Draft.

PROVING PRIMALITY IN ESSENTIALLY QUARTIC EXPECTED TIME

DANIEL J. BERNSTEIN

ABSTRACT. This paper presents a randomized algorithm that, given a prime n, finds and verifies a proof of the primality of n in expected time $(\lg n)^{4+o(1)}$.

1. INTRODUCTION

This paper presents a randomized algorithm that proves primality in $(4+o(1))$-power expected time:

- Section 2 defines certificates, and proves that n is a prime power if it has a certificate.
- Section 3 presents a randomized algorithm that, given a prime n, finds a reasonably small certificate for n in expected time $(\lg n)^{2+o(1)}$.
- Section 4 presents an algorithm to verify a reasonably small certificate in time $(\lg n)^{4+o(1)}$.

One can check whether n is a perfect power in time $(\lg n)^{1+o(1)}$, as explained in [6], so prime-power proving is tantamount to prime proving.

For comparison: The cyclotomic primality-proving method in [2] and [9] has time exponent on the scale of $\lg \lg \lg n$. The elliptic-curve primality-proving method in [11] and [5] is conjectured to take polynomial expected time, but with a larger exponent than $4 + o(1)$. The hyperelliptic-curve primality-proving method in [3] takes polynomial expected time, but with a larger exponent and a much more difficult run-time proof.

The algorithm in this paper is inspired by the recent Agrawal-Kayal-Saxena algorithm in [4], which proves primality in polynomial time. The improvement of the time exponent to $4+o(1)$ relies on an idea by Berrizbeitia in [8], twisting $x-s$ into $\zeta x-s$, $\zeta^2 x-s$, etc. Berrizbeitia used this idea to prove primality in $(4+o(1))$-power expected time for a sparse set of primes, namely the n's for which n^2-1 is divisible by some power of 2 near $(\lg n)^2$.

Qi Cheng in [10] proposed a primality-proving algorithm that is *conjectured* to take $(4+o(1))$-power expected time. Cheng adapted Berrizbeitia's idea to prove primality in $(4+o(1))$-power expected time for a larger set of primes, namely the n's for which $n-1$ is divisible by a prime $e \approx (\lg n)^2$; Cheng then used one elliptic-curve primality-proving step to prove primality of any n, using an auxiliary prime

Date: 20030306.
1991 *Mathematics Subject Classification.* Primary 11Y16.
The author was supported by the National Science Foundation under grant DMS–0140542, and by the Alfred P. Sloan Foundation. He used the libraries at the Mathematical Sciences Research Institute and the University of California at Berkeley.

in the larger set. The conjecture is that the elliptic-curve primality-proving method will not take long to find a suitable auxiliary prime.

The best case for the algorithm in this paper is an even larger set of primes, namely the n's for which $n-1$ has any divisor $e \approx (\lg n)^2$. Perhaps the algorithm will set speed records in that case for tractable values of n. See Section 5.

The $4+o(1)$ theorem in this paper relies on a further generalization from $n-1$ (and n^2-1) to n^d-1 for any $d \in n^{o(1)}$; a standard result from analytic number theory implies that every prime n has a suitable divisor of n^d-1 for some small d. Unfortunately, the time grows quite noticeably as d increases; I doubt that the case $d>1$ will be better in practice than using Cheng's approach to reduce to the case $d=1$.

Historical notes. My generalization was independent of Cheng's paper. I read Berrizbeitia's paper on 26 January 2003 and promptly sent email to a few people saying how I expected it to generalize to any n. I was then told about Cheng's paper, which had been published on 16 January 2003. I published a draft of this paper, with a detailed proof of Theorem 2.2, on 28 January 2003, and announced the result on the NMBRTHRY mailing list on 29 January 2003.

Mihailescu and Avanzi then published a much longer proof of Theorem 2.2, in the special case $S=\{-1\}$, $c_-=0$, $c=0$. They claimed incorrectly that my result was "not proved to work on any input n" and that my result was only for $d=1$.

2. Certificates

Definition 2.1. *Let n, d, and e be positive integers. Let c and c_- be nonnegative integers. Let f be a monic polynomial in $(\mathbf{Z}/n)[y]$ of degree d. Define R as the ring $(\mathbf{Z}/n)[y]/f$. Let r be an element of R. Let S be a subset of R. Assume that*

- *e divides n^d-1;*
- *$r^{n^d-1}=1$ in R;*
- *$r^{(n^d-1)/q}-1$ is a unit in R for each prime q dividing e;*
- *s is a unit in R for all $s \in S$;*
- *$s^e-(s')^e$ is a unit in R for all distinct $s,s' \in S$;*
- *s^e-r is a unit in R for all $s \in S$;*
- *$\binom{e\#S}{c_-}\binom{c}{c_-}\binom{e\#S-c_-+e-1-c}{e-1-c} \ge n^{d\lceil\sqrt{e/3}\rceil}$; and*
- *$(x-s)^{n^d}=r^{(n^d-1)/e}x-s$ in the ring $R[x]/(x^e-r)$ for all $s \in S$.*

