An Extended Quadratic Frobenius Primality Test with Average Case Error Estimates *

Ivan Bjerre Damgård Gudmund Skovbjerg Frandsen

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

DRAFT - November 16, 2001

Abstract

We present an Extended Quadratic Frobenius Primality Test (EQFT), which is related to the Miller-Rabin test and the Quadratic Frobenius test (QFT) by Grantham. EQFT is well-suited for generating large, random prime numbers since on a random input number, it takes time about equivalent to 2 Miller-Rabin tests, but has much smaller error probability. EQFT extends QFT by verifying additional algebraic properties related to the existence of elements of order 3 and 4. We obtain a simple closed expression that upper bounds the probability of acceptance for any input number. This in turn allows us to give strong bounds on the average-case behaviour of the test: consider the algorithm that repeatedly chooses random odd k bit numbers, subjects them to t iterations of our test and outputs the first one found that passes all tests. We obtain numeric upper bounds for the error probability of this algorithm as well as a general closed expression bounding the error. For instance, it is at most 2^{-143} for $k = 500, t = 2$. Compared to earlier similar results for the Miller-Rabin test, the results indicates that our test in the average case has the effect of 9 Miller-Rabin tests, while only taking time equivalent to about 2 such tests. We also give bounds for the error in case a prime is sought by incremental search from a random starting point. While EQFT is slower than the average case on a small set of inputs, we present a variant that is always fast, i.e. takes time about 2 Miller-Rabin tests. The variant has slightly larger worst case error probability than EQFT, but still improves on previous proposed tests.

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

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

1 Introduction

Efficient methods for primality testing are extremely important, in theory as well as in practice. In public-key cryptography, for instance, efficient methods for generating large, random primes are indispensable tools. Although tests that always return correct results do exist, tests that accept composite numbers with bounded probability continue to be much more efficient. This paper presents and analyses one such test.

Virtually all known probabilistic tests are built on the same basic principle: from the input number n, one defines an Abelian group in such a way that *if* n is prime, then the structure of the group (e.g., its order and the number of cyclic components) is known. But if n is in fact composite, the actual group structure is different. We then do a number of computations in the group, to see if the structure we expect if n is prime is actually present. If not, we know for sure that n is composite and reject, otherwise we accept, but n may still be composite because we may have failed to detect that the group had a "bad" structure.

The well-known Miller-Rabin test uses the group Z_n^* in exactly this way. A natural alternative is to try a quadratic extension of Z_n, that is, we look at the ring $Z_n[x]/(f(x))$ where $f(x)$ is a degree 2 polynomial chosen such that it is guaranteed to be irreducible if n is prime. In that case the ring is isomorphic to the finite field with n^2 elements, $GF(n^2)$. This approach was used successfully by Grantham[6], who proposed the Quadratic Frobenius Test (QFT), and showed that it accepts a composite with probability at most 1/7710, i.e. a better bound than may be achieved using 6 independent Miller-Rabin tests, while asymptotically taking time approximately equivalent to only 3 such tests. Müller proposes a different approach based on computation of square roots, the MQFT [7] which takes the same time as QFT and has error probability essentially 1/131040[1]. Just as for the Miller-Rabin test, however, it seems that most composites would be accepted with probability much smaller than the worst-case numbers. A precise result quantifying this intuition would allow us to analyse the average case behaviour of the test, i.e., when it is used to test numbers chosen at random, say, from some interval. Such an analysis has been done by Damgård, Landrock and Pomerance for the Miller-Rabin test, but no corresponding result for QFT or MQFT is known.

In this paper, we propose a new test that can be seen as an extension of QFT. We call this the Extended Quadratic Frobenius test (EQFT). Under the ERH, our test takes expected time 2/3 of the time needed for QFT/MQFT (the ERH is only used to bound the run time and does not affect the error probability). The error probability is analysed both in average case and worst case scenarios:

For the average case analysis: consider an algorithm that repeatedly chooses random odd k-bit numbers, subject each number to t iterations of our test, and outputs the first number found that passes all t tests. Let $q_{k,t}$ be the probability that a composite is output. We derive numeric upper bounds for $q_{k,t}$, e.g., we show $q_{500,2} \leq 2^{-143}$, and also show a general upper bound, namely for $2 \leq t \leq k-1$, $q_{k,t}$ is $O(k^{3/2}2^{(\sigma_t+1)t}t^{-1/2}4^{-\sqrt{2\sigma_t tk}})$ with an easily computable big-O constant, where $\sigma_t = \log_2 24 - 2/t$.

[1]The test and analysis results are a bit different, depending on whether the input is 3 or 1 modulo 4, see [7] for details

A similar analysis has been carried out by Damgård, Landrock and Pomerance for the Miller-Rabin test. One result of that paper was that the corresponding error probability $p_{k,t}$ for Miller-Rabin is $O(k^{3/2}2^t t^{-1/2} 4^{-\sqrt{tk}})$ for $2 \leq t \leq k/9$. This indicates that for $t \geq 2$, our test in the average case roughly speaking has the effect of between 7 and 9 Miller-Rabin tests, while only taking time equivalent to 2 such tests.

We also analyze the error probability when a random k-bit prime is instead generated using incremental search from a random starting point, still using (up to) t iterations of our test to distinguish primes from composites.

Concerning worst case analysis, it can be shown that t iterations of EQFT err with probability at most $256/331776^t$ except for an explicit finite set of numbers. However, we have to consider, in addition to the worst case error probability, the worst case running time. Unfortunately, EQFT is up to 4 times slower on worst case inputs than in the average case, namely on numbers n where very large powers of 2 and 3 divide $n^2 - 1$. We therefore present a variant of EQFT that always takes time equivalent to about 2 Miller-Rabin tests (still assuming ERH) but has a worst case error slightly weaker than EQFT. For this variant, we show that if a composite n has no prime factors less than 2^7, or if $n \geq 2^{67.5}$, and if we do t iterations of the test, we err with probability at most $16/4096^t$. For comparison with Granthams test, assume that we are willing to spend the same fixed amount of time testing an input number. Then our test gives asymptotically a better bound on the error probability: using time approximately corresponding to $6t$ Miller-Rabin test, we get error probability $1/7710^{2t} \approx 1/19.8^{6t}$ using QFT, $1/131040^{2t} \approx 1/50.8^{6t}$ using MQFT, and $16/4096^{3t} = 16/64^{6t}$ using the modified version of EQFT. Note that since we can recognize in negligible time numbers on which EQFT will take unusually long time, we can choose intially which version of EQFT to run, and in this way obtain error probability about $256/331776^{3t} = 256/576^{6t}$ for most input numbers, at no significant cost in running time.

2 Extended Quadratic Frobenius Test (EQFT)

We start by giving an intuitive explanation of the basic ideas behind EQFT. The easiest way to understand the test is to think of it as a way to generalize the Miller-Rabin test so that instead of Z_n^*, we use the quadratic extension $Z_n[x]/(f(x))$.

The basic strategy is unchanged: choose a random element z in the group and test if z has some number of properties. In the Miller-Rabin case, part of what we do is the Fermat-test, which is based on the fact that Z_n^* has order $n - 1$ if n is prime, so we verify that $z^{n-1} = 1$. In the quadratic extension we expect to have a group of order n^2-1, so one might expect that we should verify that $z^{n^2-1} = 1$. However, if n is prime $Z_n[x]/(f(x))$ is not just a group, but an extension field and so offers additional structure on top of the multiplicative group. In particular, it has three nontrivial maps, the *Frobenius automorphism:* $z \mapsto z^n$; the *norm:* $z \mapsto N(z)$ (a multiplicative homomorphism mapping $GF(n^2)$ to the subfield $GF(n)$); and *conjugation:* $z \rightarrow \overline{z}$ (a standard notion that is defined below). It turns out that we can verify instead that $z^n = \overline{z}$. For any invertible z, this implies $z^{n^2-1} = 1$ and is faster to check. On top of this, we can use an additional idea, namely instead of choosing z uniformly, we only choose values such that $N(z)$ has Jacobi symbol 1. In other words, we make sure that z "looks like a square" in the sense that z is guaranteed to be a square if n is a prime. We can therefore expect such a z to have order a factor 2 smaller than otherwise, and this turns out to improve the error probability by a factor of $2^{1-\omega}$, where ω is the number of distinct prime factors in n.

The second main part of the Miller-Rabin test is the part where we look for non-trivial square roots of 1. This is based on the fact that if the Fermat test was passed, we have a random z for which $z^{n-1} = 1$. Since $n-1$ can be assumed to be even, we can use this to construct a square root of 1, i.e., an element of order 1 or 2, chosen among all such elements in the group. If n is prime, there are only 2 such elements, namely $1, -1$, whereas in general there are 2^ω of them because Z_n^* is the direct product of ω cyclic subgroups. The probability of running into ± 1 if we choose uniformly among the elements of order 1 or 2 is $2^{1-\omega}$, and this is the reason why the error probability of the Miller Rabin test is at most the probability of passing the Fermat test times $2^{1-\omega}$ [2].

With the quadratic extension, we have a group that has order n^2-1 if n is prime. Since we constrained our choice of z to a subgroup of index 2 ($N(z)$ must have Jacobi symbol 1), the group we expect to be working in is cyclic of order $(n^2-1)/2$ if n is prime. If we make sure that n's divisible by 2 or 3 are excluded, it is always the case that $2^2 \cdot 3$ divides $(n^2-1)/2$. So if n is prime there are exactly 4 elements of order 1, 2 or 4, namely $1, -1, \xi_4, -\xi_4$ where ξ_4 has order 4. And there are exactly 3 elements of order 1 or 3: $1, \xi_3, \xi_3^{-1}$. Now, assuming we have found a random z with $z^{(n^2-1)/2} = 1$, then we can use z to produce an element $R_4(z)$ of order 1, 2 or 4 and $R_3(z)$ of order 1 or 3, just like we produced square roots of 1 in the Miller-Rabin test. Assume for the moment that we are given elements ξ_4, ξ_3 of order 4 and 3. Then, if $R_4(z)$ is not one of $1, -1, \xi_4, -\xi_4$ or $R_3(z)$ is not one of $1, \xi_3, \xi_3^{-1}$, then n is composite. The quadratic extension typically has 4^ω, resp. 3^ω elements of order $1, 2, 4$, resp. $1, 3$ [3]. So a random choice will reveal that n is composite with probability $(3 \cdot 4)^{1-\omega}$. Together with the factor of $2^{1-\omega}$ that we gained by constraining the choice of z, this gives a factor of $24^{1-\omega}$ on the error probability. This seems to be what we can naturally expect: for Miller-Rabin, we used that $n-1$ is always divisible by 2, and gained a factor of $2^{1-\omega}$. Here, we have used that n^2-1 is always divisible by 24, and this is the maximal divisor that can always be guaranteed.

However, we did not yet address the problem that we do not know the elements ξ_4, ξ_3 a priori. However, as we have already seen, if z passes the Fermat-like part of the test, then we have a chance of producing elements of order 3 and 4 from z. So if we iterate the test several times using independent choices of z but the same quadratic extension, as soon as an iteration finds an element of order 3 or 4, this can be used as ξ_4 or ξ_3 by subsequent iterations. A detailed analysis shows that, although initial iterations may be weaker than $24^{1-\omega}$, the overall probability is almost as good as if we had known ξ_4, ξ_3 from the beginning: we loose a factor of at most $4^{\omega-1}$, for any number of iterations. To show this result, we exploit that some partial testing of $R_4(z), R_3(z)$ is possible even if we do not know suitable elements ξ_3, ξ_4. For instance, if we see

[2] For some n's the distribution will be biased towards non-trivial square roots of 1, this only makes the test stronger

[3] There may be less than that, but then the Fermat-like part of the test is much stronger than otherwise, so we only have to consider the maximal case.

an element of order 2, different from ± 1, it is already clear that n is composite. This is detailed below.

To facilitate comparison, we include some comments on the similarities and difference between EQFT and Grantham's QFT. In QFT the quadratic extension, that is, the polynomial $f(x)$, is randomly chosen, whereas the element corresponding to our z is chosen deterministically, given $f(x)$. Other than that, the Fermat part of QFT is transplanted almost directly to EQFT. For the test for roots of 1, QFT does something directly corresponding to the square root of 1 test from Miller-Rabin, but does nothing relating to elements of order 3 or 4. In fact, our idea of using elements produced in one iteration of the test in other executions cannot be directly applied to QFT because $f(x)$ changes between iterations. As for the running time, since our error analysis works for any (i.e. a worst case) quadratic extension, we can pick one that has a particularly fast implementation of arithmetic, and this is the basis for the earlier mentioned difference in running time between EQFT and QFT. As for the error analysis, using a fixed polynomial but a random choice in the group seems to simplify the analysis for EQFT, in particular concrete expressions for the error probability follow directly from knowledge of the group structure of $Z_n[x]/(f(x))$.

