \prod_{\substack{J \in \mathcal{J}_{p^k} \\ e_{\pi,p^k,n_p} \neq 0}} J^{e_{\pi,p^k,n_p}} \in \mathbf{Z}[\zeta_{p^k}]/n\mathbf{Z}[\zeta_{p^k}]$$

and

$$J^{\#} = \chi(-1) \cdot q \cdot \prod_{\substack{J \in \mathcal{J}_{p^k} \\ e_{\pi,p^k,p^k} \neq 0}} J^{e_{\pi,p^k,p^k}} \in \mathbf{Z}[\zeta_{p^k}]/n\mathbf{Z}[\zeta_{p^k}],$$

where $\chi(-1) = -1$ if $q \equiv 3 \bmod 4$ and p even, and $\chi(-1) = 1$ if $q \equiv 1 \bmod 4$ or p odd.

In this step the elements $J^* = \tau(\chi)^{n_p - \sigma_{n_p}}$ and $J^\# = \tau(\chi)^{p^k}$ are calculated as elements of $\mathbf{Z}[\zeta_{p^k}]/n\mathbf{Z}[\zeta_{p^k}]$.

(c2) Let $J^* = (a_i^*)_{0 \le i < \phi(p^k)}$ and $J^\# = (a_i^\#)_{0 \le i < \phi(p^k)}$. Put $\gamma^* = \gamma_{p,k}$, where $\gamma_{p,k}$ is as in (5.3)(e1).

The $\gamma_{p,k}$ are p^k-th roots of unity, calculated in step (5.3)(e1).

Put $i = \phi(p^k) - 1$, and put $J_\alpha = a_i^*$ and $J_\beta = a_i^\#$. For $i = \phi(p^k) - 2, \phi(p^k) - 3, \ldots, 0$ in succession, replace J_α by $J_\alpha \cdot \gamma^* + a_i^*$ and J_β by $J_\beta \cdot \gamma^* + a_i^\#$.

The elements $J^\#$ and J^* have to be transformed to elements in an extension of degree $u_{p,k}$ of $\mathbf{Z}/n\mathbf{Z}$. This is done by performing a Horner-scheme, substituting γ^* for ζ_{p^k}. The resulting expressions, J_α and J_β are elements in an extension of degree $u_{p,k}$ of $\mathbf{Z}/n\mathbf{Z}$, where J_α equals $\tau(\chi)^{n_p - \sigma_{n_p}}$ and J_β equals $\tau(\chi)^{p^k}$.

(c3) Put $\alpha^* = J_\alpha \cdot (J_\beta)^{\lfloor n_r / p^k \rfloor}$, and $\beta^* = (J_\beta)^{r/p^k}$.

The element α^* equals $\tau(\chi)^{n_p - \sigma_{n_p} + n_r - n_p} = \tau(\chi)^{n_r - \sigma_{n_p}}$ and β^* equals $\tau(\chi)^r$.

(c4) For all $\tilde{u} = l^{\tilde{e}_l} \parallel u^*$ perform steps (c4a) and (c4b) if $\tilde{u}$ does not divide $u_{p,k}$ and if l divides $u_{p,k}$.

In this step the elements α^* and β^*, which are elements in an extension of $\mathbf{Z}/n\mathbf{Z}$ of degree $u_{p,k}$ will be transformed to elements in a possibly larger extension of $\mathbf{Z}/n\mathbf{Z}$ of degree u^*. The huge exponentiation, mentioned in (4.4)(a), will be performed in the extension of degree u^*. The transformation will be done in the following way. For all prime powers $\tilde{u} = l^{\tilde{e}_l} \parallel u^*$ with $\tilde{u} \nmid u_{p,k}$ and $l \mid u_{p,k}$ let $u = l^e \parallel u_{p,k}$. First the element η_u is lifted to the extension of degree $\tilde{u}$. If m_l is prime then

$$\eta_u = \sum_{\substack{0 \le i < \tilde{u} \\ i \equiv 1 \bmod u}} \sigma_n^i(\eta_{\tilde{u}}) = \sum_{\substack{0 \le i < \tilde{u} \\ i \equiv 1 \bmod u}} \varsigma_{n,i,\tilde{u}}.$$

In this case one simply has to apply $S^*_{\tilde{u},m_l}$ to a $\tilde{u}$-dimensional vector with only elements equal to 1 at the coordinates with index equivalent to 1 mod u. All other coefficients will be equal to 0. If m_l is not prime then

$$\eta_u = \sigma_n^0(\eta_u) = \varsigma_{n,u/l,u} = \varsigma_{n,u/l,\tilde{u}}.$$

In this case one simply has to apply $S^*_{\tilde{u},m_l}$ to a $\tilde{u}$-dimensional vector with only elements equal to 1 at the coordinate $i = u/l$. All other coefficients will be equal to 0. Next α^* and β^* are written as polynomials of $\eta_{\tilde{u}}$ by applying a Horner-scheme.

