Joachim von zur Gathen & Igor Shparlinski (2013). Generating safe primes. *Journal of Mathematical Cryptology* **7**(4), 333–365. ISSN 1862-2984 (Online) 1862-2976 (Print). URL https://dx.doi.org/10.1515/jmc-2013-5011.

Generating safe primes

Joachim von zur Gathen
B-IT, Universität Bonn
53113 Bonn, Germany
gathen@bit.uni-bonn.de

Igor E. Shparlinski
Department of Computing
Macquarie University
NSW 2109 , Australia
igor.shparlinski@unsw.edu.au

September 6, 2013

Abstract

Safe primes and safe RSA moduli are used in several cryptographic schemes. The most common notion is that of a prime p, where $(p-1)/2$ is also prime. The latter is then a *Sophie Germain prime*. Under appropriate heuristics, they exist in abundance and can be generated efficiently. But the modern methods of analytic number theory have—so far—not even allowed to prove that there are infinitely many of them. Also for other notions of safe primes, there is no algorithm in the literature that is unconditionally proven to terminate, let alone to be efficient.

This paper considers a different notion of safe primes and moduli. They can be generated in polynomial time, without any unproven assumptions, and are good enough for the cryptographic applications that we are aware of.

1 Introduction

There are various notions of *safe primes* p and *safe RSA moduli* $N = pq$, where p and q are safe primes, in the cryptographic literature. Two standard conditions on an integer (greater than 1) are y-smooth (with all prime factors at most y) and y-rough (with all prime factors greater than y). The condition "y-rough" implies "not y-smooth", but not conversely. A Sophie Germain prime ℓ is such that $p = 2\ell + 1$ is also prime. In order to describe the safe primes, we call a product ℓ of exactly $i \geq 1$ distinct primes a *Sophie Germain$_i$ integer* if $p = 2\ell + 1$ is prime, and denote by SG_i the set of the latter primes. Thus the Sophie Germain primes are precisely the Sophie Germain$_1$ integers. (We feel bound by the traditional nomenclature, although for our purposes it would be nicer to have a catchy name for the primes in SG_i.) We also abbreviate $\mathrm{SG}_{\leq 2} = \mathrm{SG}_1 \cup \mathrm{SG}_2$, the set of primes p with $(p-1)/2$ having either one or two prime factors. The set SG_1 appears in several cryptographic protocols, but it is still unknown whether it is infinite—although there is no reason to doubt this. It seems unlikely that this long-standing open problem will be resolved in the near future.

Our safe primes are some special primes in $\mathrm{SG}_{\leq 2}$. The purpose of this paper is to show that they

- exist in abundance,
- can be effectively generated,
- can be successfully used instead of SG_1 primes in many protocols.

In order to place this in context, we assemble four properties of primes p from the literature in the following table, where $\ell = (p-1)/2$.

Sophie Germain	ℓ prime
Sophie Germain$_{\leq 2}$	ℓ is prime or product of two primes
rough	ℓ is a product of large primes
not smooth	ℓ has a large prime divisor

We have not quantified the four notions, and at this point have the implications

$$\text{Sophie Germain} \Longrightarrow \text{Sophie Germain}_{\leq 2} \Longrightarrow \text{rough} \Longrightarrow \text{not smooth}.$$

We now add more detail. The most stringent notion asks for k to be a *Sophie Germain prime*. This is also the most common one in the cryptographic literature. It is put forth in Menezes *et al.* (1997), Section 4.6.1, and Galbraith (2012), and used in the Wikipedia entry on "safe primes", in Shoup (2000) and in Hofheinz *et al.* (2012). Furthermore, Naccache (2003) also uses this notion, but assumes incorrect heuristics for the probability of random p and $(p-1)/2$ to be prime. Damgård & Koprowski (2001) first appeal to this notion, but say, correctly, on their page 153 that "we do not even know if there are infinitely many safe primes". Indeed, it is conjectured that there are about $c\,x/\ln^2 x$ Sophie Germain primes up to x, for some explicit constant c; see Conjecture 14 in Section 6. If the conjecture is true, then one can efficiently generate such primes. This works quite well in practice. However, the currently available methods of analytic number theory have not even allowed to show that there are infinitely many Sophie Germain primes. Thus no cryptosystem assuming an infinite supply of them can be proven unconditionally to work.

As usual, we assume a security parameter n. Then "large" usually means exponential in n, for example primes with about $n/2$ bits, and moduli with about n bits. "Small" means polynomial in n, that is, with $O(\log n)$ bits.

The Sophie Germain$_{\leq 2}$ integers are well-known in number theory, but have not turned up in cryptography. We show in this paper that they provide a useful concept: it can be efficiently sampled, and is good enough for the cryptographic applications that we are aware of. Obviously, any Sophie Germain prime satisfies this condition.

In the third condition, y-rough is used with a small y, in the sense above. The signature scheme of Gennaro *et al.* (1999) works with this notion, as does the distributed moduli generation of Damgård & Koprowski (2001) and of Fouque & Stern (2001); they assume in the proof of their Theorem 2 that primes are uniformly distributed. The roughness condition is also employed in Joye & Paillier (2006), Damgård & Koprowski (2001) and Nishide & Sakurai (2011), where y is the number of players in a certain multiparty computation, and Ong & Kubiatowicz (2005), Section 3.5, say that "safe primes are unfortunately less dense than unrestricted primes", but no lower bound is attempted in this experimental paper. Joye & Paillier (2006) give a heuristic improvement for this task by sieving modulo the product of small primes greater than 2 and making sure that $(p-1)/2$ has no small factor by p being a nonsquare modulo each small prime. Clearly the second notion implies this one, even with a value for y that may be as "large" as is allowed in the second one.

The fourth notion requires that k have at least one "large" prime factor. This used to be required for RSA moduli in some scenarios, in order to resist certain factorization methods. It follows from the third notion when the smoothness and roughness parameters are chosen identically, but is now considered obsolete. A result of Baker & Harman (1998) implies that the set of primes p for which $p-1$ is not $p^{0.677}$-smooth is of positive relative density in the set of all primes.

Now take some $p \in \mathrm{SG}_{\leq 2}$ with only large prime factors in $(p-1)/2$, and a different prime q with the same property. Then the modulus $N = pq$ enjoys the following properties:

- N is a Blum integer,
- $\phi(N)/4$ has only large prime factors,
- the square of a random element in the residue ring $\mathbb{Z}_N$ is a generator of the subgroup of squares in the unit group $\mathbb{Z}_N^\times$ with probability exponentially close to 1.

For the Sophie Germain$_{\leq 2}$ integers, the topic of this paper, a quantified version of these properties is proven in Theorem 11 and Corollary 12. Algorithmically, the crux is to show that a certain rejection sampling process producing such integers works in polynomial time, that is, the primes we generate form an inverse polynomial fraction of all integers up to some bound. Our main technical tool is a result of Heath-Brown (1986).

Our interest in this theme has been spawned by the cryptosystem of Hofheinz *et al.* (2012), whose breaking (in the sense of IND-CCA2 security) is equivalent to factoring the modulus N. This reduction is free of any unproved hypotheses. They present two versions. One of them uses the Goldreich-Levin predicate. The other one, simpler and more natural, takes the modulus as $N = (2k_1 + 1)(2k_2 + 1)$, where k_1 and k_2 are two distinct Sophie Germain primes. As noted above, this works well heuristically, but no proof for it is in sight. We show here that modern analytic number theory is still powerful enough to close the only remaining gap in the argument of Hofheinz *et al.* (2012), and thus make it fully rigorous.

In the following, the letters ℓ, p, and q with or without subscripts denote prime numbers, and for real $x < y$, we use $[x \mathrel{..} y]$ and $[x \mathrel{..} y]_{\mathbb{R}}$ to denote the sets of integers and reals between x and y, respectively. We also use

$(x \mathinner{..} y]$ and $(x \mathinner{..} y]_{\mathbb{R}}$ in similar meanings. All our random choices are made uniformly from finite sets, unless explicitly stated otherwise. We use the big-Oh notation and its relatives as sets, so that

$$O(g) = \{f \colon \exists c \ |f(x)| \le c \cdot g(x) \text{ for sufficiently large } x\},$$

where f and g are real functions, and pointwise operators, so that, for example, $g + O(h) = \{g + f \colon f \in O(h)\}$.

Section 2 through 5 build up the required algorithmic machinery in several steps. Just after Algorithm 5, we explain why a naive sampling method fails and something like the approach of Section 2 is needed; see the end of Section 7 for a simpler but less efficient version. Sections 6 and 7 present extensions and variations of our method, heuristics for the number of Sophie Germain$_{\le 2}$ integers, and, assuming these heuristics, more precise information about the runtime of our algorithms.

2 Sampling the harmonic distribution

Our goal in this section is to sample the *harmonic distribution* which gives to an integer k in some finite interval a probability proportional to $1/k$. This yields an approximately uniform sampling of integer points under a hyperbola in Algorithm 3, which in turn leads to the random unbalanced moduli in Algorithm 5.

We first recall the *inversion method* for the continuous version $\mathcal{D}^*$ of this distribution; see Knuth (1981), Section 3.4.1, and Devroye (1986), Section III.2.2 B. We have two positive real numbers $A < B$, set

$$a^* = \frac{1}{\ln(B/A)}, \tag{1}$$

and take the continuous distribution $\mathcal{D}^*$ on $(A \mathinner{..} B]_{\mathbb{R}}$ with (cumulative) density function

$$F^*(x) = \begin{cases} 0 & \text{if } x < A, \\ a^* \ln(x/A) & \text{if } A \le x \le B, \\ 1 & \text{if } x > B. \end{cases}$$

Thus, choosing x according to $\mathcal{D}^*$ is equivalent to $\text{prob}(x \le y) = F^*(y)$ for all $y \in (A \mathinner{..} B]_{\mathbb{R}}$. On $(A \mathinner{..} B]_{\mathbb{R}}$, F^* takes values between 0 and 1, increases strictly monotonically, and its functional inverse is

$$G^*(x) = Ae^{x/a^*}.$$

In particular, $G^*(x) \in (A \mathinner{..} B]$ for $x \in (0 \mathinner{..} 1]_\mathbb{R}$.

The inversion method takes a sample u from the uniform real distribution on $(0 \mathinner{..} 1]_\mathbb{R}$ and sets $x = G^*(u)$. Then for any $y \in (A \mathinner{..} B]_\mathbb{R}$, we have

$$\text{prob } (x \le y) = \text{prob } (G^*(u) \le y) = \text{prob } (u \le F^*(y)) = F^*(y).$$

Thus x samples $\mathcal{D}^*$.

In several steps, we now transform this method into a discrete algorithm that approximately samples the harmonic distribution. In a first step, we take positive real numbers $A < B$, the harmonic number

$$H_n = \sum_{1 \le k \le n} 1/k$$

for an integer n, set

$$a = \frac{1}{H_{\lfloor B \rfloor} - H_{\lfloor A \rfloor}} \tag{2}$$

and consider the discrete harmonic distribution $\mathcal{D}$ with

$$\text{prob}_\mathcal{D}(k) = \frac{a}{k}$$

on the integers $k \in (A \mathinner{..} B]$. We use the Euler-Mascheroni constant $\gamma \approx 0.57721$ and the bounds

$$0 < H_n - (\ln n + \gamma) < \frac{1}{2n}; \tag{3}$$

see Guo & Qi (2011) for sharper estimates which imply (3).