A final comment relates to the comparison in running times between Miller-Rabin, Grantham's and our test. We stated above that Granthams test (our test) takes time approximately equivalent to 3 (2) Miller-Rabin tests. What we mean by this more precisely is that the running time of Miller-Rabin, resp. Grantham's, resp. our test is $\log n + o(\log n)$ resp. $3 \log n + o(\log n)$ resp. $2 \log n + o(\log n))$ multiplications in $\mathbf{Z}_n$, this is also consistent with the way running times have been stated earlier in the literature. However, taking a closer look, we find that the running time of Miller-Rabin is actually $\log n$ *squarings* $+o(\log n)$ multiplications in $\mathbf{Z}_n$, while the $3 \log n$ $(2 \log n)$ multiplications mentioned for the other tests are really a mix of squarings and multiplications. So for an accurate comparison we should consider how the times for modular multiplications and squarings compare. In turns out that on a standard, say, 32 bit architecture, a modular multiplication takes time about 1.25 times that of a modular squaring if the numbers involved are very large. However, if we use the fastest known modular multiplication method (which is Montgomery's in this case, where n stays constant over many multiplications), the factor is smaller for numbers in the range of practical interest. Concrete measurements using highly optimized C code shows that it is between 1 and 1.08 for numbers of length 500-1000 bits. This is due to the fact that optimizing squarings by avoiding computation of some partial products requires additional bookkeeping that eats up the savings unless the numbers contain more than 40-50 words. Finally, when using dedicated hardware the factor is exactly 1 in most cases. So we conclude that the comparisons we stated are quite accurate also for practical purposes.

2.1 The ring $R(n, c)$ and the extended quadratic Frobenius test

Definition 1 *Let n be an odd integer and let c be a unit modulo n.*
Let $R(n, c)$ denote the ring $\mathbf{Z}[x]/(n, x^2 - c)$.

More concretely, an element $z \in R(n, c)$ can be thought of as a degree 1 polynomial $z = ax + b$, where $a, b \in \mathbf{Z}_n$, and arithmetic on polynomials is modulo $x^2 - c$ where coefficients are computed on modulo n.

Let p be an odd prime. If c is not a square modulo p, i.e. $(c/p) = -1$, then the polynomial $x^2 - c$ is irreducible modulo p and $R(p, c)$ is isomorphic to $GF(p^2)$.

Definition 2 *Define the following multiplicative homomorphisms on $R(n, c)$ (assume $z = ax + b$):*

$$\bar{\cdot} : \quad R(n,c) \mapsto R(n,c), \quad \overline{z} = -ax + b \tag{1}$$

$$N(\cdot) : \quad R(n,c) \mapsto \mathbf{Z}_n, \quad N(z) = \overline{z} \cdot z = b^2 - ca^2 \tag{2}$$

and define the map $(\cdot/\cdot) : \mathbf{Z} \times \mathbf{Z} \mapsto \{-1, 0, 1\}$ *to be the Jacobi symbol.*

The maps $\bar{\cdot}$ and $N(\cdot)$ are both multiplicative homomorphisms whether n is composite or n is a prime. The primality test will be based on some additional properties that are satisfied when p is a prime and $(c/p) = -1$, in which case $R(p, c) \simeq GF(p^2)$:

Frobenius property / generalised Fermat property: Conjugation, $z \mapsto \overline{z}$, is a field automorphism on $GF(p^2)$. In characteristic p, the Frobenius map that raises to the p'th power is also an automorphism, using this it follows easily that

$$\overline{z} \quad = \quad z^p \tag{3}$$

Quadratic residue property / generalised Solovay-Strassen property: The norm, $z \mapsto N(z)$, is a surjective multiplicative homomorphism from $GF(p^2)$ to the subfield $GF(p)$. As such the norm maps squares to squares and non-squares to non-squares, it follows from the definition of the norm and (3) that

$$z^{(p^2-1)/2} \quad = \quad N(z)^{(p-1)/2} \quad = \quad (N(z)/p) \tag{4}$$

4'th-root-of-1-test / generalised Miller-Rabin property: Since $GF(p^2)$ is a field there is only four possible 4th roots of 1 namely 1, -1 and ξ_4, $-\xi_4$, the two roots of the cyclotomic polynomial $\Phi_4(x) = x^2 + 1$. In particular, this implies for $p^2 - 1 = 2^u 3^v q$ where $(q, 6) = 1$ that if $z \in GF(p^2) \setminus \{0\}$ is a square then

$$z^{3^v q} = \pm 1, \quad \text{or} \quad z^{2^i 3^v q} = \pm \xi_4 \quad \text{for some } i = 0, \ldots, u-3 \tag{5}$$

3'rd-root-of-1-test: Since $GF(p^2)$ is a field there is only three possible 3rd roots of 1 namely 1 and ξ_3, ξ_3^{-1}, the two roots of the cyclotomic polynomial $\Phi_3(x) = x^2 + x + 1$. In particular, this implies for $p^2 - 1 = 2^u 3^v q$ where $(q, 6) = 1$ that if $z \in GF(p^2) \setminus \{0\}$ then

$$z^{2^u q} \; = \; 1, \quad \text{or} \quad z^{2^u 3^i q} = \xi_3^{\pm 1} \quad \text{for some } i = 0, \ldots, v-1 \tag{6}$$

The actual test will have two parts. In the first part, a specific quadratic extension is chosen, i.e. $R(n, c)$ for an explicit c. In the second part, the above properties of $R(n, c)$ is tested for a random choice of z. When the EQFT is run several times on the same n, only the second part is executed multiple times. The second part receives two extra inputs, a 3rd and a 4th root of 1. On the first execution of the second part these are both 1. During later executions of the second part some nontrivial roots are possibly constructed. If so they are transfered to all subsequent executions of the second part. Figure 1 illustrates 4 consecutive tests, where a primitive 3rd root, ξ_3, is found immediately and a primitive 4th root, ξ_4, is found later.

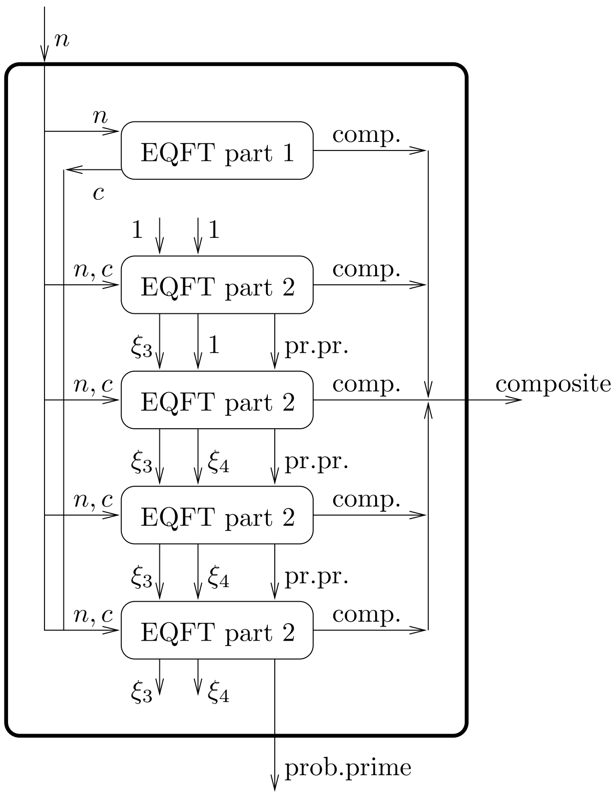

Figure 1: flowchart for 4 iterations of EQFT over a single n

Algorithm 3 *Extended Quadratic Frobenius Test (EQFT).*
First part (construct quadratic extension):
  input: *an odd number* $n \geq 5$.
  output: *"composite" or* c *satisfying* $(c/n) = -1$.

1. `if` n *is divisible by a prime less than 13* `return` *"composite"*

2. `if` n *is a perfect square* `return` *"composite"*

3. *choose a small* c *with* $(c/n) = -1$; `return` c

Second part (make actual test):
  input: $n \geq 5$ *not divisible by* 2 *or* 3.
   c *satisfying* $(c/n) = -1$
   $r_3 \in \{1\} \cup \{\xi \in R(n,c) \mid \Phi_3(\xi) = 0\}$
   $r_4 \in \{1,-1\} \cup \{\xi \in R(n,c) \mid \Phi_4(\xi) = 0\}$
  output: *"composite", or "probable prime" and*
   $s_3 \in \{1\} \cup \{\xi \in R(n,c) \mid \Phi_3(\xi) = 0\}$
   $s_4 \in \{1,-1\} \cup \{\xi \in R(n,c) \mid \Phi_4(\xi) = 0\}$
Let $n^2 - 1 = 2^u 3^v q$ *for* $(q,6) = 1$.

4. *select random* $z \in R(n,c)^*$ *with* $(N(z)/n) = 1$

5. `if` $\overline{z} \neq z^n$ *or* $z^{(n^2-1)/2} \neq 1$ `return` *"composite"*

6. `if` $z^{3^v q} \neq 1$ *and* $z^{2^i 3^v q} \neq -1$ *for all* $i = 0, \dots, u-2$ `return` *"composite"*

7. `if` *we found* $i_0 \geq 1$ *with* $z^{2^{i_0} 3^v q} = -1$ *(there can be at most one such value) then let* $R_4(z) = z^{2^{i_0-1} 3^v q}$. *Else let* $R_4(z) = z^{3^v q}$ $(= \pm 1)$;
   `if` $(r_4 \neq \pm 1$ *and* $R_4(z) \notin \{\pm 1, \pm r_4\})$ `return` *"composite"*

8. `if` $z^{2^u q} \neq 1$ *and* $\Phi_3(z^{2^u 3^i q}) \neq 0$ *for all* $i = 0, \dots, v-1$ `return` *"composite"*

9. `if` *we found* $i_0 \geq 0$ *with* $\Phi_3(z^{2^u 3^{i_0} q}) = 0$ *(there can be at most one such value) then let* $R_3(z) = z^{2^u 3^{i_0} q}$ *else let* $R_3(z) = 1$;
   `if` $(r_3 \neq 1$ *and* $R_3(z) \notin \{1, r_3^{\pm 1}\})$ `return` *"composite"*

10. `if` $r_3 = 1$ *and* $R_3(z) \neq 1$ *then let* $s_3 = R_3(z)$ *else let* $s_3 = r_3$;
    `if` $r_4 = \pm 1$ *and* $R_4(z) \neq \pm 1$ *then let* $s_4 = R_4(z)$ *else let* $s_4 = r_4$;
    `return` *"probable prime",* s_3, s_4

Remark. Line 1 ensures that $24 \mid n^2 - 1$.

Line 2 of the algorithm is necessary, since no c with $(c/n) = -1$ exists when n is a perfect square.

Line 3 of the algorithm ensures that $R(n,c) \simeq GF(n^2)$ when n is a prime. Lemma 5 defines more precisely what "small" means.

Line 4 makes sure that z is a square, when n is a prime.

Line 5 checks equations (3) and (4), the latter in accordance with the condition enforced in line 4.

Line 6 checks equation (5) to the extent possible without having knowledge of ξ_4, a primitive 4th root of 1.

Line 7f continues the check of equation (5) by using any ξ_4 given on the input.

Line 8 checks equation (6) to the extent possible without having knowledge of ξ_3, a primitive 3rd root of 1.

Line 9f continues the check of equation (6) by using any ξ_3 given on the input.

2.2 Implementation of the test

High powers of elements in $R(n,c)$ may be computed efficiently when c is (numerically) small. Represent $z \in R(n,c)$ in the natural way by $((A_z, B_z) \in \mathbf{Z}_n \times \mathbf{Z}_n$, i.e. $z = A_z x + B_z$.

Lemma 4 *Let $z, w \in R(n,c)$:*

1. *$z \cdot w$ may be computed from z and w using 3 multiplications and $O(\log c)$ additions in $\mathbf{Z}_n$*
2. *z^2 may be computed from z using 2 multiplications and $O(\log c)$ additions in $\mathbf{Z}_n$*

Proof. For 1, we use the equations

$$\begin{aligned} A_{zw} &= m_1 + m_2 \\ B_{zw} &= (cA_z + B_z)(A_w + B_w) - (cm_1 + m_2) \end{aligned}$$

with

$$\begin{aligned} m_1 &= A_z B_w \\ m_2 &= B_z A_w \end{aligned}$$

For 2, we need only observe that in the proof of 1, $z = w$ implies that $m_1 = m_2$. ■

We also need to argue that it is easy to find a small c with $(c/n) = -1$. One may note that if $n = 3 \bmod 4$, then $c = -1$ can always be used, and if $n = 5 \bmod 8$, then $c = 2$ will work. In general, we have the following:

Lemma 5 *Let n be an odd composite number that is not a perfect square. Let $\pi_-(x,n)$ denote the number of primes $p \leq x$ such that $(p/n) = -1$, and, as usual, let $\pi(x)$ denote the total number of primes $p \leq x$. Assuming the Extended Riemann Hypothesis (ERH), there exists a constant C (independent of n) such that*

$$\frac{\pi_-(x,n)}{\pi(x)} > \frac{1}{3} \quad \textit{for all } x \geq C(\log n \log\log n)^2$$

Proof. $\pi_-(x,n)$ counts the number of primes outside the group $G = \{x \in \mathbf{Z}_n^* \mid (x/n) = 1\}$. When n is not a perfect square, then G has index 2 in $\mathbf{Z}_n^*$, and by [1, th.8.4.6], the ERH implies that

$$\pi_-(x,n) = \frac{1}{2}\mathrm{li}(x) + O(\sqrt{x}(\log x + \log n)) \tag{7}$$

similarly, by [1, th.8.3.3], the Riemann Hypothesis implies that

$$\pi(x) = \mathrm{li}(x) + O(\sqrt{x}\log x) \tag{8}$$

where $\mathrm{li}(x) = \int_2^x dt/\ln t$ satisfies that

$$\mathrm{li}(x) = \Theta(x/\log x) \tag{9}$$

In addition the constants implied by the $O(\cdot)$-notation are all universal and therefore one may readily verify that for any $\epsilon > 0$ there is a universal constant C_ϵ such that

$$\frac{\pi_-(x,n)}{\pi(x)} > \frac{1}{2} - \epsilon \quad \text{for all } x \geq C_\epsilon(\log n \log\log n)^2$$

■

Theorem 6 *Let n be a number that is not divisible by 2 or 3, and let $u \geq 3$ and $v \geq 1$ be maximal such that $n^2 - 1 = 2^u 3^v q$. There is an implementation of algorithm 3 that on input n takes expected time equivalent to $2\log n + O(u+v) + o(\log n)$ multiplications in $\mathbf{Z}_n$, when assuming the ERH.*

Remark. We can only prove a bound on the expected time, due to the random selection of an element z (in line 4) having a property that is only satisfied by half the elements, and to the selection of a suitable c (line 3), where at least a third of the candidates are usable. Although there is in principle no bound on the maximal time needed, the variance around the expectation is small because the probability of failing to find a useful z and c drops exponentially with the number of attempts. We emphasize that the ERH is only used to bound the running time (of line 3) and does not affect the error probability, as is the case with the original Miller test.

