Primality Testing Revisited*

J.H. Davenport
School of Mathematical Sciences
University of Bath
Bath BA2 7AY
England
jhd@maths.bath.ac.uk

Abstract. Rabin's algorithm is commonly used in computer algebra systems and elsewhere for primality testing. This paper presents an experience with this in the Axiom* computer algebra system. As a result of this experience, we suggest certain strengthenings of the algorithm.

Introduction

It is customary in computer algebra to use the algorithm presented by Rabin [1980] to determine if numbers are prime (and primes are needed throughout algebraic algorithms). As is well known, a single iteration of Rabin's algorithm, applied to the number N, has probability at most 0.25 of reporting "N is probably prime", when in fact N is composite. For most N, the probability is much less than 0.25. Here, "probability" refers to the fact that Rabin's algorithm begins with the choice of a "random" seed x, not congruent to 0 modulo N. In practice, however, true randomness is hard to achieve, and computer algebra systems often use a fixed set of x — for example Axiom release 1 uses the set

$$\{3, 5, 7, 11, 13, 17, 19, 23, 29, 31\}. \tag{1}$$

As Pomerance *et al.* [1980] point out, there is some sense in using primes as the x-values: for example the value $x = 4$ gives no more information than the value $x = 2$, and the value $x = 6$ can only give more information than 2 and 3 under rare circumstances (in particular, we need the 2-part of the orders of 2 and 3 to differ, but be adjacent. However, this argument is heuristic: Pinch [1993] reports that, up to 10^{12}, the set $\{2, 3, 5, 7\}$ gives nine false reports, whereas the set $\{2, 3, 7, 10\}$ only gives seven). By Rabin's theorem, a group-theoretic proof of which is given in Davenport & Smith [1987], 10 elements in the set gives a probability less than 1 in 10^6 of giving the wrong answer. In fact, it is possible to do rather better than this: for example Damgård & Landrock [1991] show that, for 256-bit integers, six tests give a probability of less than 2^{-51} of giving the wrong answer.

Nevertheless, given any such fixed set of x values, there are probably some composite N for which all the x in the set report "N is probably prime". In particular Jaeschke [1991] reports that the 29–digit number

$$56897193526942024370326972321 = 137716125329053 \cdot 413148375987157$$

has this property for the set (1) (and indeed also for the base 2). For brevity, let us call this number J — "the Jaeschke number", and its factors J_1 and J_2 respectively. Now

$$J = 1 + 2^5 \cdot 1778037297716938261572717885,$$

* An earlier version of this paper was published in Proc. ISSAC 92 (ed. P.S. Wang, ACM, New York, 1992) pp. 123–129. Permission to republish is gratefully acknowledged. Axiom is a trade mark of NAG Ltd.

so Rabin's algorithm will begin by raising each element of (1) to the power

$$1778037297716938261572717885$$

(modulo J), thus getting

$$\begin{array}{rcccc}
3 & \longrightarrow & 1 & \overset{\text{squaring}}{\longrightarrow} & 1 \\
5 & \longrightarrow & 4199061068131012714084074012 & \overset{\text{squaring}}{\longrightarrow} & J-1 \\
7 & \longrightarrow & 40249683417692701270027867121 & \overset{\text{squaring}}{\longrightarrow} & J-1 \\
11 & \longrightarrow & 40249683417692701270027867121 & \overset{\text{squaring}}{\longrightarrow} & J-1 \\
13 & \longrightarrow & 52698132458811011656242898309 & \overset{\text{squaring}}{\longrightarrow} & J-1 \\
17 & \longrightarrow & 4199061068131012714084074012 & \overset{\text{squaring}}{\longrightarrow} & J-1 \\
19 & \longrightarrow & 40249683417692701270027867121 & \overset{\text{squaring}}{\longrightarrow} & J-1 \\
23 & \longrightarrow & 16647510109249323100299105200 & \overset{\text{squaring}}{\longrightarrow} & J-1 \\
29 & \longrightarrow & 40249683417692701270027867121 & \overset{\text{squaring}}{\longrightarrow} & J-1 \\
31 & \longrightarrow & 1 & \overset{\text{squaring}}{\longrightarrow} & 1
\end{array}$$

Hence, for all these x-values, Rabin's algorithm will say "J is probably prime", since we arrive at a value of 1 in our repeated squaring, either directly ($x = 3$ and $x = 31$) or via $J - 1$. However, this table indicates (to the suspicious human eye) two things.

(A) The first is that -1 appears to have four square roots modulo J, viz.

$$\begin{array}{ll}
4199061068131012714084074012, & 40249683417692701270027867121, \\
16647510109249323100299105200, & 52698132458811011656242898309.
\end{array}$$

This contradicts Lagrange's theorem, so J cannot be a prime.

