BRICS

Basic Research in Computer Science

BRICS RS-03-9 Damgård & Frandsen: An Extended Quadratic Frobenius Primality Test

An Extended Quadratic Frobenius Primality Test with Average and Worst Case Error Estimates

Ivan B. Damgård
Gudmund Skovbjerg Frandsen

BRICS Report Series RS-03-9

ISSN 0909-0878 February 2003

See back inner page for a list of recent BRICS Report Series publications. Copies may be obtained by contacting:

BRICS
Department of Computer Science
University of Aarhus
Ny Munkegade, building 540
DK–8000 Aarhus C
Denmark

Telephone: +45 8942 3360
Telefax: +45 8942 3255
Internet: BRICS@brics.dk

BRICS publications are in general accessible through the World Wide Web and anonymous FTP through these URLs:

http://www.brics.dk
ftp://ftp.brics.dk
This document in subdirectory RS/03/9/

An Extended Quadratic Frobenius Primality Test with Average and Worst Case Error Estimates [*]

Ivan Bjerre Damgård Gudmund Skovbjerg Frandsen

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

{ivan,gudmund}@brics.dk

February, 2003

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 takes time about equivalent to 2 Miller-Rabin tests, but has much smaller error probability, namely $256/331776^t$ for t iterations of the test in the worst case. EQFT extends QFT by verifying additional algebraic properties related to the existence of elements of order dividing 24. We also give 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

[*] 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.

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.

1 Introduction

Efficient methods for primality testing are important, in theory as well as in practice. 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. Primality tests are used, for instance, in public-key cryptography, where efficient methods for generating large, random primes are indispensable tools. Here, it is important to know how the test behaves in the average case. But there are also scenarios (e.g., in connection with Diffie-Hellman key exchange) where one needs to test if a number n is prime and where n may have been chosen by an adversary. Here the worst case performance of the test is important.

Virtually all known probabilistic tests are built on the same basic principle: from the input number n, one defines an Abelian group and then tests if the group structure we expect to see if n is prime, is actually present. 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 [7], 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 [8] which takes the same time as QFT and has error probability essentially[1] 1/131040. Just as for the Miller-Rabin test, however, it seems

[1] The test and analysis results are a bit different, depending on whether the input

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 give better results on 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). EQFT comes in two variants, EQFTac which works well in an average case analysis and EQFTwc, which is better for applications where the worst case behavior is important.

For the average case analysis: consider an algorithm that repeatedly chooses random odd k-bit numbers, subject each number to t iterations of EQFTac, and outputs the first number found that passes all t tests. Under the ERH, each iteration takes expected time equivalent to about 2 Miller-Rabin tests, or 2/3 of the time for QFT/MQFT (the ERH is only used to bound the run time and does not affect the error probability). 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 $\sigma_t = \log_2 24 - 2/t$. Comparison to the similar analysis by Damgård, Landrock and Pomerance for the MR test indicates that for $t \geq 2$, our test in the average case roughly speaking has the effect of 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, we show that t iterations of EQFTwc err with probability at most $256/331776^t$ except for an explicit finite set of numbers[2]. The same worst case error probability can be shown for EQFTac, but this variant 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 $n^2 - 1$. For EQFTwc, on the other hand, t iterations take time equivalent to about $2t+2$ MR tests on all inputs (still assuming ERH). For comparison with EQFT/MQFT, assume that we are willing

is 3 or 1 modulo 4, see [8] for details

[2] namely if n has no prime factors less than 118, or if $n \geq 2^{42}$

to spend the same fixed amount of time testing an input number. Then EQFTwc 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 $256/331776^{3t-1} \approx 1/576^{6t}$ using EQFTwc.

2 The Intuition behind EQFT

2.1 A simple initial idea

Given the number n to be tested, we start by constructing a quadratic extension $Z_n[X]/(f(X))$, which is kept fixed during the entire test (across all iterations). We let H be the multiplicative group in this extension ring. If n is prime, the quadratic extension is a field, and so H is cyclic of order $n^2 - 1$. We may of course assume that n is not divisible by 2 or 3, which implies that $n^2 - 1$ is always divisible by 24. Let H_{24} be the subgroup of elements of order dividing 24. If H is cyclic, then clearly $|H_{24}| = 24$. On the other hand, if n is not prime, H is the direct product of a number of subgroups, one for each distinct prime factor in n, and we may have $H_{24} >> 24$.

Now, suppose we are already given an element $r \in H$ of order 24. Then a very simple approach to a primality test could be the following: Choose a random element z in H, and verify that $z^n = \bar{z}$, where $\bar{z}$ refers to the standard conjugate (explained later). This implies $z^{n^2-1} = 1$ for any invertible z and so is similar to the classical Fermat test. It is, however, in general a much stronger test than just checking the order of z. Then, from z construct an element z' chosen from H_{24} with some "suitable" distribution. For this intuitive explanation, just think of z' as being uniform in H_{24}. Now check that $z' \in< r >$, i.e. is a power of r. This must be the case if n is prime, but may fail if n is composite. This is similar to the part of the MR test that checks for existence of elements of order 2 different from -1.

To estimate the error probability, let ω be the number of distinct prime factors in n. Since H is the direct product of ω subgroups, H_{24} is typically of order 24^{ω}. It may be smaller, but then the Fermat-like part of the test is stronger than otherwise, so we only consider the maximal case in this section. As one might then expect, it can be shown that the error probability of the test is at most $24/24^{\omega}$ times the probability that $z^n = \bar{z}$. The factor $24^{1-\omega}$ corresponds to the factor of $2^{1-\omega}$ one obtains for the MR test.

2.2 Some problems and two ways to solve them

In the above, it is not clear how to construct an element of order 24 (if it exists at all), and we have not specified how to construct z' from z. We present two different approaches to these problems.

2.2.1 EQFTwc

In this approach, we run a start-up procedure that may discover that n is composite. But if not, it constructs an element of order 24 and also guarantees that H contains ω distinct subgroups, each of order divisible by $2^u 3^v$, where $2^u, 3^v$ are the maximal 2- and 3-powers dividing $n^2 - 1$. This procedure runs in expected time $O(1)$ Miller-Rabin tests. Details on the idea behind it are given in Section 5. Having run the start-up procedure, we construct z' as $z' = z^{(n^2-1)/24}$. Note that without the condition on the subgroups of H, we could have $z' = 1$ always which would clearly be bad. Each z can be tested in time approximately 2 MR tests, for any n. This leads to the test we call EQFTwc (since it works well in a worst case analysis).

2.2.2 EQFTac

The other approach avoids spending time on the start-up. This comes at the cost that the test becomes slower on n's where u, v are very large. But this only affects a small fraction of the potential inputs and is not important when testing randomly chosen n, since then the expected values of u, v are constant.

The basic idea is the following: we start choosing random z's immediately, and instead of trying to produce an element in H_{24} from z, we look separately for an element of order dividing 3 and one of order dividing 8. For order 3, we compute $z^{(n^2-1)/3^v}$ and repeatedly cube this value at most v times. This is guaranteed to produce an element of order 3, if 3 divides the order of z, and we do not need to assume that 3^v divides the order of any cyclic component. If we already know an element ξ_3 of order 3, we can check that the new element we produce is in the group generated by ξ_3, and if not, n is composite. Of course, we do not know an element of order 3 from the start, but note that the computations we do on each z may produce such an element. So if we do several iterations of the test, as soon as an iteration produces an element of order 3, this can be used as ξ_3 by subsequent iterations. A similar idea can be applied to elements of order 8.

This leads to a test of strength comparable to EQFTwc, except for one problem: the iterations we do before finding elements of the right order may have larger error probability than the others. This can be compensated for by a number of further tricks: first, rather than choosing z uniformly, we require that $N(z)$ has Jacobi symbol 1, where $N()$ is a fixed homomorphism from H to Z_n^* defined below. This means we can expect z to have order a factor 2 smaller than otherwise[3], and this turns out to improve the error probability of the Fermat-like part of the test by a factor of $2^{1-\omega}$. Moreover, some partial testing of the elements we produce is always possible: for instance, we know n is composite if we see an element of order 2 different from -1. These tricks imply that the test, up to a small constant factor on the error probability, is as good as if we had known ξ_3, ξ_4 from the start. This version of the test is called EQFTac (since it works well in an average case analysis). We show that it satisfies the same upper bound on the error probability as we have for EQFTwc.

