

Grained integers and applications to cryptography

DISSERTATION

zur

Erlangung des Doktorgrades (Dr. rer. nat.)

der

Mathematisch-Naturwissenschaftlichen Fakultät

der

Rheinischen Friedrich-Willhelms-Universität Bonn

vorgelegt von

Daniel Loebenberger

aus

Nürnberg

Bonn, Januar 2012

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen
Fakultät der Rheinischen Friedrich-Wilhelms-Universität Bonn

1. Gutachter: Prof. Dr. Joachim von zur Gathen

2. Gutachter: Prof. Dr. Andreas Stein

Tag der Promotion: 16.05.2012

Erscheinungsjahr: 2012

Οἱ πρῶτοι ἀριθμοὶ πλείους εἰσὶ παντὸς τοῦ
 προτένθους πλήθους πρώτων ἀριθμῶν.¹
 (EUCLID)

Problema, numeros primos a compositis dignoscendi,
 hosque in factores suos primos resolvendi, ad gravissima
 ac ultissima totius arithmeticae pertinere [...] tam notum est,
 ut de hac re copiose loqui superfluum foret. [...] Praetereaque
 scientiae dignitas requirere videtur, ut omnia subsidia ad solutionem
 problematis tam elegantis ac celeberrimi sedulo excolantur.²
 (C. F. GAUSS)

L'algèbre est g n reuse, elle donne souvent plus qu'on lui demande.³
 (J. D'Alembert)

¹Die Menge der Primzahlen ist gr o er als jede vorgelegte Menge von Primzahlen.

²Dass die Aufgabe, die Primzahlen von den zusammengesetzten zu unterscheiden und letztere in ihre Primfaktoren zu zerlegen, zu den wichtigsten und n tzlichsten der gesamten Arithmetik geh rt und die Bem hungen und den Scharfsinn sowohl der alten wie auch der neueren Geometer in Anspruch genommen hat, ist so bekannt, dass es  berfl ssig w re, hier ber viele Worte zu verlieren. [...] Weiter d rfte es die W rde der Wissenschaft erheischen, alle Hilfsmittel zur L sung jenes so eleganten und ber hmten Problems fleissig zu vervollkommen.

³Die Algebra ist gro z gig, sie gibt h ufig mehr als man von ihr verlangt.

Selbständigkeitserklärung

Ich versichere, dass ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Daniel Loebenberger
Bonn, den 07.01.2012

Zusammenfassung

Um den Ansprüchen der modernen Kommunikationsgesellschaft gerecht zu werden, ist es notwendig, kryptographische Techniken gezielt einzusetzen. Dabei sind insbesondere solche Techniken von Vorteil, deren Sicherheit sich auf die Lösung bekannter zahlentheoretischer Probleme reduzieren lässt, für die bis heute keine effizienten algorithmischen Verfahren bekannt sind. Demnach führt jeglicher Einblick in die Natur eben dieser Probleme indirekt zu Fortschritten in der Analyse verschiedener, in der Praxis eingesetzten kryptographischen Verfahren.

In dieser Arbeit soll genau dieser Aspekt detaillierter untersucht werden: Wie können die teils sehr anwendungsfernen Resultate aus der reinen Mathematik dazu genutzt werden, völlig praktische Fragestellungen bezüglich der Sicherheit kryptographischer Verfahren zu beantworten und konkrete Implementierungen zu optimieren und zu bewerten? Dabei sind zwei Aspekte besonders hervorzuheben: Solche, die Sicherheit gewährleisten, und solche, die von rein kryptanalytischem Interesse sind.

Nachdem wir — mit besonderem Augenmerk auf die historische Entwicklung der Resultate — zunächst die benötigten analytischen und algorithmischen Grundlagen der Zahlentheorie zusammengefasst haben, beschäftigt sich die Arbeit zunächst mit der Fragestellung, wie die Punktaddition auf elliptischen Kurven spezieller Form besonders effizient realisiert werden kann. Die daraus resultierenden Formeln sind beispielsweise für die Kryptanalyse solcher Verfahren von Interesse, für deren Sicherheit es notwendig ist, dass die Zerlegung großer Zahlen einen hohen Berechnungsaufwand erfordert. Der Rest der Arbeit ist solchen Zahlen gewidmet, deren Primfaktoren nicht zu klein, aber auch nicht zu groß sind. Inwiefern solche Zahlen in natürlicher Art und Weise in kryptographischen und kryptanalytischen Verfahren auftreten und wie deren Eigenschaften für die Beantwortung sehr konkreter, praktischer Fragestellungen eingesetzt werden können, wird anschließend anhand von zwei Anwendungen diskutiert: Der Optimierung einer Hardware-Realisierung des Kofaktorisierungsschrittes des allgemeinen Zahlkörpersiebs, sowie der Analyse verschiedener, standardisierter Schlüssel-erzeugungsverfahren.

Synopsis

To meet the requirements of the modern communication society, cryptographic techniques are of central importance. In modern cryptography, we try to build cryptographic primitives whose security can be reduced to solving a particular number theoretic problem for which no fast algorithmic method is known by now. Thus, any advance in the understanding of the nature of such problems indirectly gives insight in the analysis of some of the most practical cryptographic techniques.

In this work we analyze exactly this aspect much more deeply: How can we use some of the purely theoretical results in number theory to answer very practical questions on the security of widely used cryptographic algorithms and how can we use such results in concrete implementations? While trying to answer these kinds of security-related questions, we always think two-fold: From a cryptographic, security-ensuring perspective and from a cryptanalytic one.

After we outlined — with a special focus on the historical development of these results — the necessary analytic and algorithmic foundations of number theory, we first delve into the question how point addition on certain elliptic curves can be done efficiently. The resulting formulas have their application in the cryptanalysis of crypto systems that are insecure if factoring integers can be done efficiently. The rest of the thesis is devoted to the study of integers, all of whose prime factors are neither too small nor too large. We show with the help of two applications how one can use the properties of such kinds of integers to answer very practical questions in the design and the analysis of cryptographic primitives: The optimization of a hardware-realization of the cofactorization step of the General Number Field Sieve and the analysis of different standardized key-generation algorithms.

Acknowledgments

The results in this thesis would not be the same without the long discussions I had with my friends and colleagues of the working group *cosec* at the Bonn-Aachen International Center for Information Technology. In particular, I want to thank Dr. Michael Nüsken for guiding me through some of the imponderabilities when working on the results presented in this thesis and for adding his invaluable experience to the work I was doing. Additionally, I would like to thank my supervisor, Prof. Dr. Joachim von zur Gathen, for giving me the opportunity to work on the topic presented here, by pointing me to interesting projects related to this work, and for selecting the right people that lead to the good working climate at *cosec*.

Special thanks go to Konstantin Ziegler, who introduced me to the computer algebra system **sage**, a tool I really learned to appreciate, and to Yona Raekow and Cláudia Oliveira Coelho for always being there for me when I needed someone to talk to. I would also like to thank Dr. Jérémie Detrey for the long fruitful discussions we had while he was working at *cosec* and for his invaluable comments on various topics. I also want to thank my former supervisor Dr. Helmut Meyn for his support.

Last but not least I would like to thank Martin Hielscher and the rest of my family for supporting me all the time. I dedicate this thesis to Dr. Herta Bergmann, to Liselotte Ammon and in particular to Dr. Fotini Pfab.

Daniel Loebenberger
07.01.2012

Contents

1	Introduction	1
1.1	Development of the results	1
1.2	Structure of the thesis	3
2	Prime numbers	5
2.1	The distribution of primes	5
2.2	More on analytic number theory	12
2.3	Counting other classes of integers	22
3	Algorithmic number theory	27
3.1	Basic algorithms	27
3.2	Newton: Recognizing perfect powers	37
3.3	Primality testing	39
3.4	Factoring algorithms by sieving	51
3.5	Factoring algorithms using elliptic curves	57
4	Differential addition in Edwards form	69
4.1	State of the art	69
4.2	Edwards form	72
4.3	Representing points in Edwards form	72
4.4	A tripling formula	75
4.5	Recovering the x -coordinate	76
4.6	A parametrization using squares only	78
5	Public key cryptography	81
5.1	Diffie and Hellman: New directions in cryptography	81
5.2	Doing it: RSA	82
5.3	The ubiquity of grained integers	83
6	Coarse-grained integers	85
6.1	The recursion	88
6.2	Using estimates	90

6.3	Approximations	93
6.4	Solving the recursion for $\tilde{\lambda}^k$	96
6.5	Estimating the estimate $\hat{\lambda}^k$	110
6.6	Reestimating $\hat{\lambda}^k$ without Riemann	114
6.7	Improvements	116
6.8	Non-squarefree numbers are negligible	120
6.9	Results on coarse-grained integers	122
6.10	Numeric evaluation	125
7	Hardware for the GNFS	129
7.1	Framework	129
7.2	Modelling the cluster system	130
7.3	Concrete statistical analyses	134
7.4	Generalizations to an arbitrary number of clusters	137
7.5	Connection to the theoretical results	139
8	RSA integers	141
8.1	Framework	141
8.2	RSA integers in general	143
8.3	Toolbox	146
8.4	Some common definitions for RSA integers	156
8.5	Arbitrary notions	162
8.6	Complexity theoretic considerations	166
9	Generalized RSA integers	171
9.1	Framework and toolbox	172
9.2	Some results	173
10	Standards for RSA integers	177
10.1	Generating RSA integers properly	177
10.2	Output entropy	182
10.3	Information-theoretical efficiency	185
10.4	Impact on standards and implementations	186
11	Future work and open problems	195
	Bibliography	197
	Players	213
	Index	221

Chapter 1

Introduction

1.1. Development of the results

The results in this thesis originated from a project the working group *cosec* jointly ran between 2006 and 2008 together with the University of Bochum, the University of Duisburg-Essen and the Siemens AG Munich on the factorization of large integers, funded by the Bundesamt für Sicherheit in der Informationstechnik (BSI). The goal was to analyze whether highly specialized hardware clusters like the COPACOBANA could be used for more efficient realizations of several parts of the General Number Field Sieve such as the cofactorization step or the linear algebra step.

Since the working group in Bochum realized together with the University of Kiel the specific implementation aspects of the hardware cluster (later published in Güneysu *et al.* 2008), our obligation was to find a way to optimize their implementation without seeing and without touching it. We asked Thorsten Kleinjung, who held the factoring record at that time together with Jens Franke (see Franke & Kleinjung 2005), to send us a list of sample inputs to the cofactorization step. Since these samples were to be fed into an elliptic curve factorization algorithm that we were supposed to optimize, we were hoping that some experiments with the data would point us to the right direction. To our surprise the inputs followed a certain highly non-uniform distribution, which enabled us to reduce the runtime of the hardware implementation by roughly 20%, given only the premise that the modules used are scalable (see Chapter 7).

After the successful optimization, we continued the work on the intermediate step of the General Number Field Sieve and figured out that the structure of the distribution was closely related to the count of integers that have prime factors from a certain interval only. The lower and upper bounds

on the size of the prime factors were specified by the concrete implementation of the General Number Field Sieve and the choice of the sieving method, see Franke & Kleinjung (2006). As it turned out, these parameter choices did not only imply bounds on the size of the prime factors but also on the *number* of prime factors.

There are many results in analytic number theory on *smooth* integers, i.e. integers with small prime factors only. The article Granville (2008) summarizes the state of the art on the analysis of such integers nicely. The dual problem on the count of *rough* integers, that are integers with large prime factors only, was shortly mentioned there, but it seemed that this problem did draw to it much less attention. Also it seemed that no one had ever considered the combined problem – counting integers that are simultaneously smooth and rough — we called such kind of integers *grained*, in analogy to the existing notions of smooth and rough. This motivated further studies in this direction.

It turned out that we were able to solve the counting problem in a satisfactory way (see Chapter 6). However, we observed that a very special case of our results were proper definitions for RSA integers which one could also find in some textbooks. About the same time Decker & Moree (2008) published an article on the count of RSA integers but following a completely different definition for those kind of integers. Puzzled by this discrepancy of our results and their results, we started analyzing in 2010 how *all possible* definitions for RSA integers compare to each other, peaking in the proof of a conjecture Benne de Weger stated in 2009, saying that the count of such integers is closely related to the area of the region the prime factors are taken from. A further interesting aspect of our work was to actually find out which kinds of standards and implementations use which of the definitions for RSA integers. Since our theoretical results were able to give precise estimates on the count of such integers, it turned out that the same results together with some very general observations on the algorithmic aspects enabled us to estimate the output entropy of all possible kinds of RSA key-generators. We knew that such kind of estimates have already been known for various types of prime number generators, see for example Brandt & Damgård (1993) or Joye & Paillier (2006), but it seemed we had been the first ones that successfully adapted the techniques to RSA key-generators (see Chapter 8 to Chapter 10).

While we were working on the project on factoring large integers, Harold M. Edwards published a groundbreaking article in 2007 by introducing a new normal form for elliptic curves. Since in our project the Elliptic Curve Method was employed for factoring moderately large integers in the cofac-

torization step of the General Number Field Sieve, a natural question was how one could use Edwards's ideas to obtain further speed-up. Montgomery (1987) showed how to employ certain representations of points on ordinary elliptic curves to obtain highly efficient arithmetic, but for elliptic curves in Edwards form there were little results in this direction. Thus, naturally, it occurred to us that a similar approach could lead to nice results in this direction for the new kind of elliptic curves. Unfortunately, we were not the first ones with this idea: Castryck *et al.* (2008) showed that the so-called *differential addition* on elliptic curves in Edwards form could be realized efficiently when a particular curve parameter equalled one. Actually, we were able to extend the results, but it turned out that using very heavy machinery Gaudry & Lubicz (2009) were able to obtain similar results using Riemann ϑ functions – a branch of number theory quite inaccessible to us. Due to the different kind of framework, our findings gave a much more elementary derivation for differential addition on elliptic curves in generalized Edwards form (see Chapter 4).

1.2. Structure of the thesis

In Chapter 2 and Chapter 3 we lay the needed mathematical and algorithmic foundations of number theory. Here, we focus in particular on the historical development of the different techniques, to be able to understand better how the important concepts around nowadays evolved over the past centuries. In Chapter 4 we analyze more deeply differential addition on elliptic curves in (generalized) Edwards form.

Afterwards, we give a short overview of the *cryptographic* concepts in Chapter 5. Chapter 6 is devoted to the number-theoretic study of coarse-grained integers: By employing explicit results on the number of primes not exceeding a given bound, we will obtain estimates of this type on the count of coarse-grained integers. Clearly, while doing so, we need to analyze carefully the error we make in our approximations.

In the subsequent Chapters 7 to 10, we study some example applications in which coarse-grained integers occur naturally and answer some very practical questions in the design and the analysis of cryptographic primitives. More specifically, we first study in Chapter 7 how we can use the specific distribution of the inputs to the cofactorization step in the General Number Field Sieve to obtain a considerable speed-up when realizing this step using resizable hardware modules.

Another application of some of the results on coarse-grained integers is the comparative study of all (reasonable) definitions for RSA integers in

Chapter 8. We propose a model that is able to capture the number-theoretic properties that we will later need for the analysis of concrete standards and implementations.

A slight generalization of the results from Chapter 8 will shortly be discussed in Chapter 9: Instead of ordinary RSA integers (that are the product of two different primes), we sketch how one can use our techniques to tackle integers that have exactly two distinct prime factors (one might want to call them *generalized RSA integers*). Such integers have their application in some fast variants of RSA or the Okamoto-Uchiyama cryptosystem.

In Chapter 10 we employ our results from Chapter 8 to analyze several concrete RSA key-generators (as specified in relevant standards and implementations) to obtain estimates on the entropy of the output distribution and the (information-theoretical) efficiency. We finish the chapter with a thorough comparison of the obtained results.

After discussing some open problems and future work in Chapter 11, we finish with a bibliography, a list of historically relevant people and an index.

Chapter 2

Prime numbers

2.1. The distribution of primes

Prime numbers are the basis of all arithmetic. Even though the concept of primality was already known in the ancient world, there are still many unsolved problems concerning them. The Riemann hypothesis (see Section 2.2.2) as part of problem 8 of Hilbert's list of 23 unsolved problems, stated in 1900, and one of the seven millennium problems of the Clay Mathematics Institute (2000), is just the tip of the iceberg. On the other side — especially from an algorithmic aspect — there was also considerable progress during the 20th century. We will describe now the most important results on prime numbers, starting from results from the ancient world up to some explicit estimates concerning primes in arithmetic progressions. We follow in the exposition of this chapter mainly Crandall & Pomerance (2005). Most of the historical facts are taken from the amazing little book Edwards (1974).

2.1.1. Euclid and Eratosthenes: Results from the ancient world.

Prime numbers were already studied in the ancient world. In fact, one of the most important theorems in arithmetic — the fundamental theorem of arithmetic — has its roots also in these times: it says that every natural number has a unique prime factorization. The proof of this theorem naturally comes in two steps: First, the *existence* has to be proven. Yet, this is simple to show: take the smallest number n that does not have a prime factorization. Since the number 1 and any prime do have a prime factorization, n can be written as a product of smaller numbers that by assumption do have a prime factorization. But then, also n has one by combining the prime factorizations of the two factors found. For *uniqueness* a theorem already

known to Εὐκλείδης ὁ Ἀλεξάνδρεια (365–300 BC)⁴ can be employed:

THEOREM 2.1.1 (Euclid Elements, book VII, proposition 31). *Let m, n be two integers. If a prime p divides $m \cdot n$, then p divides either m or n (or both).*

PROOF. Suppose p divides $m \cdot n$ but does not divide m . We need to show that then p divides n . As p does not divide m and p is prime, there are integers $s, t \in \mathbb{Z}$ such that $sp + tm = 1$, see Bézout's Identity 3.1.7. By multiplying this by n we get $spn + tmn = n$. Since p divides $m \cdot n$, it also divides tmn and thus also n . \square

From this we can deduce the

FUNDAMENTAL THEOREM OF ARITHMETIC 2.1.2. *Let n be a natural number. Then there is a unique prime factorization*

$$n = p_1 p_2 \cdots p_k,$$

where $p_1 \leq p_2 \leq \cdots \leq p_k$ are (not necessarily distinct) prime numbers.

PROOF. We have shown the existence of a prime factorization above. For uniqueness, consider the smallest counterexample, i.e. the smallest number n that does have at least two different prime factorizations $n = p_1 p_2 \cdots p_k = q_1 q_2 \cdots q_\ell$. Both q_1 and $q_2 \cdots q_\ell$ must have unique prime factorizations due to the minimality of n . By Theorem 2.1.1, p_1 divides q_1 or p_1 divides $q_2 \cdots q_\ell$. Therefore $p_1 = q_i$ for some $1 \leq i \leq \ell$. Removing p_1 and q_i from the two prime factorizations gives now a smaller natural number $n' = n/p_1$ with at least two different prime factorizations, contradicting the minimality of n . \square

The Fundamental Theorem of Arithmetic 2.1.2 has an algorithmic analog (one could call it the fundamental *problem* of arithmetic), namely the

FACTORIZATION PROBLEM 2.1.3. *Given a natural number $n \in \mathbb{N}_{\geq 2}$, find its prime factorization.*

Algorithms that solve the problem are called *factorization algorithms*. Note that it is not known up to date whether the problem can be solved in (probabilistic) polynomial time. For a more thorough discussion, see Section 3.4 and Section 3.5.

Euclid was also able to answer the question concerning the number of primes. He proved essentially

⁴Euclid of Alexandria

THEOREM 2.1.4 (Euclid Elements, book IX, proposition 20). *There are infinitely many primes.*

PROOF. Assume there are only finitely many primes p_1, \dots, p_k . Then the number $n = 1 + p_1 p_2 \cdots p_k$ is not divisible by any of the primes p_1, \dots, p_k . Thus, by the Fundamental Theorem of Arithmetic 2.1.2, the integer n has a prime factor that is different from p_i for all $1 \leq i \leq k$. \square

The theorem gives rise to the following algorithmic problem:

PROBLEM 2.1.5. *Given an integer $n \in \mathbb{N}_{\geq 2}$, decide whether n is prime.*

An algorithmic method that decides this problem is called a *primality test*. A very simple algorithm, going back to Ἐρατοσθένης ὁ Κυρηνάιος (276–194 BC)⁵, is to perform trial division for all primes up to \sqrt{n} . This is, of course, highly inefficient, as the number of necessary arithmetic operations is exponential in the size of n . More practical methods are to be discussed in Section 3.3.

The first person that seemed to be aware of the fact that the Factorization Problem 2.1.3 and Problem 2.1.5 are indeed two fundamentally different problems was François Édouard Anatole Lucas (1842–1891), who remarked in 1878 that

“Cette méthode de vérification des grands nombres premiers, qui repose sur le principe que nous venons de démontrer, est la seule méthode directe et pratique, connue actuellement, pour résoudre le problème en question; elle est opposée, pour ainsi dire, à la méthode de vérification d’Euler.”⁶

The principle Lucas was referring to is the following: Consider an easily checkable condition that holds for all prime numbers and only few composites. Then given some integer n , one can simply check if the condition holds for n . If it does not, we can be sure that n is composite, otherwise we might think (even though it is not proven) that n is prime. Indeed, this is the approach that all *practical* primality tests employ nowadays, see Section 3.3. It also gives an efficient method for *finding* a large prime of a given length by selecting repeatedly an integer of that length until our test finds a number that looks like a prime. For this procedure to be efficient

⁵Eratosthenes of Cyrene

⁶This method of verification of large prime numbers, based on the principle that we have just demonstrated, is the only direct and practical method currently known to solve the problem in question and it is opposed, so to speak, to Euler’s verification method.

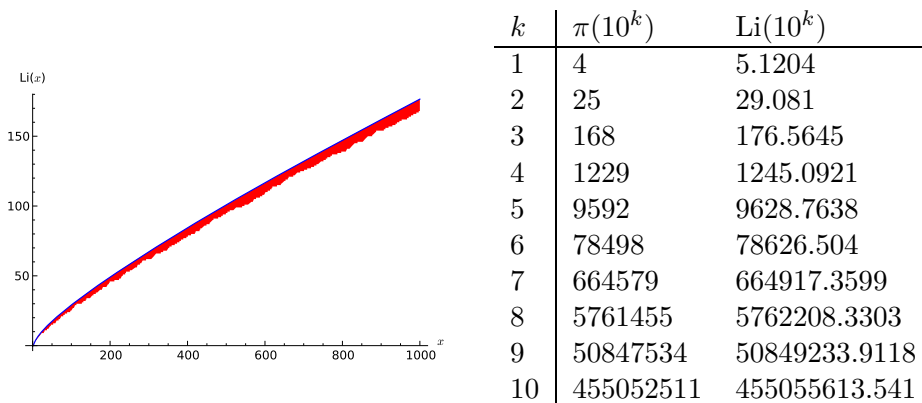


Figure 2.1.1: The left-hand picture shows the values of the logarithmic integral (blue) in comparison to the prime counting function (red). The right-hand table the first few values of the prime counting function and the logarithmic integral at powers of ten.

it is required that the prime numbers lie somewhat dense in the set of all integers. What can we say about that?

2.1.2. Gauß and the prime number theorem. For a long time there was little known about the number of primes up to a real bound x , traditionally denoted by $\pi(x)$. In 1737, Leonhard Paul Euler (1707–1783) gave a groundbreaking proof that there are infinitely many primes, by showing that the sum of reciprocals of primes diverges (see Section 2.2). This was the corner stone of rigorous analytic number theory.

Around 1792, Johann Carl Friedrich Gauß (1777–1855) conjectured, while being still a teenager, that the asymptotic behavior of the prime counting function is asymptotically equal to

$$\pi(x) \approx \frac{x}{\ln x}$$

or more precisely

$$\pi(x) \approx \text{Li}(x) := \int_2^x \frac{1}{\ln t} dt,$$

where $\text{Li}(x)$ is called the *logarithmic integral*, see Figure 2.1.1. This conjecture is now known as the famous prime number theorem (PNT).

It was, however, not published until 1849, when Gauß wrote the result in a letter to Johann Franz Encke (1791–1865). Independently, a similar conjecture, namely $\pi(x) \approx \frac{x}{\ln x - A}$, for a constant A close to one, was given

in 1830 by Adrien-Marie Legendre (1752–1833). It is striking to observe that Gauß' conjecture is by far the better one. Indeed, it subsumes Legendre's formula, since by Taylor expansion we have $\text{Li}(x) = \frac{x}{\ln x} + \frac{x}{\ln^2 x} + \mathcal{O}\left(\frac{x}{\ln^3 x}\right)$ and $\frac{x}{\ln x} \approx \frac{x}{\ln x - A}$ for all real constants A .

Roughly twenty years later, Пафну́тий Льво́вич Чебышёв (1821–1894)⁷ proved a theorem that was already a big step in the direction of the prime number theorem:

THEOREM 2.1.6 (Чебышёв 1852). *There exist real constants B and C such that for all $x \geq 3$ we have*

$$\frac{Bx}{\ln x} < \pi(x) < \frac{Cx}{\ln x}.$$

The question was finally resolved in 1896 independently by Jacques Salomon Hadamard (1865–1963) and Charles-Jean Étienne Gustave Nicolas, Baron de la Vallée Poussin (1866–1962), more than one century after Gauß' conjecture:

THEOREM 2.1.7 (Hadamard 1896, de la Vallée Poussin 1896). *We have for x tending to infinity*

$$\pi(x) \approx \frac{x}{\ln x}.$$

More precisely, we have

$$\pi(x) \in \text{Li}(x) + \mathcal{O}\left(xe^{-C\sqrt{\ln x}}\right). \quad \square$$

The prime counting function was successively refined and comes nowadays in many different variants (see Figure 2.1.2), which we sum up in the

PRIME NUMBER THEOREM 2.1.8. *There are the following results on the distribution of primes.*

- (i) *Hadamard (1896), de la Vallée Poussin (1896) and Walfisz (1963), conjectured by Gauß (1849):*

$$\pi(x) \in \text{Li}(x) + \mathcal{O}\left(x \exp\left(-\frac{A(\ln x)^{3/5}}{(\ln \ln x)^{1/5}}\right)\right) \subset \text{Li}(x) + \mathcal{O}\left(\frac{x}{\ln^k x}\right)$$

for any k . The presently best known value for A is $A = 0.2098$ (Ford 2002a, p. 566). Here, the logarithmic integral Li is given by $\text{Li}(x) := \int_2^x \frac{dt}{\ln t}$.

⁷Pafnuty Lvovich Chebyshev

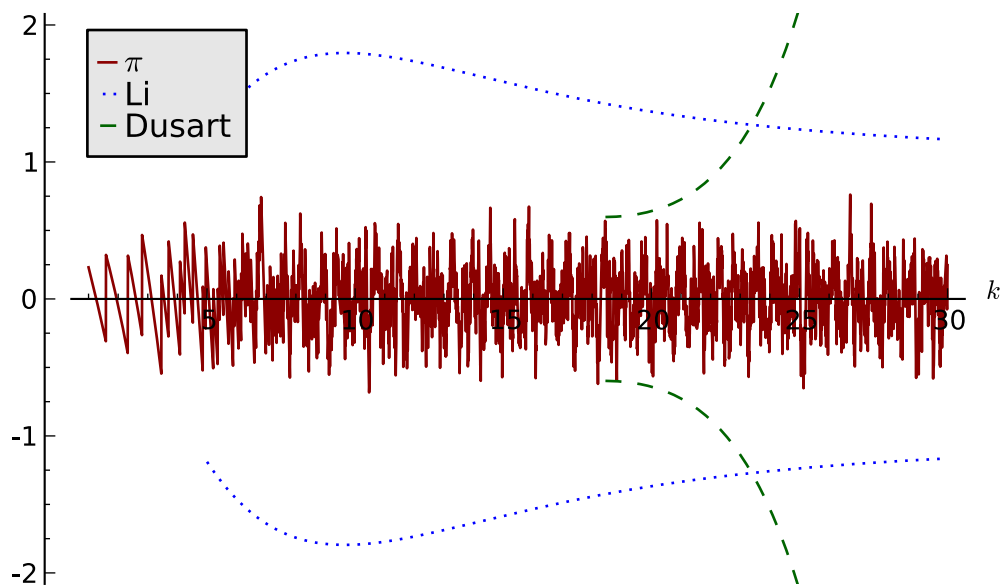


Figure 2.1.2: Various results on the distribution of primes. The red line shows a normalized variant of the prime counting function $\pi(2^k)$. The blue dotted line shows the same normalization of Gauß' approximation using the logarithmic integral (Prime Number Theorem 2.1.8(i)). The unconditional Dusart bound (Prime Number Theorem 2.1.8(ii)) is shown by the dotted green line. It was proven by Littlewood (1914) that the red line crosses the blue one infinitely often.

(ii) Dusart (1998, Théorème 1.10, p. 36): For $x > 355\,991$ we have

$$\frac{x}{\ln x} + \frac{x}{\ln^2 x} < \pi(x) < \frac{x}{\ln x} + \frac{x}{\ln^2 x} + 2.51 \frac{x}{\ln^3 x}.$$

(iii) Von Koch (1901), Schoenfeld (1976): If (and only if) the Riemann hypothesis holds then for $x \geq 1451$ we have

$$|\pi(x) - \text{Li}(x)| < \frac{1}{8\pi} \sqrt{x} \ln x.$$

□

2.1.3. Dirichlet's theorem for arithmetic progressions. Once there were results on the density of primes, a reasonable step was to consider questions on the density of *primes with certain properties*. In fact, even today there are still many open problems in this direction. One particular problem of historical relevance was the question how primes that lie in a particular residue class a modulo a natural number $m \in \mathbb{N}_{\geq 2}$ are distributed. Clearly, if a and m have a common prime factor, then this prime number divides every element of the residue class, and the class can contain at most this single prime. The proof that all other classes contain infinitely many primes was given by Johann Peter Gustav Lejeune Dirichlet (1805–1859):

THEOREM 2.1.9 (Dirichlet 1837). *If a and m are integers without common prime factor, then there are infinitely many primes in the arithmetic progression $\{a, a + m, a + 2m, \dots\}$.* □

In modern days there were many more results on primes in arithmetic progressions, like the following famous theorem Arnold Walfisz (1892–1962) proved in 1936, based on previous work of Carl Ludwig Siegel (1896–1981). The statement involves an important number-theoretic function dating back to Euler:

DEFINITION 2.1.10 (Euler φ -function). *For an integer $m \geq 2$ we write $\varphi(m)$ to denote the number of integers in the set $\{0, \dots, m - 1\}$ that are coprime to m .*

We are ready to state:

THEOREM 2.1.11 (Siegel-Walfisz). *Write $\pi_{a+m\mathbb{Z}}(x)$ for the number of primes up to a real bound x that are in the residue class a modulo m . Then, for*

any real $\eta > 0$ there exists a positive real $C(\eta)$, such that for all coprime natural numbers a, m with $m < \ln^\eta x$ we have

$$\pi_{a+m\mathbb{Z}}(x) \in \frac{1}{\varphi(m)} \text{Li}(x) + \mathcal{O}\left(x \exp(-C(\eta)\sqrt{\ln x})\right).$$

In this expression, $\varphi(m)$ denotes the Euler φ -function (see Definition 2.1.10) and the constant hidden in the big- \mathcal{O} notation is absolute. \square

2.2. More on analytic number theory

As mentioned at the beginning of Section 2.1.2, Euler proved the infinitude of the number of primes, by establishing what is nowadays known as the *Euler product formula*. It relates the function

$$(2.2.1) \quad \zeta(s) := \sum_{n=1}^{\infty} \frac{1}{n^s}$$

to a product over prime numbers only (Euler, of course, took the value of the variable s to be real):

EULER PRODUCT FORMULA 2.2.2 (Euler 1737). For $\Re(s) > 1$ we have

$$(2.2.3) \quad \zeta(s) = \prod_{p \text{ prime}} \frac{1}{1 - p^{-s}}.$$

PROOF. By expanding each factor of the right-hand side of the formula above into a geometric series, we have

$$\frac{1}{1 - p^{-s}} = 1 + p^{-s} + p^{-2s} + \dots.$$

For $\Re(s) > 1$, we have $|p^{-s}| < 1$, and the series converges absolutely. Multiplying all those factors gives terms of the form $\prod_{p \text{ prime}} p^{-e(p)s}$, where each $e(p)$ is either zero or a positive integer, and all but finitely many $e(p)$ are non-zero. Thus, by the Fundamental Theorem of Arithmetic 2.1.2, each term is of the form n^{-s} for some natural number n , and each n occurs exactly once in the expanded product. \square

Euler then took the argument further, showing (by using the product formula above) that the sum of the reciprocals of primes up to a bound x diverges like $\ln \ln x$. Inspired by Euler's theorem, Georg Friedrich Bernhard

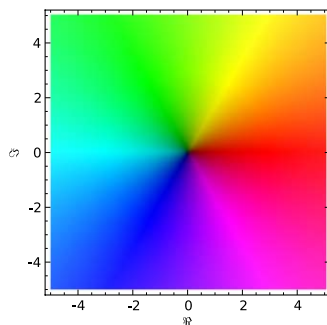


Figure 2.2.1: The complex coloring of the complex plots, i.e. the complex plot of the identity function. The absolute value of the output is indicated by the brightness (with zero being black and infinity being white), while the argument is represented by the hue.

Riemann (1826–1866) managed in the mid 19th century to introduce complex analysis to number theory, laying the foundations for *analytic number theory*: His brilliant idea was to allow the zeta function (2.2.1) to attain *complex* values. This allows to understand properties of the (by nature *discrete*) set of primes, by employing methods from an area that studies purely *continuous* objects. For example, he related in his seminal work from 1859 the zeros of the zeta function to the distribution of primes, leading to the famous, and still unproven, Riemann hypothesis, see Section 2.2.2.

2.2.1. Riemann’s zeta function. As mentioned above, Riemann (as one of the founders of complex analysis) naturally considered the zeta function as a function in a *complex* variable s . Clearly, in the half-plane $\Re(s) > 1$, both sides of the Euler Product Formula 2.2.2 converge. One of Riemann’s great achievements was to realize that the function they define is meaningful for *all* s (even though both sides of (2.2.3) diverge for $\Re(s) > 1$), except for a pole at $s = 1$ (then the left hand side in (2.2.1) is nothing but the harmonic series). To be able to extend the range for s , we first need some basic facts about another famous function Euler introduced in 1730. It is an extension of the factorial function $n! = 1 \cdot 2 \cdots (n - 1) \cdot n$, defined for non-negative integers n , to all real numbers greater than -1 via the equality

$$n! = \int_0^{\infty} e^{-x} x^n dx.$$

It holds for all non-negative integers n , and can be proven by integration by parts. Euler observed that the integral converges also for real values n ,

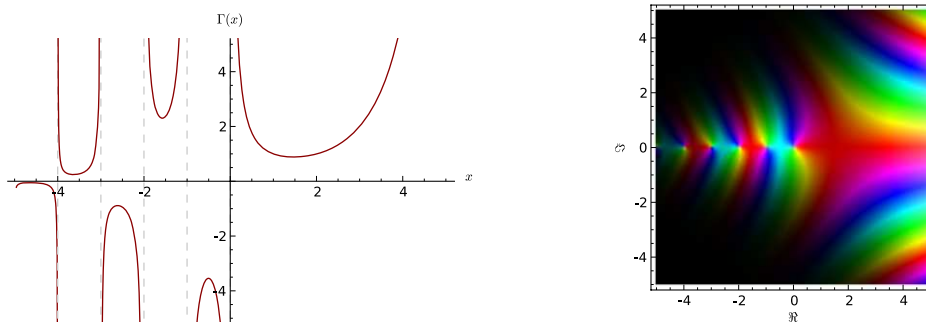


Figure 2.2.2: Plots of the gamma function. The left-hand picture shows the values of the gamma function on the real axis, the right-hand one a complex plot of the function with complex coloring defined in Figure 2.2.1.

provided $n > -1$. This leads to the definition of the function

$$(2.2.4) \quad \Gamma(s+1) = \int_0^{\infty} e^{-x} x^s \, dx$$

whose analytic continuation to all complex numbers is called the *gamma function*. Some plots of the gamma function can be found in Figure 2.2.2. We state

LEMMA 2.2.5 (Properties of the Gamma function). *We have for $s > -1$*

- (i) $\Gamma(s) = \lim_{n \rightarrow \infty} \frac{1 \cdot 2 \cdots n}{s \cdot (s+1) \cdots (s+n)} n^s$.
- (ii) $\Gamma(1+s) = s\Gamma(s)$,
- (iii) $\frac{\pi s}{\Gamma(1+s)\Gamma(1-s)} = \sin(\pi s)$. □

For proofs of these facts, see for example Königsberger (2001, chapter 17).

We are now ready to extend (2.2.1) to a formula that is, as Riemann states, “valid for all s ”, and proceed as follows: First, substitute nx for x in (2.2.4), giving

$$\int_0^{\infty} e^{-nx} x^{s-1} \, dx = \frac{1}{n^s} \Gamma(s)$$

for $s > 0$ and $n \in \mathbb{N}$. Now, sum over all n using the formula for the geometric series, obtaining

$$(2.2.6) \quad \int_0^{\infty} \frac{x^{s-1}}{e^x - 1} \, dx = \Gamma(s) \sum_{n=1}^{\infty} \frac{1}{n^s}.$$

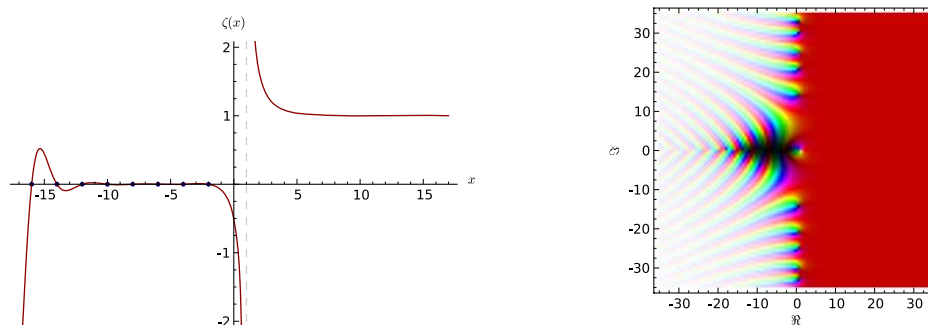


Figure 2.2.3: Plots of the zeta function. The left-hand picture shows the values of the zeta function on the real axis (note the the trivial zeros at the negative even integers), the right-hand one a complex plot with complex coloring defined in Figure 2.2.1.

Here, one needs to check the convergence of the integral on the left and the validity of the exchange of integration and summation. Consider now the contour integral

$$\int_P \frac{(-x)^{s-1}}{e^x - 1},$$

where the path P starts at $+\infty$, travels down the positive real axis, circles the origin in counterclockwise direction, and travels back to the positive real axis to $+\infty$. One can show that this integral equals $(e^{i\pi s} - e^{-i\pi s}) \int_0^\infty \frac{x^{s-1}}{e^x - 1} dx$ (see for example Edwards 1974, section 1.4). Combining with (2.2.6) yields

$$\int_P \frac{(-x)^{s-1}}{e^x - 1} = 2i \sin(\pi s) \Gamma(s) \sum_{n=1}^{\infty} \frac{1}{n^s},$$

which reads (after applying Lemma 2.2.5) as

$$(2.2.7) \quad \zeta(s) = \frac{\Gamma(1-s)}{2\pi i} \int_P \frac{(-x)^{s-1}}{e^x - 1},$$

now valid for all $s \neq 1$ (see Edwards 1974). This function is known as the famous *Riemann zeta function*. Some plots of this function can be found in Figure 2.2.3 and Figure 2.2.5.

It is relatively easy to give expressions for the value of $\zeta(s)$ for even integers $s = 2n$ and non-positive integers $s = -n$ (for $n \in \mathbb{N}_{\geq 0}$) in terms of so called Bernoulli numbers, named after Jacob Bernoulli (1654–1705), who described them first in his book “*Ars Conjectandi*”, posthumously published

n	0	1	2	4	6	8	10	12	14
B_n	1	$-\frac{1}{2}$	$\frac{1}{6}$	$-\frac{1}{30}$	$\frac{1}{42}$	$-\frac{1}{30}$	$\frac{5}{66}$	$-\frac{691}{2730}$	$\frac{7}{6}$

Table 2.2.1: The first non-zero Bernoulli numbers.

in 1713. They are defined as follows: We start from the function $\frac{x}{e^x-1}$ and perform power-series expansion around $x = 0$ (this is valid, since the function is analytic near 0), obtaining an expression of the form

$$\frac{x}{e^x - 1} = \sum_{n=0}^{\infty} B_n \frac{x^n}{n!},$$

valid for $|x| < 2\pi$. The coefficients B_n are called Bernoulli numbers. The odd Bernoulli numbers are always zero (except $B_1 = -\frac{1}{2}$), since for all x we have

$$\frac{x}{e^x - 1} + \frac{x}{2} = \frac{-x}{e^{-x} - 1} + \frac{-x}{2}.$$

The first few non-zero values are listed in Table 2.2.1. The Bernoulli numbers can be used to give explicit formulas for the values of the zeta function at negative and at even integers. We have

$$(2.2.8) \quad \zeta(-n) = (-1)^n \frac{B_{n+1}}{n+1}$$

and

$$(2.2.9) \quad \zeta(2n) = (-1)^{n+1} \frac{(2\pi)^{2n} B_{2n}}{2 \cdot (2n)!}.$$

For details on how to obtain the first expressions from (2.2.7), see Edwards (1974, section 1.5). The second expression is due to Euler (1755). Note that there is no simple closed form known for positive odd integer arguments. From (2.2.8) follows directly from the properties of the Bernoulli numbers that the zeta function vanishes at the even negative integers. These are called the *trivial zeros* of the zeta function.

The values of the Riemann zeta function at integer arguments are called *zeta constants*. These constants occur frequently in many different areas, such as probability theory or physics. We list a few zeta constants in Table 2.2.2.

Riemann also deduced in 1859 the functional equation

$$\zeta(s) = 2^s \pi^{s-1} \sin\left(s \cdot \frac{\pi}{2}\right) \Gamma(1-s) \zeta(1-s)$$

n	-4	-3	-2	-1	0	1	2	3	4
$\zeta(n)$	0	$\frac{1}{120}$	0	$-\frac{1}{12}$	$-\frac{1}{2}$	$\frac{1}{6} \pi^2$	$\frac{1}{90} \pi^4$	$\frac{1}{945} \pi^6$	$\frac{1}{9450} \pi^8$

Table 2.2.2: The first zeta constants.

for the zeta function, or by defining

$$(2.2.10) \quad \xi(s) = \frac{1}{2} s(s-1) \pi^{-s/2} \Gamma(s/2) \zeta(s)$$

the much simpler functional equation

$$(2.2.11) \quad \xi(s) = \xi(1-s).$$

For a proof of these facts, see Edwards (1974, section 1.6–1.8). Since $\xi(s)/\zeta(s)$ has only a single zero at $s = 1$, it follows that the remaining zeros of ξ coincide with the zeros of zeta. By employing the Euler Product Formula 2.2.2, the zeta function does not have any zero ρ with $\Re(\rho) > 1$ (since otherwise a convergent infinite product of non-zero factors would be zero). Due to the functional equation just described, (2.2.11), this immediately implies that there are also no zeros ρ with $\Re(\rho) < 0$. Thus, we can immediately conclude that *all* non-trivial zeros lie in the *critical strip* $0 \leq \Re(\rho) \leq 1$.

It turns out that the exact distribution of these zeros in the critical strip is closely related to the prime counting function $\pi(x)$: For that, let $\mu(n)$ be the so called Möbius function, systematically investigated by August Ferdinand Möbius (1790–1868) in 1832. It is defined by

$$(2.2.12) \quad \mu(n) = \begin{cases} 1, & \text{if } n \text{ is squarefree with an even number of prime factors,} \\ 0, & \text{if } n \text{ is not squarefree,} \\ -1, & \text{if } n \text{ is squarefree with an odd number of prime factors.} \end{cases}$$

Riemann showed that

$$(2.2.13) \quad \pi(x) = \sum_{n=1}^{\infty} \frac{\mu(n)}{n} J(x^{1/n})$$

with

$$J(x) = \text{Li}(x) - \sum_{\Im(\rho) > 0} (\text{Li}(x^\rho) + \text{Li}(x^{1-\rho})) - \ln 2 + \int_x^\infty \frac{1}{t(t^2-1) \ln t} dt$$

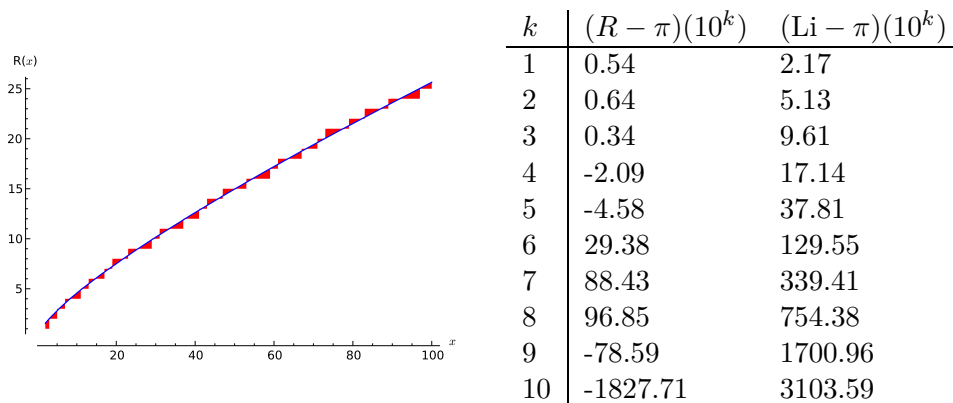


Figure 2.2.4: The left-hand picture shows the values of the Riemann function $R(x)$ (blue) in comparison to the prime counting function (red), the right-hand a table of errors at powers of ten.

and the sum in the second term runs over the nontrivial roots of zeta while summing in order of increasing imaginary part $\Im(\rho)$.

By plugging the definition of $J(x)$ into (2.2.13) and taking just the terms into account that grow as x does, we arrive at Riemann's famous prime count approximation

$$\pi(x) \approx R(x) := \sum_{n \geq 1} \frac{\mu(n)}{n} \text{Li}(x^{1/n}).$$

Note that the first term of Riemann's approximation equals Gauß' approximation $\pi(x) \approx \text{Li}(x)$, but the resulting estimate is (empirically) much better, see Figure 2.2.4. For details on how one deduces this estimate, see (Edwards 1974, section 1.11–1.17).

2.2.2. The Riemann hypothesis. Several properties of the Riemann zeta function (2.2.7) are still unproven. The following conjecture, already posed by Riemann in 1859, became one of the most important questions in number theory:

RIEMANN HYPOTHESIS 2.2.14. *For all zeros ρ of the Riemann zeta function (2.2.7) with $0 < \Re(\rho) < 1$ we have $\Re(\rho) = \frac{1}{2}$.*

In other words the hypothesis says that all zeros of the Riemann zeta function in the critical strip already lie on the *critical line*. The hypothesis has been numerically verified for the first 10^{13} zeros (see Saouter *et al.* 2011). Figure 2.2.5 shows two plots of zeta on the critical line.

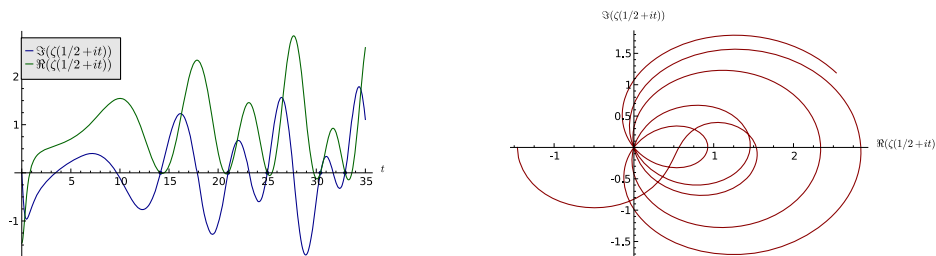


Figure 2.2.5: The left-hand picture shows the imaginary and real part, the right-hand a parametric plot of zeta along the critical line.

There are many conjectures in number theory that are equivalent to the Riemann Hypothesis 2.2.14. One of them is based on properties of a function Franz Mertens (1840–1927) introduced in 1897, namely the function

$$M(x) = \sum_{n \leq x} \mu(n),$$

where $\mu(n)$ is the Möbius function (2.2.12). By employing the Euler Product Formula 2.2.2 for $\frac{1}{\zeta(s)}$ on the one hand and the so called *Mellin-transform*

$$(\mathcal{M}f)(s) = \int_0^\infty x^{s-1} f(x) \, dx$$

of $\frac{1}{\zeta(s)}$, named after Robert Hjalmar Mellin (1854–1933), on the other hand, one obtains the following relation between the Mertens function $M(x)$ and the Riemann zeta function $\zeta(s)$:

$$\frac{1}{\zeta(s)} = \sum_{n=1}^{\infty} \frac{\mu(n)}{n^s} = s \int_1^\infty \frac{M(x)}{x^{s+1}} \, dx$$

valid (at least) for $\Re(s) > 1$. We have

THEOREM 2.2.15 (Littlewood 1912). *The Prime Number Theorem 2.1.8 is equivalent to*

$$M(x) \in o(x),$$

while the Riemann Hypothesis 2.2.14 is equivalent to

$$M(x) \in \mathcal{O}\left(x^{\frac{1}{2}+\varepsilon}\right)$$

for any fixed $\varepsilon > 0$.

□

Indeed, Niels Fabian Helge von Koch (1870–1924) showed in 1901 that for any fixed $\varepsilon > 0$ one has

$$(2.2.16) \quad \pi(x) \in \text{Li}(x) + \mathcal{O}\left(x^{\frac{1}{2}+\varepsilon}\right)$$

if and only if the Riemann Hypothesis 2.2.14 holds. The assertion (2.2.16) was later slightly strengthened and made much more explicit by Lowell Schoenfeld (1920–2002) in 1976, yielding the famous explicit version Prime Number Theorem 2.1.8(iii) that states that we have for $x \geq 1451$ the inequality

$$|\pi(x) - \text{Li}(x)| \leq \frac{1}{8\pi} \sqrt{x} \ln x$$

if and only if the Riemann Hypothesis 2.2.14 holds. The beauty of such a statement lies in the fact that it is completely explicit, in contrast to many theorems in number theory, where the main term is explicitly given, but the error term depends on some (often unknown) constant. To get rid of those hidden constants, one has to go through the analytic proofs and handle quite complicated error terms (see also Chapter 6). The benefit of such an approach is, however, twofold: First, one can make statements like “sufficiently large” precise and tell exactly when such an inequality starts to hold. Second, such explicit inequalities allow computer-aided verification of unproven conjectures like the Riemann Hypothesis 2.2.14: If the inequality fails to hold for a certain value of $x \geq 1451$, then also the Riemann hypothesis must be false.

One might ask now if there are also explicit versions for the number of primes in arithmetic progressions, discussed in Section 2.1.3. Indeed, we are not aware of any *unconditional* explicit version of Theorem 2.1.11. What we actually *do* have is an explicit version that is true if the so called extended Riemann hypothesis holds. We will state the theorem first, and afterwards give a short discussion of the hypothesis:

THEOREM 2.2.17 (Oesterlé 1979). *Write $\pi_{a+m\mathbb{Z}}(x)$ for the number of primes up to a real bound $x \geq 2$ that are in the residue class a modulo $m \geq 2$ with $\gcd(a, m) = 1$. Then, if the Extended Riemann Hypothesis 2.2.20 is true, we have*

$$\left| \pi_{a+m\mathbb{Z}}(x) - \frac{1}{\varphi(m)} \text{Li}(x) \right| \leq \sqrt{x} (\ln x + 2 \ln m),$$

where $\varphi(m)$ is the Euler φ -function (see Definition 2.1.10). □

The extended Riemann hypothesis is a conjectured property of so called Dirichlet L -functions, which are the analogues of the zeta function for primes in arithmetic progressions. Their definition depends on

DEFINITION 2.2.18 (Dirichlet character). *Let M be a positive integer and χ be a function from the integers to the complex numbers. We call χ a Dirichlet character modulo M if it is multiplicative, periodic modulo M and $\chi(n) \neq 0$ if and only if n is coprime to M .*

One example of a Dirichlet character for an odd positive integer M is the Jacobi-symbol $\left(\frac{\cdot}{M}\right)$ (see Section 3.1.4). It turns out that if χ_1 is a Dirichlet character modulo M_1 and χ_2 is a Dirichlet character modulo M_2 , then $\chi_1\chi_2$ is a Dirichlet character modulo $\text{lcm}(M_1, M_2)$, where we define $\chi_1\chi_2(n) := \chi_1(n)\chi_2(n)$. This in turn implies that Dirichlet characters modulo M are closed under multiplication and, in fact, form a multiplicative group: The identity is the character χ_0 for which $\chi_0(n) = 1$ if and only if n and M are coprime and $\chi_0(n) = 0$ otherwise. The multiplicative inverse of a character χ is its complex conjugate $\bar{\chi}$, defined as $\bar{\chi}(n) := \overline{\chi(n)}$. We are now ready for

DEFINITION 2.2.19 (Dirichlet L -function). *Let χ be a Dirichlet character modulo M . Then the function*

$$L(s, \chi) = \sum_{n=1}^{\infty} \frac{\chi(n)}{n^s}$$

is called a Dirichlet L -function.

The sum converges in the region $\Re(s) > 1$ and if χ is non-principal then the domain of convergence is $\Re(s) > 0$. Analogous to the Euler Product Formula 2.2.2, we have

$$L(s, \chi) = \prod_{p \text{ prime}} \frac{1}{1 - \chi(p)p^{-s}}.$$

Now, if $\chi = \chi_0$ is principal modulo M then $L(s, \chi) = \zeta(s) \cdot \prod_{p|M} (1 - p^{-s})$, which directly shows the connection to the zeta function. Clearly, if χ is the unique character modulo 1, the L -series is *exactly* the Riemann zeta function. We arrive at the

EXTENDED RIEMANN HYPOTHESIS 2.2.20. *Let χ be any Dirichlet character modulo M . Then for all zeros ρ of $L(s, \chi)$ with $\Re(\rho) > 0$ we have $\Re(\rho) = \frac{1}{2}$.*

The conjecture is also of central importance in *algorithmic* number theory. One beautiful example is the strong primality test, a test for compositeness

which might (with small probability) give a wrong answer. It was shown in Miller (1976) that if the Extended Riemann Hypothesis 2.2.20 is true, then the test will always answer correctly, implying that the set of primes can be decided in deterministic polynomial time (see Section 3.3.3). It is interesting to note that it took almost 30 years to remove the dependence on the Extended Riemann Hypothesis 2.2.20. For more information of this fact, see Section 3.3.4.

2.3. Counting other classes of integers

Besides counting primes with various properties in the spirit of Dirichlet, it was natural to look for (asymptotic) results on other types of integers. Examples that we will present in this section are integers that are a product of exactly two (not necessarily distinct) primes, integers with very small prime factors only and integers that have large prime factors only. Such kind of results are of central importance in the study of the complexity of various flavors of factorization algorithms (see also Section 3.4 and Section 3.5).

2.3.1. Landau: Counting semi-primes. Consider the problem of counting integers that are a product of exactly two distinct primes. The problem seems to have been first solved by Edmund Georg Hermann Landau (1877–1938) in 1909. To be more precise, consider the function

$$\pi_2(x) = \# \{n = pq \leq x \mid p \neq q \text{ prime}\}.$$

By definition, $\pi_2(x)$ equals half of the number of solutions for $pq \leq x$. Thus

$$(2.3.1) \quad \pi_2(x) = \sum_{p \leq x} \pi\left(\frac{x}{p}\right) - \pi(\sqrt{x}),$$

since the first summand counts the number of solutions for $pq \leq x$, where p, q are not necessarily distinct primes and $\pi(\sqrt{x})$ counts exactly the number of prime-squares up to x . The main tool in tackling the sum in (2.3.1) is Lemma 6.2.1, to be explained later, from which we just need a very special case here, namely

COROLLARY 2.3.2 (Special prime sum approximation). *We have*

$$\sum_{p \leq x} \pi(x/p) = \sum_{p \leq \frac{x}{2}} \pi(x/p) \approx \int_2^{\frac{x}{2}} \frac{x}{p \ln p \ln x/p} dp. \quad \square$$

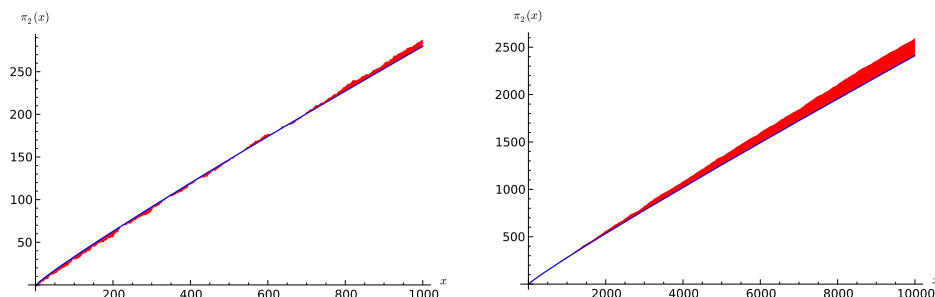


Figure 2.3.1: Landau's approximation (blue) in comparison to the exact count (red).

It remains to compute the integral on the right-hand side:

$$\begin{aligned}
 \int_2^{\frac{x}{2}} \frac{x}{p \ln p \ln x/p} dp &= \int_{\ln 2}^{\ln x - \ln 2} \frac{x}{\varrho(\ln x - \varrho)} d\varrho \\
 &= \frac{x}{\ln x} \int_{\ln 2}^{\ln x - \ln 2} \frac{1}{\varrho} - \frac{1}{\ln x - \varrho} d\varrho \\
 &= \frac{x}{\ln x} (\ln(\ln x - \ln 2) - \ln \ln 2 - \ln \ln 2 + \ln(\ln x - \ln 2)) \\
 &\approx \frac{2x \ln \ln x}{\ln x}.
 \end{aligned}$$

Since $\pi(\sqrt{x}) \in \mathcal{O}\left(\frac{\sqrt{x}}{\ln x}\right)$ in (2.3.1), it follows

THEOREM 2.3.3 (Landau 1909). *For x tending to infinity, we have*

$$\pi_2(x) \approx \frac{2x \ln \ln x}{\ln x}.$$

In Figure 2.3.1 two plots of the Landau approximation can be found.