(B) The second is that, if J were prime, we would expect about half of the elements of (1) to be quadratic non-residues, and hence to need five squarings to reach 1 (4 to reach $J - 1$), about a quarter to be quadratic residues, but quartic non-residues, hence needing three squarings to reach $J - 1$, and only an eighth to be octic residues or better, and to reach $J - 1$ in at most one squaring. Hence, if J were prime, we have observed an event of probablility $(1/8)^{10}$ — less than 1 in 10^9.

Much of the paper is taken up with a detailed exploration of these observations and their generalisations. We observe that, at least in principle, we are only concerned with the problem of testing relatively large numbers: numbers less than $25 \cdot 10^9$ are covered by Pomerance *et al.* [1980], less than 10^{12} and less that 10^{13} by Pinch [1993] and less than 341550071728321 by Jaeschke [1993].

Rabin revisited.

Throughout this paper, we assume that all integers to be tested for primality are positive and odd. We use the standard notation

$$\phi(n) = |\{x : 0 < x \leq n; \gcd(x, n) = 1\}|$$

from which the Chinese Remainder Theorem gives (here and always, we assume in such formulae that the p_i are distinct primes)

$$\phi(\prod p_i^{\alpha_i}) = \prod p_i^{\alpha_i - 1}(p_i - 1).$$

In addition, we introduce

$$\hat{\phi}(\prod p_i^{\alpha_i}) = \mathrm{lcm}\left(p_i^{\alpha_i - 1}(p_i - 1)\right).$$

Clearly $\hat{\phi}(n)|\phi(n)$.

The Fermat-Euler Theorem states that, if $\gcd(x, n) = 1$, then $x^{\phi(n)} \equiv 1 \pmod{n}$. This leads to what might be called the Fermat primality test: pick some $x \not\equiv 0 \pmod{n}$ and compute $x^{n-1} \pmod{n}$. If this is not 1, then $n - 1 \neq \phi(n)$, so n cannot be prime. If $x^{n-1} \equiv 1 \pmod{n}$, but n is not prime, we say that n is a pseudoprime(x). All composite numbers are pseudoprime(1).

However, the Chinese Remainder Theorem implies a stronger result than the Fermat-Euler Theorem, viz. the following.

Lemma 1. *If* $\gcd(x, n) = 1$, *then* $x^{\hat{\phi}(n)} \equiv 1 \pmod{n}$. *Furthermore,* $\hat{\phi}(n)$ *is minimal with this property.*

A non-prime number N for which $\hat{\phi}(N)|N - 1$ is called a Carmichael number. Any Carmichael number has to have at least three prime factors. (If pq were a Carmichael number, then $pq \equiv 1 \pmod{p-1}$, so $q \equiv 1 \pmod{p-1}$ and $q \geq p$. Similarly, $p \equiv 1 \pmod{q-1}$ and $p \geq q$. So $p = q$, but $\hat{\phi}(p^2) = p(p-1)$, which can never divide $p^2 - 1$.) These numbers, of which we now know that there are infinitely many [Alford *et al.*, 1992], are the bane of the Fermat primality test, since, unless we hit on an x with $\gcd(x, n) \not\equiv 1$, we will always have $x^{N-1} \equiv 1 \pmod{N}$.

Hence we need a stronger test: Rabin's test, which is finer than the Fermat test since, instead of computing x^{N-1}, it writes $N - 1 = 2^k \cdot l$ with l odd, and then considers each of $x^l, x^{2l}, \ldots, x^{2^k l}$ (each obtained by squaring the previous one, and all computed modulo N). If the last is not 1, we have a non-prime by the Fermat test. If the first is 1 or $N - 1$, we know nothing and say "N is probably prime". If, however,the first 1 is preceded by a number other than $N - 1$, we can assert that N is definitely composite, since we have found a square root of unity other than 1 and $N - 1$.

Another way of seeing the difference between Rabin's test and the Fermat test is to say that we are analysing the 2-part of the order of x modulo N more carefully. We reply "N is definitely not prime" if the order of x has different 2-parts modulo different factors of N.