The value H_A has a standard definition also for non-integral A. In order to avoid confusion, we write $H(A)$ for $H_{\lfloor A \rfloor}$ in the following. For our rounded arguments, we use $|\ln(1+z) - z| \le z^2$ if $|z| \le 1/2$, and for $A \ge 4$

$$\begin{aligned} |H(A) - (\ln A + \gamma)| &= |H_{\lfloor A \rfloor} - \ln A + \gamma| \\ &\le |H_{\lfloor A \rfloor} - (\ln \lfloor A \rfloor + \gamma)| + |\ln \frac{\lfloor A \rfloor}{A}| \\ &\le \frac{1}{2\lfloor A \rfloor} + \left|\ln\left(1 - \frac{A - \lfloor A \rfloor}{A}\right)\right| \le \frac{1}{2\lfloor A \rfloor} + \frac{1}{A^2} < \frac{1}{A}. \end{aligned}$$

For $4 \le A < B$, we have

$$\begin{aligned} |H(B) - H(A) - \ln(B/A)| &= |H(B) - (\ln B + \gamma) - (H(A) - (\ln A + \gamma))| \\ &< B^{-1} + A^{-1} < 2A^{-1}. \end{aligned}$$

If also

$$\ln(B/A) \ge A^{-1} + (1 + A^{-2})^{1/2}, \tag{4}$$

then $(\ln(B/A) - 2A^{-1}) \cdot \ln(B/A) \ge 1$ and

$$\begin{aligned} |a - a^*| &= |\frac{1}{H(B) - H(A)} - \frac{1}{\ln(B/A)}| \\ &= \frac{|\ln(B/A) - (H(B) - H(A))|}{|H(B) - H(A)| \cdot \ln(B/A)} \\ &< \frac{2A^{-1}}{|\ln(B/A) - 2A^{-1}| \cdot \ln(B/A)} \le 2A^{-1}. \end{aligned} \tag{5}$$

One can check that the densities of $\mathcal{D}^*$ on integers and $\mathcal{D}$ agree closely, but we do not need this here.

We now consider the inversion method, where $u \in (0 \mathinner{..} 1]_{\mathbb{R}}$ is chosen uniformly at random and the integer

$$t^*(u) = \lfloor G^*(u) \rfloor = \lfloor A e^{u/a^*} \rfloor$$

is produced. Then $A \le t^*(u) \le B$. One can check that any $k \in [A \mathinner{..} B]$ is returned with probability $a^* \ln(1 + 1/k) \sim a^*/k \cdot (1 + O(A^{-1}))$.

In the next step, we replace u by a discrete approximation and again round $G^*(u)$ down. So we have a (large) positive integer M, choose an integer $v \in (0 \mathinner{..} M]$ uniformly at random, so that v/M is an approximation of u as above, and produce

$$t(v) = \lfloor G^*(v/M) \rfloor = \lfloor A e^{v/a^* M} \rfloor. \tag{6}$$

For $k \in (A \mathinner{..} B]$, we let

$$V = \{v \in (0 \mathinner{..} M] \colon t(v) = k\} = t^{-1}(k),$$

so that $b_1(k) = \#V/M$ is the probability with which $k = t(v)$. We now claim that $b_1(k)$ is close to a^*/k, and provide an error estimate for the approximation quality. For real $x < y$, the number of integers in $(x \mathinner{..} y]$ satisfies

$$|\#(x \mathinner{..} y] - (x - y)| \le 1. \tag{7}$$

Using $F^*(k+1) - F^*(k) = a^* \ln(1+k^{-1})$, we have

$$\begin{aligned} v \in V &\iff k \le G^*(\frac{v}{M}) < k+1 \\ &\iff F^*(k) \le \frac{v}{M} < F^*(k+1) \\ &\iff MF^*(k) \le v < MF^*(k+1). \end{aligned}$$

Therefore

$$|\#V - a^* M \ln(1+k^{-1})| \le 1.$$

For $k \ge 2$ we find

$$\begin{aligned} \left| b_1(k) - \frac{a^*}{k} \right| &= \left| \frac{\#V}{M} - \frac{a^*}{k} \right| \\ &\le \left| \frac{\#V}{M} - a^* \ln(1+\frac{1}{k}) \right| + \left| a^* \ln(1+\frac{1}{k}) - \frac{a^*}{k} \right| \\ &\le \frac{1}{M} + \frac{a^*}{k^2} = \frac{a^*}{k} \left(\frac{1}{k} + \frac{k}{a^* M} \right) \le \frac{a^*}{k} \left(\frac{1}{A} + \frac{B}{a^* M} \right). \end{aligned} \tag{8}$$

None of the methods sketched so far can be literally implemented on a computer, since we cannot compute a real number like $e^{v/aM}$ exactly. So we now consider floating-point computations with real numbers using m_0 bits of precision. We take some n so that all quantities in the algorithm are absolutely at most 2^n. We assume $m_0 > n$ and set $m = m_0 - n$. Then if a real value r is to be computed, the algorithm computes some $\widetilde{r}$ with $|r - \widetilde{r}| < 2^{-m} = \varepsilon$, which we call an m-bit approximation (which in standard parlance is an m_0-bit approximation). We assume some standard representation where $\lfloor \widetilde{r} \rfloor$ can be computed exactly, by truncating after the decimal (or binary) point. This leads to the following algorithm.

Algorithm 1. *Sampling the harmonic distribution.*

INPUT: *Positive real numbers A and B with $4 \le A < B$, and positive integers M and m.*

OUTPUT: *An integer in $[A \,..\, B]$.*

1. *Choose an integer $v \in [1 \,..\, M]$ uniformly at random.*
2. *Calculate an m-bit approximation $\widetilde{s}$ to $Ae^{v \ln(B/A)/M}$.*

3. Return $\lfloor \widetilde{s} \rfloor$*.*

Theorem 2. *We assume that* $B < M < 2^n$ *and*

$$m \geq \log_2 \left(\frac{M}{A \ln(B/A)} \right), \tag{9}$$

and we calculate numerically with precision $m + n$*.*

(i) Let $k \in (A \,..\, B]$ *and*

$$\delta_1 = \frac{a^* + 2}{a^* A} + \frac{3B}{a^* M} + \frac{1}{2^{m-2}}, \tag{10}$$

and assume that (4) *holds. Then the probability* $b_2(k)$ *that* k *is returned by the algorithm satisfies*

$$|b_2(k) - \frac{a}{k}| \leq \frac{a^* \delta_1}{k}.$$

(ii) Any output of Algorithm 1 is in $[A \,..\, B]$ *and the algorithm uses time polynomial in* $m + n$*.*

Proof. For $1 \leq v \leq M$, we write $s(v) = A(B/A)^{v/M} = Ae^{v/a^*M} = G^*(v/c)$, $t(v) = \lfloor s(v) \rfloor$ as in (6), and $\widetilde{s}(v)$ for the approximation to $s(v)$ calculated in step 2. Since rounding down is exact, $\widetilde{t}(v) = \lfloor \widetilde{s}(v) \rfloor$ is returned in step 3.

(ii): By assumption, we have $A/a^*M \geq \varepsilon = 2^{-m}$. For any $v \geq 1$, we have

$$\widetilde{s}(v) \geq s(v) - \varepsilon \geq s(1) - \varepsilon = Ae^{1/a^*M} - \varepsilon \geq A(1 + \frac{1}{a^*M}) - \varepsilon \geq A$$

and $\widetilde{t}(v) \geq A$. For any $v \leq M$, we have

$$\widetilde{s}(v) \leq s(v) + \varepsilon \leq s(M) + \varepsilon = Ae^{\ln(B/A)} + \varepsilon = B + \varepsilon$$

and $\widetilde{t}(v) \leq B$. It is well known how to compute numerically the required approximations to $\ln(B/A)$, $e^{v \ln(B/A)/M} = (B/A)^{v/M}$, and $s(v)$ in time polynomial in $m + n$; see Brent & Zimmermann (2011), Sections 4.2.5 and 4.4, for some details.

(i): We take some $k \in (A \,..\, B]$ and want to show that $b_2(k)$ is close to a/k. We define the five real intervals

$$\begin{aligned}
S &= [k \,..\, k+1)_{\mathbb{R}}, \\
S_{0,+} &= [k \,..\, k+\varepsilon)_{\mathbb{R}}, \\
S_{0,-} &= [k-\varepsilon \,..\, k)_{\mathbb{R}}, \\
S_{1,+} &= [k+1 \,..\, k+1+\varepsilon)_{\mathbb{R}} \\
S_{1,-} &= [k+1-\varepsilon \,..\, k+1)_{\mathbb{R}}.
\end{aligned}$$

We take $t^{-1}(k) = \{v \in [1 \,..\, M] \colon t(v) = k\}$, and similarly for $\widetilde{t}$. Then

$$\begin{gathered}
t^{-1}(k) = s^{-1}(S), \\
s^{-1}(S \smallsetminus (S_{0,+} \cup S_{1,-})) \subseteq \widetilde{t}^{-1}(k) \subseteq s^{-1}(S \cup S_{0,-} \cup S_{1,+}).
\end{gathered} \tag{11}$$

We start by considering the case $s(v) \in S_{0,+}$. Then $s(v) = k + \gamma_0$ with $0 \le \gamma_0 < \varepsilon$. Setting $\gamma_1 = \gamma_0/k$, we have $0 \le \gamma_1 < \varepsilon/k < 1/2$ and $G^*(v/M) = s(v) = k(1+\gamma_1)$. Furthermore,

$$\frac{v}{M} = F^* \circ G^* \left(\frac{v}{M}\right) = F^*(k(1+\gamma_1)) = F^*(k) + a^* \ln(1+\gamma_1)$$

and

$$0 \le \ln(1+\gamma_1) \le 2\gamma_1 < \frac{2\varepsilon}{k}.$$

Therefore

$$v = MF^*(k) + a^*M \ln(1+\gamma_1) \in \left[MF^*(k) \,..\, MF^*(k) + \frac{2\varepsilon a^* M}{k}\right)$$

which in turn implies that

$$\# s^{-1}(S_{0,+}) \le \frac{2\varepsilon a^* M}{k} + 1.$$

One finds the same bound for $\# s^{-1}(S_{1,-})$, $\# s^{-1}(S_{0,-})$, and $\# s^{-1}(S_{1,+})$. It now follows that

$$\# s^{-1}(S_{0,+} \cup S_{1,-}), \# s^{-1}(S_{0,-} \cup S_{1,+}) \le \frac{4\varepsilon a^* M}{k} + 2.$$

Let $\mathcal{A}$ be the algorithm described by (6), which is just the exact version of Algorithm 1, with $s(v)$ in step 2 calculated exactly and output $t(v)$. Now

$\mathcal{A}$ works well and returns k with probability $b_1(k)$ close to a^*/k, namely satisfying (8). By (11), this happens if and only if $s(v) \in S$, so that

$$\left|\frac{\#s^{-1}(S)}{M} - \frac{a^*}{k}\right| = \left|b_1(k) - \frac{a^*}{k}\right| \leq \frac{a^*}{k}\left(\frac{1}{A} + \frac{B}{a^*M}\right),$$