2.3.2. Dickman and the count of smooth numbers. We are now going to present several classical results on integers that have only very small prime factors. We follow in our exposition Crandall & Pomerance (2005), Granville (2008), and Hildebrand & Tenenbaum (1993).

DEFINITION 2.3.4 (smooth integer). *Let n be a positive integer. Then n is called y -smooth if every prime factor of n does not exceed y .*

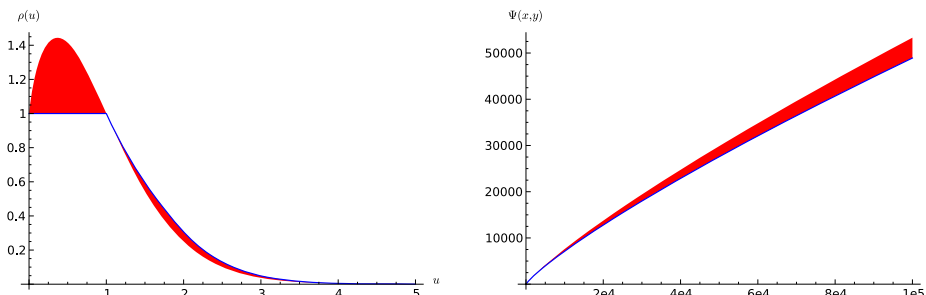


Figure 2.3.2: The left-hand picture shows the Dickman rho function (blue) in comparison to the Ramaswami-approximation (red), the right-hand one a plot of the function $x \cdot \varrho(u)$ with $u = \ln(x)/\ln(y)$ and $y = 10^3$ (blue) in comparison to the precise count $\Psi(x, y)$.

Smooth integers occur in many parts of algorithmic number theory and cryptography as the success of many factoring algorithms depends on questions concerning smooth integers (see Section 3.4 or Section 3.5).

To be more precise, we will consider the function

$$\Psi(x, y) := \#\{n \leq x \mid n \text{ is } y\text{-smooth}\}.$$

A remarkable result on the order of magnitude of $\Psi(x, y)$ was proven by Karl Dickman (ca. 1862–1940). He proved

THEOREM 2.3.5 (Dickman 1930). *Let $u > 0$ be a constant. Then there is a real number $\varrho(u) > 0$ with*

$$\Psi(x, x^{\frac{1}{u}}) \approx \varrho(u)x.$$

More precisely, this holds for

$$\varrho(u) = \begin{cases} 1, & \text{if } 0 < u \leq 1, \\ \frac{1}{u} \cdot \int_{u-1}^u \varrho(t) dt, & \text{if } u > 1. \end{cases} \quad \square$$

It is (moderately) easy to compute $\varrho(u)$ numerically (see Figure 2.3.2) using, for example, the trapezoid method. For $1 < u \leq 2$, we have the explicit expression $\varrho(u) = 1 - \ln u$, but there is no closed form known for $u > 2$ (using elementary functions only). Dickman himself did not give a rigorous (quantitative) proof of Theorem 2.3.5. The first quantitative results were given by Ramaswami (1949), who showed that

$$(2.3.6) \quad \ln \varrho(u) \in -(1 + o(1))u \ln u.$$

Dickman's Theorem 2.3.5 can be employed as an estimate for $\Psi(x, y)$ as long as $u = \ln x / \ln y$ is fixed (or at least bounded). However, in many applications that will pop up later, it is necessary to have estimates for wider ranges of u . The first step in this direction was done by de Bruijn (1951). There, it was shown that for any $\varepsilon > 0$ the estimate

$$\Psi(x, y) \in \left(1 + \mathcal{O}\left(\frac{\ln(u+1)}{\ln y}\right)\right) \varrho(u)x$$

holds uniformly in the interval $1 \leq u \leq (\ln y)^{\frac{3}{5}-\varepsilon}$. This was substantially improved by Hildebrand (1986), who showed that the statement even holds uniformly in the range

$$1 \leq u \leq \exp\left((\ln y)^{\frac{3}{5}-\varepsilon}\right).$$

Due to our inability to find closed forms for $\varrho(u)$ there were also investigations how far estimates in the spirit of (2.3.6) hold. Canfield, Erdős & Pomerance proved in 1983 an estimate which is extremely useful for algorithmic number theory. We have (as x tends to infinity) uniformly for $u < (1 - \varepsilon) \frac{\ln x}{\ln \ln x}$ that

$$\Psi(x, x^{\frac{1}{u}}) \in u^{-u+o(u)}x.$$

It is interesting to note that finding an estimate $\Psi(x, x^{\frac{1}{u}}) \approx \varrho(u)x$ in such a wide range, would readily imply the Riemann Hypothesis 2.2.14, see Hildebrand (1985).

2.3.3. Buhštab: Results on rough integers. In contrast to smooth integers, which we described in the previous section, the dual problem did, by far, not attract that much attention.

DEFINITION 2.3.7 (rough integer). *Let n be a positive integer. Then n is called y -rough if every prime factor of n exceeds y .*

Note that there are integers that are neither y -smooth nor y -rough.

EXAMPLE 2.3.8. The integer $6 = 2 \cdot 3$ is neither 2-smooth nor 2-rough. \diamond

The corresponding counting function for those integers is

$$\Phi(x, y) = \#\{n \leq x \mid n \text{ is } y\text{-rough}\}.$$

Александр Адольфович Бухштаб (1905–1990)⁸ showed in 1937

⁸Aleksandr Adolfovich Buhštab

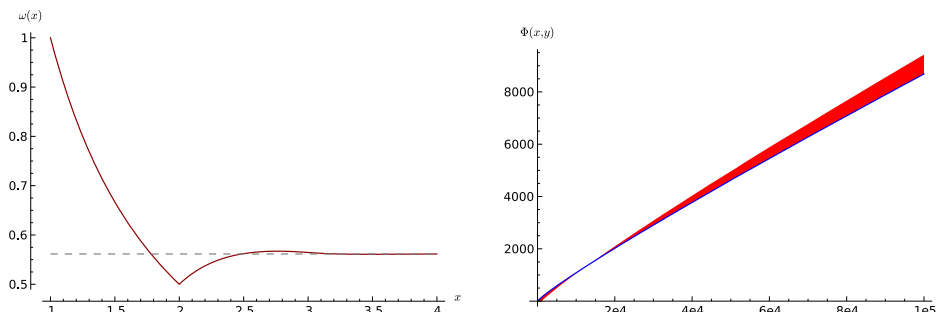


Figure 2.3.3: The left-hand picture shows the Бухштаб omega function together with its asymptote $\exp(-\gamma)$, the right-hand one plot of the function $\omega(u) \frac{x}{\ln y}$ with $u = \ln(x)/\ln(y)$ and $y = 10^3$ (blue) in comparison to the precise count $\Phi(x, y)$.

THEOREM 2.3.9 (Бухштаб 1937). *Let $u > 1$ be a constant. Then there is a real number $\omega(u) > 0$ with*

$$\Psi(x, x^{\frac{1}{u}}) \approx \omega(u) u \frac{x}{\ln x}.$$

More precisely,

$$\omega(u) = \begin{cases} \frac{1}{u}, & \text{if } 1 < u \leq 2, \\ \frac{1}{u} \cdot \left(1 + \int_1^{u-1} \omega(t) dt\right), & \text{if } u > 2. \end{cases} \quad \square$$

In Figure 2.3.3 the Бухштаб omega function is depicted. Rough numbers occur in many parts of algorithmic number theory and cryptography, interestingly often in the same context as smooth numbers do. Questions on how numbers that are simultaneously C -smooth and B -rough (you might want to call them *grained*) behave are to be discussed in Chapter 6.

Chapter 3

Algorithmic number theory

3.1. Basic algorithms

After having explored various results in analytic number theory, we will now delve into the *computational* aspects of number theory. The methods presented in the sequel have (unsurprisingly) their roots in ancient times, but starting with the emergence of computers, the field experienced a rapid development, leading for example to sub-exponential factorization algorithms (tackling the Factorization Problem 2.1.3) and a deterministic polynomial-time algorithms for primality testing (tackling Problem 2.1.5). As alluded in the introduction to Section 2.3, one needs a thorough understanding of several *analytic* aspects of number theory, in order to really understand the issues of *algorithmic* number theory. These include, in particular, the complexity of primality tests and factorization algorithms.

3.1.1. Euclid and the greatest common divisor. Computing the greatest common divisor of two numbers is one of the oldest problem in computational number theory. We have the following basic theorem, dating back to Euclid:

THEOREM 3.1.1 (Euclid Elements, book VII, proposition 2). *Let a, b be two integers, where b is non-zero. Then have $\gcd(a, b) = \gcd(b, a) = \gcd(b, a \bmod b)$. For $b = 0$ we have $\gcd(a, b) = a$. \square*

This is the basis of one of the oldest computational methods, the *Euclidean algorithm*, which efficiently computes the greatest common divisor of two integers. The idea behind it is to successively apply the above theorem until the second parameter equals zero:



Figure 3.1.1: On the left one finds a plot of the greatest common divisor for $1 \leq a, b \leq 50$, where the size of the result is indicated by the darkness of the pixel. The right-hand picture shows the runtime of the Euclidean Algorithm 3.1.2 on input (a, b) , for $1 \leq a, b \leq 100$. The large black area (maximal number of steps) lies around the line $b = \varphi a$, where $\varphi = \frac{1+\sqrt{5}}{2}$ is the golden ratio.

EUCLIDEAN ALGORITHM 3.1.2.

Input: Two positive integers a, b .

Output: $\gcd(a, b)$.

1. While $b > 0$ do
2. Set $(a, b) = \gcd(b, a \bmod b)$.
3. Return a .

Even though the algorithm is as simple as it can possibly be, its runtime analysis is a little bit delicate. We will prove an upper bound on the number of steps of the Euclidean Algorithm 3.1.2. Interestingly, the analysis will involve a linearly recurrent sequence studied by Leonardo Fibonacci (ca. 1170–1250) in his “Liber abbaci” (1202), which are nowadays known as the *Fibonacci numbers* F_n . These are defined by setting $F_0 = 0$, $F_1 = 1$ and

$$F_n = F_{n-1} + F_{n-2}.$$

The Fibonacci numbers are closely related to the golden ratio $\varphi = \frac{1+\sqrt{5}}{2} = 1.6180\mathbf{4}$ by a formula already discovered by Abraham de Moivre (1667–1754) in 1730.⁹ It is nowadays named after Jacques Philippe Marie Binet (1786–1856) and known as

⁹To indicate how a real number was rounded we append a special symbol. Examples: $\pi = 3.14\mathbf{4} = 3.142\mathbf{\Psi} = 3.1416\mathbf{\Upsilon} = 3.14159\mathbf{4}$. The height of the platform shows the size of the left-out part and the direction of the antenna indicates whether actual value is larger or smaller than displayed. We write, say, $e = 2.72\mathbf{\Upsilon}$ as if the shorthand were exact.

BINET'S FORMULA 3.1.3 (Binet 1843). Let $\varphi = \frac{1+\sqrt{5}}{2}$ and $\bar{\varphi} = \frac{1-\sqrt{5}}{2}$. Then we have for all $n \geq 0$ that

$$F_n = \frac{\varphi^n - \bar{\varphi}^n}{\sqrt{5}}.$$

PROOF. Consider the polynomial $f = x^2 - x - 1 \in \mathbb{R}[x]$. Any root α of f fulfills $\alpha^2 = \alpha + 1$ or equivalently for any $n \geq 1$

$$\alpha^n = F_n \cdot \alpha + F_{n-1}.$$

Now, since φ and $\bar{\varphi}$ are both roots of f , we have $\varphi^n = F_n \cdot \varphi + F_{n-1}$ and $\bar{\varphi}^n = F_n \cdot \bar{\varphi} + F_{n-1}$. Subtracting gives the claim for $n > 0$, and direct inspection shows the claim for $n = 0$. \square

There are many more connections of Fibonacci numbers and the golden ratio: An example is that the quotient of two successive Fibonacci numbers converges to the golden ratio. This can be seen by observing that by the definition of the Fibonacci numbers we have

$$\frac{F_{n+1}}{F_n} = 1 + \left(\frac{F_n}{F_{n-1}} \right)^{-1}.$$

If now the quotients converge to a positive value φ , then we would have $\varphi = 1 + \frac{1}{\varphi}$, which is exactly the defining equation for the golden ratio. For an illustration of this fact, see Figure 3.1.2. The connection of Fibonacci numbers and the runtime of the Euclidean Algorithm 3.1.2 gets clear by

LEMMA 3.1.4. Let a, b be positive integers with $a > b$. If the Euclidean Algorithm 3.1.2 on input a, b performs exactly n recurrent calls, then $a \geq F_{n+2}$ and $b \geq F_{n+1}$. \square

One can prove the lemma by induction on n . Indeed, we can also show that the Euclidean Algorithm 3.1.2 runs longest when the input are two successive Fibonacci numbers. For that let us say that a pair (a, b) is *lexicographically less* than (a', b') if $a < a'$ or $a = a'$ and $b < b'$. Using this, we have

THEOREM 3.1.5 (Lamé 1844). Let a, b be positive integers with $a > b$. If the Euclidean Algorithm 3.1.2 on input a, b performs exactly n recurrent calls, and (a, b) is lexicographically the smallest such input, then $(a, b) = (F_{n+2}, F_{n+1})$. \square

The proof of the theorem follows directly from Lemma 3.1.4 and elementary properties of the Fibonacci numbers. We arrive at the worst-case runtime estimate of the Euclidean Algorithm 3.1.2, which is

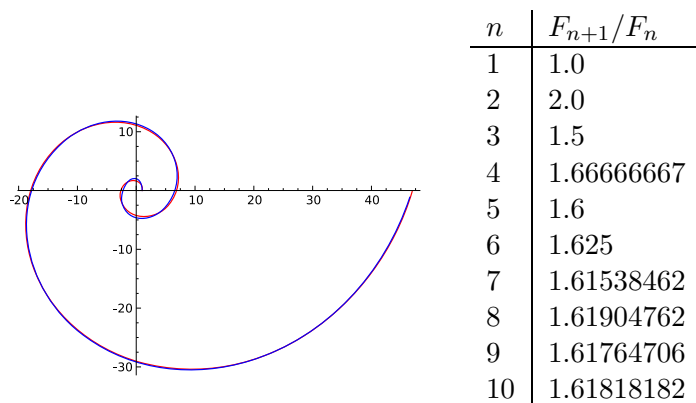


Figure 3.1.2: On the left one finds a red spiral that grows proportional to the quotient of two successive Fibonacci numbers, compared to a true golden spiral that grows always proportional to the golden ratio $\varphi = \frac{1+\sqrt{5}}{2}$. The right-hand table shows the convergence of quotients of successive Fibonacci numbers to the golden ratio.

COROLLARY 3.1.6. *The Euclidean Algorithm 3.1.2 on integers a, b with $b \leq N$ runs in at most $\log_\varphi(3 - \varphi)N \in \mathcal{O}(\log N)$ steps.*

PROOF. After one iteration of the Euclidean Algorithm 3.1.2, we have $b > a \bmod b$, thus Theorem 3.1.5 applies, and the maximum number of steps n occurs for $b = F_{n+1}$ and $a \bmod b = F_n$. Since $b = F_{n+1} < N$, it follows by Binet's Formula 3.1.3 and the fact that $\left| \frac{\varphi^n}{\sqrt{5}} \right| < 0.5$ the inequality $\frac{\varphi^{n+1}}{\sqrt{5}} < N$. Thus $n < \log_\varphi \frac{\sqrt{5}}{\varphi} N = \log_\varphi(3 - \varphi)N$. \square

This shows that the Euclidean Algorithm 3.1.2 always needs a logarithmic number of steps in the size of the second argument. Indeed, one can show that when a, b are both uniformly chosen from $[1, N]$, then heuristically the algorithm runs *on average* with

$$\frac{12 \ln 2}{\pi^2} \ln N + 0.06$$

iterations. For details, see Knuth 1998, section 4.5.3.

The problem of computing the greatest common divisor is closely related to the problem of computing the inverse of an integer. This can be easily seen by a theorem proven for polynomials by Étienne Bézout (1730–1783) in 1766, earlier proven for coprime integers by Claude Gaspar Bachet de Méziriac (1581–1638) in 1612. It is nowadays known as

BÉZOUT'S IDENTITY 3.1.7. *Let a, b be integers, not both zero. Then there are integers s, t such that*

$$as + bt = \gcd(a, b).$$

PROOF. Let g be the smallest positive value of $as + bt$ (where s, t range over all integers). We claim that $g = \gcd(a, b)$. Clearly, since $\gcd(a, b)$ divides both a and b , we have that $\gcd(a, b)$ divides g . We now show that also g divides $\gcd(a, b)$. Assume g does not divide a . Then there are integers q and r and $0 < r < g$ with $a = qg + r$. But then $r = a(1 - qs) + bqt$, contradicting the choice of g . Similarly, one shows that g divides b , implying that g divides $\gcd(a, b)$ and thus $g = \gcd(a, b)$. \square

Consider now the case $\gcd(a, b) = 1$. Then it is easy to give the inverse of a modulo b by simply taking Bézout's identity and reducing both sides of the equation modulo b , giving

$$as + bt = as = \gcd(a, b) = 1 \text{ (in } \mathbb{Z}_b\text{)}.$$

Thus s is the inverse of a modulo b . Indeed, it is possible to adapt the Euclidean Algorithm 3.1.2 to also find on the way the Bézout coefficients s and t , giving the

EXTENDED EUCLIDEAN ALGORITHM 3.1.8.

Input: Two positive integers a, b .

Output: Integers s, t, g with $g = \gcd(a, b)$ and $as + bt = g$.

1. Set $(s, t, g, u, v, h) = (1, 0, a, 0, 1, b)$.
2. While $w > 0$ do 3–4
3. Set $q = g \operatorname{div} h$.
4. Set $(s, t, g, u, v, h) = (u, v, h, s - qu, t - qv, g - qh)$.
5. Return (s, t, g) .

The runtime analysis of the algorithm is very similar to the analysis of the traditional Euclidean Algorithm 3.1.2.

3.1.2. Euler and the exponentiation. Another fundamental operation in algorithmic number theory is to take an element g in a finite (multiplicative) group G and compute g^e for an integer e . The central result is due to Joseph Louis de Lagrange (1736–1813) who proved the following

THEOREM 3.1.9 (Lagrange 1770/71). *Let G be a finite multiplicative group and H be a subgroup of G . Then the number of elements in H divides the number of elements in G .*

PROOF. This directly follows from the fact that the relation on G , by defining $g_1 \sim g_2$ if and only if there is $h \in H$, such that $g_1 = g_2 \cdot h$, is an equivalence relation. \square

From this theorem we can deduce that the *order* of an element g in G divides the group order of G . In other words: it shows that for any integer e we have $g^e = g^{e \bmod \#G}$. In the special case that G is the group of unities of the ring \mathbb{Z}_m for $m \in \mathbb{Z}_{\geq 2}$, we obtain a result which Euler proved, namely

THEOREM 3.1.10 (Euler 1760/61). *For $a \in \mathbb{Z}_m$, we have*

$$a^{\varphi(m)} = 1 \text{ in } \mathbb{Z}_m,$$

where $\varphi(m)$ is the function given in Definition 2.1.10. \square

Note that this theorem directly gives an alternative way of computing the inverse a^{-1} of an element a in \mathbb{Z}_m , by computing

$$a^{-1} = a^{\varphi(m)-1} \text{ in } \mathbb{Z}_m.$$

In order to compute it, it is of course necessary to find a better way for computing a^e than multiplying e copies of a . The following recursive algorithm works for arbitrary elements of semigroups, that is a set with a binary, associative operation. Examples for semigroups are the integers modulo m , elements of a finite field and also points on an elliptic curve. For an example of the latter see Chapter 4. We obtain the

FAST EXPONENTIATION ALGORITHM 3.1.11.

Input: An element g of a semigroup G , a non-negative integer e .

Output: $g^e \in G$.

1. If $e = 0$ then
2. Return 1.
3. If $e \bmod 2 = 0$ then
4. Let h be the output of the algorithm on input g and $e \operatorname{div} 2$.
5. Return h^2 .
6. Else
7. Let h be the output of the algorithm on input g and $(e - 1) \operatorname{div} 2$.
8. Return $g \cdot h^2$.

The runtime of the algorithm (at least the number of recursive steps) is much simpler to analyze than the Euclidean Algorithm 3.1.2: Since in every step the number of bits of e decreases by exactly one bit and the algorithm terminates when e equals zero, the algorithm terminates after exactly $\lfloor \log_2(e) \rfloor$ recursive steps. In each step we have to compute one square in G and in the case $e \bmod 2 = 1$ additionally one multiplication in G . This leads to a worst case runtime of $2(\lfloor \log_2(e) \rfloor + 1) \in \mathcal{O}(\log e)$ multiplications in G , noting that a square h^2 can be computed in any semigroup G by simply multiplying h with itself. The above estimate gets very crude while working in a domain where squaring can be done very efficiently comparing to multiplication (like a field extension of \mathbb{F}_2 represented by a normal basis).

3.1.3. Sun Tzŭ: The chinese remainder theorem. To prove properties of the integers modulo m , it is often very helpful to employ the

CHINESE REMAINDER THEOREM 3.1.12. *Let m_1, \dots, m_ℓ be positive, pairwise relatively prime integers and let $m = m_1 \cdots m_\ell$. Then for all integers a_1, \dots, a_ℓ there is exactly one integer $0 \leq a < m$, such that*

$$\begin{aligned} a &= a_1, \text{ in } \mathbb{Z}_{m_1} \\ &\vdots \\ a &= a_\ell, \text{ in } \mathbb{Z}_{m_\ell}. \end{aligned}$$

PROOF. We provide a constructive proof for finding a . The quantity $M_i = \left(\frac{m}{m_i}\right)^{\varphi(m_i)}$ satisfies $M_i = 1$ in \mathbb{Z}_{m_i} and $M_k = 0$ in \mathbb{Z}_{m_k} for $k \neq i$

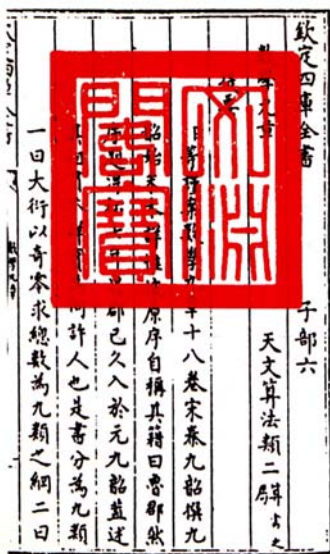


Figure 3.1.3: Front cover of Jiushao’s “Mathematical Treatise in Nine Sections”, published in 1247.

by Theorem 3.1.10. Thus

$$a = a_1M_1 + \cdots + a_\ell M_\ell$$

satisfies all conditions modulo m . \square

Following Knuth (1998) the roots of the theorem date back to ancient China, in the work of Sun Zi (ca. 400–460), published very roughly in the fifth century A.D.. It seems that the Chinese Remainder Theorem 3.1.12 was proven the first time seems to be in work of Qin Jiushao (1202–1261) in 1247.

3.1.4. Legendre and Jacobi: Quadratic (non)residues.

DEFINITION 3.1.13 (Quadratic (non)residue). *Let m be a positive integer and $a \in \mathbb{Z}_m^\times$. Then a is a quadratic residue modulo m , if there is an element $b \in \mathbb{Z}_m^\times$ such that $a = b^2$. If such b does not exist, we call a a quadratic nonresidue modulo m .*

A special symbol – fashionable for that time – was introduced by Legendre in 1798, An VI, that denotes some kind of characteristic function of quadratic residuosity modulo odd primes. It is given by

DEFINITION 3.1.14 (Legendre symbol). *Let p be an odd prime, and $a \in \mathbb{Z}$. Then we define the Legendre symbol $\left(\frac{a}{p}\right)$ by*

$$\left(\frac{a}{p}\right) = \begin{cases} 1, & \text{if } a \text{ is a quadratic residue modulo } p, \\ 0, & \text{if } p \text{ divides } a, \\ -1, & \text{if } a \text{ is a quadratic nonresidue modulo } p. \end{cases}$$

To see how we can compute the symbol easily, we can employ a special case of Euler's Theorem 3.1.10 for prime moduli. The theorem was first stated by Pierre de Fermat (1601/1607/1608–1665) in a letter to Bernard Frénicle de Bessy (ca. 1605–1675) in 1640 (as usual, without a proof):

Et cette proposition est généralement vraie en toutes progressions et en tous nombres premiers; de quoi je vous enverrois la démonstration, si je n'appréhendois d'être trop long.¹⁰

The first proof was given by Euler. The theorem is now known as

FERMAT'S LITTLE THEOREM 3.1.15 (Euler 1741). *Let p be a prime and let $a \in \mathbb{Z}$. Then $a^p = a$ in \mathbb{Z}_p .* □

From the theorem we obtain directly

EULER'S CRITERION 3.1.16 (Euler 1761). *Let p be an odd prime and $a \in \mathbb{Z}$. Then*

$$\left(\frac{a}{p}\right) = a^{\frac{p-1}{2}} \text{ modulo } p.$$

PROOF. If p divides a , the claim is true by definition. Now let a be a quadratic residue modulo p . Then there is $b \in \mathbb{Z}$ such that $b^2 = a$ modulo p . But then $a^{\frac{p-1}{2}} = b^{p-1} = 1$ by Fermat's Little Theorem 3.1.15. By the same theorem we will always have $a^{\frac{p-1}{2}} = \pm 1$, which shows the result also for quadratic nonresidues modulo p . □

A generalization to composite moduli was introduced by Carl Gustav Jacob Jacobi (1804–1851) in 1837:

¹⁰And this proposition is generally true for all progressions and for all prime numbers; the proof of which I would send to you, if I were not afraid to be too long.

DEFINITION 3.1.17 (Jacobi symbol). Let n be an odd integer with unique prime factorization $n = \prod_i p_i^{e_i}$. Then the Jacobi symbol $\left(\frac{a}{n}\right)$ is defined as

$$\left(\frac{a}{n}\right) = \prod_i \left(\frac{a}{p_i}\right)^{e_i}.$$

The computation of the Jacobi symbol is not as simple as the computation of the Legendre symbol, since a repeated application of Euler's Criterion 3.1.16 would require the factorization of n , which is in general difficult to obtain. To avoid this obstacle, one typically employs a theorem Gauß proved the first time in his "Disquisitiones Arithmeticae", articles 107-150. It is known as the

LAW OF QUADRATIC RECIPROCITY 3.1.18 (Gauß 1801). Let m, n be two coprime positive integers. Then

$$\left(\frac{m}{n}\right) \cdot \left(\frac{n}{m}\right) = (-1)^{\frac{(m-1)(n-1)}{4}}. \quad \square$$

Note that for the proper definition of the symbols used in the theorem one needs to consider the case of even moduli separately. The computation of the Jacobi symbol is now essentially a multi-application of this theorem:

ALGORITHM 3.1.19. Computing the Jacobi symbol.

Input: A positive integer m and a positive odd integer n .

Output: The Jacobi symbol $\left(\frac{m}{n}\right)$.

1. Set $m = m \bmod n$.
2. Set $t = 1$.
3. While $m \neq 0$ do 4-11
4. While $m \bmod 2 = 0$ do 5-7
5. Set $m = m/2$.
6. If $n \bmod 8 \in \{3, 5\}$ then
7. Set $t = -t$.
8. Set $(m, n) = (n, m)$.
9. If $m = n = 3$ modulo 4 then
10. Set $t = -t$.
11. Set $m = m \bmod n$.
12. If $m = 1$ then
13. Return t .
14. Return 0.

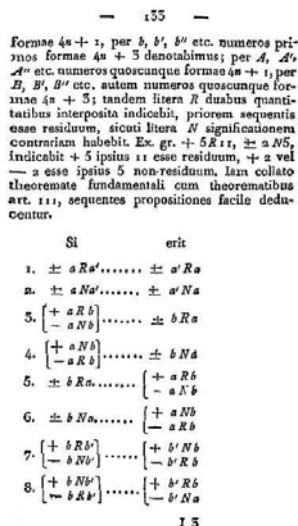


Figure 3.1.4: The different cases of quadratic reciprocity, as described by Gauss in his “Disquisitiones Arithmeticae”, in 1801.

One important property used for applications in algorithmic number theory and cryptography is that the Jacobi symbol can be efficiently computed (without knowing the prime factorization of the modulus n), although the known methods to compute it do not help in deciding quadratic (non)residuosity modulo n . This can be seen by observing that a quadratic nonresidue a modulo p_1 and p_2 is also a quadratic nonresidue modulo p_1p_2 , but for the corresponding Jacobi symbol we have $\left(\frac{a}{p_1p_2}\right) = \left(\frac{a}{p_1}\right)\left(\frac{a}{p_2}\right) = 1$.

3.2. Newton: Recognizing perfect powers

Some primality tests and factoring algorithms require that the input number n should not be a perfect power. Sir Isaac Newton (1643–1727 (greg.)) described in his famous book “The Method of Fluxions and Infinite Series with its Application to the Geometry of Curve-Lines”, published in 1671, a new method for finding a root of a polynomial equation. He essentially proposed to find a zero of a given polynomial $f(x) \in \mathbb{R}[x]$ by choosing an

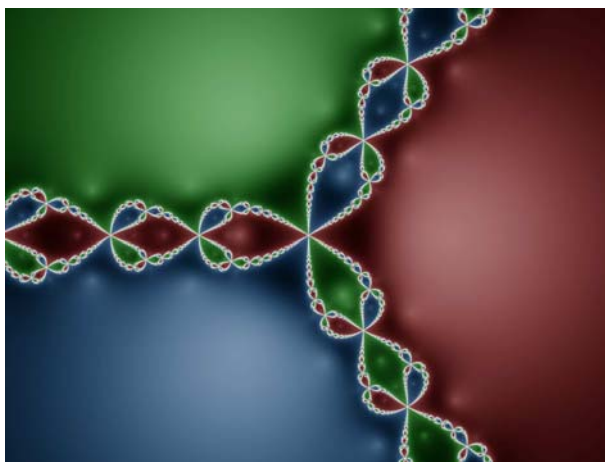


Figure 3.2.1: Illustration of the convergence of the Newton iteration for the polynomial $f(x) = x^3 - 1$ for different starting values $x_0 \in \mathbb{C}$. The colors red, green, and blue indicate a convergence to the first, second, and third root of the polynomial, respectively. The brightness visualizes the number of iteration necessary for convergence, i.e. in the white areas there is no convergence.

arbitrary starting value x_0 , and to perform the iteration

$$(3.2.1) \quad x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

Questions on the runtime, the convergence, and the numerical stability of this procedure go far beyond the scope of this thesis, especially when one allows the polynomial to attain *complex* values. A beautiful illustration of the dynamic of the convergence is given in Figure 3.2.1. Using the Newton iteration (3.2.1) it is now simple to extract the integer part of a k -th root of a given positive integer n , since the problem of extracting it is equivalent to finding a root of the polynomial $f(x) = x^k - n$. For these polynomials (3.2.1) simplifies to

$$x_{n+1} = \frac{1}{k} \left((k-1)x_n + \frac{n}{x_n^{k-1}} \right),$$

giving the following

ALGORITHM 3.2.2. Integer part of a k -th root.

Input: An integer $n \geq 0$ and a positive integer $k \geq 2$.

Output: The integer $m = \lfloor \sqrt[k]{n} \rfloor$.

1. If $n = 0$ then
2. Return 0.
3. Set $y = 2^{\lceil (\log_2 n + 1)/k \rceil}$.
4. Repeat 5–6
5. Set $x = y$.
6. Set $y = \lfloor \left((k-1)x + \left\lfloor \frac{n}{x^{k-1}} \right\rfloor \right) / k \rfloor$.
7. Until $y \geq x$
8. Return x .

As indicated above, we are not going to analyze correctness and runtime of this algorithm. Crandall & Pomerance (2005, section 9.2.2) say that it is possible to show that the algorithm uses $\mathcal{O}(\log \log n)$ iterations. We are now ready to state the

PERFECT POWER TEST 3.2.3.

Input: An integer $n \geq 0$.

Output: Either “perfect power” or “not a perfect power”.

1. For k from 2 to $\lfloor \log_2 n \rfloor$ do 2–5
2. Compute $m = \lfloor \sqrt[k]{n} \rfloor$ using Algorithm 3.2.2.
3. Compute $n' = m^k$ using the Fast Exponentiation Algorithm 3.1.11.
4. If $n' = n$ then
5. Return “perfect power”.
6. Return “not a perfect power”.

The correctness of the algorithm is easy to show, since the largest integer k for which $n = m^k$ can possibly hold is clearly $\lfloor \log_2 n \rfloor$. The runtime of the algorithm are $\mathcal{O}(\log n)$ calls of Algorithm 3.2.2. In every call of Algorithm 3.2.2 we need to compute $\mathcal{O}(\log \log n)$ k -th powers. Thus, we can estimate the overall runtime of the Perfect Power Test 3.2.3 as $\mathcal{O}(\log^2 \log n \cdot \log n)$ multiplications of integers not larger than n .

3.3. Primality testing

We are now going to explore various algorithmic methods for tackling Problem 2.1.5, the problem of deciding whether an integer is prime. All the following tests have in common that they employ an easily checkable condition

C that holds for all prime numbers. Then, as explained at the end of Section 2.1.1, given some integer n , one can simply check if the condition holds for it. If it does not, we can be sure that n is composite, otherwise n might be prime or might be composite and we call n a C probable prime. Any composite integer that fulfills the condition C is called a C -pseudoprime. The quality (that is success probability) of this method clearly depends on the proportion of C probable primes to genuine primes. Optimally there are no C -pseudoprimes and the test would run in *deterministic* polynomial time. Since all tests below follow the above approach, it would be better to call them *compositeness tests* instead of *primality tests*. We will, however, most of the time use the latter term in abuse of language.

3.3.1. Fermat's test. A very simple test that comes into mind relies on Fermat's Little Theorem 3.1.15. Recall that it says that for a prime p and any integer a we have

$$(3.3.1) \quad a^p = a \text{ modulo } p.$$

When a is coprime to p we can divide the above expression by a to obtain $a^{p-1} = 1$ in \mathbb{Z}_p , for all primes p and integers a coprime to p . Any integer n that satisfies $a^n = a$ modulo n is called a (*Fermat*) *probable prime* to the base a . If n is composite, we call n a (*Fermat*) *pseudoprime* to the base a . We exclude the base $a = 1$ and consider in the following just the case $a \geq 2$.

EXAMPLE 3.3.2. The number $15 = 3 \cdot 5$ is a Fermat pseudoprime to the base 4 since $4^{15} = 4$ in \mathbb{Z}_{15} . Similarly, 4 is a pseudoprime to the base 5. Table 3.3.1 lists the first pseudoprimes for several choices of a . \diamond

Indeed, it is possible to show that pseudoprimes are rare compared to genuine primes as stated by the following

THEOREM 3.3.3 (Erdős 1950, Li 1997). *For each fixed integer $a \geq 2$ the number of Fermat pseudoprimes to the base a up to a bound x is $o(\pi(x))$ for x tending to infinity.* \square

The famous algorithm reads as the

FERMAT TEST 3.3.4.

Input: An integer $n > 3$ and an integer $1 < a < n - 1$.

Output: Either “probable prime to the base a ” or “composite”.

k	$\pi(10^k)$	$\text{Li}(10^k)$
1	4	5.1204
2	25	29.081
3	168	176.5645
4	1229	1245.0921
5	9592	9628.7638
6	78498	78626.504
7	664579	664917.3599
8	5761455	5762208.3303
9	50847534	50849233.9118
10	455052511	455055613.541

Table 3.3.1: The first ten pseudoprimes to the bases $a = 2 \dots 7$.

1. Compute $b = a^{n-1}$ in \mathbb{Z}_n using the Fast Exponentiation Algorithm 3.1.11.
2. If $b = 1$ then
3. Return “probable prime to the base a ”.
4. Else
5. “composite”.

The asymptotic runtime of the algorithm is clearly the same as the runtime of the Fast Exponentiation Algorithm 3.1.11, namely $\mathcal{O}(\log n)$ arithmetic operations modulo n . But what about the error probability of the algorithm? We have seen in Theorem 3.3.3 that the number of Fermat pseudoprimes to the base a are rare when compared to the number of primes. If there were only finitely many such pseudoprimes (for any fixed a) our algorithm would work well! However, we have

THEOREM 3.3.5. *For every integer $a \geq 2$ there are infinitely many Fermat pseudoprimes to the base a .* \square

For a proof of the theorem see for example Crandall & Pomerance (2005, section 3.4.1).

So, what can we do? We could try to run the Fermat Test 3.3.4 for several choices of a and hope that there are few (if not no) numbers that are simultaneously pseudoprime to these different bases.

EXAMPLE 3.3.6. We have seen that the number $15 = 3 \cdot 5$ is a Fermat pseudoprime to the base 4. But it is not a pseudoprime to the base 5. Similarly 4 is a pseudoprime to the base 5 but not to the base 4. \diamond

Indeed, all the reasoning above works when we are not especially unlucky and get as an input a so called Carmichael number, named after Robert Daniel Carmichael (1879–1967), who first studied them in 1909/10. These numbers have the interesting property that they are pseudoprime to *any* basis a , motivating

DEFINITION 3.3.7 (Carmichael number). *A (composite) number for which $a^n = a$ modulo n for all integers a is called a Carmichael number.*

EXAMPLE 3.3.8. The integer $n = 561 = 3 \cdot 11 \cdot 17$ is the smallest Carmichael number. Also the numbers

1105, 1729, 2465, 2821, 6601, 8911, 10585, 15841, 29341

are Carmichael numbers. ◇

Given the prime factorization of n it is simple to tell whether n is a Carmichael number. The criterion was found by Alwin Reinhold Korselt (1864–1947), in 1899, more than a decade before Carmichael gave the first example. It reads as the

KORSELT CRITERION 3.3.9. *An integer n is a Carmichael number if and only if n is positive, composite, squarefree, and for each prime p dividing n the number $p - 1$ divides $n - 1$.*

For a proof of the theorem see, for example, Crandall & Pomerance (2005, section 3.4.2). By the above discussion there are numbers for which the Fermat Test 3.3.4 completely fails. This would be not that bad if there were only finitely many Carmichael numbers, or if all Carmichael numbers would have at least one small prime factor. The latter was disproven in Alford *et al.* (1994a) under a certain number theoretic conjecture, and the former was disproven unconditionally and is summed up in

THEOREM 3.3.10 (Alford *et al.* 1994b). *There are infinitely many Carmichael numbers. More precisely, for a sufficiently large x the number $C(x)$ of Carmichael numbers up to x satisfies $C(x) > x^{2/7}$. □*

So this somewhat destroys all hope to use the Fermat Test 3.3.4 for testing primality. How can we fix that? We could try to find a stronger condition than the Fermat condition that holds for all primes and hope that then there are not infinitely many composites on which the test always fails.

a	pseudoprimes to the base a
2	341, 561, 645, 1105, 1387, 1729, 1905, 2047, 2465, 2701
3	91, 121, 286, 671, 703, 949, 1105, 1541, 1729, 1891
4	15, 85, 91, 341, 435, 451, 561, 645, 703, 1105
5	4, 124, 217, 561, 781, 1541, 1729, 1891, 2821, 4123
6	35, 185, 217, 301, 481, 1105, 1111, 1261, 1333, 1729
7	6, 25, 325, 561, 703, 817, 1105, 1825, 2101, 2353

Table 3.3.2: The first ten Euler pseudoprimes to the bases $a = 2 \dots 7$.

3.3.2. The Solovay-Strassen test. Consider again the Fermat equation (3.3.1). From it we have deduced Euler's Criterion 3.1.16 that says that for any prime p we have $\left(\frac{a}{p}\right) = a^{\frac{p-1}{2}}$ modulo p . Can we use this for a primality test? Similarly to our definitions above, let us call an odd integer n an *Euler-Jacobi probable prime* to the base a , or short an *Euler probable prime* to the base a , if

$$(3.3.11) \quad \left(\frac{a}{n}\right) = a^{\frac{n-1}{2}} \text{ modulo } n.$$

Accordingly, we call a composite integer n that satisfies (3.3.11) an *Euler pseudoprime* to the base a . Also here we exclude the base $a = 1$ and consider just the case $a \geq 2$. Additionally, we now need to require that a and n are coprime. We do not need to explicitly compute a greatest common divisor here, since the computation is already subsumed by checking whether the value of the Jacobi symbol equals zero. From the definitions it becomes directly clear that the notion of an Euler probable prime is stronger than the notion of a Fermat probable prime: Indeed, by squaring both sides of (3.3.11) we see that any Euler probable prime to the base a is also a Fermat probable prime to the base a . The converse is, however, not true. Also the statement becomes false if we drop the coprimality condition on a . This is due to the fact that when both sides of (3.3.11) are zero (making the equation true), then squaring will not yield the Fermat equation (3.3.1).

EXAMPLE 3.3.12. As shown in Example 3.3.2 the number $n = 15$ is a Fermat pseudoprime to the base $a = 4$. It is, however, not an Euler pseudoprime to the base 4 since $4^{\frac{15-1}{2}} = 4 \neq 1 = \left(\frac{4}{15}\right)$ modulo 15. In Table 3.3.2 one finds the first Euler pseudoprimes for several choices of a . \diamond

SOLOVAY-STRASSEN TEST 3.3.13.

Input: An odd integer $n > 3$ and an integer $1 < a < n - 1$.

Output: Either “Euler probable prime to the base a ” or “composite”.

1. Compute $b = \left(\frac{a}{n}\right)$ using Algorithm 3.1.19.
2. If $b = 0$ then
3. Return “composite”.
4. Compute $c = a^{\frac{n-1}{2}}$ in \mathbb{Z}_n using the Fast Exponentiation Algorithm 3.1.11.
5. If $b = c$ then
6. Return “Euler probable prime to the base a ”.
7. Else
8. “composite”.

The asymptotic runtime of the algorithm is the same as the asymptotic runtime of the Fermat Test 3.3.4, namely $\mathcal{O}(\log n)$ arithmetic operations modulo n . What can we say about the error probability of the algorithm?

THEOREM 3.3.14. *For all integers n , define the set*

$$E(n) = \left\{ a \in \mathbb{Z}_n^\times \mid \left(\frac{a}{n}\right) = a^{\frac{n-1}{2}} \text{ modulo } n \right\}.$$

Then $E(n) = \mathbb{Z}_n^\times$ if and only if n is prime. For composite n , we have $\#E(n) \leq \frac{1}{2}\varphi(n)$, where $\varphi(m)$ is the Euler φ -function (see Definition 2.1.10).

PROOF. Assume n is an odd prime. Then $\left(\frac{a}{n}\right) = a^{\frac{n-1}{2}}$ modulo n by (3.3.11). On the other hand, assume that n is composite but $E(n) = \mathbb{Z}_n^\times$. Then n is a Carmichael number since for all $a \in \mathbb{Z}_n^\times$, we have $a^{n-1} = \left(\frac{a}{n}\right)^2 = 1$. Thus by the Korselt Criterion 3.3.9 the integer n is composite and squarefree and we can write $n = p \cdot m$ for a prime p and an integer $m > 1$ coprime to p . Take some quadratic non-residue $b \in \mathbb{Z}_p^\times$, i.e. $\left(\frac{b}{p}\right) = -1$. Then there is an integer a such that $a = b$ in \mathbb{Z}_p and $a = 1$ in \mathbb{Z}_m . By the definition of the Jacobi symbol, we have

$$\left(\frac{a}{n}\right) = \left(\frac{a}{p}\right) \cdot \left(\frac{a}{m}\right) = \left(\frac{b}{p}\right) \cdot \left(\frac{1}{m}\right) = -1.$$

Thus by assumption $a^{\frac{n-1}{2}} = -1$ modulo m , contradicting $a = 1$ modulo m . This implies that $a \in \mathbb{Z}_n^\times \setminus E(n)$ and since $E(n)$ is a subgroup of \mathbb{Z}_n^\times the theorem is proven. \square

The theorem implies that the Solovay-Strassen Test 3.3.13 will answer correctly “composite” on a composite input n in at least half of the choices of a . Thus we can decide compositeness of an integer n in random polynomial time (by calling the Solovay-Strassen Test 3.3.13 repeatedly with randomly selected parameter a coprime to n), obtaining

THEOREM 3.3.15 (Solovay & Strassen 1977). *The set of composites can be decided in randomized polynomial time.* \square

The drawback of this approach is that if one analyzes closely the running time of the algorithm one notes that the algorithm is only roughly half as fast as the Fermat Test 3.3.4, since besides a power one also needs to compute the Jacobi symbol. Can we get rid of this tiny last obstacle?

3.3.3. Miller and Rabin: The strong primality test. Let us take another closer look at the Fermat equation (3.3.1). Assuming our parameter a is coprime to the odd number we wish to test for primality, then the exponent $n - 1$ to which we take a is even and we will have to perform several squarings on the way of computing $b = a^{n-1}$. Now if the number we are testing is indeed prime, then the square roots of 1 are ± 1 . This leads to

THEOREM 3.3.16. *Let p be a prime and $a \in \mathbb{Z}$ such that a is not divisible by p . Write $p - 1 = 2^s \cdot t$, with t odd. Then*

$$\begin{cases} a^t = 1 \text{ in } \mathbb{Z}_p, \text{ or} \\ a^{2^i t} = -1 \text{ in } \mathbb{Z}_p \text{ for some } 0 \leq i < s - 1. \end{cases} \quad \square$$

This fact was first used for a primality test by M. M. Артюхов¹¹ in 1966/67. A decade later the test was rediscovered by Brillhart, Lehmer & Selfridge (1975). Crandall & Pomerance attribute the rediscovery to John Lewis Selfridge (1927–2010). At about the same time the test was published again in Miller (1975) as a *deterministic* primality test (see below) and made probabilistic in Rabin (1980).

As before, we call an integer n with $n - 1 = 2^s \cdot t$ for an odd t a *strong probable prime* to the base a , if

$$(3.3.17) \quad \begin{cases} a^t = 1 \text{ in } \mathbb{Z}_n, \text{ or} \\ a^{2^i t} = -1 \text{ in } \mathbb{Z}_n, \text{ for some } 0 \leq i < s - 1, \end{cases}$$

¹¹M. M. Artjuhov

a	Euler pseudoprimes to the base a
2	561, 1105, 1729, 1905, 2047, 2465, 3277, 4033, 4681, 6601
3	121, 703, 1729, 1891, 2821, 3281, 7381, 8401, 8911, 10585
4	341, 561, 645, 1105, 1387, 1729, 1905, 2047, 2465, 2701
5	781, 1541, 1729, 5461, 5611, 6601, 7449, 7813, 11041, 12801
6	217, 481, 1111, 1261, 1729, 2701, 3589, 3913, 5713, 6533
7	25, 325, 703, 2101, 2353, 2465, 3277, 4525, 11041, 13665

Table 3.3.3: The first ten strong pseudoprimes to the bases $a = 2 \dots 7$.

where we exclude the base $a = 1$ and consider just the case $a \geq 2$. If the strong probable prime n is composite, we call n a *strong pseudoprime* to the base a . One can see that each strong pseudoprime is also an Euler pseudoprime from

THEOREM 3.3.18 (Pomerance *et al.* 1980). *Any strong pseudoprime n to the base a is also Euler pseudoprime to the base a .* \square

EXAMPLE 3.3.19. As shown in Example 3.3.12 the Carmichael number $n = 561 = 1 + 2^4 \cdot 35$ is a Euler pseudoprime to the base $a = 2$. It is, however, not a strong pseudoprime to the base 2 since $2^{35} \not\equiv \pm 1$ and $2^{70} \not\equiv 2^{140} \not\equiv -1$ but $2^{280} \equiv 1$ modulo 561. In Table 3.3.3 one finds the first strong pseudoprimes for several choices of a . \diamond

STRONG TEST 3.3.20.

Input: An odd integer $n > 3$, written as $n - 1 = 2^s \cdot t$, and an integer $1 < a < n - 1$.

Output: Either “Strong probable prime to the base a ” or “composite”.

1. Compute $b = a^t$ in \mathbb{Z}_n using the Fast Exponentiation Algorithm 3.1.11.
2. If $b = \pm 1$ then
3. Return “Strong probable prime to the base a ”.
4. For i from 2 to $s - 1$ do 5–7
5. Set $b = b^2$ in \mathbb{Z}_n .
6. If $b = -1$ then
7. Return “Strong probable prime to the base a ”.
8. Return “composite”.

The asymptotic runtime of the algorithm is clearly the same as the asymptotic runtime of the Fermat Test 3.3.4, namely $\mathcal{O}(\log n)$ arithmetic operations modulo n . By Theorem 3.3.18, the error probability is bounded by $\frac{1}{2}$. One can show a bit more, which we state as

THEOREM 3.3.21 (Monier 1980, Rabin 1980). *For all integers n , define the set*

$$S(n) = \{a \in \mathbb{Z}_n^\times \mid n \text{ is a strong probable prime to the base } a\}.$$

Then $S(n) = \mathbb{Z}_n^\times$ if and only if n is prime. For composite n we have $\#S(n) \leq \frac{1}{4}\varphi(n)$. \square

We call any element from $\mathbb{Z}_n^\times \setminus S(n)$ a *strong witness* (for the compositeness) of n . As already indicated at the end of Section 2.2.2 the Strong Test 3.3.20 can be made deterministic under the Extended Riemann Hypothesis 2.2.20, thus conditionally deciding primality in deterministic polynomial time:

MILLER PRIMALITY TEST 3.3.22.

Input: An odd integer $n > 3$.

Output: Either “prime” or “composite”.

1. Set $W = \min(\lfloor 2 \ln^2 n \rfloor, n - 1)$.
2. For a from 2 to W do 3–5
3. Call the Strong Test 3.3.20 on input n and a .
4. If n is “composite” then
5. Return “composite”.
6. Return “prime”.

The asymptotic runtime of the algorithm is $\mathcal{O}(\log^3 n)$ arithmetic operations modulo n , since for $n > 17$ we will always have $2 \ln^2 n \leq n - 1$. For the correctness of the algorithm we state

THEOREM 3.3.23 (Miller 1975). *If we assume the Extended Riemann Hypothesis 2.2.20 then the least witness for an odd composite integer n is smaller than $2 \ln^2 n$. \square*

We can also construct a random compositeness test by calling the Strong Test 3.3.20 on a randomly selected parameter a , obtaining the

RANDOM COMPOSITENESS TEST 3.3.24.

Input: An integer $n > 3$.

Output: Either “Strong probable prime to the base a ” or “composite with witness a ”.

1. Select a uniformly at random from the interval $[2, n - 2]$.
2. Call the Strong Test 3.3.20 on input n and a .
3. If n is “Strong probable prime to the base a ” then
4. Return “Strong probable prime to the base a ”.
5. Return “Composite with witness a ”.

To understand the runtime of the algorithm, it is necessary to first analyze how long it will take to generate an integer a uniformly at random from the interval $[2, n - 2]$. A straightforward approach is to repeatedly generate an integer with $k = \lfloor \log_2 n \rfloor + 1$ bits, until the result is in the desired interval. How often will we have to iterate? *Potentially* an unbound number of times! But the *expected* number of iterations can be estimated as follows: If we are running a while loop that exits independently with probability p , then it is not difficult to show that the expected number of iterations is $\frac{1}{p}$: Write $q = 1 - p$ and denote by X the number of iterations until the loop exits. Then for any positive integer i we have

$$\text{prob}(X = i) = q^{i-1} \cdot p.$$

By the definition of the expected value $\mathcal{E}X$ we obtain

$$\mathcal{E}X = p \cdot \sum_{i \geq 0} i \cdot q^{i-1} = p \cdot \sum_{i \geq 0} \frac{\partial}{\partial q} \cdot q^i = p \cdot \frac{\partial}{\partial q} \frac{1}{1 - q} = \frac{1}{p}.$$

Thus, in the above selection process we have an *expected runtime* of at most 2 iterations, which gives for the runtime of the Random Compositeness Test 3.3.24 an expected runtime of $\mathcal{O}(\log n)$ arithmetic operations modulo n . The error probability of the algorithm is by Theorem 3.3.21 at most $\frac{1}{4}$. To decrease this probability as much as we wish, we can call the Random Compositeness Test 3.3.24 repeatedly, giving after t calls of the test an error probability of at most 4^{-t} . We can therefore use the

PRACTICAL PRIME GENERATOR 3.3.25.

Input: A bound x and an error probability ε .

Output: A randomly selected number $n \leq x$ that is prime with probability at least $1 - \varepsilon$.

1. Set $t = \frac{\ln \ln x - \ln \varepsilon}{\ln 4}$.
2. Repeat
3. Choose an odd integer n at random from the interval $[3, x]$.
4. Until the Strong Test 3.3.20 returns on input n “Strong probable prime to the base a ”, a number of t times in succession.
5. Return n .

It can be shown that the algorithm’s output is indeed prime with probability at least $1 - \varepsilon$ (where the probability is taken over the random choices of the algorithm, since, of course, any fixed number is prime with probability 1 or 0, depending on whether it actually *is* prime or not). The algorithm uses an expected number of $\mathcal{O}(k)$ calls to the Strong Test 3.3.20, where k is the bitlength of x . For details, see for example Bach & Shallit (1996, section 9.7).

3.3.4. The AKS test. We finish the section on primality testing with a short description of the famous AKS test (Agrawal, Kayal & Saxena 2004), which is a deterministic polynomial time algorithm for testing primality. It is based on the following simple

LEMMA 3.3.26. *Let $n \geq 2$ be a positive integer and $a \in \mathbb{Z}$ coprime to n . Then n is prime if and only if*

$$(3.3.27) \quad (x + a)^n = x^n + a \text{ in } \mathbb{Z}_n[x]$$

PROOF. Consider the expansion of $(x + a)^n$. For $0 \leq j \leq n$ the coefficient of x^j is $\binom{n}{j} a^{n-j}$. If n is prime then $\binom{n}{j} = 0$ in \mathbb{Z}_n for $1 < j < n$ and $\binom{n}{0} = \binom{n}{n} = 1$. Consider now a composite n , let q be a prime and $k \in \mathbb{N}_{>0}$ maximal, such that q^k divides n . Then q^k does not divide $\binom{n}{q}$ and is coprime to a^{n-q} and thus nonzero in \mathbb{Z}_n . This proves the lemma. \square

Note that equation (3.3.27) holds *if and only if* n is prime. The problem is, however, that the expansion of $(x + a)^n$ in $\mathbb{Z}[x]$ has far too many terms to be evaluated quickly. Agrawal, Kayal & Saxena’s idea was now to consider equation (3.3.27) modulo small degree polynomials $f \in \mathbb{Z}_n[x]$, i.e. we check

$$(3.3.28) \quad (x + a)^n = x^n + a \text{ in } \mathbb{Z}_n[x]/(f(x))$$

This has the huge benefit that it enables us to actually evaluate the condition quickly, but it might destroy the equivalence of the statement and the primality of n . Consider, for example, the case $a = 1$ and the polynomial $f = x + 1$. Then equation (3.3.28) reads as

$$2^n = 2 \text{ in } \mathbb{Z}_n$$

which is nothing but the Fermat condition (3.3.1) to the base 2. The brilliant idea was now to consider equation (3.3.28) using a polynomial of the form $x^r - 1$ and check the condition for several choices of a giving the

AKS TEST 3.3.29.

Input: An integer $n \geq 2$.

Output: Either “prime” or “composite”.

1. Decide whether n is a perfect power using the Perfect Power Test 3.2.3.
2. If n is a perfect power then
3. Return “composite”.
4. Find the smallest r , such that the order of n in \mathbb{Z}_r^\times exceeds $\log_2^2 n$.
5. If n has a proper factor in $[2, \sqrt{\varphi(r)} \log n]$ then
6. Return “composite”.
7. For a from 1 to $\sqrt{\varphi(r)} \log n]$ do 8–9
8. If $(x + a)^n \neq x^n + a$ in $\mathbb{Z}_n[x]/(x^r - 1)$ then
9. Return “composite”.
10. Return “prime”.

Following Crandall & Pomerance (2005, section 4.5.2), the runtime of the algorithm can be estimated as $\mathcal{O}(\ln^{16.5} n)$ bit operations. The correctness follows from

THEOREM 3.3.30 (Agrawal *et al.* 2004). *Let $n \geq 2$ and r be integers such that the order of n modulo r exceeds $\log_2^2 n$. If the congruence $(x + a)^n = x^n + a$ holds in $\mathbb{Z}_n[x]/(x^r - 1)$ for all integers a with $0 \leq a \leq \sqrt{\varphi(r)} \log_2 n$ and n has a prime factor exceeding $\sqrt{\varphi(r)} \log_2 n$, then n is a perfect power. In other words, if n is not a perfect power and has only prime factors in the interval $[2, \sqrt{\varphi(r)} \log_2 n]$, then n must be prime. \square*

COROLLARY 3.3.31 (Agrawal *et al.* 2004). *The set of primes can be decided in deterministic polynomial time. \square*

3.4. Factoring algorithms by sieving

We now focus on computational aspects of the Factorization Problem 2.1.3. There are two fundamentally different approaches for factoring: sieving algorithms, in which one ultimately wishes to find a non-trivial congruence of squares, and group based approaches, where one tries to exploit special properties of the size of certain groups like elliptic curves. We start with a discussion of the former kind of algorithms following in our exposition Pomerance (1996).

Suppose you wish to factor a given integer $n \geq 2$. A very old approach was described first by Fermat in a letter to Marin Mersenne (1588–1648) in 1647: After being asked by Mersenne whether the number $n = 100895598169$ was prime, he responded in this letter that it was indeed not prime and explained how he factored it. He noticed that $x^2 = 32 \cdot n + 1$ is a perfect square, i.e. $x^2 - 1 = 32 \cdot n$. He then deduced that $32 \cdot n = (x-1)(x+1)$ and also from it one easily obtains $n = 898423 \cdot 112303$. The key step in his argumentation was to express a multiple of n as a difference of two squares, which is the basis for all sieving algorithms, explained in the following sections.

3.4.1. Pomerance and the Quadratic Sieve. The basic problem we have with Fermat's method for factoring n is to find a way to construct integers x and y such that $x^2 - y^2$ is a multiple of n or, in other words, how to find a nontrivial congruence $x^2 = y^2$ modulo n . We can safely assume here that n is neither even nor a perfect power, since deciding either property can be done efficiently (by computing division with remainder by 2 and by using Algorithm 3.2.2, respectively). The basic idea in Pomerance (1985) was to try many congruences of the form $x_i^2 = a_i$ such that $\prod_i a_i = y^2$ is a square and to factor n by computing $\gcd(x - y, n)$.