2.3 Comparison to other tests

We give 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 $f(x)$. This seems to simplify the error analysis for EQFT. 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 higher order. In fact, our idea of testing membership in a fixed subgroup 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.

A final comment relates to the comparison in running times be-

[3] This also means that we should look for an element ξ_4 of order 4 (and not 8) in the part of the test that produces elements of order a 2-power

tween Miller-Rabin, Grantham's and our test. Using the standard way to state running times in the literature, the Miller-Rabin, resp. Grantham's, resp. our test run in time $\log n + o(\log n)$ resp. $3 \log n + o(\log n)$ resp. $2 \log n + o(\log n))$ multiplications in Z_n. However, taking a closer look, we find that the running time of Miller-Rabin is actually $\log n$ *squarings* $+o(\log n)$ multiplications in Z_n, while the $3 \log n$ $(2 \log n)$ multiplications mentioned for the other tests are a mix of squarings and multiplications. So for an accurate comparison we should compare the times for modular multiplications and squarings. 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.

2.4 The ring $R(n, c)$ and EQFTac

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 (assume $z = ax + b$):*

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

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

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

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

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

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

*4'th-root-of-*1*-test / generalised Miller-Rabin property:* Since $GF(p^2)$ is a field there are 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

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

*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

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

The actual test will have two parts (see algorithm 1). 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 $R(n,c)$ are tested for a random choice of z. When the EQFTac 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 transferred 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.

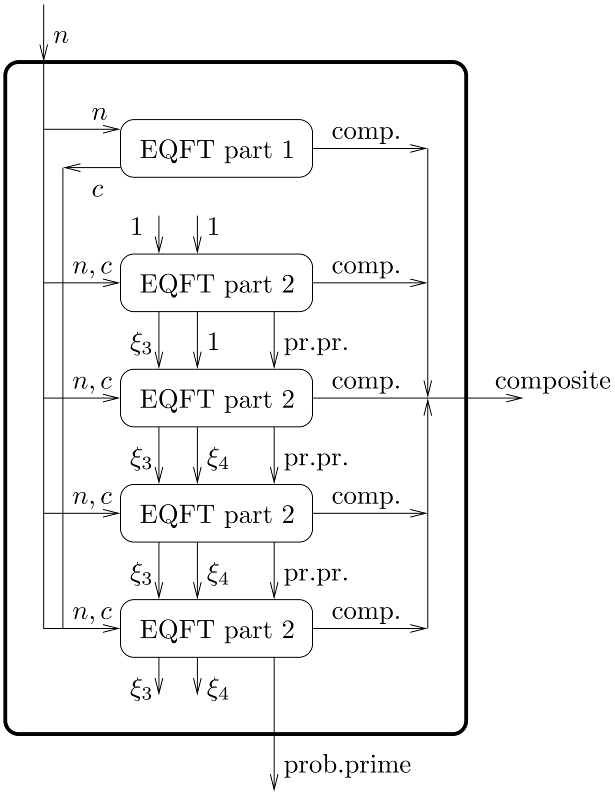

Figure 1: flowchart for 4 iterations of EQFTac over a single n

Here follows som more detailed comments to algorithm 1:

Line 1 ensures that $24 \mid n^2 - 1$. In addition, we will use that n has no small prime factors in the later error analysis.

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 4 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.

Algorithm 1 Extended Quadratic Frobenius Test (EQFTac).

```
First part (construct quadratic extension):
Require: input is odd number $n \geq 5$
Ensure: output is “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):
Require: input is $n, c, r_3, r_4$, where $n \geq 5$ not divisible by 2 or 3, $(c/n) = -1$, $r_3 \in \{1\} \cup \{\xi \in R(n,c) \mid \Phi_3(\xi) = 0\}$ and $r_4 \in \{1, -1\} \cup \{\xi \in R(n,c) \mid \Phi_4(\xi) = 0\}$
    Let $u, v$ be defined by $n^2 - 1 = 2^u 3^v q$ for $(q, 6) = 1$.
Ensure: output is “composite”, or “probable prime”, $s_3, s_4$, where $s_3 \in \{1\} \cup \{\xi \in R(n,c) \mid \Phi_3(\xi) = 0\}$ and $s_4 \in \{1, -1\} \cup \{\xi \in R(n,c) \mid \Phi_4(\xi) = 0\}$
 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 $i = 0, \ldots, u-2$ 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 $i = 0, \ldots, v-1$ 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”, $s_3$, $s_4$
```

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.

2.5 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 3 *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

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

with

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

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 4 *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 5 *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 1 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 1 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 4, 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 3 for the implementation of arithmetic in $R(n,c)$.

Define s, r by $n = 2^u 3^v s + r$ for $0 < r < 2^u 3^v$. 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 $0 \leq i \leq u-2$. If there is $i_0 \geq 1$ 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 $(0 \leq i \leq v)$ such that $z^{2^u 3^i q} = 1$. Let i_0 be the smallest such i. If $i_0 \geq 1$ 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 including the quadratic residuosity test (line 5 of algorithm 1). 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 1).

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

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

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 $z \equiv z_0 \bmod p^m$.

$$\begin{tikzcd}
R(n,c)^* \arrow[d, dash] & \simeq & R(p_1^{m_1},c)^* \arrow[dd, dash] & \times & \cdots & \times & R(p_\omega^{m_\omega},c)^* \arrow[dd, dash] \\
U(n,c) \arrow[d, dash] & & & & & & \\
G(n,c) & \simeq & G(n,p_1,c) & \times & \cdots & \times & G(n,p_\omega,c)
\end{tikzcd}$$

Figure 2: Subgroup and isomorphism relations

Expressed in terms of these definitions, the EQFTac draws a random $z \in U(n,c)$ and in line 5 of algorithm 1 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 7 *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*

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

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

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

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

$z^p = z$ 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*

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

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

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

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 $\overline{ax + b} = (ad + b, -ad + b)$ 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 8 *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 9 *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 7, $\overline{z_0} = z_0^p$ 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 7, $z_0 = z_0^p$, 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 7, $(w_2,w_1) = \overline{(w_1,w_2)} = (w_1,w_2)^n = (w_1^n,w_2^n)$. 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 7, $G(n,p^m,c)$ 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 10 *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 1 is*

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

$$\begin{aligned} &\le \; 2^{1-\omega}\prod_{i=1}^{\omega} p_i^{2(1-m_i)}\mathrm{sel}[(c/p_i), \frac{(n/p_i-1,(p_i^2-1)/2)}{(p_i^2-1)/2}, \frac{2}{p_i-1}] \\ &\le \; 2^{1-\Omega} \end{aligned}$$

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 7, 8 and 9. 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

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

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

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

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 $(c_1,\ldots,c_\omega,1,\ldots,1,1)$ such that $ord(c_1)=\cdots=ord(c_\omega)$, and $T_3(n,c)$ is the set of elements of form $(1,\ldots,1,d_1,\ldots,d_\omega,1)$ such that $ord(d_1)=\cdots=ord(d_\omega)$.

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 then $r_3 = r_4 = 1$ 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 $a_{min} = min\{a_i|\ i=1,\ldots,\omega\}$. 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 $2 + 2\cdot 2^{0\cdot\omega} + 2\cdot 2^{1\cdot\omega} + \cdots + 2\cdot 2^{(a_{min}-2)\omega}$. It follows that if $a_{min} > 1$, we have

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

Summarizing, we have

Lemma 11 *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 $b_{min} = min\{b_i|\ i = 1,\cdots,\omega\}$. 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 $1 + 2 \cdot 3^{0 \cdot \omega} + 2 \cdot 3^{1 \cdot \omega} + \cdots + 2 \cdot 3^{(b_{min}-1)\omega}$. We therefore have:

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

This leads to

Lemma 12 *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 13 *If $a_{min} = 1$, we have*

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

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

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

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

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

Proof. For the first claim, we have by Lemma 11 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 11, 12 for $a_{min} > 1, b_{min} > 0$ immediately implies:

Theorem 14 *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 EQFTac (algorithm 1) is bounded by

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

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 EQFTac 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 15 *Let n be an odd composite number with ω distinct prime factors.*

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

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

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

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 $\mathrm{EQFTac}^t(R)$ denote t iterations of EQFTac using random input R (used in choosing z-values, for instance). Let $\mathrm{EQFTac}^t_O(R)$ 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 14 it is immediate that $\mathrm{EQFTac}^t_O(R)$ 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 $O_1, \ldots, O_{4^{\omega-1}}$ where each oracle outputs its own pair of values.