Our starting code for Axiom's implementation (slightly modified from that distributed with Axiom release 1, in particular to split out the auxiliary function `rabinProvesComposite`, but using the same algorithm) is given below, where we have numbered the lines for ease of reference. We remind the reader that Axiom comments begin with `--`, and continue to the end of the line. `I` is the datatype of n, and can be thought of as being the integers. `smallPrimes` is a list of the primes up to 313, and `nextSmallPrime` is therefore 317.

```
[ 1]   prime? n ==
[ 2]      n < two => false
[ 3]      n < nextSmallPrime => member?(n, smallPrimes)
[ 4]      not one? gcd(n, productSmallPrimes) => false
[ 5]      n < nextSmallPrimeSquared => true
```

```
[ 6]      nm1:=n-1
[ 7]      q := (nm1) quo two
[ 8]      for k in 1.. while not odd? q repeat q := q quo two
[ 9]      -- q = (n-1) quo 2**k for largest possible k
[10]      mn := minIndex smallPrimes
[11]      for i in mn+1..mn+10 repeat
[12]          rabinProvesComposite(smallPrimes i,n,nm1,q,k) => return false
[13]      true
[14]
[15]   rabinProvesComposite(p,n,nm1,q,k) ==
[16]         -- nm1 = n-1 = q*2**k; q odd
[17]         -- probability false for  n composite is < 1/4
[18]         -- for most n this probability is much less than 1/4
[19]         t := powmod(p, q, n)
[20]         -- neither of these cases tells us anything
[21]         if not (one? t or t = nm1) then
[22]            for j in 1..k-1 repeat
[23]                t := mulmod(t, t, n)
[24]                one? t => return true
[25]                -- we have squared something not -1 and got 1
[26]                t = nm1 =>
[27]                    leave
[28]            not (t = nm1) => return true
[29]         false
```

Non-square-free numbers

If Rabin's algorithm is handed a number N with a repeated prime factor p^k, then the factor of p^{k-1} in $\hat{\phi}(N)$ will certainly be coprime to $N-1$. This means that we will return "N is definitely not prime" unless we use an x-value which is actually a perfect p^{k-1}-st power — an event with probability $1/p^{k-1}$. This probability is less than 0.25 except in the case $p=3$, $k=2$, when we can calculate explicitly that the probability of incorrectly saying "N is probably prime" is exactly 0.25 in the case $N=9$.

In the implementation given above, then test at line `[ 4]` ensures that N has no factors less than 317, and, *a fortiori*, no such repeated factors. Hence the probability that an x-value would be a perfect p-th power is at most $1/317$. This compares favourably with some of the probabilities that will be analysed later, and shows the practical utility of this preliminary test.

The Jaeschke number analysed

Let us analyse the number J more closely. To begin with, both J_1 and J_2 are prime. These numbers can be written as

$$
\begin{aligned}
J_1 &= 1 + 2^2 \cdot 3^2 \cdot 829 \cdot 4614533083 \\
J_2 &= 1 + 2^2 \cdot 3^3 \cdot 829 \cdot 4614533083 \\
J &= 1 + 2^5 \cdot 3^2 \cdot 5 \cdot 11 \cdot 59 \cdot 829 \cdot 34849 \cdot 456679 \cdot 4614533083
\end{aligned}
$$

J is not a Carmichael number, but it is "fairly close", since it is only the factor of 3^3, rather than 3^2, in J_2-1 which prevents it from being so. In addition, J is a product of two primes, of the form $(K+1)\cdot(rK+1)$ (with $r=3$) — a form observed by Pomerance *et al.* [1980] to account for nearly all pseudoprimes.

Why does Rabin's test (using the primes (1)) think that J is prime? To begin with, all the primes in the set (1) are actually perfect cubes modulo J_2, so their orders divide $(J_2-1)/3$, and hence $J-1$. Put another way, J is a pseudoprime(p) for all the p in (1): these 10 primes all cause the Fermat test to be satisfied. Assuming that $3|p-1$, 1/3 of non-zero congruence classes are perfect cubes modulo p.

For J to pass Rabin's test, we must also ensure that, for every p in (1), the 2-part of the order of p modulo J_1 is equal to the 2-part of the order of p modulo J_2. 3 and 31 are both quadratic residues modulo both J_1 and J_2, whilst the other primes are all non-residues. For the non-residues, the 2-part is maximal, viz. 2^2 modulo both these factors, so these eight primes all cause J to pass Rabin's test, as well as Fermat's. 3 and 31 are, in fact, not only quadratic residues, but also quartic residues for both J_1 and J_2, so their orders have 2-part 2^0, and hence also cause J to pass Rabin's test.

Since $J_2 \equiv J_1 \equiv 1 \pmod 4$, the quadratic character $(a|J_i) = (J_i|a)$, and so depends only on the value of $J_i \pmod a$ (in general, one might have to work modulo $4a$). $J_2 = 3J_1 - 2$, so the two are not independent, but we would expect 1/4 of congruence classes of $J_1 \pmod a$ to make a a non-residue for both J_1 and J_2. Another 1/4 would have a a quadratic residue for both, but it would then be necessary to investigate quartic properties, and so on. For a given a, asymptotically, about 1/3 of the values of J_1 will arrange that the quadratic, quartic, octic etc. characters of a modulo J_1 and J_2 are compatible with passing Rabin's tightening of the Fermat test.