EXAMPLE 3.4.1. Suppose we wish to factor $n = 5029$. We have

$$\begin{aligned} 71^2 &= 5041 = 12 \text{ modulo } n, \\ 72^2 &= 5184 = 155 \text{ modulo } n, \\ 73^2 &= 5329 = 300 \text{ modulo } n. \end{aligned}$$

Note that $12 \cdot 300 = 3600 = 60^2$, so we have $(71 \cdot 73)^2 = 60^2$ modulo n . Note that $71 \cdot 73 = 154 \neq 60$ modulo n , so we can factor n by computing $\gcd(154 - 60, n) = 47$, giving the factorization $n = 47 \cdot 107$. \diamond

The question that remains is now to compute a subset of the x_i , such that the product of the corresponding a_i is a square, when given lots of congruences

of the form $x_i^2 = a_i$ modulo n . Reconsider the values of Example 3.4.1: We took two of the three congruences to create a square. When considering the factorizations of the right hand sides of the congruences, it is striking to observe that we actually kept those that are composed of very small prime factors only ($12 = 2^2 \cdot 3$ and $300 = 2^2 \cdot 3 \cdot 5^2$), and threw away the one with a relatively large prime factor ($155 = 5 \cdot 31$). Making this idea systematic, let us restrict the set of x_i we keep for searching our subset to those x_i for which $x_i^2 = a_i$ has small prime factors only, i.e. we keep only those a_i that are B -smooth for some fixed bound B (see Definition 2.3.4).

How many B -smooth numbers do we have to collect until we can be sure that a product of one subset of them is a perfect square? The answer to this question was first given in Morrison & Brillhart (1975): Let us associate to each B -smooth number $m = \prod_{1 \leq i \leq \pi(B)} p_i^{e_i}$, with all $e_i \geq 0$, an exponent vector $\vec{e}(m) = (e_1, e_2, \dots, e_{\pi(B)})$. If m_1, \dots, m_k are all B -smooth, then $\prod_{1 \leq j \leq k} m_j$ is a square if and only if $\sum_{1 \leq j \leq k} \vec{e}(m_j)$ has even coordinates only. Thus, it makes sense to consider only the exponent vector modulo 2, and the search for a subset of B -smooth integers, whose product is a square boils down to finding a linear combination of vectors, whose sum is zero (modulo 2). The amazing advantage of this point of view is that we can readily say how many vectors we need to collect until we can be sure that one subset sums up to zero: Namely, we need to collect more vectors than the dimension of the vector-space, i.e. $\pi(B) + 1$ of them. Also the task of finding a linear dependent combination of vectors can be easily accomplished by a row-reduction of the matrix formed by these vectors.

There is still a tradeoff we can make: the choice of B . If we select B to be very small, then the number of B -smooth numbers we have to search for is very small, but it is also very difficult to find a *single* B -smooth number (since extracting square roots modulo n is difficult). If on the other hand B is selected too large, then we need to collect many relations, even though finding a single relation is comparatively easy to find. It was heuristically shown in Canfield *et al.* (1983) that for an optimal choice of B the heuristic runtime of the sketched *Quadratic Sieve* algorithm is

$$(3.4.2) \quad L(n) = \exp\left(\sqrt{\ln n \ln \ln n}\right),$$

and the optimal choice of B is $L(n)^{\frac{1}{2}}$.

3.4.2. Pollard's idea. A considerable speedup of the Quadratic Sieve was given in Pollard (1988). He noticed that if n is very close to a power, it is easy to find a polynomial f that is irreducible over the integers and an integer m , such that $f(m) = 0$ modulo n .

EXAMPLE 3.4.3. Let $n = 2^{2^9} + 1$. We have $8n = 2^{515} + 8$, so $m = 2^{103}$ and $f(x) = x^5 + 8$ will do the job. \diamond

Consider a complex root α of the polynomial f . The ring $\mathbb{Z}[\alpha]$ contains all polynomial expressions in α with integer coefficients. Furthermore, since $f(\alpha) = f(m) = 0$ modulo n , the map

$$\varphi: \begin{array}{ccc} \mathbb{Z}[\alpha] & \longrightarrow & \mathbb{Z}_n, \\ \alpha & \longmapsto & m \end{array}$$

is a ring homomorphism. Also, by construction, this map is well defined! If we have now a finite set S of coprime integer pairs (a, b) with the properties that

1. the product of the algebraic integers $a - b\alpha \in \mathbb{Z}[\alpha]$ is a square, say γ^2 ,
2. the product of all integers $a - bm \in \mathbb{Z}$ is a square, say v^2 ,

then if we replace each occurrence in α in γ by m , calling the resulting integer u , we have

$$\begin{aligned} u^2 &= \varphi(\gamma)^2 = \varphi(\gamma^2) = \varphi\left(\prod_{(a,b) \in S} (a - b\alpha)\right) \\ &= \prod_{(a,b) \in S} \varphi(a - b\alpha) = \prod_{(a,b) \in S} (a - bm) = v^2 \text{ modulo } n \end{aligned}$$

obtaining a (hopefully) nontrivial congruence of squares. How can we obtain such a set of pairs S ? The second property is somewhat simple to achieve by using exponent vectors and a sieve by, for example, fixing b and running for a through some given interval, changing b and start our search for a all over again. But how can we possibly simultaneously assure the first property? It was Pollard's idea that when $\mathbb{Z}[\alpha]$ is the full ring of algebraic integers in $\mathbb{Q}[\alpha]$, and if $\mathbb{Z}[\alpha]$ admits unique factorization and we know a basis for the units, then we could also create exponent vectors for the algebraic integers $a - b\alpha$. Thus, in order to ensure both properties simultaneously, we just need longer exponent vectors: for the small primes numbers, for the sign of the algebraic integer and for the small primes in $\mathbb{Z}[\alpha]$ (whatever these are, for now).

The procedure, called the *Special Number Field Sieve*, seems neat and to be working, but relies heavily on assumptions on the structure of the ring $\mathbb{Z}[\alpha]$ that do not hold in general! So, what can we do?

3.4.3. Towards the General Number Field Sieve. Let us review the sieving process in the Special Number Field Sieve: There we were looking for pairs (a, b) such that the product of the $a - bm$ is a square in \mathbb{Z} and, additionally, that the product of the $a - b\alpha$ is a square in $\mathbb{Z}[\alpha]$. The former can be rephrased as follows: Write $G(a, b) = a - bm$. This is a degree one homogeneous polynomial and we sieve this polynomial for smooth values (e.g. by letting b run up to some large bound M for each choice of a up to M). In this procedure, the so called cofactorization step, we throw away all pairs (a, b) with $\gcd(a, b) > 1$ to avoid trivial redundancies. But what can we do on the algebraic side?

Let us look again at the second condition to have the product of the $a - b\alpha$ being a square in $\mathbb{Z}[\alpha]$. Write $\alpha_1, \dots, \alpha_d$ for the complex roots of the chosen polynomial f with $\alpha = \alpha_1$. Then the norm $N(\beta)$ of any element $\beta = \sum_{0 \leq i < d} s_i \alpha^i \in \mathbb{Q}[\alpha]$ is the product of all conjugates of β , i.e.

$$N(\beta) = \prod_{0 \leq j < d} \left(\sum_{0 \leq i < d} s_i \alpha_j^i \right).$$

An important property of the norm is that it is always a rational number and indeed an integer if the coefficients s_i are integers. Additionally it is easy to show that the norm is a multiplicative function, i.e. $N(\beta\beta') = N(\beta)N(\beta')$ for all $\beta, \beta' \in \mathbb{Q}[\alpha]$. This implies that if β is a square in the number field, then $N(\beta)$ is the square of an *integer*. Thus, we have found a *necessary* condition for β being a square in the number field! We can rephrase this condition in the following way: Let us reconsider the definition of the norm of a field element $a - b\alpha$. We have

$$N(a - b\alpha) = \prod_{0 \leq i < d} (a - b\alpha_i) = b^d f(a/b).$$

The right hand expression is nothing but the homogeneous form $F(a, b)$ of the polynomial f . Thus, we can explicitly represent the norm of an element by the evaluation of a given bivariate polynomial F . Correspondingly, we call $a - b\alpha$ a B -smooth element, if $F(a, b)$ is B -smooth. We have

LEMMA 3.4.4 (Buhler *et al.* 1993). *Let \mathcal{S} be a set of pairs of coprime integers a, b such that $a - b\alpha$ is B -smooth, and $\prod_{(a,b) \in \mathcal{S}} (a - b\alpha)$ is a square in the ring of algebraic integers in $\mathbb{Q}[\alpha]$. Then*

$$\sum_{(a,b) \in \mathcal{S}} \vec{e}(a - b\alpha) = 0 \text{ modulo } 2,$$

where the vector \vec{e} is defined component-wise for entries (p, r) as $\vec{e}[(p, r)] = 0$ if $a \neq br$ modulo p and the exponent of p in the prime factorization of $F(a, b)$ otherwise. \square

There are still several technical problems to overcome. One of them is that the ring $\mathbb{Z}[\alpha]$ will in general be a subset of the set of algebraic integers in $\mathbb{Q}[\alpha]$. This in turn could lead to the problem that even if we have $\beta = \gamma^2$ it could still be that γ is not an element of $\mathbb{Z}[\alpha]$, which we need for Pollard's idea (see Section 3.4.2). To overcome this issue, we state

LEMMA 3.4.5. *If $f(x)$ is a monic irreducible polynomial in $\mathbb{Z}[x]$ with complex root α , then for any algebraic integer β of the number field $\mathbb{Q}(\alpha)$ the element $f'(\alpha)\beta$ is in $\mathbb{Z}[\alpha]$.* \square

For a proof, see Crandall & Pomerance (2005, section 6.2.4). We will use Lemma 3.4.4 as follows: Instead of looking for coprime integers a, b for which $\prod_{(a,b) \in \mathcal{S}} (a - b\alpha)$ is a square in $\mathbb{Z}[\alpha]$, we search in the same way for an element γ that is a square in the *algebraic integers of $\mathbb{Q}(\alpha)$* . Then we apply Lemma 3.4.5 to find out that the element $f'(\alpha)^2 \prod_{(a,b) \in \mathcal{S}} (a - b\alpha)$ is a square in $\mathbb{Z}[\alpha]$.

The last issue we briefly discuss here is to extend our techniques to have a *sufficient* condition for finding a square, since Lemma 3.4.4 only supplies us with a *necessary* one. The trick is to employ the following heuristic method: Suppose you wish to decide whether an integer m is a square. Then (heuristically) it is sufficient to check whether the integer is a square modulo several primes which in turn can be easily done by evaluating repeatedly a Legendre symbol (see Definition 3.1.14). A suitable extension of the idea that also works for algebraic integers can be found in Crandall & Pomerance (2005, section 6.2.4).

Using the methods we just sketched, we are able to adapt Pollard's idea to work for arbitrary integers. Using the notation (3.4.2), heuristically the algorithm runs with

$$L(n)\sqrt{(2+2\varepsilon)/d+o(1)}$$

operations, where d is the degree of the selected polynomial f and all coefficients of f are bounded by $n^{\varepsilon/d}$. This runtime is (heuristically) much better than the one of the Quadratic Sieve in Section 3.4.1 if the polynomial f is selected suitably (see Kleinjung 2006). For more details on the General Number Field Sieve, see Lenstra & Lenstra (1993) and in particular the article of Buhler *et al.* (1993).

3.4.4. Excursus: Index calculus for discrete logarithms. It is interesting to observe that any advance in the complexity of the Factorization Problem 2.1.3, also lead to a corresponding advance in computing discrete logarithms in the multiplicative group of finite fields. It boils down to the

DISCRETE LOGARITHM PROBLEM 3.4.6. *Let $G = \langle g \rangle$ be a cyclic group. Given an element $a \in G$, find an integer $\alpha \geq 0$, such that $a = g^\alpha$.*

It was proven by Shoup (1997) that this problem is hard, by showing that any generic algorithm computing the discrete logarithm in the group G must run $\Omega(p^{\frac{1}{2}})$ group operations, where p is the largest prime dividing the order of the group. Indeed, there are many such generic algorithms around, like the Baby-step-giant-step algorithm (Shanks 1969) or the ρ -method (Pollard 1978).

In practice, however, we will typically use either the multiplicative group of a finite field \mathbb{F}_q or the group of points on an elliptic curve (see Section 3.5.1). In all of these specific choices of the group we have much more information about the group law in hand than merely a black box. In the case of elliptic curves, this fact did so far not give any faster algorithms for the Discrete Logarithm Problem 3.4.6, but in the case of the multiplicative group of a finite field there are better algorithms known. For such fields the best algorithm known to solve this problem is the *index-calculus method*. The origin of the method dates back of the beginning to the 20th century (Kraitchik 1922), but the algorithm as we know it today was published by Adleman (1979). For simplicity of exposition, we will sketch the algorithm for prime fields only.

The idea for computing the discrete logarithm of $a = g^\alpha$ (to the base g) is very similar to the Quadratic Sieve described in Section 3.4.1, but instead of searching for a non-trivial congruence of squares, we search for congruences of the form

$$g^\beta = p_1^{\beta_1} \cdots p_k^{\beta_k} \text{ modulo } p,$$

where p_1, \dots, p_k are small primes. These congruences give in turn raise to congruences for the *exponents* via

$$\beta = \beta_1 \log_g(p_1) \cdots \beta_k \log_g(p_k) \text{ modulo } p - 1,$$

and we can solve this system using linear algebra for the unknown values $\log_g(p_1), \dots, \log_g(p_k)$. Once we know them, we look for a relation of the form $ag^\gamma = p_1^{\gamma_1} \cdots p_k^{\gamma_k}$ giving

$$\alpha = \log_g a = -\gamma + \gamma_1 \log_g p_1 + \cdots + \gamma_k \log_g p_k,$$

which is the desired discrete logarithm of a to the base g . One can show that the resulting algorithm has a runtime of $L(p)^{2+o(1)}$ using the notation (3.4.2). For details on the analysis, see Crandall & Pomerance (2005, section 6.4.1).

3.5. Factoring algorithms using elliptic curves

We now describe a method for factoring integers that is best suited for “medium-sized” integers n , first proposed in Lenstra (1987). The concepts behind the algorithm are substantially different from the ones used in the sieving algorithms described above: Instead of searching for a nontrivial congruence of squares modulo n , we will use certain smoothness properties of the size of an underlying group we are working in. Let us illustrate this by recalling Pollard’s $(p-1)$ -method, invented in 1974: For simplicity of exposition assume you are given an integer $n = pq$, where p and q are large primes (for a detailed discussion of these kinds of integers, see Chapter 8). Select randomly an element $a \in \mathbb{Z}_n^\times$ and a sufficiently large bound B and compute

$$b = a^{B!} \text{ in } \mathbb{Z}_n.$$

The algorithm then returns $g = \gcd(b-1, n)$ hoping that it will give a non-trivial factor of n . When does this procedure work? Assume we have selected B such that $\#\mathbb{Z}_p = p-1$ is B -smooth. Then it is very likely that $B!$ is a multiple of $p-1$ and in \mathbb{Z}_p we would have by Theorem 3.1.9 that $b = a^{B!} = 1$ modulo p . If additionally $q-1$ is *not* B -smooth, say a prime $\ell > B$ divides it, it would on the other hand be very likely that the order of a in \mathbb{Z}_q^\times is a multiple of ℓ from which we could readily deduce that $b = a^{B!} \neq 1$ modulo q . In other words, $b-1$ is a multiple of p but not a multiple of q , implying that $g = \gcd(b-1, n) = p$ is a non-trivial factor of n .

The problem with this procedure is, of course, to find an appropriate bound B . Also, if $p-1$ and $q-1$ both have a large prime factor (these are then called *strong primes*), the $p-1$ method would not succeed in efficiently finding a non-trivial factor. Additionally, the structure of the groups \mathbb{Z}_p^\times and \mathbb{Z}_q^\times are inherent to the input of the algorithm, which we cannot control. Fortunately, Lenstra’s elliptic curve factorization method provides a way out of this problem (see Section 3.5.4).

3.5.1. Arithmetic in Weierstraß form. In algebraic geometry, an *elliptic curve* is a smooth, projective, algebraic curve of genus one. We will, indeed, need that later when we consider elliptic curves in Edwards form (see Chapter 4). Such curves were already studied in the ancient world,



Figure 3.5.1: Left: The page of the 1621 edition of Diophant’s “Arithmetica” with the famous margin on it. Right: The 1670 edition of the same page including Fermat’s comment.

notably by Διόφαντος ὁ Ἀλεξανδρεύς (ca. 200–284)¹² in his multi-volume epos called “Arithmetica” (most of which is lost nowadays). Indeed, this work was the inspiration for *Fermat’s last theorem* on non-trivial integer solutions of the equation $x^n + y^n = z^n$ for $n \geq 3$. He noted on the margin in his own copy of the “Arithmetica” that he would have a marvelous proof that no such non-trivial integer-triples exist (Figure 3.5.1 shows the famous margin as well as an annotated reprint of this particular page of Fermat’s copy). The proof of Fermat’s last theorem was finally given in Wiles (1995).

More than two centuries after Fermat, Karl Theodor Wilhelm Weierstraß (1815–1897) studied elliptic curves in much more detail (see Weierstraß (1895a) and Weierstraß (1895b)), even though his research focused more on the complex analytic aspects of these objects. Elliptic curves were finally introduced to public key cryptography, independently, in Koblitz (1987) and Miller (1986). When one consults the main textbooks on the topic, like Silverman (1986), an elliptic curve in the above sense (smooth, projective, and of genus one) is introduced as a nonsingular cubic defined over a field F ,

¹²Diophantus of Alexandria

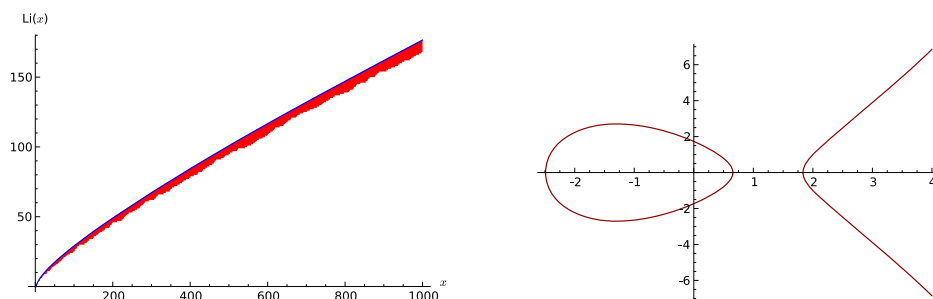


Figure 3.5.2: Real plots of the curves $y^2 = x^3 - 5x + 3$ (left) and $y^2 = x^3 - 3x + 5$ (right).

where we typically require that the characteristic of F is neither 2 or 3. It turns out that we can in this case write any such cubic in (*short*) Weierstraß form

$$y^2 = x^3 + ax + b,$$

where $a, b \in F$ and $4a^3 + 27b^2 \neq 0$. In Figure 3.5.2 one finds two real plots of elliptic curves in Weierstraß form. Actually, there is another point on the curve hidden in the real picture which lies far along the y -axis: a single point \mathcal{O} at infinity. This point serves a special purpose:

ELLIPTIC CURVE ADDITION LAW 3.5.1. *Let F be a field, $a, b \in F$ and let E be an elliptic curve defined over F by the equation $y^2 = x^3 + ax + b$. Let $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ be two points on the curve and let \mathcal{O} be the point at infinity on the curve. Define the following operation on E :*

- (i) $-\mathcal{O} = \mathcal{O}$.
- (ii) $-P = (x_1, -y_1)$.
- (iii) $P + \mathcal{O} = P$.
- (iv) If $P = -Q$ then $P + Q = \mathcal{O}$.
- (v) If $P \neq -Q$ then $P + Q = R = (x_3, y_3)$, with

$$x_3 = m^2 - x_1 - x_2, \quad y_3 = m(x_1 - x_3) - y_1,$$

where the slope m is defined as

$$m = \begin{cases} \frac{3x_1^2 + a}{2y_1} & , \text{ if } x_2 = x_1 \\ \frac{y_2 - y_1}{x_2 - x_1} & , \text{ if } x_2 \neq x_1. \end{cases}$$

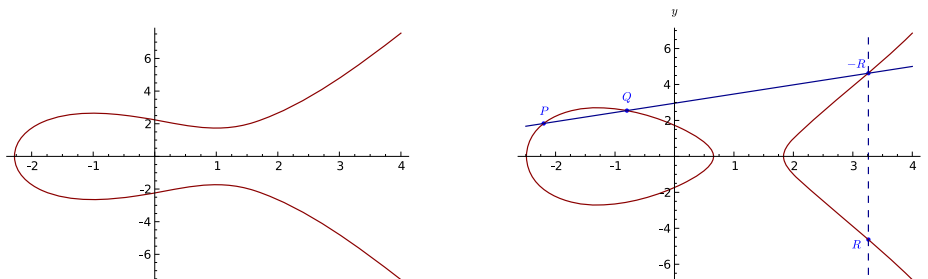


Figure 3.5.3: The addition law on elliptic curves in Weierstrass form.

The intuitive view of the operation just defined is the following: Suppose the two points P and Q are on the curve E . Then, since the curve has degree three, drawing a line through those two points will yield one further point $-R$ on the curve (if P equals Q , we take the tangent at P). Reflecting along the x -axis gives the sum R of the points P and Q . If the line between P and Q is vertical then we end up at the point at infinity, whose negative is defined to be itself. An illustration of this operation can be found in Figure 3.5.3. It turns out that the points on the curve with this operation form a commutative group. More precisely we have

THEOREM 3.5.2 (Cassels 1966). *Let E be an elliptic curve over a field F . Then E forms with the operation given in the Elliptic Curve Addition Law 3.5.1 a commutative group. If $F = \mathbb{F}_q$ is a finite field with $q = p^e$ for a prime p and $e \geq 1$, then the group is either cyclic or isomorphic to a direct product*

$$E \simeq \mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2}$$

with $m_1 \mid m_2$ and $m_1 \mid q - 1$. □

3.5.2. Hasse's theorem and point counting. If $F = \mathbb{F}_q$ is a finite field then the number of points on E is also finite and it makes sense to talk about the group order $\#E$ of the curve (it makes sometimes also sense to talk about the group order of E for some infinite fields, like the rational numbers \mathbb{Q} , but this shall not be of any concern to us). For many cryptographic and cryptanalytic applications it is necessary to be able to compute the order of a given curve E over a finite field \mathbb{F}_q . If $q = p^e$ is small enough this can be easily done by evaluating manually a sum of the form

$$(3.5.3) \quad \#E = q + 1 + \sum_{x \in \mathbb{F}_q} \chi(x^3 + ax + b),$$

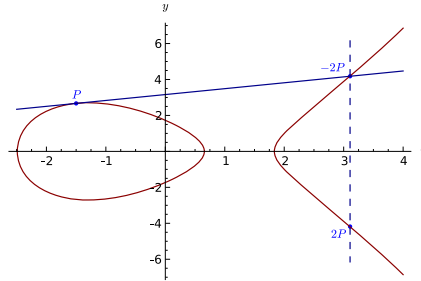


Figure 3.5.4: The number of elliptic curves E over \mathbb{F}_{1009} with exactly m points for all admissible Weierstraß parameters $a, b \in \mathbb{F}_{1009}$. For each m in the Hasse interval $[947, 1073]$ there is at least one elliptic curve with order m .

where χ is the quadratic character of \mathbb{F}_q , similar to the Legendre symbol for prime q (see Definition 3.1.14). One can show that when one knows $\#E$ over any finite field \mathbb{F}_q then we can also compute $\#E$ over any field extension of \mathbb{F}_q efficiently (see for example Washington 2003, section 4.3.1).

How large should we expect $\#E$ to be? If we evaluate χ for a randomly chosen value x , we can expect heuristically that for roughly half of the choices x we have $\chi(x^3 + ax + b) = 1$ while for the other half we have $\chi(x^3 + ax + b) = -1$. This implies that the sum in (3.5.3) is heuristically roughly 0 and thus $\#E \approx q + 1$. That this is indeed the case was first proven by Helmut Hasse (1898–1979) and reads as

THEOREM 3.5.4 (Hasse 1933). *For any elliptic curve E over a finite field \mathbb{F}_q , we have*

$$|\#E - (q + 1)| \leq 2\sqrt{q}. \quad \square$$

Indeed, we have

THEOREM 3.5.5 (Deuring 1941). *For $m \in]q + 1 - 2\sqrt{q}, q + 1 + 2\sqrt{q}[$ there is at least one elliptic curve over \mathbb{F}_q with $\#E = m$. \square*

An illustration of this fact is shown in Figure 3.5.4. But how can we actually compute the size of a given elliptic curve efficiently? We can restrict our attention to primes p since, by the above remark, this directly gives us the result for arbitrary finite fields \mathbb{F}_q . Even though we have formula (3.5.3), it cannot be evaluated for large $q = p$ since the number of summands in there grows exponentially in the size of p . An observation of Schoof (1995) was that we can actually compute $\#E$ modulo ℓ for small primes ℓ and deduce

from it the correct order $\#E$ via the Chinese Remainder Theorem 3.1.12: One can show that if $\#E = q + 1 - t$, then we have for the Frobenius endomorphism $\varphi: (x, y) \rightarrow (x^p, y^p)$, named after Ferdinand Georg Frobenius (1849–1917), the equation

$$\varphi^2 - t\varphi + p = 0,$$

in the endomorphism ring of E . Schoof's observation was now that for any point P on the curve E with coordinates in the algebraic closure of \mathbb{F}_p and with order dividing ℓ we have

$$\varphi^2(P) - (t \bmod \ell)\varphi(P) + (p \bmod \ell) = \mathcal{O}.$$

Using this we can, essentially by trial and error, deduce the group order modulo ℓ via this equation and can recombine the correct order $\#E$ with the Chinese Remainder Theorem 3.1.12. For details, one can consult for example Crandall & Pomerance (2005, section 7.5.2).

3.5.3. Speeding up the arithmetic. For efficient arithmetic on elliptic curves in the spirit of the Elliptic Curve Addition Law 3.5.1, it is important to think about the *representation* of the points on an elliptic curve E . So far, we have only encountered one representation: *Affine* coordinates (x, y) of points on E , where both x and y are elements from the field we are considering the curve over. The advantage of this representation is that it is very intuitive, but the point at infinity \mathcal{O} does not fit very well into this framework. To find a representation that can also express the point at infinity, we switch in this section to *projective* coordinates. To do so, let us reconsider the curve equation $y^2 = x^3 + ax + b$ of E . We can homogenize this equation to obtain

$$(3.5.6) \quad Y^2Z = X^3 + aXZ^2 + bZ^3.$$

Now, for any point (X, Y, Z) on this curve, we will also have that $(\lambda X, \lambda Y, \lambda Z)$ lies on the curve for any non-zero constant λ in the field. Thus, we can regard all such points as equivalent and we write $(X : Y : Z)$ to indicate this fact. It is also easy to identify an affine point (x, y) with the corresponding projective point $(x : y : 1)$. The other way round, we identify a projective point $(X : Y : Z)$ as either the affine point $(X/Z, Y/Z)$, if $Z \neq 0$ or the point at infinity \mathcal{O} , if $Z = 0$. Interestingly, it turns out that there is indeed just *a single* projective point on the curve with $Z = 0$, i.e. a *single* point at infinity: Suppose we have such a point $\mathcal{O} = (X : Y : 0)$ on the curve. Then from (3.5.6) it immediately follows $X = 0$ and thus $\mathcal{O} = (0 : Y : 0) = (0 : 1 : 0)$.

How can this point of view help us to speed up the elliptic curve arithmetic? If one considers the Elliptic Curve Addition Law 3.5.1, then the most expensive operations involved are field inversions. It turns out that by switching to projective coordinates, we get completely rid of inversions and can perform the arithmetic with multiplications, squarings, additions and negations only!

DEFINITION 3.5.7 (Elliptic curve addition law: projective coordinates). *Let F be a field, $a, b \in F$, and let E be an elliptic curve defined over F by the equation $Y^2Z = X^3 + aXZ^2 + bZ^3$. Let $P = (X_1 : Y_1 : Z_1)$ and $Q = (X_2 : Y_2 : Z_2)$ be two points on the curve and let $\mathcal{O} = (0 : 1 : 0)$ be the point at infinity on the curve. Define the following operation on E :*

$$(i) \quad -(X_1 : Y_1 : Z_1) = (X_1 : -Y_1 : Z_1).$$

$$(ii) \quad \text{If } P = -Q \text{ then } P + Q = (0 : 1 : 0).$$

$$(iii) \quad \text{If } P \neq Q \text{ then } P + Q = R = (X_3 : Y_3 : Z_3), \text{ with}$$

$$X_3 = BC, \quad Y_3 = A(B^2X_1Z_2 - C) - B^3Y_1Z_2, \quad Z_3 = B^3Z_1Z_2,$$

where

$$A = Y_2Z_1 - Y_1Z_2, \quad B = X_2Z_1 - X_1Z_2, \quad C = A^2Z_1Z_2 - B^3 - 2B^2X_1Z_2.$$

$$(iv) \quad \text{If } P = Q \text{ then } P + Q = R = (X_3 : Y_3 : Z_3), \text{ with}$$

$$X_3 = 2BD \quad Y_3 = A(4C - D) - 8B^2Y_1^2, \quad Z_3 = 8B^3,$$

where

$$A = aZ_1^2 + 3X_1^2, \quad B = Y_1Z_1, \quad C = BX_1Y_1, \quad D = A^2 - 8C.$$

Denote by \mathbf{M} and \mathbf{S} the cost of a multiplication and a squaring in the field, respectively. Then an addition can be done with $12\mathbf{M} + 2\mathbf{S}$ and a doubling with $7\mathbf{M} + 5\mathbf{S}$. In fact, we can obtain even more efficient arithmetic, if we allow slightly different curves: Montgomery (1987) considered curves of the form

$$(3.5.8) \quad by^2 = x^3 + ax^2 + x$$

and found out that on such kinds of curves scalar multiplication of points on the curve can be done extremely efficiently. In Figure 3.5.5 a plot of such

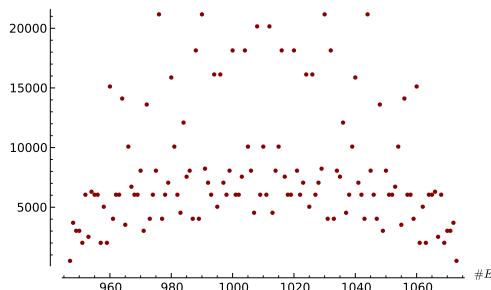


Figure 3.5.5: Real plot of the curve $\frac{1}{43}y^2 = x^3 - \frac{41}{43}x^2 + x$. The reason why we selected this particular curve will be explained in Chapter 4.

a Montgomery curve can be found. Consider a point $P = (x_1, y_2)$ on the curve (3.5.8). In projective coordinates, write $P = (X_1 : Y_1 : Z_1)$ and let $mP = (X_m : Y_m : Z_m)$ and $nP = (X_n : Y_n : Z_n)$ with $m \geq n$. Then the sum $(m+n)P$ is given by

$$\begin{aligned} X_{m+n} &= Z_{m-n}((X_m - Z_m)(X_n + Z_n) + (X_m + Z_m)(X_n - Z_n))^2, \\ Z_{m+n} &= X_{m-n}((X_m - Z_m)(X_n + Z_n) - (X_m + Z_m)(X_n - Z_n))^2, \end{aligned}$$

when $m \neq n$ and in the case $m = n$ by

$$\begin{aligned} X_{2n} &= (X_n + Z_n)^2 \cdot (X_n - Z_n)^2, \\ Z_{2n} &= 4X_n Z_n (X_n - Z_n)^2 + \frac{a+2}{4} \cdot (4X_n Z_n), \end{aligned}$$

where $4X_n Z_n = (X_n + Z_n)^2 - (X_n - Z_n)^2$. Here an addition takes $4\mathbf{M} + 2\mathbf{S}$ and a doubling costs $3\mathbf{M} + 2\mathbf{S}$. Note that the Y -coordinate is completely absent, but it can be reconstructed if necessary (see for example Cohen & Frey 2006, section 13.2.3).

For a thorough list of possible efficient representations of points on elliptic curves, one can consult the Explicit-Formulas Database (see Bernstein & Lange 2011).

3.5.4. Lenstra's Elliptic Curve Method. We are now ready to describe the elliptic curve method, as proposed in Lenstra (1987), for factoring integers n . Similar to Pollard's $(p-1)$ -method (see Pollard 1974). In the algorithm one computes a multiple $Q = kP$ for some reasonably large integer k and a point P on a curve E . Contrasting our definitions before we consider the curve E over \mathbb{Z}_n , an object that is not an elliptic curve when

n is composite (since there are points P and Q on E for which the addition $P + Q$ is not defined). You might call such a curve an *elliptic pseudocurve*. The idea of the algorithm is very simple: We choose randomly a curve with a point on it and compute $Q = kP$ using some addition chain. If we end up on the way of computing Q at a point where we have an undefined addition, this step will yield in many cases a non-trivial factor of n . We describe now the algorithm and give afterwards an analysis of it.

ELLIPTIC CURVE METHOD 3.5.9.

Input: An integer $n \geq 2$ with $\gcd(n, 6) = 1$ and n not a perfect power.

Output: A proper factor of n .

1. Choose a bound B , e.g. $B = 10000$.
2. Repeat 3–5
3. Choose randomly $x, y, a \in \mathbb{Z}_n$.
4. Set $b = (y^2 - x^3 - ax)$ modulo n .
5. Compute $g = \gcd(4a^3 + 27b^2, n)$.
6. While $g = n$.
7. If $g > 1$ then
8. Return g .
9. Set $P = (x, y)$.
10. Try to compute $Q = (B!)P$.
11. If the computation failed then
12. Return a proper factor of n or start from the beginning.
13. Increment B and start from the beginning.

It is striking to observe that this algorithm works differently than any other algorithm we have encountered so far: It runs until we end up with an invalid computation. With this we have hopefully achieved our goal of finding a non-trivial factor of n . Let us have a closer look at step 10: When computing $Q = (B!)P$, we need to employ some addition chain. While going through the chain, we have to compute several slopes m following the Elliptic Curve Addition Law 3.5.1. In each slope computation we have to invert some d modulo n . If d is not coprime to n then the Extended Euclidean Algorithm 3.1.8 (which we employ for computing the inverse) will instead produce a hopefully non-trivial factor of n .

EXAMPLE 3.5.10 (From Washington 2003). Suppose we wish to factor $n = 4453$. Let E be defined by $y^2 = x^3 + 10x - 2$ and let $P = (1, 3)$. Let us try to compute $(3!)P = 6P$. To do so, we compute $2P$, $3P = P + 2P$ and then

$6P = 2 \cdot (3P)$. The slope of the tangent at P is

$$\frac{3x^3 + 10}{2y} = \frac{13}{6} = 3713 \text{ modulo } 4453.$$

Thus, $2P = (4432, 3230)$ by the Elliptic Curve Addition Law 3.5.1. The next step is to obtain $3P = P + 2P$. To compute it, we need to evaluate the slope

$$m = \frac{3230 - 3}{4332 - 1} = \frac{3227}{4331} \text{ modulo } 4453.$$

But $\gcd(4331, 4453) = 61$, so when trying to compute 4331^{-1} modulo 4453, we end up with a non-trivial factor of n . \diamond

Let us try to understand when the Elliptic Curve Method 3.5.9 gives us a proper factor of n . For simplicity of exposition, suppose $n = p \cdot q$, where p and q are prime numbers. In this case we can view an elliptic pseudocurve over \mathbb{Z}_n as two elliptic curves: one modulo p and one modulo q , respectively. By Theorem 3.5.4, we know that over \mathbb{F}_p we have

$$p + 1 - 2\sqrt{p} < \#E < p + 1 + 2\sqrt{p},$$

while over \mathbb{F}_q we have

$$q + 1 - 2\sqrt{q} < \#E < q + 1 + 2\sqrt{q}.$$

Furthermore, we also know by Theorem 3.5.5 that each group order occurs at least once for some elliptic curve. Heuristically, we expect that the density of B -smooth numbers in the Hasse interval modulo p is high enough, and that the distribution of group orders $\#E$ in this interval is sufficiently uniform. Thus, when running the algorithm, we can expect that — for a lucky choice of the curve parameters — the group order of E modulo p is B -smooth, in which case it is very likely that $(B!)P = \mathcal{O}$ modulo p . On the other hand it is very *unlikely* that this group order is also B -smooth modulo q , in which case often $Q = (B!)P \neq \mathcal{O}$ modulo q . Therefore, when we try to compute $(B!)P$ modulo n , we will with high probability find a slope whose denominator is a multiple of p but not a multiple of q , and we will find p using the Extended Euclidean Algorithm 3.1.8.

EXAMPLE 3.5.11 (From Washington 2003). Consider again Example 3.5.10. There for $n = 4453$ the computation of $3P$ for $P = (1, 3)$ yielded the non-trivial factorization $n = 61 \cdot 73$. Let us look at the situation modulo $p = 61$ and $q = 73$ separately:

$$P = (1, 3), \quad 2P = (1, 58), \quad 3P = \mathcal{O}, \quad 4P = (1, 3), \quad \dots \quad (\text{modulo } 61)$$

and

$$P = (1, 3), \quad 2P = (25, 18), \quad 3P = (28, 44), \quad \dots, \quad 64P = \mathcal{O} \pmod{73}.$$

This shows nicely why the computation of the slope m for computing $P + 2P$ was infinite modulo p and finite modulo q , giving the non-trivial factor $p = 61$. \diamond

The benefit of the Elliptic Curve Method 3.5.9 is, when compared to Pollard's $(p - 1)$ -method, that we have now access to a large number of groups we can work in (for each valid parameter choice one group). While in the $p - 1$ method, the groups were given by the input, we can now discard elliptic curves that do not help us in achieving our goal of factoring.

What about the runtime of the Elliptic Curve Method 3.5.9? One can show that it depends mainly in the size of the *smallest* prime factor p of n . More precisely, using the notation of (3.4.2), the heuristic expected runtime of the algorithm is

$$(3.5.12) \quad L(p)^{1+o(1)} = \exp((1 + o(1))\sqrt{\ln p \cdot \ln \ln p}),$$

(see Crandall & Pomerance 2005, section 7.4.1). In the worst case, when n is the product of two roughly equally sized primes, this boils down to a runtime of $L(n)^{1+o(1)}$.

There are many practical tweaks of this algorithm:

1. Use special curves with representations of the points that allow fast arithmetic (see Section 3.5.3 or Chapter 4).
2. Use good (differential) addition chains to compute scalar multiples of point on the curve.
3. Modify the algorithm such that it will also find a nontrivial factor of n when the group order has exactly one prime factor exceeding B (see for example Crandall & Pomerance 2005, section 7.4.2).

Chapter 4

Differential addition on elliptic curves in generalized Edwards form

In this chapter we describe two new parametrizations of points in the spirit of Montgomery (see Section 3.5.3). They allow fast arithmetic on special types of curves, namely elliptic curves in *Edwards form*, proposed in Edwards (2007). The results in this chapter were presented at IWSEC 2010 in Kobe, Japan (see Justus & Loebenberger 2010). Our coauthor found on our suggestion some less efficient differential addition formulas and the formula for recovering the X coordinate. All of the differential formulas presented here as well as the discovery of the second parametrization using squares only are our own findings.

4.1. State of the art

In Table 4.1.1 one finds a selection of the most efficient representations of points on elliptic curves. As in Section 3.5.3, the notation \mathbf{M} and \mathbf{S} refer to a multiplication or a squaring in the field, respectively. We ignore multiplications by a small constant and the additions in the field, since their cost is negligible when compared to the cost of multiplication or squaring.

With the advent of Edwards coordinates in Edwards (2007), extensive work like Bernstein *et al.* (2008a), Bernstein *et al.* (2008b), Bernstein & Lange (2007a), or Bernstein & Lange (2007b), has provided formulas for addition on elliptic curves in Edwards form that are more efficient (by a constant factor) than what is known for other representations. This makes the Edwards form particularly interesting for cryptographic applications.

Forms	Coordinates	Addition	Doubling
Short Weierstraß	$(X : Y : Z) = (X/Z^2, Y/Z^3)$	12M + 4S	4M + 5S
Montgomery form	$(X : Z)$	4M + 1S	2M + 3S
Edwards form	$(X : Y : Z)$	10M + 1S	3M + 4S
Inverted Edwards	$(X : Y : Z) = (Z/X, Z/Y)$	9M + 1S	3M + 4S
<i>Differential Edwards</i> ($c = 1$ and d square)	$(Y : Z)$	4M + 4S	4S
	$(Y^2 : Z^2)$	4M + 2S	4S

Table 4.1.1: Some coordinate choices with fast arithmetic

Castnyck *et al.* (2008) present doubling formulas for elliptic curves in Edwards form with $c = 1$, like the one given in Corollary 4.3.5. They do not consider the case $c \neq 1$ and do not provide a general (differential) addition formula.

Gaudry & Lubicz (2009) present general efficient algorithms for a much broader class of curves. In order to adapt their ideas to the context of elliptic curves in generalized Edwards form, one needs to explicitly express the group law in terms of Riemann’s ϑ functions. Due to our inability to do so, we derive in this work formulas for elliptic curves in generalized Edwards form directly. We are in good company here; Castnyck, Galbraith and Farashahi write: “This is an euphemistic rephrasing of our ignorance about Gaudry and Lubicz’ result, which is somewhat hidden in a different framework.”

Special cases of our result can also be found on the Explicit-Formulas Database (see Bernstein & Lange (2011)): There are several formulas given for $c = 1$ under the assumption that the curve parameter d is a square in the field. The formulas on the Explicit-Formulas Database are on one hand consequences of Gaudry & Lubicz (2009) but can also be deduced from our general formulas in Theorem 4.3.1 and Corollary 4.3.5, as explained at the end of Section 4.3.

In the following, we will give differential addition and doubling formulas for elliptic curves in Edwards curves having arbitrary curve parameters c and d . The restriction $c = 1$ is of less importance in practice, since every curve in *generalized Edwards form* can be transformed into an isomorphic curve with $c = 1$ via the map $(x, y) \mapsto (cx, cy)$. The curve parameter d , on the other hand, is of greater importance: If d is a square in the ground field, the group law, as described in Section 4.2, will not be complete anymore, i.e. the formulas defining the addition on the curve are not valid for all possible input points anymore due to a division by zero.

We will use two parametrizations for elliptic curves in generalized Edwards form to obtain efficient arithmetic: In the first parametrization the

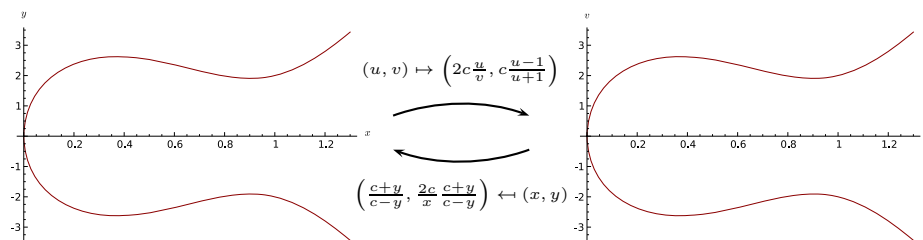


Figure 4.1.1: Transformation of the elliptic curve in Montgomery form $\frac{1}{1-c^4d}v^2 = u^3 + 2\frac{1+c^4d}{1-c^4d}u + u$ to the Edwards form $x^2 + y^2 = c^2(1 + dx^2y^2)$.

projective coordinate $(Y : Z)$ represents a point on the curve. Notice that the X -coordinate is absent, so we cannot distinguish P from $-P$. This is indeed similar to Montgomery’s approach in Montgomery (1987), where he represents a point in Weierstraß-coordinates by omitting the Y -coordinate (see Section 3.5.3). The parametrization used here leads to a differential addition formula, a doubling formula, and a tripling formula on elliptic curves in generalized Edwards form. The addition formula can be computed using $6\mathbf{M} + 4\mathbf{S}$ ($5\mathbf{M} + 4\mathbf{S}$ in the case $c = 1$), the doubling formula using $1\mathbf{M} + 4\mathbf{S}$ ($5\mathbf{S}$ when $c = 1$), and the tripling formula using $4\mathbf{M} + 7\mathbf{S}$. We also provide methods for recovering the missing X -coordinate. Compared to earlier work like Castryck *et al.* (2008), Gaudry & Lubicz (2009), or the formulas on the Explicit-Formulas Database, we explicitly consider all formulas also for the case $c \neq 1$, even though one would typically use in applications curves with $c = 1$.

The second parametrization also omits the X -coordinate. Additionally it uses the squares of the coordinates of the points only. On elliptic curves in generalized Edwards form, addition can be done with $5\mathbf{M} + 2\mathbf{S}$ and point doubling with $5\mathbf{S}$. We also provide a tripling formula for this second representation. For point doubling we get completely rid of multiplications and employ squarings in the ground field only. This is desirable since squarings can be done slightly faster than generic multiplications (see for example Avanzi *et al.* (2006)). This second representation is best suited when employed in a scalar multiplication. Again, we explicitly consider all formulas also for the case $c \neq 1$. Several formulas for this parametrization can be found on the Explicit-Formulas Database, but only for the special case $c = 1$ and d being a square in the ground field. The idea of this representation can already be found in Gaudry & Lubicz (2009), section 6.2.

We will first describe the basics of Edwards coordinates in the following

section and describe the addition and the doubling formula in Section 4.3. The tripling formulas are deduced in Section 4.4. A formula for recovering the X -coordinate is given in Section 4.5. The parametrization of the points that uses the squares of the coordinates only is analyzed in Section 4.6.

4.2. Edwards form

We describe now the basics of elliptic curves in generalized Edwards form. More details can be found, for example, in Bernstein & Lange (2007a) and Bernstein & Lange (2007b). Such curves are given by equations of the form

$$E_{c,d} : x^2 + y^2 = c^2(1 + dx^2y^2),$$

where c, d are curve parameters in a field k of characteristic different from 2. These kind of equations indeed define an elliptic curve in the algebraic-geometric sense (see Section 3.5.1). When $c, d \neq 0$ and $dc^4 \neq 1$, the addition law is defined by

$$(4.2.1) \quad (x_1, y_1), (x_2, y_2) \mapsto \left(\frac{x_1y_2 + y_1x_2}{c(1 + dx_1x_2y_1y_2)}, \frac{y_1y_2 - x_1x_2}{c(1 - dx_1x_2y_1y_2)} \right).$$

For this addition law, the point $(0, c)$ is the neutral element. The inverse of a point $P = (x, y)$ is $-P = (-x, y)$. In particular, $(0, -c)$ has order 2; $(c, 0)$ and $(-c, 0)$ are the points of order 4. When the curve parameter d is not a square in k , then the addition law (4.2.1) is complete (i.e. defined for all inputs). As noted earlier, every curve in generalized Edwards form can be transformed into a curve with $c = 1$ via the map $(x, y) \mapsto (cx, cy)$. For an illustration of the addition law, see Figure 4.2.1.

4.3. Representing points in Edwards form

As explained in the introduction, we represent a point P on the curve $E_{c,d}$ using projective coordinates $P = (Y_1 : Z_1)$. Write $nP = (Y_n : Z_n)$. Then we have

THEOREM 4.3.1 (Justus & Loebenberger 2010). *Let $E_{c,d}$ be an elliptic curve in generalized Edwards form defined over a field k , such that $\text{char}(k) \neq 2$ and $c, d \neq 0$, $dc^4 \neq 1$ and d is not a square in k . Then for $m > n$ we have*

$$\begin{aligned} Y_{m+n} &= Z_{m-n} \left(Y_m^2(Z_n^2 - c^2dY_n^2) + Z_m^2(Y_n^2 - c^2Z_n^2) \right), \\ Z_{m+n} &= Y_{m-n} \left(dY_m^2(Y_n^2 - c^2Z_n^2) + Z_m^2(Z_n^2 - c^2dY_n^2) \right). \end{aligned}$$

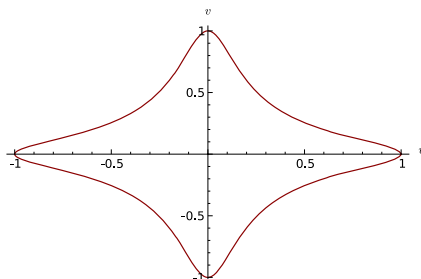


Figure 4.2.1: Real plot of the addition law on the elliptic curve in Edwards form $x^2 + y^2 = c^2(1 + dx^2y^2)$. The addition law is given by hyperbolas, not by simple lines, as in the case of elliptic curves in Weierstraß form.

It can be computed using $6\mathbf{M} + 4\mathbf{S}$. When $n = m$, the doubling formula is given by

$$\begin{aligned} Y_{2n} &= -c^2 d Y_n^4 + 2Y_n^2 Z_n^2 - c^2 Z_n^4, \\ Z_{2n} &= d Y_n^4 - 2c^2 d Y_n^2 Z_n^2 + Z_n^4, \end{aligned}$$

which can be computed using $1\mathbf{M} + 4\mathbf{S}$.

On the Explicit-Formulas Database one finds related formulas for $c = 1$ and d being a square in k . We defer a detailed study of the relationship between the formulas given there and ours at the end of this section.

PROOF. Let $P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2)$ be two different points on the curve $E_{c,d}$. Since the curve parameter d is not a square in k , the addition law (4.2.1) is defined for all inputs. Let $P_1 + P_2 = (x_3, y_3)$ and $P_1 - P_2 = (x_4, y_4)$. Then the addition law (4.2.1) gives

$$\begin{aligned} y_3 c(1 - dx_1 x_2 y_1 y_2) &= y_1 y_2 - x_1 x_2, \\ y_4 c(1 + dx_1 x_2 y_1 y_2) &= y_1 y_2 + x_1 x_2. \end{aligned}$$

After multiplying the two equations above, we obtain

$$(4.3.2) \quad y_3 y_4 c^2 (1 - d^2 x_1^2 x_2^2 y_1^2 y_2^2) = y_1^2 y_2^2 - x_1^2 x_2^2.$$

Next, we substitute $x_1^2 = \frac{c^2 - y_1^2}{1 - c^2 d y_1^2}$ and $x_2^2 = \frac{c^2 - y_2^2}{1 - c^2 d y_2^2}$ (obtained from the curve equation) in (4.3.2), yielding

$$(4.3.3) \quad y_3 y_4 (-d y_1^2 y_2^2 + c^2 d y_1^2 + c^2 d y_2^2 - 1) = c^2 d y_1^2 y_2^2 - y_1^2 - y_2^2 + c^2.$$

After switching to projective coordinates, we see that for $m > n$ the formula for adding $mP = (Y_m, Z_m)$ and $nP = (Y_n, Z_n)$ becomes

$$(4.3.4) \quad \frac{Y_{m+n} Y_{m-n}}{Z_{m+n} Z_{m-n}} = \frac{Y_m^2(Z_n^2 - c^2 d Y_n^2) + Z_m^2(Y_n^2 - c^2 Z_n^2)}{d Y_m^2(Y_n^2 - c^2 Z_n^2) + Z_m^2(Z_n^2 - c^2 d Y_n^2)}.$$

This proves the addition formula. If $P_1 = P_2$, we obtain by the addition law (4.2.1)

$$y_3 c(1 - dx_1^2 y_1^2) = y_1^2 - x_1^2.$$

Similarly, if we substitute $x_1^2 = \frac{c^2 - y_1^2}{1 - c^2 dy_1^2}$ into the equation above, we obtain

$$y_3(cdy_1^4 - 2c^3 dy_1^2 + c) = -c^2 dy_1^4 + 2y_1^2 - c^2.$$

This proves the doubling formula in Theorem 4.3.1 after switching to projective coordinates. \square

We obtain additional savings in the case $c = 1$:

COROLLARY 4.3.5. *Assume the same as in Theorem 4.3.1. If $c = 1$, we have for $m > n$*

$$\begin{aligned} Y_{m+n} &= Z_{m-n} \left((Y_m^2 - Z_m^2)(Z_n^2 - dY_n^2) - (d-1)Y_n^2 Z_m^2 \right), \\ Z_{m+n} &= -Y_{m-n} \left((Y_m^2 - Z_m^2)(Z_n^2 - dY_n^2) + (d-1)Y_m^2 Z_n^2 \right), \end{aligned}$$

which can be computed using $5\mathbf{M} + 4\mathbf{S}$. For doubling we obtain

$$\begin{aligned} Y_{2n} &= -(Y_n^2 - Z_n^2)^2 - (d-1)Y_n^4, \\ Z_{2n} &= (dY_n^2 - Z_n^2)^2 - d(d-1)Y_n^4, \end{aligned}$$

which can be computed using $5\mathbf{S}$. \square

REMARK 4.3.6. *A simple induction argument shows that the computation of the 2^j -fold of a point can be computed using $5j\mathbf{S}$.*

A slight variant of the doubling formula in this Corollary is given by Castryck *et al.* (2008) in their section 3. Similar doubling formulas can be found on the Explicit-Formulas Database, but only for the special case of d being a square in the ground field. For general c the formulas of Theorem 4.3.1 do not seem to be found in the literature.

In the remainder of this section we will explore this relationship in more detail. We focus here in particular on Corollary 4.3.5 since the Explicit-Formulas Database covers the case $c = 1$ only. As on the Explicit-Formulas Database we assume now that $d = r^2$ for some $r \in k$. Then we can write

$$y_{2n} = \frac{-r^2 Y_{2n}^4 + 2Y_{2n}^2 Z_{2n}^2 - Z_{2n}^4}{r^2 Y_{2n}^4 - 2r^2 Y_{2n}^2 Z_{2n}^2 + Z_{2n}^4},$$

where y_{2n} denotes the corresponding affine y -coordinate of the point. Thus we have

$$r y_{2n} = \frac{2r/(r-1) \cdot (r^2 Y_{2n}^4 - 2Y_{2n}^2 Z_{2n}^2 + Z_{2n}^4)}{-2/(r-1) \cdot (r^2 Y_{2n}^4 - 2r^2 Y_{2n}^2 Z_{2n}^2 + Z_{2n}^4)}.$$

If we set $A := \frac{1+r}{1-r}(rY_{2n}^2 - Z_{2n}^2)^2$ and $B := (rY_{2n}^2 + Z_{2n}^2)^2$ we can write the numerator of the last expression as $B - A$ and the denominator as $B + A$, yielding the formulas `db1-2006-g` and `db1-2006-g-2` from the Explicit-Formulas Database. This can be computed with `4S`, but only for those restricted curve parameters. The addition formulas `dadd-2006-g` and `dadd-2006-g-2` from the Explicit-Formulas Database can be deduced in a similar way from our differential addition formula in Corollary 4.3.5.

4.4. A tripling formula

One also obtains a tripling formula that can be computed using `4M + 7S`. This is cheaper than by doing first a doubling and afterwards an addition, which costs `7M + 8S` (`5M + 9S` when $c = 1$).

THEOREM 4.4.1 (Justus & Loebenberger 2010). *Assume the same as in Theorem 4.3.1. Furthermore, let $\text{char}(k) \neq 3$. Then we have*

$$\begin{aligned} Y_{3n} &= Y_n(c^2(3Z_n^4 - dY_n^4)^2 - Z_n^4(8c^2Z_n^4 + (Y_n^2(c^3d + c^{-1}) - 2cZ_n^2)^2 \\ &\quad - c^{-2}(c^4d + 1)^2Y_n^4)), \\ Z_{3n} &= Z_n(c^2(Z_n^4 - 3dY_n^4)^2 + dY_n^4(4c^2Z_n^4 - (Y_n^2(c^3d + c^{-1}) - 2cZ_n^2)^2 \\ &\quad + c^{-2}((c^4d + 1)^2 - 12c^4d)Y_n^4)), \end{aligned}$$

which can be computed using `4M + 7S`.

PROOF. Let $(x_3, y_3) = 3(x, y) = 2(x, y) + (x, y)$. Using the addition law (4.2.1), we obtain an expression for y_3 . Inside the expression, make the

substitution $x^2 = \frac{c^2 - y^2}{1 - c^2 dy^2}$ and simplify to obtain an expression in y only. Then we have

$$y_3 = \frac{y(c^2 d^2 y^8 - 6c^2 dy^4 + 4(c^4 d + 1)y^2 - 3c^2)}{-3c^2 d^2 y^8 + 4d(c^4 d + 1)y^6 - 6c^2 dy^4 + c^2}.$$

Switch to projective coordinates $y = Y/Z$ and rearrange terms. The formula follows. \square

COROLLARY 4.4.2. *Assume the same as in Theorem 4.3.1. Furthermore, let $\text{char}(k) \neq 3$ and assume $c = 1$. Then we have*

$$\begin{aligned} Y_{3n} &= Y_n((dY_n^4 - 3Z_n^4)^2 - Z_n^4((2Z_n^2 - (1+d)Y_n^2)^2 + 8Z_n^4 - (1+d)^2 Y_n^4)), \\ Z_{3n} &= Z_n((Z_n^4 - 3dY_n^4)^2 - dY_n^4((2Z_n^2 - (1+d)Y_n^2)^2 - 4Z_n^4 + (12d - (1+d)^2)Y_n^4)), \end{aligned}$$

which can be computed using $4\mathbf{M} + 7\mathbf{S}$. \square

4.5. Recovering the x -coordinate

In some cryptographic applications it is important to have at some point both coordinates, x - and y . Theorem 4.5.4 shows how to obtain them. There have been results Okeya & Sakurai (2001) and Brier & Joye (2002) in this direction for other forms of elliptic curves. To recover the (affine) x -coordinate, we need the following

LEMMA 4.5.1. *Fix an elliptic curve $E_{c,d}$ in generalized Edwards form, such that $\text{char}(k) \neq 2$ and $c, d \neq 0$, $dc^4 \neq 1$ and d is not a square in k . Let $Q = (x, y)$, $P_1 = (x_1, y_1)$ be two points on $E_{c,d}$. Define $P_2 = (x_2, y_2)$ and $P_3 = (x_3, y_3)$ by $P_2 = P_1 + Q$ and $P_3 = P_1 - Q$. Then we have*

$$(4.5.2) \quad x_1 = \frac{2yy_1 - cy_2 - cy_3}{cdxyy_1(y_3 - y_2)},$$

provided the denominator does not vanish.