Then (d,e,c,c_-,f,r,S) is a **certificate for** *n.*

For example, $(1,840,419,246,y,17,\{1\})$ is a certificate for

$$314159265358979323846264338327950288 41,$$

and $(1,2430,1214,928,y,2,\{1,2\})$ is a certificate for

$$2718281828459045235360287471352662497757247093699959574966967627724076630353547594571.$$

Theorem 2.2. *Let n, d, and e be positive integers. Let c and c_- be nonnegative integers. Let f be a monic polynomial in $(\mathbf{Z}/n)[y]$ of degree d. Define R as the ring $(\mathbf{Z}/n)[y]/f$. Let r be an element of R. Let S be a subset of R. Assume that (d,e,c,c_-,f,r,S) is a certificate for n. Then n is a power of a prime.*

In particular, if n has a certificate and is not a perfect power, then n is prime. This theorem improves on the theorems of Berrizbeitia and Cheng in two basic ways:

- d is allowed to be any positive integer. Berrizbeitia considered only $d \in \{1, 2\}$, and Cheng considered only $d = 1$. Larger d's are important for the $(\lg n)^{4+o(1)}$ theorem in this paper. On the other hand, as discussed in Section 1, the case $d = 1$ is the fastest case, and might end up being the only case used in practice.
- e is allowed to be any positive divisor of $n^d - 1$. Berrizbeitia considered only powers of 2 (although with slightly more general moduli $x^{2^i e} - r$), and Cheng considered only primes e; the proofs relied on e having only one prime divisor. Arbitrary e's are important for the $(\lg n)^{4+o(1)}$ theorem in this paper, and also save time in practice, because they allow many more n's to be handled with $d = 1$.

This theorem also incorporates several smaller time-saving features. It uses negative powers (i.e., allows $c > 0$) as suggested by Voloch, with the optimization suggested by Vaaler. It uses $\sqrt{e/3}$ as suggested by Lenstra, instead of $2\sqrt{e}$ or $\sqrt{e}$ or $\sqrt{e/2}$. It allows $\#S$ to vary; Berrizbeitia and Cheng considered only $\#S = 1$ (or $\#S = 2^i$ for modulus $x^{2^i e} - r$).

Proof. If $n = 1$ then n is a power of a prime, so assume that $n \geq 2$. Let p be a prime divisor of n.

Find an irreducible polynomial g in $\mathbf{F}_p[y]$ dividing the image of f. Then $k = \mathbf{F}_p[y]/g$ is a field.

Write $N = n^d$ and $P = \#k = p^{\deg g}$. Note that P divides N. If $N = P$ then n must be a power of p, so assume that $N > P$.

Define ζ as the image of $r^{(N-1)/e}$ in k. Then ζ has order e in k; consequently e divides $P - 1$. (Indeed, $r^{N-1} = 1$ in R by hypothesis, so $\zeta^e = 1$ in k. Furthermore, if q is a prime dividing e, then $r^{(N-1)/q} - 1$ is a unit in R by hypothesis, so its image $\zeta^{e/q} - 1$ in k is a unit; hence $\zeta^{e/q} \neq 1$ in k.)

If $s \in S$ then $(x - s)^N = r^{(N-1)/e}x - s$ in $R[x]/(x^e - r)$ by hypothesis, so $(x - s)^N = \zeta x - s$ in $k[x]/(x^e - r)$. Substitute $\zeta^m x$ for x for any integer m: $(\zeta^m x - s)^N = \zeta^{m+1}x - s$ in $k[x]/((\zeta^m x)^e - r) = k[x]/(x^e - r)$. Thus $(\zeta^m x - s)^{N^i} = \zeta^{m+i}x - s$ in $k[x]/(x^e - r)$ for any integer $i \geq 0$.

The nonzero element $r^{(P-1)/e}$ of k has eth power $r^{P-1} = 1$, so $r^{(P-1)/e} = \zeta^\ell$ in k for some integer ℓ. Thus $x^P = x^{P-1}x = r^{(P-1)/e}x = \zeta^\ell x$ in $k[x]/(x^e - r)$; consequently $(\zeta^m x - s)^{N^i P^j} = \zeta^{m+i+j\ell}x - s$ in $k[x]/(x^e - r)$ for any integer $j \geq 0$. Define $T = \{\zeta^m x - s : s \in S, m \in \mathbf{Z}\} \subseteq k[x]$; then $t^{N^i P^j} = t(\zeta^{i+j\ell}x)$ in $k[x]/(x^e - r)$ for all $t \in T$, $i \geq 0$, and $j \geq 0$.