Consider the following experiment: on input n, we run $\mathrm{EQFTac}^t(R)$ and also $\mathrm{EQFTac}^T_{O_i}(R)$ for $i = 1, \ldots, 4^{\omega-1}$. The probability that for some i, $\mathrm{EQFTac}^t_{O_i}(R)$ accepts, is at most $4^{\omega-1}\beta(n,c)^t$. So it is enough to show that if $\mathrm{EQFTac}^t(R)$ accepts, then for some i, $\mathrm{EQFTac}^t_{O_i}(R)$ accepts. To see this, consider some R for which all z-values chosen in $\mathrm{EQFTac}^t(R)$ lead to trivial values of auxiliary input, i.e., $R_3(z) = R_4(z) = 1$ in all iterations. In this case, if $\mathrm{EQFTac}^t(R)$ accepts, so does every $\mathrm{EQFTac}^t_{O_i}(R)$ because no comparisons with the values from the oracle take place. On the other hand, if R is such that some iterations in $\mathrm{EQFTac}^t(R)$ 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 $\mathrm{EQFTac}^t(R)$ accepts, then $\mathrm{EQFTac}^t_{O_i}(R)$ will also accept. A similar argument shows that if a non-trivial value of only r_3 or only r_4 is produced, then $\mathrm{EQFTac}^t_{O_i}(R)$ 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 EQFTac 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 EQFTac, 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 5 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 EQFTac. 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 EQFTac 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 15 and Theorem 14 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 EQFTac 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 EQFTac 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

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

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:

Theorem 16 *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

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

$k \setminus t$	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 $-\log_2 q_{k,t}$

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 17 *For $2 \leq t \leq k-1$, we have that*

$$q_{k,t} \ \textit{is} \ O(k^{3/2} 2^{(\sigma_t+1)t} t^{-1/2} 4^{-\sqrt{2\sigma_t tk}})$$

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 EQFTac 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 [4]. 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 $n_0, n_0+2, n_0+4, \ldots$ 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 [3]). On the other hand, the analysis we did above depends on the assumption that candidates are independent. In [2], 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 EQFTac 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 EQFTac on any of the inputs $n_0, n_0 + 2, \ldots, n_0 + 2(s-1)$. 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 $n_0, n_0 + 2, \ldots, n_0 + 2(s-1)$ 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 5 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 EQFTac is used. In practice, one would use test division and other tricks to eliminate some of the non primes faster than EQFTac 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 EQFTac 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 18 *$D_{\sigma,m-1,k,s} \subset D_{\sigma,m,k,s}$ 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 19 *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 18, 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 16, 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

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

$k \setminus t$	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 $c = 10$.

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 [2] 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

We present in this section the version of our test (EQFTwc) that has the best performance for worst case n in that there is an upper bound of $2\log n + o(\log n)$ on the expected number of multiplications in $\mathbf{Z}_n$ needed for making the test on a worst case n. In addition there is an expected start up cost of $\leq 2\log n + o(\log n)$ multiplications in $\mathbf{Z}_n$ for the first iteration of the test. The error probability is bounded by $4^{\omega-1}24^{t(1-\Omega)}$, i.e. the same as for EQFTac.

The classic 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 well known worst case error bound 2^{-2}.

[4] even though this was shown in [2] 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

time in MR-units	large t
MR	4^{-t}
Grantham [7]	19.8^{-t}
Müller [8]	50.8^{-t}
EQFTwc	$(\approx 576)^{-t}$

Table 3: Worst case error bounds per time spent on the test (disregarding start-up cost)

For the Frobenius test, one can define a concept of generalised Carmichael numbers, i.e. numbers n for which $z^n = \overline{z}$ for all $z \in R(n,c)^*$ for some c with $(c/n) = -1$. It appears unkown whether any such numbers exist, but Grantham [7] essentially proved that only numbers with at least $\Omega = 5$ factors can be bad for the Frobenius test. In this section we give a different and slightly stronger formulation of this result, implying that t iterations of the EQFTwc has a worst case error bound of $4^4 24^{-4t} = 256/331776^t$, except for an explicit finite set of small numbers.

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 (EQFTwc) 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 $\approx 24^{-4t/2} = 1/576^t$ using our test and disregarding start-up cost.

5.1 The idea behind EQFTwc

To explain our EQFTwc it will be helpful to first make a version of the usual Miller-Rabin algorithm following the same principles. Let us call the resulting algorithm MR′. The first iteration of MR′ will be special, but subsequent iterations will be very simple: select a random $z \in \mathbf{Z}_n^*$ and verify that $z^{(n-1)/2} = \pm 1$. For this test to have a low error probability it is vital that all Sylow-2 subgroups of $z \in \mathbf{Z}_n^*$ have order $\geq 2^u$, where 2^u is the maximal power of 2 dividing $n-1$. This will be ensured by the first iteration that selects a random $z \in \mathbf{Z}_n^*$ subject to the restriction $(z/n) = -1$ and then verifies that $z^{(n-1)/2} = -1$. Each iteration of the MR′ test (including the first) has error probability $\leq 2^{1-\omega}$.

Similarly, we make EQFTwc with a special first iteration that allows very simple subsequent iterations each of which consists in taking a random $z \in R(n,c)$ and checking whether $\overline{z} = z^n$ and $z^{(n^2-1)/24} \in \{r_{24}^i \mid i = 0,\ldots,23\ \}$, where $r_{24} \in R(n,c)$ is a primitive 24th root of 1 that is constructed in the first iteration. The first iteration must in addition ensure that all Sylow-2 subgroups and Sylow-3 subgroups (exist and) are sufficiently large to ensure a low error probability in all iterations. This is done by selecting a random $z \in R(n,c)^*$ that looks like both a nonsquare and a noncube, computing $r_{24} = z^{(n^2-1)/24}$, and checking that $z^{(n^2-1)/2} = -1$ and that $z^{(n^2-1)/3} = r^{\pm 1}$, where r is a primitive 3rd root of 1.

How does one find the necessary element that looks like a nonsquare and a noncube? Computation of the Jacobi symbol will let us recognize 1/2 of all elements as nonsquares. One might expect that computation of the corresponding cubic residuosity symbol will let us recognize 2/3 of all elements as noncubes. Unfortunately, this technique for recognizing noncubes seems to fail for some composite n, though it does work, when n is a prime.

To handle this problem, we take a pragmatic solution: Run a Miller-Rabin test and a search for noncubes in parallel. If n is prime then the search for a noncube will succeed, and if n is composite then the MR-test (or the noncube search) will succeed.

Before presenting the details of the algorithms, we give the necessary preliminaries.

5.2 Quadratic and cubic (non)residuosity in $R(n.c)$

Let us first deal with quadratic nonresiduosity since that is very simple. All we need to know is included in the following lemma:

Lemma 20 *Let n, c satisfy $(c/n) = -1$.*

$$|\{z \in R(n,c)^* \mid (N(z)/n) = -1\ \}| = \frac{1}{2}|R(n,c)^*|$$

Proof. Since the map $z \mapsto (N(z)/n)$ is a multiplicative group homomorphism $R(n,c)^* \mapsto \{-1,1\}$ it suffices to argue that there is at least one $z \in R(n,c)^*$ such that $(N(z)/n) = -1$. Let $n = \prod_{i=1}^{\omega} p_i^{m_i}$, i.e. $(N(z)/n) = \prod_{i=1}^{\omega}(N(z)/p_i)^{m_i}$. Since $(c/n) = -1$, there must be some i for which $(c/p_i) = -1$ and m_i is odd. Modulo p_i, $N(z)$ maps $GF(p_i^2)$ onto $GF(p_i)$, and therefore there is z with $(N(z)/p_i) = -1$, and by the Chinese remainder theorem it is possible to get $(N(z)/n) = \prod_{i=1}^{\omega}(N(z)/p_i)^{m_i} = -1$ for some z. ■

To speak about and compute cubic residuosity it is necessary to introduce $\mathbf{Z}[\zeta]$, the ring of integers extended with a primitive third root of unity ζ (complex root of $z^2 + z + 1$).

The following definitions and facts may be found in a paper by Scheidler and Williams [9].

Define the two conjugate mappings $\sigma_i : \mathbf{Z}[\zeta] \mapsto \mathbf{Z}[\zeta]$ by $\sigma_i(\zeta) = \zeta^i$ for $i = 1, 2$. The rational integer $N(\alpha) = \sigma_1(\alpha)\sigma_2(\alpha)$ is called the norm of $\alpha \in \mathbf{Z}[\zeta]$.