PROOF. By the addition law (4.2.1), we have

$$\begin{aligned} c(1 - dx_1yy_1)y_2 &= yy_1 - xx_1, \\ c(1 + dx_1yy_1)y_3 &= yy_1 + xx_1. \end{aligned}$$

Adding the two equations and solving them for x_1 gives the claim. \square

The following lemma provides a simple criterion, which tells us when the denominator in formula (4.5.2) does not vanish.

LEMMA 4.5.3. *Assume the same as in Lemma 4.5.1. Furthermore, let P_1, Q be points whose order does not divide 4. Then the formula (4.5.2) holds.*

PROOF. The points P_1 and Q have orders that are not 1, 2, 4, so $x, x_1, y, y_1 \neq 0$. Suppose now, $y_2 = y_3$ (i.e. y -coordinates of $P_1 + Q$ and $P_1 - Q$ are the same). By the addition law (4.2.1), this implies

$$\frac{yy_1 - xx_1}{c(1 - dxx_1yy_1)} = \frac{yy_1 + xx_1}{c(1 + dxx_1yy_1)}.$$

By solving for d it follows that $dy^2y_1^2 = 1$, which is a contradiction since d is not a square in k . \square

We are now ready to prove

THEOREM 4.5.4 (Justus & Loebenberger 2010). *Let $E_{c,d}$ be an elliptic curve in generalized Edwards form defined over a field k such that $\text{char}(k) \neq 2$, $c, d \neq 0$, $dc^4 \neq 1$, and d is not a square in k . Let $P = (x, y)$ be a point, whose order does not divide 4. Let y_n, y_{n+1} be the affine y -coordinates of the points $nP, (n+1)P$ respectively. Then we have*

$$x_n = \frac{2yy_ny_{n+1} - cC_n - cy_{n+1}^2}{cdxyy_n(C_n - y_{n+1}^2)},$$

where

$$\begin{aligned} A &= 1 - c^2dy^2, \\ B &= y^2 - c^2, \\ C_n &= \frac{Ay_n^2 + B}{dBy_n^2 + A}. \end{aligned}$$

PROOF. Let $nP = (x_n, y_n)$, where P is not a 4-torsion point on $E_{c,d}$. Our task is to recover x_n . By Lemma 4.5.1 with $P_1 = nP$ and $Q = (x, y)$, we may write

$$(4.5.5) \quad x_n = \frac{2yy_n - cy_{n-1} - cy_{n+1}}{cdxyy_n(y_{n-1} - y_{n+1})},$$

where y_{n-1}, y_{n+1} are the y -coordinates of the points $(n-1)P$ and $(n+1)P$ respectively. Now the variable y_{n-1} can be eliminated because of (4.3.4). Indeed, using (4.3.4) we may write in affine coordinates

$$(4.5.6) \quad y_{n-1}y_{n+1} = \frac{Ay_n^2 + B}{dBy_n^2 + A},$$

where

$$A = 1 - c^2dy^2, \quad B = y^2 - c^2.$$

Now from (4.5.6), y_{n-1} can be isolated and put back in (4.5.5). This gives

$$x_n = \frac{2yy_ny_{n+1}(dBy_n^2 + A) - c(Ay_n^2 + B) - cy_{n+1}^2(dBy_n^2 + A)}{cdxyy_n(Ay_n^2 + B - y_{n+1}^2(dBy_n^2 + A))}.$$

The claim follows. \square

4.6. A parametrization using squares only

The formulas in Theorem 4.3.1 show that for the computation of Y_{m+n}^2 , and Z_{m+n}^2 it is sufficient to know the squares of the coordinates of the points $(Y_m : Z_m)$, $(Y_n : Z_n)$ and $(Y_{m-n} : Z_{m-n})$ only. This gives

THEOREM 4.6.1 (Justus & Loebenger 2010). *Assume the same as in Theorem 4.3.1. Then, for $m > n$ we have*

$$\begin{aligned} Y_{m+n}^2 &= Z_{m-n}^2 ((A+B)/2)^2, \\ Z_{m+n}^2 &= Y_{m-n}^2 \left((A-B)/2 + (d-1)Y_m^2(Y_n^2 - c^2Z_n^2) \right)^2, \end{aligned}$$

with

$$\begin{aligned} A &:= (Y_m^2 + Z_m^2)((1 - dc^2)Y_n^2 + (1 - c^2)Z_n^2), \\ B &:= (Y_m^2 - Z_m^2)((1 + c^2)Z_n^2 - (1 + dc^2)Y_n^2). \end{aligned}$$

This addition can be computed using $5\mathbf{M} + 2\mathbf{S}$, if one stores the squares of the coordinates only. When $n = m$, we obtain

$$\begin{aligned} Y_{2n}^2 &= \left((1 - c^2d)Y_n^4 + (1 - c^2)Z_n^4 - (Y_n^2 - Z_n^2)^2 \right)^2, \\ Z_{2n}^2 &= \left(dc^2(Y_n^2 - Z_n^2)^2 - d(c^2 - 1)Y_n^4 + (c^2d - 1)Z_n^4 \right)^2, \end{aligned}$$

which can be computed using $5\mathbf{S}$, if one stores the squares of the coordinates only.

PROOF. This follows directly from Theorem 4.3.1 and elementary calculus. \square

A direct adaptation of Corollary 4.3.5 does not give any speedup. Again on the Explicit-Formulas Database one finds related formulas for $c = 1$ and d being a square in k .

We will now sketch the computation of a scalar multiple sP in this parametrization. Assume P has affine coordinates $(x : y)$. Then one would proceed as follows: After changing to projective coordinates $(X : Y : Z)$, two squares (one for each of the coordinates Y and Z) have to be computed. Now, a differential addition chain is employed to compute the multiple sP . During all but the last step of the computation we store the squares of the coordinates of the intermediate points only. The last step plays a special role now, since at the end we wish to obtain the coordinates of the point sP and not the square of the coordinates. To do so, we run the last step using the first parametrization. If we construct from the beginning the differential addition chain, such that for each computation of P_{m+n} we have that $m - n = 1$, we obtain an efficient algorithm for computing the scalar multiple sP on an elliptic curve in generalized Edwards form using the second parametrization. In order to recover the x -coordinate, one would have to compute also the scalar multiple $(s + 1)P$ and use the recovering formula from Theorem 4.5.4.

Also the tripling formula given in Theorem 4.4.1 can be adapted to this second parametrization. Namely, we have

COROLLARY 4.6.2. *Assume the same as in Theorem 4.3.1. Furthermore, we assume $\text{char}(k) \neq 3$. Then we have*

$$Y_{3n}^2 = Y_n^2(c^2(3Z_n^4 - dY_n^4)^2 - Z_n^4(8c^2Z_n^4 + (Y_n^2(c^3d + c^{-1}) - 2cZ_n^2)^2 - c^{-2}(c^4d + 1)^2Y_n^4))^2,$$

$$Z_{3n}^2 = Z_n^2(c^2(Z_n^4 - 3dY_n^4)^2 + dY_n^4(4c^2Z_n^4 - (Y_n^2(c^3d + c^{-1}) - 2cZ_n^2)^2 + c^{-2}((c^4d + 1)^2 - 12c^4d)Y_n^4))^2,$$

which can be computed using $4\mathbf{M} + 7\mathbf{S}$, if one stores the squares of the coordinates only. \square

Chapter 5

Public key cryptography

We will now turn our attention to some of the *cryptographic* applications of the concepts described in the previous chapters. Naturally, these concepts help on one hand to *build* cryptographic systems and on the other hand have wide applications in *cryptanalysis*. After having described the most basic crypto systems in the following section, we will show that in almost all of them naturally integers turn up, whose prime factors are not too small and not too large – we will call such integers *grained*.

5.1. Diffie and Hellman: New directions in cryptography

In Diffie & Hellman (1976), a groundbreaking article on a new concept for cryptography was published: Instead of requiring that the two communicating parties have to agree on a common secret in advance (as was common practice at that point in time), they proposed to have two keys: a public key for encryption, made available on, say, a key-server and a private key for decryption, only known to the recipient of the encrypted message. In this setting, it should be infeasible to compute the private key from the publicly available information. The advantage of this approach is obvious: We get completely rid of a secure channel that allows transmission of a pre-shared secret! The same technique also allows to create a digital analog to written signatures: The signature on a document is created using the private key, while the validity of the signature is checked using the corresponding public key.

Diffie & Hellman did not describe an actual realization of their idea, but they proposed a key exchange protocol, known as the Diffie-Hellman key exchange. If two users, Alice and Bob, wish to agree on a common secret, they first decide on a cyclic group to work in (say the multiplicative group

of a finite field \mathbb{F}_q with a generator g). Alice selects randomly an element a from \mathbb{F}_q^\times and transmits $A = g^a$ to Bob. Bob himself selects randomly an element b from \mathbb{F}_q^\times and transmits $B = g^b$ to Alice. After having exchanged these values, Alice computes $B^a = g^{ab}$, while Bob computes $A^b = g^{ab}$, the common secret. Why is this secure? Diffie & Hellman realized that in a finite field \mathbb{F}_q it is easy to compute the value $b = a^e$ (for example using the Fast Exponentiation Algorithm 3.1.11) for a given element $a \in \mathbb{F}_q$ and an integer exponent e . On the other hand, it seems difficult to solve the Discrete Logarithm Problem 3.4.6, i.e. to find e when given just a and b . The best algorithm known nowadays is the index-calculus method (see Section 3.4.4), which has sub-exponential complexity in the size of the inputs. The problem we still face is that we cannot *prove* that discrete logarithms cannot be computed efficiently. Indeed, it would directly imply $\mathcal{P} \neq \mathcal{NP}$, another millennium problem of the Clay Mathematics Institute (2000). There is, however, yet another issue: One does not necessarily need to compute discrete logarithms to destroy the security of the Diffie-Hellman key exchange! If one manages to compute g^{ab} from the knowledge of g , g^a and g^b only (this is the so-called *computational Diffie-Hellman problem*), an attacker could deduce the common secret by observing public information only! For more details on the relation of these different problems, see for example Galbraith (2011, chapter 21).

5.2. Doing it: RSA

The just described ideas were for the first time realized in Rivest, Shamir & Adleman (1978). Contrasting the Diffie & Hellman (1976) article, the security of this system did not depend on the difficulty of the Discrete Logarithm Problem 3.4.6, but on the Factorization Problem 2.1.3. The *RSA system* can be described as follows: A public key consists of an integer $n = pq$, where p and q are prime and a public encryption exponent $e \in \mathbb{Z}_{\varphi(n)}^\times$, where $\varphi(n)$ is the Euler φ -function (see Definition 2.1.10). The corresponding private key contains the integer n and a private *decryption* exponent $d \in \mathbb{Z}_{\varphi(n)}^\times$ with the property that $d = e^{-1}$ modulo $\varphi(n)$. A message $m \in \mathbb{Z}_n$ can now be encrypted by computing $c = m^e$ modulo n , and the receiver can recover the message $c^d = m^{ed} = m$ modulo n .

There are many interesting questions related to this system: First of all, the security is only given if it is infeasible to recover the private key (n, d) from the public information (that includes the public key (n, e)). This, in turn, boils down to the problem of computing $d = e^{-1}$ modulo $\varphi(n)$, a task that is very simple to achieve using the Extended Euclidean Algorithm 3.1.8,

if we know $\varphi(n) = (p-1)(q-1)$. Since computing this quantity is polynomial time equivalent to factoring n , it is necessary for security of RSA to assume that factoring a product of two primes is hard. It is clear that this assumption is not sufficient for the security of RSA. Indeed, it might be possible to recover the message m from the public key (n, e) and the encrypted message $c = m^e$ only (this is the so called *RSA problem*). For more details on the relation of these different problems see Galbraith (2011, chapter 24).

A completely different aspect of the security of the system is the *concrete* choice of how to generate the public as well as the private key. Indeed, just the selection of an appropriate integer $n = pq$ motivates several questions on the security of RSA (see for example Chapter 8).

5.3. The ubiquity of grained integers

We have seen that in the RSA crypto system, described in the last section, one constructs a modulus n that is a product of two roughly equally sized primes. This already implies that the integer n cannot be factored efficiently using trial division, i.e. the prime factors p and q are larger than some poly-logarithmic bound B . But this requirement gives also an *upper bound* on the size of p and q , since both p and q are by construction also smaller than $C = n/B$. Thus any integer $n = pq$ used in the RSA crypto system has prime factors within interval $[B, C]$, i.e. they are simultaneously B -rough and C -smooth. In real-world implementations of RSA the bound B is not only poly-logarithmic in n , but actually very close to \sqrt{n} . This has, by the same reasoning as above, the consequence that also the bound C is very close to \sqrt{n} .

Integers with prime factors from a certain interval $[B, C]$ also pop up in factorization algorithms: If we consider the Quadratic Sieve (see Section 3.4.1) or the General Number Field Sieve (see Section 3.4.3) and use the Elliptic Curve Method 3.5.9 (ECM) for the factorization of the intermediate sieving results, we will typically first try to factor those sieving results via trial division up to some bound B . If this does not succeed, i.e. the number we are currently trying to factor is B -rough, then we start the Elliptic Curve Method 3.5.9 on this integers and run it until we think that we may not succeed (that is after, say, one second of computation time, we abort and discard the number). Now, since the runtime of the ECM basically relies on the size of the smallest prime factor of the input only (cf. the runtime estimate (3.5.12)), such a procedure would naturally be able to factor such numbers that are C -smooth, where the bound C depends

on the runtime-bound we are imposing on our algorithm. Thus, also there the surviving numbers (i.e. those which actually *can* be factored using the trial-division/ECM approach) will with high probability be simultaneously B -rough and C -smooth.

These two examples show that such *grained* integers often occur naturally in the design and the analysis of cryptographic algorithms (especially those whose security depends on Factorization Problem 2.1.3), and it turns out that we can use this additional property to say something more about the behavior of the algorithms using them and answer some very practical questions concerning them (see Chapter 7 and Chapter 10).

Chapter 6

Coarse-grained integers

The results in this section arose from a fruitful collaboration with Michael Nüsken between 2007 and 2011. A preprint of the results was published on the math arXiv, see Loebenberger & Nüsken (2010). It is a bit difficult to actually tell which results are the coauthors' work only. We were often working together on certain aspects of the analysis. The results that were mainly our coauthor's work are marked with the citation "Nüsken (2006-2011)".

Let us call an integer \mathcal{A} -grained if and only if all its prime factors are in the set \mathcal{A} . If \mathcal{A} is an interval $[B, C]$, we call the integer $[B, C]$ -grained. Then an integer is C -smooth if and only if it is $]0, C]$ -grained, and B -rough if and only if it is $]B, \infty[$ -grained. You may want to call an integer family *coarse-grained* if it is $]B, C]$ -grained and the number of factors is bounded (alternatively, we then call such an integer $[B, C]$ -grained). We consider the number of integers up to a real positive bound x that are $]B, C]$ -grained and have a given number k of factors, in formulae:

$$(6.0.1) \quad \pi_{B,C}^k(x) := \#\left\{n \leq x \mid \exists p_1, \dots, p_k \in \mathbb{P} \cap]B, C]: n = p_1 \cdots p_k\right\}.$$

We always assume that $C > B$, since otherwise the set under consideration is empty.

Such numbers occur for example in an intermediate step in the general number field sieve when trying to factor large numbers. During the sieving integers are constructed as random values of carefully chosen small-degree polynomials and made B -rough by dividing out all smaller factors. The remaining number is fed into the Elliptic Curve Method 3.5.9 with a time limit that should allow to find factors up to C . To tune the overall algorithm it is vital to know the probability that the remaining number is C -smooth. As-

suming heuristically that the polynomials output truly random numbers on random inputs the counting task we deal with is the missing link, since estimates for counting B -smooth numbers are known, see Section 2.3.2. There is plenty of work on smooth integers. A brilliant overview article on the state of the art in this area is Granville (2008). Rough numbers on the other hand have drawn much less attention. The only result we are aware of is the very old article Бухштраб (1937). The combined question of considering $[B, C]$ -grained integers seems not have been studied anywhere in the literature. This might be due to the fact that the counting problem is trivial for fixed bounds on B, C since the number of coarse-grained integers is in this case finite, see below.

To avoid difficulties with non-squarefree numbers, instead of numbers we count lists of primes

$$(6.0.2) \quad \kappa_{B,C}^k(x) := \#\left\{(p_1, \dots, p_k) \in (\mathbb{P} \cap]B, C])^k \mid p_1 \cdots p_k \leq x\right\}.$$

It turns out that there are anyways only a few non-squarefree numbers counted by $\pi_{B,C}^k$, namely $k! \cdot \pi_{B,C}^k(x) \approx \kappa_{B,C}^k(x)$. This would actually be an equality if there were no non-squarefree numbers in the count. We defer a precise treatment until Section 6.8.

We head for determining precise bounds $\kappa_{B,C}^k(x)$ or $\pi_{B,C}^k(x)$ that can be used in practical situations. However, to understand these bounds we additionally consider the asymptotical behaviors. This is tricky since we have to deal with the three parameters B, C and x simultaneously. To guide us in considering different asymptotics, we usually write $B = x^\beta$, $C = x^\gamma$ and $\gamma = \beta(1 + \alpha)$. In particular, $C = B^{1+\alpha}$. So we replace (B, C, x) with (x, β, γ) or (x, α, β) , similar to the considerations when counting smooth or rough numbers in the literature. Alternatively, it seems also natural to fix x somehow in the interval $]B^k, C^k]$ by introducing a parameter ξ by

$$x = B^{k-\xi}C^\xi = B^{k+\xi\alpha}.$$

Now the parameters are (B, C, ξ) or (B, α, ξ) .

As a first observation, note that $\kappa_{B,C}^k(x)$ is constantly 0 for $x < B^k$ and constantly $(\pi(C) - \pi(B))^k$ for $x \geq C^k$ and grows monotonically when x goes through $]B^k, C^k]$. Here π denotes the prime counting function, see Chapter 2. For ‘middle’ x -values the asymptotics can be derived from our main result:

COROLLARY 6.0.3. Fix $k \in \mathbb{N}_{\geq 2}$, $\alpha > 0$ and $\varepsilon > 0$. Then for large $B = x^\beta$ and $C = B^{1+\alpha}$ we have uniformly for $x \in [B^k(1+\varepsilon), C^k(1-\varepsilon)]$ that

$$\kappa_{B,C}^k(x) \in \Theta\left(\frac{x}{\ln B}\right) = \Theta\left(\frac{x}{\beta \ln x}\right),$$

where the error between the approximation and the function is of order $\mathcal{O}\left(\frac{x}{\sqrt{B}}\right)$ and the hidden constants depend on ε, k and α only. \square

This is in contrast to the asymptotics at $x = C^k$:

$$\kappa_{B,C}^k(C^k) \approx \frac{C^k}{\ln^k C} = \frac{B^{k(1+\alpha)}}{(1+\alpha)^k \ln^k B} \in \Theta\left(\frac{x}{\beta^k \ln^k x}\right).$$

This observation is explained as follows: Note that roughly half of the numbers up to x are in the interval $[\frac{1}{2}x, x]$ and similarly for primes. Thus the behavior of those candidates largely rule $\kappa_{B,C}^k(x)/x$. For an $x = B^{k-\xi}C^\xi$ with a fixed $\xi \in]0, k[$, it is mostly determined by the requirement that the counted numbers are B -rough, and we thus observe a comparatively large fraction of $]B, C]$ -grained numbers. In the extreme case $x = C^k$, most candidates are ruled out by the requirement to be C -smooth and thus we see a much smaller fraction of $]B, C]$ -grained numbers.

The case that the intervals $]B^k, C^k]$ are disjoint for considered values k is especially nice, as then the number of prime factors of a B -rough and C -smooth number n can be derived from the number n . So we assume in the entire paper that $C < B^s$ for some fixed $s > 1$. (Clearly, we cannot have any fixed s that grants disjointness for all intervals. But for the first few we can.) In the inspiring number field sieve application we have $C < B^{1.2327}$. This ensures that the intervals $]B^k, C^k]$ for $k \leq 5$ are disjoint.

A further application is related to RSA. Decker & Moree (2008) give estimates for the number of RSA-integers. However, there are many possible ways of constructing RSA-integers. A discussion and further calculations to adapt our results to the different shape are needed. We treat these issues in Chapter 8.

As our basic field of interest is cryptography and there the largest occurring numbers are actually small in the number theorist's view, we assume the Riemann Hypothesis 2.2.14 throughout the entire paper. We will always use the Prime Number Theorem 2.1.8(iii), namely the explicit

PRIME NUMBER THEOREM (Von Koch 1901, Schoenfeld 1976). *If (and only if) the Riemann Hypothesis 2.2.14 holds then for $x \geq 1451$*

$$|\pi(x) - \text{Li}(x)| < \frac{1}{8\pi} \sqrt{x} \ln x,$$

where $\text{Li}(x) = \int_2^x \frac{1}{\ln t} dt$.

We have numerically verified this inequality for $x \leq 2^{40} \approx 1.1 \cdot 10^{12}$ based on Kulsha's tables (Kulsha 2008) and extensions built using the segmented siever implemented by Oliveira e Silva (2003). We are confident that we can extend this verification much further. In the inspiring application we have $x < 2^{37}$ and so for those x we can take this theorem for granted even if the Riemann Hypothesis 2.2.14 should not hold.

We arrive at the following description of the desired count:

THEOREM. *Let $B < C = B^{1+\alpha}$ with $\alpha \geq \frac{\ln B}{\sqrt{B}}$ and fix $k \geq 2$. Then for any (small) $\varepsilon > 0$ and B tending to infinity we have for $x \in [B^k(1+\varepsilon), C^k(1-\varepsilon)]$ a value $\tilde{c} \in \left[\frac{\alpha^{k-1} \delta_k}{k!(1+\alpha)^k}, \frac{1}{k!} \right]$ with $\delta_k = \min \left(2^{-4\varepsilon^k}, 2^{-k} \frac{\varepsilon^{k-1}}{(k-1)!} \right)$ such that*

$$\left| \pi_{B,C}^k(x) - \tilde{c} \frac{x}{\ln B} \right| \leq (2^k - 1) \alpha^{k-2} (1 + \alpha) \cdot \frac{x}{\sqrt{B}} + 2^{k-1} \frac{x}{B}.$$

Also without assuming the Riemann Hypothesis 2.2.14 we can achieve meaningful results provided we use a good unconditional version of the prime number theorem whose error estimate is at least in $\mathcal{O} \left(\frac{x}{\ln^3 x} \right)$. The famous work by Rosser & Schoenfeld (1962); Rosser & Schoenfeld (1975) is not sufficient. Yet, Dusart (1998) provides an explicit error bound of order $\mathcal{O} \left(\frac{x}{\ln^3 x} \right)$, and Ford (2002a) provides explicit error bounds of order $\mathcal{O} \left(x \exp \left(-\frac{A(\ln x)^{3/5}}{(\ln \ln x)^{1/5}} \right) \right)$ though this only applies for x beyond 10^{171} or even much later depending on A and the \mathcal{O} -constant, see Fact 6.6.1.

6.1. The recursion

The essential basis for the analysis of the counting functions $\kappa_{B,C}^k$ is the following simple description.

LEMMA 6.1.1. *For all $k \in \mathbb{N}_{>0}$ we have the recursion*

$$\kappa_{B,C}^k(x) = \sum_{p_k \in \mathbb{P} \cap]B, C]} \kappa_{B,C}^{k-1}(x/p_k)$$

based on

$$\kappa_{B,C}^0(x) = \begin{cases} 0 & \text{if } x \in [0, 1[, \\ 1 & \text{if } x \in [1, \infty[. \end{cases}$$

PROOF. In case $k > 0$ we have

$$\begin{aligned} \kappa_{B,C}^k(x) &= \# \left\{ (p_1, \dots, p_k) \in (\mathbb{P} \cap]B, C])^k \mid p_1 \cdots p_k \leq x \right\} \\ &= \# \bigcup_{p \in \mathbb{P} \cap]B, C]} \left\{ (p_1, \dots, p_k) \in (\mathbb{P} \cap]B, C])^k \mid \begin{array}{l} p_1 \cdots p_{k-1} \leq x/p_k, \\ p_k = p \end{array} \right\} \\ &= \sum_{p_k \in \mathbb{P} \cap]B, C]} \kappa_{B,C}^{k-1}(x/p_k). \end{aligned}$$

The case $k = 0$ is immediate from the definition. \square

From the definition (6.0.2) or from Lemma 6.1.1, it is clear that

$$\kappa_{B,C}^1(x) = \begin{cases} 0 & \text{if } x \in [0, B[, \\ \pi(x) - \pi(B) & \text{if } x \in [B, C[, \\ \pi(C) - \pi(B) & \text{if } x \in [C, \infty[. \end{cases}$$

This reveals that the case distinction in $\kappa_{B,C}^0$ leaves its traces on higher $\kappa_{B,C}^k$. For further calculations it is vital that we make this precise. This will enable us later to do our estimates. For $k \in \mathbb{N}$ we distinguish $k + 2$ cases:

$$\begin{aligned} x \text{ is in case } (k, -1) &: \iff x \in [0, B^k[, \\ x \text{ is in case } (k, j) &: \iff x \in [B^{k-j}C^j, B^{k-1-j}C^{j+1}[, \\ x \text{ is in case } (k, k) &: \iff x \in [C^k, \infty[, \end{aligned}$$

where $j \in \mathbb{N}_{<k}$. Note that most cases are characterized by the exponent of C at the left end of the interval.

LEMMA 6.1.2. For $k, j \in \mathbb{N}$, $0 \leq j < k$ and x in case (k, j) , we have

$$\kappa_{B,C}^k(x) = \sum_{p_k \in \mathbb{P} \cap \left] \frac{x}{B^{k-1-j}C^j}, C \right]} \kappa_{B,C}^{k-1}(x/p_k) + \sum_{p_k \in \mathbb{P} \cap \left] B, \frac{x}{B^{k-1-j}C^j} \right]} \kappa_{B,C}^{k-1}(x/p_k)$$

where in the first sum x/p_k is in case $(k-1, j-1)$ and in the second in case $(k-1, j)$. For $j = -1$, and $j = k$ we do not split the sum, as then all x/p_k are in one case anyways. For $j = 0$ the left part is zero, so that the splitting there is less visible.

PROOF. We only have to verify that $x/p_k \in [B^{k-1-j}C^j, B^{k-j-2}C^{j+1}[$ for $p_k \in \mathbb{P} \cap]B, x/B^{k-1-j}C^j]$ and $x \in [B^{k-j}C^j, B^{k-1-j}C^{j+1}[$. Similarly, the statement for the second sum is established. \square

For example, we obtain

$$(6.1.3) \quad \kappa_{B,C}^2(x) = \begin{cases} 0 & \text{if } x \in [0, B^2[, \\ \sum_{p_2 \in \mathbb{P} \cap]B, \frac{x}{B}] \sum_{p_1 \in \mathbb{P} \cap]B, \frac{x}{p_2}] 1 & \text{if } x \in [B^2, BC[, \\ \sum_{p_2 \in \mathbb{P} \cap]\frac{x}{C}, C] \sum_{p_1 \in \mathbb{P} \cap]B, \frac{x}{p_2}] 1 & \text{if } x \in [BC, C^2[, \\ + \sum_{p_2 \in \mathbb{P} \cap]B, \frac{x}{C}] \sum_{p_1 \in \mathbb{P} \cap]B, C] 1 & \\ \sum_{p_2 \in \mathbb{P} \cap]B, C] \sum_{p_1 \in \mathbb{P} \cap]B, C] 1 & \text{if } x \in [C^2, \infty[. \end{cases}$$

So for $\kappa_{B,C}^2(x)$ we have four cases with four 2-fold sums. In general $\kappa_{B,C}^k(x)$ has $k+2$ cases with 2^k k -fold sums. Well, we better stop unfolding here.

Based on the intuition that a randomly selected integer n is prime with probability $\frac{1}{\ln n}$ we can replace $\sum_{p \in \mathbb{P} \cap]B, C] f(p)$ with $\int_B^C \frac{f(p)}{\ln p} dp$. This directly leads to the approximation function. We prefer however to follow a better founded way to them which will also give information about the error term.

6.2. Using estimates

From the recursion for $\kappa_{B,C}^k(x)$ it is clear that we have to compute terms like

$$\sum_{p \in \mathbb{P} \cap]B, C] f(p) \quad \text{or} \quad \sum_{p \in \mathbb{P} \cap]B, C] f(x/p).$$

To get good estimates for such a sum we follow the classic path, as Rosser & Schoenfeld (1962): we rewrite the sum as a Lebesgue-Stieltjes-integral over the prime counting function $\pi(p)$. Then we substitute $\pi(t) = \text{Li}(t) + E(t)$, keeping in mind that we know good bounds on the error term $E(t)$ by the Prime Number Theorem 2.1.8(iii). Finally, we integrate by parts, estimate

and integrate by parts back:

$$\begin{aligned}
& \sum_{p \in \mathbb{P} \cap]B, C]} f(p) \\
&= \int_B^C f(t) d\pi(t) \\
&= \int_B^C f(t) d\text{Li}(t) + \int_B^C f(t) dE(t) \\
&= \int_B^C \frac{f(t)}{\ln t} dt + f(C)E(C) - f(B)E(B) - \int_B^C E(t) df(t).
\end{aligned}$$

The existence of all integrals follow from the existence of the first. If the sum kernel f is differentiable with respect to t we can rewrite $\int_B^C E(t) df(t) = \int_B^C f'(t)E(t) dt$. Now we can use the estimate on the error term $E(t)$:

$$\begin{aligned}
& \left| \sum_{p \in \mathbb{P} \cap]B, C]} f(p) - \int_B^C \frac{f(t)}{\ln t} dt \right| \\
& \leq |f(C)|\widehat{E}(C) + |f(B)|\widehat{E}(B) + \int_B^C |f'(t)|\widehat{E}(t) dt.
\end{aligned}$$

What remains is, given the concrete f , to determine the occurring integrals. For the counting functions $\kappa_{B,C}^k$ — as one would guess — this task is more and more complicated the larger k is. Clearly, smoothness properties of f must be considered carefully.

During all this we make sure that the involved functions stay sufficiently smooth:

LEMMA 6.2.1 (Prime sum approximation). *Let $f, \widetilde{f}, \widehat{f}$ be functions $\mathbb{R}_{>0} \rightarrow \mathbb{R}_{\geq 0}$ such that \widetilde{f} and \widehat{f} are piece-wise continuous, $f + \widehat{f}$ is increasing, and*

$$|f(x) - \widetilde{f}(x)| \leq \widehat{f}(x)$$

for $x \in \mathbb{R}_{>0}$. Further, let $\widehat{E}(p)$ be a positive valued, increasing, smooth function of p bounding $|\pi(p) - \text{Li}(p)|$ on $[B, C]$. (For example, under the Riemann Hypothesis 2.2.14 we can take $\widehat{E}(p) = \frac{1}{8\pi} \sqrt{p} \ln p$ provided $p \geq 1451$.) Then for $x \in \mathbb{R}_{>0}$

$$\left| \sum_{p \in \mathbb{P} \cap]B, C]} f(x/p) - \widetilde{g}(x) \right| \leq \widehat{g}(x)$$

where

$$\begin{aligned}\tilde{g}(x) &= \int_B^C \frac{\tilde{f}(x/p)}{\ln p} dp, \\ \hat{g}(x) &= \int_B^C \frac{\hat{f}(x/p)}{\ln p} dp + 2(\tilde{f} + \hat{f})(x/B)\hat{E}(B) + \int_B^C (\tilde{f} + \hat{f})(x/p)\hat{E}'(p) dp.\end{aligned}$$

Moreover, \tilde{g} and \hat{g} are piece-wise continuous, and $\tilde{g} + \hat{g}$ is increasing.

PROOF. The assumption immediately implies

$$\left| \sum_{p \in \mathbb{P} \cap]B, C]} f(x/p) - \sum_{p \in \mathbb{P} \cap]B, C]} \tilde{f}(x/p) \right| \leq \sum_{p \in \mathbb{P} \cap]B, C]} \hat{f}(x/p).$$

Using the techniques just sketched we obtain $\sum_{p \in \mathbb{P} \cap]B, C]} \tilde{f}(x/p) = \int_B^C \frac{\tilde{f}(x/p)}{\ln p} dp + \int_B^C \tilde{f}(x/p) dE(p)$ with $E(p) = \pi(p) - \text{Li}(p)$. Shifting the second term to the correspondingly transformed error bound, we now have

$$\left| \sum_{p \in \mathbb{P} \cap]B, C]} f(x/p) - \underbrace{\int_B^C \frac{\tilde{f}(x/p)}{\ln p} dp}_{=\tilde{g}(x)} \right| \leq \int_B^C \frac{\hat{f}(x/p)}{\ln p} dp + \int_B^C (\hat{f} + \tilde{f})(x/p) dE(p).$$

This bound always holds, yet we still have to estimate $E(p)$ in it. Abbreviating $h(p) = (\hat{f} + \tilde{f})(x/p)$ we estimate the last integral:

$$\begin{aligned}\int_B^C h(p) dE(p) &= h(C)E(C) - h(B)E(B) - \int_B^C E(p) dh(p) \\ &\leq h(C)\hat{E}(C) + h(B)\hat{E}(B) - \int_B^C \hat{E}(p) dh(p) \\ &= +h(C)\hat{E}(C) + h(B)\hat{E}(B) \\ &\quad - h(C)\hat{E}(C) + h(B)\hat{E}(B) \\ &\quad + \int_B^C h(p) d\hat{E}(p).\end{aligned}$$

For the inequality we use that $h(p)$ is *decreasing* in p . Collecting gives the claim.

Finally, \tilde{g} and \hat{g} being obviously piece-wise continuous it remains to show that $\tilde{g} + \hat{g}$ is increasing. By the (defining) equation

$$(\tilde{g} + \hat{g})(x) = \int_B^C (\tilde{f} + \hat{f})(x/p) \left(\frac{1}{\ln p} + \hat{E}'(p) \right) dp + 2(\tilde{f} + \hat{f})(x/B)\hat{E}(B)$$

this follows from $\tilde{f} + \hat{f}$ and \hat{E} being increasing. \square

6.3. Approximations

Based on Lemma 6.2.1 we recursively define approximation functions $\tilde{\kappa}_{B,C}^k$ and error bounding functions $\hat{\kappa}_{B,C}^k$. Recursive application of Lemma 6.2.1 will lead to a good estimate in Theorem 6.3.2. For the understanding we further need to determine the asymptotic order of the functions defined here, which we start in the remainder of this section. Theorem 6.3.5 relates the functions $\tilde{\kappa}_{B,C}^k$ and $\hat{\kappa}_{B,C}^k$ to some easier manageable functions. These in turn are computed or estimated, respectively, in Section 6.4 and Section 6.5.

DEFINITION 6.3.1. For $x \geq 0$ we define

$$\tilde{\kappa}_{B,C}^0(x) := \kappa_{B,C}^0(x), \quad \hat{\kappa}_{B,C}^0(x) := 0$$

and recursively for $k > 0$

$$\begin{aligned} \tilde{\kappa}_{B,C}^k(x) &:= \int_B^C \frac{\tilde{\kappa}_{B,C}^{k-1}(x/p_k)}{\ln p_k} dp_k, \\ \hat{\kappa}_{B,C}^k(x) &:= \int_B^C \frac{\hat{\kappa}_{B,C}^{k-1}(x/p_k)}{\ln p_k} dp_k \\ &\quad + 2 \left(\tilde{\kappa}_{B,C}^{k-1} + \hat{\kappa}_{B,C}^{k-1} \right) (x/B) \hat{E}(B) \\ &\quad + \int_B^C \left(\tilde{\kappa}_{B,C}^{k-1} + \hat{\kappa}_{B,C}^{k-1} \right) (x/p_k) \hat{E}'(p_k) dp_k. \end{aligned}$$

These functions now describe the behavior of $\kappa_{B,C}^k$ nicely:

THEOREM 6.3.2. Given $x \in \mathbb{R}_{>0}$ and $k \in \mathbb{N}$. Then the inequality

$$\left| \kappa_{B,C}^k(x) - \tilde{\kappa}_{B,C}^k(x) \right| \leq \hat{\kappa}_{B,C}^k(x)$$

holds.

PROOF. Using Lemma 6.2.1 the claim together with the fact that $\tilde{\kappa}_{B,C}^k + \hat{\kappa}_{B,C}^k$ is increasing follow simultaneously by induction on k based on $\tilde{\kappa}_{B,C}^0 = \kappa_{B,C}^0$ and $\hat{\kappa}_{B,C}^0 = 0$. \square

In order to give a first impression we calculate $\tilde{\kappa}_{B,C}^1$ and $\hat{\kappa}_{B,C}^1$. Analogous to Lemma 6.1.2, we split the integration at $x/B^{k-1-j}C^j$ so that the parts fall entirely into case $(k-1, j-1)$ or into case $(k-1, j)$:

$$\tilde{\kappa}_{B,C}^k(x) = \int_{x/B^{k-1-j}C^j}^C \tilde{\kappa}_{B,C}^{k-1}(x/p_k) dp_k + \int_B^{x/B^{k-1-j}C^j} \tilde{\kappa}_{B,C}^{k-1}(x/p_k) dp_k.$$

Also for $\hat{\kappa}_{B,C}^k$ this can be done, simply split the occurring integrals at $x/B^{k-1-j}C^j$. Now unfolding the recursive definition of $\tilde{\kappa}_{B,C}^1$ gives:

$$\tilde{\kappa}_{B,C}^1(x) = \begin{cases} 0 & \text{if } x \in [0, B[, \\ \int_B^x \frac{1}{\ln p_1} dp_1 & \text{if } x \in [B, C[, \\ \int_B^C \frac{1}{\ln p_1} dp_1 & \text{if } x \in [C, \infty[, \end{cases}$$

$$\hat{\kappa}_{B,C}^1(x) = \begin{cases} 0 & \text{if } x \in [0, B[, \\ \hat{E}(x) + \hat{E}(B) & \text{if } x \in [B, C[, \\ \hat{E}(C) + \hat{E}(B) & \text{if } x \in [C, \infty[, \end{cases}$$

which corresponds exactly to the approximation of the prime counting function by the logarithmic integral Li. In case $k=2$ we obtain

$$(6.3.3) \quad \tilde{\kappa}_{B,C}^2(x) = \begin{cases} 0 & \text{if } x \in [0, B^2[, \\ \int_B^{\frac{x}{B}} \int_B^{\frac{x}{p_2}} \frac{1}{\ln p_1 \ln p_2} dp_1 dp_2 & \text{if } x \in [B^2, BC[, \\ \int_{\frac{x}{C}}^C \int_B^{\frac{x}{p_2}} \frac{1}{\ln p_1 \ln p_2} dp_1 dp_2 & \text{if } x \in [BC, C^2[, \\ + \int_B^{\frac{x}{C}} \int_B^C \frac{1}{\ln p_1 \ln p_2} dp_1 dp_2 & \\ \int_B^C \int_B^C \frac{1}{\ln p_1 \ln p_2} dp_1 dp_2 & \text{if } x \in [C^2, \infty[. \end{cases}$$

This is now exactly the transformed version of (6.1.3), as announced there. You may ask where we can find a display of the error term corresponding to (6.3.3). Well, we have computed it. But the resulting terms are so complex that we didn't really learn much from it. Here is an expression for $x \in [B^2, BC[$:

$$\hat{\kappa}_{B,C}^2(x) = \int_B^{\frac{x}{B}} \int_B^{\frac{x}{p}} \left(\frac{\hat{E}'(q)}{\ln p} + \frac{\hat{E}'(p)}{\ln q} + \hat{E}'(p)\hat{E}'(q) \right) dq dp$$

$$+ 4\hat{E}(B) \int_B^{\frac{x}{B}} \frac{1}{\ln p} dp + 4\hat{E}(B)\hat{E}(x/B).$$

Though this term can still be handled, it becomes apparent that things get more and more complicated with increasing k . We escape from this issue by

loosening the bonds and weakening our bounds slightly. The first aim will be to obtain easily computable terms while retaining the asymptotic orders, the second aim will be to still retain meaningful bounds for the fixed values $B = 1100 \cdot 10^6$, $C = 2^{37} - 1$, $k \in \{2, 3, 4\}$ from our inspiring application.

Our next task is to describe the orders of $\tilde{\kappa}_{B,C}^k$ and $\hat{\kappa}_{B,C}^k$. The main problem in an exact calculation is that most of the time we cannot elementary integrate a function with a logarithm occurring in the denominator. But using $B \leq p \leq C$ we can obtain a suitably good approximation instead by replacing $\frac{1}{\ln p}$ with $\frac{1}{\ln B}$ in the integrals. At this point we start using $C \leq B^s$ and rewrite $C = B^{1+\alpha}$ where α is a new parameter (bounded by $s - 1$). For the time being we can consider α as a constant, but actually we make no assumption on it. This leads to the following

DEFINITION 6.3.4. For $x \geq 0$ we let

$$\tilde{\lambda}^0(x) := \kappa_{B,C}^0(x), \quad \hat{\lambda}^0(x) := 0,$$

and recursively for $k > 0$

$$\begin{aligned} \tilde{\lambda}^k(x) &:= \int_B^C \frac{\tilde{\lambda}^{k-1}(x/p_k)}{\ln B} dp_k, \\ \hat{\lambda}^k(x) &:= \int_B^C \frac{\hat{\lambda}^{k-1}(x/p_k)}{\ln B} dp_k \\ &\quad + 2 \left(\tilde{\lambda}^{k-1} + \hat{\lambda}^{k-1} \right) (x/B) \hat{E}(B) \\ &\quad + \int_B^C \left(\tilde{\lambda}^{k-1} + \hat{\lambda}^{k-1} \right) (x/p_k) \hat{E}'(p_k) dp_k. \end{aligned}$$

We observe that $\ln B \leq \ln p_k \leq \ln C = (1 + \alpha) \ln B$. Thus we obtain $\int_B^C \frac{f(p)}{\ln p} dp \in \left[\frac{1}{1+\alpha}, 1 \right] \int_B^C \frac{f(p)}{\ln B} dp$ for any positive integrable function f . By induction on k we obtain

THEOREM 6.3.5. Write $C = B^{1+\alpha}$ and fix $k \in \mathbb{N}_{>0}$. Then for $x \in \mathbb{R}_{>0}$ we have

$$\begin{aligned} \tilde{\kappa}_{B,C}^k(x) &\in \left[\frac{1}{(1+\alpha)^k}, 1 \right] \tilde{\lambda}^k(x), \\ \hat{\kappa}_{B,C}^k(x) &\in \left[\frac{1}{(1+\alpha)^k}, 1 \right] \hat{\lambda}^k(x). \quad \square \end{aligned}$$

In order to determine at least the asymptotic orders of $\tilde{\kappa}_{B,C}^k$ and $\hat{\kappa}_{B,C}^k$ (along with precise estimates) it remains to solve the recursions for $\tilde{\lambda}^k$ and $\hat{\lambda}^k$, or at least to estimate these functions. As a first step, we rewrite all integrals in Definition 6.3.4 in terms of ϱ defined by $p_k = B^{1+\varrho\alpha}$.

DEFINITION 6.3.4 CONTINUED. Rewrite $x = B^{k+\xi\alpha}$ with a new parameter ξ and let $\tilde{\lambda}^k \langle \xi \rangle := \tilde{\lambda}^k(B^{k+\xi\alpha})$ and $\hat{\lambda}^k \langle \xi \rangle := \hat{\lambda}^k(B^{k+\xi\alpha})$. (Actually, you may think of $f^k \langle \xi \rangle := f^k(B^{k+\xi\alpha})$ for any family of functions f^k .)

LEMMA 6.3.6. For $k > 0$ we now have the recursion

$$\begin{aligned}\tilde{\lambda}^k \langle \xi \rangle &= \alpha \int_0^1 \tilde{\lambda}^{k-1} \langle \xi - \varrho \rangle B^{1+\varrho\alpha} d\varrho, \\ \hat{\lambda}^k \langle \xi \rangle &= \alpha \int_0^1 \hat{\lambda}^{k-1} \langle \xi - \varrho \rangle B^{1+\varrho\alpha} d\varrho \\ &\quad + 2 \left(\tilde{\lambda}^{k-1} + \hat{\lambda}^{k-1} \right) \langle \xi \rangle \hat{E}(B) \\ &\quad + \alpha \ln B \int_0^1 \left(\tilde{\lambda}^{k-1} + \hat{\lambda}^{k-1} \right) \langle \xi - \varrho \rangle \hat{E}'(B^{1+\varrho\alpha}) B^{1+\varrho\alpha} d\varrho. \quad \square\end{aligned}$$

6.4. Solving the recursion for $\tilde{\lambda}^k$

To construct a useful description of the function $\tilde{\lambda}^k$ we make a small excursion and consider the following family of piece-wise polynomial functions.

DEFINITION 6.4.1 (Polynomial hills). Initially, define the integral operator \mathcal{M} by

$$(\mathcal{M}f)(\xi) = \int_0^1 f(\xi - \varrho) d\varrho = \int_{\xi-1}^{\xi} f(\varrho) d\varrho$$

for any integrable function $f: \mathbb{R} \rightarrow \mathbb{R}$. Now for $k \in \mathbb{N}_{>1}$ let the k -th polynomial hill be

$$\tilde{m}^k := \mathcal{M}\tilde{m}^{k-1}$$

based on the rectangular function \tilde{m}^1 given by $\tilde{m}^1(\xi) = 1$ for $\xi \in [0, 1[$ and $\tilde{m}^1(\xi) = 0$ otherwise.

Actually, $\tilde{m}^1 = \mathcal{M}\tilde{m}^0$ if we let $\tilde{m}^0 = \delta$ be the ‘left lopsided’ Dirac delta distribution defined by its integral $\int_{-\infty}^{\xi} \delta(t) dt$ being 0 for $\xi < 0$ and 1 for $\xi \geq 0$. Contrastingly, the standard Dirac delta function is balanced and has $\int_{-\infty}^0 \delta(t) dt = \frac{1}{2}$. However, we will stick to the lopsided variant throughout the entire paper. By \mathcal{D}_{ξ} we denote the differential operator with respect to ξ .

LEMMA 6.4.2 (Polynomial hills). (i) $\tilde{m}^k(\xi) = 0$ for $\xi < 0$ or $\xi \geq k$.

(ii) $\tilde{m}^k(\xi) = \frac{1}{(k-1)!}\xi^{k-1}$ for $\xi \in [0, 1[$ and $\tilde{m}^k(\xi) = \frac{1}{(k-1)!}(k-\xi)^{k-1}$ for $\xi \in [k-1, k[$.

(iii) \tilde{m}^k restricted to $[j, j+1[$ is a polynomial function of degree $k-1$ for $j \in \{0, \dots, k-1\}$. In particular, \tilde{m}^k is smooth for $\xi \in \mathbb{R} \setminus \{0, \dots, k\}$.

(iv) \tilde{m}^k is $(k-2)$ -fold continuously differentiable.

(v) Conversely, the conditions (i) through (iv) uniquely determine \tilde{m}^k .

(vi) $\mathcal{D}_\xi^i \tilde{m}^k = \mathcal{M} \mathcal{D}_\xi^i \tilde{m}^{k-1}$ as long as $i \leq k-2$ and even for $i = k-1$ when read for distributions.

(vii) The function \tilde{m}^k is symmetric to $\frac{k}{2}$: $\tilde{m}^k(\xi) = \tilde{m}^k(k-\xi)$.

(viii) For $\xi \in [j, j+1[$ (corresponding to case (k, j)) we have

$$\mathcal{D}_\xi^{k-1} \tilde{m}^k(\xi) = (-1)^j \cdot \binom{k-1}{j}.$$

(ix) The next observation can only be correctly described as a linear combination of Dirac delta distributions:

$$\mathcal{D}_\xi^k \tilde{m}^k(\xi) = \sum_{0 \leq j \leq k} (-1)^j \binom{k}{j} \cdot \delta(\xi - j).$$

(x) For any $0 \leq \ell < k$ we obtain the following explicit description:

$$\mathcal{D}_\xi^\ell \tilde{m}^k(\xi) = \frac{1}{(k-1-\ell)!} \sum_{0 \leq i \leq \lfloor \xi \rfloor} \binom{k}{i} (-1)^i (\xi - i)^{k-1-\ell}.$$

(xi) Further, $\tilde{m}^k(\xi) = \frac{\xi}{k-1} \tilde{m}^{k-1}(\xi) + \frac{k-\xi}{k-1} \tilde{m}^{k-1}(\xi-1)$ holds.

Curiosity: The function \tilde{m}^k can also be described as the volume of a slice through a $(k-1)$ -dimensional unit hypercube of thickness $\frac{1}{\sqrt{k}}$ orthogonal to a main diagonal. That's the same as saying it is the $(k-1)$ -volume of the cut between a k -dimensional hypercube and the hyperplane $\sum_{1 \leq i \leq k} \varrho_i = \xi$. This last interpretation makes it obvious that

$$\int_0^k \tilde{m}^k(\xi) d\xi = 1,$$

since this is the volume of the k - hypercube.

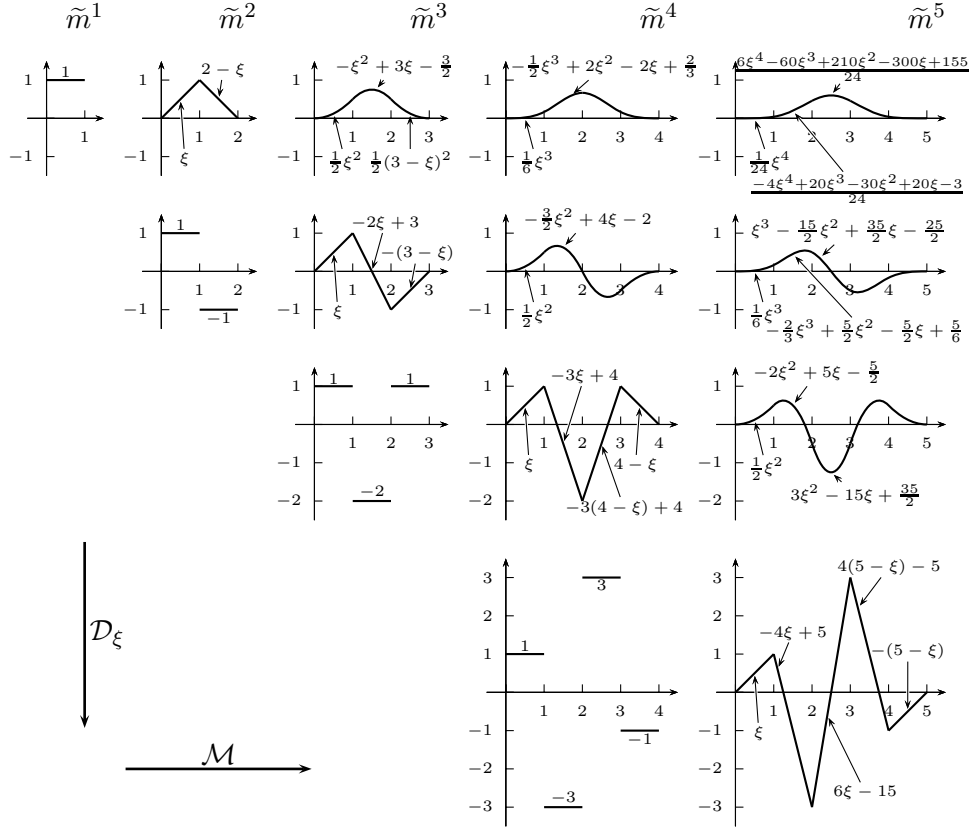


Figure 6.4.1: Graphs of the polynomial hills \tilde{m}^k for $k \leq 5$ and their derivatives

PROOF. From the definition (i) through (vii) follow directly. Further, note that $(\mathcal{M}\mathcal{D}_\xi f)(\xi) = f(\xi) - f(\xi - 1)$. This is obvious from $(\mathcal{M}f)(\xi) = \int_{\xi-1}^\xi f(\varrho) d\varrho$.

We prove (viii) by induction on k . For $k = 1$ this is true by definition. So consider $k > 1$. The recursion for \tilde{m} differentiated $k - 1$ times yields for $\xi \in [j, j + 1[$

$$\begin{aligned} \mathcal{D}_\xi^{k-1} \tilde{m}^k(\xi) &= \mathcal{D}_\xi \mathcal{M} \mathcal{D}_\xi^{k-2} \tilde{m}^{k-1}(\xi) \\ &= \mathcal{D}_\xi^{k-2} \tilde{m}^{k-1}(\xi) - \mathcal{D}_\xi^{k-2} \tilde{m}^{k-1}(\xi - 1) \\ &= (-1)^j \binom{k-2}{j} - (-1)^{j-1} \binom{k-2}{j-1} = (-1)^j \binom{k-1}{j}. \end{aligned}$$

(ix) follows similarly.

To prove (x) consider $h(\xi) = \frac{1}{(k-1)!} \sum_{0 \leq i \leq \lfloor \xi \rfloor} \binom{k}{i} (-1)^i (\xi - i)^{k-1}$. Then $\mathcal{D}_\xi^{k-1} h = \mathcal{D}_\xi^{k-1} \tilde{m}^k$ using (viii). And obviously $\mathcal{D}_\xi^\ell h(0) = 0 = \mathcal{D}_\xi^\ell \tilde{m}^k(0)$ for $0 \leq \ell < k-1$, so that inductively (with falling ℓ) we get $\mathcal{D}_\xi^\ell h = \mathcal{D}_\xi^\ell \tilde{m}^k$.

As we do not need (v) and (xi), we leave these proofs to the interested reader. \square

Most of the following is easier if we first renormalize $\tilde{\lambda}^k$. So we let

$$(6.4.3) \quad \tilde{\lambda}_{\text{norm}}^k \langle \xi \rangle := \frac{1}{\alpha^k B^{k+\xi\alpha}} \tilde{\lambda}^k \langle \xi \rangle.$$

The recursion for $\tilde{\lambda}^k$ now turns into

$$\tilde{\lambda}_{\text{norm}}^k = \mathcal{M} \tilde{\lambda}_{\text{norm}}^{k-1}.$$

THEOREM 6.4.4 (Approximation order, Nüsken 2006-2011). *For any $\xi \in \mathbb{R}$ we have*

$$\begin{aligned} \tilde{\lambda}_{\text{norm}}^k \langle \xi \rangle &= \int_0^\xi B^{-\varrho\alpha} \tilde{m}^k(\xi - \varrho) d\varrho = B^{-\xi\alpha} \int_0^\xi B^{\varrho\alpha} \tilde{m}^k(\varrho) d\varrho, \\ \tilde{\lambda}_{\text{norm}}^k \langle \xi \rangle &= \frac{1}{(-\alpha \ln B)^k} \left(\sum_{0 \leq i \leq \lfloor \xi \rfloor} \binom{k}{i} (-1)^i B^{-(\xi-i)\alpha} \right. \\ &\quad \left. - \sum_{0 \leq \ell \leq k-1} (-\alpha \ln B)^\ell \cdot \mathcal{D}_\xi^{k-\ell-1} \tilde{m}^k(\xi) \right), \\ \tilde{\lambda}_{\text{norm}}^k \langle \xi \rangle &= \sum_{0 \leq i \leq \lfloor \xi \rfloor} \binom{k}{i} (-1)^i \frac{\text{cutexp}_k(-(\xi-i)\alpha \ln B)}{(-\alpha \ln B)^k}, \end{aligned}$$

where $\text{cutexp}_k(\zeta) = \exp(\zeta) - \sum_{0 \leq \ell \leq k-1} \frac{\zeta^\ell}{\ell!} = \sum_{\ell \geq k} \frac{\zeta^\ell}{\ell!}$. We can also express cutexp_k using the incomplete Gamma function $\Gamma(k, \zeta) = \int_\zeta^\infty e^{-\varrho} \varrho^{k-1} d\varrho$ by $\text{cutexp}_k(\zeta) = \exp(\zeta) - \frac{\Gamma(k, \zeta)}{\exp(-\zeta)\Gamma(k, 0)}$.

PROOF. The definition for $\tilde{\lambda}^0$ turns into

$$\tilde{\lambda}_{\text{norm}}^0 \langle \xi \rangle = \int_0^\infty B^{-\varrho\alpha} \delta(\xi - \varrho) d\varrho.$$

Now, since \mathcal{M} commutes with this integration this immediately implies that

$$\tilde{\lambda}_{\text{norm}}^k \langle \xi \rangle = \int_0^\infty B^{-\varrho\alpha} \tilde{m}^k(\xi - \varrho) d\varrho$$

which is the first stated equality noting that \tilde{m}^k is zero outside $[0, k]$. By partial integration we obtain

$$\begin{aligned}\tilde{\lambda}_{\text{norm}}^k \langle \xi \rangle &= \frac{1}{\alpha \ln B} \tilde{m}^k(\xi) - \frac{1}{\alpha \ln B} \int_0^\infty B^{-\varrho\alpha} \mathcal{D}_\xi \tilde{m}^k(\xi - \varrho) \, \mathrm{d}\varrho \\ &= \sum_{1 \leq i \leq k} \frac{(-1)^{i-1}}{(\alpha \ln B)^i} \mathcal{D}_\xi^{i-1} \tilde{m}^k(\xi) + \frac{(-1)^k}{(\alpha \ln B)^k} \int_0^\infty B^{-\varrho\alpha} \mathcal{D}_\xi^k \tilde{m}^k(\xi - \varrho) \, \mathrm{d}\varrho.\end{aligned}$$

Using the description of $\mathcal{D}_\xi^k \tilde{m}^k$ from Lemma 6.4.2(ix) the last integral turns into the claimed sum of the second stated equality. Expressing $\mathcal{D}_\xi^{k-\ell-1} \tilde{m}^k(\xi - \varrho)$ using Lemma 6.4.2(x) and rearranging slightly yields the third equality. \square

We are going to estimate the estimation of the error in the next section. To that aim we first need to estimate $\tilde{\lambda}_{\text{norm}}^k$. If $k > 0$ then, based on Theorem 6.4.4 and $\tilde{m}^k(\xi) \leq 1$, we obtain the upper bound

$$\tilde{\lambda}_{\text{norm}}^k \langle \xi \rangle = \int_0^\infty B^{-\varrho\alpha} \tilde{m}^k(\xi - \varrho) \, \mathrm{d}\varrho \leq \frac{1}{\alpha \ln B}.$$

For $k = 0$ we have $\tilde{\lambda}_{\text{norm}}^0 \langle \xi \rangle = B^{-\xi\alpha}$ for $\xi \geq 0$ and so $\tilde{\lambda}_{\text{norm}}^0 \langle \xi \rangle \leq 1$ will do for all $\xi \in \mathbb{R}$.