Note that $\#T = e\#S$. (There are e powers of ζ and $\#S$ choices of s.) Note also that distinct elements of T are coprime in $k[x]$. (If $\zeta^m x - s$ and $\zeta^{m'}x - s'$ are not coprime then $s\zeta^{m'} = s'\zeta^m$ in k, so $s^e = (s')^e$ in k. If $s \neq s'$ then $s^e - (s')^e$ is a unit in R by hypothesis, so it is a unit in k; contradiction. Thus $s = s'$; so $s\zeta^{m'} = s\zeta^m$; also s is a unit in R by hypothesis, so $\zeta^{m'} = \zeta^m$.)

Define L as the set of $(\alpha, \beta) \in \mathbf{Z} \times \mathbf{Z}$ such that e divides $\alpha + (\beta - \alpha)\ell$; then L is a lattice of determinant e. Define C as the set of $(\alpha, \beta) \in \mathbf{R} \times \mathbf{R}$ such that $\max\{|\alpha| \lg(N/P), |\beta| \lg P, |\alpha \lg(N/P) + \beta \lg P|\} \leq \sqrt{e/3} \lg N$; then C is a closed convex symmetric set of area $3(e/3)(\lg N)^2/(\lg P)\lg(N/P) \geq 4e$. By Minkowski's theorem, there is a nonzero point $(\alpha, \beta) \in L \cap C$. Assume without loss of generality that $\alpha \geq 0$.

$$\begin{tikzcd}
 & k[x]/h \\
R[x]/(x^e - r) \arrow[r, two heads] & k[x]/(x^e - r) \arrow[u, two heads] \\
R[x] \arrow[r, two heads] \arrow[u, two heads] & k[x] \arrow[u, two heads] \\
R = (\mathbf{Z}/n)[y]/f \arrow[r, two heads] \arrow[u] & k = \mathbf{F}_p[y]/g \arrow[u] \\
(\mathbf{Z}/n)[y] \arrow[r, two heads] \arrow[u, two heads] & \mathbf{F}_p[y] \arrow[u, two heads] \\
\mathbf{Z}/n \arrow[r, two heads] \arrow[u] & \mathbf{F}_p \arrow[u]
\end{tikzcd}$$

If $\beta \geq 0$, define $u = (N/P)^\alpha P^\beta$ and $v = 1$; then u and v are positive integers, $\lg u = \alpha \lg(N/P) + \beta \lg P \leq \sqrt{e/3} \lg N$ by definition of C, and $t^{uP^\alpha} = t^{N^\alpha P^\beta} = t(\zeta^{\alpha+\beta\ell}x) = t(\zeta^{\alpha\ell}x) = t^{P^\alpha} = t^{vP^\alpha}$ in $k[x]/(x^e - r)$ for all $t \in T$. If $\beta < 0$, define $u = (N/P)^\alpha$ and $v = P^{-\beta}$; then u and v are positive integers, $\lg u = \alpha \lg(N/P) \leq \sqrt{e/3} \lg N$ and $\lg v = -\beta \lg P \leq \sqrt{e/3} \lg N$ by definition of C, and $t^{uP^\alpha} = t^{N^\alpha} = t(\zeta^\alpha x) = t(\zeta^{(\alpha-\beta)\ell}x) = t^{P^{\alpha-\beta}} = t^{vP^\alpha}$ in $k[x]/(x^e - r)$ for all $t \in T$.

Find an irreducible polynomial h in $k[x]$ dividing the image of $x^e - r$. Then $k[x]/h$ is a field, and $t^{uP^\alpha} = t^{vP^\alpha}$ in $k[x]/h$ for all $t \in T$. Thus $t^u = t^v$ in $k[x]/h$ for all $t \in T$, since Pth powering is invertible in $k[x]/h$.

Each element of T is a unit in $k[x]/h$. (The remainder $(x^e - r) \bmod (\zeta^m x - s)$ is $(s\zeta^{-m})^e - r = s^e - r$, which is a unit in k by hypothesis; so $x^e - r$ and $\zeta^m x - s$ are coprime in $k[x]$; so h and $\zeta^m x - s$ are coprime in $k[x]$.)

Consider functions $a : T \to \mathbf{Z}$ such that, first, $\#\{t \in T : a(t) < 0\} = c_-$; second, $\sum_t -a(t)[a(t) < 0] \leq c$; and, third, $\sum_t a(t)[a(t) \geq 0] \leq e - 1 - c$. There are $\binom{\#T}{c_-}\binom{c}{c_-}\binom{\#T - c_- + e - 1 - c}{e - 1 - c} \geq N^{\lceil \sqrt{e/3} \rceil} \geq N^{\sqrt{e/3}} > |u - v|$ such functions.