The detailed implementation of algorithm 3 may be optimized in various ways. The implementation given in the proof that follows this remark has focused on simplicity more than saving a few multiplications. However, we are not aware of any implementation that avoids the $O(u+v)$ term in the complexity analysis.

Proof. We will first argue that only lines 5-9 in the algorithm have any significance in the complexity analysis.

line 2. By Newton iteration the square root of n may be computed using $O(\log\log n)$ multiplications.

line 3. By lemma 5, we expect to find a c of size $O((\log n \log\log n)^2)$ such that $(c/n) = -1$ after three attempts (or discover that n is composite).

line 4. z is selected randomly from $R(n,c) \setminus \{0\}$. We expect to find z with $(N(z)/n) = 1$ after two attempts (or discover that n is composite).

line 5-9. Here we need to explain how it is possible to simultaneously verify that $\overline{z} = z^n$, and do both a 4'th-root-of-1-test and a 3'rd-root-of-1-test without using too many multiplications. We refer to lemma 4 for the implementation of arithmetic in $R(n,c)$.

Define s, r by $n = 2^u 3^v s + r$ for $0 \leq r < 2^u 3^v$. A simple calculation confirms that

$$q = ns + rs + (r^2 - 1)/(2^u 3^v), \tag{10}$$

where the last fraction is integral. Go through the following computational steps using the z selected in line 4 of the algorithm:

1. compute z^s.

   This uses $2\log n + o(\log n)$ multiplications in $\mathbf{Z}_n$.

2. compute z^n.

   Starting from step 1 this requires $O(v+u)$ multiplications in $\mathbf{Z}_n$.

3. verify $z^n = \overline{z}$.

4. compute z^q.

   One may compute z^q from step 1 using $O(v+u)$ multiplications in $\mathbf{Z}_n$, when using (10) and the shortcut $z^{ns} = \overline{z^s}$, where the shortcut is implied by step 3 and exponentiation and conjugation being commuting maps.

5. compute $z^{3^v q}, z^{2\cdot 3^v q}, z^{2^2 3^v q}, \ldots, z^{2^{u-2}3^v q}$.

   Starting from step 4 this requires $O(v+u)$ multiplications in $\mathbf{Z}_n$.

6. verify that $z^{3^v q} = 1$ or $z^{2^i 3^v q} = -1$ for some $0 \le i \le u-2$. If there is $i_0 \ge 1$ with $z^{2^{i_0} 3^v q} = -1$ and if ξ_4 is present, verify that $z^{2^{i_0-1}3^v q} = \pm\xi_4$.

7. compute $z^{2^u q}, z^{2^u 3q}, z^{2^u 3^2 q}, \ldots, z^{2^u 3^{v-1} q}$.

   Starting from step 4 this requires $O(v+u)$ multiplications in $\mathbf{Z}_n$.

8. By step 6 there must be an i $(0 \le i \le v)$ such that $z^{2^u 3^i q} = 1$. Let i_0 be the smallest such i. If $i_0 \ge 1$ verify that $z^{2^u 3^{i_0-1} q}$ is a root of x^2+x+1. If ξ_3 is present, verify in addition that $z^{2^u 3^{i_0-1} q} = \xi_3^{\pm 1}$

■

3 An expression bounding the error probability

The analysis of our primality test falls in two parts. In the first subsection, we deduce an expression describing the probability of passing the basic Frobenius test (line 5 of algorithm 3). In the second subsection this analysis is augmented to encompass the 4'th-root-of-1 and 3'rd-root-of-1 tests (lines 6-9f of algorithm 3).

3.1 The Frobenius test

The analysis of the Frobenius test is based on understanding the structure of the following groups and thereby constructing expressions for bounding the absolute and relative sizes of them.

Definition 7 *Let n be an odd number, let c be a unit modulo n.*

$$U(n,c) \stackrel{\text{def}}{=} \{z \in R(n,c)^* \mid (N(z)/n) = 1\}$$
$$G(n,c) \stackrel{\text{def}}{=} \{z \in U(n,c) \mid \overline{z} = z^n \quad \textit{and} \quad z^{(n^2-1)/2} = 1\}$$

For prime power p^m dividing n, let $G(n,p^m,c)$ denote the set of those $z_0 \in R(p^m,c)$ for which there exists $z \in G(n,c)$ satisfying that $z \equiv z_0 \bmod p^m$.

$$\begin{array}{ccccccc}
R(n,c)^* & \simeq & R(p_1^{m_1},c)^* & \times & \cdots & \times & R(p_\omega^{m_\omega},c)^* \\
| & & | & & & & | \\
U(n,c) & & | & & & & | \\
| & & | & & & & | \\
G(n,c) & \simeq & G(n,p_1,c) & \times & \cdots & \times & G(n,p_\omega,c)
\end{array}$$

Figure 2: Subgroup and isomorphism relations

Expressed in terms of these definitions, the EQFT draws a random $z \in U(n,c)$ and in line 5 of algorithm 3 it checks that $z \in G(n,c)$, which should be the case if n is a prime and $(c/n) = -1$. Hence, the probability of not discovering a composite n in line 5 alone is

$$\frac{|G(n,c)|}{|U(n,c)|} \tag{11}$$

It is fairly clear from the definitions that $U(n,c)$, $G(n,c)$ and $G(n,p^m,c)$ are all groups.

Figure 2 illustrates the subgroup and isomorphism relations that holds (assuming $n = \prod_{i=1}^{\omega} p_i^{m_i}$). We will in turn characterise the structure and size of $R(n,c)$ and $G(n,c)$.

Lemma 8 *Let n be an odd integer and let c be a unit modulo n.*

1. *if p is a prime and $(c/p) = -1$ then*

$$R(p,c)^* \quad \simeq \quad \mathbf{Z}_{p^2-1}$$

and $z^p = \overline{z}$ for $z \in R(p,c)$

2. *if p is a prime and $(c/p) = 1$ then*

$$R(p,c)^* \quad \simeq \quad \mathbf{Z}_{p-1} \times \mathbf{Z}_{p-1},$$

$z^p = z$ and $\overline{(z_1,z_2)} = (z_2,z_1)$ for $z = (z_1,z_2) \in R(p,c)$

3. *if p^m is a prime power divisor of n, then*

$$R(p^m,c)^* \quad \simeq \quad \mathbf{Z}_{p^{m-1}} \times \mathbf{Z}_{p^{m-1}} \times R(p,c)^*$$

4. *If n has prime power factorisation $n = \prod_{i=1}^{\omega} p_i^{m_i}$ then*

$$R(n,c)^* \quad \simeq \quad R(p_1^{m_1},c)^* \times \cdots \times R(p_\omega^{m_\omega},c)^*$$

Proof. 1. The condition $(c/p) = -1$ implies that $x^2 - c$ is irreducible over $\mathbf{Z}_p$, and hence $R(p,c)$ is isomorphic to $GF(p^2)$, the finite field with p^2 elements. In this field the map $z \mapsto z^p$ is a field automorphism (it is the identity map on the subfield $GF(p)$). Hence, If $z = ax + b$ then

$$z^p = (ax+b)^p = ax^p + b = ac^{(p-1)/2}x + b = a(c/p)x + b = -ax + b = \overline{z}$$

2. The condition $(c/p) = 1$ implies that c has a square root $d \in \mathbf{Z}_p$, i.e. $x^2 - c = (x-d)(x+d)$. Hence, by Chinese remaindering

$$R(p,c) \simeq \mathbf{Z}[x]/(p, x-d) \times \mathbf{Z}[x]/(p, x+d) \simeq GF(p) \times GF(p)$$

Let $(z_1, z_2) \in R(p,c)$. The map $z \mapsto z^p$ is the identity map on $GF(p)$. Hence, $(z_1, z_2)^p = (z_1^p, z_2^p) = (z_1, z_2)$. Let $(z_1, z_2) = ax + b$. Using that $ax + b = (ad+b, -ad+b)$ and $-ax+b = (-ad+b, ad+b)$, we find that $\overline{(z_1, z_2)} = (z_2, z_1)$.

3. Define the sets $A = \{(1+p)^i \mid i = 1, \ldots, p^{m-1}\}$ and $B = \{(1+px)^i \mid i = 1, \ldots, p^{m-1}\}$. It is easy to verify that $A \cap B = \{1\}$, and each of A and B are cyclic subgroups of $R(n,c)^*$ of order p^{m-1}. Define the homomorphism $h : R(p^m, c)^* \mapsto R(p,c)^*$ by $h(z) = z \bmod p$. Clearly h is surjective, and hence $R(p,c)^*$ is isomorphic to a subgroup of $R(p^m,c)^*$. It suffices to prove that the kernel of h is $A \times B$. Clearly, $A \times B \subseteq h^{-1}(1)$, and since also $|A \times B| = p^{2(m-1)} = |h^{-1}(1)|$, the proof is complete.

4. By Chinese remaindering. ■

Lemma 9 *Let n be an odd number, and let c satisfy that $(c/n) = -1$. Then $U(n,c)$ is a subgroup of $R(n,c)^*$, and*

$$|U(n,c)| \geq \frac{1}{2}|R(n,c)^*|$$

Proof. The map $h : z \mapsto (N(z)/n)$ is a multiplicative homomorphism from $R(n,c)^*$ to $\{-1, 1\}$. Hence, $U(n,c) = h^{-1}(1)$ must be a subgroup of $R(n,c)^*$ of index 2 or 1. ■

Lemma 10 *Let n be an odd number, let c be a unit modulo n.*

1. *If prime p divides n then $G(n,p,c)$ is a cyclic subgroup of $R(p,c)^*$ of size*

$$|G(n,p,c)| = \begin{cases} \gcd(n/p - 1, (p^2-1)/2), & \text{if } (c/p) = -1 \\ \gcd((n^2/p^2 - 1)/2, p-1), & \text{if } (c/p) = 1 \end{cases}$$

2. *If prime power p^m divides n then $G(n, p^m, c) \simeq G(n,p,c)$*

3. *If n has prime power factorisation $n = \prod_{i=1}^{\omega} p_i^{m_i}$ then*

$$G(n,c) = G(n, p_1, c) \times \cdots \times G(n, p_\omega, c).$$

Proof. For 1, let $z \in G(n,c)$, and define $z_0 \in R(p,c)$ by $z \equiv z_0 \bmod p$. Since $z \in G(n,c)$, we know that $z_0^n = \overline{z_0}$ and $z_0^{(n^2-1)/2} = 1$. The argument is divided in cases:

Consider first the case $(c/p) = -1$. By lemma 8, $\overline{z_0} = z_0^p$ implying that the order of z_0 divides $\gcd(n-p, (n^2-1)/2) = \gcd(n/p-1, (p^2-1)/2)$. Since the multiplicative subgroup of $R(p,c) \simeq GF(p^2)$ is cyclic, the stated bound on the size of $|G(n,p,c)|$ follows.

Consider next the case $(c/p) = 1$. By lemma 8, $z_0 = z_0^p$, i.e. the order of z_0 in $R(p,c)$ divides $\gcd((n^2-1)/2, p-1) = \gcd((n^2/p^2-1)/2, p-1)$. Since $R(p,c) \simeq GF(p) \times GF(p)$, one may represent z_0 by $(w_1, w_2) \in GF(p) \times GF(p)$, implying that w_1 is in the unique multiplicative subgroup of $GF(p)$ of order

$\gcd((n^2/p^2-1)/2, p-1)$. In addition w_2 is uniquely determined by w_1, since by lemma 8, $(w_2, w_1) = \overline{(w_1, w_2)} = (w_1, w_2)^n = (w_1^n, w_2^n)$. Part 1 of the lemma follows.

For 2, it is enough to argue that p does not divide the order of any element $z \in G(n, p^m, c)$, since, by lemma 8, $G(n, p^m, c)$ is a subgroup of $R(p^m, c)^* \simeq \mathbf{Z}_{p^{m-1}} \times \mathbf{Z}_{p^{m-1}} \times R(p, c)^*$. By definition, $z \in G(n, p^m, c)$ satisfy that $z^{n^2-1} = 1$, and since $p|n$ it follows that $p \not| \, n^2 - 1$.