A unit in $\mathbf{Z}[\zeta]$ is an element of norm 1. There are 6 units in $\mathbf{Z}[\zeta]$: $\pm 1, \pm\zeta, \pm\zeta^2$. Two elements $\alpha, \beta \in \mathbf{Z}[\zeta]$ are said to be associates if there exists a unit ϵ such that $\alpha = \epsilon\beta$.

A prime π in $\mathbf{Z}[\zeta]$ is a non unit such that for any $\alpha, \beta \in \mathbf{Z}[\zeta]$, if $\pi|\alpha\beta$, then $\pi|\alpha$ or $\pi|\beta$.

$1 - \zeta$ is a prime in $\mathbf{Z}[\zeta]$ and $N(1 - \zeta) = 3$. If $\pi \neq 1 - \zeta$ is a prime in $\mathbf{Z}[\zeta]$, then $N(\pi) = p^k$ where p is a prime in $\mathbf{Z}$ and $k \in 1, 2$ is the order of p modulo 3, hence $N(\pi) \equiv 1(\mod 3)$.

If $n \equiv 1(\mod 3)$ is a prime in $\mathbf{Z}$ and $r \in \mathbf{Z}$ satisfies that $r^2 + r + 1 \equiv 0 \bmod n$, then $\gcd(n, r - \zeta)$ is a prime $\pi \in \mathbf{Z}[\zeta]$ of norm n.

The cubic residuosity symbol

$$[\cdot/\cdot] : \mathbf{Z}[\zeta] \times (\mathbf{Z}[\zeta] - (1 - \zeta)\mathbf{Z}[\zeta]) \mapsto \{0, 1, \zeta, \zeta^{-1}\}$$

is a multiplicative homomorphism in both arguments. For prime $\pi \in \mathbf{Z}[\zeta]$ where $\pi \neq 1 - \zeta$ and $\alpha \in \mathbf{Z}[\zeta]$:

$$[\alpha/\pi] = \alpha^{\frac{N(\pi)-1}{3}} \bmod \pi$$

We are now ready to make the connection between $R(n, c)$ and $\mathbf{Z}[\zeta]$.

Definition 21 *For arbitrary $n \geq 5$ with $(n, 6) = 1$, for arbitrary c with $(c/n) = -1$, assume there exists an $r = gx + h \in R(n, c)$ with $r^2 + r + 1 = 0$, and if $n \equiv 1 \bmod 3$ assume in addition that $r \in \mathbf{Z}_n$, i.e. $g = 0$.*

Define $res_3 : R(n, c)^* \mapsto \{1, \zeta, \zeta^2\} \subseteq \mathbf{Z}[\zeta]$ by

$$res_3(ax + b) = \begin{cases} [b^2 - ca^2 \ / \ \gcd(n, r - \zeta)], & \text{if } n \equiv 1 \bmod 3 \\ [(b + a(\zeta - h)/g) \ / \ n], & \text{if } n \equiv 2 \bmod 3 \end{cases}$$

The function res_3 is a nice multiplicative homomorphism but for same composite numbers (f.x. $n = 5^3 7$, $c = -3$ and $r = -x/2 - 1/2$) it is trivial. Fortunately, we only need it to be nontrivial for prime n. In addition, we can speed up some later computations, when we know that it is also nontrivial for composite $n \equiv 1 \bmod 3$:

Lemma 22 *For arbitrary $n \geq 5$ with $(n, 6) = 1$, for arbitrary c with $(c/n) = -1$, for arbitrary $r \in R(n, c)$ with $r^2 + r + 1 = 0$, we have res_3 is a multiplicative homomorphism $res_3 : R(n, c)^* \mapsto \{1, \zeta, \zeta^2\} \subseteq \mathbf{Z}[\zeta]$.*

If n is a prime, or if $n \equiv 1 \bmod 3$ and n is not a perfect cube then res_3 is surjective, i.e. $|res_3^{-1}(1)| = |R(n, c)^*|/3$.

Proof. For proof of the homomorphism property, observe that the second argument of the cubic residuosity symbol is either n or a divisor of n, and in any case is constant and independent of $ax + b$. Hence, it suffices to prove that the mapping of $ax + b$ to the first argument of the cubic residuosity symbol is a multiplicative homomorphism when taken modulo n.

For the case of $n \equiv 1 \bmod 3$, it suffices to note that $b^2 - ca^2$ is the norm of $ax + b$ in $R(n, c)$, and taking the norm is a multiplicative homomorphism.

Next consider the case of $n \equiv 2 \bmod 3$. Note that $ax + b = b + a(r - h)/g$. In addition, for $A, B \in \mathbf{Z}$ the mapping of $Ar + B$ to $A\zeta + B$ is a multiplicative homomorphism modulo n, since r and ζ are roots of $z^2 + z + 1$ in the rings $R(n, c)$ and $\mathbf{Z}[\zeta]$, respectively.

For proof of surjectivity, it suffices to find an element that is not mapped into 1.

Consider first the case of $n \equiv 1 \bmod 3$ and n is not a perfect cube. Assume that $n = \prod_{i=1}^{\omega} p_i^{m_i}$. Since $r^2 + r + 1 = 0 \bmod n$ it must be the case that $r_i^2 + r_i + 1 = 0 \bmod p_i$ for $r_i = r \bmod p_i$, i.e. r_i is a primitive third root of 1 in $\mathbf{Z}_{p_i}$. This is only possible if $p_i \equiv 1 \bmod 3$ for all i. Let $\pi_i = \gcd(p_i, r_i - \zeta)$. Then π_i is a prime of norm p_i in $\mathbf{Z}[\zeta]$, and $n = \prod_{i=1}^{\omega}(\pi_i\bar{\pi}_i)^{m_i}$. Let $\nu = \gcd(n, r - \zeta)$. Since $n \mid N(r - \zeta)$ and $p_i \not| (r - \zeta)$ it follows that $\nu = \prod_{i=1}^{\omega} \pi_i^{m_i}$ and $N(\nu) = n$. Since n is a not a perfect cube there is m_i with $3 \not| m_i$, and we may assume $3 \not| m_1$ without loss of generality. Finally, we are ready to present an element that is not mapped into 1 by res_3. Let $z = 0 \cdot x + b \in R(n, c)$ be chosen such that $b \equiv 1 \bmod p_i$ for all $i \geq 2$ and $b \equiv k \bmod p_1$, where k is a generator of the multiplicative group $\mathbf{Z}_{p_1}^*$. In particular, this implies that $k^{(p_1-1)/3} = r_1^{\pm 1}$ modulo p_1 (and therefore also modulo π_1). Note also that $r_1 \equiv \zeta$ modulo π_1. Using that the cubic residuosity symbol is a multiplicative homomorphism, one gets:

$$res_3(z) = [b^2 \ / \ \nu] = \prod_{i=1}^{\omega} [b^2 \ / \ \pi_i]^{m_i} = [k^2 \ / \ \pi_1]^{m_1} = [k^2 \ / \ \pi_1]^{\pm 1}$$

Using the definition of the cubic residuosity symbol, we continue the computation modulo π_1:

$$[k^2 \ / \ \pi_1] = k^{2(N(\pi_1)-1)/3} = k^{2(p_1-1)/3} = r_1^{\mp 1} = \zeta^{\mp 1} \neq 1$$

Consider next the case of n prime and $n \equiv 2 \bmod 3$. In that case n is a prime in $\mathbf{Z}[\zeta]$ of norm n^2. Let $k = ax + b$ be a generator of the multiplicative group $R(n,c)^* \cong GF(n^2)^*$. In particular, this implies that $k^{(n^2-1)/3} = r^{\pm 1}$ modulo n in $R(n,c)$, or phrased differently, $(b + a(r - h)/g)^{(n^2-1)/3} = r^{\pm 1}$. As noted under the proof of the homomorphism property, such an equality is valid modulo n in $\mathbf{Z}[\zeta]$ when substituting ζ for r, i.e. $(b+a(\zeta-h)/g)^{(n^2-1)/3} = \zeta^{\pm 1}$. Using the definition of the cubic residuosity symbol, one gets the following computation modulo n:

$$\begin{aligned} res_3(k) &= [b + a(\zeta - h)/g \ / \ n] \\ &= \ (b + a(\zeta - h)/g)^{(N(n)-1)/3} = (b + a(\zeta - h)/g)^{(n^2-1)/3} = \zeta^{\pm 1} \neq 1 \end{aligned}$$

■

5.3 EQFTwc: The algorithm

An abstract version of our revised test is presented as algorithm 2.