Moreover,

$$\#s^{-1}(S) - \left(\frac{4\varepsilon a^* M}{k} + 2\right) \leq \#s^{-1}(S \setminus (S_{0,+} \cup S_{1,-})) \leq \#\widetilde{t}^{-1}(k)$$
$$\leq s^{-1}(S \cup S_{0,-} \cup S_{1,+}) \leq \#s^{-1}(S) + \frac{4\varepsilon a^* M}{k} + 2.$$

Using (5), it follows that

$$\begin{aligned}
\left|b_2(k) - \frac{a}{k}\right| &\leq \left|\frac{\#\widetilde{t}^{-1}(k)}{M} - \frac{a^*}{k}\right| + \left|\frac{a^* - a}{k}\right| \\
&\leq \left|\frac{\#\widetilde{t}^{-1}(k) - \#s^{-1}(S)}{M}\right| + \left|\frac{\#s^{-1}(S)}{M} - \frac{a^*}{k}\right| + \frac{2}{Ak} \\
&\leq \frac{4\varepsilon a^*}{k} + \frac{2}{M} + \frac{a^*}{k}\left(\frac{1}{A} + \frac{B}{a^*M}\right) + \frac{2}{Ak} \\
&\leq \frac{a^*}{k}\left(4\varepsilon + \frac{2B}{a^*M} + \frac{1}{A} + \frac{B}{a^*M} + \frac{2}{a^*A}\right) = \frac{a^*\delta_1}{k}.
\end{aligned}$$

□

Using the penultimate rather than the last bound in (8), we can replace the summand $\sigma = 1/A + 3B/a^*M$ by $\sigma_k = 1/k + 3k/a^*M$. As a function of a real variable k on $(A \,..\, B]_{\mathbb{R}}$, σ_k is convex, and σ can be replaced by $\max\{\sigma_A, \sigma_B\}$. This equals σ_B if and only if $a^*M \geq 3AB$.

We now present a method to generate almost uniformly random pairs of positive integers under a hyperbola $xy = D$.

Algorithm 3. *Random integers under a hyperbola.*

> INPUT: *Positive real numbers* $8 \leq A < B < C < D < 2^n$, *and integers* M *and* m *greater than* 1.
>
> OUTPUT: *A pair* (k_1, k_2) *of integers with* $k_1 \in [A \,..\, B]$ *and* $k_1k_2 \in [C \,..\, D]$.

1. *Call Algorithm 1 with inputs A, B, M, and m, and output k_1, calculating numerically with precision $m+n$.*

2. *Choose a uniformly random integer $k_2 \in [C/k_1 \mathrel{..} D/k_1]$.*

3. *Return (k_1, k_2).*

For any pair (k_1, k_2) of integers, we let $b_3(k_1, k_2)$ be the probability with which it is returned by the algorithm, and write

$$K = \{(k_1, k_2) \colon k_1 \in [A \mathrel{..} B],\ k_1 k_2 \in [C \mathrel{..} D]\}. \tag{12}$$

For $A = C = 1$ and $B = D$, $\#K = \sum_{k \le D} \tau(k)$ equals $2D$ times the average value of Dirichlet's divisor function τ on $[1 \mathrel{..} D]$. In our application, $B \approx D^{1/2}$ is much smaller than D.

Theorem 4. *We assume that* (4) *and* (9) *hold, a is as in* (2), *δ_1 as in* (10), *and set*

$$\delta_2 = \frac{(a+1)B + 2\delta_1(D - C + aB)}{D - C - B}. \tag{13}$$

(i) Any output of Algorithm 3 satisfies the output specification and for any $(k_1, k_2) \in K$ we have

$$\left| b_3(k_1, k_2) - \frac{1}{\#K} \right| < \frac{\delta_2}{\#K}.$$

(ii) The algorithm uses time polynomial in m, n, and $\ln M$.

Proof. For $k_1 \in [A \mathrel{..} B]$, the number $d(k_1)$ of choices in step 2 satisfies by (7)

$$\left| d(k_1) - \frac{D-C}{k_1} \right| \le 1,$$

It follows that

$$d(k_1) \ge \frac{D-C}{k_1} - 1 \ge \frac{D-C-B}{k_1}.$$

Hence

$$\begin{aligned} \left|\frac{1}{d(k_1)} - \frac{k_1}{D-C}\right| &= \left|\frac{D - C - k_1 d(k_1)}{d(k_1)(D-C)}\right| \\ &\le \frac{k_1}{d(k_1)(D-C)} \le \frac{k_1^2}{(D-C)(D-C-B)}. \end{aligned}$$

Furthermore,

$$\begin{aligned}\left|\#K - \frac{D-C}{a}\right| &= \Big|\sum_{A<k_1\le B} d(k_1) - (D-C)(H(B)-H(A))\Big| \\ &= \Big|\sum_{A<k_1\le B}\big(d(k_1) - \frac{D-C}{k_1}\big)\Big| \le \sum_{A<k_1\le B} 1 < B.\end{aligned} \tag{14}$$

Since $b_3(k_1,k_2) = b_2(k_1)/d(k_1)$, we have

$$\begin{aligned}\#K{\cdot}b_3(k_1,k_2) - 1 &= b_2(k_1)\cdot\frac{\#K}{d(k_1)} - 1 \\ &= \frac{a}{k_1}\cdot\frac{D-C}{a}\cdot\frac{k_1}{D-C} + \frac{(b_2(k_1)-a/k_1)\#K}{d(k_1)} + \frac{a(\#K-(D-C)/a)}{k_1 d(k_1)}, \\ &\quad + \frac{(D-C)(d(k_1)^{-1} - k_1/(D-C))}{k_1} - 1 \\ &= \frac{(k_1 b(k_1)-a)\#K}{k_1 d(k_1)} + \frac{a\#K-(D-C)}{k_1 d(k_1)} + \frac{(D-C)-k_1 d(k_1)}{k_1 d(k_1)}.\end{aligned}$$

Therefore,

$$\begin{aligned}k_1 d(k_1)\cdot|\#K\cdot b_3(k_1,k_2) - 1| &\le a^*\delta_1\#K + aB \\ &\quad + (D-C)d(k_1)\cdot\frac{k_1^2}{(D-C)(D-C+B)}.\end{aligned}$$

Finally, we derive

$$\begin{aligned}\#K{\cdot}\left|b_3(k_1,k_2) - \frac{1}{\#K}\right| &\le \frac{a^*\delta_1\#K + aB + k_1^2 d(k_1)/(D-C+B)}{k_1 d(k_1)} \\ &= \frac{aB + a^*\delta_1\#K}{k_1 d(k_1)} + \frac{k_1}{D-C+B} \\ &\le \frac{aB + (a+2A^{-1})\delta_1((D-C)/a+B)}{k_1\cdot(D-C-B)/k_1} + \frac{B}{D-C+B} \\ &< \frac{(a+1)B + 2\delta_1(D-C+aB)}{D-C-B} = \delta_2,\end{aligned}$$

The time bound claimed in (ii) follows from Theorem 2. □

3 Uniform unbalanced moduli

As a next step, we present an efficient algorithm to generate (almost) uniformly random unbalanced RSA moduli, where one prime factor is allowed to be considerably smaller than the other one. The natural rejection sampling process cannot be proved to be efficient in this situation, as explained after stating the algorithm.

This is used to generate safe primes in Section 4, which in turn leads to the safe moduli of Section 5, the ultimate goal of this paper.

Algorithm 5. *Random unbalanced moduli.*

INPUT: *Positive real numbers* $8 \leq A < B < C < D$.

OUTPUT: *Primes* $\ell_1 \neq \ell_2$ *so that* $A \leq \ell_1 \leq B$ *and* $C \leq \ell_1 \ell_2 \leq D$.

1. *Compute* $M = \lceil 6AB \rceil$ *and* $m = \lceil \log_2 6B \rceil$.
2. *Repeat steps 2a and 2b until primes* ℓ_1 *and* ℓ_2 *are found.*
   (a) *Call Algorithm 3 with the input values as above and output* (k_1, k_2).
   (b) *If* k_1 *and* k_2 *are distinct primes, set* $\ell_1 = k_1$ *and* $\ell_2 = k_2$.
3. *Return* (ℓ_1, ℓ_2).

For simplicity, the algorithm does not return the unbalanced modulus $N = \ell_1 \ell_2$ explicitly.

As an alternative one might consider the naive approach of generating uniformly randomly first $\ell_1 \in [A .. B]$ and then $\ell_2 \in [C/\ell_1 .. D/\ell_1]$.

This naive approach fails for the following reason. Assume that $C = o(D)$, A and B are of different orders of magnitude, smaller than that of D, and let $\pi(x)$ denote the number of primes up to x. Then there are about $\pi(D/\ell_1) \sim D/\ell_1 \ln(D/\ell_1) \sim D/\ell_1 \ln D$ possibilities for ℓ_2, and a particular ℓ_2 is chosen with probability proportional to ℓ_1. The same holds for any particular (ℓ_1, ℓ_2). This value is exponentially smaller for small values of ℓ_1 (close to A) than for large ones (close to B). Thus the distribution on (ℓ_1, ℓ_2) is highly nonuniform, as illustrated by the left heat diagram in Figure 1 which is taken from Ziegler & Zollmann (2013). In Algorithm 8 below we need (approximately) uniform (ℓ_1, ℓ_2), and this is actually delivered by Algorithm 5 as shown on the right of Figure 1.

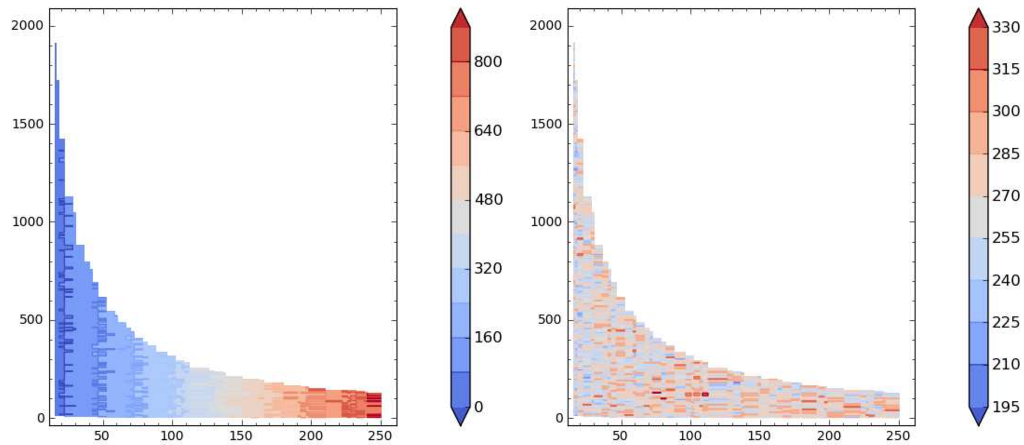

Figure 1: Distribution of 1 000 000 prime pairs (ℓ_0, ℓ_1) obtained from different sampling algorithms for $x = 2^{16}$.

In the usual RSA key generation, one typically uses primes from dyadic intervals, that is with $B = 2A$, and then this nonuniformity is not much of a problem.