For 3, we use 2. In addition we need to argue that $G(n, c)$ is the entire Cartesian product and not just a subgroup. Let $A \simeq G(n, p_1, c) \times \cdots \times G(n, p_\omega, c)$. It suffices to prove that $A \subseteq U(n, c)$. Assume to the contrary that $z \in A \setminus U(n, c)$, i.e. $(N(z)/n) = -1$. Since $(N(z)/n) = \prod_{i=1}^{\omega} (N(z)/p_i)^{m_i}$, it must be the case that $(N(z)/p) = -1$ for some $p|n$. Computing modulo p, and using that $\overline{z} = z^p$, we get $-1 = (N(z)/p) = z^{(p+1)(p-1)/2}$ in contradiction with 1. ∎

Lemma 11 *Let n be an odd number with prime power factorisation $n = \prod_{i=1}^{\omega} p_i^{m_i}$, let $\Omega = \sum_{i=0}^{\omega} m_i$, and let c satisfy that $(c/n) = -1$. The probability that n is not found to be composite in line 5 of algorithm 3 is*

$$\begin{aligned}
&\frac{|G(n,c)|}{|U(n,c)|} \\
&\quad \leq \quad 2 \cdot \prod_{i=1}^{\omega} p_i^{2(1-m_i)} \mathrm{sel}[(c/p_i), \frac{(n/p_i - 1, (p_i^2-1)/2)}{p_i^2 - 1}, \frac{((n^2/p_i^2 - 1)/2, p_i - 1)}{(p_i - 1)^2}] \\
&\quad \leq \quad 2^{1-\omega} \prod_{i=1}^{\omega} p_i^{2(1-m_i)} \mathrm{sel}[(c/p_i), \frac{(n/p_i - 1, (p_i^2-1)/2)}{(p_i^2 - 1)/2}, \frac{2}{p_i - 1}] \\
&\quad \leq \quad 2^{1-\Omega}
\end{aligned}$$

where, we have adopted the notation $\mathrm{sel}[\pm 1, E_1, E_2]$ *for a conditional expression with the semantics* $\mathrm{sel}[-1, E_1, E_2] = E_1$ *and* $\mathrm{sel}[1, E_1, E_2] = E_2$.

Proof. The first upper bound for $|G(n,c)|/|U(n,c)|$ follows from combining lemmas 8, 9 and 10. The last two inequalities are trivial simplifications. ∎

3.2 4'th-root-of-1 and 3'rd-root-of-1 tests

In this subsection, we estimate the probability that n passes the 4'th-root-of-1 and 3'rd-root-of-1 tests (lines 6-7f and 8-9f), given that it passes the Frobenius part of the test (line 5), i.e., given that $z \in G(n, c)$. These probabilities can be bounded assuming that the auxiliary inputs r_3, r_4 are "well chosen". We define below exactly which values of r_3, r_4 are good. We let $\beta(n, c)$ be the probability that the entire second part of the test (lines 5-9f) is passed assuming that r_3, r_4 are good. Also, under the same assumption, we let

$$Pr_4(n, c) = Pr(\text{4'th-root-of-1-test passed} \mid z \in G(n, c))$$

$$Pr_3(n, c) = Pr(\text{3'rd-root-of-1-test passed} \mid z \in G(n, c))$$

Let $p_1, \ldots, p_\omega$ be the distinct prime factors in n, and let C_i, respectively D_i be the Sylow-2, respectively the Sylow-3 subgroup of $G(n, p_i, c)$. Then we have

$$G(n, c) \simeq C_1 \times \ldots \times C_\omega \times D_1 \times \ldots \times D_\omega \times H$$

where $|H|$ is prime to 2,3. Recall that in the test, we write $n^2 - 1 = q2^u3^v$. If z is uniformly chosen in $G(n, c)$ and we write elements in $G(n, c)$ according to the above decomposition as $2\omega + 1$-tuples, we have

$$z^{q3^v} = (c_1, ..., c_\omega, 1, ..., 1, 1) \quad z^{q2^u} = (1, ..., 1, d_1, ..., d_\omega, 1)$$

where c_i is uniform in C_i and d_i is uniform in D_i, and so these two group elements are independently distributed. Since furthermore the result of the 4'th-root-of-1-test depends only on z^{q3^v}, r_4 and the 3'rd-root-of-1-test depends only on z^{q2^u}, r_3, we have

$$\beta(n, c) = \frac{|G(n, c)|}{|U(n, c)|} Pr_4(n, c) Pr_3(n, c)$$

We let $T_4(n, c)$ be the set of elements of form $(c_1, ..., c_\omega, 1, ..., 1, 1)$ such that $ord(c_1) = ord(c_2) = ... = ord(c_\omega)$, and $T_3(n, c)$ is the set of elements of form $(1, ..., 1, d_1, ..., d_\omega, 1)$ such that $ord(d_1) = ord(d_2) = ... = ord(d_\omega)$.

r_3, r_4 are said to be *good* if $r_4 \in T_4(n, c)$ and is a non-trivial 4'th root of 1 (different from ± 1), and if $r_3 \in T_3(n, c)$ and is a non-trivial 3'rd root of 1 (different from 1), provided that such non-trivial roots exist in $T_4(n, c), T_3(n, c)$. If not $r_3 = r_4 = 1$ is defined to be good. We now derive bounds for $Pr_4(n, c), Pr_3(n, c)$ (assuming we are given good values of r_3, r_4).

Consider first $Pr_4(n, c)$. The first part of the 4'th-root-of-1-test (line 6) starts from z^{q3^v}, performs some squarings and tests for occurrence of -1. It is easy to see that this first part is passed if and only if $z^{q3^v} \in T_4(n, c)$. Let $|C_i| = 2^{a_i}$ and define $a_{min} = min\{a_i | \ i = 1..\omega\}$. Note that $a_{min} \geq 1$. Of course, the probability that this first part of the 4'th-root-of-1-test is passed is $|T_4(n, c)|/2^{\sum_i a_i}$. Clearly, if $a_{min} = 1$, $|T_4(n, c)| = 2$.

Now assume that $a_{min} > 1$ and that $z^{q3^v} \in T_4(n, c)$. We want to count the number of possible values of $z^{q3^v} \in T_4(n, c)$ for which the second part of the 4'th-root-of-1-test (line 7) is passed, i.e., for which $R_4(z)$ is one of $1, -1, r_4, -r_4$. This happens if z^{q3^v} is ± 1 or is mapped to $\pm r_4$ by 0 or more squarings. Since squaring in the group $C_1 \times \cdots \times C_\omega$ is a 2^ω to 1 mapping, and elements in $T_4(n, c)$ have maximal order $2^{a_{min}}$, the number of such elements is $2 + 2 \cdot 2^{0 \cdot \omega} + 2 \cdot 2^{1 \cdot \omega} + ... + 2 \cdot 2^{(a_{min}-2)\omega}$. It follows that if $a_{min} > 1$, we have

$$Pr_4(n, c) = \frac{|T_4(n, c)|}{2^{\sum_i a_i}} \cdot \frac{2 + 2 \cdot 2^{0 \cdot \omega} + 2 \cdot 2^{1 \cdot \omega} + ... + 2 \cdot 2^{(a_{min}-2)\omega}}{|T_4(n, c)|} \leq 4^{1-\omega}$$

Summarizing, we have

Lemma 12 *If $a_{min} = 1$, we have $Pr_4(n, c) \leq 2^{1-\sum_i a_i}$. If $a_{min} > 1$, we have $Pr_4(n, c) \leq 4^{1-\omega}$.*

We now consider $Pr_3(n, c)$. The first part of the 3'rd-root-of-1-test (line 8) starts from z^{q2^u}, performs some cubings and tests for occurrences of roots in the third cyclotomic polynomial. This first part is passed if and only if $z^{q2^u} \in T_3(n, c)$. Let $|D_i| = 3^{b_i}$, and set $b_{min} = min\{b_i | \ i = 1..\omega\}$. Note that $b_{min} \geq 0$. The probability of passing the first part is $|T_3(n, c)|/3^{\sum_i b_i}$. This is $3^{-\sum_i b_i}$ if $b_{min} = 0$.

Now assume that $b_{min} > 0$ and that $z^{q2^u} \in T_3(n, c)$. Similar to what we did in the 4'th-root-of-1-test, we count the number of possible values for $z^{q2^u} \in$

$T_3(n,c)$, such that $R_3(z)$ is one of $1, r_3, r_3^{-1}$. This number is $1 + 2 \cdot 3^{0 \cdot \omega} + 2 \cdot 3^{1 \cdot \omega} + ... + 2 \cdot 3^{(b_{min}-1)\omega}$. We therefore have:

$$Pr_3(n,c) = \frac{|T_3(n,c)|}{3^{\sum_i b_i}} \cdot \frac{1 + 2 \cdot 3^{0 \cdot \omega} + 2 \cdot 3^{1 \cdot \omega} + ... + 2 \cdot 3^{(b_{min}-1)\omega}}{|T_3(n,c)|} \leq 3^{1-\omega}$$

This leads to

Lemma 13 *If $b_{min} = 0$, we have $Pr_3(n,c) \leq 3^{-\sum_i b_i}$. If $b_{min} > 0$, we have $Pr_3(n,c) \leq 3^{1-\omega}$.*

Clearly, these estimates for $Pr_4(n,c), Pr_3(n,c)$ combined with the formula above for $\beta(n,c)$ can be used to obtain general estimates. However, we need to split the analysis into some cases, since $a_{min} = 1$ and $b_{min} = 0$ require arguments different from the other cases. As a first step, we have

Lemma 14 *If $a_{min} = 1$, we have*

$$\frac{|G(n,c)|}{|U(n,c)|} Pr_4(n,c) \leq 4 \cdot 8^{-\omega} \prod_{i=1}^{\omega} p_i^{2(1-m_i)} \mathrm{sel}[(c/p_i), \frac{(n/p_i - 1, (p_i^2-1)/8)}{(p_i^2-1)/8}, \frac{4}{p_i - 1}]$$

If $b_{min} = 0$, we have

$$\frac{|G(n,c)|}{|U(n,c)|} Pr_3(n,c) \leq 2 \cdot 6^{-\omega} \prod_{i=1}^{\omega} p_i^{2(1-m_i)} \mathrm{sel}[(c/p_i), \frac{(n/p_i - 1, (p_i^2-1)/6)}{(p_i^2-1)/6}, \frac{6}{p_i - 1}]$$

If $a_{min} = 1$ and $b_{min} = 0$, we have

$$\frac{|G(n,c)|}{|U(n,c)|} Pr_4(n,c) Pr_3(n,c) \leq 4 \cdot 24^{-\omega} \prod_{i=1}^{\omega} p_i^{2(1-m_i)} \mathrm{sel}[(c/p_i), \frac{(n/p_i - 1, (p_i^2-1)/24)}{(p_i^2-1)/24}, \frac{12}{p_i - 1}]$$

Proof. For the first claim, we have by Lemma 12 that

$$\frac{|G(n,c)|}{|U(n,c)|} Pr_4(n,c) \leq \frac{|G(n,c)|}{|R(n,c)^*|} \frac{4}{2^{a_1} \cdots 2^{a_\omega}} = 4 \prod_{i=1}^{\omega} p_i^{2(1-m_i)} \frac{|G(n,p_i,c)|}{2^{a_i} |R(p_i,c)^*|}$$

Note that by definition of a_i, $|G(n,p_i,c)|/2^{a_i}$ is odd. Therefore we have that if the Jacobi symbol of c modulo p_i is -1,

$$\frac{|G(n,p_i,c)|}{2^{a_i} |R(p_i,c)^*|} = \frac{(n/p_i - 1, (p_i^2-1)/2)}{2^{a_i}(p_i^2-1)} \leq \frac{1}{8} \frac{(n/p_i - 1, (p_i^2-1)/8)}{(p_i^2-1)/8}$$

and if the Jacobi symbol of c modulo p_i is 1,

$$\frac{|G(n,p_i,c)|}{2^{a_i} |R(p_i,c)^*|} = \frac{((n^2/p_i^2 - 1)/2, p_i - 1)}{2^{a_i}(p_i - 1)^2} \leq \frac{1}{8} \frac{4}{p_i - 1}$$

This proves the first claim. The other two can be argued in similar ways, details are left to the reader. ■

This lemma, combined with the conclusions of Lemmas 12, 13 for $a_{min} > 1, b_{min} > 0$ immediately implies:

Theorem 15 *Let n be an odd composite number with prime power factorisation $n = \prod_{i=1}^{\omega} p_i^{m_i}$, let $\Omega = \sum_{i=0}^{\omega} m_i$, and let c satisfy that $(c/n) = -1$.*

Given good values of the inputs r_3, r_4, the error probability of a single iteration of the second part of the EQFT (algorithm 3) is bounded by

$$\begin{aligned}\beta(n,c) &\le \frac{|G(n,c)|}{|U(n,c)|} Pr_4(n,c) Pr_3(n,c) \\ &\le 24^{1-\omega} \prod_{i=1}^{\omega} p_i^{2(1-m_i)} \mathrm{sel}[(c/p_i), \frac{(n/p_i - 1, (p_i^2-1)/24)}{(p_i^2-1)/24}, \frac{12}{p_i - 1}] \\ &\le 24^{1-\Omega}\end{aligned}$$

The assumption on r_3, r_4 in the above theorem means that $r_3 \in T_3(n,c), r_4 \in T_4(n,c)$, and furthermore that both are non-trivial roots of 1, if such roots exist in $T_3(n,c), T_4(n,c)$. However, when EQFT is executed as described earlier, these auxiliary inputs are produced such that r_3 is either 1 or is $R_3(z)$ for some base z that leads to accept, and similarly for r_4. This does ensure that $r_3 \in T_3(n,c), r_4 \in T_4(n,c)$, but of course not that they are non-trivial roots. Fortunately, the probability that they are non-trivial is sufficiently large that the theorem can still be used to bound the actual error probability:

Theorem 16 *Let n be an odd composite number with ω distinct prime factors.*

For any $t \ge 1$, the error probability $\beta_t(n)$ of t iterations of EQFT (algorithm 3) is bounded by

$$\beta_t(n) \quad \le \quad \max_{(c/n)=-1} \; 4^{\omega-1}\beta(n,c)^t$$

Proof. Let $\beta_t(n,c)$ denote the probability that a composite n passes t iterations of the second part of algorithm 3 with $r_3 = r_4 = 1$ on the first iteration. Clearly, $\beta_t(n) \le \max_{(c/n)=-1} \beta_t(n,c)$, and it suffices to prove that $\beta_t(n,c) \le 4^{\omega-1}\beta(n,c)^t$

Fix any c with $(c/n) = -1$. Then the proof splits in cases, according to the values of a_{min}, b_{min}. Assume first that $a_{min} > 1, b_{min} > 0$. Then non-trivial 3'rd and 4'th roots exist in $T_3(n,c), T_4(n,c)$. Let $EQFT^t(R)$ denote t iterations of $EQFT$ using random input R (used in choosing z-values, for instance). Let $EQFT_O^t(R)$ denote t iterations, where the algorithm is given two non-trivial roots r_3, r_4 from an oracle O. By construction of the algorithm, this means that all iterations will use r_3, r_4 as auxiliary input. From Theorem 15 it is immediate that $EQFT_O^t(R)$ accepts n with probability $\beta(n,c)^t$.