We first describe the qualitative behaviour of $\tilde{\lambda}_{\text{norm}}^k$. Actually its graph looks like a slightly biased hill.

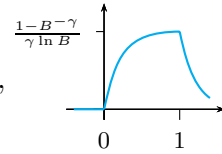
LEMMA 6.4.5 (Nüsken 2006-2011). *The function $\tilde{\lambda}_{\text{norm}}^k \langle \xi \rangle = B^{-\xi\alpha} \int_0^\xi B^{\varrho\alpha} \tilde{m}^k(\varrho) \, \mathrm{d}\varrho$ is zero at $\xi = 0$, positive at $\xi = k$, more precisely*

$$\tilde{\lambda}_{\text{norm}}^k \langle k \rangle = \left(\frac{1 - B^{-\alpha}}{\alpha \ln B} \right)^k,$$

and there is a position $\xi_{\frac{1}{2}}^k \in]0, k[$ such that it is increasing on $]0, \xi_{\frac{1}{2}}^k[$ and decreasing on $]\xi_{\frac{1}{2}}^k, \xi[$. Further, $\xi_{\frac{1}{2}}^k \geq \frac{k}{2}$.

PROOF. First, inspecting

$$\begin{aligned}\tilde{\lambda}_{\text{norm}}^1 \langle \xi \rangle &= \int_0^\xi B^{-\varrho\alpha} \tilde{m}^1(\xi - \varrho) \, \mathrm{d}\varrho \\ &= \begin{cases} 0 & \text{if } \xi < 0, \\ \frac{1 - B^{-\xi\alpha}}{\alpha \ln B} & \text{if } \xi \in [0, 1], \\ \frac{1 - B^{-\alpha}}{\alpha \ln B} B^{-(\xi-1)\alpha} & \text{if } \xi > 1. \end{cases}\end{aligned}$$



shows that for $k = 1$ all claims hold with $\xi_{\frac{1}{2}}^k = 1$. So in the remainder of this proof we assume $k > 1$.

Next, compute $\tilde{\lambda}_{\text{norm}}^k \langle k \rangle$ inductively:

$$\begin{aligned} \tilde{\lambda}_{\text{norm}}^k \langle k \rangle &= B^{-k\alpha} \int_0^k B^{\varrho\alpha} \int_0^1 \tilde{m}^{k-1}(\varrho - \varrho_k) d\varrho_k d\varrho \\ &= B^{-\alpha} \underbrace{\int_0^1 B^{\varrho_k\alpha} d\varrho_k}_{=\frac{1-B^{-\alpha}}{\alpha \ln B}} \cdot \underbrace{B^{-(k-1)\alpha} \int_0^{k-1} B^{\tau\alpha} \tilde{m}^{k-1}(\tau) d\tau}_{=\tilde{\lambda}_{\text{norm}}^{k-1} \langle k-1 \rangle} = \left(\frac{1-B^{-\alpha}}{\alpha \ln B} \right)^k. \end{aligned}$$

Here we have substituted $\tau = \varrho - \varrho_k$ and collapsed the new integration interval $[-\varrho_k, k - \varrho_k]$ for τ to $[0, k-1]$ since \tilde{m}^{k-1} vanishes on the difference. Moreover, based on $\tilde{\lambda}_{\text{norm}}^k \langle \xi \rangle = B^{-\xi\alpha} \int_0^\xi B^{\varrho\alpha} \tilde{m}^k(\varrho) d\varrho$ we obtain

$$(6.4.6) \quad \mathcal{D}_\xi \tilde{\lambda}_{\text{norm}}^k \langle \xi \rangle = -\alpha \ln B \cdot \tilde{\lambda}_{\text{norm}}^k \langle \xi \rangle + \tilde{m}^k(\xi),$$

and infer that $\mathcal{D}_\xi \tilde{\lambda}_{\text{norm}}^k \langle k \rangle = -\alpha \ln B \left(\frac{1-B^{-\alpha}}{\alpha \ln B} \right)^k$ is negative.

Finally, compute the derivate of $\tilde{\lambda}_{\text{norm}}^k$ differently

$$\mathcal{D}_\xi \tilde{\lambda}_{\text{norm}}^k \langle \xi \rangle = \mathcal{D}_\xi \left(\int_0^\xi B^{-\varrho\alpha} \tilde{m}^k(\xi - \varrho) d\varrho \right) = B^{-\xi\alpha} \int_0^\xi B^{\varrho\alpha} \mathcal{D}_\xi \tilde{m}^k(\varrho) d\varrho.$$

The integral kernel $B^{\varrho\alpha} \mathcal{D}_\xi \tilde{m}^k(\varrho)$ is positive on $]0, \frac{k}{2}[$ and negative on $]\frac{k}{2}, k[$. Thus $\int_0^\xi B^{\varrho\alpha} \mathcal{D}_\xi \tilde{m}^k(\varrho) d\varrho$ increases on $]0, \frac{k}{2}[$ and decreases on $]\frac{k}{2}, k[$. Since this term starts at zero, it is positive for some time, begins to decrease at $\frac{k}{2}$, traverses zero at some point $\xi_{\frac{1}{2}}^k$ recalling that at k the value is negative, and stays negative until $\xi = k$ since it continues to decrease. Thus the sign of $\mathcal{D}_\xi \tilde{\lambda}_{\text{norm}}^k \langle \xi \rangle$ is positive on $]0, \xi_{\frac{1}{2}}^k[$ and negative on $]\xi_{\frac{1}{2}}^k, k[$, and so $\tilde{\lambda}_{\text{norm}}^k \langle \xi \rangle$ is increasing till $\xi_{\frac{1}{2}}^k$ and decreasing afterwards. \square

LEMMA 6.4.7 (Nüsken 2006-2011). Assume $k \geq 2$, $\alpha \ln B \geq \frac{\ln 16}{k}$, and $\xi \in [0, 1]$. Then we have

$$\tilde{\lambda}_{\text{norm}}^k \langle \xi \rangle \geq \frac{\exp\left(-\frac{\ln^2 4}{\alpha \ln B}\right)}{\alpha \ln B} \cdot \frac{\xi^k}{k!}.$$

The assumptions are already true for $C = 2B$ when $k \geq 4$. Note that this is rather sharp as for $\xi \in [0, 1]$ we have $\tilde{\lambda}_{\text{norm}}^k \langle \xi \rangle \leq \frac{\xi^k}{k!}$.

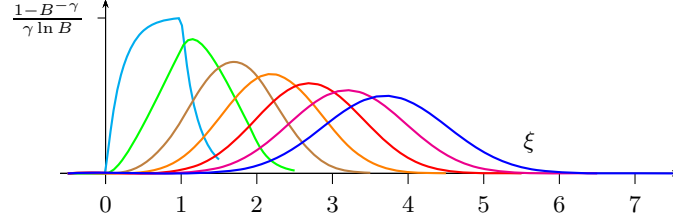
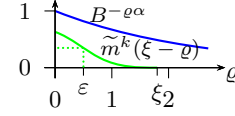


Figure 6.4.2: $\tilde{\lambda}_{\text{norm}}^k \langle \xi \rangle$ for $\xi \in [-\frac{1}{2}, k + \frac{1}{2}]$ and $k = 1, 2, 3, 4, 5, 6, 7$

PROOF. We use the integral representation from Theorem 6.4.4 and estimate the polynomial hill part $\tilde{m}^k(\xi - \varrho)$ of its kernel by a simple piece-wise constant function as indicated in the picture. We obtain

$$\begin{aligned} \tilde{\lambda}_{\text{norm}}^k \langle \xi \rangle &= \int_0^\xi B^{-\varrho\alpha} \tilde{m}^k(\xi - \varrho) d\varrho \\ &\geq \int_0^\varepsilon B^{-\varrho\alpha} d\varrho \cdot \tilde{m}^k(\xi - \varepsilon) \\ &= \frac{1}{\alpha \ln B} (1 - \exp(-\varepsilon\alpha \ln B)) \tilde{m}^k(\xi - \varepsilon). \end{aligned}$$



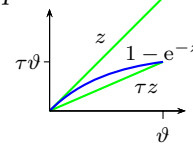
This holds for any $\varepsilon \in [0, \xi]$ since $0 \leq \xi \leq \frac{k}{2}$ ensures that \tilde{m}^k is increasing. So we can optimize ε depending on ξ . We obtain a suitable value when setting ε by

$$(6.4.8) \quad 1 - \exp(-\varepsilon\alpha \ln B) = \frac{\xi}{k}.$$

To make sure that now $\varepsilon \leq \xi$ we use the following simple fact.

FACT 6.4.9. For any $\vartheta > 0$ and $\tau = \frac{1 - \exp(-\vartheta)}{\vartheta}$ the map

$$\begin{aligned} [0, \vartheta] &\longrightarrow [0, \tau\vartheta], \\ z &\longmapsto 1 - \exp(-z), \end{aligned}$$



is bijective and increasing and for $z \in [0, \vartheta]$ we have $\tau z \leq 1 - \exp(-z) \leq z$. \triangle

Let $\vartheta_1 > 0$ be such that $1 - \exp(-\vartheta_1) = \frac{1}{k}$. Namely, $\vartheta_1 = -\ln\left(1 - \frac{1}{k}\right)$. With $\tau_1 = \frac{1 - \exp(-\vartheta_1)}{\vartheta_1}$ then $1 = \tau_1 \vartheta_1 k$ and $\frac{\xi}{k} \leq \tau_1 \vartheta_1$. Thus we have $\varepsilon\alpha \ln B \in [0, \vartheta_1]$ so that

$$(6.4.10) \quad \tau_1 \varepsilon\alpha \ln B \leq 1 - \exp(-\varepsilon\alpha \ln B) = \frac{\xi}{k} \leq \varepsilon\alpha \ln B.$$

In particular, $\varepsilon \leq \xi$ follows from $k\tau_1\alpha \ln B = \frac{1}{\vartheta_1}\alpha \ln B \geq 1$. By Fact 6.4.9 with $\vartheta = \ln 2$ we obtain $\tau = \frac{1}{2\ln 2}$ and $\left(1 - \frac{1}{k}\right)^k \geq \exp(-\frac{1}{\tau}) = \exp(-\ln 4)$ for all $k \geq 2$. Thus $k\vartheta_1 = -k \ln\left(1 - \frac{1}{k}\right) \leq \ln 4$. Further, $\alpha \ln B \geq \frac{\ln 16}{k} > \frac{\ln 4}{k} \geq \vartheta_1$. This now implies $\varepsilon \leq \xi$.

Since $\xi \in [0, 1]$ we have the explicit expression $\tilde{m}^k(\xi - \varepsilon) = \frac{(\xi - \varepsilon)^{k-1}}{(k-1)!}$ and so

$$\tilde{\lambda}_{\text{norm}}^k \langle \xi \rangle \geq \frac{\xi}{k\alpha \ln B} \tilde{m}^k(\xi - \varepsilon) = \frac{(1 - \varepsilon/\xi)^{k-1}}{\alpha \ln B} \cdot \frac{\xi^k}{k!}.$$

Since $\vartheta_1 < \alpha \ln B$ we can define ϑ_2 by $1 - \exp(-\vartheta_2) = \frac{\vartheta_1}{\alpha \ln B}$, and according to Fact 6.4.9 let $\tau_2 = \frac{1 - \exp(-\vartheta_2)}{\vartheta_2}$. Combining with (6.4.10) gives us $\left(1 - \frac{\varepsilon}{\xi}\right)^k \geq \left(1 - \frac{1}{k\tau_1\alpha \ln B}\right)^k = \exp\left(-\frac{1}{\tau_2\tau_1\alpha \ln B}\right)$ and thus simplifies our above inequality to

$$(6.4.11) \quad \tilde{\lambda}_{\text{norm}}^k \langle \xi \rangle \geq \frac{\exp\left(-\frac{1}{\tau_2\tau_1\alpha \ln B}\right)}{\alpha \ln B} \cdot \frac{\xi^k}{k!}.$$

By our choices

$$\frac{1}{\tau_1\tau_2\alpha \ln B} = k\vartheta_2 \geq \frac{k\vartheta_1}{\alpha \ln B} \geq \frac{1}{\alpha \ln B}.$$

Though these inequalities get equalities with $k \rightarrow \infty$, we need a precise estimate. Substituting k in $-k \ln\left(1 - \frac{1}{k}\right) \leq \ln 4$ with $\frac{k\alpha \ln B}{\ln 4}$ yields

$$\begin{aligned} \frac{1}{\tau_2\tau_1} &= k\alpha \ln B \cdot \vartheta_2 = -k\alpha \ln B \ln\left(1 - \frac{k\vartheta_1}{k\alpha \ln B}\right) \\ &\leq -\frac{k\alpha \ln B}{\ln 4} \ln\left(1 - \frac{\ln 4}{k\alpha \ln B}\right) \ln 4 \leq \ln^2 4 \end{aligned}$$

provided $\alpha \ln B \geq \frac{\ln 16}{k}$. □

Though we know the value of $\tilde{\lambda}_{\text{norm}}^k \langle k \rangle$, it is orders smaller than the above left lower bound. Thus let us consider $\tilde{\lambda}_{\text{norm}}^k$ on $[k-1, k]$.

LEMMA 6.4.12 (Nüsken 2006-2011). *If $k \geq 3$ then for $\xi \in [k-1, k]$ we have*

$$\tilde{\lambda}_{\text{norm}}^k \langle \xi \rangle \geq \frac{(1 - B^{-\alpha})^k}{\alpha \ln B} \cdot \frac{(k - \xi)^{k-1}}{(k-1)!}.$$

PROOF. By definition we have $\tilde{\lambda}_{\text{norm}}^k \langle \xi \rangle = \frac{\tilde{\lambda}^k \langle \xi \rangle}{\alpha^k B^{k+\xi\alpha}}$. Theorem 6.4.4's third description expresses $\tilde{\lambda}_{\text{norm}}^k$ on $[k-1, k]$ as a sum of k terms. Adding the missing term $i = k$ we obtain $\frac{\tilde{\lambda}^k \langle k \rangle}{\alpha^k B^{k+\xi\alpha}}$, noting that $\tilde{\lambda}^k$ is constant for $\xi > k$:

$$\tilde{\lambda}_{\text{norm}}^k \langle \xi \rangle = \left(\frac{1 - B^{-\alpha}}{\alpha \ln B} \right)^k B^{(k-\xi)\alpha} - \frac{\text{cutexp}_k((k-\xi)\alpha \ln B)}{(\alpha \ln B)^k}.$$

To check the claimed inequality we substitute $\tau = (k-\xi)\alpha \ln B \in [0, \alpha \ln B]$ (eliminating ξ):

$$\tilde{\lambda}_{\text{norm}}^k \langle \xi \rangle = \frac{\sum_{0 \leq \ell \leq k-1} \frac{\tau^\ell}{\ell!} - \left(1 - (1 - B^{-\alpha})^k\right) e^\tau}{(\alpha \ln B)^k}.$$

We have to show that this is at least $\frac{(1-B^{-\alpha})^k}{(\alpha \ln B)^k} \frac{\tau^{k-1}}{(k-1)!}$ which we rewrite to

$$\sum_{0 \leq \ell \leq k-2} \frac{\tau^\ell}{\ell!} \geq \left(1 - (1 - B^{-\alpha})^k\right) \left(e^\tau - \frac{\tau^{k-1}}{(k-1)!} \right).$$

Obviously $(1 - B^{-\alpha})^k \geq (1 - e^{-\tau})^k$ in our situation, with equality for $\tau = \alpha \ln B$ or $\xi = k-1$. Thus it suffices to show for any $\tau > 0$

$$(6.4.13) \quad \sum_{0 \leq \ell \leq k-2} \frac{\tau^\ell}{\ell!} \geq (1 - (1 - e^{-\tau})^k) \left(e^\tau - \frac{\tau^{k-1}}{(k-1)!} \right).$$

The remaining proof proceeds in four steps:

- High case: $\sum_{0 \leq \ell \leq k-2} \frac{\tau^\ell}{\ell!} \geq k$.
- Low case: $\frac{1-e^{-\tau}}{\tau} \geq \sqrt[k]{\frac{1+\sigma}{k!}}$, where $\frac{1}{\sigma} + 1 \leq \frac{e^{k-1}(k-1)!}{(k-1)^{k-1}}$.
- Covering: Fixing $\sigma := \frac{1}{\sqrt{2\pi(k-1)-1}}$ these cases cover all $\tau > 0$ if $k \geq 4$.
- Brute-force: Prove (6.4.13) for $k = 3$. (Actually, for $k \in \{3, 4, 5, 6, 7, 8, 9\}$.)

High case: Since $e^\tau \geq \sum_{0 \leq \ell \leq k-2} \frac{\tau^\ell}{\ell!} \geq k$ in this case we have $\tau \geq \ln k$. Employing Bernoulli's inequality $(1-T)^k \geq 1 - kT$ for $T = e^{-\tau} \leq 1$ we obtain

$$\begin{aligned} (1 - (1 - e^{-\tau})^k) \left(e^\tau - \frac{\tau^{k-1}}{(k-1)!} \right) &\leq k e^{-\tau} \left(e^\tau - \frac{\tau^{k-1}}{(k-1)!} \right) \\ &= k \left(1 - \frac{\tau^{k-1}}{(k-1)!} e^{-\tau} \right) \leq k \leq \sum_{0 \leq \ell \leq k-2} \frac{\tau^\ell}{\ell!}. \end{aligned}$$

As Bernoulli's inequality is good for T close to 0 only, it is not surprising that this only gives a sufficient result for τ large enough.

Low case: Assume $\frac{1-e^{-\tau}}{\tau} \geq \sqrt[k]{\frac{1+\sigma}{k!}}$ where σ is chosen such that $\frac{1}{\sigma} + 1 < \frac{e^{k-1}(k-1)!}{(k-1)^{k-1}}$. (Since $\frac{1-e^{-\tau}}{\tau}$ is decreasing, this is always true on some interval $[0, \tau_0]$.)

The condition on σ implies that $e^\tau - \left(\frac{1}{\sigma} + 1\right) \frac{\tau^{k-1}}{(k-1)!}$ is non-negative for all $\tau > 0$: Consider $f_0(\tau) = \frac{e^\tau(k-1)!}{\tau^{k-1}} - \left(\frac{1}{\sigma} + 1\right)$. Then f_0' vanishes at $\tau = k-1$ only and thus f_0 is minimal there. The assumption on σ is precisely $f_0(k-1) \geq 0$.

Further, note that $\frac{\tau^{k-2}}{(k-2)!} + \frac{\tau^{k-1}}{(k-1)!} \leq e^\tau \leq \sum_{0 \leq \ell \leq k-1} \frac{\tau^\ell}{\ell!} + e^\tau \frac{\tau^k}{k!}$. We use this to obtain:

$$\begin{aligned} & \left(1 - \underbrace{(1 - e^{-\tau})^k}_{\geq \frac{(1+\sigma)\tau^k}{k!}}\right) \underbrace{\left(e^\tau - \frac{\tau^{k-1}}{(k-1)!}\right)}_{\geq 0} \\ & \leq e^\tau - \frac{\tau^{k-1}}{(k-1)!} - (1+\sigma) \frac{\tau^k}{k!} e^\tau + (1+\sigma) \frac{\tau^k}{k!} \frac{\tau^{k-1}}{(k-1)!} \\ & \leq \sum_{0 \leq \ell \leq k-2} \frac{\tau^\ell}{\ell!} - \frac{\sigma \tau^k}{k!} e^\tau + (1+\sigma) \frac{\tau^k}{k!} \frac{\tau^{k-1}}{(k-1)!} \\ & = \sum_{0 \leq \ell \leq k-2} \frac{\tau^\ell}{\ell!} - \sigma \frac{\tau^k}{k!} \underbrace{\left(e^\tau - \left(\frac{1}{\sigma} + 1\right) \frac{\tau^{k-1}}{(k-1)!}\right)}_{\geq 0} \leq \sum_{0 \leq \ell \leq k-2} \frac{\tau^\ell}{\ell!}. \end{aligned}$$

Notice that the value of σ only influences the set of values of τ that fall in this case.

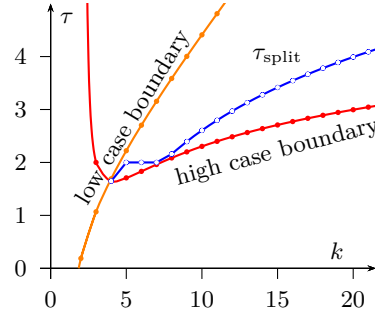
Covering: We choose $\sigma := \frac{1}{\sqrt{2\pi(k-1)-1}}$. Recall Stirling's formula: For any $n > 0$ there is a $\vartheta \in]0, 1[$ such that $n! = \left(\frac{n}{e}\right)^n \sqrt{2\pi n} e^{\frac{1}{12n+\vartheta}}$, see Robbins (1955). We thus have $\frac{1}{\sigma} + 1 = \sqrt{2\pi(k-1)} < \frac{e^{k-1}(k-1)!}{(k-1)^{k-1}}$ as required. Further, we define the value τ_{split} where we split between the low and the high case:

$$\tau_{\text{split}} := \begin{cases} 2 \ln \frac{k}{e} & \text{if } k \geq 8, \\ 2 & \text{if } 5 \leq k \leq 7, \\ \sqrt{7} - 1 & \text{if } k = 4. \end{cases}$$

We start with the treatment of the cases $k \geq 8$.

We claim that for $\tau \leq \tau_{\text{split}}$ we are in the low case, i.e.

$$(6.4.14) \quad \frac{1 - e^{-\tau}}{\tau} \geq \sqrt[k]{\frac{1+\sigma}{k!}} =: \vartheta.$$

Figure 6.4.3: Case coverage and τ_{split}

Consider $f_2(\tau) = 1 - e^{-\tau} - \vartheta\tau$. It obviously vanishes at $\tau = 0$. The derivative of f_2 shows that there is exactly one maximum at $\tau = -\ln(\vartheta)$, which is roughly $\ln k$. To prove (6.4.14) for $\tau \in]0, \tau_{\text{split}}]$ it is thus sufficient to prove that f_2 is at least 0 at the right boundary. First, note that by Stirling's formula we have $k! \geq \left(\frac{k}{e}\right)^k \sqrt{2\pi}$ and so we can estimate ϑ by $\frac{e}{k}$:

$$\vartheta = \sqrt[k]{\frac{1+\sigma}{k!}} \leq \frac{e}{k} \sqrt[k]{\frac{2}{\sqrt{2\pi}}} \leq \frac{e}{k}.$$

Well, now we have

$$\begin{aligned} k \cdot f_2(\tau_{\text{split}}) &= k - \frac{e^2}{k} - k\vartheta\tau_{\text{split}} \\ &\geq k - \frac{e^2}{k} - 2e \ln \frac{k}{e} =: f_3(k). \end{aligned}$$

Checking that $f_3(e) = 0$ and the derivative $f_3'(k) = (1 - \frac{e}{k})^2$ is positive for $k > e$ shows that $f_3(k) > 0$ for all $k \geq 3$. Thus for $\tau \leq \tau_{\text{split}}$ we are in the low case with the above choice of σ .

It remains to check that for $\tau \geq \tau_{\text{split}}$ we are in the high case. We use the Lagrange remainder estimate of the power series of the exponential function and again Stirling's formula to obtain

$$\sum_{0 \leq \ell \leq k-2} \frac{\tau^\ell}{\ell!} \geq e^\tau \left(1 - \frac{\tau^{k-1}}{(k-1)!}\right) \geq e^\tau \left(1 - \left(\frac{e\tau}{k-1}\right)^{k-1}\right) =: f_5(k, \tau).$$

As the left hand side is increasing in $\tau > 0$ we consider the smallest τ in

question: let $f_6(k) := f_5(k, \tau_{\text{split}})/k$. Therein,

$$\left(\frac{e\tau_{\text{split}}}{k-1}\right)^{k-1} = \exp\left(\underbrace{-(k-1)}_{\text{(I)}} \underbrace{\left(-\ln 2 - 1 - \ln \ln \frac{k}{e} + \ln(k-1)\right)}_{\text{(II)}}\right).$$

Observe that the term (I) is obviously positive and increasing for $k \geq 8$. The same is true for the term (II): its derivative $\frac{1}{k-1} - \frac{1}{k(\ln k-1)}$ is positive for $k \geq e^2 \approx 7.39$ and the value of term (II) at e^2 is positive. Using this we infer that f_6 is increasing and positive. Checking $f_6(10) > 1$ ($f_6(10) \in]1.19, 1.20[$) now proves $\sum_{0 \leq \ell \leq k-2} \frac{(2 \ln \frac{k}{e})^\ell}{\ell!} \geq k$ for $k \geq 10$. For $k = 8$ and $k = 9$ we just verify this inequality directly.

It remains to consider $4 \leq k \leq 7$. Here we use individual separation positions as defined above:

k	4	5	6	7	(8)	(9)	
τ_{split}	$\sqrt{7} - 1$	2	2	2	$2 \ln \frac{8}{e}$	$2 \ln \frac{9}{e}$	
$\frac{1 - e^{-\tau_{\text{split}}}}{\tau_{\text{split}}}$	0.49⊥	0.43⊥	0.43⊥	0.43⊥	0.40⊓	0.37⊓	∨ for low case
$\sqrt[k]{\frac{1+\sigma}{k!}}$	0.48∇	0.40∇	0.35∇	0.31∇	0.28⊔	0.25∇	
$\sum_{0 \leq \ell \leq k-2} \frac{\tau_{\text{split}}^\ell}{\ell!}$	4	6.33⊥	7.00⊥	7.27∇	8.60⊥	10.93∇	$\geq k$ for high case

Just check that for this τ_{split} the low and the high case conditions are both fulfilled. Summing up: the claim is proved for $k \geq 4$.

$k < 10$: For $k = 3$ we need an explicit check as the estimates done in the low and the high case are too sloppy. As the following actually is a general computational way to verify the inequality (6.4.13) we describe it in general, show computational results for $3 \leq k \leq 10$ and make the critical case $k = 3$ hand-checkable at the end. For the verification we use a small trick and brute force: First, we substitute occurrences of $e^{-\tau}$ with a new variable T . The task turns into showing that the bivariate polynomial

$$F_k(\tau, T) := \sum_{0 \leq \ell \leq k-2} \frac{\tau^\ell}{\ell!} - \frac{(1 - (1 - T)^k)}{T} \left(1 - \frac{\tau^{k-1}}{(k-1)!} T\right)$$

is non-negative at $\tau = -\ln T$ for $T \in]0, 1]$. Now, observe that F_k is increasing in $\tau > 0$ for fixed $T \in [0, 1]$. If we thus replace $-\ln T$ with a lower bound and we can show that the resulting term is still non-negative then

we are done. For $T \in]0, 1]$ we have $-\ln T \geq \sum_{1 \leq \ell \leq s} \frac{(1-T)^\ell}{\ell}$. This lower bound even converges to $-\ln T$, which actually ensures that we can always find some s that allows the following reduction. We consider the univariate polynomial

$$g_{k,s}(T) := F_k \left(\sum_{1 \leq \ell \leq s} \frac{(1-T)^\ell}{\ell}, T \right).$$

By our reasoning, the claim follows if $\forall T \in]0, 1] : g_{k,s}(T) \geq 0$ for some s . This in turn is implied by

$$(6.4.15) \quad g_{k,s}(0) > 0 \quad \wedge \quad g_{k,s}(T) \text{ has no zero for } 0 < T < 1.$$

The second statement can be checked using Sturm's theorem (Sturm 1835) by only evaluating certain rational polynomials at $T = 0$ and $T = 1$. However, this only works if s is chosen large enough. We have determined the smallest s that make (6.4.15) true:

k	3	4	5	6	7	(8)	(9)
s	4	3	4	5	6	7	8
$\deg g_{k,s}$	11	13	21	31	43	57	73
time (sec)	0.19	0.29	0.60	2.8	24	235	1704

Though we can always divide out $(1-T)^k$ from $g_{k,s}$ the degrees are in all cases quite high and the computations better done by a computer. The timings refer to our own (non-optimized) MuPad-program used to assert (6.4.15). As $k = 3$ is the only case that we do not cover otherwise we give $g_{3,4}$ here:

$$g_{3,4}(T)/(1-T)^3 = \frac{1}{12}(1-T) + \frac{5}{6}T + T(1-T) \left(\frac{481}{96}(1-T)^2 + \frac{35}{24}T^2 + T(1-T) \left(\frac{245}{96}(1-T) + \frac{103}{24}T + T(1-T) \left(\frac{119}{144}(1-T)^2 + \frac{89}{144}T^2 + T(1-T) \cdot \frac{407}{288} \right) \right) \right)$$

With this description we can easily see that it is positive on $[0, 1[$. \square

Finally, we put together the upper bound on $\tilde{\lambda}_{\text{norm}}^k$, Lemma 6.4.5, Lemma 6.4.7, and Lemma 6.4.12 in the following theorem.

THEOREM 6.4.16. For any $k \geq 1$ and $C = B^{1+\alpha}$ we have

$$\tilde{\lambda}_{\text{norm}}^k \langle \xi \rangle \leq \tilde{c}_k := \begin{cases} \frac{1}{\alpha \ln B} & \text{if } k > 0, \\ 1 & \text{if } k = 0. \end{cases}$$

Assume $\alpha \ln B \geq \max\left(\ln 2, \frac{\ln 16}{k}\right)$. Then for any $\varepsilon \in]0, 1]$ with $\varepsilon \leq k - \xi_{\frac{1}{2}}^k$ or $k < 3$ there is a $\delta_{\tilde{\lambda}^k} > 0$ such that for $\xi \in [\varepsilon, k - \varepsilon]$

$$\tilde{\lambda}_{\text{norm}}^k \langle \xi \rangle \geq \frac{\delta_{\tilde{\lambda}^k}}{\alpha \ln B}.$$

Here we can choose

$$\delta_{\tilde{\lambda}^k} = \min \left(2^{-4} \cdot \frac{\varepsilon^k}{k!}, 2^{-k} \cdot \frac{\varepsilon^{k-1}}{(k-1)!} \right).$$

Note that $\xi_{\frac{1}{2}}^k$ is the maximum of $\tilde{\lambda}_{\text{norm}}^k$ which in our experiments is always less than $k - 1$ for $k \geq 3$. However, proving that would result in a stronger version of Lemma 6.4.12, which even in the given form required quite some effort. However, we just want that to work for some small ε and that is always granted.

PROOF. The upper bound being proven, we consider the lower bound. First, keeping in mind that $\alpha \ln B \geq \ln 2$ and $\alpha \ln B \geq \frac{\ln 16}{k}$, we consider the cases $k \geq 3$. We know by Lemma 6.4.5 that $\tilde{\lambda}_{\text{norm}}^k$ on $[\varepsilon, k - \varepsilon]$ attains its minimal value at one of the boundaries since $\xi_{\frac{1}{2}}^k \in [\varepsilon, k - \varepsilon]$ (by assumption). We thus only need to consider its values at $\xi = \varepsilon$ and at $\xi = k - \varepsilon$.

On the left hand side Lemma 6.4.7 gives

$$\alpha \ln B \cdot \tilde{\lambda}_{\text{norm}}^k \langle \varepsilon \rangle \geq \exp\left(-\frac{\ln^2 4}{\alpha \ln B}\right) \frac{\varepsilon^k}{k!} \geq \exp\left(-\frac{\ln^2 4}{\ln 2}\right) \frac{\varepsilon^k}{k!} = 2^{-4} \cdot \frac{\varepsilon^k}{k!}$$

using the conditions on $\alpha \ln B$.

On the right hand side by Lemma 6.4.12 we find

$$\alpha \ln B \cdot \tilde{\lambda}_{\text{norm}}^k \langle k - \varepsilon \rangle \geq (1 - \exp(-\alpha \ln B))^k \frac{\varepsilon^{k-1}}{(k-1)!} \geq 2^{-k} \frac{\varepsilon^{k-1}}{(k-1)!}$$

using again $\alpha \ln B \geq \ln 2$. This completes the proof for the cases $k \geq 3$.

We will now show corresponding lower bounds for the cases $k = 1, 2$. For $k = 1$ we have for $\varepsilon \in]0, 1]$ using $\alpha \ln B \geq \ln 2$:

$$\alpha \ln B \cdot \tilde{\lambda}_{\text{norm}}^1 \langle \varepsilon \rangle = 1 - \exp(-\varepsilon \alpha \ln B) \geq 1 - \exp(-\varepsilon \ln 2).$$

Using Fact 6.4.9 and $\varepsilon \leq 1$, we have

$$1 - \exp(-\varepsilon \ln 2) \geq \frac{1 - \exp(-\ln 2)}{\ln 2} \varepsilon = \frac{1}{2 \ln 2} \varepsilon \geq \frac{1}{2} \varepsilon.$$

For $k = 2$ we again apply Lemma 6.4.7 for the left hand side as in the general case. For the right hand side we show that

$$\alpha^2 \ln^2 B \cdot \tilde{\lambda}_{\text{norm}}^2 \langle 2 - \varepsilon \rangle \geq \left(1 - \frac{1}{2 \ln 2}\right) \varepsilon \alpha \ln B$$

for $0 \leq \varepsilon \alpha \ln B \leq \alpha \ln B$, $\ln 2 \leq \alpha \ln B$. Since $\left(1 - \frac{1}{2 \ln 2}\right) \geq 2^{-2}$ this proves the claim. Now, using Theorem 6.4.4 we write the left hand side minus the right hand side as $f_5(\varepsilon \alpha \ln B, \alpha \ln B)$ with $f_5(\tau, \vartheta) = \exp(\tau - 2\vartheta) - 2 \exp(\tau - \vartheta) + \frac{1}{2 \ln 2} \tau + 1$. We have to show that f_5 is non-negative if $0 \leq \tau \leq \vartheta$ and $\vartheta \geq \ln 2$. The ϑ -derivative of f_5 ,

$$\frac{\partial f_5}{\partial \vartheta}(\tau, \vartheta) = 2 \exp(-\vartheta) (1 - \exp(-\vartheta)) \exp(\tau),$$

is positive for $\vartheta > 0$. Thus it suffices to show that $f_5(\tau, \vartheta) \geq 0$ for the smallest allowed ϑ , which is the larger of τ and $\ln 2$. If $\tau \geq \ln 2$ then we consider $f_5(\tau, \tau) = \exp(-\tau) + \frac{1}{2 \ln 2} \tau - 1$. This expression is increasing in this case (even for $\tau \geq \ln 2 + \ln \ln 2$) and so it is greater than or equal to $f_5(\ln 2, \ln 2) = 0$. If otherwise $0 \leq \tau \leq \ln 2$ then we consider $f_5(\tau, \ln 2) = -\frac{3}{4} \exp(\tau) + \frac{1}{2 \ln 2} \tau + 1$ which is decreasing even for $\tau \geq 0$ and so it is greater than or equal to $f_5(\ln 2, \ln 2) = 0$. \square

Summing up we obtain:

COROLLARY 6.4.17. *For any $\varepsilon \in]0, 1]$ and any $k \geq 1$ we have uniformly for $\xi \in [\varepsilon, k - \varepsilon]$*

$$\tilde{\lambda}_{\text{norm}}^k \langle \xi \rangle \in \Theta \left(\frac{1}{\alpha \ln B} \right). \quad \square$$

6.5. Estimating the estimate $\widehat{\lambda}^k$

The recurrence Lemma 6.3.6 for $\widehat{\lambda}^k$ is more complex than the one for $\tilde{\lambda}^k$, so instead of solving it we estimate it. We consider also here the normed version $\widehat{\lambda}_{\text{norm}}^k \langle \xi \rangle := \frac{\widehat{\lambda}^k \langle \xi \rangle}{\alpha^k B^{k+\xi\alpha}}$. To better understand how the error behaves we compute it for $k = 1$:

$$\widehat{\lambda}_{\text{norm}}^1 \langle \xi \rangle = \begin{cases} 0 & \text{if } \xi \in]-\infty, 0[, \\ \frac{1+\xi\alpha}{8\pi\alpha} \frac{\ln B}{B^{\frac{1}{2}+\frac{\xi\alpha}{2}}} + \frac{1}{8\pi\alpha} \frac{\ln B}{B^{\frac{1}{2}+\xi\alpha}} & \text{if } \xi \in [0, 1[, \\ \frac{1+\alpha}{8\pi\alpha} \frac{\ln B}{B^{\frac{1}{2}-\frac{\alpha}{2}+\xi\alpha}} + \frac{1}{8\pi\alpha} \frac{\ln B}{B^{\frac{1}{2}+\xi\alpha}} & \text{if } \xi \in [1, \infty[. \end{cases}$$

From this we estimate $\widehat{\lambda}_{\text{norm}}^1$ directly:

$$\widehat{\lambda}_{\text{norm}}^1 \langle \xi \rangle \leq \begin{cases} 0 & \text{if } \xi \in]-\infty, 0[, \\ \frac{(2+\alpha) \ln B}{8\pi\alpha} B^{-\frac{1+\xi\alpha}{2}} & \text{if } \xi \in [0, 1[, \\ \frac{(1+\alpha) \ln B}{8\pi\alpha} B^{-\frac{1}{2} + \frac{\alpha}{2} - \xi\alpha} + \frac{\ln B}{8\pi\alpha} \cdot B^{-\frac{1}{2} - \xi\alpha} & \text{if } \xi \in [1, \infty[. \end{cases}$$

We have also looked at precise expressions for larger k , yet they are huge and do not give rise to better bounds.

THEOREM 6.5.1. *Define values \widehat{c}_k recursively by*

$$\widehat{c}_k := \widehat{c}_{k-1} + \frac{4 + 3 \ln B}{8\pi\alpha\sqrt{B}} (\widetilde{c}_{k-1} + \widehat{c}_{k-1})$$

for $k \geq 3$ based on $\widehat{c}_0 := 0$, $\widehat{c}_1 := \frac{(2+\alpha) \ln B}{8\pi\alpha\sqrt{B}}$, and

$$\begin{aligned} \widehat{c}_2 &:= \frac{6 + 3\alpha}{8\pi\alpha^2} \frac{1}{\sqrt{B}} + \frac{4 + 3 \ln B}{8\pi\alpha\sqrt{B}} (\widetilde{c}_1 + \widehat{c}_1) \\ &= \frac{9 + 3\alpha}{8\pi\alpha^2} \frac{1}{\sqrt{B}} + \frac{1}{2\pi\alpha^2\sqrt{B} \ln B} + \frac{(2 + \alpha)(4 + 3 \ln B) \ln B}{64\pi^2\alpha^2 B}. \end{aligned}$$

Then for any k and $\xi \in \mathbb{R}$ we have

$$\widehat{\lambda}_{\text{norm}}^k \langle \xi \rangle \leq \widehat{c}_k.$$

If $\alpha \geq \frac{\ln B}{\sqrt{B}}$ we have for $k \geq 2$ and large B the inequality

$$\widehat{c}_k \leq \frac{(2^k - 1)(1 + \alpha)}{\alpha^2\sqrt{B}}.$$

For $k = 1$ the order of \widehat{c}_1 is necessarily slightly larger. More precisely, we have for large B that

$$\widehat{c}_1 \leq \frac{(1 + \alpha) \ln B}{\alpha\sqrt{B}}.$$

Instead of defining \widehat{c}_2 and \widehat{c}_1 we could have left that to the recursion. But the given values are smaller than the ones derived from the recursion based on \widehat{c}_0 only. For $k = 1$ the recursion would give $\frac{4+3\ln B}{8\pi\alpha\sqrt{B}}$. For $k \geq 2$ however the improvement due to these explicit settings is a factor of order $\ln B$.

PROOF. We first show that the value \widehat{c}_k is bounded as claimed. For $k = 1$ the claim follows directly from the definition, since we have for $B > \exp(1)$

that $(2 + \alpha) \ln B \leq 2(1 + \alpha) \ln B$ as α is positive. For $k = 2$ we have for $\ln^2 B \leq \sqrt{B}$ the inequality

$$\begin{aligned} \widehat{c}_2 &= \frac{6 + 3\alpha}{8\pi\alpha^2} \frac{1}{\sqrt{B}} + \frac{4 + 3 \ln B}{8\pi\alpha\sqrt{B}} (\widetilde{c}_1 + \widehat{c}_1) \\ &\leq \frac{1 + \alpha}{\alpha^2\sqrt{B}} + \frac{1}{\alpha^2\sqrt{B}} + \frac{(1 + \alpha) \ln^2 B}{\alpha^2 B} \\ &\leq \frac{3(1 + \alpha)}{\alpha^2\sqrt{B}}. \end{aligned}$$

For $k \geq 2$ we proceed inductively. We have

$$\begin{aligned} \widehat{c}_k &= \widehat{c}_{k-1} + \frac{4 + 3 \ln B}{8\pi\alpha\sqrt{B}} (\widetilde{c}_{k-1} + \widehat{c}_{k-1}) \\ &\leq \frac{(2^{k-1} - 1)(1 + \alpha)}{\alpha^2\sqrt{B}} + \frac{\ln B}{\alpha\sqrt{B}} \left(\frac{1}{\alpha \ln B} + \frac{(2^{k-1} - 1)(1 + \alpha)}{\alpha \ln B} \right) \\ &= \frac{(2^k - 1)(1 + \alpha)}{\alpha^2\sqrt{B}}. \end{aligned}$$

Now we prove the remaining estimate by induction on k . The case $k = 0$ is true by definition of $\widehat{\lambda}^0$ (with equality). For $k = 1$ the inspection above proves the claim. The explicit calculation of $\widehat{\lambda}^1$ also shows that a bound of order $\mathcal{O}(x/\sqrt{B})$ is impossible. We defer the case $k = 2$ to the end of the proof as most of it will be as in the general case. So assume $k \geq 3$. Using the definition of $\widehat{\lambda}^k$ from Lemma 6.3.6 we split $\widehat{\lambda}^k$ into three summands:

$$\begin{aligned} \widehat{\lambda}_{\text{norm}}^k(\xi) &= \int_0^1 \widehat{\lambda}_{\text{norm}}^{k-1}(\xi - \varrho) \, d\varrho \\ &\quad + \frac{2\widehat{E}(B)}{\alpha B} \cdot (\widetilde{\lambda}_{\text{norm}}^{k-1} + \widehat{\lambda}_{\text{norm}}^{k-1})(\xi) \\ &\quad + \int_0^1 (\widetilde{\lambda}_{\text{norm}}^{k-1} + \widehat{\lambda}_{\text{norm}}^{k-1})(\xi - \varrho) \widehat{E}'(B^{1+\varrho\alpha}) \ln B \, d\varrho. \end{aligned}$$

For $\widehat{E}(x) = \frac{1}{8\pi} \sqrt{x} \ln x$ we calculate as a preparative

$$(6.5.2) \quad \frac{2\widehat{E}(B)}{\alpha B} = \frac{\ln B}{4\pi\alpha\sqrt{B}},$$

$$(6.5.3) \quad \int_0^1 \widehat{E}'(B^{1+\varrho\alpha}) \ln B \, d\varrho = \frac{4 + \ln B}{8\pi\alpha\sqrt{B}} - \frac{4 + \ln C}{8\pi\alpha\sqrt{C}}.$$

The first summand of $\widehat{\lambda}_{\text{norm}}^k$ is at most \widehat{c}_{k-1} by induction hypothesis. The second summand we estimate using (6.5.2) by

$$\frac{\ln B}{4\pi\alpha\sqrt{B}} (\widetilde{c}_{k-1} + \widehat{c}_{k-1}).$$

The third summand is bounded by

$$(\widetilde{c}_{k-1} + \widehat{c}_{k-1}) \int_0^1 \widehat{E}'(B^{1+\varrho\alpha}) \ln B \, d\varrho.$$

By (6.5.3) the third summand is at most

$$\frac{4 + \ln B}{8\pi\alpha\sqrt{B}} (\widetilde{c}_{k-1} + \widehat{c}_{k-1}).$$

This completes the proof of the case $k \geq 3$:

$$\begin{aligned} \widehat{\lambda}_{\text{norm}}^k \langle \xi \rangle &\leq \widehat{c}_{k-1} + \frac{\ln B}{4\pi\alpha\sqrt{B}} (\widetilde{c}_{k-1} + \widehat{c}_{k-1}) + \frac{4 + \ln B}{8\pi\alpha\sqrt{B}} (\widetilde{c}_{k-1} + \widehat{c}_{k-1}) \\ &= \widehat{c}_k. \end{aligned}$$

For $k = 2$ we have to improve the estimate of the first summand only, since the other terms anyways are of order at most $\mathcal{O}(1/\sqrt{B})$. For $\xi < 0$ there is nothing to prove. For $\xi \in [0, 1[$ we find for this first summand

$$\begin{aligned} \int_0^\xi \widehat{\lambda}_{\text{norm}}^1 \langle \xi - \varrho \rangle \, d\varrho &\leq \int_0^\xi \frac{(2 + \alpha) \ln B}{8\pi\alpha} B^{-\frac{1}{2} - \frac{(\xi - \varrho)\alpha}{2}} \, d\varrho \\ &= \frac{(2 + \alpha) \ln B}{8\pi\alpha} B^{-\frac{1}{2}} \underbrace{B^{-\frac{\xi\alpha}{2}} \int_0^\xi B^{\frac{\alpha\varrho}{2}} \, d\varrho}_{= \frac{2}{\alpha \ln B} \left(1 - B^{-\frac{\xi\alpha}{2}}\right)} \\ &\leq \frac{2 + \alpha}{4\pi\alpha^2\sqrt{B}}. \end{aligned}$$

For $\xi \in [1, 2]$ we find

$$\begin{aligned} \int_0^1 \widehat{\lambda}_{\text{norm}}^1 \langle \xi - \varrho \rangle \, d\varrho &\leq \int_{\xi-1}^1 \frac{(2 + \alpha) \ln B}{8\pi\alpha} B^{-\frac{1}{2} - \frac{(\xi - \varrho)\alpha}{2}} \, d\varrho \\ &\quad + \int_0^{\xi-1} \widehat{\lambda}_{\text{norm}}^1 \langle 1 \rangle B^{1+\alpha} B^{-(1+(\xi-\varrho)\alpha)} \, d\varrho \end{aligned}$$

$$\begin{aligned}
&= \frac{(2 + \alpha) \ln B}{8\pi\alpha} B^{-\frac{1}{2} - \frac{(\xi-1)\alpha}{2}} \underbrace{B^{-\frac{\alpha}{2}} \int_{\xi-1}^1 B^{\frac{\alpha\varrho}{2}} d\varrho}_{= \frac{2}{\alpha \ln B} \left(1 - B^{-\frac{(2-\xi)\alpha}{2}}\right)} \\
&\quad + \widehat{\lambda}_{\text{norm}}^1 \langle 1 \rangle \underbrace{B^{-(\xi-1)\alpha} \int_0^{\xi-1} B^{\alpha\varrho} d\varrho}_{= \frac{1}{\alpha \ln B} (1 - B^{-(\xi-1)\alpha})} \\
&\leq \frac{2 + \alpha}{4\pi\alpha^2} B^{-\frac{1}{2} - \frac{(\xi-1)\alpha}{2}} + \frac{1 + \alpha}{8\pi\alpha^2} B^{-\frac{1}{2} - \frac{\alpha}{2}} + \frac{1}{8\pi\alpha^2} B^{-\frac{1}{2} - \alpha} \\
&\leq \frac{6 + 3\alpha}{8\pi\alpha^2 \sqrt{B}}.
\end{aligned}$$

As $\widehat{\lambda}_{\text{norm}}^1$ decreases for $\xi \geq 1$ this bound also holds for $\xi \geq 2$. Putting everything together the above defined value \widehat{c}_2 bounds $\widehat{\lambda}_{\text{norm}}^2 \langle \xi \rangle$ as claimed. \square

It is tempting to guess that we can save more $\ln B$ factors for larger k . However, inspecting $\widehat{\lambda}_{\text{norm}}^2$ shows that, say, $\widehat{\lambda}_{\text{norm}}^3 \langle \frac{1}{2} \rangle \in \Omega\left(\frac{1}{\sqrt{B}}\right)$. (For $\xi \in [0, 1[$ we find $\widehat{\lambda}_{\text{norm}}^2 \langle \xi \rangle = \frac{3}{4\pi\alpha\sqrt{B}} + \mathcal{O}\left(\frac{1}{\sqrt{B \ln B}}\right)$.)

6.6. Reestimating $\widehat{\lambda}^k$ without Riemann

If you do not want to assume the Riemann Hypothesis 2.2.14 then only weaker bounds $\widehat{E}(x)$ on $|\pi(x) - \text{Li}(x)|$ can be used. In Ford (2002a) and Ford (2002b) we found the following explicit bounds, the first one he attributes to a paper by Y. Cheng which we could not find.

FACT 6.6.1.

- For $x > 10$ we have

$$|\pi(x) - \text{Li}(x)| \leq 11.88 x (\ln x)^{\frac{3}{5}} \exp\left(-\frac{1}{57} (\ln x)^{\frac{3}{5}} (\ln \ln x)^{-\frac{1}{5}}\right).$$

- There is a constant C and a frontier x_0 such that for $x > x_0$ we have

$$|\pi(x) - \text{Li}(x)| \leq C x \exp\left(-0.2098 (\ln x)^{\frac{3}{5}} (\ln \ln x)^{-\frac{1}{5}}\right).$$

Admittedly, these bounds only start to be meaningful at large values of x (eg. the first statement around 10^{159299}). All those bounds are of the form:

For all $x > x_0$

$$|\pi(x) - \text{Li}(x)| \leq \underbrace{C x (\ln x)^{c_0} \exp(-A (\ln x)^{c_1} (\ln \ln x)^{-c_2})}_{=:\widehat{E}(x)}$$

holds. Here, $C > 0$, $x_0 > 0$, $c_0 \in \mathbb{R}$, $c_1 > 0$, $c_2 > 0$ and $A > 0$ are given parameters (which are not always known). Note that we have that

$$\widehat{E}(x)/x$$

is decreasing for large x . Actually, with the parameter sets from Fact 6.6.1 this is already true for $x \geq 5$. Moreover, the quotient of the relative errors at $x^{1+\alpha}$ and at x

$$\frac{\widehat{E}(x^{1+\alpha}) \frac{\ln x^{1+\alpha}}{x^{1+\alpha}}}{\widehat{E}(x) \frac{\ln x}{x}} = \frac{(1+\alpha)\widehat{E}(x^{1+\alpha})}{x^\alpha \widehat{E}(x)}$$

is bounded (or even tends to zero) with $x \rightarrow \infty$ for any $\alpha > 0$. This follows from $\widehat{E}(x)/x$ decreasing when α is constant, but you may also consider values for α that increase when x grows.

Revisiting the proof of Theorem 6.5.1 shows that only (6.5.2), (6.5.3), and the initial values \widehat{c}_1 and \widehat{c}_2 depend on the specific bound \widehat{E} . We now use the following recursion for the bounds:

$$\widehat{c}_k := \widehat{c}_{k-1} + \underbrace{\left(\frac{2\widehat{E}(B)}{\alpha B} + \int_0^1 \widehat{E}'(B^{1+\varrho\alpha}) \ln B \, d\varrho \right)}_{=:u} (\widehat{c}_{k-1} + \widehat{c}_{k-1})$$

for $k \geq 1$ based on $\widehat{c}_0 = 0$, and possibly values for \widehat{c}_1 and \widehat{c}_2 .

To bound u tightly the trickiest step is bounding the integral. As our interests lie elsewhere we take the easy way out. We integrate by parts and use that $\widehat{E}(x)/x$ is decreasing for the following rough estimate

$$\int_0^1 \widehat{E}'(B^{1+\varrho\alpha}) \ln B \, d\varrho = \frac{\widehat{E}(B^{1+\alpha})}{\alpha B^{1+\alpha}} - \frac{\widehat{E}(B)}{\alpha B} + \ln B \underbrace{\int_0^1 \frac{\widehat{E}(B^{1+\varrho\alpha})}{B^{1+\varrho\alpha}} \, d\varrho}_{\leq \frac{\widehat{E}(B)}{B}}$$

Thus u is bounded by

$$u \leq \left(1 + \frac{1}{\alpha \ln B} \left(1 + \underbrace{\frac{\widehat{E}(B^{1+\alpha})}{B^\alpha \widehat{E}(B)}}_{\text{bounded}} \right) \right) \frac{\widehat{E}(B) \ln B}{B}.$$

In the following we neglect the bounded term, as we can compensate its effect for example by a small additional factor. Since u is small for large B , we expect \widehat{c}_k to be dominated by $\widehat{c}_1 = u$. Precisely, for $k > 1$ we have $\widehat{c}_k = (1 + u)\widehat{c}_{k-1} + \frac{u}{\alpha \ln B}$, thus

$$\widehat{c}_k = (1 + u)^{k-1}u + \frac{(1 + u)^{k-1} - 1}{\alpha \ln B} \sim \left(1 + \frac{k-1}{\alpha \ln B}\right)u.$$

THEOREM 6.6.2 (Nüsken 2006-2011). *Assume that $\widehat{E}(x)$ bounds $|\pi(x) - \text{Li}(x)|$ and $\widehat{E}(x)/x$ is decreasing for $x > x_0$ and the relative error decreases fast, i.e.*

$$\frac{\widehat{E}(x^{1+\alpha}) \frac{\ln x^{1+\alpha}}{x^{1+\alpha}}}{\widehat{E}(x) \frac{\ln x}{x}} = \frac{(1 + \alpha)\widehat{E}(x^{1+\alpha})}{x^\alpha \widehat{E}(x)}$$

is bounded under the chosen behavior of α . Then for any $k \geq 2$ and B large we have

$$\widehat{\lambda}_{\text{norm}}^k \langle \xi \rangle \in \mathcal{O} \left(\left(1 + \frac{k-1}{\alpha \ln B}\right) \left(1 + \frac{1}{\alpha \ln B}\right) \frac{\widehat{E}(B) \ln B}{B} \right)$$

for $k \geq 2$. □

This is close to optimal, we only loose a factor of order $\ln B$ in the relative error compared to the used error bound in the prime number theorem:

$$\frac{\widehat{\lambda}_{\text{norm}}^k \langle \xi \rangle}{\widetilde{\lambda}_{\text{norm}}^k \langle \xi \rangle} \leq \frac{\left(1 + \frac{k-1}{\alpha \ln B}\right) \left(1 + \frac{1}{\alpha \ln B}\right) \frac{\widehat{E}(B) \ln B}{B}}{\frac{\delta_{\widetilde{\lambda}^k}}{\alpha \ln B}} \sim \frac{\alpha}{\delta_{\widetilde{\lambda}^k}} \ln B \cdot \frac{\widehat{E}(B) \ln B}{B}.$$

The assumptions on \widehat{E} also hold for explicit error bounds with $\widehat{E}(x) \in \mathcal{O}\left(\frac{x}{\ln^\ell x}\right)$. Due to the lost $\ln B$ the result is only meaningful if $\ell \geq 3$, so Rosser & Schoenfeld (1962) does not suffice. From Dusart (1998) we can use $\widehat{E}(x) = 2.3854 \frac{x}{\ln^3 x}$ for $x > 355\,991$, and obtain

$$\widehat{\lambda}_{\text{norm}}^k \langle \xi \rangle \in \mathcal{O} \left(\frac{1}{\ln B} \right).$$

6.7. Improvements

We have a look at the quality of Theorem 6.3.5 when applied to our inspiring application. There $B = 1100 \cdot 10^6$, $C = 2^{37} - 1$, $\alpha = \ln(C)/\ln(B) - 1 \approx 0.2327$, and the largest k of interest is $k = 4$. For these parameters we find

$$\left[\frac{1}{(1 + \alpha)^k}, 1 \right] \subset [0.434, 1].$$

That is a great loss when we try to enclose the function of interest in a small interval. Actually, we can improve the theorem for the price of a slightly more complicated recursion. The present result was based on approximating $\dot{\kappa}(\varrho) = \frac{1}{\ln p}$ for $p = B^{1+\varrho\alpha} \in]B, C]$ by $\dot{\lambda}(\varrho) = \frac{1}{\ln B}$ in the recursion of Definition 6.3.1:

$$\tilde{\kappa}_{B,C}^k \langle \xi \rangle = \alpha \ln B \int_B^C \tilde{\kappa}_{B,C}^{k-1} \langle \xi - \varrho \rangle \cdot \dot{\kappa}(\varrho) B^{1+\varrho\alpha} d\varrho.$$

We get a better bound by using

$$\dot{\nu}(\varrho) = \frac{1-\varrho}{\ln B} + \frac{\varrho}{\ln C}$$

with $p = B^{1+\varrho\alpha}$ instead.