We recall the set K of integer pairs from (12), and δ_2 from Theorem 4. For our values of A, B, C, and D, we set

$$X = \{(\ell_1, \ell_2)\colon \ell_1 \text{ and } \ell_2 \text{ distinct primes}, \\ \ell_1 \in [A \mathrel{..} B], \ell_1\ell_2 \in [C \mathrel{..} D]\} \subseteq K, \tag{15}$$

$$a_1 = \frac{1}{\ln\ln B - \ln\ln A} = \frac{1}{\ln(\log_A(B))}. \tag{16}$$

We have the following bounds on a_1 and the size of X.

Lemma 6. *Let $8 \leq A < B < C < D$ be real numbers with*

$$2a_1 \leq \ln^2 A, \tag{17}$$

$$82A\ln A \leq 41B \leq D, \tag{18}$$

$$2^{34} \leq D. \tag{19}$$

Then

$$|a_1^{-1} - \sum_{A \leq \ell \leq B} \frac{1}{\ell}| \leq \frac{1}{\ln^2 A}, \tag{20}$$

$$\frac{D}{5a_1 \ln D} \le \frac{D}{5a_1 \ln(D/A)} \le \#X \le \frac{2D}{a_1 \ln(D/B)}. \tag{21}$$

Proof. Mertens' theorem, see Rosser & Schoenfeld (1962), Theorem 5, implies that

$$\ln\ln B + c - \frac{1}{2\ln^2 B} \le \sum_{\ell \le B} \frac{1}{\ell} \le \ln\ln B + c + \frac{1}{2\ln^2 B} \tag{22}$$

for some constant c, from which (20) follows. An explicit formula and the approximation 0.26150 for c are given in (Tenenbaum, 1995, §I.1.6). There are at most $D^{1/2}$ pairs (k_1, k_2) with $k_1 = k_2$ and $k_1^2 \le D$. We first bound from below the size of X as

$$\begin{aligned} \#X &\ge \sum_{A \le \ell_1 \le B} (\pi(D/\ell_1) - \pi(C/\ell_1)) - D^{1/2} \\ &\ge \sum_{A \le \ell_1 \le B} (\pi(D/\ell_1) - \pi(D/2\ell_1)) - D^{1/2}. \end{aligned} \tag{23}$$

The asymptotic value of $\#X$ is given in Theorem 15 below. Now since $D/2\ell_1 \ge D/2B \ge 20.5$ by (18), Rosser & Schoenfeld (1962), Corollary 3, implies that

$$\pi(D/\ell_1) - \pi(D/2\ell_1) \ge \frac{3D}{5\ell_1 \ln(D/\ell_1)} > \frac{D}{2\ell_1 \ln(D/A)}. \tag{24}$$

It follows that

$$\begin{aligned} D^{1/2} + \#X &\ge \frac{D}{2\ln(D/A)} \sum_{A \le \ell_1 \le B} \frac{1}{\ell_1} \ge \frac{D}{2\ln(D/A)} \left(\frac{1}{a_1} - \frac{1}{\ln^2 A}\right) \\ &\ge \frac{D}{2\ln(D/A)} \left(\frac{1}{a_1} - \frac{1}{2a_1}\right) = \frac{D}{4a_1 \ln(D/A)}. \end{aligned}$$

The assumptions (17) and (19) yield

$$\begin{aligned} &20a_1 \ln(D/A) \le 10 \ln^2 A \ln D < 10 \ln^3 D \le D^{1/2}, \\ &\frac{D}{5a_1 \ln D} \le \frac{D}{5a_1 \ln(D/A)} \le \frac{D}{4a_1 \ln(D/A)} - D^{1/2} \le \#X. \end{aligned} \tag{25}$$

Furthermore, (17) and (Rosser & Schoenfeld, 1962, Corollary 2) imply that

$$\#X \leq \sum_{A \leq \ell_1 \leq B} \pi(\frac{D}{\ell_1}) \leq \sum_{A \leq \ell_1 \leq B} \frac{5D}{4\ell_1 \ln(D/\ell_1)} \leq \frac{5D}{4\ln(D/B)} \sum_{A \leq \ell_1 \leq B} \frac{1}{\ell_1}$$
$$\leq \frac{5D}{4\ln(D/B)}(a_1^{-1} + \frac{1}{\ln^2 A}) < \frac{2D}{a_1 \ln(D/B)}.$$

□

Theorem 7. *Let $8 \leq A < B < C < D \leq 2^n$ be real numbers, assume that* (17), (18), *and* (19) *hold, and also*

$$42 \ln B \leq A, \tag{26}$$
$$AB + C \leq D. \tag{27}$$

Then the following hold.

(i) We have $\delta_2 \leq (14 \ln B)/A \leq 1/3$ and Algorithm 5 returns an element of X, and for any pair $(\ell_1, \ell_2) \in X$, the probability $b_4(\ell_1, \ell_2)$ with which it is returned satisfies

$$|b_4(\ell_1, \ell_2) - \frac{1}{\#X}| \leq \frac{3\delta_2}{\#X}.$$

(ii) The algorithm performs an expected number in $O(a_1 n^2)$ of repetitions of steps 2a and 2b, and has expected runtime polynomial in $a_1 n$.

Proof. We begin with numerical computations that verify some assumptions in previous results. For starters, the condition (4) follows from

$$\ln(B/A) \geq \ln 4 > 8^{-1} + (1 + 8^{-2})^{1/2} \geq A^{-1} + (1 + A^{-2})^{1/2}.$$

Next, we find an upper bound on δ_1:

$$\frac{a^* + 2}{a^* A} = \frac{1 + 2\ln(B/A)}{A} < \frac{2\ln B}{A},$$
$$\frac{1}{2^{m-2}} < \frac{1}{A} < \frac{\ln B}{2A},$$
$$\frac{3B}{a^* M} < \frac{\ln B}{2A},$$
$$\delta_1 < \frac{3\ln B}{A}.$$

For the upper bound on δ_2, we have from (5) that

$$a \le a^* + 2A^{-1} = (\ln(B/A))^{-1} + 2A^{-1} \le 1/\ln 4 + 1/4 < 1,$$

so that $(a+2)B < 3B \le D - C$ and $(D - C + aB)/(D - C - B) \le 2$. Furthermore, $AB/(2 \ln B) + B \le AB/4 + B < AB \le D - C$ and

$$(a+1)B < 2B \le (2 \ln B)(D - C - B)/A,$$

which, together with (26), implies the claimed bounds on δ_2.

We also have

$$\frac{M}{a \ln(B/A)} < \frac{7AB}{a \ln(B/A)} < \frac{7B}{\ln 4} < 6B,$$

so that (9) is satisfied.

For the analysis of the algorithm we have, by equation (14) and with a as in (2), that

$$\#K \le \frac{D - C + aB}{a}.$$

Let $(k_1, k_2) \in K$. Since (4) holds, Theorem 4 says that the probability $b_3(k_1, k_2)$ that (k_1, k_2) is returned by Algorithm 3 satisfies

$$\left| b_3(k_1, k_2) - \frac{1}{\#K} \right| \le \frac{\delta_2}{\#K}. \tag{28}$$

Thus some element of X is returned in one execution of steps 2a and 2b with probability at least

$$\frac{(1-\delta_2)\#X}{\#K} \ge \frac{(1-\delta_2)aD}{5a_1(D - C + aB)\ln D} \ge \frac{(1-\delta_2)a}{10a_1 \ln D}.$$

Since $a^{-1} \in O(\ln B)$, the expected number of executions until success is at most

$$\frac{10a_1 \ln D}{(1-\delta_2)a} \in O(a_1 \ln D \cdot \ln B) \subseteq O(a_1 n^2).$$

Now let $r = \#X/\#K$ and denote as t the random variable counting the number of repetitions until success. For any $k = (k_1, k_2) \in K$, the probability for output k satisfies

$$(1-\delta_2)r \le \operatorname{prob}(k \in X) \le (1+\delta_2)r,$$
$$1 - (1+\delta_2)r \le \operatorname{prob}(k \notin X) \le 1 - (1-\delta_2)r.$$

Since the random choices in different repetitions are independent, we have for any $i \geq 0$

$$\text{prob}(t > i) = (\text{prob}(k \notin X))^i \geq (1 - (1 + \delta_2)r)^i.$$

Thus for any $(\ell_1, \ell_2) \in X$, we have

$$\begin{aligned} \text{prob}(t = i + 1 \text{ and } (\ell_1, \ell_2) \text{ is found}) &\geq (1 - (1 + \delta_2)r)^i \frac{1 - \delta_2}{\#K}, \\ b_4(\ell_1, \ell_2) \geq \frac{1 - \delta_2}{\#K} \sum_{i \geq 0} (1 - (1 + \delta_2)r)^i &= \frac{1 - \delta_2}{(1 + \delta_2)\#X} \geq \frac{1 - 2\delta_2}{\#X}. \end{aligned}$$

A similar calculation shows that

$$b_4(\ell_1, \ell_2) \leq \frac{1 + \delta_2}{(1 - \delta_2)\#X} \leq \frac{1 + 3\delta_2}{\#X}.$$

Primality of an integer can be tested in random polynomial time, see Crandall & Pomerance (2005), §3.4, and the claimed cost bound follows. □

4 Safe primes

We use the following notation, for any α with $0 \leq \alpha < 1/2$.

$$\begin{aligned} P &= \text{ set of primes}, \\ \text{SG}_1 &= \{p \in P \colon (p-1)/2 \text{ prime}\}, \\ \text{SG}_{2,\alpha} &= \{p \in P \colon (p-1)/2 = \ell_1 \ell_2 \text{ with } \ell_1 < \ell_2 \text{ primes and } (\ell_1 \ell_2)^\alpha \leq \ell_1\}. \end{aligned}$$

Furthermore, $P(x)$, $\text{SG}_1(x)$, and $\text{SG}_{2,\alpha}(x)$ are the corresponding subsets of those p with $p \leq x$, and $\pi(x)$, $\pi_1(x)$, and $\pi_{2,\alpha}(x)$ denote the respective cardinalities. An integer ℓ is a Sophie Germain prime if and only if $2\ell + 1 \in \text{SG}_1$, as defined in the Introduction, and $\text{SG}_2 = \text{SG}_{2,0}$. The last condition in our definition of $\text{SG}_{2,\alpha}$ is equivalent to $\ell_2 \leq \ell_1^{1/\alpha - 1}$ when $\alpha \neq 0$.

Algorithm 8. *Generating a safe prime.*

INPUT: *Positive bounds x and $\alpha < 1/2$.*

OUTPUT: *A prime p with $x/\ln^2 x < p \leq x$.*

1. *Compute* $y_0 = (x - \ln^2 x)/2\ln^2 x$ *and* $y_1 = (x-1)/2$.

   Repeat steps 2–5 until a prime is returned.

2. *Choose a random prime* $\ell \in [y_0 \,..\, y_1]$.

3. *If* $2\ell + 1$ *is prime then return* $p = 2\ell + 1$.

4. *Using Algorithm 5 with inputs* $(A, B, C, D) = (x^\alpha, x^{1/2}, y_0, y_1)$, *choose an approximately uniformly random sample from the pairs* (ℓ_1, ℓ_2) *of primes with* $\ell_1 \in [x^\alpha \,..\, x^{1/2}]$ *and* $\ell_1\ell_2 \in [y_0 \,..\, y_1]$.