What are the implications of this for an n-step Rabin algorithm, if our opponent, the person who is trying to find a composite N such that our use of Rabin's algorithm says "N is probably prime", chooses $N = M_1 \cdot M_2$, with M_1, M_2 prime and $M_2 - 1 = 3(M_1 - 1)$ (and hence $M_1 \equiv 1 \pmod 3$, otherwise $x = 3$ will fail Rabin's test)? Each prime p we use forces the condition that p should be a perfect cube modulo M_2 — satisfied about 1/3 of the time. In addition, the quadratic characters of p modulo M_1 and M_2 must be compatible — at best, with $M_1 - 1 \equiv 2 \pmod 4$, this happens 1/3 of the time on average. Hence each p we use imposes constraints satisfied about 1/9 of the time (assuming independence, which seems in practice to be the case). So we might expect to find a "rogue" number with M_1 about 9^n, and so N about 9^{2n}, which is 10^{19} if $n = 10$. However, we also have to insist that M_1 and M_2 are prime, which reduces our chance of finding a rogue pair quite considerably — roughly by 1/22 for each of M_1 and M_2, which would give us an estimated "time to find a rogue value" of $5 \cdot 10^{21}$. We can, in fact, be surprised that J is as large as it is — perhaps a smaller value exists.

Roots of -1

Here we look at observation (A) above — that a suspicious human being would observe more than two square roots of -1, and hence deduce that J was not prime, irrespective of the details of Rabin's algorithm. This is certainly true — how programmable, and how widely applicable, is it?

Adding it is easy: the following modifications need to be performed. A global (to `prime?` and `rabinProvesComposite`) variable `rootsMinus1` is added, whose type is a `Set` of `I`. This variable is used in the following ways.

After line `[10]`, we add

```
[10y]     rootsMinus1 := [] -- the empty set
```

After line `[22]` we add

```
[22a]               oldt := t
```

After line `[26]`, we add (`#` is the operator that counts the number of elements in a set)

```
[26a]                   rootsMinus1:=union(rootsMinus1,oldt)
[26b]                   # rootsMinus1 > 2 => return true
```

These changes certainly stop the algorithm from returning "N is probably prime" on the Jaeschke number, and do not otherwise alter the correctness of the algorithm, so might as well be incorporated. They only take effect when $k > 1$, since only then is the loop at [22] onwards executed.

If $k > 1$ then these changes certainly may be executed. But if all the prime factors p_i of N have small 2-part in $\phi(p_i)$, in particular if the 2-part of $\hat{\phi}(N) = 2^1$, then these changes will not take effect (but those proposed in the next section will). In general it is hard to analyse the precise contribution of these changes, other than to be certain that it is never negative.

The "maximal 2-part" test

Here we attempt to generalise observation (B) above. Let us suppose that N is still the composite number that we wish to prove is composite, and that $N = \prod_{i=1}^{n} p_i$ with the p_i distinct. Write $N = 1 + 2^k l$ with l odd, and $p_i = 1 + 2^{k_i} l_i$ with l_i odd. Clearly $k \geq \min_i k_i$. If N were prime, we would know that half the residue classes modulo N were quadratic non-residue, and hence we would expect half the x-values chosen to have 2-order k. Conversely, if all the k_i were equal to each other and to k, we would expect X to be a quadratic non-residue about half the time *with respect to each* p_i, and so about 1 in 2^n of the x-values will have maximal 2-rank.

One very simple variant on this test that can be imposed is to insist that, before deciding that "N is probably prime", we actually observe an element of 2-order k. If N actually were prime, we would have a chance of 1023/1024 of observing this before finishing the loop starting on line [11], so this test is extremely unlikely to slow down the performance of the system on primes. On non-primes, it may slow us down, but increases the chance of our giving the "correct" answer.

Adding it is easy: in fact our solution collects more information than is strictly necessary. The following modifications need to be performed. A global (to `prime?` and `rabinProvesComposite`) variable `count2Order` is added, whose type is a `Vector` of `NonNegativeInteger`s. This variable is used to count the number of elements of each 2-order: more precisely it is used in the following ways.

After line [10], we add

```
[10z]      count2Order := new(k,0) -- vector of k zeroes
```

After line 12, we insert the following lines

```
[12e]      currPrime:=smallPrimes(mn+10)
[12f]      while count2Order(k) = 0 repeat
[12g]            currPrime := nextPrime currPrime
[12h]            rabinProvesComposite(currPrime,n,nm1,q,k) => return false
```

After line [19] we insert

```
[19a]         if t=nm1 then count2Order(1):=count2Order(1)+1
```

After line [26] we insert

```
[26c]                     count2Order(j+1):=count2Order(j+1)+1
```

Again, this modification to the Rabin algorithm proves that the Jaeschke number is not prime.

How would one defeat these modifications?

It is all very well to propose new algorithms, and demonstrate that they are "better" than the old ones, but might they really have loop-holes just as large? The "maximal 2-part" requirement defeats a whole family of pseudoprimes — all those of the form $(K+1)\cdot(rK+1)$ with r odd, since then $N-1$ has a higher 2-part than $\hat{\phi}(N)$. This test is therefore useful in general, and defeats any straight-forward generalisation of the Jaeschke number to larger sets of x.