There are 2^ω possible non-trivial values of r_3 in $T_3(n,c)$. For each such r_3, using r_3^{-1} as auxiliary input instead leads to the same behavior of the test, so there are $2^{\omega-1}$ essentially different choices of r_3. Similar reasoning shows that there are $2^{\omega-1}$ essentially different choices of r_4. Hence we can make in a natural way $4^{\omega-1}$ essentially different pairs (r_3, r_4), and define oracles $O_1, ..., O_{4^{\omega-1}}$ where each oracle outputs its own pair of values.

Consider now the following experiment: on input n, we run $EQFT^t(R)$ and also $EQFT_{O_i}^T(R)$ for $i = 1, ..., 4^{\omega-1}$. The probability that for some i, $EQFT_{O_i}^t(R)$ accepts, is at most $4^{\omega-1}\beta(n,c)^t$. So it is enough to show that if $EQFT^t(R)$ accepts, then for some i, $EQFT_{O_i}^t(R)$ accepts. To see this, consider some R for which all z-values chosen in $EQFT^t(R)$ lead to trivial values of auxiliary input, i.e., $R_3(z) = R_4(z) = 1$ in all iterations. In this case, if $EQFT^t(R)$

accepts, so does every $EQFT^t_{O_i}(R)$ because no comparisons with the values from the oracle take place. On the other hand, if R is such that some iterations in $EQFT^t(R)$ produce non-trivial roots, then the first such values found, say r_3, r_4, will be used for comparison in all following iterations. Furthermore, there exists some i for which O_i outputs $(r_3^{\pm 1}, \pm r_4)$, and if $EQFT^t(R)$ accepts, then $EQFT^t_{O_i}(R)$ will also accept. A similar argument shows that if a non-trivial value of only r_3 or only r_4 is produced, then $EQFT^t_{O_i}(R)$ will accept for $2^{\omega-1}$ values of i.

This finishes the case $a_{min} > 1, b_{min} > 0$. For $a_{min} = 1, b_{min} > 0$, observe that there are then no non-trivial 4'th roots of 1 in $T_4(n,c)$. We can then run the same argument, but this time with $2^{\omega-1}$ oracles ranging over essentially different values of non-trivial 3'rd roots of 1. In this case, we get that $\beta_t(n,c) \leq 2^{\omega-1}\beta(n,c)^t$, and the same results follows if $a_{min} > 1, b_{min} = 0$. Finally, for $a_{min} = 1, b_{min} = 0$, there is nothing to prove since there are no non-trivial roots, and we have $\beta_t(n,c) = \beta(n,c)^t$. ■

4 Average Case Behaviour

This section analyses what happens when EQFT is applied to generate random probable prime numbers.

4.1 Uniform Choice of Candidates

Let M_k be the set of odd k-bit integers ($2^{k-1} < n < 2^k$). Consider the algorithm that repeatedly chooses random numbers in M_k, until one is found that passes t iterations of EQFT, and outputs this number.

The expected time to find a "probable prime" with this method is at most tT_k/p_k, where T_k is the expected time for running the test on a random number from M_k, and p_k is the probability that a such a number is prime. Suppose we choose n at random and let $n^2 - 1 = 2^u 3^v q$, where q is prime to 2 and 3. It is easy to see that the expected values of u and v are constant, and so it follows from Theorem 6 that T_k is $2k + o(k)$ multiplications modulo a k bit number. This gives approximately the same time needed to generate a probable prime, as if we had used $2t$ iterations of the Miller-Rabin test in place of t iterations of $EQFT$. But, as we shall see, the error probability is much smaller than with $2t$ MR tests.

Let $q_{k,t}$ be the probability that the algorithm above outputs a composite number. The rest of this section is aimed at finding estimates for $q_{k,t}$. We recall that the EQFT algorithm tests if primes less than 13 divide n, so numbers with such small prime factors are always rejected, this will be useful below.

When running t iterations of our test on input n, it follows from Theorem 16 and Theorem 15 that the probability $\beta_t(n)$ of accepting n satisfies

$$\beta_t(n) \leq 4^{\omega-1} 24^{t(1-\Omega)} \max\{\frac{(n/p-1,(p^2-1)/24)}{(p^2-1)/24}, \frac{12}{p-1}\}^t$$

where p is the largest prime factor in n and Ω is the number of prime factors in n, counted with multiplicity (and where of course $\beta_t(n) = 0$ if n is divisible by primes less than 13). Let $\sigma_t = \log_2 24 - 2/t$. Using this and $\omega \leq \Omega$, we can rewrite the estimate to

$$\beta_t(n) \leq (2^{\sigma_t})^{t(1-\Omega)} \max\{\frac{(n/p-1,(p^2-1)/24)}{(p^2-1)/24}, \frac{12}{p-1}\}^t$$

Define $\beta_\sigma(n)$, for any positive σ, by: $\beta_\sigma(n) = 0$ if n is divisible by a prime less than 13, and otherwise

$$\beta_\sigma(n) = 2^{\sigma(1-\Omega)} \max\{\frac{(n/p-1,(p^2-1)/24)}{(p^2-1)/24}, \frac{12}{p-1}\} \tag{12}$$

For any t and any composite n, the above estimate of $\beta_t(n)$ shows that t iterations of EQFT accept n with probability no larger than $\beta_{\sigma_t}(n)^t$.

Now assume we have a (hypothetical) primality test that always accepts a prime and accepts a composite n with probability $\beta_\sigma(n)$. Suppose we used this test in place of EQFT when generating a probable prime, and let $q_{\sigma,k,t}$ be the resulting error probability. It is then clear that $q_{k,t} \leq q_{\sigma_t,k,t}$. So to estimate $q_{k,t}$, it is enough to estimate $q_{\sigma,k,t}$ for all $t \geq 1$ and all σ with $\log_2 24 - 2 \leq \sigma \leq \log_2 24$.

We define $C_{\sigma,m}$ to be the class of odd composite integers with $\beta_\sigma(n) > 2^{-m}$. Since $\beta_\sigma(n) \leq 2^{\sigma(1-\Omega)}$ we have for $n \in C_{\sigma,m}$ that $\Omega < m/\sigma + 1$. Let $N(m,k,j)$ be the set of integers in $C_{\sigma,m} \cap M_k$ with $\Omega = j$. Then trivially,

$$|C_{\sigma,m} \cap M_k| = \sum_{2 \leq j < m/\sigma+1} |N(m,k,j)| \tag{13}$$

The goal in the following will be to estimate $|N(m,k,j)|$ and use the above to estimate $|C_{\sigma,m} \cap M_k|$.

For an $n \in N(m,k,j)$ we have $n > 2^{k-1}$ and $\Omega = j$. This implies for the largest prime factor p in n that $p > 2^{(k-1)/j}$, and so, for $p > 3$, we have $1/(p-1) \leq 2^{-(k-1)/j} \cdot 4/3$.

Now, let us assume that $m + \sigma + 4 \leq \sqrt{4\sigma(k-1)}$. In general, it holds for any positive j that $\sqrt{4\sigma(k-1)} \leq \sigma j + (k-1)/j$. This, together with the above estimate on $1/(p-1)$, gives us $12/(p-1) \leq 2^{-m-\sigma(1-j)}$.

Now, (12) gives us that any $n \in N(m,k,j)$ must satisfy

$$\max\{\frac{(n/p-1,(p^2-1)/24)}{(p^2-1)/24}, \frac{12}{p-1}\} > 2^{-m-\sigma(1-j)}$$

Inserting the estimate on $12/(p-1)$, we get

$$\frac{(n/p-1,(p^2-1)/24)}{(p^2-1)/24} > 2^{-m-\sigma(1-j)}$$

If we define

$$d(p,n) = \frac{(p^2-1)/24}{(n/p-1,(p^2-1)/24)},$$

we have $d(p,n) < 2^{m+\sigma(1-j)}$.

This means that for any prime $p > 2^{(k-1)/j}$ and integer $d|(p^2-1)/24$ with $d < 2^{m+\sigma(1-j)}$, we can count the number of $n \in M_k$ with the property that $p|n$,

$d = d(p, n)$ and n is composite. This is at most the number of solutions to the system

$$n = 0 \bmod p, \quad n = p \bmod \frac{p^2-1}{24d}, \quad p < n < 2^k$$

By the Chinese remainder theorem, the number of solutions is at most

$$\frac{2^k\ 24d}{p(p^2-1)}$$

We therefore have

$$\begin{aligned} |N(m,k,j)| &\leq \sum_{p>2^{(k-1)/j}} \quad \sum_{d<2^{m+\sigma(1-j)},\ (24d)|(p^2-1)} \frac{2^k\ 24d}{p(p^2-1)} \\ &= 2^k \sum_{d<2^{m+\sigma(1-j)}} \quad \sum_{p>2^{(k-1)/j},\ (24d)|(p^2-1)} \frac{24d}{p(p^2-1)} \end{aligned}$$

Taking only the inner sum in this, and define $T(24d)$ to be the number of solutions $x \in \{1, 2, \ldots, 24d\}$ to the congruence $x^2 = 1 \bmod 24d$, we get

$$\begin{aligned} \sum_{p>2^{(k-1)/j},\ (24d)|p^2-1} \frac{24d}{p(p^2-1)} &\leq \frac{9}{8} \sum_{p>2^{(k-1)/j},\ p^2\equiv 1 \bmod 24d} \frac{24d}{p^3} \\ &\leq \frac{9}{8}\, T(24d) \sum_{u=0}^{\infty} \frac{24d}{(24du + 2^{(k-1)/j})^3} \\ &\leq \frac{9}{8}\, \frac{T(24d)}{(24d)^2} \sum_{u=0}^{\infty} (u + \frac{2^{(k-1)/j}}{24d})^{-3} \end{aligned}$$

To bound the latter, we use the assumptions $d < 2^{m+\sigma(1-j)}$ and $\sigma j + (k-1)/j \geq \sqrt{4\sigma(k-1)} \geq m + \sigma + 4$:

$$\frac{2^{(k-1)/j}}{24d} > 24^{-1} \cdot 2^{-m+\sigma(j-1)+(k-1)/j} \geq \frac{2^4}{24} = \frac{2}{3}$$

For $c > 2/3$, it holds that $\sum_{u=0}^{\infty}(u+c)^{-3} < c^{-3} + \int_c^{\infty} x^{-3}dx = c^{-2}(1/c + 1/2) \leq 2c^{-2}$. Using this, our inner sum above can be estimated as

$$\sum_{p>2^{(k-1)/j},(24d)|p^2-1} \frac{24d}{p(p^2-1)} \leq T(24d)\ 3^2\ 2^{-2(k-1)/j-2}$$

Inserting into the expression for $|N(m,k,j)|$, we get

$$\begin{aligned} |N(m,k,j)| &\leq 2^k\ 3^2\ 2^{-2(k-1)/j-2} \sum_{d<2^{m+\sigma(1-j)}} T(24d) \\ &\leq 2^k\ 3^2\ 2^{3\sigma/2+1+3m/2-3\sigma j/2-2(k-1)/j} \end{aligned}$$

Here, we have used that $T(24d) = 2^{1+\omega(24d)} \leq 2^{3+\log_5 d} < 8\sqrt{d}$. Inserting the estimate for $|N(m,k,j)|$ in (13), we get:

$k \setminus t$	1	2	3	4
300	42	105	139	165
400	49	125	165	195
500	57	143	187	221
600	64	159	208	245
1000	86	212	276	325

Table 1: Lower bounds on $-\log_2 q_{k,t}$

Theorem 17 *Let m, k be positive integers with $m+\sigma+4 \leq \sqrt{4\sigma(k-1)}$. Then we have*

$$|C_{\sigma,m} \cap M_k| \leq 2^{k+3\sigma/2+1}\ 3^2\ 2^{3m/2} \sum_{2\leq j\leq m/\sigma+1} 2^{-3\sigma j/2-2(k-1)/j}$$

Let us now choose some M with $3 \leq M \leq \sqrt{4\sigma(k-1)} - \sigma - 4$ (this is possible if $k \geq 10$). Using exactly the same arguments as in Prop. 1 of [5], we get that

$$q_{\sigma,k,t} \leq \frac{2^{-Mt}|M_k \setminus C_{\sigma,M}| + \sum_{m=3}^{M} 2^{-(m-1)t}|M_k \cap C_{\sigma,m}|)}{\pi(2^k) - \pi(2^{k-1})}$$

Prop. 2 of [5] says that $\pi(2^k) - \pi(2^{k-1}) \geq 0.71867 \cdot 2^k/k$. Let $f(k) = 0.71867 \cdot 2^k/k$. Then inserting the result of the theorem and changing summation order, we have

$$\begin{aligned} f(k)q_{\sigma,k,t} &\leq 2^{-Mt+k-2} + \\ &\quad 2^{k+3\sigma/2+1}\ 3^2 \sum_{j=2}^{M/\sigma+1} \sum_{m=\sigma(j-1)}^{M} 2^{-(m-1)t+3m/2-3\sigma j/2-2(k-1)/j} \\ &= 2^{-Mt+k-2} + \\ &\quad 2^{t+k+3\sigma/2+1}\ 3^2 \sum_{j=2}^{M/\sigma+1} 2^{-3\sigma j/2-2(k-1)/j} \sum_{m=\sigma(j-1)}^{M} 2^{m(3/2-t)} \end{aligned}$$

Numerical estimates for $q_{k,t} \leq q_{\sigma_t,k,t}$ can obtained from this by choosing an optimal value of M within the range given. Some sample results are shown in the table 1, which contains $-\log_2$ of the estimates, so we assert that, e.g., $q_{500,2} \leq 2^{-143}$.