DEFINITION 6.7.1. For $x \geq 0$ we let $\tilde{\nu}^0 := \kappa_{B,C}^0$, and recursively for $k > 0$

$$\tilde{\nu}^k \langle \xi \rangle := \alpha \ln B \int_0^1 \tilde{\nu}^{k-1} \langle \xi - \varrho \rangle \left(\frac{1-\varrho}{\ln B} + \frac{\varrho}{\ln C} \right) B^{1+\varrho\alpha} d\varrho.$$

THEOREM 6.7.2 (Nüsken 2006-2011). Write $C = B^{1+\alpha}$ and fix $k \in \mathbb{N}_{>0}$. Then for $x \in \mathbb{R}_{>0}$ we have

$$\tilde{\kappa}_{B,C}^k(x) \in \left[\left(\frac{1+\alpha}{\left(1+\frac{\alpha}{2}\right)^2} \right)^k, 1 \right] \tilde{\nu}^k(x). \quad \square$$

In the light of the inspiring application we now find

$$\left[\left(\frac{1+\alpha}{\left(1+\frac{\alpha}{2}\right)^2} \right)^k, 1 \right] \subset [0.957, 1].$$

When we started to think about solving the recursion in Definition 6.7.1 our first trial was to reuse the polynomial hills \tilde{m}^k . Yet, that didn't want to fit nicely. Instead we learned from our calculations that an exponential density instead of a linear one would be easier to connect to the polynomial hills. So we tried to approximate like this and got

$$\dot{\kappa}(\varrho) = \frac{1}{\ln p} = \frac{1}{(1-\varrho)\ln B + \varrho\ln C} \approx e^{-\ln \ln B - \varrho \ln(1+\alpha)} =: \dot{\eta}(\varrho).$$

The exponent in $\dot{\eta}$ is chosen such that for $\varrho = 0$ and $\varrho = 1$ we have equality. It turns out that this approximation is even better than the one before and at the same time easier to handle. Thus we replace the functions $\tilde{\nu}^k$ with another family $\tilde{\eta}^k$:

DEFINITION 6.7.3. For $\xi \in \mathbb{R}$ we let $\tilde{\eta}^0 := \kappa_{B,C}^0$, and recursively for $k > 0$

$$\tilde{\eta}^k \langle \xi \rangle := \alpha \ln B \int_0^1 \tilde{\eta}^{k-1} \langle \xi - \varrho \rangle \underbrace{\frac{(1+\alpha)^{-\varrho}}{\ln B}}_{=\hat{\eta}(\varrho)} B^{1+\varrho\alpha} d\varrho.$$

THEOREM 6.7.4 (Nüsken 2006-2011). Write $C = B^{1+\alpha}$ and fix $k \in \mathbb{N}_{>0}$. Then for $x \in \mathbb{R}_{>0}$ we have

$$\tilde{\kappa}_{B,C}^k(x) \in \left[\left(\frac{\ln(1+\alpha)}{\alpha} (1+\alpha)^{\frac{1}{\ln(1+\alpha)} - \frac{1}{\alpha}} \right)^k, 1 \right] \tilde{\eta}^k(x).$$

PROOF. To prove this we have to relate the function $\tilde{\kappa}: [0, 1] \rightarrow \mathbb{R}_{>0}$, $\varrho \mapsto 1/\ln(B^{1+\varrho\alpha})$ occurring in the definition of $\tilde{\kappa}_{B,C}^k$ to the function $\hat{\eta}: [0, 1] \rightarrow \mathbb{R}_{>0}$, $\varrho \mapsto \hat{\eta}(\varrho)$ replacing it in the definition of $\tilde{\eta}^k$. Routine calculus shows that the function $\tilde{\kappa}/\hat{\eta}$ is at most 1, namely at $\varrho = 0$ and $\varrho = 1$, and assumes its minimum value $\frac{\ln(1+\alpha)}{\alpha} (1+\alpha)^{\frac{1}{\ln(1+\alpha)} - \frac{1}{\alpha}}$ at $\varrho = \frac{\alpha - \ln(1+\alpha)}{\alpha + \ln(1+\alpha)}$. \square

Testing this with the parameters $\alpha \approx 0.232\mathbb{T}$ and $k = 4$ from our inspiring application we obtain

$$\left[\left(\frac{\ln(1+\alpha)}{\alpha} (1+\alpha)^{\frac{1}{\ln(1+\alpha)} - \frac{1}{\alpha}} \right)^k, 1 \right] \subset [0.978, 1].$$

To get a better impression we have plotted the lower interval boundary as a function of α for all three cases in Figure 6.7.1. We see that for small values of α we obtain good approximations of $\tilde{\kappa}_{B,C}$ and all our attempts give only weak results for large α , but the one with $\hat{\eta}$ is always best.

If we now rewrite the recursion to one for

$$\tilde{\eta}_{\text{norm}}^k \langle \xi \rangle = \frac{\tilde{\eta}^k \langle \xi \rangle}{\alpha^k B^{k+\xi\alpha} (1+\alpha)^{-\xi}} = \frac{\tilde{\eta}^k \langle \xi \rangle}{\alpha^k B^{k+\xi} \left(\alpha - \frac{\ln(1+\alpha)}{\ln B} \right)}$$

we find that $\tilde{\eta}_{\text{norm}}^k = \mathcal{M}\tilde{\eta}_{\text{norm}}^{k-1}$. So we'll obtain the solution from the polynomial hills as in Theorem 6.4.4 for $\tilde{\lambda}^k$:

$$\tilde{\eta}_{\text{norm}}^k \langle \xi \rangle = \int_0^\infty e^{-\varrho(\alpha \ln B - \ln(1+\alpha))} \tilde{m}^k(\xi - \varrho) d\varrho.$$

The only difference is that instead of α we have $\alpha - \frac{\ln(1+\alpha)}{\ln B}$. With this replacement Theorem 6.4.4 becomes:

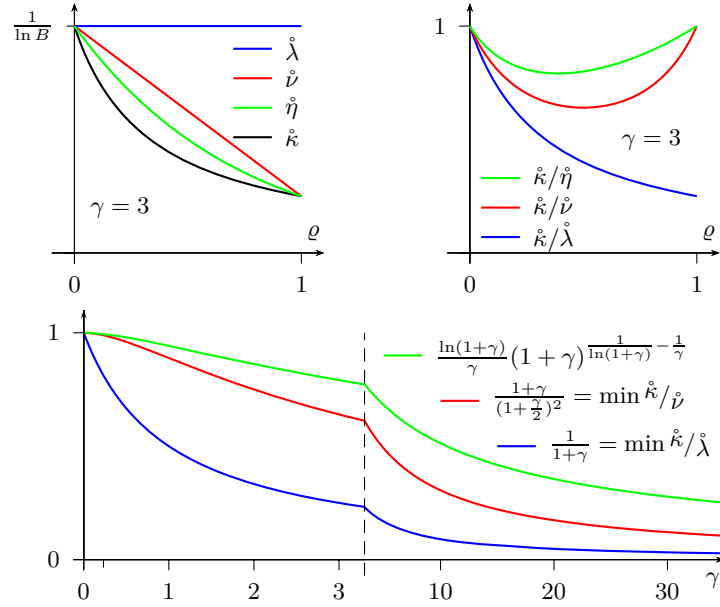


Figure 6.7.1: Comparing the quality of $\tilde{\lambda}$, $\tilde{\nu}$, $\tilde{\eta}$: the top pictures show the integral kernels and their ratios for a specific α , the lower pictures show the minimum of the ratios as a function of α

THEOREM 6.7.5 (Nüsken 2006-2011). For any $\xi \in \mathbb{R}$ we have

$$\begin{aligned}
 \tilde{\eta}_{\text{norm}}^k(\xi) &= \int_0^\xi B^{-\varrho(\alpha - \frac{\ln(1+\alpha)}{\ln B})} \tilde{m}^k(\xi - \varrho) d\varrho \\
 &= B^{-\xi(\alpha - \frac{\ln(1+\alpha)}{\ln B})} \int_0^\xi B^{\varrho(\alpha - \frac{\ln(1+\alpha)}{\ln B})} \tilde{m}^k(\varrho) d\varrho, \\
 &= \int_0^\xi \left(\frac{B^\alpha}{1+\alpha} \right)^{-\varrho} \tilde{m}^k(\xi - \varrho) d\varrho \\
 &= B^{-\xi(\alpha - \frac{\ln(1+\alpha)}{\ln B})} \int_0^\xi \underbrace{\left(\frac{B^\alpha}{1+\alpha} \right)^\varrho}_{= \frac{C/\ln C}{B/\ln B}} \tilde{m}^k(\varrho) d\varrho,
 \end{aligned}$$

$$\begin{aligned} \tilde{\eta}_{\text{norm}}^k \langle \xi \rangle &= \frac{1}{(-\alpha \ln B + \ln(1 + \alpha))^k} \\ &\quad \left(\sum_{0 \leq i \leq \lfloor \xi \rfloor} \binom{k}{i} (-1)^i B^{-(\xi-i)\left(\alpha - \frac{\ln(1+\alpha)}{\ln B}\right)} \right. \\ &\quad \left. - \sum_{0 \leq \ell \leq k-1} (-\alpha \ln B + \ln(1 + \alpha))^\ell \cdot \mathcal{D}_\xi^{k-\ell-1} \tilde{m}^k(\xi) \right), \\ \tilde{\eta}_{\text{norm}}^k \langle \xi \rangle &= \sum_{0 \leq i \leq \lfloor \xi \rfloor} \binom{k}{i} (-1)^i \frac{\text{cutexp}_k(-(\xi-i)(\alpha \ln B - \ln(1+\alpha)))}{(-\alpha \ln B + \ln(1+\alpha))^k}, \end{aligned}$$

where $\text{cutexp}_k(\zeta) = \exp(\zeta) - \sum_{0 \leq \ell \leq k-1} \frac{\zeta^\ell}{\ell!} = \sum_{\ell \geq k} \frac{\zeta^\ell}{\ell!}$.

6.8. Non-squarefree numbers are negligible

Considering $\kappa_{B,C}^k(x)$ we immediately observe that the ordering of the counted prime lists are not important and we can group together many such elements. To get a precise picture, we define the *sorting* of a tuple $P = (p_1, \dots, p_k)$ in the following way:

$$S(P) := \left(\{j \leq k \mid \text{rank}_j P = i\} \right)_{i \leq k},$$

where $\text{rank}_j P = \#\{p_\ell \mid \ell \leq k \wedge p_\ell \leq p_j\}$. Now the count $\kappa_{B,C}^k(x)$ can be partitioned using the sets

$$A_S(x) := \left\{ P = (p_1, \dots, p_k) \in (\mathbb{P} \cap]B, C])^k \left| \begin{array}{l} n = p_1 \dots p_k \leq x, \\ S(P) = S \end{array} \right. \right\},$$

namely $\kappa_{B,C}^k(x) = \sum_S \#A_S(x)$ where S runs over all possible sortings. The above intuition would imply that many of these sets are essentially equal. We group them by their *type*

$$T(S) := (\#S_i)_{i \leq k}.$$

Given any type $T = (T_1, \dots, T_r)$, there are exactly $\binom{k}{T} = \frac{k!}{T_1! \dots T_r!}$ different sortings S of type T . This corresponds to possible reorderings of a specific vector $P \in A_S(x)$ for a sorting S of type T . The type of such a vector P is defined to be $T(S)$. It is clear that the type of P is invariant under permutations, yet not its sorting.

LEMMA 6.8.1. *Let T be a type for k elements.*

- (i) *There exists a sorting $S(T)$ of type T such that all vectors in $A_{S(T)}(x)$ are increasing.*
- (ii) *If $T(S) = T$ then there is a permutation σ of k elements such that for all x we have $A_S(x) = A_{S(T)}(x)^\sigma$.*
- (iii) *More precisely, for any sorting S of k elements the following are equivalent:*
 - (a) $T(S) = T$.
 - (b) $\exists \sigma: S = S(T)^\sigma$.
 - (c) $\exists \sigma: \forall x: A_S(x) = A_{S(T)}(x)^\sigma$. □

Noting that $\#\{S \mid T(S) = T\} = \binom{k}{T}$ we have

$$\kappa_{B,C}^k(x) = \sum_T \sum_{S: T(S)=T} \#A_S(x) = \sum_T \binom{k}{T} \#A_{S(T)}(x).$$

On the other hand we have $\pi_{B,C}^k(x) = \sum_T 1 \cdot \#A_{S(T)}(x)$. In particular, we can deduce

$$\pi_{B,C}^k(x) < \kappa_{B,C}^k(x) \leq k! \cdot \pi_{B,C}^k(x).$$

Actually, for large B (and C and x) we have $\pi_{B,C}^k(x) \sim \frac{1}{k!} \kappa_{B,C}^k(x)$. This stems from the following fact that $\#A_S(x)$ is asymptotically much smaller than $\#A_{S(1,\dots,1)}(x)$ for any sorting S of k elements of type different from $(1, \dots, 1)$.

LEMMA 6.8.2. *For any sorting S of k elements of type different from $(1, \dots, 1)$ for some sorting S' of $k-1$ elements we have $\#A_S(x) \leq \#A_{S'}(x/B) \leq \frac{x}{B}$.*

PROOF. Take S as specified. Let t be a position which does not occur as a singleton in S . Further, say $t \in S_\tau$, and let $r = \#S_\tau \geq 2$. Let S^- be the sorting with S_τ removed, and S' the sorting with t removed. (Retaining the old indexing is easier, yet then indices run over $\{1, \dots, k\} \setminus S_\tau$, or $\{1, \dots, k\} \setminus \{t\}$, respectively.) Then

$$\#A_S(x) = \sum_{p_t \in \mathbb{P} \cap]B, C]} \#A_{S^-}(x/p_t^r) \leq \sum_{p_t \in \mathbb{P} \cap]B, C]} \#A_{S^-}(x/p_t B) = \#A_{S'}(x/B).$$

For the inequality note that $S^- = (S')^-$. Since obviously $\#A_S(z) \leq z$ we are done. □

Combining this with $\sum_S \#A_S(x) = \kappa_{B,C}^k(x) \sim \tilde{\kappa}_{B,C}^k(x) \in \Theta\left(\frac{x}{\ln B}\right)$, shows that there must be a large summand, which can be only $\#A_{S(1,\dots,1)}(x)$.

The number $s(k) = \sum_T \binom{k}{T}$ of sortings of k elements is called ordered Bell number. We can also recursively define them: $s(0) = 1$, $s(k) = \sum_{0 \leq r \leq k-1} \binom{k}{r} s(r)$. According to Wilf (1994), page 175f, we have $s(k) = \frac{k!}{2 \ln^{k+1} 2} + \mathcal{O}\left((0.16)^k k!\right)$. In particular, $s(k)$ is small in comparison to $2^{k-1} k!$. Using Lemma 6.8.2 for a comparison yields the — now immediate — following

LEMMA 6.8.3. *We have*

$$\left| \pi_{B,C}^k(x) - \frac{1}{k!} \kappa_{B,C}^k(x) \right| \leq \left(2^{k-1} - \frac{s(k)}{k!} \right) \frac{x}{B} < 2^{k-1} \frac{x}{B}.$$

PROOF. $\left| k! \cdot \pi_{B,C}^k(x) - \kappa_{B,C}^k(x) \right| \leq \sum_T \left(k! - \binom{k}{T} \right) \frac{x}{B} = \left(k! 2^{k-1} - s(k) \right) \frac{x}{B}$. \square

Compared to the error bound in $\left| \kappa_{B,C}^k(x) - \tilde{\kappa}_{B,C}^k(x) \right| \leq \tilde{\kappa}_{B,C}^k(x) \in \mathcal{O}\left(\frac{x}{\sqrt{B}}\right)$ this is negligible. Here we assume $k \geq 2$ since the present observations are irrelevant for $k = 1$, namely $\pi_{B,C}^1 = \kappa_{B,C}^1$. Now let $\tilde{\pi}_{B,C}^k(x) := \frac{1}{k!} \tilde{\kappa}_{B,C}^k(x)$, $\hat{\pi}_{B,C}^k(x) := \frac{1}{k!} \hat{\kappa}_{B,C}^k(x) + 2^{k-1} \frac{x}{B}$. Combining with Theorem 6.4.16 and Theorem 6.5.1 we finally arrive at Theorem 6.9.4, the overall summary of our results.

6.9. Results on coarse-grained integers

We are going to combine the results of the last section with Theorem 6.4.16 and Theorem 6.5.1 to finally arrive at our main theorem.

Analogously to Definition 6.3.1, we define an approximation function for the function $\pi_{B,C}^k(x)$.

DEFINITION 6.9.1. *For $x \geq 0$ and $k \geq 0$ we define*

$$\tilde{\pi}_{B,C}^k(x) := \frac{1}{k!} \tilde{\kappa}_{B,C}^k(x)$$

and

$$\hat{\pi}_{B,C}^k(x) := \frac{1}{k!} \hat{\kappa}_{B,C}^k(x) + 2^{k-1} \frac{x}{B}.$$

Similarly to Definition 6.3.1 we can also recursively define $\tilde{\pi}_{B,C}^k(x)$ by

$$\tilde{\pi}_{B,C}^0(x) = \begin{cases} 0 & \text{if } x < 1, \\ 1 & \text{if } 1 \leq x, \end{cases} \quad \tilde{\pi}_{B,C}^k(x) = \frac{1}{k} \int_B^C \frac{\tilde{\pi}_{B,C}^{k-1}(x/p_k)}{\ln p_k} dp_k.$$

It is also possible to define $\widehat{\pi}_{B,C}^k(x)$ similarly based on Definition 6.3.1. We can now describe the behavior of $\pi_{B,C}^k$ nicely and give our main result.

THEOREM 6.9.2. *Given $x \in \mathbb{R}_{>0}$ and $k \in \mathbb{N}$. Then the inequality*

$$\left| \pi_{B,C}^k(x) - \widetilde{\pi}_{B,C}^k(x) \right| \leq \widehat{\pi}_{B,C}^k(x)$$

holds.

PROOF. By Lemma 6.8.3 we have

$$\left| \pi_{B,C}^k(x) - \frac{1}{k!} \kappa_{B,C}^k(x) \right| < 2^{k-1} \frac{x}{B}.$$

Thus using the triangle inequality and Theorem 6.3.2, we obtain

$$\left| \pi_{B,C}^k(x) - \frac{1}{k!} \widetilde{\kappa}_{B,C}^k(x) \right| < \frac{1}{k!} \widetilde{\kappa}_{B,C}^k(x) + 2^{k-1} \frac{x}{B},$$

which proves the claim. \square

THEOREM 6.9.3. *Fix $k \geq 2$. Then for any $\varepsilon > 0$ and B tending to infinity, there are for $x \in [B^k(1 + \varepsilon), C^k(1 - \varepsilon)]$ values $\widetilde{s}, \widehat{s} \in \left[\frac{1}{(1+\alpha)^k}, 1 \right]$ such that*

$$\widetilde{\pi}_{B,C}^k(x) = \frac{\widetilde{s}}{k!} \widetilde{\lambda}^k(x), \quad \widehat{\pi}_{B,C}^k(x) = \frac{\widehat{s}}{k!} \widehat{\lambda}^k(x) + 2^{k-1} \frac{x}{B}.$$

PROOF. By Definition 6.9.1 we have

$$\widetilde{\pi}_{B,C}^k(x) = \frac{1}{k!} \widetilde{\kappa}_{B,C}^k(x).$$

Theorem 6.3.5 tells us that there is a value $\widetilde{s} \in \left[\frac{1}{(1+\alpha)^k}, 1 \right]$ such that

$$\widetilde{\kappa}_{B,C}^k(x) = \widetilde{s} \widetilde{\lambda}^k(x),$$

implying that for the same value \widetilde{s} we have

$$\widetilde{\pi}_{B,C}^k(x) = \frac{\widetilde{s}}{k!} \widetilde{\lambda}^k(x).$$

Considering $\widehat{\pi}_{B,C}^k(x)$ we have by Definition 6.9.1 that

$$\widehat{\pi}_{B,C}^k(x) = \frac{1}{k!} \widehat{\kappa}_{B,C}^k(x) + 2^{k-1} \frac{x}{B}.$$

Applying Theorem 6.3.5 gives a value $\widehat{s} \in \left[\frac{1}{(1+\alpha)^k}, 1 \right]$ such that

$$\widehat{\kappa}_{B,C}^k(x) = \widehat{s} \widehat{\lambda}^k(x),$$

which directly gives

$$\widehat{\pi}_{B,C}^k(x) = \frac{\widehat{s}}{k!} \widehat{\lambda}^k(x) + 2^{k-1} \frac{x}{B}.$$

This proves the theorem. \square

Unrolling our results on $\widetilde{\lambda}^k(x)$ and $\widehat{\lambda}^k(x)$, namely Theorem 6.4.16 and Theorem 6.5.1, gives a slightly weaker result.

THEOREM 6.9.4. *Let $B < C = B^{1+\alpha}$ with $\alpha \geq \frac{\ln B}{\sqrt{B}}$ and fix $k \geq 2$. Then for any (small) $\varepsilon > 0$ and B tending to infinity we have for $x \in \left[B^k(1+\varepsilon), C^k(1-\varepsilon) \right]$ a value $\widetilde{c} \in \left[\frac{\alpha^{k-1} \delta_{\widetilde{\lambda}^k}}{k!(1+\alpha)^k}, \frac{1}{k!} \right]$ with $\delta_{\widetilde{\lambda}^k} = \min \left(2^{-4} \frac{\varepsilon^k}{k!}, 2^{-k} \frac{\varepsilon^{k-1}}{(k-1)!} \right)$ such that*

$$\left| \pi_{B,C}^k(x) - \widetilde{c} \frac{x}{\ln B} \right| \leq (2^k - 1) \alpha^{k-2} (1 + \alpha) \cdot \frac{x}{\sqrt{B}} + 2^{k-1} \frac{x}{B}.$$

PROOF. By Theorem 6.9.3 we have values $\widetilde{s}, \widehat{s} \in \left[\frac{1}{(1+\alpha)^k}, 1 \right]$ such that

$$\widetilde{\pi}_{B,C}^k(x) = \frac{\widetilde{s}}{k!} \widetilde{\lambda}^k(x), \quad \widehat{\pi}_{B,C}^k(x) = \frac{\widehat{s}}{k!} \widehat{\lambda}^k(x) + 2^{k-1} \frac{x}{B}.$$

By Theorem 6.4.16 we have for ε small enough that

$$\widetilde{\lambda}^k(x) \in \left[\delta_{\widetilde{\lambda}^k}, 1 \right] \frac{\alpha^{k-1} x}{\ln B}$$

and by Theorem 6.5.1 that

$$\widehat{\lambda}^k(x) \leq (2^k - 1) \alpha^{k-2} (1 + \alpha) \cdot \frac{x}{\sqrt{B}}.$$

This gives the claim. \square

6.10. Numeric evaluation

To discuss the quality of our results we consider again the example parameters $B = 1100 \cdot 10^6$, $C = 2^{37} - 1$, $\alpha = \ln(C)/\ln(B) - 1 = 0.232\uparrow$, $k = 4$ from our inspiring application. For $k = 2$, $k = 3$ we can give similar pictures; of course, the errors are even smaller in these cases. At present we do not have efficient algorithms for computing $\kappa_{B,C}^k$ itself. However, based on our estimates we can compute values for encapsulating intervals in three variants, listed in increasing quality:

- λ estimate (Theorem 6.3.5):

$$\left[\frac{1}{(1+\alpha)^k} \tilde{\lambda}^k(x) - \hat{c}_k \alpha^k x, \quad \tilde{\lambda}^k(x) + \hat{c}_k \alpha^k x \right].$$

- η estimate (Theorem 6.7.2):

$$\left[\left(\frac{\ln(1+\alpha)}{\alpha} (1+\alpha)^{\frac{1}{\ln(1+\alpha)} - \frac{1}{\alpha}} \right)^k \tilde{\eta}^k(x) - \hat{c}_k \alpha^k x, \quad \tilde{\eta}^k(x) + \hat{c}_k \alpha^k x \right].$$

- κ estimate (Theorem 6.3.2):

$$\left[\tilde{\kappa}_{B,C}^k(x) - \hat{\kappa}_{B,C}^k(x), \quad \tilde{\kappa}_{B,C}^k(x) + \hat{\kappa}_{B,C}^k(x) \right].$$

The κ estimate was easiest to obtain and is of course the most accurate one, however, it is difficult to evaluate. The λ estimate was easy to obtain and compute. But it is of course the least accurate of the three. The η estimate was slightly more difficult to find, is as easy to evaluate as the prior one, and it is much more accurate. As usual we write $x = B^{k+\xi\alpha}$ and use ξ as a running parameter.

Figure 6.10.1 shows the absolute behavior of all estimates. We observe that the absolute errors at the right margin are huge. This is expected as also the error estimates in the prime number theorem only bound the relative error. However, the picture completely conceals information about the middle and the left part of the interval $[0, k]$.

To see more we divide by $x = B^{k+\xi\alpha}$ and therefore obtain estimates for the ratio of $]B, C]$ -grained integers x in Figure 6.10.2. This reveals a lot about the quality of our estimates. The black area indicates the best that we could hope for, namely the estimate based merely on Prime Number Theorem 2.1.8(iii). However, as this is difficult to evaluate we have to approximate once more. The λ estimate, shown in light gray, is clearly only of use to get a rough idea. The η estimate, however, is rather close to the actual behavior and may well serve as a basis for stochastic fine tuning of algorithms like the General Number Field Sieve.

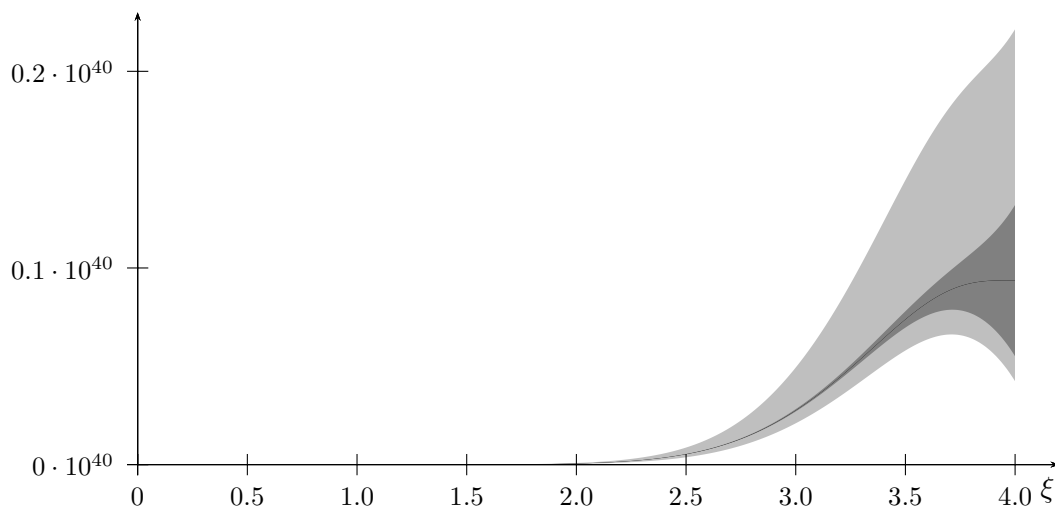


Figure 6.10.1: Absolute behavior of the estimates for $\kappa_{B,C}^k \langle \xi \rangle$. The light gray area shows the λ estimate, the dark gray area the η estimate, and the black area (well, yes) shows the κ estimate. The parameters are $B = 1100 \cdot 10^6$, $C = 2^{37} - 1$, $\alpha = \ln(C)/\ln(B) - 1 = 0.232\mathfrak{T}$, $k = 4$.

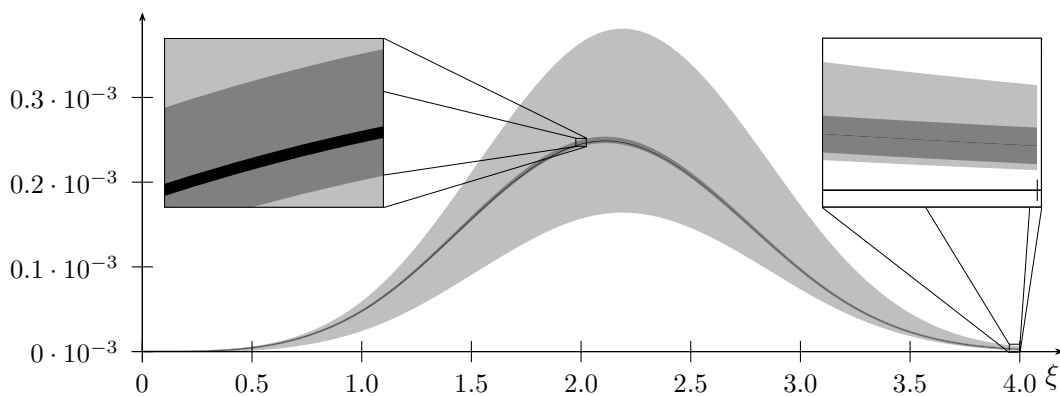


Figure 6.10.2: Behavior of the estimates relative to $x = B^{k+\xi\alpha}$. Colors and parameters are as in Figure 6.10.1.

Last, Figure 6.10.3 illustrates the size of the various errors terms relative to $\tilde{\kappa}^k$:

— $\tilde{\lambda}$ error: $\left(1 - \frac{1}{(1+\alpha)^k}\right) \tilde{\lambda}^k(x)$.

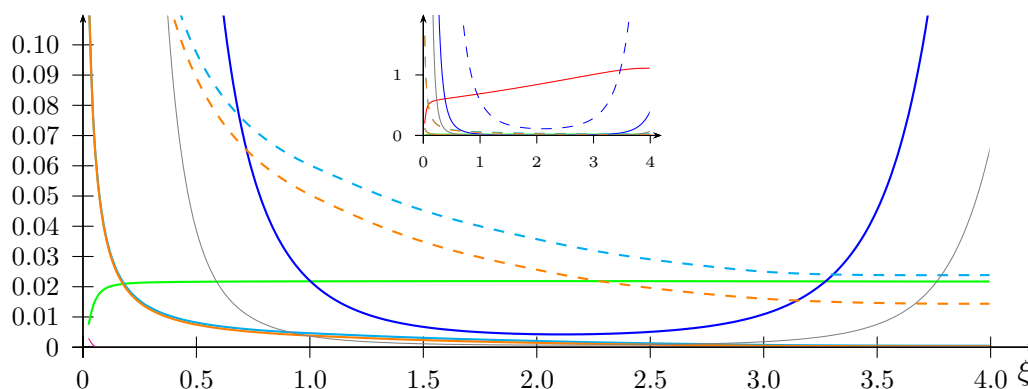


Figure 6.10.3: Errors of the various estimates relative to $\tilde{\kappa}_{B,C}^k$. Parameters are as in Figure 6.10.1.

- $\tilde{\eta}$ error: $\left(1 - \left(\frac{\ln(1+\alpha)}{\alpha}(1+\alpha)^{\frac{1}{\ln(1+\alpha)} - \frac{1}{\alpha}}\right)^k\right) \tilde{\eta}^k(x)$.
- $\hat{\lambda}$ error bound: $\hat{c}_k \alpha^k x$. - - - Unconditional $\hat{\lambda}$ error bound: $\hat{c}_k \alpha^k x$.
- $\hat{\lambda}$ error: $\hat{\lambda}^k(x)$. - - - Unconditional $\hat{\lambda}$ error: $\hat{\lambda}^k(x)$.
- $\hat{\kappa}$ error: $\hat{\kappa}_{B,C}^k(x)$. - - - Unconditional $\hat{\kappa}$ error: $\hat{\kappa}_{B,C}^k(x)$.
- Non-squarefree error bound: $\left(2^{k-1}k! - s(k)\right) \frac{x}{B}$.
- Non-squarefree error: $\left(2^{k-1}k! - s(k)\right) \frac{x}{B}$.

For the unconditional errors we use Dusart's unconditional bound on $|\pi(x) - \text{Li}(x)|$ which is given by $\tilde{E}(x) = 2.3854 \frac{x}{\ln^3 x}$ for $x \geq 355\,991$.

The figure shows that all the error terms but the $\tilde{\lambda}$ error are sufficiently small. Our best choice is the $\tilde{\eta}$ estimate which is ruled by the $\tilde{\eta}$ error and the $\hat{\lambda}$ error. Both are fairly less than 3% of the target value at least in the middle of the interval. The estimations are more difficult close to the boundaries. It is also positive that, at the parameters of our interest, most error terms are still comparative in size to the contributions of the $\hat{\kappa}$ error, which is induced by the Prime Number Theorem 2.1.8. Definitely, the η estimate, combining the $\tilde{\eta}$ error and the $\hat{\lambda}$ error, is good enough for practical purposes, as for example the fine tuning of the general number field sieve.

Chapter 7

Hardware for the General Number Field Sieve

We will now explore how to optimize hardware realizations of the cofactorization step of the General Number Field Sieve (see Section 3.4.3). The results in this chapter were published at SAC 2009 in Calgary, Canada (see Loebenberg & Putzka 2009). Our coauthor suggested to analyze the bitlength-structure of the cofactorization inputs and wrote a short section on the number field sieve (which is in this version omitted). Both, the framework for analyzing the cluster as well as the actual optimization results are our own discovery.

7.1. Framework

In recent implementations of the General Number Field Sieve (GNFS), described in Section 3.4.3, the Elliptic Curve Method 3.5.9 (ECM) is used to factor intermediate sieving results. For example in the record factorization of Franke & Kleinjung (2005) the sieving step produced intermediate numbers of length up to 128 bits. Adapting this to the factorization problem of the number RSA-768 (RSA Laboratories 2007) results in the task of factoring roughly $2 \cdot 10^{12}$ numbers of length up to 140 bit using the ECM.

Since cofactorization is a costly part of the GNFS, it is natural to think about highly specialized hardware realizations of this step, to improve the performance of the GNFS considerably. In particular, since the task consists of many very similar steps, a realization as a hardware *cluster* is suitable. On such a cluster one has many computational units running in parallel that are able to process inputs up to a certain bitlength. The question remains how many of those bitlength-specific modules should be implemented, re-

ardless of the concrete implementation of the corresponding ECM modules. A straightforward approach would be to construct only modules capable of factoring inputs of any size from the GNFS. It is clear, however, that this approach is a great waste of logical resources and that a detailed study of the bitlength-structure of the inputs to the cofactorization step results in much better performance than the naïve approach. Furthermore we quantify the gain we achieve using our optimized construction and generalize our result to arbitrary clusters.

7.2. Modelling the cluster system

Our goal is a model of a hardware cluster (e.g. a COPACOBANA, see Kumar *et al.* (2006) and Güneysu *et al.* (2008), using Virtex4 XC4VSX35 FPGAs). In our specific example the cluster has 16 slots, each containing 8 FPGAs (in the following called chips). Each chip can run several ECM-processes in parallel depending on the size of the corresponding ECM-module. We assume that each chip can only be filled with ECM modules of a particular size. This requirement is from a theoretical point of view unnecessary, but for the concrete realization we have in mind we actually have to require this, since the device controlling all the chips is in our case not able to perform otherwise. Of course modules constructed for a given bitlength can also factor shorter integers. If one wants to factor a number using the cluster, the number is forwarded to a module suitable for its bitlength. The corresponding module then attempts to find a nontrivial factor of the input number. If this succeeds after a certain number of trials (each being a separate run of the ECM with a different elliptic curve), the factor is sent back to the controlling host computer, otherwise the number is discarded. If the factor that is sent back or the remaining cofactor is still composite, another factoring attempt is made. We assume for our estimates that the effort for these additional factorizations is negligible when compared to the first factorization attempt.

The first question we have to answer is the following: From an engineering point of view it is unrealistic to build arbitrary sized ECM modules. What is the smallest bitlength $g \in \mathbb{N}$ for which such a construction is practical? We call this g the *granularity* of the implementation. Of course one cannot give a general answer to this question. The answer heavily depends on the type of the chips one is using and the concrete implementation one has in mind. In our example, we will have $g = 17$ due to the design of the Virtex4 XC4VSX35 FPGAs.

Another question is: How can we get rid of modules for which the num-

bers of integers having that bitlength is very small? In other words, if for a particular bitlength there are only very few numbers to factor, it would be better to factor such numbers using modules capable of factoring larger integers. This would ensure that we would not waste any resources on the cluster, resulting in a better runtime of the cofactorization step.

We describe now the model of the cluster: Let N denote the number of chips on the cluster, e.g. $N = 128$ in our concrete example, and let \mathcal{D} denote the set of inputs to the cofactorization step with $M := \#\mathcal{D}$. For $d \in \mathcal{D}$ let $\text{len}(d)$ denote the bitlength of the number d , i.e. $\text{len}(d) := \lfloor \log_2(d) \rfloor + 1$. Each of the input numbers can be handled by specific modules suitable for their bitlength. The size for which the modules are designed is always a multiple of g . We denote by n_i the number of parallel ECM modules for an integer having $i \cdot g$ bits and by c_i the average runtime of such an integer on the corresponding chips. We are now going to model the classes the numbers may fall into. In general, if we are given an interval $\mathcal{I} := [x, y]$ with $x, y \in \mathbb{N}$ and $x \leq y$, a *partition* of \mathcal{I} is a sequence $\mathcal{C} := (\mathcal{C}_0, \mathcal{C}_1, \dots, \mathcal{C}_k) \in \mathbb{N}^k$ for some $k \in \mathbb{N}$, with $x = \mathcal{C}_0 < \mathcal{C}_1 < \dots < \mathcal{C}_k = y$. We call k the *size* of the partition \mathcal{C} . The interval $(\mathcal{C}_{i-1}, \mathcal{C}_i]$ is called the *i -th subinterval* of \mathcal{C} . If now \mathcal{C}^1 and \mathcal{C}^2 are partitions of \mathcal{I} , we say that \mathcal{C}^2 is a *refinement* of \mathcal{C}^1 if for any $0 \leq i \leq k$ there is some j , such that $\mathcal{C}_i^1 = \mathcal{C}_j^2$. In other words that means that we have subdivided the subintervals of \mathcal{C}^1 into smaller pieces without changing already existing cuts and we write $\mathcal{C}^1 \preceq \mathcal{C}^2$. Conversely, \mathcal{C}^1 is called a *coarsening* of \mathcal{C}^2 . For our purposes we only consider partitions \mathcal{C} of the interval $\mathcal{I} = [x, y]$ where $x := \lfloor \min(\text{len}(d) \mid d \in \mathcal{D}) \rfloor_g$ and $y := \lceil \max(\text{len}(d) \mid d \in \mathcal{D}) \rceil_g$, where the notation $\lfloor \cdot \rfloor_g$ ($\lceil \cdot \rceil_g$) means that the rounding is done down to (up to) the next multiple of g . Additionally we require that for any $0 \leq i < \#\mathcal{C}$ the number \mathcal{C}_i is a multiple of g . We will call such partitions *g -partitions* of the interval induced by \mathcal{D} . In particular the finest partition we will consider is the *g -partition* $\mathcal{C}^f := (x, x + g, x + 2g, \dots, y)$ and the possible partitions we may have at the end are always coarsenings of \mathcal{C}^f .

For the following, fix a data set \mathcal{D} and define $K := \#\mathcal{C}^f - 1 = (y - x)/g$. Now given any $\mathcal{C} \preceq \mathcal{C}^f$ of size k , let $a_i(\mathcal{C}) \in \mathbb{N}$ be the number of occurrences in the i -th subinterval of \mathcal{C} , i.e. $a_i(\mathcal{C}) := \#\{d \in \mathcal{D} \mid \text{len}(d) \in (\mathcal{C}_{i-1}, \mathcal{C}_i]\}$. For later use we define the input distribution

$$\alpha(\mathcal{C}) := \left(\frac{a_1(\mathcal{C})}{M}, \dots, \frac{a_k(\mathcal{C})}{M} \right) \in \mathbb{R}^k.$$

If we consider the i th subinterval of \mathcal{C} the average cost of factoring such a number is $c_{\mathcal{C}_i/g}$. The space used for such a module is roughly $1/n_{\mathcal{C}_i/g}$. Thus

the area-time product for class i is given by

$$\vartheta_i(\mathcal{C}) := \frac{c_{\mathcal{C}_i/g}}{n_{\mathcal{C}_i/g}}.$$

A *layout* of the cluster is given by an ordered partition $\ell \vdash_k N$ of the N chips into k summands, one for each class. Thus we have

$$\ell \vdash_k N : \iff \ell = (\ell_1, \dots, \ell_k) \in \{1, \dots, N\}^k \wedge \sum_{1 \leq i \leq k} \ell_i = N,$$

with $\ell_i > 0$, implying $N \geq k$. That means we assume that the number of chips is always greater than the number of classes, which is also reasonable. Note that we have indeed two different notions of partitions here: First a partition of an interval and second an additive ordered partition of a natural number. This could of course be unified, but for our work it is preferable to have these two different notions, since for the former notion we emphasize on the variable number of subintervals while for the latter we assume a fixed number of summands.

Write $\mathcal{C}|_j$ for the restriction of \mathcal{C} on its first j subintervals. The minimal runtime for $\mathcal{C}|_j$ is given by

$$(7.2.1) \quad \mu_{\mathcal{C}}(N, j) := \min_{\ell \vdash_j N} \max_{1 \leq i \leq j} \frac{\vartheta_i(\mathcal{C}|_j) \cdot a_i(\mathcal{C}|_j)}{\ell_i}.$$

The value $\mu_{\mathcal{C}}(N, j)$ is indeed a time measurement, since c_i is given in seconds, n_i has unit $1/\text{chip}$ and ℓ_i has unit chip . We will use the following convention: If we write $\mu_{\mathcal{C}}(N)$ we actually mean $\mu_{\mathcal{C}}(N, \#\mathcal{C} - 1)$. Further we define

$$(7.2.2) \quad \tau(N) := \min_{\mathcal{C} \leq \mathcal{C}^f} \mu_{\mathcal{C}}(N).$$

Equations (7.2.1) and (7.2.2) actually depend on the data set \mathcal{D} and we write $\mu_{\mathcal{D}, \mathcal{C}}(N, j)$ and $\tau_{\mathcal{D}}(N)$, respectively, if there is more than one data set under consideration. In the following we will show how one can compute $\mu_{\mathcal{C}^f}(N)$ efficiently, namely with $\mathcal{O}(N \cdot K)$ arithmetic operations. Note that the imprecision of considering arithmetic operations only is in our case not a problem, since the size of the numbers is bounded from above by a constant.

We can compute equation (7.2.1) easily using Bellman's dynamic programming. To do so, we need to handle two things:

1. The solutions for the boundaries have to be computed (i.e. for the case $j = 1$):

$$(7.2.3) \quad \mu_{\mathcal{C}}(N, 1) = \frac{\vartheta_1(\mathcal{C}|_1) \cdot a_1(\mathcal{C}|_1)}{N}.$$

2. We need a recursion formula for $\mu_{\mathcal{C}}(N, j)$. Assume we know $\mu_{\mathcal{C}}(N', j-1)$ for all $N' < N$. Then we have

$$(7.2.4) \quad \mu_{\mathcal{C}}(N, j) = \min_{N' < N} \max \left(\mu_{\mathcal{C}}(N', j-1), \frac{\vartheta_j(\mathcal{C}|_j) \cdot a_j(\mathcal{C}|_j)}{N - N'} \right).$$

The function $\mu_{\mathcal{C}}(N, j)$ can thus be computed with $\mathcal{O}(N \cdot j)$ arithmetic operations.

Let us now compute the function $\tau(N)$. The total number of classes $\mathcal{C} \preceq \mathcal{C}^f$ is $2^K/4$. Since K will be small in all our examples of the GNFS, a straightforward algorithm would just compute $\mu_{\mathcal{C}}(N)$ for all $\mathcal{C} \preceq \mathcal{C}^f$ and select the classes with minimal runtime. Employing such an algorithm for the computation of $\tau(N)$ will use $\mathcal{O}(NK2^K)$ arithmetic operations.

We will now describe a greedy approach which will find in many cases the optimal classes using only $\mathcal{O}(K)$ evaluations of the function $\mu_{\mathcal{C}}(N)$ for various $\mathcal{C} \preceq \mathcal{C}^f$, i.e. compute $\tau(N)$ with $\mathcal{O}(N \cdot K^2)$ arithmetic operations: Let $\mathcal{C} := [\mathcal{C}_0, \mathcal{C}_1, \dots, \mathcal{C}_k]$ be any partition of the interval $\mathcal{I} = [x, y]$.

For $p \in [1, K-1]$ denote by $\mathcal{C}^{(p)}$ the refinement of \mathcal{C} at position $g \cdot p$. Our algorithm will work as follows: Starting from the partition (x, y) , we successively refine (x, y) until the optimal partition is found. In particular if we are given in step r a partition \mathcal{C} , we compute $\mu_{\mathcal{C}^{(p)}}(N)$ for all p and take in the next round the partition $\mathcal{C}^{(p)}$ with the smallest runtime $\mu_{\mathcal{C}^{(p)}}(N)$. If there are two positions p_1, p_2 with the same minimal runtime, we select one of the partitions randomly for the next step. This approach is indeed greedy, since we take in every round the best subdivision. The algorithm terminates if for all p the value $\mu_{\mathcal{C}^{(p)}}(N)$ is not strictly smaller than $\mu_{\mathcal{C}}(N)$. In this case the partition \mathcal{C} is returned. Observe that this algorithm will in general *not* find the optimal classes, since we cannot guarantee that the algorithm terminates in a local minimum. In our experiments, however, this heuristic indeed computed $\tau(N)$ in all our examples.

In order to measure the advantage of our optimization, we compare the estimated runtime of the cluster using our construction with the runtime of a naïvely constructed cluster, i.e. a cluster only containing bitlength-specific modules for numbers having y bits. On such a cluster the runtime for a data set \mathcal{D} of M numbers is bounded from below by the following expression:

$$(7.2.5) \quad \sigma_{\mathcal{D}}^-(N) := \frac{1}{N \cdot n_K} \sum_{1 \leq i \leq K} c_i a_i$$

and bounded from above by

$$(7.2.6) \quad \sigma_{\mathcal{D}}^+(N) := \frac{Mc_K}{Nn_K}$$

with $K := \#\mathcal{C}^f - 1$ as above. The first estimate is a bit optimistic since the runtime of a module does not only depend on the input but also on the arithmetic built into the module. Further the second estimate is too pessimistic, since a module running on smaller input numbers will also run faster on average.

We use the functions

$$\gamma_{\mathcal{D}}^-(N) := \frac{\sigma_{\mathcal{D}}^-(N) - \tau_{\mathcal{D}}(N)}{\sigma_{\mathcal{D}}^-(N)}$$

and

$$\gamma_{\mathcal{D}}^+(N) := \frac{\sigma_{\mathcal{D}}^+(N) - \tau_{\mathcal{D}}(N)}{\sigma_{\mathcal{D}}^+(N)}$$

as lower and upper bounds, respectively, to measure the runtime gain we achieve with our optimized cluster. This expression is exactly the runtime gain achieved by the optimization (having runtime $\tau_{\mathcal{D}}(N)$) in contrast to the naively constructed cluster (having runtime between $\sigma_{\mathcal{D}}^-(N)$ and $\sigma_{\mathcal{D}}^+(N)$).

7.3. Concrete statistical analyses

We will now perform a rigorous statistical analysis of six concrete runs of the GNFS up to the cofactorizations step for the number RSA-768 and study the function $\tau(N)$ for these particular inputs: Each data set \mathcal{D} consists of many ($2 \cdot 10^8$)-rough composite numbers of bitlength between 58 and 160, each \mathcal{D} being a specific output of the sieving step of the GNFS for different choices of a polynomial pair and the sieving region of the lattice siever. Following von zur Gathen *et al.* (2007), we estimate the number of parallel ECM modules and the runtime on the Virtex4 XC4VSX35 FPGAs according to Table 7.3.1 and Table 7.3.2, respectively. In the implementation that was used only modules for 17i bit integers were build. Note that such a module will also be capable of factoring smaller integers.

Let us have a look at the distribution $\alpha(\mathcal{C}^f)$ of the input data for the various data sets (see Table 7.3.3). Note the low standard deviation of the corresponding entries. In Figure 7.3.1 a histogram as well as the distribution on the classes \mathcal{C}^f is given for data set \mathcal{D}_1 .

We now employ our model to find an optimal layout for the cluster and compute the runtime gain we achieved with our optimization. Let the

Bitlength $17i$	17	34	51	68	85	102	119	136	153	170
Processes n_i	32	26	22	18	15	12	10	9	8	7

Table 7.3.1: Number of parallel ECM-modules per chip depending on the bitlength

Bitlength $17i$	17	34	51	68	85
Cost c_i (in μs)	491.49125	673.9225	856.35375	1038.785	1221.21625

Bitlength $17i$	102	119	136	153	170
Cost c_i (in μs)	1403.6475	1586.07875	1768.51	1950.94125	2133.3725

Table 7.3.2: Average runtime of the ECM on a Virtex4 XC4VSX35 FPGA

Bitlength	0 – 68	69 – 85	86 – 102	103 – 119	120 – 136	137 – 153
\mathcal{D}_1	0.0015	0.0553	0.4540	0.0886	0.2826	0.1181
\mathcal{D}_2	0.0007	0.0547	0.4493	0.0889	0.2823	0.1241
\mathcal{D}_3	0.0008	0.0540	0.4533	0.0881	0.2836	0.1203
\mathcal{D}_4	0.0009	0.0567	0.4440	0.0874	0.2902	0.1209
\mathcal{D}_5	0.0011	0.0518	0.4306	0.0875	0.2992	0.1299
\mathcal{D}_6	0.0009	0.0461	0.4340	0.0834	0.3031	0.1326
Mean	0.0010	0.0531	0.4442	0.0873	0.2902	0.1243
Stdev.	0.0003	0.0038	0.0099	0.0020	0.0091	0.0058

Table 7.3.3: Relative frequencies of the input data

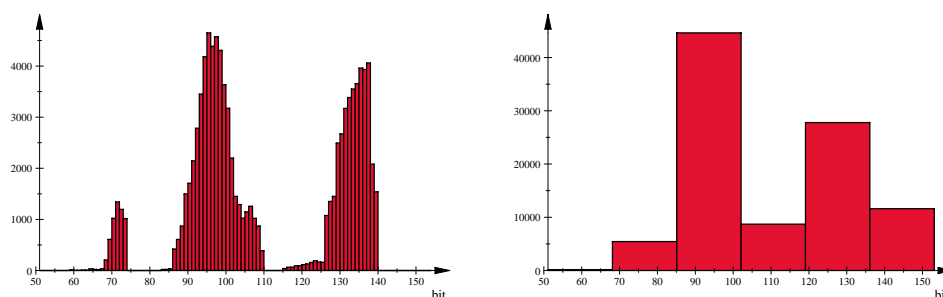


Figure 7.3.1: Left: Histogram of data set \mathcal{D}_1 . Right: Distribution onto specific modules.

notation be as in Section 7.2. In the case of the COPACOBANA we will have $N = 8 \cdot 16 = 128$. There are 351306039 ordered partitions of the number 128 in not more than 6 parts. The total number of layouts of the cluster, including the choice of the classes is in our example 402858941.

After having computed the function $\tau_{\mathcal{D}}(128)$ for all data sets \mathcal{D} we obtain for every set an optimal layout (consisting of the interval partition \mathcal{C} and the distribution of chips ℓ). If we take the result of the optimization for data set \mathcal{D}_1 , for example, we will have 47 modules for integers of up to 102 bit, 58 for integers up to 136 bit and 23 for the remaining integers (up to 153 bit). The size of the first class is in this case 102 bit, the size of the second one 34 bit and of the third class 17 bit. The results are summarized in Table 7.3.4 and 7.3.5.

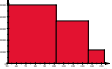
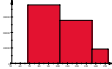
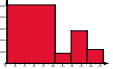
	\mathcal{D}_1	\mathcal{D}_2	\mathcal{D}_3
			
$(\mathcal{C}_{i+1} - \mathcal{C}_i)/g$	(3,2,1)	(1,2,2,1)	(3,1,1,1)
ℓ	(47, 58, 23)	(1, 46, 57, 24)	(48, 11, 45, 24)
$\tau_{\mathcal{D}} (\mu\text{s})$	124966.936	96137.13955	126309.5441
$\#\mathcal{D}$	98322	75013	99488
$\tau_{\mathcal{D}}/\#\mathcal{D}$	1.271	1.2816	1.2696

Table 7.3.4: Optimal partitions for the data sets \mathcal{D}_1 , \mathcal{D}_2 and \mathcal{D}_3

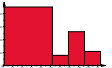
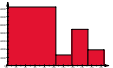
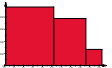
	\mathcal{D}_4	\mathcal{D}_5	\mathcal{D}_6
			
$(\mathcal{C}_{i+1} - \mathcal{C}_i)/g$	(3,1,1,1)	(3,1,1,1)	(3,2,1)
ℓ	(47, 11, 46, 24)	(45, 11, 47, 25)	(44, 59, 25)
$\tau_{\mathcal{D}} (\mu\text{s})$	113592.0763	37653.16612	65015.11716
$\#\mathcal{D}$	90141	29719	50273
$\tau_{\mathcal{D}}/\#\mathcal{D}$	1.2602	1.267	1.2932

Table 7.3.5: Optimal partitions for the data sets \mathcal{D}_4 , \mathcal{D}_5 and \mathcal{D}_6

In order to measure the advantage of our optimization, we use the estimates from Section 7.2. We have here at maximum 153 bit numbers and use the values in the tables above. The result of our optimization is shown

in Table 7.3.6.

	\mathcal{D}_1	\mathcal{D}_2	\mathcal{D}_3	\mathcal{D}_4	\mathcal{D}_5	\mathcal{D}_6
$\gamma_{\mathcal{D}}^-$	17.47	16.97	17.66	18.38	18.4	16.88
$\gamma_{\mathcal{D}}^+$	33.29	32.73	33.36	33.86	33.5	32.12

Table 7.3.6: Performance gain for the different data sets (in percent) of the optimized cluster

7.4. Generalizations to an arbitrary number of clusters

Fix one data set \mathcal{D} . In this section we analyze the behavior of the function $\gamma^-(N)$ for $N \rightarrow \infty$.

In practice a growing N would mean that we employ not only one CO-PACOBANA, but a whole collection of these, running simultaneously, and optimize over the whole set of chips. We will now show that the runtime gain achieved by this collection of clusters converges to roughly 21% when compared to a collection of naïvely constructed clusters. It is clear that the actual gain however will strongly depend on the input data \mathcal{D} .

Now let's say we are going to build m clusters and we wish to optimize the number of bitlength specific ECM modules as above. The formulae in Section 7.2 are still valid, except that we will have $N = 128m$ chips in a collection of m clusters instead of $N = 128$ as above.

We wish to compute $\lim_{N \rightarrow \infty} \gamma^\pm(N)$. To do so, we first need to compute $\tau(N)$ for $N \rightarrow \infty$. Unfortunately, the dynamic programming approach used above is only useful if we consider fixed N , but does not tell us anything about the limit. In Figure 7.4.1 the value of $\gamma^-(N)$ is plotted for the case of $m \in \{1, \dots, 100\}$ clusters using data set \mathcal{D}_1 . Note that this observation follows our intuition, since with an increasing number of clusters one cannot expect more runtime gain.

Assume we are given classes $\mathcal{C} \preceq \mathcal{C}^f$. Set $k := \#\mathcal{C} - 1$. In order to be able to compute the limit, we look at the problem of computing $\mu_{\mathcal{C}}(N)$ over the reals, i.e. we will have $\ell \in \mathbb{R}^k$. With this simplifications it is clear that the expression

$$\max_{1 \leq i \leq k} \frac{\vartheta_i(\mathcal{C}) \cdot a_i(\mathcal{C})}{\ell_i}$$

is minimal if and only if

$$\frac{\vartheta_i(\mathcal{C}) \cdot a_i(\mathcal{C})}{\ell_i} = \frac{\vartheta_j(\mathcal{C}) \cdot a_j(\mathcal{C})}{\ell_j} \text{ for all } i, j \in \{1, \dots, k\}.$$

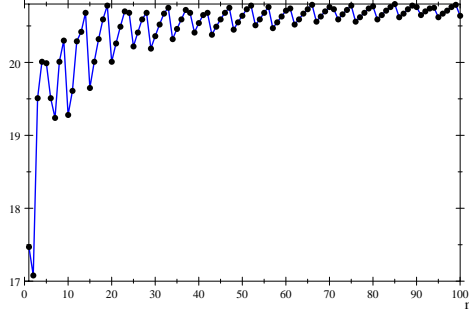


Figure 7.4.1: Lower bound on the runtime gain for an increasing number m of clusters

Write $\vartheta'_i(\mathcal{C}) := \vartheta_i(\mathcal{C}) \cdot a_i(\mathcal{C})$. We end up in solving the following system of equations:

$$\begin{aligned} \ell_1 + \cdots + \ell_k &= N, \\ \vartheta'_1(\mathcal{C}) \cdot \ell_2 &= \vartheta'_2(\mathcal{C}) \cdot \ell_1, \\ &\vdots \\ \vartheta'_1(\mathcal{C}) \cdot \ell_k &= \vartheta'_k(\mathcal{C}) \cdot \ell_1, \end{aligned}$$

This system of k equations is linear in the k unknowns ℓ_1, \dots, ℓ_k , having the solution

$$\ell_i = \frac{\vartheta'_i(\mathcal{C})N}{\vartheta'_1(\mathcal{C}) + \cdots + \vartheta'_k(\mathcal{C})}.$$

We could have used this approach also for our computation of $\mu_{\mathcal{C}}(n)$ in Section 7.2. There we would have computed the approximate partition of N (being a vector of reals) and would then have rounded the results appropriately. To find the minimum we would have then to round 2^k times resulting in an algorithm that would have used $\mathcal{O}(k \cdot 2^k)$ arithmetic operations, which is of course preferable if k is small compared to N . Back to our question of computing the limit we have

$$\lim_{N \rightarrow \infty} \mu_{\mathcal{C}}(N) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{1 \leq i < \# \mathcal{C}} \vartheta'_i(\mathcal{C}) \quad \text{and} \quad \lim_{n \rightarrow \infty} \tau(N) = \min_{\mathcal{C} \leq \mathcal{C}^f} \lim_{N \rightarrow \infty} \mu_{\mathcal{C}}(N).$$

Furthermore

$$\lim_{N \rightarrow \infty} \sigma^-(N) = \lim_{N \rightarrow \infty} \frac{1}{N \cdot n_K} \sum_{1 \leq i \leq K} a_i \cdot c_i \quad \text{and} \quad \lim_{N \rightarrow \infty} \sigma^+(N) = \lim_{N \rightarrow \infty} \frac{Mc_K}{Nn_K}$$

	\mathcal{D}_1	\mathcal{D}_2	\mathcal{D}_3	\mathcal{D}_4	\mathcal{D}_5	\mathcal{D}_6
$\lim_{N \rightarrow \infty} \gamma_{\mathcal{D}}^-$	20.81	20.58	20.70	20.56	20.00	19.81
$\lim_{N \rightarrow \infty} \gamma_{\mathcal{D}}^+$	35.99	35.66	35.82	35.63	34.80	34.51

Table 7.4.1: Bounds on the limit of the runtime gain (in percent) for the various data sets

Together

$$\lim_{N \rightarrow \infty} \gamma^-(N) = \min_{\mathcal{C} \preceq \mathcal{C}^f} 1 - \frac{n_K \sum_{1 \leq i < \#\mathcal{C}} \vartheta'_i(\mathcal{C})}{\sum_{1 \leq i \leq K} c_i \cdot a_i}$$

and

$$\lim_{N \rightarrow \infty} \gamma^+(N) = \min_{\mathcal{C} \preceq \mathcal{C}^f} 1 - \frac{n_K \sum_{1 \leq i < \#\mathcal{C}} \vartheta'_i(\mathcal{C})}{Mc_K}.$$

Table 7.4.1 shows the results for our six test sets. We observe again that the corresponding values for the different data sets are very similar. Thus it seems that only the distribution of the inputs is crucial for the outcome of the optimization.

7.5. Connection to the theoretical results

It is striking to observe that the distribution in the statistical analyses from Section 7.3 directly resemble the results we obtained in Chapter 6. Indeed the whole work we were doing there was motivated by the statistical analyses from this chapter. Unfortunately, there is still a gap between the results presented in Chapter 6 and the results presented here.