(c4a) Let $e \ge 1$ be such that $l^e \parallel u_{p,k}$, and let $u = l^e$. If m_l is prime let $y = (y_i)_{i=0}^{\tilde{u}-1}$ be such that $y_i = 1$ if $i \equiv 1 \bmod u$ and $y_i = 0$ otherwise. If m_l is not prime let $y = (y_i)_{i=0}^{\tilde{u}-1}$ be such that $y_i = 1$ if $i = u/l$ and $y_i = 0$ otherwise. Replace y by $S^*_{\tilde{u},m_l} \cdot y$ (cf. (4.2)(b3)).

(c4b) For $\delta^* = \alpha^*, \beta^*$ do the following. First, write δ^* as a u-dimensional vector $(\delta_i^*)_{i=0}^{u-1}$, i.e., write δ^* as a polynomial in η_u. Next, put $\delta = \delta^*_{u-1}$, and for $i = u-2, u-3, \ldots, 0$ in succession, replace δ by $\delta \cdot y + \delta_i^*$. Finally, replace δ^* by δ.

(c5) The elements α^* and β^* are now elements in an extension of degree u^* of $\mathbf{Z}/n\mathbf{Z}$. Replace α and β by $\alpha \cdot \alpha^*$ and $\beta \cdot \beta^*$, respectively.

The α^* and β^* now satisfy

$$\alpha^* = \prod_{p|r} \tau(\chi_{p,q})^{n_r - \sigma_{n_p}},$$

and

$$\beta^* = \prod_{p|r} \tau(\chi_{p,q})^{r},$$

where $\chi_{p,q}$ is the character of order $p^{\hat{k}_p}$ and conductor q such that $(q,p,h) \in \mathcal{S}$, with $h = o_p(u_{p,\hat{k}_p})$.

(d) Compute $\delta = \alpha \cdot \beta^c$ (cf. (b)). Perform steps (d1) and (d2) for all $p^k \parallel r$.

The element δ satisfies

$$\delta = \prod_{p|r} \tau(\chi_{p,q})^{r\cdot c + n_r - \sigma_{n_p}} = \prod_{p|r} \tau(\chi_{p,q})^{n - \sigma_{n_p}} = \prod_{p|r} \tau(\chi_{p,q})^{n-\sigma_n}.$$

In this step one has to check that δ is a power of an r-th root of unity.

(d1) Compute $\delta^* = \delta^{r/p^k}$ and next perform step (d1a) for those prime powers $\bar{u} = l^{\bar{e}} \parallel u^*$ such that $\bar{u}$ does not divide $u_{p,k}$.

(d1a) Let e be such that $l^e \parallel u_{p,k}$, and let $u = l^e$. Represent δ^* as a $\bar{u}$-dimensional column vector, i.e., write δ^* as a polynomial in $\eta_{\bar{u}}$. Apply S^*_{u,m_l} to the vector consisting of the first u coordinates of the vector $S_{\bar{u},m_l} \cdot \delta^*$, and replace δ^* by the result (cf. (5.2)(b3a) and (5.2)(b3b)). Put $\bar{\delta} = \delta^*$.

This step will only be performed for those prime powers $\bar{u}$ which are not divisors of $u_{p,k}$; in particular if $u^* = u_{p,k}$ this step is skipped. This is in fact the same transformation as performed in step (5.3)(e1b). Explanation can be found there.

(d2) Put $\bar{\gamma} = \gamma_{p,k}$, where $\gamma_{p,k}$ is as in (5.3)(e1). Check that $\bar{\delta} = \bar{\gamma}^i$ for some $i \in \{0, 1, \ldots, p^k - 1\}$; if such an integer i does not exist, then n is composite and the primality test is terminated.