There are various possible constructions which these modifications do not defeat. We could make our pseudoprime N take the form $(K+1)\cdot(6K+1)$ with $K \equiv 2 \pmod 4$. Then the 2-part of $\hat{\phi}(N)$ would be 2^2, whereas that of $N-1$ would be 2^1 (and so the "roots of -1" enhancement would never operate). A value x would pass Rabin's test, with the "maximal 2-part" enhancement, if it were

(1) a cubic residue modulo $6K+1$;
(2) a quadratic residue modulo $6K+1$;
(3) a quartic non-residue modulo $6K+1$;
(4) a quadratic non-residue modulo $K+1$.

On average, one K-value in 24 will have these properties for a fixed x.

A value x would also pass Rabin's test, but would not contribute to the "maximal 2-part", if it were

(1') a cubic residue modulo $6K+1$;
(2') a quadratic residue modulo $6K+1$;
(3') a quartic residue modulo $6K+1$;
(4') a quadratic residue modulo $K+1$.

Again, on average, one K-value in 24 will have these properties for a fixed x.

We note, therefore, that we might expect 50% of x-values causing N to pass Rabin's test to have 2-part 2^1 and 50% to have 2-part 2^0: the same distribution as for a prime value of N (with $k=1$). If we use n different x-values, we might expect K to have to be of the order of 12^n, and N to be of the order of 144^n. In addition, both $K+1$ and $6K+1$ have to be prime. For $n=10$, the probability of this is about $1/25$, so we might expect to find such an N at around $2 \cdot 10^{24}$.

Leech's attack

Leech [1992] has suggested an attack of the form $N=(K+1)\cdot(2K+1)\cdot(3K+1)$. If the three factors are prime (which incidentally forces $K=2$, a case we can discard, or $K \equiv 0 \pmod 6$), then these numbers are certainly Carmichael*, and hence a good attack on the original version of Rabin's algorithm. Indeed, almost 25% of seed values will yield the result "N is probably prime".

Fortunately, we are saved by the "maximal 2-part" variant. Suppose $K=3\cdot n\cdot 2^m$ with n odd (and m at least 1). Then the maximal 2-part we can actually observe is 2^m, whereas

$$N-1 = 162\ n^3\ (2^m)^3 + 99\ n^2\ (2^m)^2 + 18\ n\ 2^m,$$

which is divisible by 2^{m+1}. Hence we will never observe an element of the maximal 2-part, and the loop at line `[12f]` will run until a counter-example to primality is found.

In fact, if $m=1$, $N-1$ is divisible by 8, and if $m>1$, $N-1$ is divisible by 2^{m+1}, which is at least 8. Hence the "roots of -1" test also acts, and reduces the probability of passing the modified Rabin well below 25%.

Other forms of attack are certainly possible, e.g. taking $N=(K+1)\cdot(3K+1)\cdot(5K+1)$. Here the "maximal 2-part" does not help us, since the 2-part of $N-1$ is equal to that of $\hat{\phi}(N)$. However these numbers are not generally Carmichael, only "nearly Carmichael", since 5 does not divide $N-1$. Hence we would need to insist that all our seed values were quintic residues modulo $5K+1$ as well as having the same 2-part modulo all the factors, and so on. These more complex families seem to create more problems for the inventor of counter-examples, so we can probably

* The first few numbers of this form are 1729, 294409, 56052361, 118901521, 172947529, 216821881, 228842209, 1299963601, 2301745249, 9624742921, 11346205609, 13079177569, 21515221081, 27278026129, 65700513721, 71171308081, 100264053529, 168003672409, 172018713961, 173032371289, 464052305161, 527519713969, 663805468801, 727993807201, 856666552249, 1042789205881 and 1201586232601.

say that taking one prime for every factor of 100 in N probably makes the systematic construction of counter-examples by this technique impossible.

However, if we also force $K \equiv 12 \pmod{30}$, Leech [1992] has pointed out that N is Carmichael. By construction, the factors are conguent to each other, and to their product, modulo 12, so the quadratic characters of 3 modulo the different factors are compatible. In fact we also need $K \equiv 0 \pmod 7$, since $K \equiv 1, 3, 5$ gives incompatible quadratic characters for 7 modulo the different factors, and $K \equiv 2, 4, 6$ gives non-prime factors. However, the three factors are congruent respectively to 3, 2 and 1 modulo 5, and so 5 will be a quadratic non-residue modulo the first two factors, but a residue modulo the last, hence ensuring that Rabin's algorithm with $x = 5$ always says "N is certainly composite".