The reason is that the data sets we considered in Section 7.3 are generated by evaluating homogeneous polynomials in a lattice sieving fashion and dividing out all the small primes, see Section 3.4.3. Afterwards, a procedure is applied that in advance tries to filter out those candidates for which the probability for the Elliptic Curve Method 3.5.9 to factor the candidate was too low. The distribution that is obtained in such a way differs from the one studied in Chapter 6 in the sense that it resembles the inputs to the cofactorization step, while our results from the previous chapter describe those integers that really help in the factorization effort. Working out the details is still to be done and could be a starting point for future work, see Chapter 11.

Chapter 8

RSA integers

In this chapter we describe a theoretical framework that is capable of analyzing rigorously various definitions for integers used in the RSA crypto system (see Section 5.2). The results were first published in a conference version at AfricaCrypt 2011 in Dakar, Senegal (see Loebenberger & Nüsken 2011a). The results presented here were clearly influenced by our coauthor, but most of the details in this section (including the proof of the central prime sum approximation lemma) are our own findings, even though they would not be in the same shape without our coauthor.

8.1. Framework

An *RSA integer* is an integer that is suitable as a modulus for the RSA crypto system as proposed by Rivest, Shamir & Adleman (1978) and described in Section 5.2. On their page 6 they write:

“You first compute n as the product of two primes p and q :

$$n = p \cdot q.$$

These primes are very large, ‘random’ primes. Although you will make n public, the factors p and q will be effectively hidden from everyone else due to the enormous difficulty of factoring n .”

Also in earlier literature such as Ellis (1970) or Cocks (1973) one does not find any further restrictions. In subsequent literature people define RSA integers similarly to Rivest, Shamir & Adleman: Crandall & Pomerance (2005) note that it is “fashionable to select approximately equal primes but sometimes one runs some further safety tests”. In more applied works such

as Schneier (1996) or Menezes *et al.* (1997) one can read that for maximum security one chooses two (distinct) primes of equal length. Also von zur Gathen & Gerhard (1999) follow a similar approach. Decker & Moree (2008) define an RSA integer to be a product of two primes p and q such that $p < q < rp$ for some parameter $r \in \mathbb{R}_{>1}$. Real world implementations, however, require *concrete algorithms* that specify in detail how to generate RSA integers. This has led to a variety of standards, notably the standards PKCS#1 (Jonsson & Kaliski 2003), ISO 18033-2 (International Organization for Standards 2006), IEEE 1363-2000 (IEEE working group 2000), ANSI X9.44 (Accredited Standards Committee X9 2007), FIPS 186-3 (NIST 2009), the standard of the RSA foundation (RSA Laboratories 2000), the standard set by the German Bundesnetzagentur (Wohlmacher 2009), and the standard resulting from the European NESSIE project (NESSIE working group 2003). All of those standards define more or less precisely how to generate RSA integers and all of them have substantially different requirements. This reflects the intuition that it does not really matter how one selects the prime factors in detail, the resulting RSA modulus will do its job. But what is needed to show that this is really the case?

Following Brandt & Damgård (1993) a quality measure of a generator is the entropy of its output distribution. In abuse of language we will most of the time talk about the *output entropy* of an algorithm. To compute it, we need estimates of the probability that a certain outcome is produced. This in turn needs a thorough analysis of how one generates RSA integers of a specific form. If we can show that the outcome of the algorithm is roughly uniformly distributed, the output entropy is closely related to the count of RSA integers it can produce. It will turn out that in all reasonable setups this count is essentially determined by the desired length of the output, see Section 8.5. For primality tests there are several results in this direction (see for example Joye & Paillier 2006) but we are not aware of any related work analyzing the output entropy of algorithms for generating RSA integers.

Another requirement for the algorithm is that the output should be ‘hard to factor’. Since this statement does not even make sense for a single integer, this means that one has to show that the restrictions on the shape of the integers the algorithm produces do not introduce any further possibilities for an attacker. To prove this, a *reduction* has to be given that reduces the problem of factoring the output to the problem of factoring a product of two primes of similar size, see Section 8.6. Also there it is necessary to have results on the count of RSA integers of a specific form to make the reduction work. As for the entropy estimations, we do not know any related work on this.

In the following section we will develop a formal framework that can handle all possible definitions for RSA integers. After discussing the necessary number theoretic tools in Section 8.3, we give explicit formulas for the count of such integers which will be used later for entropy estimations of the various standards for RSA integers. In Section 8.4 we show how our general framework can be instantiated, yielding natural definitions for several types of RSA integers (as used later in the standards). Section 10.1 gives a short overview on generic constructions for fast algorithms that generate such integers almost uniformly. At this point we will have described all necessary techniques to compute the output entropy, which we discuss in Section 10.2. The following section resolves the second question described above by giving a reduction from factoring special types of RSA integers to factoring a product of two primes of similar size. We finish by applying our results to various standards for RSA integers in Section 10.4.

8.2. RSA integers in general

If one generates an RSA integer it is necessary to select for each choice of the security parameter the prime factors from a certain region. This security parameter is typically an integer k that specifies (roughly) the size of the output. We use a more general definition by asking for integers from the interval $]x/r, x]$, given a *real* bound x and a parameter r (possibly depending on x). Clearly, this can also be used to model the former selection process by setting $x = 2^k - 1$ and $r = 2$. Let us in general introduce a *notion of RSA integers with tolerance r* as a family

$$\mathcal{A} := \langle \mathcal{A}_x \rangle_{x \in \mathbb{R}_{>1}}$$

of subsets of the positive quadrant $\mathbb{R}_{>1}^2$, where for every $x \in \mathbb{R}_{>1}$

$$\mathcal{A}_x \subseteq \left\{ (y, z) \in \mathbb{R}_{>1}^2 \mid \frac{x}{r} < yz \leq x \right\}.$$

The tolerance r shall always be larger than 1. We allow here that r varies (slightly) with x , which of course includes the case that r is a constant. Typical values used for RSA are $r = 2$ or $r = 4$ which fix the bit-length of the modulus more or less. Now an \mathcal{A} -integer n of size x — for use as a modulus in RSA — is a product $n = pq$ of a prime pair $(p, q) \in \mathcal{A}_x \cap (\mathbb{P} \times \mathbb{P})$, where \mathbb{P} denotes the set of primes. They are counted by the associated *prime pair counting function* $\#\mathcal{A}$ for the notion \mathcal{A} :

$$\begin{aligned} \#\mathcal{A}: \quad \mathbb{R}_{>1} &\longrightarrow \mathbb{N}, \\ x &\longmapsto \#\{(p, q) \in \mathbb{P} \times \mathbb{P} \mid (p, q) \in \mathcal{A}_x\}. \end{aligned}$$

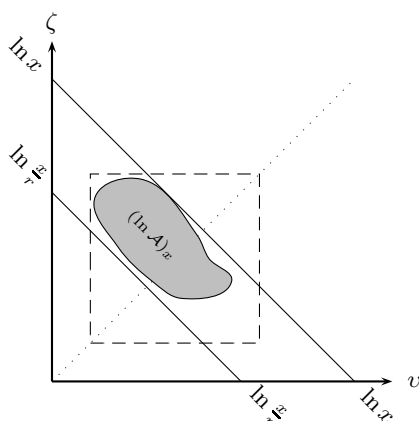


Figure 8.2.1: A generic notion of RSA integers with tolerance r . The gray area shows the parts of the $(\ln y, \ln z)$ -plane which is counted. It lies between the tolerance bounds $\ln x$ and $\ln \frac{x}{r}$. The dashed lines show boundaries as imposed by $[c_1, c_2]$ -balanced. The dotted diagonal marks the criterion for symmetry.

Thus every \mathcal{A} -integer $n = pq$ is counted once or twice in $\#\mathcal{A}(x)$ depending on whether only $(p, q) \in \mathcal{A}_x$ or also $(q, p) \in \mathcal{A}_x$, respectively. We call a notion *symmetric* if for all choices of the parameters the corresponding area in the (y, z) -plane is symmetric with respect to the main diagonal, i.e. that $(y, z) \in \mathcal{A}_x$ implies also $(z, y) \in \mathcal{A}_x$. If to the contrary $(y, z) \in \mathcal{A}_x$ implies $(z, y) \notin \mathcal{A}_x$ we call the notion *antisymmetric*. If we are only interested in RSA integers we can always require symmetry or antisymmetry, yet many algorithms proceed in an asymmetric way.

Certainly, we will also need restrictions on the shape of the area we are analyzing: If one considers any notion of RSA integers and throws out exactly the prime pairs one would be left with a prime-pair-free region and any approximation for the count of such a notion based on the area would necessarily have a tremendously large error term. However, for practical applications it turns out that it is enough to consider regions of a very specific form. Actually, we will most of the time have regions whose boundary can be described by graphs of certain smooth functions.

For RSA, people usually prefer two prime factors of roughly the same size, where size is understood as bit length. Accordingly, we call a notion of

RSA integers $[c_1, c_2]$ -balanced iff additionally for every $x \in \mathbb{R}_{>1}$

$$\mathcal{A}_x \subseteq \left\{ (y, z) \in \mathbb{R}_{>1}^2 \mid y, z \in [x^{c_1}, x^{c_2}] \right\},$$

where $0 < c_1 \leq c_2$ can be thought of as constants or — more generally — as smooth functions in x defining the amount of allowed divergence subject to the side condition that x^{c_1} tends to infinity when x grows. If $c_1 > \frac{1}{2}$ then \mathcal{A}_x is empty, so we will usually assume $c_1 \leq \frac{1}{2}$. In order to prevent trial division from being a successful attacker it would be sufficient to require $y, z \in \Omega(\ln^k x)$ for every $k \in \mathbb{N}$. Our stronger requirement still seems reasonable and indeed equals the condition Maurer (1995) required for secure RSA moduli, as the supposedly most difficult factoring challenges stay within the range of our attention. As a side-effect this greatly simplifies our approximations later. The German Bundesnetzagentur (see Wohlmacher 2009) uses a very similar restriction in their algorithm catalog. There it is additionally required that the primes p and q are not too close to each other. We ignore this issue here, since the probability that two primes are *very close* to each other would be tiny if the notion from which (p, q) was selected is sufficiently large. If necessary, we are able to modify our notions such that also this requirement is met. We can — for a fixed choice of parameters — easily visualize any notion of RSA integers by the corresponding region \mathcal{A}_x in the (y, z) -plane. It is favorable to look at these regions in logarithmic scale, i.e. if we write $y = e^v$ and $z = e^\zeta$ and denote by $(\ln \mathcal{A})_x$ the region in the (v, ζ) -plane corresponding to the region \mathcal{A}_x in the (y, z) -plane, i.e. $(v, \zeta) \in (\ln \mathcal{A})_x : \Leftrightarrow (y, z) \in \mathcal{A}_x$, we obtain a picture like in Figure 8.2.1.

Often the considered integers $n = pq$ are also subject to further side conditions, like $\gcd((p-1)(q-1), e) = 1$ for some fixed public RSA exponent e . Most of the number theoretic work below can easily be adapted, but for simplicity of exposition we will often present our results without those further restrictions and just point out when necessary how to incorporate such additional properties.

In order to count the number of \mathcal{A} -integers we have to evaluate

$$\#\mathcal{A}(x) = \sum_{\substack{(p,q) \in \mathcal{A}_x \\ p,q \in \mathbb{P}}} 1.$$

If we follow the intuitive view that a randomly generated number n is prime with probability $\frac{1}{\ln n}$, we expect that we have to evaluate integrals like

$$\iint_{\mathcal{A}_x} \frac{1}{\ln y \ln z} dz dy,$$

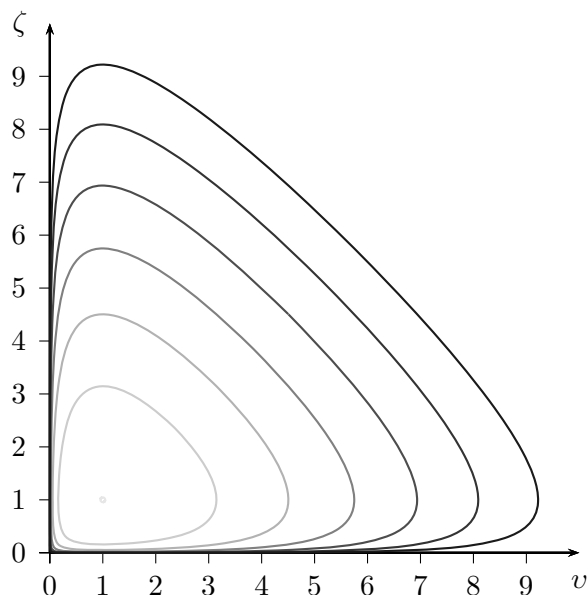


Figure 8.2.2: Levels e^k of the function $\frac{e^{v+\zeta}}{v\zeta}$ for $k \in \{2 + \varepsilon, 3, \dots, 8\}$. The darker the line the higher is the value of k .

while carefully considering the error between those integrals and the above sums. In logarithmic scale we obtain expressions of the form

$$\iint_{(\ln \mathcal{A})_x} \frac{e^{v+\zeta}}{v\zeta} d\zeta dv.$$

To get an understanding of these functions, in Figure 8.2.2 the contour lines of the inner function is depicted. From the figure we observe that pairs (v, ζ) , where $v + \zeta$ is large has a higher weight in the overall count.

As we usually deal with balanced notions the considered regions are somewhat centered around the main diagonal. We will show in Section 8.6 that if factoring products of two primes is hard then it is also hard to factor integers generated from such notions.

8.3. Toolbox

We will now develop the necessary number theoretic concepts to obtain formulas for the count of RSA integers that will later help us to estimate the output entropy of the various standards for RSA integers, see Chapter 10. In related articles, like Decker & Moree (2008) one finds counts for *one particular* definition of RSA integers. We believe that in the work presented

here for the first time a sufficiently general theorem is established that allows to compute the number of RSA integers for *all* reasonable definitions.

We assume the Riemann Hypothesis 2.2.14 throughout the entire chapter. The main terms are the same without this assumption, but the error bounds one obtains are then much weaker. We first state a quite technical lemma, very similar to Lemma 6.2.1, that enables us to do our approximations:

LEMMA 8.3.1 (Prime sum approximation). *Let f, \tilde{f}, \hat{f} be functions $[B, C] \rightarrow \mathbb{R}_{>1}$, where $B, C \in \mathbb{R}_{>1}$ such that \tilde{f} and \hat{f} are piece-wise continuous, $\tilde{f} + \hat{f}$ is either weakly decreasing, weakly increasing, or constant, and for $p \in [B, C]$ we have the estimate*

$$|f(p) - \tilde{f}(p)| \leq \hat{f}(p).$$

Further, let $\hat{E}(p)$ be a positive valued, continuously differentiable function of p bounding $|\pi(p) - \text{li}(p)|$ on $[B, C]$. (For example, under the Riemann Hypothesis 2.2.14 we can take $\hat{E}(p) = \frac{1}{8\pi}\sqrt{p} \ln p$ provided $B \geq 1451$.) Then

$$\left| \sum_{p \in \mathbb{P} \cap [B, C]} f(p) - \tilde{g} \right| \leq \hat{g}$$

with

$$\begin{aligned} \tilde{g} &= \int_B^C \frac{\tilde{f}(p)}{\ln p} dp, \\ \hat{g} &= \int_B^C \frac{\hat{f}(p)}{\ln p} dp + 2(\tilde{f} + \hat{f})(B)\hat{E}(B) + 2(\tilde{f} + \hat{f})(C)\hat{E}(C) + \int_B^C (\tilde{f} + \hat{f})(p)\hat{E}'(p) dp. \end{aligned}$$

In the special case that $\tilde{f} + \hat{f}$ is constant we have the better bound

$$\hat{g} = \int_B^C \frac{\hat{f}(p)}{\ln p} dp + (\tilde{f} + \hat{f})(B)(\hat{E}(B) + \hat{E}(C)).$$

PROOF. The proof can be done analogously to the proof of Lemma 6.2.1. \square

Next we formulate a lemma specialized to handle RSA notions. We cannot expect to obtain an approximation of the number of prime pairs by the area of the region unless we make certain restrictions.

The following definition describes the restrictions that we use. As the reader will notice, it essentially enforces a certain monotonicity that allows the error estimation.

DEFINITION 8.3.2. Let \mathcal{A} be a notion of RSA integers with tolerance r .

- (i) The notion \mathcal{A} is graph-bounded if and only if there are (at least) integrable boundary functions $B_1, C_1: \mathbb{R}_{>1} \rightarrow \mathbb{R}_{>1}$ and $B_2, C_2: \mathbb{R}_{>1}^2 \rightarrow \mathbb{R}_{>1}$ such that we can write

$$\mathcal{A}_x = \left\{ (y, z) \in \mathbb{R}_{>1}^2 \mid \begin{array}{l} B_1(x) < y \leq C_1(x) \\ B_2(y, x) < z \leq C_2(y, x) \end{array} \right\},$$

where for all $x \in \mathbb{R}_{>1}$ and all $y \in]B_1(x), C_1(x)[$ we have $1 < B_1(x) \leq C_1(x) \leq x$ and $1 < B_2(y, x) < C_2(y, x) \leq x$.

- (ii) The notion \mathcal{A} is monotone at x (relative to the error bound \widehat{E}) for some $x \in \mathbb{R}_{>1}$ if and only if it is graph-bounded and the function

$$\int_{B_2(p, x)}^{C_2(p, x)} \frac{1}{\ln q} dq + \widehat{E}(B_2(p, x)) + \widehat{E}(C_2(p, x))$$

is either weakly increasing, weakly decreasing, or constant as a function in p restricted to the interval $[B_1(x), C_1(x)]$. If not mentioned otherwise we refer to the error bound given by $\widehat{E}(p) = \frac{1}{8\pi} \sqrt{p} \ln p$.

We call the notion \mathcal{A} monotone if and only if it is monotone at each $x \in \mathbb{R}_{>1}$ where $\mathcal{A}_x \neq \emptyset$.

- (iii) The notion \mathcal{A} is piece-wise monotone iff there is a parameter $m \in \mathbb{N}$ such that

$$\mathcal{A}_x := \bigsqcup_{j=1}^m \mathcal{A}_{j,x},$$

where $\mathcal{A}_{j,\cdot}$ are all monotone notions of RSA integers of tolerance r . Note that we may also allow m depending on x . In the light of a multi-application of Lemma 8.3.5 we would be on the safe side if we require $m \in \ln^{\mathcal{O}(1)} x$. At the extreme $m \in o\left(c_1 x^{\frac{1-c}{4}} \ln x\right)$ with $c = \max(2c_2 - 1, 1 - 2c_1)$ is necessary for any meaningful result generalizing Lemma 8.3.5.

For (i) note that $B_1(x) = C_1(x)$ allows to describe an empty set \mathcal{A}_x , and otherwise the inequality $B_2(y, x) \neq C_2(y, x)$ makes sure that all four bounding functions are determined by \mathcal{A}_x as long as $y \in]B_1(x), C_1(x)[$. This condition enforces that \mathcal{A}_x is (path) connected. We do not need that but also it does no harm. As in particular (ii) is rather weird to verify we provide an easily checkable, sufficient condition for monotonicity of a notion.

LEMMA 8.3.3. Assume \mathcal{A} is a graph-bounded notion of RSA integers with tolerance r given by continuously differentiable functions $B_1, C_1: \mathbb{R}_{>1} \rightarrow \mathbb{R}_{>1}$ and $B_2, C_2: \mathbb{R}_{>1}^2 \rightarrow \mathbb{R}_{>1}$. Finally, let $x \in \mathbb{R}_{>1}$ be such that

- the function $B_2(p, x)$ is weakly decreasing in p and
- the function $C_2(p, x)$ is weakly increasing in p

for $p \in]B_1(x), C_1(x)]$, or vice versa. As usual let $\widehat{E}(p)$ be the function given by $\widehat{E}(p) = \frac{1}{8\pi} \sqrt{p} \ln p$. Then the notion \mathcal{A} is monotone at x (relative to \widehat{E}).

PROOF. The goal is to show that the function

$$h(p) := \int_{B_2(p,x)}^{C_2(p,x)} \frac{1}{\ln q} dq + \widehat{E}(B_2(p,x)) + \widehat{E}(C_2(p,x))$$

is weakly increasing or weakly decreasing in p . We write $B_2'(p, x)$ and $C_2'(p, x)$, respectively, for the derivative with respect to p . Note that

$$h'(p) := \underbrace{\left(\frac{1}{\ln C_2(p,x)} + \frac{2 + \ln C_2(p,x)}{16\pi \sqrt{C_2(p,x)}} \right)}_{>0} C_2'(p,x) - \underbrace{\left(\frac{1}{\ln B_2(p,x)} - \frac{2 + \ln B_2(p,x)}{16\pi \sqrt{B_2(p,x)}} \right)}_{>0} B_2'(p,x).$$

Some calculus shows that the second underbraced term is always positive since $B_2(p, x) > 1$. Thus if $B_2(p, x)$ is weakly decreasing we have by assumption that $C_2(p, x)$ is weakly increasing and $h(p)$ is weakly increasing. If on the other hand $B_2(p, x)$ is weakly increasing, it follows that $C_2(p, x)$ is weakly decreasing and $h(p)$ is weakly decreasing. \square

Clearly, the conditions of the lemma are not necessary. We can easily extend it, for example, as follows:

LEMMA 8.3.4. Assume \mathcal{A} is a graph-bounded notion of RSA integers with tolerance r given by continuously differentiable functions $B_1, C_1: \mathbb{R}_{>1} \rightarrow \mathbb{R}_{>1}$ and $B_2, C_2: \mathbb{R}_{>1}^2 \rightarrow \mathbb{R}_{>1}$. Further, individually for each $x \in \mathbb{R}_{>1}$, the functions $B_2(p, x)$ and $C_2(p, x)$ are both weakly increasing in p for $p \in]B_1(x), C_1(x)]$. Then there are two monotone notions \mathcal{A}^1 and \mathcal{A}^2 with tolerance r , both having $\mathcal{A}_x^i \subseteq \mathbb{R}_{\geq B_1(x)} \times \mathbb{R}_{\geq B_2(B_1(x), x)}$ for all x , such that $\mathcal{A} = \mathcal{A}^1 \setminus \mathcal{A}^2$.

PROOF. Let $A(x) := B_2(B_1(x), x)$. We define two $[c_1, c_2]$ -balanced graph-bounded notions $\mathcal{A}^1, \mathcal{A}^2$ of RSA integers by the following: the first notion \mathcal{A}^1 is defined by the functions $B_1^1 := B_1, C_1^1 := C_1, B_2^1(p, x) := A(x)$ and $C_2^1 := C_2$. The second notion \mathcal{A}^2 is defined by the functions $B_1^2 := B_1, C_1^2 := C_1, B_2^2(p, x) := A(x)$ and $C_2^2 := B_2$. Since $x/r < B_1(x)B_2(B_1(x), x) = B_1(x)A(x)$ both new notions have tolerance r as well. Then $\mathcal{A}^1, \mathcal{A}^2$ are by Lemma 8.3.3 both monotone and $\mathcal{A} = \mathcal{A}^1 \setminus \mathcal{A}^2$. \square

A similar result with B_2 and C_2 both weakly decreasing is more difficult to obtain while simultaneously retaining the tolerance. A particularly difficult example is the maximal notion \mathcal{M} given by

$$\mathcal{M}_x = \left\{ (y, z) \in \mathbb{R}_{>1}^2 \mid \frac{x}{r} < yz \leq x \text{ and } y, z \geq x^{c_1} \right\}.$$

The following lemma covers all the estimation work.

LEMMA 8.3.5 (Prime sum approximation for monotone notions). *Assume that we have a monotone $[c_1, c_2]$ -balanced notion \mathcal{A} of RSA integers with tolerance r , where $0 < c_1 \leq c_2$. (The values r, c_1, c_2 are allowed to vary with x .) Then under the Riemann Hypothesis 2.2.14 there is a value $\tilde{a}(x) \in \left[\frac{1}{4c_2^2}, \frac{1}{4c_1^2} \right]$ such that*

$$\#\mathcal{A}(x) \in \tilde{a}(x) \cdot \frac{4 \operatorname{area}(\mathcal{A}_x)}{\ln^2 x} + \mathcal{O}\left(c_1^{-1} x^{\frac{3+c}{4}}\right),$$

where $c = \max(2c_2 - 1, 1 - 2c_1)$.

Note that the following proof gives a precise expression for $\tilde{a}(x)$, namely

$$\tilde{a}(x) = \frac{\iint_{\mathcal{A}_x} \frac{1}{\ln p \ln q} \, dp \, dq}{4 \iint_{\mathcal{A}_x} \frac{1}{\ln^2 x} \, dp \, dq}.$$

It turns out that we can only evaluate $\tilde{a}(x)$ numerically in our case and so we tend to estimate also this term. Then we often obtain $\tilde{a}(x) \in 1 + o(1)$. Admittedly, this mostly eats up the advantage obtained by using the Riemann Hypothesis 2.2.14. However, we accept this because it still leaves the option of going through that difficult evaluation and obtain a much more precise answer. If we do not use the Riemann Hypothesis 2.2.14 we need to replace $\mathcal{O}\left(c_1^{-1} x^{\frac{3+c}{4}}\right)$ with $\mathcal{O}\left(\frac{x}{\ln^k x}\right)$ for any $k > 2$ of your choice.

PROOF. Fix any $x \in \mathbb{R}_{>1}$. In case $\operatorname{area}(\mathcal{A}_x) = 0$ the claim holds with any desired $\tilde{a}(x)$ and zero big-Oh term. We can thus assume that the area is

positive. As the statement is asymptotic and x^{c_1} tends to ∞ with x we can further assume that $x^{c_1} \geq 1451$. Abbreviating $\tilde{h}(x) = \frac{4 \text{area}(\mathcal{A}_x)}{\ln^2 x}$, we prove that there exists a value $\tilde{a}(x) \in \left[\frac{1}{4c_2^2}, \frac{1}{4c_1^2}\right]$ such that

$$\left| \#\mathcal{A}(x) - \tilde{a}(x) \cdot \tilde{h}(x) \right| \leq \hat{h}(x)$$

with

$$\hat{h}(x) = \frac{1}{4\pi c_1} \left(7 - 6c_2 + \frac{12}{\ln x} \right) x^{\frac{1+c_2}{2}} + \frac{1}{8\pi^2} \cdot x^{\frac{1}{2} + \frac{2 \ln \ln x}{\ln x}} + \frac{1}{4\pi c_1} \left(1 + \frac{4}{\ln x} \right) x^{1 - \frac{c_1}{2}}.$$

This is slightly more precise and implies the claim.

Since the given notion is $[c_1, c_2]$ -balanced with tolerance r for any $(y, z) \in \mathcal{A}_x$ we have $\frac{x}{r} \leq yz \leq x$ and $y, z \in [x^{c_1}, x^{c_2}]$ which implies $\ln y, \ln z \in [c_1, c_2] \ln x$. Equivalently, we have

$$(8.3.6) \quad x^{c_1} \leq B_1(x) \leq C_1(x) \leq x^{c_2}$$

and for $y \in]B_1(x), C_1(x)[$ we have

$$(8.3.7) \quad \frac{x}{ry} \leq B_2(y, x) < C_2(y, x) \leq \frac{x}{y}$$

and

$$(8.3.8) \quad x^{c_1} \leq B_2(y, x) < C_2(y, x) \leq x^{c_2}.$$

From (8.3.7) we infer that for all $y \in]B_1(x), C_1(x)[$ we have

$$(8.3.9) \quad \frac{x}{r} \leq yB_2(y, x) \leq x \quad \text{and} \quad \frac{x}{r} \leq yC_2(y, x) \leq x.$$

In order to estimate

$$\#\mathcal{A}(x) = \sum_{p \in \mathbb{P} \cap]B_1(x), C_1(x)[} \sum_{q \in \mathbb{P} \cap]B_2(p, x), C_2(p, x)[} 1,$$

we apply Lemma 8.3.1 twice. Since $x^{c_1} \geq 1451$ and so $B_2(p, x) \geq 1451$ for the considered p we obtain for the inner sum

$$\left| \sum_{q \in \mathbb{P} \cap]B_2(p, x), C_2(p, x)[} 1 - \tilde{g}_1(p, x) \right| \leq \hat{g}_1(p, x),$$

where

$$\begin{aligned}\tilde{g}_1(p, x) &= \int_{B_2(p, x)}^{C_2(p, x)} \frac{1}{\ln q} dq, \\ \hat{g}_1(p, x) &= \hat{E}(B_2(p, x)) + \hat{E}(C_2(p, x)),\end{aligned}$$

since we can use the special case of constant functions in Lemma 8.3.1. Because we are working under the restriction that the notion is monotone, i.e. $\tilde{g}_1(p, x) + \hat{g}_1(p, x)$ is monotone, we are able to apply the lemma a second time. Since $x^{c_1} \geq 1451$ and so $B_1(x) \geq 1451$ we obtain

$$\left| \sum_{p \in \mathbb{P} \cap]B_1(x), C_1(x)]} \sum_{q \in \mathbb{P} \cap]B_2(p, x), C_2(p, x)]} 1 - \tilde{g}_2(x) \right| \leq \hat{g}_2(x),$$

where

$$\begin{aligned}\tilde{g}_2(x) &= \int_{B_1(x)}^{C_1(x)} \int_{B_2(p, x)}^{C_2(p, x)} \frac{1}{\ln p \ln q} dq dp, \\ \hat{g}_2(x) &= \frac{1}{8\pi} \int_{B_1(x)}^{C_1(x)} \left(\sqrt{B_2(p, x)} \ln B_2(p, x) + \sqrt{C_2(p, x)} \ln C_2(p, x) \right) \cdot \left(\frac{1}{\ln p} + \frac{\ln p + 2}{2\sqrt{p}} \right) dp \\ &\quad + \frac{1}{4\pi} \sqrt{B_1(x)} \ln B_1(x) \int_{B_2(B_1(x), x)}^{C_2(B_1(x), x)} \frac{1}{\ln q} dq \\ &\quad + \frac{1}{4\pi} \sqrt{C_1(x)} \ln C_1(x) \int_{B_2(C_1(x), x)}^{C_2(C_1(x), x)} \frac{1}{\ln q} dq \\ &\quad + \frac{1}{32\pi^2} \sqrt{B_1(x)} \ln B_1(x) \left(\sqrt{B_2(B_1(x), x)} \ln (B_2(B_1(x), x)) + \sqrt{C_2(B_1(x), x)} \ln (C_2(B_1(x), x)) \right) \\ &\quad + \frac{1}{32\pi^2} \sqrt{C_1(x)} \ln C_1(x) \left(\sqrt{B_2(C_1(x), x)} \ln (B_2(C_1(x), x)) + \sqrt{C_2(C_1(x), x)} \ln (C_2(C_1(x), x)) \right) \\ &\quad + \frac{1}{8\pi} \int_{B_1(x)}^{C_1(x)} \int_{B_2(p, x)}^{C_2(p, x)} \frac{\ln p + 2}{2\sqrt{p} \ln q} dq dp.\end{aligned}$$

It remains to estimate $\tilde{g}_2(x)$ and $\hat{g}_2(x)$ suitably sharp.

For $(p, q) \in \mathcal{A}_x$ we frequently use the estimate $\ln p, \ln q \in [c_1, c_2] \ln x$. For the main term we obtain

$$\tilde{g}_2(x) \in \left[\frac{1}{4c_2^2}, \frac{1}{4c_1^2} \right] \frac{4 \text{ area}(\mathcal{A}_x)}{\ln^2 x}.$$

We also read off the exact expression $\tilde{a}(x) = \frac{\ln^2 x}{4 \text{ area}(\mathcal{A}_x)} \tilde{g}_2(x)$.

We treat the error term $\widehat{g}_2(x)$ part by part. For the first term we obtain

$$\begin{aligned} & \frac{1}{8\pi} \int_{B_1(x)}^{C_1(x)} \left(\sqrt{B_2(p,x)} \ln B_2(p,x) + \sqrt{C_2(p,x)} \ln C_2(p,x) \right) \cdot \left(\frac{1}{\ln p} + \frac{\ln p + 2}{2\sqrt{p}} \right) dp \\ & \leq \frac{1}{4\pi} \int_{x^{c_1}}^{x^{c_2}} \sqrt{\frac{x}{p}} \ln \left(\frac{x}{p} \right) \cdot \frac{3}{\ln p} dp \\ & \leq \frac{3}{4\pi} \frac{1}{c_1 \ln x} \int_{x^{c_1}}^{x^{c_2}} \sqrt{\frac{x}{p}} \ln \left(\frac{x}{p} \right) dp \\ & \leq \frac{3}{2\pi} \frac{1}{c_1} \left(1 - c_2 + \frac{2}{\ln x} \right) x^{\frac{1+c_2}{2}} \in \mathcal{O} \left(c_1^{-1} x^{\frac{1+c_2}{2}} \right), \end{aligned}$$

where we used in the second line that $\frac{\ln p + 2}{2\sqrt{p}} \leq \frac{2}{\ln p}$ for all $p \geq 2$. Basic calculus shows that $\frac{\ln p(\ln p + 2)}{2\sqrt{p}}$ is maximal at $p = \exp(\sqrt{5} + 1)$, where it is less than 1.68. For the fourth line note that

$$\int \sqrt{\frac{x}{p}} \ln \left(\frac{x}{p} \right) dp = 2p\sqrt{\frac{x}{p}} \left(\ln \left(\frac{x}{p} \right) + 2 \right).$$

The definite integral is not greater than this function evaluated at $p = x^{c_2}$ since $c_1 \leq \frac{1}{2}$. Using $c_2 \geq 0$ gives the claim.

The second term yields

$$\begin{aligned} & \frac{1}{8\pi} \sqrt{B_1(x)} \ln B_1(x) \int_{B_2(B_1(x),x)}^{C_2(B_1(x),x)} \frac{1}{\ln q} dq \\ & \leq \frac{1}{8\pi c_1 \ln x} \sqrt{B_1(x)} C_2(B_1(x),x) \ln B_1(x) \\ & \leq \frac{1}{8\pi c_1} x^{\frac{1+c_2}{2}} \in \mathcal{O} \left(c_1^{-1} x^{\frac{1+c_2}{2}} \right), \end{aligned}$$

since we have $\sqrt{B_1(x)C_2(B_1(x),x)}\sqrt{C_2(B_1(x),x)} \leq x^{\frac{1+c_2}{2}}$ and $\ln B_1(x) \leq \ln x$.

Similarly we obtain for the third term

$$\begin{aligned} & \frac{1}{8\pi} \sqrt{C_1(x)} \ln C_1(x) \int_{B_2(C_1(x),x)}^{C_2(C_1(x),x)} \frac{1}{\ln q} dq \\ & \leq \frac{1}{8\pi c_1} x^{\frac{1+c_2}{2}} \in \mathcal{O} \left(c_1^{-1} x^{\frac{1+c_2}{2}} \right), \end{aligned}$$

using $\sqrt{C_1(x)C_2(C_1(x),x)}\sqrt{C_2(C_1(x),x)} \leq x^{\frac{1+c_2}{2}}$ and $\ln C_1(x) \leq \ln x$.

The fourth term yields

$$\begin{aligned} & \frac{1}{32\pi^2} \sqrt{B_1(x)} \ln B_1(x) \left(\sqrt{B_2(B_1(x), x)} \ln B_2(B_1(x), x) + \sqrt{C_2(B_1(x), x)} \ln C_2(B_1(x), x) \right) \\ & \leq \frac{1}{16\pi^2} \sqrt{x} \ln^2 x \in \mathcal{O} \left(c_1^{-1} x^{\frac{1+c_2}{2}} \right), \end{aligned}$$

where we used (8.3.9) and the (very weak) bound $\ln B_1(x), \ln C_2(p, x) \leq \ln x$. The fifth term can be treated similarly. We finish by observing for the sixth term

$$\begin{aligned} & \frac{1}{8\pi} \int_{B_1(x)}^{C_1(x)} \int_{B_2(p,x)}^{C_2(p,x)} \frac{\ln p + 2}{2\sqrt{p} \ln q} dq dp \\ & \leq \frac{1}{8\pi} \frac{1}{c_1 \ln x} \int_{B_1(x)}^{C_1(x)} \int_{B_2(p,x)}^{C_2(p,x)} \frac{\ln p}{\sqrt{p}} dq dp \\ & \leq \frac{1}{8\pi} \frac{1}{c_1 \ln x} \int_{x^{c_1}}^{x^{c_2}} \frac{\ln p}{\sqrt{p}} \int_0^{\frac{x}{p}} dq dp \\ & \leq \frac{1}{8\pi} \frac{1}{c_1 \ln x} \cdot x \cdot \int_{x^{c_1}}^{x^{c_2}} \frac{\ln p}{p^{3/2}} dp \\ & \leq \frac{1}{4\pi} \frac{1}{c_1} \left(1 + \frac{4}{\ln x} \right) x^{1-\frac{c_1}{2}} \\ & \in \mathcal{O} \left(c_1^{-1} x^{1-\frac{c_1}{2}} \right), \end{aligned}$$

using $B_1(x) \geq x^{c_1}$, $c_1 \leq \frac{1}{2}$, and

$$\int \frac{\ln p}{p^{3/2}} dp = \frac{-2(\ln p + 2)}{\sqrt{p}}.$$

This completes the proof. \square

In specific situations one may obtain better estimates. In particular, when we substitute $C_2(p, x)$ by x/p in the estimation of the sixth summand of the error we may loose much.

Of course we can generalize this lemma to notions composed of few monotone ones. We leave the details to the reader. As mentioned before, in many standards the selection of the primes p and q is additionally subject to the side condition that $\gcd((p-1)(q-1), e) = 1$ for some fixed public exponent e of the RSA crypto system. To handle these restrictions, we prove

THEOREM 8.3.10 (Loebenberger & Nüsken 2011a). *Let $e \in \mathbb{N}_{>2}$ be a public RSA exponent. Then we have for the number $\pi_e(x)$ of primes $p \leq x$ with $\gcd(p-1, e) = 1$ that for x tending to infinity*

$$\pi_e(x) \in \frac{\varphi_1(e)}{\varphi(e)} \cdot \text{Li}(x) + \mathcal{O}(\sqrt{x} \ln x),$$

where $\text{Li}(x) = \int_2^x \frac{1}{\ln t} dt$ is the integral logarithm, $\varphi(e)$ is Euler's totient function and

$$(8.3.11) \quad \frac{\varphi_1(e)}{\varphi(e)} = \prod_{\substack{\ell|e \\ \ell \text{ prime}}} \left(1 - \frac{1}{\ell-1}\right).$$

PROOF. We first show that the number of elements in $\mathbb{Z}_e^\times \cap (1 + \mathbb{Z}_e^\times)$ is exactly $\varphi_1(e)$. Write $e = \prod_{\substack{\ell|e \\ \ell \text{ prime}}} \ell^{f(\ell)}$. Observe that by the Chinese Remainder Theorem 3.1.12 we have

$$\mathbb{Z}_e^\times \cap (1 + \mathbb{Z}_e^\times) = \bigoplus_{\substack{\ell|e \\ \ell \text{ prime}}} \left(\mathbb{Z}_{\ell^{f(\ell)}}^\times \cap (1 + \mathbb{Z}_{\ell^{f(\ell)}}^\times)\right)$$

and each factor in this expression has size $(\ell-2)\ell^{f(\ell)-1}$. Multiplying up all factors gives

$$\#(\mathbb{Z}_e^\times \cap (1 + \mathbb{Z}_e^\times)) = \prod_{\substack{\ell|e \\ \ell \text{ prime}}} \left(1 - \frac{1}{\ell-1}\right) \left(1 - \frac{1}{\ell}\right) \ell^{f(\ell)} = \varphi_1(e).$$

To show the result for $\pi_e(x)$ note that by Theorem 2.2.17, a quantitative version of Dirichlet's Theorem 2.1.9 on the number $\pi_{a+e\mathbb{Z}}(x)$ of primes $p \leq x$ with $p \equiv a \in \mathbb{Z}_e$ when $\gcd(a, e) = 1$, we have

$$\pi_{a+e\mathbb{Z}}(x) \in \frac{1}{\varphi(e)} \cdot \text{Li}(x) + \mathcal{O}(\sqrt{x} \ln x)$$

and we have to count $\varphi_1(e)$ residue classes. Summing up everything, we obtain

$$\pi_e(x) \in \frac{\varphi_1(e)}{\varphi(e)} \cdot \text{Li}(x) + \mathcal{O}(\varphi_1(e)\sqrt{x} \ln x),$$

which proves the claim. \square

This theorem shows that the prime pair approximation in Lemma 8.3.5 can be easily adapted to RSA integers whose prime factors satisfy the conditions of Theorem 8.3.10, since the density of such primes differs for every fixed e just by a constant.

8.4. Some common definitions for RSA integers

We will now give formal definitions of three specific notions of RSA integers. In particular, we consider the following example definitions within our framework:

- The number theoretically inspired notion following Decker & Moree. Note that this occurs in no standard and no implementation.
- The simple construction given by just choosing two primes in given intervals. This construction occurs in several standards, like the standard of the RSA foundation (RSA Laboratories 2000), the standard resulting from the European NESSIE project (NESSIE working group 2003) and the FIPS 186-3 standard (NIST 2009). Also open source implementations of `OpenSSL` (Cox *et al.* 2009), `GnuPG` (Skala *et al.* 2009) and the GNU crypto library `GNU Crypto` (Free Software Foundation 2009) use some variant of this construction.
- An algorithmically inspired construction which allows one prime being chosen arbitrarily and the second is chosen such that the product is in the desired interval. This was for example specified as the IEEE standard 1363 (IEEE working group 2000), Annex A.16.11. However, we could not find any implementations following this standard.

8.4.1. A number theoretically inspired notion. In Decker & Moree (2008) on suggestion of Benne de Weger, the number $\mathcal{C}_r(x)$ of RSA integers up to x was defined as the count of numbers whose two prime factors differ by at most a factor r , in formulas

$$\mathcal{C}_r(x) := \# \left\{ n \in \mathbb{N} \mid \begin{array}{l} \exists p, q \in \mathbb{P}: \\ n = pq \wedge p < q < rp \wedge n \leq x \end{array} \right\}.$$

There is also work of Hashimoto (2009), who pursued generalizations of such integers, by allowing the factors to differ by an arbitrary function instead of a factor.

Formulated as a notion of RSA integers in the sense above, we analyze

$$(8.4.1) \quad \mathcal{A}^{\text{DM}(r)} := \left\langle \left\{ (y, z) \in \mathbb{R}^2 \mid \frac{y}{r} < z < ry \wedge \frac{x}{r} < yz \leq x \right\} \right\rangle_{x \in \mathbb{R}_{>1}}.$$

Note that the prime pair counting function of this notion is closely related to the function $\mathcal{C}_r(x)$: Namely we have

$$\#\mathcal{A}^{\text{DM}(r)}(x) = 2 \left(\mathcal{C}_r(x) - \mathcal{C}_r\left(\frac{x}{r}\right) \right) + \left(\pi(\sqrt{x}) - \pi\left(\sqrt{\frac{x}{r}}\right) \right),$$

where the last part is comparatively small. We now analyze the behavior of the function $\#\mathcal{A}^{\text{DM}(r)}(x)$ under the Riemann Hypothesis 2.2.14. Following Decker & Moree (2008), we rewrite

$$(8.4.2) \quad \begin{aligned} \frac{1}{2} \cdot \#\mathcal{A}^{\text{DM}(r)}(x) &= \sum_{p \in \mathbb{P} \cap \left] \frac{\sqrt{x}}{r}, \sqrt{\frac{x}{r}} \right]} \sum_{q \in \mathbb{P} \cap \left] \frac{x}{rp}, rp \right]} 1 \\ &+ \sum_{p \in \mathbb{P} \cap \left] \sqrt{\frac{x}{r}}, \sqrt{x} \right]} \sum_{q \in \mathbb{P} \cap \left] p, \frac{x}{p} \right]} 1 + \frac{\pi(\sqrt{x}) - \pi\left(\sqrt{\frac{x}{r}}\right)}{2}. \end{aligned}$$

With these bounds we obtain using Lemma 8.3.5:

THEOREM 8.4.3 (Loebenberger & Nüsken 2011a). *We have for $\ln r \in o(\ln x)$ under the Riemann Hypothesis 2.2.14 for x tending to infinity*

$$\#\mathcal{A}^{\text{DM}(r)}(x) \in \tilde{a}(x) \frac{4x}{\ln^2 x} \left(\ln r - \frac{\ln r}{r} \right) + \mathcal{O}\left(x^{\frac{3}{4}} r^{\frac{1}{2}}\right)$$

$$\text{with } \tilde{a}(x) \in \left[\left(1 - \frac{\ln r}{\ln x + \ln r}\right)^2, \left(1 + \frac{2 \ln r}{\ln x - 2 \ln r}\right)^2 \right] \subseteq 1 + o(1).$$

You may want to sum this up as $\#\mathcal{A}^{\text{DM}(r)}(x) \in (1 + o(1)) \frac{4x}{\ln^2 x} \left(\ln r - \frac{\ln r}{r} \right)$. However, one then forgoes the option of actually calculating $\tilde{a}(x)$.

PROOF. Consider x large enough such that all sum boundaries are beyond 1451, i.e. $\frac{\sqrt{x}}{r} \geq 1451$. By definition $\mathcal{A}^{\text{DM}(r)}$ is a notion of tolerance r . Further it is $[c_1, c_2]$ -balanced with $c_1 = \log_x\left(\frac{\sqrt{x}}{r}\right) = \frac{1}{2} - \frac{\ln r}{\ln x}$ and $c_2 = \log_x(\sqrt{rx}) = \frac{1}{2} + \frac{\ln r}{2 \ln x}$. As depicted next to (8.4.1), we treat the upper half of the notion as the union of those two notions matching the two double sums in (8.4.2), which both inherit being $[c_1, c_2]$ -balanced of tolerance r . Considering the inner bounds $\frac{x}{rp}$ to rp and p to $\frac{x}{p}$, respectively, as a function of the outer variable p , we observe that the lower and upper bound in each case have opposite monotonicity behavior and thus by Lemma 8.3.3 each part is a monotone notion. We can thus apply Lemma 8.3.5. Since $\ln r \in o(\ln x)$ we have $c_1, c_2 \in \frac{1}{2} + o(1)$, which implies that $\frac{1}{c_i^2} \in 4(1 + o(1))$ for both $i \in \{1, 2\}$. Computing the area of the two parts yields

$$\int_{\frac{\sqrt{x}}{r}}^{\sqrt{\frac{x}{r}}} \int_{\frac{x}{rp}}^{rp} 1 \, dq \, dp = \frac{1}{2} \cdot x \left(1 - \frac{\ln r}{r} - \frac{1}{r} \right)$$

and

$$\int_{\sqrt{\frac{x}{r}}}^{\sqrt{x}} \int_p^{\frac{x}{p}} 1 \, dq \, dp = \frac{1}{2} \cdot x \left(\ln r - 1 + \frac{1}{r} \right).$$

For the error term we obtain $\mathcal{O}(x^{\frac{3}{4}}r^{\frac{1}{2}})$ noting that the number $\pi(\sqrt{x})$ of prime squares up to x is at most \sqrt{x} . \square

Actually, we can prove that the error term is even in $\mathcal{O}(x^{\frac{3}{4}}r^{\frac{1}{4}})$. We lost this in the last steps of the proof of Lemma 8.3.5 when we replaced $C_2(p, x) = rp$ by x/p .

8.4.2. A fixed bound notion. A second possible definition for RSA integers can be stated as follows: We consider the number of integers smaller than a real positive bound x that have exactly two prime factors p and q , both lying in a fixed interval $]B, C]$, in formulas:

$$\pi_{B,C}^2(x) := \# \left\{ n \in \mathbb{N} \left| \begin{array}{l} \exists p, q \in \mathbb{P} \cap]B, C]: \\ n = pq \wedge n \leq x \end{array} \right. \right\}.$$

To avoid problems with rare prime squares, which are also not interesting when talking about RSA integers, we instead count

$$\kappa_{B,C}^2(x) := \# \left\{ (p, q) \in (\mathbb{P} \cap]B, C])^2 \mid pq \leq x \right\}.$$

We discussed such functions in Chapter 6.

In the context of RSA integers we consider the notion

$$(8.4.4) \quad \mathcal{A}^{\text{FB}(r,\sigma)} := \left\langle \left\{ (y, z) \in \mathbb{R}_{>1}^2 \left| \sqrt{\frac{x}{r}} < y, z \leq \sqrt{r^\sigma x} \wedge yz \leq x \right. \right\} \right\rangle_{x \in \mathbb{R}_{>1}}.$$

with $\sigma \in [0, 1]$. The parameter σ describes the (relative) distance of the restriction $yz \leq x$ to the center of the rectangle in which y and z are allowed.

We split the corresponding counting function into two double sums:

$$(8.4.5) \quad \begin{aligned} \#\mathcal{A}^{\text{FB}(r,\sigma)}(x) = & \sum_{p \in \mathbb{P} \cap]\sqrt{\frac{x}{r}}, \sqrt{\frac{x}{r^\sigma}}] } \sum_{q \in \mathbb{P} \cap]\sqrt{\frac{x}{r}}, \sqrt{r^\sigma x}] } 1 \\ & + \sum_{p \in \mathbb{P} \cap]\sqrt{\frac{x}{r^\sigma}}, \sqrt{r^\sigma x}] } \sum_{q \in \mathbb{P} \cap]\sqrt{\frac{x}{r}}, \frac{x}{p}] } 1. \end{aligned}$$

The next theorem follows directly from Loebenberger & Nüsken (2010) but we can also derive it from Lemma 8.3.5 similar to Theorem 8.4.3.

THEOREM 8.4.6 (Loebenberger & Nüsken 2011a). *We have for $\ln r \in o(\ln x)$ under the Riemann Hypothesis 2.2.14 for x tending to infinity*

$$\#\mathcal{A}^{\text{FB}(r,\sigma)}(x) \in \tilde{a}(x) \frac{4x}{\ln^2 x} \left(\sigma \ln r + 1 - \frac{2}{r^{\frac{1-\sigma}{2}}} + \frac{1}{r} \right) + \mathcal{O}\left(x^{\frac{3}{4}} r^{\frac{1}{4}}\right)$$

with $\tilde{a}(x) \in \left[\left(1 - \frac{\sigma \ln r}{\ln x + \sigma \ln r}\right)^2, \left(1 + \frac{\ln r}{\ln x - \ln r}\right)^2 \right] \subseteq 1 + o(1)$.

If we drop the condition $\ln r \in o(\ln x)$, we still obtain the same result, but the asymptotics of the function $\tilde{a}(x)$ changes.

PROOF. Let x be such that all sum boundaries are beyond 1451. By definition $\mathcal{A}^{\text{FB}(r,\sigma)}$ is a notion of tolerance r . Further it is for all $\sigma \in [0, 1]$ clearly $[c_1, c_2]$ -balanced with $c_1 = \log_x \sqrt{\frac{x}{r}} = \frac{1}{2} - \frac{\ln \sqrt{r}}{\ln x}$ and $c_2 = \log_x \sqrt{r^\sigma x} = \frac{1}{2} + \frac{\ln r^{\sigma/2}}{\ln x}$. As depicted next to (8.4.4), we treat the notion as the union of two notions corresponding to the two double sums in (8.4.5), which are both $[c_1, c_2]$ -balanced of tolerance r .

Consider the inner bounds $\sqrt{\frac{x}{r}}$ to $\sqrt{r^\sigma x}$ and $\sqrt{\frac{x}{r}}$ to $\frac{x}{p}$ respectively, as a function of the outer variable p (while σ is fixed): We observe that the lower and upper bound in the first case are constant and in the second case consist of a constant lower bound and an antitone upper bound. Thus by Lemma 8.3.3 each part is a monotone notion and we can apply Lemma 8.3.5. As for the number theoretically inspired notion we have $\frac{1}{c_i^2} \in 4(1 + o(1))$ for both $i \in \{1, 2\}$. Computing the area of the two parts yields

$$\int_{\sqrt{\frac{x}{r^\sigma}}}^{\sqrt{r^\sigma x}} \int_{\sqrt{\frac{x}{r^\sigma}}}^{\frac{x}{p}} 1 \, dq \, dp = x \left(\sigma \ln r + \frac{1}{r^{(1+\sigma)/2}} - \frac{1}{r^{(1-\sigma)/2}} \right)$$

and

$$\int_{\sqrt{\frac{x}{r}}}^{\sqrt{\frac{x}{r^\sigma}}} \int_{\sqrt{\frac{x}{r}}}^{\sqrt{r^\sigma x}} 1 \, dq \, dp = x \left(1 - \frac{1}{r^{(1-\sigma)/2}} - \frac{1}{r^{(1+\sigma)/2}} + \frac{1}{r} \right).$$

For the error term we obtain $\mathcal{O}\left(x^{\frac{3}{4}} r^{\frac{1}{4}}\right)$. □

Note that in Justus (2009) one finds a similar estimate for such kinds of integers. Justus attended a talk of mine on 28 May 2009 at the *cosec* Oberseminar, where I presented our findings. It seems that he then decided to work on his own on the topic and published it in the Albanian Journal of Mathematics. Pieter Moree made us aware of this fact in late 2011. We will now discuss how far our findings differ from Justus' ones: First of all in the work of Justus only two particular notions are discussed. A number theoretically

inspired one in the sense of Decker & Moree and the fixed-bound notion, presented in this section. He does neither provide a result like Lemma 8.3.5 for arbitrary notions nor provides explicit estimates for the error term. Additionally, his findings do not allow to easily modify his estimates in presence of different versions of the Prime Number Theorem 2.1.8. Furthermore, his chapter on RSA generators are mere estimates on the density of RSA integers and not put into any context of specific algorithms. In our work we have a much broader framework in which special cases give the results from Justus (2009).

8.4.3. An algorithmically inspired notion. A third option to define RSA integers is the following notion: Assume you wish to generate an RSA integer between $\frac{x}{r}$ and x , which has two prime factors of roughly equal size. Then algorithmically we might first generate the prime p and afterward select the prime q such that the product is in the correct interval. As we will see later, this procedure does — however — not produce every number with the same probability, see Section 10.1. Formally, we consider the notion

$$(8.4.7) \quad \mathcal{A}^{\text{ALG}(r)} := \left\langle \left\{ (y, z) \in \mathbb{R}_{>1}^2 \left| \begin{array}{l} \frac{\sqrt{x}}{r} < y \leq \sqrt{x}, \\ \frac{x}{ry} < z \leq \frac{x}{y}, \\ \frac{x}{r} < yz \leq x \end{array} \right. \right\} \right\rangle_{x \in \mathbb{R}_{>1}}.$$

We proceed with this notion similar to the previous one. By observing

$$(8.4.8) \quad \begin{aligned} \#\mathcal{A}^{\text{ALG}(r)}(x) &= \sum_{p \in \mathbb{P} \cap \left] \frac{\sqrt{x}}{r}, \sqrt{x} \right]} \sum_{q \in \mathbb{P} \cap \left] \sqrt{x}, \frac{x}{p} \right]} 1 \\ &+ \sum_{p \in \mathbb{P} \cap \left] \frac{\sqrt{x}}{r}, \sqrt{x} \right]} \sum_{q \in \mathbb{P} \cap \left] \frac{x}{rp}, \sqrt{x} \right]} 1, \end{aligned}$$

and again applying Lemma 8.3.5 and Lemma 8.3.3 we obtain

THEOREM 8.4.9 (Loebenberger & Nüsken 2011a). *We have for $\ln r \in o(\ln x)$ under the Riemann Hypothesis 2.2.14 for x tending to infinity:*

$$\#\mathcal{A}^{\text{ALG}(r)}(x) \in \tilde{a}(x) \frac{4x}{\ln^2 x} \left(\ln r - \frac{\ln r}{r} \right) + \mathcal{O}\left(x^{\frac{3}{4}} r^{\frac{1}{2}}\right)$$

with $\tilde{a}(x) \in \left[\left(1 - \frac{2 \ln r}{\ln x + 2 \ln r}\right)^2, \left(1 + \frac{2 \ln r}{\ln x - 2 \ln r}\right)^2 \right] \subseteq 1 + o(1)$.

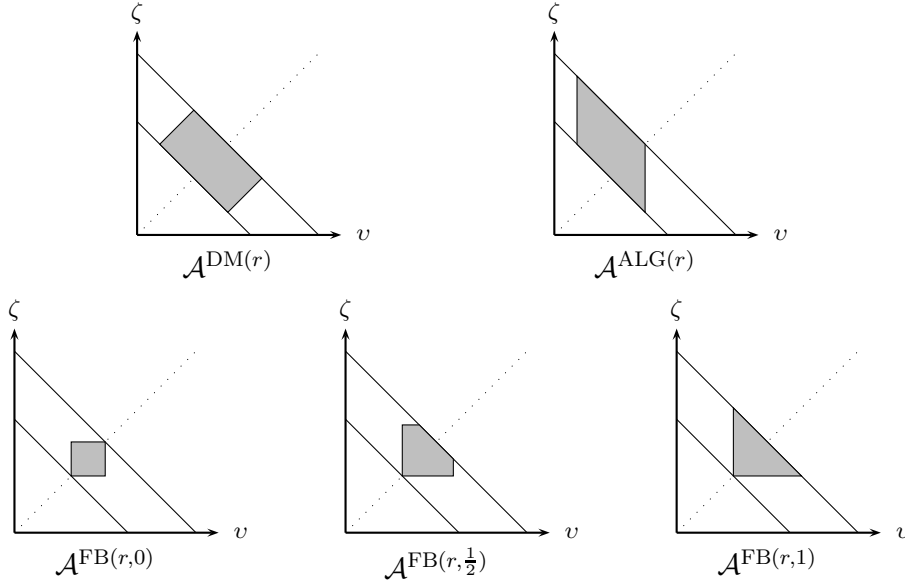


Figure 8.4.1: Three notions of RSA integers.

PROOF. Again let x be such that all sum boundaries are beyond 1451. By definition $\mathcal{A}^{\text{ALG}(r)}$ is a notion of tolerance r . Further it is clearly $[c_1, c_2]$ -balanced with $c_1 = \log_x \frac{\sqrt{x}}{r} = \frac{1}{2} - \frac{\ln r}{\ln x}$ and $c_2 = \log_x r\sqrt{x} = \frac{1}{2} + \frac{\ln r}{\ln x}$. As depicted next to (8.4.7), we treat the notion as the union of two notions corresponding to the two double sums in (8.4.8), which are both $[c_1, c_2]$ -balanced of tolerance r .