5. *If* $2\ell_1\ell_2 + 1$ *is prime then return* $p = 2\ell_1\ell_2 + 1$.

The interesting question is how many repetitions we expect to perform.

Theorem 9. *(i) Any* p *returned by the algorithm satisfies the output specification, and* $p \equiv 3 \bmod 4$. *For any output* p, $(p-1)/2$ *is squarefree with at most two prime divisors, and each of them is at least* x^α.

(ii) Let $0.25 \le \alpha < 0.276$ *and* n *be sufficient large. For an input* $x \in [2^{n-1} \,..\, 2^n]_{\mathbb{R}}$, *the expected number of repetitions made by Algorithm 8 until an output is returned is* $O(n)$, *and the expected runtime of the algorithm is polynomial in* n.

Proof. An ℓ leading to an output in step 3 satisfies $\ell \ge y_0 > x^\alpha$. Suppose that (ℓ_1, ℓ_2) is chosen in step 4 and that $p = 2\ell_1\ell_2 + 1$ is returned in step 5. Then $x^\alpha \le \ell_1 \le x^{1/2}$ and $\ell_2 \ge y_0/\ell_1 \ge (x - \ln^2 x)/2x^{1/2}\ln^2 x > x^\alpha$. Also $\ell_1 \ne \ell_2$, since 3 divides $2\ell^2 + 1$ for every prime $\ell \ne 3$. Thus any output has the stated properties. The primality tests can be performed in polynomial time. The assumptions of Theorem 7 are easily checked, and thus also one execution of step 4 uses polynomial time. It is sufficient to show the claim about the expected number of repetitions.

Our main tool is a result of Heath-Brown (1986), Lemma 1, taken with $k = 1$, $K = 2$, any $u \in \{3, 7, 11, 15\}$, and $v = 16$, which implies that

$$\pi_1(x) + \pi_{2,0.276}(x) \ge \frac{cx}{\ln^2 x} \tag{29}$$

for some constant $c > 0$. No explicit value for c is known.

By (29), at least one (possibly both) of the alternatives

$$\pi_1(x) \geq \frac{cx}{2\ln^2 x}, \tag{30}$$

and

$$\pi_{2,0.276}(x) \geq \frac{cx}{2\ln^2 x} \tag{31}$$

holds. In this and the next section, we make no attempt to optimize constants, and even use the trivial form

$$\frac{x}{2\ln x} \leq \pi(x) = \#P(x) \leq \frac{2x}{\ln x}$$

of the Prime Number Theorem.

We consider the "shifted multiplication" map μ with $\mu((\ell_1, \ell_2)) = 2\ell_1\ell_2 + 1$, and

$$\begin{aligned}
X_1 &= \{2\ell+1 \colon \frac{x}{\ln^2 x} < 2\ell+1 \leq x, \ell \text{ prime}\},\\
Y_1 &= X_1 \cap P,\\
X_2 &= \{(\ell_1,\ell_2) \colon \ell_1 \in [x^\alpha \mathinner{..} x^{1/2}] \text{ and } \ell_2 \in \left[\frac{y_0}{\ell_1} \mathinner{..} \frac{y_1}{\ell_1}\right] \text{ primes}\},\\
Y_2 &= \mu(X_2) \cap P.
\end{aligned} \tag{32}$$

By the Prime Number Theorem, we have $\#X_1 \leq \pi(x/2) < 2x/\ln x$. Since $2y_0 + 1 = x/\ln^2 x$, we have $\mathrm{SG}_1(x) \subseteq Y_1 \cup P(x/\ln^2 x)$, the integers $2\ell + 1$ tested in step 3 are uniformly distributed in X_1, and $2\ell + 1$ is returned if it is in Y_1. Since n is sufficiently large, we may assume $n > 1 + 18/c$ and $x > e^{12/c}$, so that for

$$a_2 = \frac{1}{c - 6/\ln x}$$

we have $0 < a_2 \leq 2/c$. If (30) holds, then with

$$\begin{aligned}
\#Y_1 &\geq \pi_1(x) - \pi\left(\frac{x}{\ln^2 x}\right) \geq \frac{cx}{2\ln^2 x} - \frac{2x}{\ln^2 x \cdot \ln(x/\ln^2 x)}\\
&\geq \frac{cx}{2\ln^2 x} - \frac{3x}{\ln^3 x} = \frac{x}{2a_2\ln^2 x} \geq \frac{cx}{4\ln^2 x}.
\end{aligned} \tag{33}$$

Thus the expected number of repetitions of steps 2 and 3 is at most

$$\frac{\#X_1}{\#Y_1} < \frac{2x/\ln x}{cx/4\ln^2 x} = \frac{8\ln x}{c} \in O(n).$$

The claim follows in this case.

Now we assume that (30) does not hold, so that (31) does. We have

$$1.4 < a_1 = \frac{1}{-\ln 2\alpha} < 1.7. \tag{34}$$

Now X_2 is the set X defined in (15) for our input parameters $(A, B, C, D) = (x^\alpha, x^{1/2}, y_0, y_1)$. By Lemma 6, we have

$$\#X_2 \leq \frac{2x}{a_1 \ln((x-1)/2x^{1/2})} \leq \frac{2x}{a_1(0.5 \ln x - 1)} \leq \frac{4x}{\ln x}.$$

An asymptotic formula for $\#X_2$ is provided in Theorem 15, but at this stage, the above upper bound suffices.

Now let $\beta = 0.276$ and $p = 2\ell_1\ell_2 + 1 \in \mathrm{SG}_{2,\beta}(x)$ with $\ell_1 < \ell_2$. Then either $\ell_1\ell_2 < y_0$ or ($y_0 \leq \ell_1\ell_2 \leq y_1$ and $\ell_1 < \ell_2 \leq \ell_1^{1/\beta-1}$). In the latter case, we have $\ell_1^2 < \ell_1\ell_2 \leq y_1$ and $\ell_1 \leq y_1^{1/2} < x^{1/2}$. Furthermore, $\alpha < \beta$ and $x^\alpha < y_0^\beta \leq (\ell_1\ell_2)^\beta \leq \ell_1$, so that $(\ell_1, \ell_2) \in X_2$. It follows that $p \in Y_2$, and hence $\mathrm{SG}_{2,\beta}(x) \subseteq Y_2 \cup P(2y_0 + 1)$ and, similar to (33),

$$\#Y_2 \geq \pi_{2,\beta}(x) - \pi(2y_0 + 1) \geq \frac{cx}{2\ln^2 x} - \pi(\frac{x}{\ln^2 x}) \geq \frac{cx}{4\ln^2 x}.$$

Since $Y_2 \subseteq \mu(X_2)$, we have $\mu(\mu^{-1}(Y_2)) = Y_2$ and $\#\mu^{-1}(Y_2) \geq \#Y_2$. Success occurs in step 5 if $(\ell_1, \ell_2) \in \mu^{-1}(Y_2)$, and any (ℓ_1, ℓ_2) is chosen in step 4 with probability at least $(1 - 3\delta_2)/\#X_2$ by Theorem 7, whose assumptions are readily checked. Then the probability of success is at least

$$\frac{1 - 3\delta_2}{\#X_2} \cdot \#\mu^{-1}(Y_2) \geq \frac{(1 - 3\delta_2)\#Y_2}{\#X_2}.$$

It follows that the expected number of repetitions of steps 4 and 5 until success is at most

$$\frac{\#X_2}{(1 - 3\delta_2)\#Y_2} \leq \frac{4x/\ln x}{(1 - 3\delta_2) \cdot cx/4\ln^2 x} \in O(n),$$

which concludes the proof. □

More detailed calculations show that any $n \geq \max\{18/c, 155\}$ is sufficiently large for the conclusions to hold, for $\alpha = 1/4$.

On the other hand, for an asymptotic result we can replace the lower bound in the output specification by $p \geq a(y) \cdot y / \ln y$ for any function $a \in o(1)$.

An output p from step 3 is uniformly random in the set Y_1 from (32), and by Theorem 7 (i), an output from step 5 is approximately random in Y_2. After Conjecture 16 below, we address the question of how to make p in either case uniformly random in $Y_1 \cup Y_2$.

5 Safe moduli

We generate a safe modulus $N = pq$ from two executions of Algorithm 8.

Algorithm 10. *Generating a safe modulus.*

INPUT: *Positive bounds y and α with $1/4 \leq \alpha < 0.276$.*

OUTPUT: *A modulus N with $y/\ln^2 y \leq N \leq y$.*

1. *Repeat steps 2–4 until a modulus is returned.*
2. *Call Algorithm 8 with inputs $x = y^{1/2}$ and α, and output p.*
3. *Call Algorithm 8 with inputs $x = y/p$ and α, and output q.*
4. *If $\gcd((p-1)/2, (q-1)/2) = 1$, then return $N = pq$.*

A *Blum integer* is a product of two primes, both congruent to 3 modulo 4. These were introduced in Blum, Blum & Shub (1986).

Theorem 11. *(i) Any output N satisfies the output specification and is a Blum integer. Furthermore, $\varphi(N)/4$ is squarefree with at most four prime factors, and each of these is at least $y^{\alpha/2}$.*

(ii) For n sufficiently large and $y \in [2^{n-1} .. 2^n]_{\mathbb{R}}$, the expected number of repetitions in Algorithm 10 until an output N is returned is at most $1 + y^{-1/8} \ln^2 y < 2$, and the expected runtime of the algorithm is polynomial in n.

Proof. (i) The claims follow from Theorem 9, using the fact that $y/p \geq y^{1/2}$.
(ii) We write ℓ_0 or (ℓ_1, ℓ_2) for the choice that leads to an output p in step 2, depending on whether step 3 or step 5 of the call to Algorithm 8 is successful, and similarly r_0 or (r_1, r_2) for step 3. We denote as R_0, R_1 and R_2 the sets of possible values for r_0, r_1, and r_2, respectively. Since $\ell_1 \neq \ell_2$ and $r_1 \neq r_2$, the gcd condition in step 4 is violated only if one of r_0, r_1, r_2 equals one of ℓ_0, ℓ_1, ℓ_2. We write b_5 for the probability of this event, conditioned on some output p of step 2, and distinguish two cases.

In the first one, the second call to Algorithm 8 returns from its step 3. Then r_0 has to avoid at most two values, so that $b_5 \leq 2/\#R_0$, since r_0 is chosen uniformly in R_0.