5.4 EQFTwc: implementation details

We need a primitive third root of unity in $R(n,c)$, since it is used for computing the cubic residuosity in line 5 (and for convenience, explicitly in line 4)

Lemma 23 *For arbitrary $n \geq 5$ with $(n,6) = 1$, for arbitrary c with $(c/n) = -1$, we may either*

- *find some $r \in R(n,c)$ with $r^2 + r + 1 = 0$ (and if $n \equiv 1 \bmod 3$ then $r \in \mathbf{Z}$), or*
- *discover that n is composite,*

using an expected time equivalent to $O(\log n)$ multiplications in $\mathbf{Z}_n$.

If $n \equiv 1 \bmod 3$, we need only use expected time equivalent to $2\log n + o(\log n)$ multiplications in $\mathbf{Z}_n$.

If $n \equiv 2 \bmod 3$ and $c = -3$ we need only use time equivalent to $O(1)$ multiplications in $\mathbf{Z}_n$.

Algorithm 2 Extended Quadratic Frobenius Test (EQFTwc).

```
*First iteration:*
Require: input is an odd number $n \geq 5$
Ensure: output is "composite", or "probable prime", $c \in \mathbf{Z}_n$, $r_{24} \in R(n,c)^*$, where $(c/n) = -1$ and $\Phi_{24}(r_{24}) = 0$.
if $n$ is divisible by 2 or 3 return "composite"
if $n$ is a perfect square or a perfect cube return "composite"
choose a small $c$ with $(c/n) = -1$
compute $r \in R(n,c)$ satisfying $r^2 + r + 1 = 0$ (may return "composite")
a: if $n \equiv 1 \bmod 3$ then select a random $z \in R(n,c)^*$ with $(N(z)/n) = -1$ and $res_3(z) \neq 1$.
b: if $n \equiv 2 \bmod 3$ then repeat
        Make a Miller-Rabin primality test on $n$ (may return "composite")
        select a random $z \in R(n,c)^*$ with $(N(z)/n) = -1$ and compute $res_3(z)$
   until either the Miller-Rabin test returns composite or the selected $z$ satisfies that $res_3(z) \neq 1$
if $\overline{z} \neq z^n$ return "composite".
Let $r_{24} = z^{(n^2-1)/24}$. If $r_{24}^8 \neq r^{\pm 1}$ or $r_{24}^{12} \neq -1$ return "composite".
return "probable prime", $c$, $r_{24}$
*Subsequent iterations:*
Require: input is $n, c, r_{24}$, where $n \geq 5$ is not divisible by 2 or 3, $(c/n) = -1$, and $\Phi_{24}(r_{24}) = 0$
Ensure: output is "composite" or "probable prime"
select random $z \in R(n,c)^*$
if $\overline{z} \neq z^n$ return "composite"
if $z^{(n^2-1)/24} \notin \{r_{24}^i \mid i = 0, \ldots, 23\}$ return "composite"
return "probable prime"
```

Proof. When n is a prime, -3, $-3/c$ are squares modulo n for $n \equiv 1, 2 \bmod 3$ respectively, and one may verify by a simple computation modulo $(n, x^2 - c)$ that

$$r = \begin{cases} -\frac{1}{2} \pm \frac{1}{2}\sqrt{-3}, & \text{if } n \equiv 1 \bmod 3 \\ -\frac{1}{2} \pm \frac{1}{2}\sqrt{\frac{-3}{c}}x, & \text{if } n \equiv 2 \bmod 3 \end{cases}$$

If n is composite the above square roots modulo n may fail to exist. We want to compute the square roots using an algorithm that is expected to terminate fast and either declares n composite or returns the wanted square root (in which case n may still be composite).

Cipolla's algorithm (see [1, ch.7]) for computing square roots in a finite field will work when n is prime. When n is composite it may still work, but if it does not, it will be clear that the compositeness is the cause. Cipolla's algorithm takes time corresponding to $O(\log n)$ multiplications over $\mathbf{Z}_n$.

To obtain the claimed better bound for $n \equiv 2 \bmod 3$ and $c = -3$, it suffices to note that the root extraction $\sqrt{-3/c}$ becomes trivial. For $n \equiv 1 \bmod 3$ it would be too slow to use Cipolla's algorithm for computing $\sqrt{-3}$. In stead we use a variant of the Cantor-Zassenhaus polynomial factorization algorithm to factor $(x^2 + 3)$ into $(x - \sqrt{-3})(x + \sqrt{-3})$. The details are given in Algorithm 3.

Algorithm 3 Compute squareroot of d modulo n

```
Require: $0 < d < n$ and $(d/n) = 1$
Ensure: $s^2 \equiv d \bmod n$ if $n$ is prime
Select arbitrary $t$ with $(t^2 - d/n) = -1$.
$f \leftarrow (x + t)^{(n-1)/2} \bmod (x^2 - d) \bmod n$.
Let $s$ be defined by $f = s^{-1}x$. (if $f$ is not a monomial of degree 1
    then $n$ is composite)
```

For proving the correctness of this algorithm, we may assume that n is a prime. Observe that by Chinese remaindering, $\mathbf{Z}[x]/(n, x^2 - d) \simeq \mathbf{Z}[x]/(n, x - \sqrt{d}) \times \mathbf{Z}[x]/(n, x + \sqrt{d}) \simeq \mathbf{Z}_n \times \mathbf{Z}_n$. The corresponding representation of $x + t \in \mathbf{Z}[x]/(n, x^2 - d)$ by Chinese remaindering is $(t + \sqrt{d}, t - \sqrt{d})$, and the representation of $(x+t)^{(n-1)/2}$ is $((t+\sqrt{d})^{(n-1)/2}, (t-\sqrt{d})^{(n-1)/2}) = ((t+\sqrt{d}/n), (t-\sqrt{d}/n)$. Since t is selected with $(t^2 - d/n) = -1$, we know that $(t + \sqrt{d}/n) = -(t - \sqrt{d}/n) = \pm 1$. But $(1, -1)$ and $(-1, 1)$ represents $(\pm\sqrt{d})^{-1}x$.

For proving the time bound, we observe that t can be selected fast since at least 1/3 of all $0 < t < n$ will have $(t^2 - d/n) \neq 1$. The remaining bottleneck is the exponentiation for computing f. But if d is small then arithmetic modulo $x^2 - d$ may be implemented analogously to our efficient implementation of arithmetic in $R(n, c)$, resulting in the exponentiation taking time corresponding to $2 \log n + o(\log n)$ multiplications in $\mathbf{Z}_n$. In particular, $d = -3$ is small in this sense. ■

Lemma 24 *For arbitrary $n \geq 5$ with $(n, 6) = 1$, for arbitrary c with $(c/n) = -1$, assume we are given an $r = gx + h \in R(n, c)$ with $r^2 + r + 1 = 0$, and if $n \equiv 1 \bmod 3$ assume in addition that $r \in \mathbf{Z}_n$, i.e. $g = 0$.*

For arbitrary $z \in R(n, c)$ we may compute $res_3(z)$ in time $O(\log^2 n)$.

Proof. By the definition of res_3, we only need efficient algorithms for computing gcd and cubic residuosity over $\mathbf{Z}[\zeta]$. Such algorithms are described and analysed in [6]. ■

Theorem 25 *There is an implementation of algorithm 2 that on input n takes expected time equivalent to at most $2 \log n + o(\log n)$ multiplications in $\mathbf{Z}_n$ per iteration, when assuming the ERH.*

The first iteration has an additional expected start up cost equivalent to at most $2 \log n + o(\log n)$ multiplications in $\mathbf{Z}_n$.

Remark. Compared to Theorem 5 the $O(u + v)$ term has disappeared from the runtime bound, but a start-up cost of $2 \log n$ has entered.

Proof. The special start up cost has different causes dependent upon whether $n \equiv 1$ or $2 \mod 3$. In the first case line 4 with construction of r is costly and in the second case line 5b running an MR test takes extra time.

The implementation of lines 1 and 2 are trivial.