The $(K + 1) \cdot (2K + 1)$ attack

This attack has been used recently by Arnault [1991] to defeat the set of x-values

$$\{2, 3, 5, 7, 11, 13, 17, 19, 23, 29\}. \tag{2}$$

The number

$$1195068768795265792518361315725116351898245581 =$$

$$48889032896862784894921 \cdot 24444516448431392447461$$

passes all these tests. In effect, the requirement is that x be a quadratic residue modulo $2K + 1$, and that the quadratic character of x modulo $K + 1$ should equate to the quartic character of x modulo $2K + 1$. These conditions are satisfied for approximately 25% of K-values. In addition, of course, $K+1$ and $2K+1$ must be prime. It would almost certainly be possible to construct a much smaller number than Arnault's, with the same properties — he fixed the congruences classes he was considering: for example he chose one class modulo 116, rather than examining all 29 satisfactory classes.

This form of attack is particularly worrying, since it is much easier to use than the other attacks in the previous sections. Indeed, one should probably consider $\log_4 N$ values x to test a number N if one is to defend against this attack. Fortunately, we have a simpler defence: we can check explicitly if the number we are given is of this form.

It is worth noting that Damgård & Landrock [1991] prove the following theorem, a slightly weaker version of which was earlier proved by McDonnell [1989].

Theorem. *If N is an odd composite number, such that N is not divisible by 3, and more than 1/8 of the x-values yield "N is probably prime" then one of the following holds:*

- *N is a Carmichael number with precisely three prime factors;*
- *$3N + 1$ is a perfect square;*
- *$8N + 1$ is a perfect square.*

$8(K + 1) \cdot (2K + 1) + 1 = (4K + 3)^2$, so this attack is a special case of the above theorem. There seems no reason not to test both the exceptional conditions in the Damgård-Landrock Theorem — such numbers are always composite, except for trivial cases ruled out by lines before `[ 5]`.

This requires the following modifications. The following lines are inserted after line `[12]` (but before those from the "maximal 2-part" modification).

```
[12a]       import IntegerRoots(I)
[12b]       q > 1 and perfectSquare?(3*n+1) => false
[12c]       ((n9:=n rem (9::I))=1 or n9 = -1) and perfectSquare?(8*n+1) => false
[12d]       -- Both previous tests from Damgard & Landrock
```

Note that we have saved on the average number of calls to `perfectSquare?` by the use of some elementary congruences. This is perhaps somewhat otiose, since in theory testing for being a perfect square takes time $O(\log^2 N)$, and our algorithm for primality testing is at least $O(\log^3 N)$.

Further Attacks

We were unable to construct a number which defeated our enhanced version of Rabin's algorithm, and in the earlier version of our paper we wrote as follows.

> "It should certainly be possible to do so [defeat the algorithm], if the set of x-values is fixed. In general, the number of primes used should be proportional to $\log N$, and we have made some suggestions as to what the constant of proportionality should be. A better constant of proportionality can be used if we test explicitly for numbers of the form $(K+1)\cdot(2K+1)$, probably via the Damgård-Landrock Theorem. This approach converts Rabin's algorithm from a $O(\log^3 N)$ test to a $O(\log^4 N)$, but we believe that a general-purpose system needs the additional security."

Our expectation that the fixed test with 10 bases would be defeated was shown to be justified by Arnault [1993], who suggested a $(K+1)\cdot(37K+1)\cdot(41K+1)$ attack. With K=1242260225201226, we get the number 2908193646321516347729985612962472286446235377 of 46 digits, which is first proved composite (among the odd prime values) by the x-value 43. Bleichenbacher [1993] used a $(K+1)\cdot(5K+1)\cdot(21K+1)$ attack with K=867416450123298078 to deduce the 56-digit number 68528663395046912244223605902738356719751082784386681071, which is first proved composite by the base 101.

Conclusions

It is certainly possible to draw more information from a fixed set of x-values than Rabin's original algorithm does, and we have explained two ways of doing this. However, it seems likely that any *fixed* set of bases can be led to produce a composite number for which they all report "probably prime". We therefore suggest that the number of bases used should grow with N, and suggest $\log_{100} N$ as the right number of bases to use (subject to a minimum of 10, except when relying on the exhaustive checks of Pomerance *et al.* [1980]. Pinch [1993] and Jaeschke [1993]).

It is perhaps worth noting that $\log_{100} N$ is consistent with the exhaustive data referred to above. The pseudoprimes from Pomerance *et al.* which need more than the bases 2, 3 and 5 are all larger than 100^3, and all are proved composite by the base 7. The first number to need five bases (2, 3, 5, 7 and 11) is 118670087467 from Pinch, and that is greater than 100^4*. The first number to need six bases is 21515221081, which is greater than 100^5, and the first to need seven bases is 3474749660383, which is greater than 100^6 — though not by much, and we see that picking a larger constant than 100 would not be suggsted by these exhaustive data.