By taking δ to the power r/p^k, one kills all other $\tau(\chi)^{n-\sigma_n}$ in δ, for $\chi \neq \chi_{p,q}$, since these will then be units in $\mathbf{Z}/n\mathbf{Z}$. The exponent j in $\tau(\chi)^{n-\sigma_n} = \gamma^j_{p,k}$ will then be equal to $j = i \cdot (r/p^k)^{-1} \bmod p^k$.

6. FINAL TRIAL DIVISIONS.

Let s, t, u, v, and w be as determined in step (4.1). Let $s_1 = s/\prod_{p|t} p^{o_p(s)}$ and $t_2 = \prod_{p|t} p^{o_p(n^u-1)}$ (cf. (3.4)). Put $\bar{s} = \operatorname{lcm}(s_1 \cdot t_2, v)$, and $\bar{t} = \operatorname{lcm}(t, u, w)$. Check that $\gcd(G, n) = 1$. If this does not hold, then n is composite and the primality test is terminated. If $\mu = \frac{1}{2}$ perform step (a), and if $\mu = \frac{1}{3}$ perform step (b).

If the number n passed all tests described in Sections 3–5, one can prove that all divisors of n that are congruent to $n^i \bmod \bar{s}$ for $i \in \{1, 2, \ldots, \bar{t}\}$, as shown in II.7. The chance that n passed all tests in steps 3–5 without being prime is practically zero. Therefore checking the remaining possibilities for the divisors of n will usually not yield any non-trivial factor of n, but is needed to complete the proof of the primality of n.

All gcd-operations, which were necessary in the course of the algorithm, are done simultaneously, by performing only one gcd. The result of this gcd cannot be 0, since every time one changes G, it is checked that $G \bmod n \neq 0$. So, if $\gcd(G, n) \neq 1$ (which is not very likely), then one is able to find a non-trivial divisor of n.

In the optimization part of the algorithm, it is determined which type of final trial division will be performed in this section. If $\mu = \frac{1}{2}$, the algorithm will examine all numbers $r \equiv n^i \bmod \bar{s}$, for $i = 1, \ldots, \bar{t}$. If $\mu = \frac{1}{3}$, the algorithm presented in [88] will be used to find all divisors in the residue classes $r \equiv n^i \bmod \bar{s}$, for $i = 1, \ldots, \bar{t}$, (cf. II.9).

(a) Let $\bar{s}$ and $\bar{t}$ be as above. Put $\tilde{n} = n \bmod \bar{s}$ with $1 \leq \tilde{n} < \bar{s}$ and let $r = \tilde{n}$. Repeat step (a1) until $\tilde{n} = 1$ but at most $\bar{t}$ times.

In this way a divisor which is at most $\sqrt{n}$, if it exists, will be found. Since $\operatorname{ord}(n \bmod s_1)$ divides t, $\operatorname{ord}(n \bmod t_2) = u$, $\operatorname{ord}(n \bmod v) = w$, and $\operatorname{lcm}(s_1, t_2, v) = \operatorname{lcm}(s_1 \cdot t_2, v) = \bar{s}$, it follows that $\operatorname{ord}(n \bmod \bar{s})$ divides $\operatorname{lcm}(t, u, w) = \bar{t}$, and that step (a1) will be performed at most $\bar{t}$ times.

(a1) If $r = 1$, then n is prime and the primality test is terminated. Otherwise, if $r \leq \sqrt{n}$, check if $r \mid n$; if so, then n is composite and the primality test is terminated. Replace r by $(\tilde{n} \cdot r) \bmod \bar{s}$ in such a way that the new value of r satisfies $0 \leq r < \bar{s}$.

If n is composite, at least one divisor does not exceed $\sqrt{n}$. So in order to find a divisor in this step, one only has to check those values r, which do not exceed $\sqrt{n}$. This observation speeds up this step considerably. The value of r is probably of the same magnitude as $\bar{s}$. Therefore it probably does not make sense to incorporate the value of r in G; otherwise one has to reduce G modulo n each time it is multiplied by a value of r.

(b) Let $\bar{s}$ and $\bar{t}$ be as above. Put $\tilde{n} = n \bmod \bar{s}$ and put $\bar{n} = n^{-1} \bmod \bar{s}$ with $1 \leq \tilde{n}, \bar{n} < \bar{s}$ and let $r = \tilde{n}$, $r^* = \bar{n}$ and $r' = 1$.