For line 3, we select $c = -3$, when $n \equiv 2 \bmod 3$, in order to benefit from lemma 23. For $n \equiv 1 \bmod 3$, the implementation described in the proof of theorem 5 is used. It is expected to take $o(\log n)$ multiplications in $\mathbf{Z}_n$

For line 4, we use lemma 23. Hence line 4 is trivial when $n \equiv 2 \bmod 3$, but contributes an extra expected start up time corresponding to $2 \log n + o(\log n)$ multiplications in $\mathbf{Z}_n$, when $n \equiv 1 \bmod 3$.

For line 5a, we note that 1/3 of all $z \in R(n, c)^*$ are usable by lemmas 20 and 22, and we therefore expect to find one in 3 attempts, which takes time corresponding to $o(\log n)$ multiplications in $\mathbf{Z}_n$ by lemma 24.

For line 5b, we note that if n is composite we expect to use at most 4/3 iterations of Miller-Rabin to discover this, and if n is prime we expect to use at most 3/2 attempts to find a z with $res_3(z) \neq 1$ (by lemma 22). The bottleneck is running the Miller-Rabin algorithm, which takes time equivalent to $\log n$ multiplications in $\mathbf{Z}_n$ per run. In total, line 5b contributes an extra expected start up time corresponding to at most $3/2 \log n + o(\log n)$ multiplications in $\mathbf{Z}_n$, when $n \equiv 2 \bmod 3$.

In lines 6 and 7 we must

1. verify $z^n = \overline{z}$,
2. compute $z^{(n^2-1)/24}$,

We will argue that this takes $2 \log n + o(\log n)$ multiplications in $\mathbf{Z}_n$.

We may assume that $n = 24A + B$ for $0 < B < 24$

- compute z^A (This uses the allowed $2 \log n + o(\log n)$ multiplications in $\mathbf{Z}_n$)
- compute $z^n = z^{24A+B}$ (using $O(1)$ multiplications) and verify $z^n = \overline{z}$.
- compute $z^{(n^2-1)/24} = \overline{z^A} z^{AB+(B^2-1)/24}$. This takes only $O(1)$ multiplications and the used identity follows from $\overline{z^i} = z^{ni}$ (that is implied by $z^n = \overline{z}$). Note that $(B^2-1)/24$ is integral.

The only remaining nontrivial lines 10 and 11 are similar to lines 6 and 7. ■

5.5 EQFTwc: error analysis

The error analysis of section 3 must be reformulated.

We have earlier characterised the structure of $R(n, c)$. We need a characterisation of $H(n, c) = \{ z \in R(n, c)^* \mid z^n = \overline{z} \}$. For prime power p^m dividing n, let $H(n, p^m, c)$ denote the set of those $z_0 \in R(p^m, c)$ for which there exists $z \in H(n, c)$ satisfying that $z \equiv z_0 \bmod p^m$.

The proof of the following lemma is quite similar to the proof of lemma 9, but for completeness we include the details.

Lemma 26 *Let n be an odd number, let c be a unit modulo n.*

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

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

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

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

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

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

Consider first the case $(c/p) = -1$. By lemma 7, $\overline{z_0} = z_0^p$ implying that the order of z_0 divides $\gcd(n-p, n^2-1) = \gcd(n/p-1, p^2-1)$. Since the multiplicative subgroup of $R(p,c) \simeq GF(p^2)$ is cyclic, the stated bound on the size of $|H(n,p,c)|$ follows.

Consider next the case $(c/p) = 1$. By lemma 7, $z_0 = z_0^p$, i.e. the order of z_0 in $R(p,c)$ divides $\gcd(n^2-1, p-1) = \gcd(n^2/p^2-1, p-1)$. 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 $\gcd(n^2/p^2-1, p-1)$. In addition w_2 is uniquely determined by w_1, since by lemma 7, $(w_2,w_1) = \overline{(w_1,w_2)} = (w_1,w_2)^n = (w_1^n, w_2^n)$. Part 1 of the lemma follows.

For 2, it is enough to argue that p does not divide the order of any element $z \in H(n,p^m,c)$, since, by lemma 7, $H(n,p^m,c)$ 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, $z \in H(n,p^m,c)$ satisfy that $z^{n^2-1} = 1$, and since $p|n$ it follows that $p \not| \, n^2-1$.

For 3, we use 2 and Chinese Remaindering. ■

We may precisely state the probability of passing the conjugation tests of both the first and subsequent iterations (lines 6 and 10).

Define

$$Pr_{conj}^{first}(n,c) = Pr(\ z^n = \overline{z} \mid z \in R(n,c)^*,\ (N(z)/n) = -1,\ res_3(z) \neq 1\ )$$

$$Pr_{conj}^{subs}(n,c) = Pr(\ z^n = \overline{z} \mid z \in R(n,c)^*\ )$$

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 $(c/n) = -1$. The probability that n is not found to be composite in line 6 (respectively line 10) of algorithm 2 is*

$$
\begin{aligned}
Pr_{conj}^{first}(n,c) \leq Pr_{conj}^{subs}(n,c) &= \frac{|H(n,c)|}{|R(n,c)^*|} \\
\leq & \prod_{i=1}^{\omega} p_i^{2(1-m_i)} \mathrm{sel}[(c/p_i), \frac{(n/p_i-1, p_i^2-1)}{p_i^2-1}, \frac{(n^2/p_i^2-1, p_i-1)}{(p_i-1)^2}] \\
\leq & \prod_{i=1}^{\omega} p_i^{2(1-m_i)} \mathrm{sel}[(c/p_i), \frac{(n/p_i-1, p_i^2-1)}{p_i^2-1}, \frac{1}{p_i-1}]
\end{aligned}
$$

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. In a subsequent iteration, z is chosen randomly from $R(n,c)^*$, and it suffices to combine lemmas 7 and 26. For the first iteration, we need to argue that z being chosen with the restrictions $(N(z)/n) = -1$ and $res_3(z) \neq 1$ does not adversely effect the probability. Note that the maps $(N(\cdot)/n)$ and $res_3(\cdot)$ are group homomorphisms from $R(n,c)^*$ to $\{-1, 1\}$ and $\{\zeta, \zeta^{-1}, 1\}$ respectively. Therefore $|(N(\cdot)/n)^{-1}(-1) \cap res_3^{-1}(\zeta^{\pm 1})| = |R(n,c)^*|/3$ (when it is nonempty). But since $H(n,c)$ is a subgroup of $R(n,c)^*$ it follows that $|H(n,c) \cap (N(\cdot)/n)^{-1}(-1) \cap res_3^{-1}(\zeta^{\pm 1})| \leq |H(n,c)|/3$. ■

Next we estimate the probability of passing the 24'th-root-of-1-tests in lines 7 and 11 under the assumption that the conjugation test is passed (lines 6 and 10). The 24'th-root-of-1-test is equivalent to an 8'th-root-of-1-test combined with a 3'rd-root-of-1-test. Note that with an argument similar to the one used when analysing the EQFTac, the contributions of the 8'th-root-of-1-test and the 3'rd-root-of-1-tests are independent. For the analysis, we treat each separately, starting with the 8'th-root-of-1-test.

Define

$$
\begin{aligned}
& Pr_8^{first}(n,c) \\
&= Pr(\ z^{(n^2-1)/2} = -1 \mid z \in H(n,c),\ (N(z)/n) = -1,\ res_3(z) \neq 1\ )
\end{aligned}
$$

and

$$
Pr_8^{subs}(n,c) = Pr(\ z^{(n^2-1)/8} \in \{r_{24}^0, r_{24}^3, r_{24}^6, \ldots, r_{24}^{21}\} \mid z \in H(n,c)\ )
$$

Lemma 28 *Let n be an odd number with prime power factorisation $n = \prod_{i=1}^{\omega} p_i^{m_i}$, let c satisfy that $(c/n) = -1$.*

$$Pr_8^{first}(n, c) \leq 2^{1-\omega}$$

$$Pr_8^{subs}(n, c) \leq 8^{1-\omega}$$

Proof. let C_i be the Sylow-2 subgroup of $H(n, p_i, c)$. Then we have

$$H(n, c) \simeq C_1 \times \cdots \times C_\omega \times H_2$$

where $|H_2|$ is prime to 2. Let 2^u be the maximal power of 2 dividing $n^2 - 1$. If z is uniformly chosen in $H(n, c)$ and we write elements in $H(n, c)$ according to the above decomposition as $\omega + 1$-tuples, we have

$$z^{(n^2-1)/2^u} = (c_1, \ldots, c_\omega, 1)$$

where c_i is uniform in C_i. Let $|C_i| = 2^{a_i}$ and define $a_{min} = min\{a_i| \ i = 1, \ldots \omega\}$.