Assume that a, b are two such functions with $\prod_{t \in T} t^{a(t)} = \prod_{t \in T} t^{b(t)}$ in $(k[x]/h)^*$. Clear denominators to obtain polynomials $A = \prod_{t \in T} t^{a(t)[a(t) \geq 0] - b(t)[b(t) < 0]} \in k[x]$ and $B = \prod_{t \in T} t^{b(t)[b(t) \geq 0] - a(t)[a(t) < 0]} \in k[x]$ with $A = B$ in $k[x]/h$. Each element $t \in T$ satisfies $t^{N^i} = t(\zeta^i x)$ in $k[x]/h$, so $A(\zeta^i x) = A^{N^i} = B^{N^i} = B(\zeta^i x)$ in $k[x]/h$. Thus $A - B$ has roots $x, \zeta x, \zeta^2 x, \ldots, \zeta^{e-1} x$ in $k[x]/h$; these roots are distinct, since x is invertible in $k[x]/h$; but $A - B$ has degree at most $c + e - 1 - c < e$, so it cannot have e roots unless it is zero. Thus $A = B$ in $k[x]$; so $a(t)[a(t) \geq 0] - b(t)[b(t) < 0] = b(t)[b(t) \geq 0] - a(t)[a(t) < 0]$ by unique factorization into coprimes; so $a(t) = b(t)$.

Thus there are more than $|u - v|$ products $\prod_{t \in T} t^{a(t)}$ in $(k[x]/h)^*$. Each product π satisfies $\pi^u = \pi^v$ in $(k[x]/h)^*$, hence $\pi^{|u-v|} = 1$; but a field cannot have more than $|u - v|$ roots of $\pi^{|u-v|} = 1$ if $|u - v| > 0$. Thus $u = v$. In other words, $N^\alpha = P^{\alpha-\beta}$. If $\alpha = 0$ then $P^{-\beta} = 0$ so $\beta = 0$, but (α, β) was nonzero by construction; contradiction. Hence n is a power of p. $\square$

Notes on the proof. If $N = P^2$ then n must be a power of p; if $N \neq P^2$ then the area of C cannot equal $4e$; so one can use a slightly simpler form of Minkowski's theorem.

The proof would still work if $N^{\lceil\sqrt{e/3}\rceil}$ were replaced by $N^{\sqrt{e/3}}$ in the definition of a certificate. However, this change would complicate certificate testing, and would have very little benefit.

The Agrawal-Kayal-Saxena proof considered the subgroup of $(k[x]/h)^*$ generated by images of elements of T; the subgroup has size larger than $|u-v|$, so a generator g of the subgroup has order larger than $|u-v|$, so the equation $g^u = g^v$ implies that $u = v$. Kiran Kedlaya pointed out to me the simpler argument that $\pi^u = \pi^v$ has at most $|u-v|$ nonzero roots in a field unless $u = v$; this avoids constructing a generator.

Can we prove better lower bounds on the group size, i.e., on the number of products $\prod_t t^{a(t)}$ in $(k[x]/h)^*$? This is the same as the number of products in $k[x]/(x^e - r)$: if $A = B$ in $k[x]/h$ then $A - B$ has roots $x, \zeta x, \zeta^2 x, \ldots, \zeta^{e-1}x$ in $k[x]/h$ as above, so $A - B$ is divisible by $x^e - r$. It would be surprising for the number of products to be much smaller than P^e. A lower bound close to P^e, with $\#S$ small, would accelerate the algorithm from essentially quartic time to essentially cubic time.

3. Finding a certificate

Every prime n has a certificate of the form $(d, e, 0, 0, f, r, \{1\})$ with $d \in (\lg n)^{o(1)}$ and $e \in (\lg n)^{2+o(1)}$. Furthermore, this certificate can be found in expected time $(\lg n)^{2+o(1)}$. This section discusses the construction of d and e, then the construction of f, and finally the construction of r.

As discussed in Section 4, one can then verify that this is a certificate for n in time $(\lg n)^{4+o(1)}$.

As discussed in Section 5, one can reduce the $o(1)$ by choosing certificates more carefully.

Finding d and e. There is a positive integer d such that $n^d - 1$ has a divisor $e \geq 6$ between $d^2 \lceil \lg n \rceil^2$ and $(d+1)d^2 \lceil \lg n \rceil^2$, by Theorem 3.1. The smallest such d is in $\exp(O(\lg\lg\lg n \lg\lg\lg\lg n))$ by Theorem 3.2.

To compute the smallest d, one can try $d = 1$, then $d = 2$, etc.; success will occur within $(\lg n)^{o(1)}$ tries. For each d, there are $(\lg n)^{2+o(1)}$ possible divisors e between $d^2 \lceil \lg n \rceil^2$ and $(d+1)d^2 \lceil \lg n \rceil^2$, each e having $(\lg n)^{o(1)}$ bits. One can compute $n^d - 1$ modulo all these e's simultaneously in time $(\lg n)^{2+o(1)}$; see, e.g., [7, Section 18].