It must be emphasised that we have not produced a guaranteed $O(\log^4 N)$ primality test: merely one that we do not believe we can break by the technology we know. It would be tempting to conjecture that, with an appropriate constant of proportionality, this test is guaranteed never to return "N is probably prime" when in fact it is composite. The first result in this area is a statement by Ankeny [1952] that, assuming the Extended Riemann Hypothesis (ERH), the number of tests required is $O(\log^2 N)$. Lenstra [1980] (see also Koblitz [1987]) shows, under the same assumptions, that $70\log^2 N$ values suffice, which would give a $O(\log^5 N)$ primality test. Under ERH, Bach [1985; 1992] has reduced the constant from 70 to 2, and in fact $1 + o(1)$. Bach & Huelsbergen [1993] suggest that the correct number of test required is at most $\log_2 N \log\log N$, which would give a $O(\log^4 N \log\log N)$ primality test.

We should note that Baillie & Wagstaff [1980] propose that, and Arnault [1993] gives further reasons why, a composite test, testing for both Fermat pseudoprimes and Lucas pseudoprimes, might be more efficient. No counter-examples are known to this test, but there is currently less

* 354864744877, from the same list, also needs five bases for the standard Rabin test, but only four with our "roots of -1" modification.

theoretical background, even as nebulous as that presented here, to suggest where counter-examples might be found, or how many tests one should do for a number of a given size.

Timings. Consider the prime $(2^{3539}+1)/3)$, of 1065 decimal digits. Morain [1989] proved its primality, using a distributed implementation of the elliptic curve primality test, requiring 319 days of SUN 3 time. On an IBM RS/6000 model 530H, the original implementation took 1625 seconds, and our modified $O(\log^4 N)$ algorithm took 86783 seconds (marginally over a day), as we might expect since it will use 530 seeds rather than 10.

Acknowledgements. The author is grateful to Barry Trager for passing on Herr Jaeschke's comment, to Patrizia Gianni for her hospitality while the original investigations were carried out, to Geoff Smith for many discussions, to Guy Robin for pointing out Arnault's work, to the late John Leech for pointing out his attack, to the University of Limoges where the first draft of this paper was written, to the Cambridge Arithmetic Seminar for many useful comments, in particular Richard Pinch who pointed out an error in previous drafts, and to Eric Bach for pointing out his work. The UK SERC provided funding under grant GR/H/11587. Above all, thanks are due to Herr Jaeschke for his painstaking construction of J.

References

[Alford *et al.*, 1992] Alford,W.R., Granville,A. & Pomerance,C., There are Infinitely Many Carmichael Numbers. Preprint, 27 December 1992.

[Ankeny, 1952] Ankeny,N.C., The Least Quadratic Non Residue. Ann. Math. (2) **55** (1952) pp. 65–72.

[Arnault, 1991] Arnault,F., Le Test de Primalité de Rabin-Miller: Un nombre composé qui le "passe". Prepublication 61, Université de Poitiers Département de Mathématiques, November 1991.

[Arnault, 1992] Arnault,F., Rabin-Miller Primality Test: Composite Numbers which Pass it. Submitted Sept. 1992 to Math. Comp.

[Arnault, 1993] Arnault,F., Carmichaels fortement pseudo-premiers, pseudo-premiers de Lucas. Prepublication 73, Université de Poitiers Département de Mathématiques, January 1993.

[Bach, 1985] Bach,E., Analytic Methods in the Analysis and Design of Number-theoretic Algorithms. MIT Press, Cambridge, Mass., 1985.

[Bach, 1992] Bach,E., Explicit Bounds for Primality Testing and Related Problems. Math. Comp. **55** (1992) pp. 355–380.

[Bach & Huelsbergen, 1993] Bach,E., & Huelsbergen,L., Statistical Evidence for Small Generating Sets. To appear in Math. Comp.

[Baillie & Wagstaff, 1980] Baillie,R. & Wagstaff,S.S., Jr., Lucas Pseudoprimes. Math. Comp. **35** (1980) pp. 1391–1417.

[Bleichenbacher, 1993] Bleichenbacher,D., Pseudoprimes too strong for Maple. News article `1993Apr27.141249.29080@neptune.inf.ethz.ch`.

[Damgård & Landrock, 1991] Damgård,I. & Landrock,P., Improved Bounds for the Rabin Primality Test. To appear in Proc. 3rd IMA conference on Coding and Cryptography, ed. M. Ganley, OUP.

[Davenport & Smith, 1987] Davenport,J.H. & Smith,G.C., Rabin's Primality Testing Algorithm — a Group Theory View. University of Bath Technical Report 87–04.

[Jaeschke, 1991] Jaeschke,G., Private Communication. April 1991.

[Jaeschke, 1993] Jaeschke,G., On Strong Pseudoprimes to Several Bases. To appear in October 1993 Math. Comp.