In the second case, Algorithm 8 returns from its step 5. Then Theorem 7 implies that (r_1, r_2) assumes any specific value with probability at most $(1 + 3\delta_2)/\#X < 2/\#X$ with X from (15). There are at most two values that both r_1 and r_2 have to avoid. This makes for a total of at most $2\#R_1 + 2\#R_2$ pairs to be avoided by $\#R_1$ choices and

$$b_5 \leq \frac{4\#R_1 + 4\#R_2}{\#X}. \tag{35}$$

We now prove upper bounds on the three $\#R_i$. By Theorem 9, the output p of step 2 satisfies

$$\frac{4y^{1/2}}{\ln^2 y} = \frac{y^{1/2}}{\ln^2 y^{1/2}} \leq p \leq y^{1/2}.$$

In step 2 of Algorithm 8, when called in step 3 of Algorithm 10, we have $y_0 \leq y_1/2$ and

$$\begin{aligned} y_1 &= \frac{y/p - 1}{2}, \\ \#R_0 &= \pi(y_1) - \pi(y_0) \geq \pi(y_1) - \pi(y_1/2) \\ &\geq \frac{3 \cdot y_1/2}{5 \cdot \ln(y_1/2)} \geq \frac{y^{1/2}}{10 \ln(y^{1/2}/6)} > \frac{y^{1/2}}{5 \ln y}, \\ b_5 &< \frac{10 \ln y}{y^{1/2}} < \frac{\ln y}{2y^{1/8}}. \end{aligned}$$

For the second case, we have in step 4 of the same call to Algorithm 8

$$
\begin{aligned}
(y/p)^{\alpha} \leq r_1 \leq (y/p)^{1/2}\,, \\
\#R_1 = \pi\left((y/p)^{1/2}\right) - \pi\left((y/p)^{\alpha/2}\right) < \pi\left(\frac{y^{1/4}\ln y}{2}\right) < 4y^{1/4}, \\
\frac{y/p-1}{2r_1} < \frac{y}{2pr_1} \leq \frac{y^{1-\alpha}}{2p^{1-\alpha}} \leq \frac{y^{(1-\alpha)/2}(\ln y)^{2-2\alpha}}{5} \leq \frac{y^{3/8}\ln^{3/2} y}{5}.
\end{aligned}
$$

Finally

$$
\begin{aligned}
\#R_2 &= \pi\left(\frac{y/p-1}{2r_1}\right) - \pi\left(\frac{y/p-\ln^2(y/p)}{2r_1\ln^2(y/p)}\right) \\
&< \pi\left(\frac{y^{3/8}\ln^{3/2} y}{5}\right) \leq 2y^{3/8}\ln^{1/2} y.
\end{aligned}
$$

From (21) with the parameters $B = (y/p)^{1/2} = A^{1/2\alpha}$ and $D = y/p \geq y^{1/2}$ we obtain

$$
\#X \geq \frac{y^{1/2}\cdot(-\ln(2\alpha))}{5\cdot \ln y^{1/2}} > \frac{y^{1/2}}{2\ln y}.
$$

It follows that in the second case we have in (35)

$$
b_5 \leq \frac{4\cdot(4y^{1/4} + 2y^{3/8}\ln^{1/2} y)\cdot 2\ln y}{y^{1/2}} \leq \frac{\ln^2 y}{2y^{1/8}}.
$$

The gcd condition holds in any case with probability at least $1 - b_5$, and the number of iterations is at most $(1-b_5)^{-1} \leq 1 + 2b_5 \leq 1 + y^{-1/8}\ln^2 y$. □

Section 7 presents more details on the runtime of Algorithms 8 and 10.

Corollary 12. *Let N be an output of Algorithm 10. For uniformly random $a \in \mathbb{Z}_N^{\times}$, a^2 generates the group of squares in $\mathbb{Z}_N^{\times}$ with probability at least $1 - 4N^{-1/8}$.*

Proof. We have

$$
\begin{aligned}
\mathbb{Z}_p^{\times} &\cong \mathbb{Z}_2 \times \mathbb{Z}_{(p-1)/2}, \\
\mathbb{Z}_N^{\times} &\cong \mathbb{Z}_2^2 \times \mathbb{Z}_{(p-1)/2} \times \mathbb{Z}_{(q-1)/2} \cong \mathbb{Z}_2^2 \times \mathbb{Z}_{\varphi(N)/4},
\end{aligned}
$$

by the condition in step 4.

The set $\square_N$ of squares in $\mathbb{Z}_N^\times$ is isomorphic to $\mathbb{Z}_{\varphi(N)/4}$ and hence cyclic with $\varphi(\varphi(N)/4)$ generators. The squaring map from $\mathbb{Z}_N^\times$ to $\square_N$ is 4-to-1, and for a uniformly random $a \in \mathbb{Z}_N^\times$ we have

$$\text{prob}\,\{a \in \mathbb{Z}_N^\times : a^2 \text{ generates } \square_N\} = \frac{\varphi(\varphi(N)/4)}{\varphi(N)/4}. \tag{36}$$

We first consider the case where both p and q come from step 5 of the respective call to Algorithm 8. We can then write $\varphi(N)/4 = \ell_1\ell_2r_1r_2$, with four distinct primes ℓ_1, ℓ_2, r_1, r_2, by Theorem 9 and the condition in step 4 of Algorithm 10. These primes are all at least $y^{\alpha/2}$, and

$$\begin{aligned}\frac{\varphi(\varphi(N)/4)}{\varphi(N)/4} &= \left(1-\frac{1}{\ell_1}\right)\left(1-\frac{1}{\ell_2}\right)\left(1-\frac{1}{r_1}\right)\left(1-\frac{1}{r_2}\right)\\ &\geq 1-\left(\frac{1}{\ell_1}+\frac{1}{\ell_2}+\frac{1}{r_1}+\frac{1}{r_2}\right)\\ &\geq 1-\frac{4}{y^{\alpha/2}} \geq 1-\frac{4}{N^{\alpha/2}} \geq 1-\frac{4}{N^{1/8}}.\end{aligned}$$

The last estimate also holds for the other possibilities for the factors of $p-1$ and $q-1$, including the case where $\gcd((p-1)/2,(q-1)/2) = 1$. Together with (36), this concludes the proof. $\square$

As y grows, the lower bound of Corollary 12 comes exponentially close to 1.

Corollary 13. *Let n be a sufficiently large integer. Then we can generate in expected time polynomial in n an RSA modulus $N = pq$ so that*

- $2^n/n^2 \leq N \leq 2^n$,
- $\varphi(N)/4 = (p-1)(q-1)/4$ *is squarefree with at most four prime factors,*
- *each such prime factor is at least* $2^{n/8}$,
- *for uniformly random* $a \in \mathbb{Z}_N^\times$, a^2 *generates the group of squares in* $\mathbb{Z}_N^\times$ *with probability at least* $1 - 4 \cdot 2^{-n/8}$.

It follows that the moduli presented here can be used in the encryption scheme of Hofheinz *et al.* (2012). If g generates $\square_N$, then $|g|$ generates the group $\mathbb{QR}_N^+$ of signed quadratic residues, in their notation. The factoring assumption then refers to the moduli generated by Algorithm 10.

For various notions of RSA integers, their number is estimated in de Weger (2008), Decker & Moree (2008), and Loebenberger & Nüsken (2013).

6 Heuristic estimates and extended range

We want to compare the number of Sophie Germain primes to that of the Sophie Germain$_2$ integers generated in steps 4 and 5 of Algorithm 8. For Sophie Germain primes, we take a prime $\ell \leq x$, of which there are about $x/\ln x$ many. If $2\ell+1$ is also prime, then ℓ is a Sophie Germain prime. Since the density of primes up to $2x$ is about $1/\ln(2x) \sim 1/\ln x$, one might naively expect there to be about $x/\ln^2 x$ of them. Although this gives the right order of magnitude, the asymptotics is false as it ignores so-called "local" (or divisibility) conditions. The argument of Bateman & Horn (1962) in this special case suggests the following.

We take a prime $q \neq 2, \ell$. Then $2\ell + 1 \not\equiv 1 \bmod q$, while general primes are allowed to be 1 mod q. This consideration also applies to twin primes and leads to the standard heuristics on the number of Sophie Germain primes, namely that there are about $2C_2x/\ln^2 x$ of them up to x, where

$$C_2 = \prod_{p\geq 3}\left(1 - \frac{1}{(p-1)^2}\right) \approx 0.66016 \tag{37}$$

is the *twin prime constant*. In our situation, x denotes an upper bound on $2\ell + 1$, so that we consider the Sophie Germain primes $\ell \leq x/2$. We thus rephrase the heuristics as follows for our purposes.

We use $\sim$ to denote the asymptotic equality of two quantities, so that $f \sim g$ means that $|f(x) - g(x)| \in o(g(x))$ as $x \to \infty$. This relation is symmetric in f and g.

Conjecture 14 (Sophie Germain prime conjecture)**.** *We have*

$$\pi_1(x) \sim \frac{2C_2}{\ln x} \cdot \pi(x/2) \sim \frac{C_2\, x}{\ln^2 x}.$$

The local behavior of $2\ell_1\ell_2+1$, as (ℓ_1, ℓ_2) ranges over the set X_2 as defined in (32), is the same as that of $2\ell+1$ in this range. The following conjecture is the natural analog of the previous one:

$$\pi_{2,\alpha}(x) \sim \frac{2C_2}{\ln x} \cdot \#\mu(X_2). \tag{38}$$

where the map μ is as in the proof of Theorem 9. It now remains to estimate the magnitude of $\#\mu(X_2)$. This can be done unconditionally.

Theorem 15. *We have*

$$\#\mu(X_2) \sim \#X_2 \sim \frac{\ln(\alpha^{-1}-1)}{2} \cdot \frac{x}{\ln x}.$$

Proof. We start with $\#X_2$ and set

$$r_1 = \sum_{x^\alpha \le \ell \le x^{1/2}} \pi\left(\frac{x}{2\ell}\right),$$
$$r_2 = \sum_{x^\alpha \le \ell \le x^{1/2}} \pi\left(\frac{x}{2\ell \ln^2 x}\right).$$

Then the asymptotic version of (23), ignoring the term $D^{1/2}$ of small order, says that

$$\#X_2 \sim r_1 - r_2.$$

For r_1, the prime number theorem and partial summation imply

$$\begin{aligned} r_1 &\sim \frac{x}{2}\int_{x^\alpha}^{x^{1/2}} \frac{1}{\lambda \ln(x/2\lambda)\ln\lambda}\, \mathrm{d}\lambda \sim \frac{x}{2}\int_{x^\alpha}^{x^{1/2}} \frac{1}{\lambda\ln(x/\lambda)\ln\lambda}\, \mathrm{d}\lambda \\ &= \frac{x}{2}\int_{x^\alpha}^{x^{1/2}} \frac{1}{(\ln x - \ln\lambda)\ln\lambda}\, \mathrm{d}\ln\lambda = \frac{x}{2}\int_{\alpha u}^{u/2} \frac{1}{(u-v)v}\, \mathrm{d}v, \end{aligned}$$

where $u = \ln x$. Therefore,

$$r_1 \sim \frac{x}{2u}\int_{\alpha u}^{u/2}\left(\frac{1}{u-v} + \frac{1}{v}\right) d\,v = \frac{x}{2u}\left(\ln(1-\alpha) - \ln\alpha\right) = \frac{\ln(\alpha^{-1}-1)}{2} \cdot \frac{x}{\ln x}.$$

For r_2, we recall from (16) and (20) that

$$\sum_{x^\alpha \le \ell \le x^{1/2}} \frac{1}{\ell} \sim \ln\frac{1}{2\alpha}.$$

Thus

$$\begin{aligned} r_2 &\leq \sum_{x^\alpha \leq \ell \leq x^{1/2}} \frac{x}{2\ell \ln^2 x \cdot \ln(x/2\ell \ln^2 x)} \\ &\leq \frac{x}{2\ln^2 x} \sum_{x^\alpha \leq \ell \leq x^{1/2}} \frac{1}{\ell \ln(x^{1/2}/2\ln^2 x)} \leq \frac{x \ln(1/2\alpha)}{\ln^3 x} \in o(r_1). \end{aligned}$$

It follows that

$$\# X_2 \sim r_1. \tag{39}$$

Now it remains to determine the size of $\mu(X_2)$. The map $\mu\colon X_2 \to \mu(X_2)$ is sometimes 1-to-1 and sometimes 2-to-1. The latter happens if and only if $(\ell_1, \ell_2), (\ell_2, \ell_1) \in X_2$. For every such pair (ℓ_1, ℓ_2), we have $\ell_1, \ell_2 \leq x^{1/2}$, and thus there are at most $\pi(x^{1/2})^2 \in O(x/\ln^2 x) \subseteq o(r_1)$ of them. Now (39) implies that $\#\mu(X_2) \sim \# X_2 \sim r_1$. $\square$

We have $\ln((1/4)^{-1} - 1) = \ln 3 \approx 1.098$. The methods of Loebenberger & Nüsken (2013) provide upper and lower bounds on $\#\mu(X_2)$ of the form const $\cdot\, / \ln x$. From the heuristic argument (38), we derive the following.