Theorem 3.1. *Let n be an integer with $n \geq 2$. Then there exists a positive integer d such that $n^d - 1$ has a divisor $e \geq 6$ with $d^2 \lceil \lg n \rceil^2 \leq e < (d+1)d^2 \lceil \lg n \rceil^2$.*

Proof. Observe that $n^2 - 1 \geq \lceil \lg n \rceil^2$ and $n^6 + n^4 + n^2 + 1 \geq 64$. Thus $e \geq 64 \lceil \lg n \rceil^2$ where $e = n^8 - 1$. Define d as the largest multiple of 8 with $d^2 \leq e/\lceil \lg n \rceil^2$. Then $d \geq 8$, and e divides $n^d - 1$. Furthermore, $d^3 \geq 16d + 64$, so $d^3 + d^2 \geq d^2 + 16d + 64 = (d+8)^2 > e/\lceil \lg n \rceil^2$. □

Theorem 3.2. *There are constants n_0 and α such that, for every prime number $n \geq n_0$, there is a positive integer $d \leq \exp(\alpha \log(3 \log\lg n) \log\log(3 \log\lg n))$ such that $n^d - 1$ has a divisor $e \geq 6$ with $d^2 \lceil \lg n \rceil^2 \leq e < (d+1)d^2 \lceil \lg n \rceil^2$.*

This is a typical application of a well-known theorem of Adleman, Pomerance, Rumely, and Odlyzko; see [2, Theorem 3]. The point is that the product of the small primes dividing $n^d - 1$ grows, at a minimum, almost exponentially with d. Older theorems suffice for the bound $d \in (\lg n)^{o(1)}$.

It is overkill to assume that n is prime; what matters is that n has no tiny prime divisors.

Proof. Choose α such that d below always exists, and choose $n_0 > 8$ such that H below always exists.

Given $n \ge n_0$, select a real number $H > 16$ such that $H \le (\lg n)^3$, $D + 1 < n$, and $H/D^2 \ge \lceil \lg n \rceil^2$, where $D = \exp(\alpha \log\log H \log\log\log H)$. Asymptotically one can take H in $(\lg n)^{2+o(1)}$, and thus D in $(\lg n)^{o(1)}$, satisfying $H/D^2 \ge \lceil \lg n \rceil^2$, so the extra constraints $H \le (\lg n)^3$ and $D + 1 < n$ are automatically satisfied for n large enough. Note that $D \le \exp(\alpha \log(3 \log \lg n) \log\log(3 \log \lg n))$.

By [2, Theorem 3], there is a positive integer $d \le D$ such that $H \le \pi$, where π is the product of the primes q with $q - 1$ dividing d.

Now $d^2 \lceil \lg n \rceil^2 \le D^2 \lceil \lg n \rceil^2 \le H \le \pi$. Find the smallest positive integer $e \ge d^2 \lceil \lg n \rceil^2$ dividing π; note that $e \ge 6$ since $n > 8$. Each prime q is at most $d + 1$, so e must be smaller than $(d + 1)d^2 \lceil \lg n \rceil^2$.

Finally, e divides $n^d - 1$. Indeed, each prime q is at most $d + 1 \le D + 1 < n$, so q does not divide n, so q divides $n^{q-1} - 1$, hence $n^d - 1$. □

Finding f. For every prime number n and positive integer d, there is a monic irreducible polynomial $f \in (\mathbf{Z}/n)[y]$ of degree d.

One standard way to find f is to generate a uniform random monic polynomial f of degree d, see if it is irreducible, and try again if not. There are many choices of f that work: the expected number of trials is approximately d, which is in $(\lg n)^{o(1)}$ in Theorem 3.2.

Another standard way to find f is to search systematically through polynomials with small coefficients. This avoids randomness, and produces polynomials f that take very little space to write down. It has the disadvantage that the number of trials is no longer guaranteed to be small.

One way to check the irreducibility of a single f is to see whether f has factors in common with $x^n - x, x^{n^2} - x, \ldots, x^{n^{d-1}} - x$. Each nth powering in $(\mathbf{Z}/n)[y]/f$ takes time $(\lg n)^{2+o(1)}$ if $d \in (\lg n)^{o(1)}$, so the total time for an irreducibility test is $(\lg n)^{2+o(1)}$.

There is much more to say about the construction of irreducible polynomials. I should cite some recent survey. I should say a little about the impact of GRH, and about the improvements available as d grows.

Finding r. For every prime number n, positive integer d, positive integer e dividing $n^d - 1$, and monic irreducible polynomial $f \in (\mathbf{Z}/n)[y]$ of degree d, there is an element r of the field $R = (\mathbf{Z}/n)[y]/f$ such that $r^{(n^d-1)/e}$ has order e; for example, any generator r of R^*. Furthermore, if $e \ge 6$ and $e \ge d^2 \lceil \lg n \rceil^2$ as in Theorems 3.1 and 3.2, then $(d, e, 0, 0, f, r, \{1\})$ is a certificate for n by Theorem 3.3.