Find the first K odd primes that do not divide $\bar{s}$ and group them into products m_j, such that $m_j < \sqrt{M}/3$, where M is the maximal representable single precision integer. Let k denote the number of products m_j, and $h = \prod_{j=1}^{k} m_j$. Next, for each product m_j, put $f_{j,l} = 0$ if l is a square modulo m_j, and put $f_{j,l} = 1$ if l is not a square modulo m_j, for $l = 0, \ldots, m_j - 1$, and put $n_j = n \bmod m_j$ and

$\bar{s}_j = \bar{s} \bmod m_j$, for $j = 1, \ldots, k$.

For $l = 1, \ldots, \lfloor \bar{t}/2 \rfloor$ perform steps (b1) through (b3) until $\tilde{n} = 1$.

The most expensive part of the final trial division using the method presented by [88] is solving a system of two equations in two variables. Solving such a system of equations can be reduced to solving a single quadratic equation in one variable. Therefore finding a solution can essentially be done by applying Newton's method. Before applying Newton's method on a number, it is cheaper to first determine if the number is a perfect square modulo a few small primes. To determine this, a table of all squares modulo these primes, or in fact modulo a product of these primes will be determined.

A divisor which is congruent to $r \equiv n^l \bmod \bar{s}$, if it exists, will be found. A divisor which is congruent to r modulo $\bar{s}$ also implies that there exists a divisor congruent to $r' \equiv r^* \cdot n \bmod \bar{s}$. Since $\operatorname{ord}(n \bmod s_1)$ divides t, $\operatorname{ord}(n \bmod t_2) = u$, $\operatorname{ord}(n \bmod v) = w$, and $\operatorname{lcm}(s_1, t_2, v) = \operatorname{lcm}(s_1 \cdot t_2, v) = \bar{s}$, it follows that $\operatorname{ord}(n \bmod \bar{s})$ divides $\operatorname{lcm}(t, u, w) = \bar{t}$. Secondly, since $r \equiv n^l \bmod \bar{s}$ and $r' \equiv n^{\bar{t}+1-l} \bmod \bar{s}$, that divisor would also be found in the $(\bar{t} + 1 - l)$-th step. Therefore we have that the (non-)existence of divisors congruent to $n^l \bmod \bar{s}$ for $l = 1, \ldots, \bar{t}/2$ implies the (non-)existence of divisors congruent to $n^l \bmod \bar{s}$ for $l = 1, \ldots, \bar{t}$.

(b1) Put $r_j = r \bmod m_j$ and $r'_j = r' \bmod m_j$, for $j = 1, \ldots, k$, by first reducing r and r' modulo h and next reducing its result modulo all the m_j, for $j = 1, \ldots, k$.

Since one does not expect to find any divisors, the expensive step of finding the actual divisors is preceded by a step to check whether it is plausible that n has any divisors. Therefore all operations first will be performed modulo small products of primes.

(b2) Put $a_0 = \bar{s}$, $b_0 = 0$ and $c_0 = 0$, $a_1 = r' \cdot r^* \bmod \bar{s}$ with $0 < a_1 \leq \bar{s}$, $b_1 = 1$ and $c_1 = ((n - r \cdot r')/\bar{s}) \cdot r^* \bmod \bar{s}$, with $0 \leq c_1 < \bar{s}$. Next put $a_{i,j} = a_i \bmod m_j$, $b_{i,j} = b_i \bmod m_j$ and $c_{i,j} = c_i \bmod m_j$ for $j = 1, \ldots, k$ and $i = 0, 1$.

Perform step (b2d) for $i = 0$ and $i = 1$, and steps (b2a) through (b2d) until $a_i = 0$.

In this step we will perform the Euclidean-like algorithm to find the divisors that are congruent $r = n^l \bmod \bar{s}$ as described in [88]. This comes down to solving for each triple (a_i, b_i, c_i) the system of equations

$$a_i \cdot x + b_i \cdot y = c_i$$
$$(x \cdot \bar{s} + r) \cdot (y \cdot \bar{s} + r') = n.$$

(b2a) Put $q = \lfloor a_{i-2}/a_{i-1} \rfloor$ and $a_i = a_{i-2} - q \cdot a_{i-1}$. If $a_i = 0$ and i is odd, replace a_i by $a_i + a_{i-1}$ and q by $q - 1$. Next put $b_i = b_{i-2} - q \cdot b_{i-1}$ and $c_i = c_{i-2} - q \cdot c_{i-1}$. Put $\bar{q} = \lfloor c_i/\bar{s} \rfloor$ and replace c_i by $c_i - \bar{q}\bar{s}$. If $c_i < 0$ then replace c_i by $c_i + \bar{s}$ and $\bar{q}$ by $\bar{q} - 1$.