Conjecture 16 (SG_2 prime conjecture)**.** *For $1/4 \leq \alpha < 1/2$, we have*

$$\pi_{2,\alpha}(x) \sim \frac{C_2 \ln(\alpha^{-1} - 1)\, x}{\ln^2 x}.$$

Under the two conjectures, there are roughly 10% more SG_2- than SG_1-primes for $\alpha = 1/4$.

We now extend the range of applicability of our method. Lemma 6 and thus Theorem 9 depend on a version of Mertens' theorem over certain intervals, that is, on good estimates of the sum

$$M(A, B) = \sum_{\ell \in (A \,..\, B]} \frac{1}{\ell}.$$

In our present application, B is substantially large than A, so that classical results allow us to handle this sum. However, for future extensions and possible ramifications of our ideas and results, it might be useful to study this sum in a range of A and B which is as wide as possible. For two quantities x and y, we write $x \asymp y$ if for some positive constants $c_1 \leq c_2$ we have $c_1 y \leq x \leq c_2 y$.

Theorem 17. *For real A and B with $B \geq A + A^{0.525} \geq 3$ we have*

$$M(A, B) \asymp \ln \frac{\ln B}{\ln A}.$$

Proof. We assume that A is large enough and let $\Delta = B - A$. A result of Baker *et al.* (2001) on primes in short interval (see also Harman (2007), Theorem 7.2) implies that

$$\pi(A + \Delta) - \pi(A) \asymp \Delta / \ln A \tag{40}$$

for $\Delta \geq A^{0.525}$. Only the lower bound on the left hand side in (40) requires this restriction; the upper bound holds for any Δ by the celebrated Brun-Titchmarsh theorem, see Iwaniec & Kowalski (2004), Theorem 6.6.

As a first case, we consider $A + A^{0.525} \leq B < A + A/\ln A$. Then

$$\frac{1}{B} (\pi(B) - \pi(A)) \leq M(A, B) \leq \frac{1}{A} (\pi(B) - \pi(A)),$$

and (40) implies

$$M(A, B) \asymp \frac{\Delta}{A \ln B}. \tag{41}$$

Since $\Delta = o(A)$ and $\ln(1 + z) \sim z$ as $z \to 0$, we also have

$$\frac{\ln B}{\ln A} = 1 + \frac{\ln(B/A)}{\ln A} = 1 + \frac{\ln(1 + \Delta/A)}{\ln A} \sim 1 + \frac{\Delta}{A \ln A},$$

$$\ln \frac{\ln A}{\ln B} \sim \frac{\Delta}{A \ln A} \sim \frac{\Delta}{A \ln B}.$$

Together with (41), Theorem 17 follows in this case. We now come to the case where $A + A/\ln A \leq B$. Then

$$\frac{\ln B}{\ln A} \geq \frac{\ln A + \ln(1 + 1/\ln A)}{\ln A} \sim \frac{\ln A + 1/\ln A}{\ln A} = 1 + \frac{1}{\ln^2 A}$$

and $1/\ln^2 A \in O(\ln(\ln B/\ln A))$. A slight modification of the argument in Виноградов (1963), coupled with Tenenbaum (1995), Theorem 8, Section I.1, implies that

$$\sum_{p \leq x} \frac{1}{p} \in \ln \ln x + \gamma + O\left(\exp\left(-c_0 (\ln x)^{3/5}\right)\right)$$

for some absolute constant $c_0 > 0$. Furthermore, we have

$$\exp\left(-c_0(\ln A)^{3/5})\right) \in o\left(\frac{1}{\ln^2 A}\right) \subseteq o\left(\ln\frac{\ln B}{\ln A}\right).$$

Thus we find

$$M(A,B) \in \ln\frac{\ln B}{\ln A} + O\left(\exp\left(-c_0(\ln A)^{3/5})\right)\right).$$

Hence

$$M(A,B) \sim \ln\frac{\ln B}{\ln A},$$

which concludes the proof of Theorem 17 also in this case. □

Until the result of Baker *et al.* (2001) is improved, there is clearly no chance to extend the above range of A and B.

A famous result of Huxley (1972), see also Heath-Brown (1988), shows that for $\Delta \geq A^{7/12}$, we can replace $\asymp$ in (40) by $\sim$. That is, in the $\asymp$ notation, both "constants" c_1 and c_2 can be chosen in $1 + o(1)$. Arguing as above, we find that in this range, Theorem 17 also holds with $\sim$. We have $7/12 = 0.58333\ldots$

7 Cost estimates

We now analyze the cost of Algorithm 8 more closely. We denote by $\mathsf{M}(n)$ a number of bit operations with which arithmetic (addition, multiplication, division with remainder) can be performed on n-bit integers. Thus we have $\mathsf{M}(n) \in O(n^2)$ with classical and $\mathsf{M}(n) \in O^\sim(n)$ with fast arithmetic, where the $O^\sim$ notation hides logarithmic factors. We also denote as $\mathsf{T}(n)$ the cost of testing an n-bit integer for primality. There are several choices.

- deterministic: $\mathsf{T}(n) \in O^\sim(n^6)$ (see Crandall & Pomerance (2005), § 4.5),
- probabilistic primality test: $O^\sim(n^4)$ (see Crandall & Pomerance (2005), § 4.5),
- probabilistic compositeness test: $\mathsf{T}(n) \in O(tn\mathsf{M}(n))$ for some t, the number of iterations of a single test. These are actually compositeness tests. See von zur Gathen & Gerhard (2003), Theorem 18.6, or Crandall & Pomerance (2005), Algorithm 3.4.7).

For practical purposes, one will use the last type of test in the algorithm. After Algorithm 10 has produced an output, one can test the pseudoprimes involved—p, q, and the factors of $p-1$ and $q-1$—by a probabilistic primality test. This cost is within the time bound of the algorithm.

We examine separately SG_1 *prime generation*, that is, Algorithm 8 with steps 4 and 5 removed, and SG_2 *prime generation*, that is, Algorithm 8 with steps 2 and 3 removed.

In SG_1 prime generation, we choose a uniformly random $k \le x$, test it for primality, and on success, test $2k+1$ for primality. The probability finding a prime $p = 2\ell + 1$ with ℓ prime is approximately $\pi_1(x)/x$ and the expected cost of producing such a p is $x/\pi_1(x) \cdot \mathsf{T}(n)$.

In SG_2 prime generation, we choose uniformly random $(k_1, k_2) \in K$, test both for primality, and on success, test $2k_1k_2 + 1$ for primality. The probability finding a prime p is approximately $\pi_2(x)/\#K$ and the expected cost of producing such a p is $\#K/\pi_2(x) \cdot \mathsf{T}(n)$. For a from (2), we have $a \approx 1/\ln(x^{1/2}/x^{\alpha}) = 2/(1-2\alpha)\ln x$. Using (14), we find $\#K \approx x/a \approx ((1-2\alpha)x\ln x)/2$.

We know from (29) that at least one of $\pi_1(x)$ and $\pi_1(x)$ is bounded from below by $cx/2\ln^2 x$. Algorithm 8 runs both SG_1 and SG_2 generation in tandem and halts whenever one of them succeeds. Its expected cost is therefore the minimum of the two costs. We have shown the following.

Theorem 18. *Algorithms 8 and 10 for the generation of safe primes and safe moduli with nearly n bits, respectively, take an expected number of $O(n^3\mathsf{T}(n))$ bit operations.*

With probabilistic compositeness tests, this comes to $O(n^4\mathsf{M}(n))$ operations, that is, $O(n^6)$ operations with classical and $O^{\sim}(n^5)$ with fast arithmetic.

If Conjectures 14 and 16 hold, then the cost comes to $O(n^3\mathsf{M}(n))$ for SG_1 prime generation and $O(n^4\mathsf{M}(n))$ for SG_2. We might run SG_1 prime generation n times for each execution of SG_2 prime generation. Then we are in the best of two worlds:

- The algorithm provably terminates.
- If the Sophie Germain Conjecture holds, then it does not take much more time than pure SG_1 prime generation.

bit-length	128	256	512	1024	2048
SG_1	0.2 s	1.1 s	12.6 s	298.1 s	5798.8 s
SG_2	20.8 s	164.7 s	2582.5 s	31892.8 s	558600.2 s
SG_2 (fast)	0.4 s	1.8 s	11.7 s	144.9 s	2147.5 s

Table 1: Time needed for finding a safe prime, depending on the bit-length of x. (Average over at least 100 findings, except for SG_2 of bit-lengths ≥ 1024; Hardware: single core Intel Xeon, 3.00GHz)

The advantage is that pure SG_1 prime generation is not proven to terminate.

In a "fast" variant of SG_2 generation, instead of rejecting (k_1, k_2) if one of them is composite, we first test k_1 and on success keep generating values for k_2 until we find a prime. This works well in practice (Table 1 from Ziegler & Zollmann (2013)), but samples a slightly non-uniform distribution of primes p.

The primes generated have n or slightly fewer bits. If exactly n bits are required, one can reject the smaller ones. Under the conjectures, this also works in polynomial time.

For a simplified version of our method, we recall that Algorithms 3 and 5 rely on an asymptotically uniform sampling of points under a hyperbola given by Algorithm 1. We now exhibit a slower but simpler "dyadic" algorithm, using y_0 and y_1 from Algorithm 8.

Let $n = \lceil \log_2 x \rceil$ and $j_0 = \lfloor \alpha \log_2 x \rfloor$. We know that either (30) or (31) holds. In the latter case, for some $j \in [j_0 \,..\, n/2]$ there are at least

$$\frac{\pi_{2,\alpha}(x)}{n/2 - j_0} \geq \frac{cx}{n \ln^2 x}$$

primes $p = 2\ell_1\ell_2+1 \in SG_{2,\alpha}(x)$ with primes ℓ_1 and ℓ_2 such that $\ell_1 \in [2^j, 2^{j+1}]$. A simple version of Algorithm 8 repeats on input x and α the following steps until success:

- choose $\ell \in [y_0 \,..\, y_1]$ uniformly at random and test ℓ and $2\ell + 1$ for primality,
- for $j \in [j_0 \,..\, n/2 - 1]$, choose $\ell_1 \in [2^j \,..\, 2^{j+1}]$ and $\ell_2 \in [y_0/\ell_1 \,..\, y_1/\ell_1]$ uniformly at random and test ℓ_1, ℓ_2, and $2\ell_1\ell_2 + 1$ for primality.