We may assume that $a_{min} \geq u$. If that was not the case, then n will definitely not pass the 8'th-root-of-1-test in the first iteration. To see this, consider the base z selected in the first iteration and suppose that z passes the tests in line 6,7. Let $z^{(n^2-1)/2^u} = (z_1, \ldots, z_\omega, 1)$ in the above decomposition, and observe that since $z^{(n^2-1)/2} = -1$, each z_i must have order 2^u, i.e. $a_i \geq u$ for all i and $a_{min} \geq u$.

We may regard any subsequent 8'th-root-of-1-test (line 11) as starting from $z^{(n^2-1)/2^u}$, performing $u-3$ squarings and testing for the occurrence of one out of 8 permitted values. Since raising to the power 2^i in the group $C_1 \times \cdots \times C_\omega$ is a $2^{\omega i}$ to 1 mapping (for $i \leq a_{min}$), it follows that

$$Pr_8^{subs}(n, c) = 8 \cdot \frac{2^{\omega(u-3)}}{\prod_{i=1}^{\omega} |C_i|} \leq 8^{1-\omega} \cdot \frac{2^{\omega u}}{\prod_{i=1}^{\omega} 2^{a_i}} \leq 8^{1-\omega}$$

A very similar argument applies to the first iteration. We make $u - 1$ squarings from $z^{(n^2-1)/2^u}$ and tests whether the result is -1. In addition, z is not chosen randomly from $H(n, c)$ but with a restriction corresponding to $(c_1, \ldots, c_\omega)$ being chosen from outside a subgroup of index 2 in the Sylow-2 subgroup. That can at most worsen the probability by a factor 2:

$$Pr_8^{first}(n, c) \leq 1 \cdot \frac{2^{\omega(u-1)}}{\prod_{i=1}^{\omega} |C_i|} \cdot 2 \leq 2^{1-\omega} \cdot \frac{2^{\omega u}}{\prod_{i=1}^{\omega} 2^{a_i}} \leq 2^{1-\omega}$$

■

The probability of passing the 3'rd-root-of-1-test may be bounded with an analogous argument.

Define

$$\begin{aligned} &Pr_3^{first}(n,c) \\ &\quad = \ Pr(\ z^{(n^2-1)/3} = r_3^{\pm 1} \mid z \in H(n,c),\ (N(z)/n) = -1,\ res_3(z) \neq 1\ ) \end{aligned}$$

and

$$Pr_3^{subs}(n,c) \ = \ Pr(\ z^{(n^2-1)/3} \in \{r_{24}^0, r_{24}^9, r_{24}^{18}\} \mid z \in H(n,c)\ )$$

Lemma 29 *Let n be an odd number with prime power factorisation $n = \prod_{i=1}^{\omega} p_i^{m_i}$, let c satisfy that $(c/n) = -1$.*

$$Pr_3^{first}(n,c) \leq 3^{1-\omega}$$

$$Pr_3^{subs}(n,c) \leq 3^{1-\omega}$$

Proof. let D_i be the Sylow-3 subgroup of $H(n, p_i, c)$. Then we have

$$H(n,c) \simeq D_1 \times \cdots \times D_\omega \times H_3$$

where $|H_3|$ is prime to 3. Let 3^v be the maximal power of 3 dividing n^2-1. If z is uniformly chosen in $H(n,c)$ and we write elements in $H(n,c)$ according to the above decomposition as $\omega+1$-tuples, we have

$$z^{(n^2-1)/3^v} = (d_1, \ldots, d_\omega, 1)$$

where d_i is uniform in D_i. Let $|D_i| = 3^{b_i}$ and define $b_{min} = min\{b_i| \ i = 1, \ldots \omega\}$.

We may assume that $b_{min} \geq v$. If that was not the case, then n will definitely not pass the 3'rd-root-of-1-test in the first iteration. To see this, consider the base z selected in the first iteration and suppose that z passes the tests in line 6,7. Let $z^{(n^2-1)/3^v} = (z_1, \ldots, z_\omega, 1)$ in the above decomposition, and observe that since $z^{(n^2-1)/3} = r_3^{\pm 1}$, each z_i must have order 3^v, i.e. $b_i \geq v$ for all i and $b_{min} \geq v$.

We may regard any subsequent 3'rd-root-of-1-test (line 11) as starting from $z^{(n^2-1)/3^v}$, raising to the power 3^{v-1}, and testing for the occurrence of one out of 3 permitted values. Since raising to the power 3^i in the group $D_1 \times \cdots \times D_\omega$ is a $3^{\omega i}$ to 1 mapping (for $i \leq b_{min}$), it follows that

$$Pr_3^{subs}(n,c) = 3 \cdot \frac{3^{\omega(v-1)}}{\prod_{i=1}^{\omega} |D_i|} \leq 3^{1-\omega} \cdot \frac{3^{\omega v}}{\prod_{i=1}^{\omega} 3^{b_i}} \leq 3^{1-\omega}$$

A very similar argument applies to the first iteration. We raise $z^{(n^2-1)/3^v}$ to the power 3^{v-1} and tests whether the result is $r_3^{\pm 1}$. In addition, z is not chosen randomly from $H(n,c)$ but with a restriction corresponding to $(d_1,\ldots,d_\omega)$ being chosen from outside a subgroup of index 3 in the Sylow-3 subgroup. That can at most worsen the probability by a factor 3/2:

$$Pr_3^{first}(n,c) \leq 2 \cdot \frac{3^{\omega(u-1)}}{\prod_{i=1}^{\omega}|C_i|} \cdot 3/2 \leq 3^{1-\omega} \cdot \frac{3^{\omega v}}{\prod_{i=1}^{\omega} 3^{b_i}} \leq 3^{1-\omega}$$

■

Theorem 30 *Let n be an odd composite number with prime power factorisation $n = \prod_{i=1}^{\omega} p_i^{m_i}$, let $\Omega = \sum_{i=0}^{\omega} m_i$.*

Let $\gamma_t(n)$ denote the probability that n passes t iterations of EQFTwc (algorithm 2). $\gamma_t(n)$ satisfies the following inequalities:

$$\begin{aligned}
\gamma_t(n) &\leq 4^{\omega-1}24^{t(1-\omega)}(\frac{|H(n,c)|}{|R(n,c)^*|})^t \\
&\leq \max_{(c/n)=-1} 4^{\omega-1}(24^{1-\omega}\prod_{i=1}^{\omega} p_i^{2(1-m_i)}\mathrm{sel}[(c/p_i), \frac{(n/p_i-1, p_i^2-1)}{p_i^2-1}, \\
&\quad \frac{(n^2/p_i^2-1, p_i-1)}{(p_i-1)^2}])^t \\
&\leq 4^{\omega-1}24^{t(1-\Omega)}
\end{aligned}$$

Proof. The theorem follows from lemmas 27, 28 and 29 combined with trivial inequalities.

■

Theorem 31 *Let n be an odd composite number. The probability that t iterations of the test of algorithm 2 result in the output "probable prime" when input n is bounded by*

$$\gamma_t(n) \leq 4^4 24^{-4t} \approx 2^{8-18.36t}$$

if either n has no prime factor ≤ 118 or $n \geq 2^{42}$.

Proof. By theorem 30, $\gamma_t(n) \leq (4 \cdot 24^{-t})^{(\Omega-1)}$. Hence, we need only consider numbers with at most 4 prime factors. By theorem 30, it suffices to prove that

$$4^{\omega-1}24^{t(1-\omega)}(\frac{|H(n,c)|}{|R(n,c)^*|})^t \leq 4^4 24^{-4t}$$

for such numbers, which is implied by

$$\frac{|H(n,c)|}{|R(n,c)^*|} \leq 24^{\omega-5}$$

To prove this inequality, we will use that $|H(n,c)|/|R(n,c)^*|$ is bounded in lemmas 35 and 36 for numbers with few prime factors.

For numbers with no small prime divisors, we consider the table of lemma 35. 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 $\leq 24^{\omega-5}$. It turns out that the bottleneck is the case $\Omega = \omega = 2$, requiring $p > 118$.

For large n, we analogously consider the table of lemma 36. Here the bottleneck is the case $\Omega = \omega = 4$, requiring $n > 2^{42}$. ■

5.6 Error probability without the 24'th-root-of-1-test

For composite numbers with at most 4 prime factors, it is possible to get good bounds on the error probability $|H(n,c)|/|R(n,c)^*|$ of the basic Frobenius test (line 6 or line 10 of algorithm 2) alone, i.e. omitting the 24'th-root-of-1-test.