To get an explicit expression, we can use the general inequality that for $t \geq 2$, $\sum_{m=x}^{M} 2^{m(3/2-t)} \leq 2^{x(3/2-t)}/(1-2^{3/2-t})$. We use this with $x = \sigma(j-1)$. Moreover, we want to use the estimate for $q_{\sigma,k,t}$ we derived above with $M = \sqrt{8\sigma(k-1)/t}$. Up to an additive constant, we can do this for all $2 \leq t \leq k-1$. Inserting this and substituting σ_t for σ, one easily obtains

Theorem 18 *For $2 \leq t \leq k-1$, we have that $q_{k,t}$ is $O(k^{3/2}2^{(\sigma_t+1)t}t^{-1/2}4^{-\sqrt{2\sigma_t tk}})$*

In comparison, results in [5] for the corresponding probability $p_{k,t}$ for the Miller-Rabin test say, for instance, that $p_{k,t}$ is $O(k^{3/2}2^t t^{-1/2}4^{-\sqrt{tk}})$ for $2 \leq t \leq$

$k/9$. In our case, σ_t is at least $\log_2 24 - 2$ and approaches $\log_2 24$ as t increases. Since $2\log_2 24 \simeq 9.2$, this analysis indicates that if several iteration of EQFT are performed, then roughly speaking each iteration has the effect of 9 Miller-Rabin tests, while only taking time equivalent to about 2 MR tests.

Note that [5] contains sharper numeric estimates for the MR test than what the above type of analysis implies, and also more work has been done in this direction after [5], for instance [2]. However, such methods for better estimates on the MR test could also be applied to our test so that the relative strengths of the tests is likely to remain the same.

4.2 Incremental Search

The algorithm we have just analysed is in fact seldom used in practice. Most real implementations will not want to choose candidates for primes uniformly at random. Instead one will choose a random starting point n_0 in M_k and then test $n_0, n_0+2, n_0+4, ..$ for primality until one is found that passes t iterations of the test. Many variations on this theme are possible, such as other step sizes, various types of sieving, but the basic principle remains the same. The reason for applying such an algorithm is that test division by small primes can be implemented much more efficiently because one can exploit the fact that different candidates are related (see for instance [4]). On the other hand, the analysis we did above depends on the assumption that candidates are independent. In [3], a way to get around this problem for the Miller-Rabin test was suggested. We apply an extension of that technique here.

We will analyse the following example algorithm which depends on parameters t and s:

1. Choose n_0 uniformly in M_k, set $n = n_0$, and execute the following loop until it stops:
   (a) Run up to t iterations of EQFT on n, if n passes all iterations, output n and exit loop.
   (b) Otherwise, set $n = n + 2$. If $n \geq n_0 + 2s$, exit loop, else go to step 1a.
2. If the loop in the previous step produced a number n, output n and stop. Otherwise, go to step 1.

So this algorithm tries incremental search from a random starting point until s candidates have been examined. If no probable prime was found, it tries again with a new starting point.

To estimate the expected running time of this method, let $T_k(n_0, s)$ be the maximal running time of EQFT on any of the inputs $n_0, n_0+2, ..., n_0+2(s-1)$. We shall see below that under the prime r-tuple conjecture, if we choose s to be $\theta(k)$, then the expected number of starting points we need to try is constant, in fact very close to 1 for the value we recommend below, namely $s = 10\ln(2^k)$. For such a choice of s, the expected run time is at most $O(stE[T_k(n_0, s)])$, where $E[\cdot]$ refers to the expectation over the choice of n_0, and in practice an upper bound is $stE[T_k(n_0, s)]$ if we choose $s = 10\ln(2^k)$.

To estimate $E[T_k(n_0, s)]$, we need to look at a random set of numbers $n_0, n_0+2, ..., n_0+2(s-1)$ and estimate the maximal powers of 2 and 3 that divide n^2-1

where n is any of the numbers in our set. For any 2-power 2^u where $u > 2$, it holds that $2^u | n^2 - 1 = (n+1)(n-1)$ only if n is 1 or -1 modulo 2^{u-1}. So this always happens for some n in the set if $2^{u-1} \leq 2s$ (since then the values $n+1, n-1$ cover all even residues modulo 2^{u-1}), whereas for larger values the probability drops exponentially with u. It follows that the expected value for the maximal u such that 2^u divides one of our numbers $n^2 - 1$, is $O(\log s)$. A similar argument holds for 3-powers. We conclude from this and Theorem 6 that $E[T_k(n_0, s)]$ is $O(k)$ multiplications, and so the expected time to find a probable prime by the above algorithm is at most $O(tk^2)$ multiplications modulo k bit numbers, if s is $\theta(k)$. As mentioned, practice shows that for $s = 10 \ln 2^k$, we need almost all the time only one value of n_0, and so $st(2k+o(k))$ multiplications is an upper bound. Of course, this refers to the run time when only the EQFT is used. In practice, one would use test division and other tricks to eliminate some of the non primes faster than EQFT can do it. This may reduce the run time significantly. Any such method can be used without affecting the error estimates, as long as no primes are rejected.

Let $q_{k,t,s}$ be the probability that one execution of the loop (steps 1a-1b) outputs a composite number. To do this, we consider again the hypothetical test from the previous subsection, that accepts composites with probability $\beta_\sigma(n)$, and analyse what happens if we use this test in place of EQFT in the algorithm. We let $q_{\sigma,k,t,s}$ be the probability that one execution of the loop outputs a composite in this case. Then, in the same way as before, it follows that $q_{k,t,s} \leq q_{\sigma_t,k,t,s}$.

Recall that we defined $C_{\sigma,m}$ to be the set of odd composites with $\beta_\sigma(n) > 2^{-m}$. From this, we define: $D_{\sigma,m,k,s} = \{n \in M_k | \; [n..n+2s[\cap C_{\sigma,m} \neq \emptyset\}$, for $m \geq 3$. Of course $D_{\sigma,2,k,s} = \emptyset$ by the worst-case error bound.

Since a number in $C_{\sigma,m}$ can be in at most s different intervals of form $[n..n+2s[$, we clearly have

Lemma 19 *$D_{\sigma,m-1,k,s} \subset D_{\sigma,m,k,s}$ and $|D_{\sigma,m,k,s}| \leq s \cdot |M_k \cap C_{\sigma,m}|$*

The idea with defining the sets $D_{\sigma,m,k,s}$ is that if we are lucky enough to choose a starting point n_0 for the inner loop which is *not* in $D_{\sigma,m,k,s}$, then we know that all composites we will test before the loop exits will pass with probability at most 2^{-m}. This translates into a bound on $q_{\sigma,k,t,s}$ as follows:

Lemma 20 *Let $s = c \cdot \ln(2^k)$ for some constant c. Then for any $M \geq 3$, we have*

$$q_{\sigma,k,t,s} \leq 0.5(ck)^2 \sum_{m=3}^{M} \frac{|C_{\sigma,m} \cap M_k|}{|M_k|} 2^{-t(m-1)} + 0.7ck2^{-tM}$$

Proof. Let E be the event that we output a composite, and identify $D_{\sigma,m,k,s}$ with the event that the starting point n_0 is in $D_{\sigma,m,k,s}$. Then we have

$$\begin{aligned} q_{\sigma,k,t,s} &= \sum_{m=3}^{M} P(E \cap (D_{\sigma,m,k,s} \setminus D_{\sigma,m-1,k,s})) + P(E \cap \neg D_{\sigma,M,k,s}) \\ &\leq \sum_{m=3}^{M} P(D_{\sigma,m,k,s}) P(E | (D_{\sigma,m,k,s} \setminus D_{\sigma,m-1,k,s})) + P(E \cap \neg D_{\sigma,M,k,s}) \end{aligned}$$

Consider the case where some fixed $n_0 \notin D_{\sigma,m,k,s}$ was chosen as starting point. Then no candidate n we test will be in $M_k \cap C_{\sigma,m}$, and so will pass all tests with probability at most 2^{-mt}. The probability of outputting a composite in such a case is clearly maximal when all numbers in the interval we consider are composite. In this case, we accept one of the candidates with probability at most $s \cdot 2^{-mt}$. From this and Lemma 19, we get

$$\begin{aligned} q_{\sigma,k,t,s} &\leq s^2 \sum_{m=3}^{M} \frac{|C_{\sigma,m} \cap M_k|}{|M_k|} 2^{-t(m-1)} + s \cdot 2^{-tM} \\ &\leq 0.5(ck)^2 \sum_{m=3}^{M} \frac{|C_{\sigma,m} \cap M_k|}{|M_k|} 2^{-t(m-1)} + 0.7ck2^{-tM} \end{aligned}$$

■

From this lemma and Theorem 17, we can directly get numeric estimates of $q_{k,t,s} \leq q_{\sigma_t,k,t,s}$ for any value of s, by choosing an optimal value of M.

To analyse the overall error probability of the algorithm, observe that the inner loop always terminates when the starting point is a prime. This happens with probability $(\pi(2^k) - \pi(2^{k-1}))/|M_k| \geq 2.8/k$, by the estimates we gave earlier. Moreover, the error probability of our algorithm cannot be worse than that of a procedure that runs the inner loop up to k^2 times and outputs a composite if all executions of the loop output "fail". Clearly, the error probability of this modified algorithm is at most

$$Q_{k,t,s} = k^2 q_{k,t,s} + (1 - 2.8/k)^{k^2},$$

and so we have an estimate of the overall error, for any value of s.

What remains is to consider the choice of s. Based on Hardy and Littlewoods prime r-tuple conjecture, it is shown in [3] that when $s = c \cdot \ln(2^k)$, the probability of failure is less than $2\exp(-2c)$ for all large enough k (and is in fact essentially $\exp(-2c)$). Overwhelming heuristic evidence shows that this is an accurate estimate for realistic values of k [4]. So for instance, for $c = 10$, we fail with probability about 2^{-28}, or once in 256 million times. In other words, with such a choice of c, the algorithm will almost always terminate after one execution of the inner loop, so this gives us all the efficiency advantages of the incremental search method. Table 2 shows sample numerical results of the analysis, with $c = 10$.

5 Worst case analysis

The basic Fermat test is known to have a very bad worst case performance because of the existence of Carmichael numbers. Such numbers have at least $\Omega = 3$ factors. Combined with the Miller-Rabin error bound, $2^{1-\Omega}$, this gives the wellknown worst case error bound 2^{-2}.

For the Frobenius test, one can define a similar concept of generalised Carmichael numbers. Much less is known about them, but Grantham [6] essentially proved

[4] even though this was shown in [3] in connection with the MR test, the result applies to any algorithm of the form we consider here, as long as the test used always accepts a prime number

$k \setminus t$	1	2	3	4
300	18	74	107	133
400	26	93	132	162
500	34	109	153	186
600	40	125	174	210
1000	62	176	239	288

Table 2: Estimates of the overall error probability with incremental search, lower bounds on $-\log_2 Q_{k,t,s}$ using $s = c \cdot \ln(2^k)$ and $c = 10$.

time in MR-units	6	12	18	...	large t
MR	4^{-6}	4^{-12}	4^{-18}	...	4^{-t}
Grantham	19.8^{-6}	19.8^{-12}	19.8^{-18}	...	19.8^{-t}
Müller	50.8^{-6}	50.8^{-12}	50.8^{-18}	...	50.8^{-t}
EQFT w/o $\sqrt[3]{1}$-part	40.3^{-6}	50.7^{-12}	54.8^{-18}	...	$(\approx 64)^{-t}$

Table 3: Worst case error bounds per time spent on the test

that only generalised Carmichael numbers with at least $\Omega = 5$ factors will be bad for the Frobenius test (if they exist). In this section we give a different and slightly stronger formulation of this result. Combined with theorems 15 and 16 this implies that t iterations of the EQFT has a worst case error bound of $4^4 24^{-4t} = 256/331776^t$, except for an explicit finite set of small numbers.

However, this error bound is not impressive considering that the test can be quite slow in the worst case, when large powers of 2 and 3 divides $n^2 - 1$ (cfr. theorem 6).