Finding elements of specified order is analogous to (and in many ways tied to) finding irreducible polynomials of specified degree. One standard way to find r is to generate a uniform random element r of $R - \{0\}$, see if $r^{(n^d-1)/e}$ has order e, and try again if not. There are many choices of r that work: the expected number of trials is the product of $q/(q-1)$ for primes q dividing e, which is in $(\lg n)^{o(1)}$ if $e \in (\lg n)^{2+o(1)}$.

Another standard way to find r is to search systematically through elements of $(\mathbf{Z}/n)[y]/f$ with small coefficients. This avoids randomness, and produces r's that take very little space to write down; the reader may have noticed that the examples of certificates in Section 2 are very short. It has the disadvantage that the number of trials is no longer guaranteed to be small.

One way to check whether $r^{(n^d-1)/e}$ has order e is to check that $r^{n^d-1} = 1$ and that $r^{(n^d-1)/q} \neq 1$ for each prime q dividing e. There are only $(\lg n)^{o(1)}$ such primes q if $e \in (\lg n)^{2+o(1)}$, and all of them are easy to find since e is small; the main work is to compute $r^{(n^d-1)/e}$ in the first place, which takes time $(\lg n)^{2+o(1)}$.

I should, again, point to the literature: combining orders, using GRH, merging exponentiations, etc.

Theorem 3.3. *Let n be a prime number. Let d be a positive integer. Let $e \geq 6$ be a divisor of $n^d - 1$ such that $d^2 \lceil \lg n \rceil^2 \leq e$. Let f be a monic irreducible polynomial in $(\mathbf{Z}/n)[y]$ of degree d. Let r be an element of the ring $(\mathbf{Z}/n)[y]/f$ such that $r^{(n^d-1)/e}$ has order e. Then $(d, e, 0, 0, f, r, \{1\})$ is a certificate for n.*

Proof. Write $R = (\mathbf{Z}/n)[y]/f$. Observe that R is a field.

By hypothesis, n, d, and e are positive integers; e divides $n^d - 1$; $r^{n^d-1} = (r^{(n^d-1)/e})^e = 1$; if q is a prime dividing e, then $r^{(n^d-1)/q} - 1 = (r^{(n^d-1)/e})^{e/q} - 1 \neq 0$, so $r^{(n^d-1)/q} - 1$ is a unit; and $e > 1$, so $r^{(n^d-1)/e} \neq 1$, so $r \neq 1$, so $1^e - r = 1 - r$ is a unit.

Furthermore, $e \geq 6$, so $\binom{2e-1}{e-1} \geq 2^e$ and $e \geq (\sqrt{e/3} + 1)^2$. Thus $(\lg \binom{2e-1}{e-1})^2 \geq e^2 \geq (\sqrt{e/3}+1)^2 e \geq (\sqrt{e/3}+1)^2 d^2 \lceil \lg n \rceil^2 \geq \lceil \sqrt{e/3} \rceil^2 d^2 (\lg n)^2$; i.e., $\binom{2e-1}{e-1} \geq n^{d\lceil \sqrt{e/3} \rceil}$.

Finally, $(x-1)^{n^d} = x^{n^d} - 1 = x^{n^d-1}x - 1 = r^{(n^d-1)/e}x - 1$ in $R[x]/(x^e - r)$. □

4. Checking a certificate

This section presents an algorithm that decides whether (d, e, c, c_-, f, r, S) is a certificate for n, given positive integers n, d, e, nonnegative integers c and c_-, a monic degree-d polynomial $f \in (\mathbf{Z}/n)[y]$, an element r of $R = (\mathbf{Z}/n)[y]/f$, and a subset S of R.

This algorithm takes time $(\lg n)^{4+o(1)}$ for reasonably small inputs. "Reasonably small" means that d is in $(\lg n)^{o(1)}$; $\#S$ is in $(\lg n)^{o(1)}$; e is at most $(\lg n)^{2+o(1)}$; and the product $\binom{e\#S}{c_-}\binom{c}{c_-}\binom{e\#S-c_-+e-1-c}{e-1-c}$ has at most $(\lg n)^{2+o(1)}$ digits.

Note that the certificates $(d, e, 0, 0, f, r, \{1\})$ constructed in Section 3, with $d \in (\lg n)^{o(1)}$ and $e \in (\lg n)^{2+o(1)}$, are reasonably small. The product of binomial coefficients is $\binom{2e-1}{e-1}$, which has $O(e)$ digits.

The reader is assumed to be familiar with fast multiplication. See, e.g., [7].

The basic conditions. Computing $n^d - 1$, and checking that it is divisible by e, takes time $(\lg n)^{1+o(1)}$.