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 5.5 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 [7]. We give a different proof and a slightly sharper result in lemma 33. 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 31.

5.6.1 Technical lemmas

Lemma 32 *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 $(c/n) = -1$.*

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

Proof. This follows from lemma 27. ■

Lemma 33 *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*

$$1 < \frac{\prod_{i=1}^{3}(n/p_i - 1)}{\prod_{i=1}^{3}(p_i^2 - 1)} < 1 + \frac{1}{p_1^2 - 1} \tag{14}$$

and

$$\prod_{i=1}^{3} \frac{\gcd(n/p_i - 1, p_i^2 - 1)}{p_i^2 - 1} < \frac{1}{p_1^2 - 1} \tag{15}$$

Proof. We start by proving (14). Define

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

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

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

and

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

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

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

and

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

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

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

which proves the lemma. ■

Lemma 34 *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*

$$p_1^3 < \frac{\prod_{i=2}^{4} (n/p_i - 1)}{\prod_{i=2}^{4} (p_i^2 - 1)} < p_1^3 (1 + \frac{3}{p_2^2 - 1}) \tag{16}$$

and

$$\prod_{i=2}^{4} \frac{\gcd(n/p_i - 1, p_i^2 - 1)}{p_i^2 - 1} < \frac{3 p_1^3}{p_2^2 - 1} \tag{17}$$

Proof. We start by proving the upper bound of (16).

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

The lower bound is implied by (14):

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

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. ■

5.6.2 Worst case bound parametrised by smallest prime factor

Lemma 35 *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 $|H(n,c)|/|R(n,c)^*|$ is bounded by the entries of the following table

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

Proof. All the entries with $\Omega > \omega$ are consequences of lemma 32. 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 32. For the entry with $\Omega = \omega = 2$, we have $n = p_1p_2$, and without loss of generality, we may assume that $(c/p_1) = -(c/p_2) = -1$. By lemma 32, we have

$$\frac{|H(n,c)|}{|R(n,c)^*|} \;\le\; \frac{\gcd(p_2 - 1, p_1^2 - 1)}{p_1^2 - 1} \cdot \frac{\gcd(p_1^2 - 1, p_2 - 1)}{(p_2 - 1)^2} \;\le\; \frac{1}{p_1^2 - 1}$$

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 32,

$$\frac{|H(n,c)|}{|R(n,c)^*|} \leq \frac{1}{p_2-1} \cdot \frac{1}{p_3-1} \leq \frac{1}{p^2-1}$$

In the former case, $(c/p_i) = -1$ for all i, and by the inequality of lemma 33, we have

$$\frac{|H(n,c)|}{|R(n,c)^*|} \leq \frac{1}{p^2-1}$$

■

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

Lemma 36 *Let n be an odd composite number having complete prime factorisation $n = \prod_{i=1}^{\omega} p_i^{m_i}$, let $\Omega = \sum_{i=1}^{\omega} m_i \leq 4$, and let c satisfy that $(c/n) = -1$. $|H(n,c)|/|R(n,c)^*|$ is bounded by the entries of the following table*

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

Proof. The entries for $\omega = 1$ are a consequence of lemma 32, 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 32, and for $n = p_1p_2p_3$ and $n = p_1p_2p_3p_4$, we need additional technical results (lemma 33 and lemma 34).

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,

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

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

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

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 34 and the part of the previous argument that holds also when $p_1 < p_2$, we find

$$\begin{aligned} \frac{|H(n,c)|}{|R(n,c)^*|} &\leq \frac{1}{p_1-1} \cdot \min\{1, \frac{p_1p_2p_4-1}{p_3^2-1} \cdot \frac{p_1p_2p_3-1}{p_4^2-1}, \frac{3p_1^3}{p_2^2-1}\} \\ &\leq 2n^{-2/15} \end{aligned}$$

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 $(c/p_1) = -1$:

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

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 33,

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

Case $n = p_1^2p_2p_3$:

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

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

Case $n = p_1p_2$:

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).

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

Case $n = p_1^2p_2$:

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

$$\frac{|H(n,c)|}{|R(n,c)^*|} \;\leq\; p_1^{-2} \cdot \min\{1, \frac{p_1^2-1}{p_2^2-1}\} \;\leq\; n^{-2/3}$$

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,

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

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

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

■

References

[1] Eric Bach and Jeffrey Shallit. *Algorithmic number theory. Vol. 1.* Foundations of Computing Series. MIT Press, Cambridge, MA, 1996. Efficient algorithms.

[2] Jørgen Brandt and Ivan Damgård. On generation of probable primes by incremental search. In *Advances in cryptology—CRYPTO '92 (Santa Barbara, CA, 1992)*, Vol. 740 of *Lecture Notes in Comput. Sci.*, pp. 358–370. Springer, Berlin, 1993.

[3] Jørgen Brandt, Ivan Damgård, and Peter Landrock. Speeding up prime number generation. In *Advances in cryptology—ASIACRYPT '91 (Fujiyoshida, 1991)*, Vol. 739 of *Lecture Notes in Comput. Sci.*, pp. 440–449. Springer, Berlin, 1993.

[4] Ronald Joseph Burthe, Jr. Further investigations with the strong probable prime test. *Math. Comp.* **65**(213) (1996), 373–381.

[5] Ivan Damgård, Peter Landrock, and Carl Pomerance. Average case error estimates for the strong probable prime test. *Math. Comp.* **61**(203) (1993), 177–194.

[6] Ivan B. Damgård and Gudmund Skovbjerg Frandsen. Efficient algorithms for gcd and cubic residuosity in the ring of Eisenstein integers. Research Series RS-03-8, BRICS, Department of Computer Science, University of Aarhus, February 2003.

[7] Jon Grantham. A probable prime test with high confidence. *J. Number Theory* **72**(1) (1998), 32–47.

[8] Siguna Müller. A probable prime test with very high confidence for $n \equiv 1 \bmod 4$. In *Advances in cryptology—ASIACRYPT 2001 (Gold Coast)*, Vol. 2248 of *Lecture Notes in Comput. Sci.*, pp. 87–106. Springer, Berlin, 2001.

[9] Renate Scheidler and Hugh C. Williams. A public-key cryptosystem utilizing cyclotomic fields. *Des. Codes Cryptogr.* **6**(2) (1995), 117–131.

Recent BRICS Report Series Publications

RS-03-9 Ivan B. Damgård and Gudmund Skovbjerg Frandsen. *An Extended Quadratic Frobenius Primality Test with Average and Worst Case Error Estimates*. February 2003. 53 pp.

RS-03-8 Ivan B. Damgård and Gudmund Skovbjerg Frandsen. *Efficient Algorithms for gcd and Cubic Residuosity in the Ring of Eisenstein Integers*. February 2003. 11 pp.

RS-03-7 Claus Brabrand, Michael I. Schwartzbach, and Mads Vanggaard. *The METAFRONT System: Extensible Parsing and Transformation*. February 2003. 24 pp.

RS-03-6 Giuseppe Milicia and Vladimiro Sassone. *Jeeg: Temporal Constraints for the Synchronization of Concurrent Objects*. February 2003. 41 pp. Short version appears in Fox and Getov, editors, *Joint ACM-ISCOPE Conference on Java Grande*, JGI '02 Proceedings, 2002, pages 212–221.

RS-03-5 Aske Simon Christensen, Anders Møller, and Michael I. Schwartzbach. *Precise Analysis of String Expressions*. February 2003. 15 pp.

RS-03-4 Marco Carbone and Mogens Nielsen. *Towards a Formal Model for Trust*. January 2003.

RS-03-3 Claude Crépeau, Paul Dumais, Dominic Mayers, and Louis Salvail. *On the Computational Collapse of Quantum Information*. January 2003. 31 pp.

RS-03-2 Olivier Danvy and Pablo E. Martínez López. *Tagging, Encoding, and Jones Optimality*. January 2003. To appear in Degano, editor, *Programming Languages and Systems: Twelfth European Symposium on Programming*, ESOP '03 Proceedings, LNCS, 2003.

RS-03-1 Vladimiro Sassone and Pawel Sobocinski. *Deriving Bisimulation Congruences: 2-Categories vs. Precategories*. January 2003. To appear in Gordon, editor, *Foundations of Software Science and Computation Structures*, FoSSaCS '03 Proceedings, LNCS, 2003.