This dyadic method has an expected cost of $O(n^4 \mathsf{M}(n))$ bit operations. If one performs about n iterations of the first step for each execution of the second one, this bound still holds and the algorithm provably terminates. But if Conjecture 14 is true, this version uses about as much time as SG_1 prime generation. Its advantage is its greater simplicity, when compared to Algorithm 8.

8 Comments and open questions

From the point of view of algorithmic applications, it would be nice to have a version of Lemma 1 in Heath-Brown (1986) with an effective (or, even better, an explicitly computed) constant c for which (29) holds. This, however, may be a nontrivial task and may only work for values of β smaller than 0.276.

The proofs in this paper rely on fairly deep results in analytic number theory. But the resulting algorithm is quite simple. In any efficient prime generation method, one will need a (probabilistic) primality test. This is sufficient for Sophie Germain prime generation. For SG_2 prime generation, one only needs in addition a variable-precision numerical package to approximate $A(B/A)^{v/M}$ in step 2 of Algorithm 1 with sufficient accuracy. It would be interesting to see how close our comparison in Section 7 comes to the reality of a fair implementation.

Besides making our main results and algorithms stronger, Algorithm 1 may have more applications. For example, one can consider various approximate counting problems with positive integer points (m, n) in the hyperbolic domain $mn \leq x$. The exact determination of the total number of such points is treated in (Tao *et al.*, 2012, Theorem 2.1 and Section 2.1). This number can also be approximated, with the currently best known error bound $x^{131/416+o(1)}$, see Huxley (2003). However, these methods do not apply to counting integer points (m, n) under a hyperbola if some additional restrictions are imposed on m and n that might be expressed as congruence conditions or properties of b-ary expansions (to some fixed base $b \geq 2$) or a combination of both. For such questions, Algorithm 1 may lead to effective probabilistic estimation algorithms.

Acknowledgments

The authors thank Richard Brent and Paul Zimmermann for hints about numerical approximations, and Konstantin Ziegler and Johannes Zollmann for useful discussions and permission to use some figures and tables they produced.

The first author's work was supported by the B-IT Foundation and the Land Nordrhein-Westfalen. The second author was supported in part by the Australian Research Council grant DP110100628 and Macquarie University grant MQRDG1465020.

References

R. C. Baker & G. Harman (1998). Shifted primes without large prime factors. *Acta Arithmetica* **83**(4), 331–361.

R. C. Baker, G. Harman & J. Pintz (2001). The difference between consecutive primes. II. *Proceedings of the London Mathematical Society* **83**(3), 532–562.

Paul T. Bateman & Roger A. Horn (1962). A heuristic asymptotic formula concerning the distribution of prime numbers. *Mathematics of Computation* **16**, 363–367.

L. Blum, M. Blum & M. Shub (1986). A simple unpredictable pseudo-random number generator. *SIAM Journal on Computing* **15**(2), 364–383. URL http://dx.doi.org/10.1137/0215025.

Richard P. Brent & Paul Zimmermann (2011). *Modern Computer Arithmetic*, volume 18 of *Applied and Computational Mathematics*. Cambridge University Press, Cambridge, UK.

Richard Crandall & Carl Pomerance (2005). *Prime numbers – A computational perspective*. Springer-Verlag, 2nd edition. ISBN 0-387-25282-7.

Ivan Damgård & Maciej Koprowski (2001). Practical Threshold RSA Signatures without a Trusted Dealer. In *Advances in Cryptology: Proceedings of EUROCRYPT 2001,* Aarhus, Denmark, volume 2045 of *Lecture Notes in Computer Science*, 152–165. Springer-Verlag, Berlin, Heidelberg. ISBN 3-540-42070-3. ISSN 0302-9743. URL http://dx.doi.org/10.1007/3-540-44987-6_10.

ANDREAS DECKER & PIETER MOREE (2008). Counting RSA-integers. *Results in Mathematics* **52**, 35–39. URL http://dx.doi.org/10.1007/s00025-008-0285-5.

LUC DEVROYE (1986). *Non-Uniform Random Variate Generation*. Springer-Verlag, New York. ISBN 0-387-96305-7. 843 pages.

PIERRE-ALAIN FOUQUE & JACQUES STERN (2001). Fully Distributed Threshold RSA under Standard Assumptions. In *Advances in Cryptology: Proceedings of ASIACRYPT 2001,* Gold Coast, Australia, volume 2248 of *Lecture Notes in Computer Science*, 310–330. Springer-Verlag, Berlin, Heidelberg. ISBN 3-540-42987-5. ISSN 0302-9743. URL http://dx.doi.org/10.1007/3-540-45682-1_19.

STEVEN GALBRAITH (2012). *Mathematics of Public Key Cryptography*. Cambridge University Press, New York. ISBN 978-1-107-01392-6. URL http://www.math.auckland.ac.nz/~sgal018/crypto-book/crypto-book.html.

JOACHIM VON ZUR GATHEN & JÜRGEN GERHARD (2003). *Modern Computer Algebra*. Cambridge University Press, Cambridge, UK, Second edition. ISBN 0-521-82646-2, 800 pages. URL http://cosec.bit.uni-bonn.de/science/mca/. Other available editions: first edition 1999, Chinese edition, Japanese translation.

ROSARIO GENNARO, SHAI HALEVI & TAL RABIN (1999). Secure Hash-and-Sign Signatures Without the Random Oracle. In *Advances in Cryptology: Proceedings of EUROCRYPT 1999,* Prague, Czech Republic, J. STERN, editor, volume 1592 of *Lecture Notes in Computer Science*, 123–139. Springer-Verlag, Berlin, Heidelberg. ISBN 3-540-65889-0. ISSN 0302-9743. URL http://www.springerlink.com/content/bryhef8g51vwbl10/?p=8d42fdca76d6472ca31db56d3b833c55&pi=1.

BAI-NI GUO & FENG QI (2011). Sharp bounds for harmonic numbers. *Applied Mathematics and Computation* **218**, 991–995. URL http://dx.doi.org/10.1016/j.amc.2011.01.089.

GLYN HARMAN (2007). *Prime-detecting sieves*. Princeton University Press, Princeton, NJ.

D. R. HEATH-BROWN (1986). Artin's conjecture for primitive roots. *Quarterly Journal of Mathematics* **37**, 27–38.

D. R. HEATH-BROWN (1988). The number of primes in a short interval. *Journal für die reine und angewandte Mathematik* **389**, 22–63.

DENNIS HOFHEINZ, EIKE KILTZ & VICTOR SHOUP (2012). Practical Chosen Ciphertext Secure Encryption from Factoring. *To appear in Journal of Cryptology* URL `http://www.shoup.net/papers/cca-fac.pdf`.

M. N. HUXLEY (1972). On the Difference between Consecutive Primes. *Inventiones mathematicae* **15**, 164–170.

M. N. HUXLEY (2003). Exponential sums and lattice points III. *Proceedings of the London Mathematical Society* **87**(3), 591–609.

H. IWANIEC & E. KOWALSKI (2004). *Analytic Number Theory*. American Mathematical Society, Providence, RI.

MARC JOYE & PASCAL PAILLIER (2006). Fast Generation of Prime Numbers on Portable Devices: An Update. In *Cryptographic Hardware and Embedded Systems, Workshop, CHES'06,* Yokohama, Japan, LOUIS GOUBIN & MITSURU MATSUI, editors, volume 4249 of *Lecture Notes in Computer Science*, 160–173. Springer-Verlag, Berlin, Heidelberg. ISBN 978-3-540-46559-1. ISSN 0302-9743. URL `http://dx.doi.org/10.1007/11894063_13`.

DONALD E. KNUTH (1981). *The Art of Computer Programming, vol.2, Seminumerical Algorithms*. Addison-Wesley, Reading MA, 2nd edition.

DANIEL LOEBENBERGER & MICHAEL NÜSKEN (2013). Notions for RSA integers. *To appear in International Journal of Applied Cryptography,* 23 pages. ISSN 1753-0571 (online), 1753-0563 (print). URL `http://arxiv.org/abs/1104.4356`.

ALFRED J. MENEZES, PAUL C. VAN OORSCHOT & SCOTT A. VANSTONE (1997). *Handbook of Applied Cryptography*. CRC Press, Boca Raton FL. ISBN 0-8493-8523-7. URL `http://www.cacr.math.uwaterloo.ca/hac/`.

DAVID NACCACHE (2003). Double-Speed Safe Primes Generation. *Cryptology ePrint Archive* **2003/175**, 2 pages.

Takashi Nishide & Kouichi Sakurai (2011). Distributed Paillier Cryptosystem without Trusted Dealer. In *Information Security Applications*, Yongwha Chung & Moti Yung, editors, volume 6513 of *Lecture Notes in Computer Science*, 44–60. Springer-Verlag, Berlin, Heidelberg. ISBN 3-642-17954-1. ISSN 0302-9743. URL http://dx.doi.org/10.1007/978-3-642-17955-6_4.

Emil Ong & John Kubiatowicz (2005). Optimizing Robustness while Generating Shared Secret Safe Primes. In *8th International Workshop on Practice and Theory in Public Key Cryptography,* Switzerland, Serge Vaudenay, editor, volume 3386 of *Lecture Notes in Computer Science*, 120–137. Springer-Verlag. URL http://dx.doi.org/10.1007/b105124.

J. Barkley Rosser & Lowell Schoenfeld (1962). Approximate formulas for some functions of prime numbers. *Illinois Journal of Mathematics* **6**, 64–94.

Victor Shoup (2000). Practical Threshold Signatures. In *Advances in Cryptology: Proceedings of EUROCRYPT 2000,* Bruges, Belgium, B. Preneel, editor, volume 1807 of *Lecture Notes in Computer Science*, 207–220. Springer-Verlag, Berlin, Heidelberg. ISBN 3-540-67517-5. ISSN 0302-9743. URL http://dx.doi.org/10.1007/3-540-45539-6_15.

Terence Tao, Ernest Croot & Harald Helfgott (2012). Deterministic methods to find primes. *Mathematics of Computation* **81**(278), 1233–1246.

Gérald Tenenbaum (1995). *Introduction to analytic and probabilistic number theory.* Number 46 in Cambridge studies in advanced mathematics. Cambridge University Press, Cambridge, UK.

А. И. Виноградов (1963). Об остатке в формуле Мертенса (A. I. Vinogradov, On the remainder in Mertens' formula). *Dokl. Akad. Nauk SSSR* **148**(2), 262–263. (Russian).

Benne de Weger (2008). Estimates for RSA Moduli Counting Functions. Preprint, March 20, 2008.

Konstantin Ziegler & Johannes Zollmann (2013). Fast and uniform generation of safe RSA moduli — Extended Abstract. In *WEWoRC 2013*

— *Conference Records*, 15–19. Karlsruher Institut für Technologie (KIT), Karlsruhe. URL http://2013.weworc.eu/docs/program_full.pdf.