(b2b) If $q < \sqrt{M}/3$ put $q_j = q \bmod m_j$ for $j = 1, \ldots, k$. Otherwise put $q_j = q$ for $j = 1, \ldots, k$. If $\bar{q} < \sqrt{M}/3$ put $\bar{q}_j = \bar{q} \bmod m_j$ for $j = 1, \ldots, k$. Otherwise put $\bar{q}_j = \bar{q}$ for $j = 1, \ldots, k$.

If q and $\bar{q}$ are less than $\sqrt{M}/3$, all operations modulo the m_j can be done without reducing q and $\bar{q}$ modulo the m_j.

(b2c) For $j = 1, \ldots, k$, put $a_{i,j} = (a_{i-2,j} - q_j \cdot a_{i-1,j}) \bmod m_j$, put $b_{i,j} = (b_{i-2,j} - q_j \cdot b_{i-1,j}) \bmod m_j$, and put $c_{i,j} = (c_{i-2,j} - q_j \cdot c_{i-1,j} - \bar{q}_j \cdot \bar{s}_j) \bmod m_j$.

This step performs the same operations as in step (b2a) modulo the m_j.

(b2d) If i is odd, perform steps (b2d1) and (b2d2) for $z = 0$ and $z = 1$.

If i is even and $c_i = 0$ perform steps (b2d1) and (b2d2) for $z = 0$ and in the case that i is even and $c_i > 0$ perform steps (b2d1) and (b2d2) for $z = 0$ and $z = -1$.

In this step all possible values for c will be checked modulo the m_j, for $j = 1, \ldots, k$. In fact, for i odd only values for c with $2a_i \cdot b_i \le c_i \le n^2/s + a_i \cdot b_i$ are possible. This implies that there is at most one value in this range. It seems to be faster to check first if a c possibly gives rise to a solution and next to check whether c is in the correct range than vice versa.

(b2d1) For $j = 1, \ldots, k$ calculate $d_{1,j} = ((c_{i,j} + z \cdot \bar{s}_j) \cdot \bar{s}_j + a_{i,j} r_j + b_{i,j} r'_j) \bmod m_j$ and $d_{2,j} = a_{i,j} b_{i,j} \bmod m_j$, put $e_j = (d_{1,j}^2 - 4 \cdot d_{2,j} \cdot n) \bmod m_j$, and check if $f_{j,e_j} = 0$. If $f_{j,e_j} \ne 0$ for some $j \le k$ then terminate step (b2d) for this value of z.

In this step it is checked whether the system of two equations gives rise to a solution. This is done by checking if $(c \cdot \bar{s} + a_i \cdot r + b_i \cdot r')^2 - 4a_i \cdot b_i \cdot n$ is a square modulo all m_j, for $j = 1, \ldots, k$. Here c is equal to $c_i + z \cdot \bar{s}$. If the expression is not a square then no solution can be found in this step.

(b2d2) Calculate $d_2 = a_i \cdot b_i$. If i is even or if both i is odd and $2 \cdot d_2 \le c + z \cdot \bar{s} \le n^2/\bar{s} + d_2$, try to find a positive integer e such that $e^2 = d_1^2 - 4 \cdot d_2 \cdot n$, with $d_1 = (d_i + z \cdot \bar{s}) \cdot \bar{s} + a_i r + b_i r'$. If such an integer exists, check whether $(d_1 + e)/(2a_i)$ or $(d_1 - e)/(2a_i)$ are non-trivial divisors of n. If one of these possibilities is a non-trivial divisor of n, then n is composite and the primality test is terminated.

In this step an attempt to find an actual candidate for a divisor of n is made. This is done by solving the system of equations. By identifying u with $a_i \cdot (x \cdot \bar{s} + r)$ and v with $b_i \cdot (y \cdot \bar{s} + r')$, solving the system of equations comes down to finding values for u or v which are the positive roots of the quadratic polynomial $X^2 - (c \cdot \bar{s} + a_i \cdot r + b_i \cdot r')X + a_i \cdot b_i \cdot n$.

(b3) Replace r by $r \cdot \tilde{n} \bmod \bar{s}$, r' by r^*, and r^* by $r^* \cdot \bar{n} \bmod \bar{s}$, with $1 \le r, r^* < \bar{s}$.