Multiplying in $\mathbf{Z}/n$ takes time $(\lg n)^{1+o(1)}$. Thus multiplying in R takes time $(\lg n)^{1+o(1)}$. Computing the $n^d - 1$ power of r in R takes $(\lg n)^{1+o(1)}$ multiplications in R, hence time $(\lg n)^{2+o(1)}$.

The units. There are $(\lg n)^{o(1)}$ primes q dividing e; finding them by trial division takes time $(\lg n)^{1+o(1)}$. Computing the $(n^d-1)/q$ power of r in R takes time $(\lg n)^{2+o(1)}$. Checking whether $r^{(n^d-1)/q}-1$ is a unit in R takes time $(\lg n)^{1+o(1)}$.

Computing s^e in R for each $s \in S$ takes time $(\lg n)^{1+o(1)}$. Checking all the remaining units takes time $(\lg n)^{1+o(1)}$.

The binomial coefficients. Computing the product of binomial coefficients takes time $(\lg n)^{2+o(1)}$. I should say more about the binomial-coefficient computation; maybe I should simplify the definition of "reasonably small."

Computing $n^{d\lceil\sqrt{e/3}\rceil}$ takes time $(\lg n)^{2+o(1)}$.

The big exponentiation. Multiplying in $R[x]/(x^e-r)$ takes time $(\lg n)^{3+o(1)}$. Computing each $(x-s)^{n^d}$ in $R[x]/(x^e-r)$ takes $(\lg n)^{1+o(1)}$ multiplications in $R[x]/(x^e-r)$, hence time $(\lg n)^{4+o(1)}$.

5. Optimizations and practical performance

This section looks at verification speed more closely in the important case $d=1$.

Choosing c and c_-. The choice $c=c_-=0$ in Section 3 is far from optimal. If $c\approx\alpha e$ and $c_-\approx\beta e$ then the product $\binom{e\#S}{c_-}\binom{c}{c_-}\binom{e\#S-c_-+e-1-c}{e-1-c}$ is approximately $\exp(e\gamma)$ where $\gamma=(\#S-\beta+1-\alpha)\log(\#S-\beta+1-\alpha)+\#S\log\#S+\alpha\log\alpha-2(\#S-\beta)\log(\#S-\beta)-2\beta\log\beta-(\alpha-\beta)\log(\alpha-\beta)-(1-\alpha)\log(1-\alpha)$.

One can, either with a computer program or by hand, easily find α and β that maximize γ for any given $\#S$. Any choice of $c\approx\alpha e$ and $c_-\approx\beta e$ is reasonable; a small amount of additional searching will locate the optimal c and c_-.

It turns out that the optimal α and β have simple expressions: $\alpha=1/2$ and $\beta=(\#S+1-\sqrt{\#S^2+1})/2$. For example, say $\#S=1$. The product of binomial coefficients is about $5.828427\ldots^e$ if one takes $c\approx e/2$ and $c_-\approx(2-\sqrt{2})e/2=(0.2928932\ldots)e$. For comparison: The product of binomial coefficients is about 4^e if one takes $c=0$ and $c_-=0$.

Choosing e and $\#S$. Say there are many possibilities for $(e,\#S)$—or, in Cheng's method, many possibilities for an auxiliary $(n,e,\#S)$—such that the maximized product of binomial coefficients exceeds $n^{\lceil\sqrt{e/3}\rceil}$. One should choose the possibility that minimizes verification time.

As a first approximation, this means minimizing $e\#S$: verification time can be crudely modeled as $(\lg n)^2 e\#S$. The following table shows $e\#S/(\lg n)^2$ as a function of $e/(\lg n)^2$, when $\#S$ is chosen as small as possible:

$\#S$	works for $e/(\lg n)^2$ between about	and about	so $e\#S/(\lg n)^2$ is between about	and about
1	0.051540...	∞	0.051540...	∞
2	0.027664...	0.051540...	0.055328...	0.103081...
3	0.020415...	0.027664...	0.061247...	0.082992...
4	0.016832...	0.020415...	0.067328...	0.081663...
5	0.014653...	0.016832...	0.073269...	0.084160...
6	0.013169...	0.014653...	0.079017...	0.087923...
7	0.012082...	0.013169...	0.084575...	0.092187...
8	0.011244...	0.012082...	0.089958...	0.096658...

If e drops substantially below $0.01(\lg n)^2$ then $e\#S$ explodes: $\#S = 100$ works for $e/(\lg n)^2$ down to about $0.004037\ldots$; $\#S = 1000$ works for $e/(\lg n)^2$ down to about $0.002164\ldots$; $\#S = 10000$ works for $e/(\lg n)^2$ down to about $0.001347\ldots$; and so on.