[Koblitz, 1987] Koblitz,N., A Course in Number Theory and Cryptography. Springer-Verlag, 1987.

[Leech, 1992] Leech,J., Private Communication. 18 March and 7 April, 1992.

[Lenstra, 1981] Lenstra,H.W.,Jr., Primality Testing Algorithms (after Adleman, Rumely and Pomerance). Séminaire Bourbaki 1980/81 (Springer Lecture Notes in Mathematics 901, Springer-Verlag, Berlin-Heidelberg, 1981) pp. 243–257.

[McDonnell, 1989] McDonnell,F.J., Rabin's Algorithm and the Proportion of Liars for Various Families of Numbers. Technical Report, University of Warwick Mathematics Institute, March 1989.

[Morain, 1989] Morain,F., Distributed primality proving and the primality of $(2^{3539}+1)/3$. INRIA Research Report 1152, Dec. 1989.

[Pinch, 1993] Pinch,R.G.E., Primality Testing. Lecture Notes, University of Cambridge, Lent 1993.

[Pomerance *et al.*, 1980] Pomerance,C., Selfridge,J.L. & Wagstaff,S.S.,Jr, The pseudoprimes up to $25 \cdot 10^9$. *Math. Comp.* **35** (1980) pp. 1003–1026.

[Rabin, 1980] Rabin,M.O., Probabilistic Algorithm for Testing Primality. J. Number Theory **12** (1980) pp. 128–138.

The Pomerance *et al.* [1980], Pinch [1993] and Jaeschke [1993] Modifications

The following global declarations are made.

```
[ 0a]  PomeranceList:= [25326001::I, 161304001::I, 960946321::I, 1157839381::I,
[ 0b]                   -- 3215031751::I, -- has a factor of 151
[ 0c]                   3697278427::I, 5764643587::I, 6770862367::I,
[ 0d]                    14386156093::I, 15579919981::I, 18459366157::I,
[ 0e]                     19887974881::I, 21276028621::I ]::(List I)
[ 0f]  PomeranceLimit:=27716349961::I  -- replaces (25*10**9) due to Pinch
[ 0g]  PinchList:= [3215031751::I, 118670087467::I, 128282461501::I,
[ 0h]     354864744877::I, 546348519181::I, 602248359169::I, 669094855201::I ]
[ 0i]  PinchLimit:= (10**12)::I
[ 0j]  PinchList2:= [2152302898747::I, 3474749660383::I]
[ 0k]  PinchLimit2:= (10**13)::I
[ 0l]  JaeschkeLimit:=341550071728321::I
```

The following lines are placed after line 10.

```
[10a]      n < JaeschkeLimit =>
[10b]          rabinProvesCompositeSmall(2::I,n,nm1,q,k) => return false
[10c]          rabinProvesCompositeSmall(3::I,n,nm1,q,k) => return false
[10d]
[10e]          n < PomeranceLimit =>
[10f]              rabinProvesCompositeSmall(5::I,n,nm1,q,k) => return false
[10g]              member?(n,PomeranceList) => return false
[10h]              true
[10i]
[10j]          rabinProvesCompositeSmall(7::I,n,nm1,q,k) => return false
[10k]          n < PinchLimit =>
[10l]              rabinProvesCompositeSmall(10::I,n,nm1,q,k) => return false
[10m]              member?(n,PinchList) => return false
[10n]              true
[10o]
[10p]          rabinProvesCompositeSmall(5::I,n,nm1,q,k) => return false
[10q]          rabinProvesCompositeSmall(11::I,n,nm1,q,k) => return false
[10r]          n < PinchLimit2 =>
[10s]              member?(n,PinchList2) => return false
[10t]              true
[10u]
[10v]          rabinProvesCompositeSmall(13::I,n,nm1,q,k) => return false
```

```
[10w]          rabinProvesCompositeSmall(17::I,n,nm1,q,k) => return false
[10x]          true
```

Here, `rabinProvesCompositeSmall` is a variant of `rabinProvesComposite` without the "Roots of -1" or the "Maximal 2-part" modifications. This, and the careful ordering of these lines compared to `[10y]` and `[10z]` suggested earlier means that recursive calls of `prime?` do not disturb the data structures set up for those modifications *unless* we recurse on primes greater than the Jaeschke limit, which would happen if we wished to test numbers with more than $6 \cdot 10^{14}$ decimal digits — a contingency we can regard as remote.

$O(log^4 N)$ Modifications

The "Maximal 2-part" modifications are replaced by the following.

```
[12e]     currPrime:=smallPrimes(mn+10)
[12f]     probablySafe:=tenPowerTwenty
[12g]     while count2Order(k) = 0 or n > probablySafe repeat
[12h]         currPrime := nextPrime currPrime
[12i]         probablySafe:=probablySafe*(100::I)
[12j]         rabinProvesComposite(currPrime,n,nm1,q,k) => return false
```