By omitting the 3'rd-root-of-1-test from algorithm 3, it is possible to give a much better worst case bound on the runtime combined with a slightly weakened bound on the error probability. In fact, a test omitting the 3'rd-root-of-1-test takes time $2 \log n + o(\log n)$ per iteration for a worst case n (with high probability and assuming the ERH) and t iterations err with probability bounded by $2^4 8^{-4t} = 16/4096^t$. In an actual implementation, one may then first compute the powers of 2 and 3 that divide $n^2 - 1$ (this takes negligible time compared to EQFT itself) and based on this decide whether the full EQFT or the version without 3'rd-root-of-1-test is best on this n.

For comparison of our test with the earlier tests of Grantham, Müller and Miller-Rabin, assume that we are willing to spend the same fixed amount of time testing an input number. Table 3 shows that our test (the EQFT without the 3'rd-root-of-1-test) gives asymptotically a better bound on the error probability: using time approximately corresponding to t Miller-Rabin test, we get a bound of $1/7710^{t/3} \approx 1/19.8^t$ using Granthams test and a bound of $4/64^t$ using our test.

5.1 improved implementation without the 3'rd-root-of-1-test

It is convenient to describe the implementation using an extra multiplicative homomorphism in addition to the norm $N(\cdot)$:

Definition 21 *Define the multiplicative homomorphism*

$$P(\cdot): \quad R(n,c) \mapsto R(n,c), \quad P(z) = \overline{z}/z = \overline{z}^2/N(z) \tag{14}$$

The homomorphisms $N(\cdot)$ and $P(\cdot)$ allow some short cut in the implementation:

Lemma 22 *Let n be an odd number, let c be a unit modulo n, and let $z \in R(n,c)^*$. Let $n+1 = 2^v r$ for r odd, and let $n-1 = 2^u s$ for s odd (i.e. $n^2-1 = 2^{v+u} rs$ for rs odd). Define*

$$w_0 = \begin{cases} N(z^{(s-1)/2}) z^{(n+1)/2} & , \textit{for } n \equiv 1 \bmod 4 \\ P(z^{(r-1)/2}) z^{(n-1)/2} & , \textit{for } n \equiv 3 \bmod 4 \end{cases}$$

$$w_i = \begin{cases} N(z^{s2^{i-1}}) & , \textit{for } n \equiv 1 \bmod 4, \textit{ and } i = 1,2,\ldots,u-1 \\ P(z^{r2^{i-1}}) & , \textit{for } n \equiv 3 \bmod 4, \textit{ and } i = 1,2,\ldots,v-1 \end{cases}$$

If $z^n = \overline{z}$ then $w_i = z^{2^i rs}$ for $i = 0,1,2,\ldots$

Proof. The essential observation is that exponentiation regarded as a map commute with the maps $N(\cdot)$ and $P(\cdot)$, since these are multiplicative homomorphisms. The rest is trivial computation, which we divide in 2 cases. Assume first that $n \equiv 1 \bmod 4$, in which case $N(z) = z^{n+1} = z^{2r}$. One may compute $w_0 = N(z^{(s-1)/2}) z^r = N(z)^{(s-1)/2} z^r = z^{rs}$, and $w_i = N(z^{s2^{i-1}}) = N(z)^{s2^{i-1}} = z^{rs2^i}$ for $i \geq 1$. Similarly, if $n \equiv 3 \bmod 4$, it is the case that $P(z) = z^{n-1} = z^{2s}$. One may compute $w_0 = P(z^{(r-1)/2}) z^s = P(z)^{(r-1)/2} z^s = z^{rs}$ and $w_i = P(z^{r2^{i-1}}) = P(z)^{r2^{i-1}} = z^{rs2^i}$ for $i \geq 1$. ■

High powers of elements in $R(n,c)$ may be computed efficiently (for c small) by lemma 4 when representing $z \in R(n,c)$ in the natural way by $(A_z, B_z) \in \mathbf{Z}_n \times \mathbf{Z}_n$, i.e. $z = A_z x + B_z$. In addition, we need the following results for the homomorphisms:

Lemma 23 *Let $z, w \in R(n,c)$:*

1. *$N(z)$ may be computed from z and z^2 using 1 multiplication in $\mathbf{Z}_n$*
2. *it may be decided whether $P(z) = 1$ or $P(z) = -1$ without using arithmetic in $\mathbf{Z}_n$.*

Proof. For (1) we use that $N(z) = 2B_z^2 - B_{z^2}$.

For (2) assume that $z \neq 0$, then $P(z) = 1$ if and only if $\overline{z} = z$, i.e. $A_z = 0$. Similarly, $P(z) = -1$ if and only if $\overline{z} = -z$, i.e. $B_z = 0$. ■

Theorem 24 *There is an implementation of algorithm 3 (omitting the 3'rd-root-of-1-test of lines 8-9f) that with very high probability uses* $2\log n + o(\log n)$ *multiplications in* $\mathbf{Z}_n$*, when assuming the ERH.*

Proof. We need only argue why the $O(u+v)$ term in the complexity bound of theorem 6 can be omitted. It suffices to explain how it is possible to simultaneously verify that $\overline{z} = z^n$, and do a 4'th-root-of-1-test without using too many multiplications.

By lemma 22 this is equivalent to testing $z^n = \overline{z}$ and looking for a minimal i with $w_i = 1$ and $2^{i+1} \mid n^2 - 1$. If such $i \geq 1$ exists, we must, in addition check that $w_{i-1} = -1$ and (possibly if $i \geq 2$) $w_{i-2} = \pm\xi_4$ for some given ξ_4. By the representation of lemma 4 and lemma 23 this may all be done using $(2 + o(1))\log n$ multiplications in $\mathbf{Z}_n$, when taking a bit of care:

First consider the case of $n \equiv 1 \bmod 4$, and let $n - 1 = 2^u s$ for s odd. The algorithm need only calculate the following powers of z

$$z^{(s-1)/2}, z^s, z^{2s}, \ldots, z^{n-1}, z^n \tag{15}$$

To avoid potentially many costly $N(\cdot)$ computations, we can make a binary search for a minimal i with $w_i = 1$, i.e. in total $O(\log\log n)$ additional multiplications.

Next consider the case of $n \equiv 3 \bmod 4$, and let $n + 1 = 2^v r$, for r odd. The algorithm need only calculate the following powers of z

$$z^{(r-1)/2}, z^r, z^{2r}, \ldots, z^{n+1} \tag{16}$$

To avoid any additional divisions, one may note that $(N(z)/n) = 1$ implies z being invertible in $R(n,c)$. Hence, the test $z^n = \overline{z}$ is equivalent to $z^{n+1} = N(z)$, and the test $P(z^{(r-1)/2})z^{(n-1)/2} = \alpha$ is equivalent to $N(z^{(r-1)/2})z^{(n+1)/2} = \alpha \cdot z^r$. ■

5.2 Error probability without the 3'rd-root-of-1-test

The error analysis of section 3 is simplified when omitting the 3'rd-root-of-1-test. We omit the details and state only an analogue of theorems 15 and 16:

Theorem 25 *Let* n *be an odd number with prime power factorisation* $n = \prod_{i=1}^{\omega} p_i^{m_i}$*, let* $\Omega = \sum_{i=0}^{\omega} m_i$*, and let* c *satisfy that* $(c/n) = -1$*.*

1. *Given a good value of the input* r_4*, the error probability* $\gamma(n,c)$ *of a single iteration of the second part of algorithm 3 (omitting the 3'rd-root-of-1-test of line 8-9f) is bounded by*

$$\begin{aligned}
\gamma(n,c) &\leq \frac{|G(n,c)|}{|U(n,c)|} Pr_4(n,c) \\
&\leq 8^{1-\omega} \prod_{i=1}^{\omega} p_i^{2(1-m_i)} \mathrm{sel}[(c/p_i), \frac{(n/p_i - 1, (p_i^2-1)/8)}{(p_i^2-1)/8}, \frac{4}{p_i - 1}] \\
&\leq 8^{1-\Omega}
\end{aligned}$$

2. *the error probability $\gamma_t(n)$ of t iterations of algorithm 3 (omitting the 3'rd-root-of-1-test of line 8-9f) is bounded by*

$$\gamma_t(n) \quad \leq \quad \max_{(c/n)=-1} \; 2^{\omega-1}\gamma(n,c)^t$$

Proof. Omitted. ■

Theorem 26 *Let n be an odd composite number. The probability that t iterations of the test of algorithm 3 (omitting the 3'rd-root-of-1-test of line 8-9f) result in the output "probable prime" when input n is bounded by*

$$\gamma_t(n) \;\leq\; 2^{4-12t}$$

if either n has no prime factor $\leq 2^7$ or $n \geq 2^{67.5}$.

Remark. the bounds $\leq 2^7$ and $\geq 2^{67.5}$ are not optimal. In their proof independent bounds on the basic Frobenius test and the 4'th-root-of-1-test are combined. If a simultaneous bound on the two tests were considered (analogously to lemma 14), better bounds on the exceptional set of small numbers are possible. The reason for presenting a nonoptimal result is technical simplicity. We can reuse the analysis of section 3 unchanged.

Proof. By theorem 25, $\gamma_t(n) \leq 2^{(\Omega-1)(1-3t)}$. Hence, we need only consider numbers with at most 4 prime factors. For such numbers it suffices to prove that $\gamma(n,c) \leq 2^{-12}$. For this we use that $\gamma(n,c) = |G(n,c)|/|U(n,c)| \cdot Pr_4(n,c)$. By lemma 12, $Pr_4(n,c) \leq 2^{1-\omega}$, and $|G(n,c)|/|U(n,c)|$ is bounded in lemmas 30 and 31 for numbers with few prime factors.

For numbers with no small prime divisors, we consider the table of lemma 30. By solving an inequality for each entry in the table, we can find a bound on the smallest prime factor p, that makes all entries $\leq 2^{-12}2^{\omega-1}$. It turns out that the bottleneck is the case $\Omega = \omega = 4$, requiring $p > 2^7$.

For large n, we analogously consider the table of lemma 31. Here the bottleneck is also the case $\Omega = \omega = 4$, requiring $n > 2^{67.5}$. ■

5.3 Error probability without the 4'th-root-of-1 and 3'rd-root-of-1 tests

For composite numbers with at most 4 prime factors, it is possible to get good bounds on the error probability $|G(n,c)|/|U(n,c)|$ of the basic Frobenius test (line 5 of algorithm 3) alone, i.e. omitting both the 4'th-root-of-1 and 3'rd-root-of-1 tests.

The bound can be parametrised either by the smallest prime factor or by the size of n. This result is a simple consequence of the analysis of section 3 except in the case of n having an odd number of all distinct prime factors. For 3 distinct prime factors, the proof hinges on a technical result by Grantham [6]. We give a different proof and a slightly sharper result in lemma 28. We haven't found a way to parametrise the error bound for numbers with 5 distinct prime factors, but a result in that direction would allow an improvement of the absolute worst case bound stated in theorem 26.

5.3.1 Technical lemmas

Lemma 27 *Let n be an odd number with prime power factorisation $n = \prod_{i=1}^{\omega} p_i^{m_i}$, let $\Omega = \sum_{i=0}^{\omega} m_i$, and let c satisfy that $(c/n) = -1$.*

$$\begin{aligned}
\frac{|G(n,c)|}{|U(n,c)|} &\leq 2\prod_{i=1}^{\omega} p_i^{2(1-m_i)} \mathrm{sel}[(c/p_i), \\
&\quad \frac{\gcd(n/p_i - 1, (p_i^2-1)/2)}{p_i^2 - 1}, \frac{\gcd((n^2/p_i^2 - 1)/2, p_i - 1)}{(p_i - 1)^2}] \\
&\leq 2\prod_{i=1}^{\omega} p_i^{2(1-m_i)} \mathrm{sel}[(c/p_i), \min\{\frac{1}{2}, \frac{n/p_i - 1}{p_i^2 - 1}\}, \min\{\frac{1}{p_i - 1}, \frac{n^2/p_i^2 - 1}{(p_i - 1)^2}\}]
\end{aligned}$$

Proof. This follows from lemma 11. ∎

Lemma 28 *Let n be an odd composite number that is the product of 3 distinct primes $n = \prod_{i=1}^{3} p_i$. Assume that $p_1 < p_2 < p_3$, then the following inequality holds*

$$1 < \frac{\prod_{i=1}^{3}(n/p_i - 1)}{\prod_{i=1}^{3}(p_i^2 - 1)} < 1 + \frac{1}{p_1^2 - 1} \tag{17}$$

and

$$\prod_{i=1}^{3} \frac{\gcd(n/p_i - 1, p_i^2 - 1)}{p_i^2 - 1} < \frac{1}{p_1^2 - 1} \tag{18}$$

Proof. We start by proving (17). Define

$$f(p_1, p_2, p_3) = \frac{(p_1p_2 - 1)(p_1p_3 - 1)(p_2p_3 - 1)}{(p_1^2 - 1)(p_2^2 - 1)(p_3^2 - 1)}$$

When differentiating f with respect to p_3, one easily finds that under the assumption $1 < p_1 < p_2 < p_3$ then

$$f(p_1, p_2, p_3) < \lim_{p_3 \to \infty} f(p_1, p_2, p_3) = \frac{(p_1p_2 - 1)p_1p_2}{(p_1^2 - 1)(p_2^2 - 1)}$$

and

$$f(p_1, p_2, p_3) > f(p_1, p_2, p_2) = \frac{(p_1p_2 - 1)^2}{(p_1^2 - 1)(p_2^2 - 1)}$$