A more precise model of verification time includes logarithmic factors that grow with e but not with $\#S$. Reducing e at the expense of $\#S$ often saves time even if it increases $e\#S$.

Multiplying quickly. One can square an element of $(\mathbf{Z}/n)[x]/(x^e - r)$ as follows:

- Lift to $\mathbf{Z}[x]$, obtaining polynomials of degree at most $e-1$ with coefficients between $-n/2$ and $n/2$.
- Choose p so that $2^p > e(n/2)^2$, and map to $\mathbf{Z}[x]/(x - 2^p) \cong \mathbf{Z}$, obtaining an integer with approximately $2e \lg n$ bits.
- Square in $\mathbf{Z}$.
- Recover the product in $\mathbf{Z}[x]$.
- Reduce modulo $x^e - r$. This is particularly easy if r is small.
- Reduce each coefficient modulo n.

The overall speed of certificate verification depends crucially on the details of these steps.

For example, one might square a polynomial with the following C code, using the GMP 4.1.2 library:

```
mpz_set_si(t1,0);
for (j = 0;j < e;++j)
  for (b = 0;b < nbits;++b)
    if (mpz_tstbit(poly[j],b))
      mpz_setbit(t1,j * padbits + b);
mpz_mul(t1,t1,t1);
for (j = 0;j < e;++j) {
  mpz_set_si(t2,0);
  for (b = 0;b < padbits;++b)
    if (mpz_tstbit(t1,(j + e) * padbits + b))
      mpz_setbit(t2,b);
  mpz_set_si(t3,r);
  mpz_mul(t2,t2,t3);
  mpz_set_si(t3,0);
  for (b = 0;b < padbits;++b)
    if (mpz_tstbit(t1,j * padbits + b))
      mpz_setbit(t3,b);
  mpz_add(t2,t3,t2);
  mpz_mod(poly[j],t2,n);
}
```

This code uses approximately $2\cdot 10^{11}$ clock cycles on a Pentium III-800 to verify the aforementioned certificate $(1, 2430, 1214, 928, y, 2, \{1, 2\})$ for the prime $\lfloor 10^{84} \exp 1 \rfloor$. A large fraction of the time is spent testing and setting bits; GMP does not offer any good way to copy a stretch of bits from one number to another. Another large fraction of the time is spent in integer squaring, which can be sped up considerably. I would not be surprised to see an order of magnitude speed improvement.

References

[1] —, *Proceedings of the 18th annual ACM symposium on theory of computing*, Association for Computing Machinery, New York, 1986. ISBN 0–89791–193–8.

[2] Leonard M. Adleman, Carl Pomerance, Robert S. Rumely, *On distinguishing prime numbers from composite numbers*, Annals of Mathematics **117** (1983), 173–206. ISSN 0003–486X. MR 84e:10008.

[3] Leonard M. Adleman, Ming-Deh A. Huang, *Primality testing and abelian varieties over finite fields*, Lecture Notes in Mathematics, 1512, Springer-Verlag, Berlin, 1992. ISBN 3–540–55308–8. MR 93g:11128.

[4] Manindra Agrawal, Neeraj Kayal, Nitin Saxena, *PRIMES is in P* (2002). Available from `http://www.cse.iitk.ac.in/news/primality.html`.

[5] A. O. L. Atkin, Francois Morain, *Finding suitable curves for the elliptic curve method of factorization*, Mathematics of Computation **60** (1993), 399–405. ISSN 0025–5718. MR 93k:11115.

[6] Daniel J. Bernstein, *Detecting perfect powers in essentially linear time*, Mathematics of Computation **67** (1998), 1253–1283. ISSN 0025–5718. MR 98j:11121. Available from `http://cr.yp.to/papers.html`.

[7] Daniel J. Bernstein, *Fast multiplication and its applications*. Available from `http://cr.yp.to/papers.html`.

[8] Pedro Berrizbeitia, *Sharpening PRIMES is in P for a large family of numbers* (2002). Available from `http://arxiv.org/abs/math.NT/0211334`.

[9] Wieb Bosma, Marc-Paul van der Hulst, *Primality proving with cyclotomy*, Ph.D. thesis, Universiteit van Amsterdam, 1990.

[10] Qi Cheng, *Primality proving via one round in ECPP and one iteration in AKS* (2003). Available from `http://www.cs.ou.edu/~qcheng/`.

[11] Shafi Goldwasser, Joe Kilian, *Almost all primes can be quickly certified*, in [1] (1986), 316–329; see also newer version in [12].

[12] Shafi Goldwasser, Joe Kilian, *Primality testing using elliptic curves*, Journal of the ACM **46** (1999), 450–472; see also older version in [11]. ISSN 0004–5411. MR 2002e:11182.

Department of Mathematics, Statistics, and Computer Science (M/C 249), The University of Illinois at Chicago, Chicago, IL 60607–7045

E-mail address: `djb@cr.yp.to`