When differentiating the simplified expressions with respect to p_2 one finds that (assuming $1 < p_1 < p_2$)

$$f(p_1, p_2, p_3) < \lim_{p_2 \to \infty} \frac{(p_1p_2 - 1)p_1p_2}{(p_1^2 - 1)(p_2^2 - 1)} = 1 + \frac{1}{p_1^2 - 1}$$

and

$$f(p_1, p_2, p_3) > f(p_1, p_1, p_1) = 1$$

Finally, we consider the last inequality. Since

$$1 < \frac{\prod_{i=1}^{3}(n/p_i - 1)}{\prod_{i=1}^{3}(p_i^2 - 1)} = \frac{\prod_{i=1}^{3}(n/p_i - 1)/\gcd(n/p_i - 1, p_i^2 - 1)}{\prod_{i=1}^{3}(p_i^2 - 1)/\gcd(n/p_i - 1, p_i^2 - 1)}$$

it follows that

$$1+\frac{1}{\prod_{i=1}^{3}(p_i^2-1)/\gcd(n/p_i-1,p_i^2-1)}\le\frac{\prod_{i=1}^{3}(n/p_i-1)}{\prod_{i=1}^{3}(p_i^2-1)}<1+\frac{1}{p_1^2-1}$$

which proves the lemma. ■

Lemma 29 *Let n be an odd composite number that is the product of 4 distinct primes $n=\prod_{i=1}^{4}p_i$. Assume that $1<p_1<p_2<p_3<p_4$, then the following inequality holds*

$$p_1^3<\frac{\prod_{i=2}^{4}(n/p_i-1)}{\prod_{i=2}^{4}(p_i^2-1)}<p_1^3(1+\frac{3}{p_2^2-1}) \tag{19}$$

and

$$\prod_{i=2}^{4}\frac{\gcd(n/p_i-1,p_i^2-1)}{p_i^2-1}<\frac{3p_1^3}{p_2^2-1} \tag{20}$$

Proof. We start by proving the upper bound of (19).

$$\begin{aligned}\frac{\prod_{i=2}^{4}(n/p_i-1)}{\prod_{i=2}^{4}(p_i^2-1)} &< \frac{\prod_{i=2}^{4}n/p_i}{\prod_{i=2}^{4}(p_i^2-1)}\\ &= \frac{p_1^3}{\prod_{i=2}^{4}(p_i^2-1)/p_i^2}\\ &= p_1^3\prod_{i=2}^{4}(1+\frac{1}{p_i^2-1})\\ &< p_1^3(1+\frac{3}{p_2^2-1})\end{aligned}$$

The lower bound is implied by (17):

$$\frac{\prod_{i=2}^{4}(n/p_i-1)}{\prod_{i=2}^{4}(p_i^2-1)} > p_1^3\frac{\prod_{i=2}^{4}((n/p_1)/p_i-1)}{\prod_{i=2}^{4}(p_i^2-1)} > p_1^3$$

Finally, we consider the last inequality. Since

$$p_1^3<\frac{\prod_{i=2}^{4}(n/p_i-1)}{\prod_{i=2}^{4}(p_i^2-1)}=\frac{\prod_{i=2}^{4}(n/p_i-1)/\gcd(n/p_i-1,p_i^2-1)}{\prod_{i=2}^{4}(p_i^2-1)/\gcd(n/p_i-1,p_i^2-1)}$$

it follows that

$$p_1^3+\frac{1}{\prod_{i=2}^{4}(p_i^2-1)/\gcd(n/p_i-1,p_i^2-1)}\le\frac{\prod_{i=2}^{4}(n/p_i-1)}{\prod_{i=2}^{4}(p_i^2-1)}<p_1^3+\frac{3p_1^3}{p_1^2-1}$$

which proves the lemma. ■

5.3.2 Worst case bound parametrised by smallest prime factor

Lemma 30 *Let n be an odd composite number having complete prime factorisation $n = \prod_{i=1}^{\omega} p_i^{m_i}$, let $\Omega = \sum_{i=1}^{\omega} m_i \le 4$, and let c satisfy that $(c/n) = -1$.*

If p is the smallest prime factor of n, then $|G(n,c)|/|U(n,c)|$ is bounded by the entries of the following table

	$\Omega = 2$	$\Omega = 3$	$\Omega = 4$
$\omega = 1$		p^{-4}	p^{-6}
$\omega = 2$	$2(p^2-1)^{-1}$	$2^{-1}p^{-2}$	$2^{-1}p^{-4}$
$\omega = 3$		$2(p^2-1)^{-1}$	$2^{-2}p^{-2}$
$\omega = 4$			$2^{-2}(p-1)^{-1}$

Proof. All the entries with $\Omega > \omega$ are immediate consequences of lemma 27. For the entry with $\Omega = \omega = 4$, we argue that since $(c/n) = -1$ and 4 is an even number, we must have $(c/p_i) = 1$ for some prime factor p_i of n. Hence, the bound $2^{-2}(p_i - 1)^{-1}$ is also implied by lemma 27. For the entry with $\Omega = \omega = 2$, we have $n = p_1 p_2$, and without loss of generality, we may assume that $(c/p_1) = -(c/p_2) = -1$. By lemma 27, we have

$$\frac{|G(n,c)|}{|U(n,c)|} \le 2\frac{\gcd(p_2-1,(p_1^2-1)/2)}{p_1^2-1}\cdot\frac{\gcd((p_1^2-1)/2,p_2-1)}{(p_2-1)^2} \le \frac{2}{p_1^2-1}$$

Finally, consider the case of $\Omega = \omega = 3$, i.e. $n = p_1p_2p_3$. The assumption $(c/n) = -1$ implies that $(c/p_i) = -1$ either for all i or for precisely one i. Consider first the latter case, and assume $(c/p_1) = -1$ and $(c/p_2) = (c/p_3) = 1$. By lemma 27,

$$\frac{|G(n,c)|}{|U(n,c)|} \le \frac{1}{p_2-1}\cdot\frac{1}{p_3-1} \le \frac{1}{p^2-1}$$

In the former case, $(c/p_i) = -1$ for all i, and by the inequality of lemma 28, we have

$$\frac{|G(n,c)|}{|U(n,c)|} \le \frac{2}{p^2-1}$$

■

5.3.3 Worst case bound on the form n^{-const}

Lemma 31 *Let n be an odd composite number having complete prime factorisation $n = \prod_{i=1}^{\omega} p_i^{m_i}$, let $\Omega = \sum_{i=1}^{\omega} m_i \le 4$, and let c satisfy that $(c/n) = -1$. $|G(n,c)|/|U(n,c)|$ is bounded by the entries of the following table*

	$\Omega = 2$	$\Omega = 3$	$\Omega = 4$
$\omega = 1$		$n^{-4/3}$	$n^{-3/2}$
$\omega = 2$	$2n^{-2/3}$	$n^{-2/3}$	$2^2n^{-10/9}$
$\omega = 3$		$2n^{-2/5}$	$n^{-2/3}$
$\omega = 4$			$n^{-2/15}$

Proof. The entries for $\omega = 1$ is an immediate consequence of lemma 27, and each of the remaining entries of the table is proved by separate case analysis in the following. For all the cases, we will use the inequalities of lemma 27, and for $n = p_1p_2p_3$ and $n = p_1p_2p_3p_4$, we need additional technical results (lemma 28 and lemma 29).

Case $n = p_1p_2p_3p_4$:

The assumption $(c/n) = -1$ implies that $(c/p_i) = -1$ either for precisely three i or for precisely one i, and in the latter case we may assume that $(c/p_1) = -1$ without loss of generality,

$$\frac{|G(n,c)|}{|U(n,c)|} \leq 2 \cdot \min\{\frac{1}{2}, \frac{p_2p_3p_4 - 1}{p_1^2 - 1}\} \cdot \prod_{i=2}^{4} \frac{1}{p_i - 1} \leq 2^2 n^{-2/3}$$

In the former case, where $(c/p_i) = -1$ for precisely three i, we assume that $(c/p_1) = 1$,

$$\begin{aligned}\frac{|G(n,c)|}{|U(n,c)|} &\leq 2\frac{1}{p_1 - 1} \cdot \frac{1}{2} \cdot \min\{\frac{1}{2} \cdot \frac{1}{2}, \frac{p_1p_2p_4 - 1}{p_3^2 - 1} \cdot \frac{p_1p_2p_3 - 1}{p_4^2 - 1}\} \\ &\leq 2 \cdot \min\{\frac{1}{8(p_1 - 1)}, \frac{p_1^2p_2^3}{n}\} \\ &\leq 2^{-1}n^{-1/6} \qquad \text{for } p_1 > p_2\end{aligned}$$

We need the assumption $p_1 > p_2$ to make the above estimate. This is always possible after permutation of indices except when p_1 is the smallest prime factor. In that case, we may assume that $p_1 < p_2 < p_3 < p_4$, and by Lemma 29 and the part of the previous argument that holds also when $p_1 < p_2$, we find

$$\frac{|G(n,c)|}{|U(n,c)|} < 2 \cdot \min\{\frac{1}{8(p_1 - 1)}, \frac{p_1^2p_2^3}{n}, \frac{3p_1^3}{(p_1 - 1)(p_2^2 - 1)}\} \leq n^{-2/15}$$

Case $n = p_1p_2p_3$:

The assumption $(c/n) = -1$ implies that $(c/p_i) = -1$ either for all i or for precisely one i, and in the latter case, we may assume without loss of generality that $(c/p_1) = -1$:

$$\frac{|G(n,c)|}{|U(n,c)|} \leq 2 \cdot \min\{\frac{1}{2}, \frac{p_2p_3 - 1}{p_1^2 - 1}\} \cdot \frac{1}{p_2 - 1} \cdot \frac{1}{p_3 - 1} \leq 2^2 n^{-2/3}$$

In the former case, $(c/p_i) = -1$ for all i. Without loss of generality, we may assume that $p_1 < p_2 < p_3$, and by lemma 28,

$$\begin{aligned}\frac{|G(n,c)|}{|U(n,c)|} &\leq 2 \cdot \prod_{i=1}^{3} \frac{\gcd(n/p_i - 1, (p_i^2 - 1)/2)}{p_i^2 - 1} \\ &\leq 2 \cdot \min\{\frac{1}{p_1^2 - 1}, \frac{1}{2} \cdot \frac{p_1p_3 - 1}{p_2^2 - 1} \cdot \frac{p_1p_2 - 1}{p_3^2 - 1}\} \\ &\leq 2n^{-2/5}\end{aligned}$$

Case $n = p_1^2p_2p_3$:

Without loss of generality, $(c/p_2) = 1$ and $(c/p_3) = -1$,

$$\frac{|G(n,c)|}{|U(n,c)|} \leq 2 \cdot p_1^{-2} \cdot \frac{1}{2} \cdot \frac{1}{p_2 - 1} \cdot \min\{\frac{p_1^2p_2 - 1}{p_3^2 - 1}, \frac{1}{2}\} \leq n^{-2/3}$$

Case $n = p_1 p_2$:

Since $(c/n) = -1$, it must be the case that $(c/p_1) = -(c/p_2) = -1$ (if necessary permute p_1 and p_2).

$$\frac{|G(n,c)|}{|U(n,c)|} \leq 2 \cdot \min\{\frac{1}{2}, \frac{p_2 - 1}{p_1^2 - 1}\} \cdot \frac{1}{p_2 - 1} \leq 2n^{-2/3}$$

Case $n = p_1^2 p_2$:

It must be the case that $(c/p_2) = -1$ and therefore

$$\frac{|G(n,c)|}{|U(n,c)|} \leq 2 \cdot p_1^{-2} \cdot \frac{1}{2} \cdot \min\{\frac{1}{2}, \frac{p_1^2 - 1}{p_2^2 - 1}\} \leq n^{-2/3}$$

Case $n = p_1^3 p_2$:

There are two possibilities, either $(c/p_1) = -(c/p_2) = -1$ or $(c/p_1) = -(c/p_2) = 1$. Consider the latter possibility first,

$$\frac{|G(n,c)|}{|U(n,c)|} \leq 2 \cdot p_1^{-4} \cdot \frac{1}{p_1 - 1} \cdot \min\{\frac{1}{2}, \frac{p_1^3 - 1}{p_2^2 - 1}\} \leq 2^2 n^{-10/9}$$

We need also consider the situation when $(c/p_1) = -(c/p_2) = -1$,

$$\frac{|G(n,c)|}{|U(n,c)|} \leq 2 \cdot p_1^{-4} \cdot \frac{1}{2} \cdot \min\{\frac{p_1^6 - 1}{(p_2 - 1)^2}, \frac{1}{p_2 - 1}\} \leq 2^2 n^{-10/9}$$

■

References

[1] Eric Bach and Jeffrey Shallit. *Algorithmic Number Theory. Volume I: Efficient algorithms.* MIT Press, 1996.

[2] Ronald Burthe: *Further investigations with the strong probable prime test*, Math. Comp. 65, pp.373-381, 1996.

[3] J.Brandt and I.Damgård: *On Generation of Probable Primes by Incremental Search*, Proc. of Crypto 92, Springer Verlag LNCS series.

[4] J.Brandt, I.Damgård and P.Landrock: *Speeding up Prime Number Generation*, Proc. of AsiaCrypt 91, Springer Verlag LNCS series.

[5] Damgård, Landrock and Pomerance: *Average Case Error Estimates for the Strong Probable Prime Test*, Math. Comp., vol. 61, 1993, pp.177-194.

[6] Jon Grantham: *A Probable prime test with high confidence*, J.Number Theory 72, pp.32-47, 1998.

[7] Siguna Müller: *A probable prime test with very high confidence for* $n \equiv 1 \bmod 4$, Proceeedings of Asiacrypt 2001, Springer Verlag LNCS.