If we consider the inner bounds \sqrt{x} to $\frac{x}{p}$ and $\frac{x}{rp}$ to \sqrt{x} , respectively, as a function of the outer variable p , we observe that one of them is all the time constant and by Lemma 8.3.3 each part is a monotone notion. We can thus apply Lemma 8.3.5.

As for the previous notions we have $\frac{1}{c_i^2} \in 4(1 + o(1))$ for both $i \in \{1, 2\}$. Computing the area of the two parts yields

$$\int_{\frac{\sqrt{x}}{r}}^{\sqrt{x}} \int_{\frac{x}{\sqrt{x}}}^{\frac{x}{p}} 1 \, dq \, dp = x \left(1 - \frac{\ln r}{r} - \frac{1}{r} \right)$$

and

$$\int_{\frac{\sqrt{x}}{r}}^{\sqrt{x}} \int_{\frac{x}{rp}}^{\sqrt{x}} 1 \, dq \, dp = x \left(\ln r - 1 + \frac{1}{r} \right).$$

For the error term we obtain $\mathcal{O}\left(x^{\frac{3}{4}} r^{\frac{1}{2}}\right)$. □

Note that we also could have employed Lemma 8.3.4, but in this particular case we decided to use another split of the notion.

The IEEE 1363-2000 standard suggest a slight variant, both generalize to

$$(8.4.10) \quad \mathcal{A}^{\text{ALG}_2(r,\sigma)}(x) := \left\langle \left\{ (y, z) \in \mathbb{R}_{>1}^2 \left| \begin{array}{l} r^{\sigma-1} \sqrt{x} < y \leq r^\sigma \sqrt{x}, \\ \frac{x}{ry} < z \leq \frac{x}{y}, \\ \frac{x}{r} < yz \leq x \end{array} \right. \right\} \right\rangle_{x \in \mathbb{R}_{>1}},$$

with $\sigma \in [0, 1]$. Now, our notion above is $\mathcal{A}^{\text{ALG}_2(r,0)}$, and the IEEE variant is $\mathcal{A}^{\text{ALG}_2(r, \frac{1}{2})}$. By similar reasoning as above we obtain

THEOREM 8.4.11 (Loebenberger & Nüsken 2011a). *Assuming $\ln r \in o(\ln x)$ we have under the Riemann Hypothesis 2.2.14 for x tending to infinity*

$$\#\mathcal{A}^{\text{ALG}_2(r,\sigma)}(x) \in \tilde{a}(x) \frac{4x}{\ln^2 x} \left(\ln r - \frac{\ln r}{r} \right) + \mathcal{O}\left(x^{\frac{3}{4}} r^{\frac{1}{4}}\right),$$

with $\tilde{a}(x) \in \left[\left(1 - \frac{2\sigma' \ln r}{\ln x + 2\sigma' \ln r}\right)^2, \left(1 + \frac{2(1+\sigma) \ln r}{\ln x - 2(1+\sigma) \ln r}\right)^2 \right] \subseteq 1 + o(1)$. \square

8.4.4. Summary. As we see all notions, depicted again in Figure 8.4.1, open a slightly different view. However the outcome is not that different, at least the numbers of described RSA integers are quite close to each other.

Current standards and implementations of various crypto packages mostly use the notions $\mathcal{A}^{\text{FB}(4,0)}$, $\mathcal{A}^{\text{FB}(4,1)}$, $\mathcal{A}^{\text{FB}(2,0)}$ or $\mathcal{A}^{\text{ALG}_2(2,1/2)}$. For details see Section 10.4.

8.5. Arbitrary notions

The preceding examinations show that the order of the analyzed functions differ by a factor that only depends on the notion parameters, i.e. on r and σ , summarizing:

THEOREM. *Assuming $\ln r \in o(\ln x)$ and $r > 1$ and $\sigma \in [0, 1]$ we have for x tending to infinity*

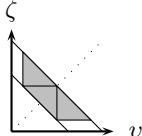
$$(i) \quad \#\mathcal{A}^{\text{DM}(r)}(x) \in (1 + o(1)) \frac{4x}{\ln^2 x} \left(\ln r - \frac{\ln r}{r} \right),$$

$$(ii) \quad \#\mathcal{A}^{\text{FB}(r,\sigma)}(x) \in (1 + o(1)) \frac{4x}{\ln^2 x} \left(\sigma \ln r + 1 - \frac{2}{r^{\frac{1-\sigma}{2}}} + \frac{1}{r} \right),$$

$$(iii) \# \mathcal{A}^{\text{ALG}(r)}(x) \in (1 + o(1)) \frac{4x}{\ln^2 x} \left(\ln r - \frac{\ln r}{r} \right). \quad \square$$

It is obvious that the three considered notions with many parameter choices cover about the same number of integers.

To obtain a much more general result, we consider the following maximal notion

$$(8.5.1) \quad \mathcal{M} := \left\langle \left\{ (y, z) \in \mathbb{R}_{>1}^2 \left| \begin{array}{l} x^{c_1} < y \leq x^{1-c_1}, \\ x^{c_1} < z \leq x^{1-c_1}, \\ \frac{x}{r} < yz \leq x \end{array} \right. \right\} \right\rangle_{x \in \mathbb{R}_{>1}} \quad \begin{array}{c} \zeta \\ \nearrow \\ \searrow \\ v \end{array}$$


All of the notions discussed in Section 8.4 are subsets of this notion. Using the same techniques as above, we obtain:

THEOREM 8.5.2 (Loebenberger & Nüsken 2011a). *We have under the Riemann Hypothesis 2.2.14 for x tending to infinity*

(i) For $c_1 \leq \frac{1}{2} - \frac{\ln r}{2 \ln x}$

$$\# \mathcal{M}(x) \in \tilde{a}(x) \frac{4x}{\ln^2 x} \left((1 - 2c_1) \left(1 - \frac{1}{r} \right) \ln x - 1 + \frac{\ln r + 1}{r} \right) + \mathcal{O} \left(c^{-1} x^{1 - \frac{c_1}{2}} \right),$$

(ii) otherwise we have in the case $\ln r \in \Omega \left(\frac{1 - 2c_1}{\ln^\ell x} \right)$ for some ℓ that

$$\tilde{a}(x) \frac{4x}{\ln^2 x} \left((1 - 2c_1) \ln x + \frac{1}{x^{1-2c_1}} - 1 \right) + \mathcal{O} \left(c_1^{-1} \cdot x^{1 - \frac{c_1}{2}} \ln^{\ell+1} x \right)$$

with $\tilde{a}(x) \in \left[\frac{1}{4(1-c_1)^2}, \frac{1}{4c_1^2} \right]$. In particular for $c_1 \in \frac{1}{2} + o(1)$ we have $\tilde{a}(x) \in 1 + o(1)$.

PROOF. As usual let x be such that all sum boundaries are beyond 1451. By definition \mathcal{M} is a notion of tolerance r . Further it is clearly $[c_1, 1 - c_1]$ -balanced. For $c_1 > \frac{1}{2} - \frac{\ln r}{2 \ln x}$ the result follows directly from Theorem 8.4.6, since \mathcal{M} is simply the fixed bound notion $\mathcal{A}^{\text{FB}(x^{1-2c_1}, 1)}$.

For $c_1 \leq \frac{1}{2} - \frac{\ln r}{2 \ln x}$ we treat the notion as the sum of several monotone, $[c_1, 1 - c_1]$ -balanced notions of tolerance r by triangulating the maximal notion as indicated in the picture above. The number of cuts m we have to make is $m = (1 - 2c_1) \frac{\ln x}{\ln r} \in \mathcal{O} \left(\ln^{\ell+1} x \right)$. This gives the claim. \square

We obtain

THEOREM 8.5.3 (Loebenberger & Nüsken 2011a). *Let $c_1, c_2 \in \frac{1}{2} + o(1)$, $r > 1$ with $\ln r \in \Omega\left(\frac{1-2c_1}{\ln^t x}\right) \cap o(\ln x)$ be possibly x -dependent values, and $a \in]0, 1[$. Consider a piece-wise monotone notion \mathcal{A} of RSA integers with tolerance r such that for large $x \in \mathbb{R}_{>1}$ we have $\text{area } \mathcal{A}_x \geq ax$. Then for x tending to infinity*

$$\#\mathcal{A}(x) = \frac{4x}{\ln^2 x} \cdot \tilde{a}(x)$$

where $\tilde{a}(x) \in o(\ln x)$ and $\tilde{a}(x) \geq a - \varepsilon(x)$ for some $\varepsilon(x) \in o(1)$.

In particular, the prime pair counts of two such notations can differ by at most a factor of order $o(\ln x)$.

PROOF. Let \mathcal{A} be as specified. Assume x to be large enough to grant that $\text{area } \mathcal{A}_x \geq ax$ and $x^{c_1} > 1451$. Without loss of generality we assume $c_1 + c_2 \leq 1$. Otherwise we replace $c_2 = 1 - c_1$. Denote $c := \max(2c_2 - 1, 1 - 2c_1)$, this now is always in $[0, 1]$. By Lemma 8.3.5 we obtain

$$\#\mathcal{A}(x) \geq a \frac{4x}{\ln^2 x} - \hat{a}(x), \quad \hat{a}(x) \in \mathcal{O}\left(x^{\frac{3+c}{4}}\right).$$

To provide an upper bound, we consider the $[c_1, 1 - c_1]$ -balanced notion maximal notion (8.5.1). As mentioned above we have for all $x \in \mathbb{R}_{>1}$ that $\mathcal{A}_x \subseteq \mathcal{M}_x$, and so $\#\mathcal{A}(x) \leq \#\mathcal{M}(x)$. Note that $c_1 \leq \frac{1}{2}$, as otherwise \mathcal{A}_x would be empty rather than having area at least ax . By assumption we have $c_1 \in \frac{1}{2} + o(1)$ and thus $0 \leq 1 - 2c_1 \in o(1)$. Now the claim follows from Theorem 8.5.2 and the assumption $\ln r \in o(\ln x)$. \square

In the following we will analyze the relation between the proposed notions in more detail. Namely, we carefully check how each of the notions can be enclosed in terms of the others. Clearly the fixed bound notions $\mathcal{A}^{\text{FB}(r,\sigma)}$ enclose each other:

LEMMA 8.5.4. *For $r \in \mathbb{R}_{>1}$, $x \in \mathbb{R}_{>1}$ and $\sigma, \sigma' \in [0, 1]$ with $\sigma \leq \sigma'$ we have*

$$\#\mathcal{A}^{\text{FB}(\sqrt{r},1)}(x/\sqrt{r}) \leq \#\mathcal{A}^{\text{FB}(r,\sigma)}(x) \leq \#\mathcal{A}^{\text{FB}(r,\sigma')}(x) \leq \#\mathcal{A}^{\text{FB}(r,1)}(x).$$

PROOF. For the first inequality simply observe that $x/\sqrt{r} \leq x$. The remaining inequalities follow from the fact that $\sqrt{r^\sigma x} \leq \sqrt{r^{\sigma'} x}$ whenever $\sigma \leq \sigma'$. \square

We can also enclose different notions by each other:

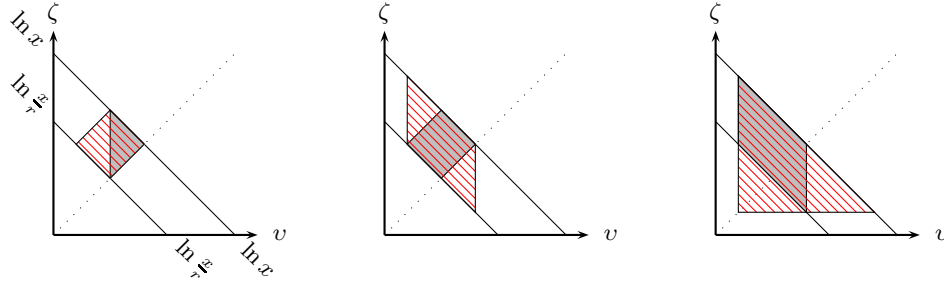


Figure 8.5.1: Enclosing notions of RSA integers using others.

LEMMA 8.5.5. For $r \in \mathbb{R}_{>1}$ and $x \in \mathbb{R}_{>1}$ we have

$$\frac{1}{2} \# \mathcal{A}^{\text{FB}(r,1)}(x) \leq \frac{1}{2} \# \mathcal{A}^{\text{DM}(r)}(x) \leq \# \mathcal{A}^{\text{ALG}(r)}(x) \leq \# \mathcal{A}^{\text{FB}(r^2,1)}(x).$$

PROOF. We prove every inequality separately. For an easier understanding of the proof a look at Figure 8.5.1 is advised:

$\frac{1}{2} \# \mathcal{A}^{\text{FB}(r,1)}(x) \leq \frac{1}{2} \# \mathcal{A}^{\text{DM}(r)}(x)$: Consider the double sum (8.4.5)

$$\frac{1}{2} \# \mathcal{A}^{\text{FB}(r,1)}(x) = \frac{1}{2} \sum_{p \in \mathbb{P} \cap]\sqrt{\frac{x}{r}}, \sqrt{rx}] } \sum_{q \in \mathbb{P} \cap]\sqrt{\frac{x}{r}}, \frac{x}{p}] } 1 = \sum_{p \in \mathbb{P} \cap]\sqrt{\frac{x}{r}}, \sqrt{x}] } \sum_{q \in \mathbb{P} \cap]p, \frac{x}{p}] } 1$$

due to the restriction $p < q$. This is exactly the second summand in (8.4.2).

$\frac{1}{2} \# \mathcal{A}^{\text{DM}(r)}(x) \leq \# \mathcal{A}^{\text{ALG}(r)}(x)$: Consider again the double sum (8.4.2).

We expand the summation area for q (thus increasing the number of prime pairs we count) in order to obtain the sum (8.4.8) for the algorithmic notion: For the first summand we obtain from $p \leq \sqrt{\frac{x}{r}}$ that $rp \leq \frac{x}{p}$ and for the second summand from the same argument that $\frac{x}{rp} \leq p$. The third summand disappears while doing this, since the squares (which are counted by the third summand) are now counted by the second summand. Thus we can bound the whole sum from above by changing the summation area for q in this way.

$\# \mathcal{A}^{\text{ALG}(r)}(x) \leq \# \mathcal{A}^{\text{FB}(r^2,1)}(x)$: We proceed as in the previous step, by replacing in the sum (8.4.8) the summation area for q : Since $p \leq \sqrt{x}$, we obtain $\frac{x}{rp} \geq \frac{\sqrt{x}}{r}$. Now since $\sqrt{x} \leq r\sqrt{x}$ the claim follows. \square

We actually can enclose the Decker & Moree notion even tighter by the fixed bound notion:

LEMMA 8.5.6. For $r \in \mathbb{R}_{>1}$ and $x \in \mathbb{R}_{>1}$ we have

$$\#\mathcal{A}^{\text{FB}(r,1)}(x) \leq \#\mathcal{A}^{\text{DM}(r)}(x) \leq \#\mathcal{A}^{\text{FB}(r^2, \frac{1}{2})}(x).$$

PROOF. Assume $\sqrt{\frac{x}{r}} < p < q \leq \sqrt{rx}$ and $pq \leq x$. Then $\frac{x}{r} < pq \leq x$ and $q \leq rp$. If on the other hand $\frac{x}{r} < pq \leq x$ and $p < q < rp$, then $\frac{x}{r^2} < \frac{1}{r}pq < p^2 < q^2 < rpq \leq rx$ and the claim follows. \square

8.6. Complexity theoretic considerations

We are about to reduce factoring products of two comparatively equally sized primes to the problem of factoring integers generated from a sufficiently large notion. As far as we know there are no similar reductions in the literature.

We consider finite sets $M \subset \mathbb{N} \times \mathbb{N}$, in our situation we actually have only prime pairs. The multiplication map μ_M is defined on M and merely multiplies, that is, $\mu_M: M \rightarrow \mathbb{N}$, $(y, z) \mapsto y \cdot z$. The random variable U_M outputs uniformly distributed values from M . An attacking algorithm F gets a natural number $\mu_M(U_M)$ and attempts to find factors inside M . Its success probability

$$(8.6.1) \quad \text{succ}_F(M) = \text{prob} \left(F(\mu_M(U_M)) \in \mu_M^{-1}(\mu_M(U_M)) \right)$$

measures its quality in any fixed-size scenario. We call a countably infinite family \mathcal{C} of finite sets of pairs of natural numbers *hard to factor* if and only if for any probabilistic polynomial time algorithm F and any exponent s for all but finitely many $M \in \mathcal{C}$ the success probability $\text{succ}_F(M) \leq \ln^{-s} x$ where $x = \max \mu_M(M)$. In other words: The success probability of any probabilistic polynomial time factoring algorithm on a number chosen uniformly from $M \in \mathcal{C}$ is negligible. That is equivalent to saying that the function family $(\mu_M)_{M \in \mathcal{C}}$ is one-way.

Integers generated from a notion \mathcal{A} are *hard to factor* iff for any sequence $(x_i)_{i \in \mathbb{N}}$ tending to infinity the family $(\mathcal{A}_{x_i} \cap (\mathbb{P} \times \mathbb{P}))_{i \in \mathbb{N}}$ is hard to factor. This definition is equivalent to the requirement that for all probabilistic polynomial time machines F , all $s \in \mathbb{N}$, there exists a value $x_0 \in \mathbb{R}_{>1}$ such that for any $x > x_0$ we have $\text{succ}_F(\mathcal{A}_x) \leq \ln^{-s} x$. Since \mathbb{R} is first-countable, both definitions are actually equal. This can be easily shown by considering the functions $g_{s,F}: \mathbb{R}_{>1} \rightarrow \mathbb{R}$, $x \mapsto \text{succ}_F(\mathcal{A}_x) \cdot \ln^s x$. The first definition says that each function $g_{s,F}$ is sequentially continuous (after swapping the initial universal quantifiers). The second definition says that each function $g_{s,F}$ is continuous. In first-countable spaces sequentially continuous is equivalent to continuous.

For any polynomial f we define the set

$$R_f = \{(m, n) \in \mathbb{N} \mid m \leq f(n) \wedge n \leq f(m)\}$$

of f -related positive integer pairs. Denote by $\mathbb{P}^{(m)}$ the set of m -bit primes. We can now formulate the basic assumption that makes the Factorization Problem 2.1.3 more precise:

ASSUMPTION 8.6.2 (Intractability of factoring). *For any unbounded positive polynomial f integers from the f -related prime pair family $(\mathbb{P}^{(m)} \times \mathbb{P}^{(n)})_{(m,n) \in R_f}$ are hard to factor.*

This is exactly the definition given by Goldreich (2001). Note that this assumption implies that factoring in general is hard, and it covers the supposedly hardest factoring instances. Now we are ready to state that integers from all relevant notions are hard to factor.

THEOREM 8.6.3 (Loebenberger & Nüsken 2011a). *Let $\ln r \in \Omega\left(\frac{1-2c_1}{\ln^k x}\right)$ and \mathcal{A} be a piece-wise monotone, $[c_1, c_2]$ -balanced notion for RSA integers of tolerance r with large area, namely, for some k and large x we have $\text{area } \mathcal{A}_x \geq \frac{x}{\ln^k x}$. Assume that factoring is difficult in the sense of Assumption 8.6.2 (or if only integers from the family of linearly related prime pairs are hard to factor). Then integers from the notion \mathcal{A} are hard to factor.*

PROOF. Assume that we have an algorithm F that factors integers generated uniformly from the notion \mathcal{A} . Our goal is to prove that this algorithm also factors polynomially related prime pairs successfully. In other words: Its existence contradicts the assumption that factoring in the form of Assumption 8.6.2 is difficult.

By assumption, there is an exponent s so that for any x_0 there is $x > x_0$ such that the assumed algorithm F has success probability $\text{succ}_F(\mathcal{A}_x) \geq \ln^{-s} x$ on inputs from \mathcal{A}_x . We are going to prove that for each such x there exists a pair (m_0, n_0) , both in the interval $[c_1 \ln x - \ln 2, c_2 \ln x + \ln 2]$, such that F executed with an input from image $\mu_{\mathbb{P}^{m_0}, \mathbb{P}^{n_0}}$ still has success probability at least $\ln^{-(s+k)} x$. By the interval restriction, m_0 and n_0 are polynomially (even linearly) related, namely $m_0 < \frac{2c_2}{c_1} n_0$ and $n_0 < \frac{2c_2}{c_1} m_0$ for large x . So that contradicts Assumption 8.6.2.

First, we cover the set \mathcal{A}_x with small rectangles. Let $S_{m,n} := \mathbb{P}^{(m)} \times \mathbb{P}^{(n)}$ and $I_x := \{(m, n) \in \mathbb{N}^2 \mid S_{m,n} \cap \mathcal{A}_x \neq \emptyset\}$ then

$$(8.6.4) \quad \mathcal{A}_x \cap \mathbb{P}^2 \subseteq \bigsqcup_{(m,n) \in I_x} S_{m,n} =: S_x.$$

Next we give an upper bound on the number $\#S_x$ of prime pairs in the set S_x in terms of the number $\#\mathcal{A}(x)$ of prime pairs in the original notion: First, since each rectangle $S_{m,n}$ extends by a factor 2 along each axis we overshoot by at most that factor in each direction, that is, we have for $c'_1 = c_1 - (1 + 2c_1)\frac{\ln 2}{\ln x}$ and all $x \in \mathbb{R}_{>1}$

$$S_x \subset \mathcal{M}[16r, c'_1]_{4x} = \left\{ (y, z) \in \mathbb{R}^2 \mid y, z \geq \frac{1}{2}x^{c_1} \wedge \frac{x}{4r} < yz \leq 4x \right\}.$$

Provided x is large enough we can guarantee by Theorem 8.5.2 that

$$\#S_x \leq \#\mathcal{M}[16r, c'_1](4x) \leq \frac{8x}{c_1^2 \ln x}.$$

On the other hand side we apply Lemma 8.3.5 for the notion \mathcal{A}_x and use that \mathcal{A}_x is large by assumption. Let $c = \max(2c_2 - 1, 1 - 2c_1)$. Then we obtain for large x with some $e_{\mathcal{A}}(x) \in \mathcal{O}\left(x^{\frac{3+c}{4}}\right)$ the inequality

$$\#\mathcal{A}(x) \geq \frac{\text{area}(\mathcal{A}_x)}{c_2^2 \ln^2 x} - e_{\mathcal{A}}(x) \geq \frac{x}{2c_2^2 \ln^{k+2} x}.$$

Together we obtain

$$(8.6.5) \quad \frac{\#\mathcal{A}(x)}{\#S_x} \geq \frac{c_1^2}{16c_2^2 \ln^{k+1} x} \geq \ln^{-(k+2)} x$$

By assumption we have $\text{succ}_F(\mathcal{A}_x) \geq \ln^{-s} x$ for infinitely many values x . Thus F on an input from S_x still has large success even if we ignore that F might be successful for elements on $S_x \setminus \mathcal{A}_x$,

$$\text{succ}_F(S_x) \geq \text{succ}_F(\mathcal{A}_x) \frac{\#\mathcal{A}(x)}{\#S_x} \geq \ln^{-(k+s+2)} x.$$

Finally choose $(m_0, n_0) \in I_x$ for which the success of F on S_{m_0, n_0} is maximal. Then $\text{succ}_F(S_{m_0, n_0}) \geq \text{succ}_F(S_x)$. Combining with the previous we obtain that for infinitely many x there is a pair (m_0, n_0) where the success $\text{succ}_F(S_{m_0, n_0})$ of F on inputs from S_{m_0, n_0} is still larger than inverse polynomial: $\text{succ}_F(S_{m_0, n_0}) \geq \ln^{-(k+s+2)} x$.

For these infinitely many pairs (m_0, n_0) the success probability of the algorithm F on S_{m_0, n_0} is at least $\ln^{-(k+s+2)} x$ contradicting the hypothesis. \square

All the specific notions that we have found in the literature fulfill the criterion of Theorem 8.6.3. Thus if factoring is difficult in the stated sense then each of them is invulnerable to factoring attacks. Note that the above reduction still works if the primes p, q are due to the side condition $\gcd((p-1)(q-1), e) = 1$ for a fixed integer e (see Theorem 8.3.10). We suspect that this is also the case if p and q are strong primes. Yet, this needs further investigation.

Chapter 9

Generalized RSA integers

We can easily extend the concepts described in Chapter 8 to answer a slightly more general problem of understanding the various definitions for *generalized RSA integers* n , i.e. integers of the form $n = p^i q^j$ with $i, j \in \mathbb{N}_{\geq 1}$. Generalized RSA integers are used for example in fast variants of RSA (Takagi 1998). This system is typically used with integers of the form $p^2 q$, even though it was introduced using integers of the form $p^i q$. Another example is the Okamoto-Uchiyama crypto system (Okamoto & Uchiyama 1998), which uses integers of the form $p^2 q$.

In order to generate generalized RSA integers one might consult one of the various standards for traditional RSA integers and adapt it to generalized ones by changing the routine for prime generation to a routine for prime power generation. By doing so, however, one needs to be careful with the selection of the security parameter, which in some standards is given by the length of the resulting product $n = p^i q^j$, in others by the length of the primes p, q .

We will now sketch an extension of the results from Chapter 8 to generalized RSA integers and show that also for generalized RSA integers the intuition is true that it does not really matter how one generates the integers in detail, the result will do its job: On the number theoretic side we give several counting formulas for generalized RSA integers and show that *all* reasonable notions contain about the same number of integers. On the algorithmic side we note that generating such integers takes always almost the same amount of time and show that factoring integers of a specific form is hard provided factoring prime power products in general is hard. Since all of the aforementioned systems are broken if the underlying integer can be factored efficiently, we thus need to assume that factoring integers of the

form $p^i q^j$ is hard for a specific selection of the parameters i and j in some suitable sense. This, however, is not reasonable in general since for example work of Boneh, Durfee & Howgrave-Graham shows that integers of the form $p^i q$ with very large $i \approx \sqrt{\ln p}$ can be factored efficiently using a lattice reduction algorithm. In our framework we do not allow that i and j grow with $p^i q^j$ which makes this attack not directly applicable to the integers we are considering. In chapter 6 of their paper we read that for $i = 2$ and $j = 1$ it is sufficient to know one third of the bits of q in order to factor the integer efficiently. However, though partial exposure of secrets is of course a realistic assumption in practice (see for example Blömer & May 2003), this is a different issue. This still implies that not all variants of generalized RSA integers are of practical importance in cryptography. We will formulate all of the relevant results for arbitrary *fixed* (i, j) and just point out required restrictions on i and j where necessary.

Though we only deal with asymptotic results here, our intention includes explicit estimations for realistic finite sizes. Experiments show that our estimates start working if the target size for $p^i q^j$ exceeds about 40 bits when $(i, j) = (1, 1)$, 70 bits when $(i, j) = (2, 1)$, 100 bits when $(i, j) = (3, 1)$ or 125 bits when $(i, j) = (3, 2)$. Even this is only achievable using the Riemann Hypothesis 2.2.14.

9.1. Framework and toolbox

We formalize a *notion of generalized RSA integers with tolerance r* as a family

$$\mathcal{A} := \langle \mathcal{A}_x \rangle_{x \in \mathbb{R}_{>1}}$$

of subsets of the positive quadrant $\mathbb{R}_{>1}^2$, where for every $x \in \mathbb{R}_{>1}$

$$\mathcal{A}_x \subseteq \left\{ (y, z) \in \mathbb{R}_{>1}^2 \mid \frac{x}{r} < y^i z^j \leq x \right\}$$

for some $i, j \in \mathbb{N}_{\geq 1}$. For the sake of a simpler presentation we frequently use the negative slope $s := \frac{i}{j}$ and the exponent sum $t := i + j = j(s + 1)$. The tolerance r shall always be larger than 1. We allow here that r varies (slightly) with x , which of course includes the case that r is a constant. Typical values used for RSA are $r = 2$ or $r = 4$ which fix the bitlength of the modulus more or less. Now a *generalized \mathcal{A} -integer of size x* is a product $n = p^i q^j$ of a prime pair (p, q) in \mathcal{A}_x , i.e., $n = p^i q^j$ for some $(p, q) \in \mathcal{A}_x \cap (\mathbb{P} \times \mathbb{P})$, where \mathbb{P} denotes the set of primes.

As before they are counted by the associated *prime pair counting function* $\#\mathcal{A}$ for the notion \mathcal{A} :

$$\#\mathcal{A}: \begin{array}{l} \mathbb{R}_{>1} \longrightarrow \mathbb{N}, \\ x \longmapsto \#\{(p, q) \in \mathbb{P} \times \mathbb{P} \mid (p, q) \in \mathcal{A}_x\}. \end{array}$$

The following central technical lemma covers all the estimation work. For the lemma to apply the considered notions must be monotone. Moreover it is convenient if the notion is suitably *large*, that is, for some k and large x we have $\text{area } \mathcal{A}_x \geq \frac{\text{area } \mathcal{M}_x^{c_1}}{\ln^k x}$, where the maximal $[c_1, \infty]$ -balanced notion \mathcal{M}^{c_1} (with tolerance x) is given by

We obtain the following

LEMMA 9.1.1 (Generalized prime sum approximation). *Assume that we have a large, monotone $[c_1, c_2]$ -balanced notion \mathcal{A} of generalized RSA integers with tolerance r , where $0 < c_1 \leq c_2$. (The values r, c_1, c_2 are allowed to vary with x as long as $c_1^{-1}x^{-\frac{c_1}{2}} \in o(1)$.) Then under the Riemann Hypothesis 2.2.14 there is a value $\tilde{a}(x) \in \left[\frac{1}{t^2 c_2^2}, \frac{1}{t^2 c_1^2}\right]$ such that we have*

$$\#\mathcal{A}(x) \in \left(1 + \mathcal{O}\left(c_1^{-1}x^{-\frac{c_1}{2}}\right)\right) \tilde{a}(x) \cdot \frac{t^2 \text{area}(\mathcal{A}_x)}{\ln^2 x}. \quad \square$$

The proof can be done analogously to the proof of Lemma 9.1.1. Using this lemma, we can easily compute the area of some of the notions similar to those discussed in Chapter 8:

9.2. Some results

The notion $\mathcal{A}^{\text{DM}(r)}$ is given by the sets

$$\mathcal{A}_x^{\text{DM}(r)} := \left\{ (y, z) \in \mathbb{R}^2 \mid y < z < ry \wedge \frac{x}{r} < y^i z^j \leq x \right\}.$$

Recall that we abbreviate the slope $s = \frac{i}{j}$ and the exponent sum $t = i + j$. By first rewriting the count into sums, we observe that

$$\begin{aligned} \#\mathcal{A}^{\text{DM}(r)}(x) = & \sum_{p \in \mathbb{P} \cap \left[x^{\frac{1}{t}} r^{-\frac{i+1}{t}}, x^{\frac{1}{t}} r^{-\frac{i}{t}} \right]} \sum_{q \in \mathbb{P} \cap \left[x^{\frac{1}{j}} p^{-s} r^{-\frac{1}{j}}, rp \right]} 1 + \\ & \sum_{p \in \mathbb{P} \cap \left[x^{\frac{1}{t}} r^{-\frac{i}{t}}, x^{\frac{1}{t}} r^{-\frac{i-1}{t}} \right]} \sum_{q \in \mathbb{P} \cap \left[x^{\frac{1}{j}} p^{-s} r^{-\frac{1}{j}}, x^{\frac{1}{j}} p^{-s} \right]} 1 + \\ & \sum_{p \in \mathbb{P} \cap \left[x^{\frac{1}{t}} r^{-\frac{i-1}{t}}, x^{\frac{1}{t}} \right]} \sum_{q \in \mathbb{P} \cap \left[p, x^{\frac{1}{j}} p^{-s} \right]} 1. \end{aligned}$$

We obtain using Lemma 9.1.1:

THEOREM 9.2.1. *Assuming $r \in \ln^{\mathcal{O}(1)} x$, we have under the Riemann Hypothesis 2.2.14 for x tending to infinity*

(i) *If $s = 1$:*

$$\#\mathcal{A}^{\text{DM}(r)}(x) \in (1 + o(1)) \frac{t^2 x^{\frac{2}{t}}}{2 \ln^2 x} \left(1 - r^{-\frac{2}{t}}\right) \ln r.$$

(ii) *If $s \neq 1$:*

$$\#\mathcal{A}^{\text{DM}(r)}(x) \in (1 + o(1)) \frac{t^2 x^{\frac{2}{t}}}{2 \ln^2 x} \frac{s+1}{s-1} \left(r^{\frac{s-1}{s+1}} - 1\right) \left(1 - r^{-\frac{2}{t}}\right).$$

□

Consider now the notion

$$\mathcal{A}^{\text{FB}(r,\sigma)} := \left\langle \left\{ (y, z) \in \mathbb{R}_{>1}^2 \left| \left(\frac{x}{r}\right)^{\frac{1}{i+j}} < y, z \leq (r^\sigma x)^{\frac{1}{i+j}} \wedge y^i z^j \leq x \right. \right\} \right\rangle_{x \in \mathbb{R}_{>1}}.$$

with $\sigma \in \mathbb{R}_{\geq 0}$. We obtain:

$$\begin{aligned} \#\mathcal{A}^{\text{FB}(r,\sigma)}(x) = & \sum_{p \in \mathbb{P} \cap \left] x^{\frac{1}{t}} r^{-\frac{1}{t}}, x^{\frac{1}{t}} r^{-\frac{1}{st} \min(\sigma, s)} \right]} \sum_{q \in \mathbb{P} \cap \left] x^{\frac{1}{t}} r^{-\frac{1}{t}}, x^{\frac{1}{t}} r^{\frac{1}{t} \min(\sigma, s)} \right]} 1 + \\ & \sum_{p \in \mathbb{P} \cap \left] x^{\frac{1}{t}} r^{-\frac{1}{st} \min(\sigma, s)}, x^{\frac{1}{t}} r^{\frac{1}{t} \min(\sigma, s-1)} \right]} \sum_{q \in \mathbb{P} \cap \left] x^{\frac{1}{t}} r^{-\frac{1}{t}}, x^{\frac{s+1}{t}} y^{-s} \right]} 1. \end{aligned}$$

The following theorem follows directly from Loebenberger & Nüsken (2010) but we can also derive it from Lemma 9.1.1 similar to Theorem 9.2.1.

THEOREM 9.2.2. *Assuming $r \in \ln^{\mathcal{O}(1)} x$, we have under the Riemann Hypothesis 2.2.14 for x tending to infinity*

(i) *If $s = 1$:*

$$\#\mathcal{A}^{\text{FB}(r,\sigma)}(x) \in (1 + o(1)) \frac{t^2 x^{\frac{2}{t}}}{\ln^2 x} \left(\frac{2\sigma \ln r}{t} + 1 - 2r^{\frac{\sigma-1}{t}} + r^{\frac{2}{t}} \right).$$

(ii) If $s \neq 1$:

$$\#\mathcal{A}^{\text{FB}(r,\sigma)}(x) \in (1 + o(1)) \frac{t^2 x^{\frac{2}{t}}}{\ln^2 x} \left(f_s + f_{s-1} + r^{-\frac{2}{t}} \right),$$

$$\text{where } f_z := \frac{s}{s-1} r^{\frac{\min(\sigma,z)(s-1)}{st}} - r^{\frac{\min(\sigma,z)-1}{t}}.$$

□

Consider the family of notions $\mathcal{A}^{\text{ALG}_2(r,\sigma)}$ given by the sets

$$\mathcal{A}_x^{\text{ALG}_2(r,\sigma)} := \left\{ (y, z) \in \mathbb{R}_{>1}^2 \left| \begin{array}{l} r^{\sigma-1} x^{\frac{1}{i+j}} < y \leq r^\sigma x^{\frac{1}{i+j}} \\ \left(\frac{x}{ry^i} \right)^{\frac{1}{j}} < z \leq \left(\frac{x}{y^i} \right)^{\frac{1}{j}}, \\ \frac{x}{r} < y^i z^j \leq x \end{array} \right. \right\}$$

with $\sigma \in [0, 1]$. We proceed with this notion similar to the previous one. By observing

$$\#\mathcal{A}^{\text{ALG}_2(r,\sigma)}(x) = \sum_{p \in \mathbb{P} \cap \left] r^{\sigma-1} x^{\frac{1}{t}}, r^\sigma x^{\frac{1}{t}} \right]} \sum_{q \in \mathbb{P} \cap \left] \left(\frac{x}{rp^{sj}} \right)^{\frac{1}{j}}, \left(\frac{x}{p^{sj}} \right)^{\frac{1}{j}} \right]} 1,$$

and again applying Lemma 9.1.1 and Lemma 8.3.3, we obtain after splitting as depicted next to Section 9.2 the notion $\mathcal{A}^{\text{ALG}_2(r,\sigma)}$ into $2i$ large, monotone notions.

THEOREM 9.2.3. *Assuming $r \in \ln^{\mathcal{O}(1)} x$, we have under the Riemann Hypothesis 2.2.14 for x tending to infinity*

(i) If $s = 1$:

$$\#\mathcal{A}^{\text{ALG}_2(r,\sigma)}(x) \in (1 + o(1)) \frac{t^2 x^{\frac{2}{t}}}{\ln^2 x} \left(1 - r^{-\frac{2}{t}} \right) \ln r$$

(ii) If $s \neq 1$:

$$\#\mathcal{A}^{\text{ALG}_2(r,\sigma)}(x) \in (1 + o(1)) \frac{t^2 x^{\frac{2}{t}}}{\ln^2 x} \left(1 - r^{\frac{s+1}{t}-1} \right) \left(1 - r^{\frac{s+1}{t}} \right) r^{(1-s)\sigma - \frac{s+1}{t}}.$$

□

Chapter 10

Analyzing standards for RSA integers

We will now apply our results from Chapter 8 to analyze concrete standards and implementation around. Also these results were first published in a conference version at AfricaCrypt 2011 in Dakar, Senegal (see Loebenberger & Nüsken 2011a). The full version is submitted to the Journal of Cryptology for publication (see Loebenberger & Nüsken 2011b). Our coauthor suggested to analyze different standards and implementations, but most of the details in this section are our own findings.

10.1. Generating RSA integers properly

We first analyze how to generate RSA integers properly. It completes the picture and we found several implementations overlooking this kind of arguments.

We wish that all the algorithms generate integers with the following properties:

- If we fix x we should with overwhelming probability generate integers that are a product of a prime pair in \mathcal{A}_x .
- These integers (not the pairs) should be selected roughly uniformly at random.
- The algorithm should be efficient. In particular, it should need only few primality tests.

For the first point note that we usually use probabilistic primality tests with a very low error probability, for example the Solovay-Strassen Test 3.3.13

or the Strong Test 3.3.20. Deterministic primality tests (like the Miller Primality Test 3.3.22 or the AKS Test 3.3.29) are also available but are at present for these purposes by far too slow.

10.1.1. Rejection sampling. Assume that \mathcal{A} is a $[c_1, c_2]$ -balanced notion of RSA integers with tolerance r . The easiest approach for generating a pair from \mathcal{A} is based on von Neumann's rejection sampling method. For this the following definition comes in handy:

DEFINITION 10.1.1 (Banner). A banner is a graph-bounded notion of RSA integers such that for all $x \in \mathbb{R}_{>1}$ and for every prime $p \in [B_1(x), C_1(x)]$ the number $f_x(p)$ of primes in the interval $[B_2(p, x), C_2(p, x)]$ is almost independent of p in the following sense: $\frac{\max\{f_x(p) \mid p \in [B_1(x), C_1(x)] \cap \mathbb{P}\}}{\min\{f_x(p) \mid p \in [B_1(x), C_1(x)] \cap \mathbb{P}\}} \in 1 + o(1)$.



For example, a rectangular notion, where $B_2(p, x)$ and $C_2(p, x)$ do not depend on p , is a banner. Now given any notion \mathcal{A} of RSA integers we select a banner \mathcal{B} of (almost) minimal area enclosing \mathcal{A}_x . Note that there may be many choices for \mathcal{B} . We can easily generate elements in $\mathcal{B}_x \cap \mathbb{N}^2$: Select first an appropriate $y \in [B_1(x), C_1(x)] \cap \mathbb{N}$, second an appropriate $z \in [B_2(p, x), C_2(p, x)] \cap \mathbb{N}$. By the banner property this chooses (y, z) almost uniformly. We obtain the following straightforward Las Vegas algorithm:

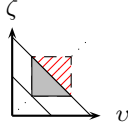
ALGORITHM 10.1.2. Generating an RSA integer (Las Vegas version).

Input: A notion \mathcal{A} , a bound $x \in \mathbb{R}_{>1}$.

Output: An integer $n = pq$ with $(p, q) \in \mathcal{A}_x$.

1. Repeat 2–4
2. Repeat
3. Select (y, z) at random from $\mathcal{B}_x \cap \mathbb{N}^2$ as just described.
4. Until $(y, z) \in \mathcal{A}_x$.
5. Until y prime and z prime.
6. $p \leftarrow y, q \leftarrow z$.
7. Return pq .

The expected repetition count of the inner loop is roughly $\frac{\text{area}(\mathcal{B}_x)}{\text{area}(\mathcal{A}_x)}$. The expected number of primality tests is about $\frac{\text{area}(\mathcal{A}_x)}{\#\mathcal{A}(x)}$. By Theorem 8.5.3 this is for many notions in $\mathcal{O}(\ln^2 x)$. We have seen implementations (for example the one of **GnuPG**) where the inner and outer loop have been exchanged. This increases the number of primality tests by the repetition count of the inner loop. For $\mathcal{A}^{\text{FB}(r,1)}$ this is a factor of about

$$\frac{\#\mathcal{A}^{\text{FB}(r^2,0)}(rx)}{\#\mathcal{A}^{\text{FB}(r,1)}(x)} \sim \frac{\frac{1}{r} - 2 + r}{\ln r + \frac{1}{r} - 1} = \frac{(r-1)^2}{r(\ln r - 1) + 1},$$


which for $r = 2$ is equal to $2.58\bar{1}$ and even worse for larger r . Also easily checkable additional conditions, like $\text{gcd}((p-1)(q-1), e) = 1$, should be checked before the primality tests to improve the efficiency.

10.1.2. Inverse transform sampling. Actually, we would like to avoid generating out-of-bound pairs completely. Then a straightforward attempt to construct such an algorithm would look the following way:

ALGORITHM 10.1.3. Generating an RSA integer (non-uniform version).

Input: A notion \mathcal{A} , a bound $x \in \mathbb{R}_{>1}$.

Output: An integer $n = pq$ with $(p, q) \in \mathcal{A}_x$.

1. Repeat
2. Select y uniformly at random from $\{y \in \mathbb{R} \mid \exists z \in \mathbb{N}: (y, z) \in \mathcal{A}_x\} \cap \mathbb{N}$.
3. Until y prime.
4. $p \leftarrow y$.
5. Repeat
6. Select z uniformly at random from $\{z \in \mathbb{R} \mid (p, z) \in \mathcal{A}_x\} \cap \mathbb{N}$.
7. Until z prime.
8. $q \leftarrow z$.
9. Return pq .

The main problem with Algorithm 10.1.3 is that the produced output typically is not uniform since the sets $\{z \in \mathbb{R} \mid (p, z) \in \mathcal{A}_x\} \cap \mathbb{N}$ do not necessarily have the same cardinality when changing p . To retain uniform selection, we need to select the primes p non-uniformly with the following distribution:

DEFINITION 10.1.4. Let \mathcal{A} be a notion of RSA integers with tolerance r . For every $x \in \mathbb{R}_{>1}$ the associated cumulative distribution function of \mathcal{A}_x is defined as

$$F_{\mathcal{A}_x}: \begin{array}{ll} \mathbb{R} & \longrightarrow [0, 1], \\ y & \longmapsto \frac{\text{area}(\mathcal{A}_x \cap ([1, y] \times \mathbb{R}))}{\text{area}(\mathcal{A}_x)}. \end{array}$$

In fact we should use the function $G_{\mathcal{A}_x}: \mathbb{R} \rightarrow [0, 1], y \mapsto \frac{\#(\mathcal{A}_x \cap (([1, y] \cap \mathbb{P}) \times \mathbb{P}))}{\#\mathcal{A}_x}$, in order to compute the density but computing $G_{\mathcal{A}_x}$ (or its inverse) is tremendously expensive. Fortunately, by virtue of Lemma 8.3.5 we know

that $F_{\mathcal{A}_x}$ approximates $G_{\mathcal{A}_x}$ for monotone, $[c_1, c_2]$ -balanced notions \mathcal{A} quite well. So we use the function $F_{\mathcal{A}_x}$ to capture the distribution properties of a given notion of RSA integers. As can be seen by inspection, in practically relevant examples this function is sufficiently easy to handle. Using this we modify Algorithm 10.1.3 such that each element from \mathcal{A}_x is selected almost uniformly at random:

ALGORITHM 10.1.5. Generating an RSA integer.

Input: A notion \mathcal{A} , a bound $x \in \mathbb{R}_{>1}$.

Output: An integer $n = pq$ with $(p, q) \in \mathcal{A}_x$.

1. Repeat
2. Select y with distribution $F_{\mathcal{A}_x}$ from $\{y \in \mathbb{R} \mid \exists z: (y, z) \in \mathcal{A}_x\} \cap \mathbb{N}$.
3. Until y prime.
4. $p \leftarrow y$.
5. Repeat
6. Select z uniformly at random from $\{z \in \mathbb{R} \mid (p, z) \in \mathcal{A}_x\} \cap \mathbb{N}$.
7. Until z prime.
8. $q \leftarrow z$.
9. Return pq .

As desired, this algorithm generates any pair $(p, q) \in \mathcal{A}_x \cap (\mathbb{P} \times \mathbb{P})$ with almost the same probability. In order to generate y with distribution $F_{\mathcal{A}_x}$ one can use inverse transform sampling, see for example Knuth (1998):

THEOREM 10.1.6 (Inverse transform sampling). *Let F be a continuous cumulative distribution function with inverse F^{-1} for $u \in [0, 1]$ defined by*

$$F^{-1}(u) := \inf \{x \in \mathbb{R} \mid F(x) = u\}.$$

If U is uniformly distributed on $[0, 1]$, then $F^{-1}(U)$ follows the distribution F' .

PROOF. We have $\text{prob}(F^{-1}(U) \leq x) = \text{prob}(U \leq F(x)) = F(x)$. □

The expected number of primality tests now is in $\mathcal{O}(\ln x)$.

PROOF. If \mathcal{A} is $[c_1, 1]$ -balanced then $F_{\mathcal{A}_x}(y) = 0$ as long as $y \leq x^{c_1}$. The exit probability of the first loop is $\text{prob}(y \text{ prime})$ where y is chosen according to the distribution $F'_{\mathcal{A}_x}$:

$$\text{prob}(y \text{ prime}) \sim \int_1^x \frac{F'_{\mathcal{A}_x}(y)}{\ln y} dy \in \left[\frac{1}{\ln x}, \frac{1}{c_1 \ln x} \right].$$

Thus we expect $\mathcal{O}(\ln x) \cap \Omega(c_1 \ln x)$ repetitions of the upper loop until y is prime. \square

Of course we have to take into account that for each trial y the inverse $F_{\mathcal{A}_x}^{-1}(y)$ has to be computed — at least approximately —, yet this cost is usually negligible compared to a primality test.

10.1.3. Other constructions. There are variants around, where we select the prime numbers differently: Take an integer randomly from a suitable interval and increase the result until the first prime is found. This has the advantage that the amount of randomness needed is considerably lower and by optimizing the resulting algorithm can also be made much faster. The price one has to pay is that the produced primes will not be selected uniformly at random: Primes p for which $p - 2$ is also prime will be selected with a much lower probability than randomly selected primes of a given length. As shown in Brandt & Damgård (1993) the output entropy of such algorithms is still almost maximal and also generators based on these kind of prime-generators might be used in practice.

10.1.4. Summary. We have seen that Algorithm 10.1.2 and 10.1.5 are practical uniform generators for any symmetric or antisymmetric notion.

Note that Algorithm 10.1.2 and 10.1.5 may, however, still produce numbers in a non-uniform fashion: In the last step of both algorithms a product is computed that corresponds to either one pair or two pairs in \mathcal{A}_x . To solve this problem we have two choices: Either we replace \mathcal{A} by its symmetric version \mathcal{S} which we define as $\mathcal{S}_x := \{(y, z) \in \mathbb{R}_{>1}^2 \mid (y, z) \in \mathcal{A}_x \vee (z, y) \in \mathcal{A}_x\}$, or by its, say, top half \mathcal{T} given by $\mathcal{T}_x := \{(y, z) \in \mathcal{S}_x \mid z \geq y\}$ before anything else.

It is now relatively simple to instantiate the above algorithms using the notions proposed in Section 8.4: Namely for an algorithm following the Las Vegas approach, one simply needs to find a suitable banner that encloses the desired notion. In order to instantiate Algorithm 10.1.5 we need to determine the inverse of the corresponding cumulative distribution function for the respective notion (see Table 10.1.1 and 10.1.2). Still Algorithm 10.1.2 and 10.1.5 are practically uniform generators for any symmetric or antisymmetric notion.

Considering runtimes we observe that Algorithm 10.1.5 is much faster, but we have to use inverse transform sampling to generate the first prime. However, despite the simplicity of the approaches some of the most common implementations use corrupted versions of Algorithm 10.1.2 or 10.1.5 as explained below.

Notion \mathcal{A}	$F_{\mathcal{A}_x}$
$\mathcal{A}^{\text{DM}(r)}$	$\begin{cases} 0 & \text{if } y \leq \frac{\sqrt{x}}{r}, \\ \frac{r^2 y^2 + x(\ln x - 2 \ln r - 2 \ln y - 1)}{x(r-1) \ln r} & \text{if } \frac{\sqrt{x}}{r} < y \leq \sqrt{\frac{x}{r}}, \\ \frac{r(x-y^2) + x(r(1+\ln r + 2 \ln y - \ln x) - \ln r)}{x(r-1) \ln r} & \text{if } \sqrt{\frac{x}{r}} < y \leq \sqrt{x}, \\ 1 & \text{if } \sqrt{x} < y. \end{cases}$
$\mathcal{A}^{\text{FB}(r,\sigma)}$	$\begin{cases} 0 & \text{if } y \leq \sqrt{\frac{x}{r}}, \\ \frac{\sqrt{r} \left(r^{\frac{1+\sigma}{2}} - 1 \right)}{\sqrt{x} \left(\sigma r \ln r + r - 2r^{\frac{1+\sigma}{2}} + 1 \right)} & \text{if } \sqrt{\frac{x}{r}} < y \leq \sqrt{\frac{x}{r^\sigma}}, \\ \frac{r\sqrt{x}(\sigma \ln r + 2 \ln y - \ln x + 2) - 2\sqrt{r}y + \sqrt{x} \left(2r^{\frac{1+\sigma}{2}} + 2 \right)}{2\sqrt{x} \left(\sigma r \ln r + r - 2r^{\frac{1+\sigma}{2}} + 1 \right)} & \text{if } \sqrt{\frac{x}{r^\sigma}} < y \leq \sqrt{r^\sigma x}, \\ 1 & \text{if } \sqrt{r^\sigma x} < y. \end{cases}$
$\mathcal{A}^{\text{ALG}(r)}$	$\begin{cases} 0 & \text{if } y \leq \frac{\sqrt{x}}{r}, \\ \frac{2 \ln r + 2 \ln y - \ln x}{2 \ln r} & \text{if } \frac{\sqrt{x}}{r} < y \leq \sqrt{x}, \\ 1 & \text{if } \sqrt{x} < y. \end{cases}$

Table 10.1.1: Some cumulative density functions.

Notion \mathcal{A}	$F'_{\mathcal{A}_x}$
$\mathcal{A}^{\text{DM}(r)}$	$\begin{cases} \frac{2(r^2 y^2 - x)}{xy(r-1) \ln r} & \text{if } \frac{\sqrt{x}}{r} < y \leq \sqrt{\frac{x}{r}}, \\ \frac{2r(x-y^2)}{xy(r-1) \ln r} & \text{if } \sqrt{\frac{x}{r}} < y \leq \sqrt{x}, \\ 0 & \text{otherwise.} \end{cases}$
$\mathcal{A}^{\text{FB}(r,\sigma)}$	$\begin{cases} \frac{\sqrt{r} \left(r^{\frac{1+\sigma}{2}} - 1 \right)}{\sqrt{x} \left(\sigma r \ln r + r - 2r^{\frac{1+\sigma}{2}} + 1 \right)} & \text{if } \sqrt{\frac{x}{r}} < y \leq \sqrt{\frac{x}{r^\sigma}}, \\ \frac{r\sqrt{x} - \sqrt{r}y}{\sqrt{x}y \left(\sigma r \ln r + r - 2r^{\frac{1+\sigma}{2}} + 1 \right)} & \text{if } \sqrt{\frac{x}{r^\sigma}} < y \leq \sqrt{r^\sigma x}, \\ 0 & \text{otherwise.} \end{cases}$
$\mathcal{A}^{\text{ALG}(r)}$	$\begin{cases} \frac{1}{y \ln r} & \text{if } \frac{\sqrt{x}}{r} < y \leq \sqrt{x}, \\ 0 & \text{otherwise.} \end{cases}$

Table 10.1.2: Derivatives of the density functions in Table 10.1.1 with respect to y .

10.2. Output entropy

The entropy of the output distribution is one important quality measure of all kinds of generators: For example the quality of a physical random generator is typically measured by the amount of entropy it produces, see Schindler (2008a) and Schindler (2008b). There are specific constructions

for such generators for which good stochastic models exist, see for example Killmann & Schindler (2008). For arbitrary construction the development of reasonable models remains a challenging task. Also for primality tests several analyses were performed, see for example Brandt & Damgård (1993) or Joye & Paillier (2006). For generators of RSA integers we are not aware of any work in this direction.

Let \mathcal{A} be any monotone notion. Consider a generator G that produces a pair of primes $(p, q) \in \mathcal{A}_x$ with distribution G_{out} . Seen as random variables, G_{out} induces two random variables P and Q by its first and the second coordinate, respectively. The entropy of the generator G is given by

$$H(G_{\text{out}}) = H(P \times Q) = H(P) + H(Q | P),$$

where H denotes the entropy and the conditional entropy is given by

$$(10.2.1) \quad H(Q | P) = - \sum_{p \in \text{im}(P)} \text{prob}(P = p) \cdot \sum_{q \in \text{im}(Q|P)} \text{prob}(Q = q | P = p) \log_2(\text{prob}(Q = q | P = p)).$$

If G_{out} is the uniform distribution U we obtain by Lemma 8.3.5 maximal entropy

$$(10.2.2) \quad H(U) = \log_2(\#\mathcal{A}(x)) \approx \log_2(\text{area}(\mathcal{A}_x)) - \log_2(\ln x) + 1,$$

with an error of very small order. The algorithms from Section 10.1, however, return the product $P \cdot Q$. The entropy of this random variable can be estimated as

$$\begin{aligned} H(P \cdot Q) &= - \sum_{\substack{n=pq \in \mathbb{N} \\ (p,q) \in \mathcal{A}_x}} \text{prob}(P \cdot Q = n) \log_2(\text{prob}(P \cdot Q = n)) \\ &\geq - \sum_{(p,q) \in \mathcal{A}_x} \text{prob}(P \times Q = (p, q)) \log_2(2 \text{prob}(P \times Q = (p, q))) \\ &= H(P \times Q) - 1. \end{aligned}$$

Some of the standards and implementations in Section 10.4 (like the standard IEEE 1363-2000 or the implementation of GNU Crypto) *do not* generate every possible outcome with the same probability. All of them have in common that the prime p is selected uniformly at random and afterwards the prime q is selected uniformly at random from an appropriate interval.

This is a non-uniform selection process since for some choices of p there might be less choices for q .

If the probability distribution G_{out} is close to the uniform distribution, say $G_{\text{out}}(p, q) \in [2^{-\varepsilon}, 2^\varepsilon] \frac{1}{\#\mathcal{A}(x)}$ for some fixed $\varepsilon \in \mathbb{R}_{>0}$, then the entropy of the resulting generator can be estimated as follows:

$$\begin{aligned} H(G_{\text{out}}) &= - \sum_{(p,q) \in \mathcal{A}_x} G_{\text{out}}(p, q) \log_2(G_{\text{out}}(p, q)) \\ &\in \sum_{(p,q) \in \mathcal{A}_x} G_{\text{out}}(p, q) [\log_2(\#\mathcal{A}(x)) - \varepsilon, \log_2(\#\mathcal{A}(x)) + \varepsilon] \\ &= [H(U) - \varepsilon, H(U) + \varepsilon] \end{aligned}$$

and since the entropy of the uniform distribution is maximal, this implies that

$$H(G_{\text{out}}) \geq H(U) - \varepsilon.$$

A measure for the quality of a generator G is the *entropy loss* with respect to an optimal one. It is defined as

$$\delta(G) = \frac{H(U) - H(G_{\text{out}})}{H(U)}.$$

In the next section we will explore yet another measure of quality, which compares the output entropy of a generator to the needed entropy for generating the results.

We will now estimate the output entropy for the three generators from Section 10.1. Since Algorithm 10.1.2 and Algorithm 10.1.5 first produce pairs of primes $(p, q) \in \mathcal{A}_x$ uniformly at random and return the product pq , their output entropy is by (10.2.2) at least

$$\log_2(\#\mathcal{A}(x)) - 1 \approx \log_2(\text{area}(\mathcal{A}_x)) - \log_2(\ln x).$$

For Algorithm 10.1.3 we use (10.2.1) to obtain an estimate. Since the prime p is selected uniformly at random, we have $H(P) = \frac{1}{\#\text{im}(P)}$. For the conditional entropy we obtain

$$\begin{aligned} H(Q|P) &= \sum_{p \in \text{im}(P)} \frac{1}{\#\text{im}(P)} \sum_{q \in \text{im}(Q|P)} \frac{1}{\#\text{im}(Q|P)} \log_2 \#\text{im}(Q|P) \\ &= \frac{1}{\#\text{im}(P)} \cdot \sum_{p \in \text{im}(P)} \log_2 \#\text{im}(Q|P). \end{aligned}$$

In case the notion \mathcal{A} is graph-bounded with boundary functions $B_1(x)$, $C_2(y, x)$, $B_2(y, x)$ and $C_1(y, x)$ we can for large x estimate the conditional entropy by

$$H(Q|P) \approx \frac{1}{f(C_1(x)) - f(B_1(x))} \int_{B_1(x)}^{C_1(x)} \frac{\log_2(f(C_2(y, x)) - f(B_2(y, x)))}{\ln y} dy,$$

where we abbreviated $f(x) = \frac{x}{\ln x}$.

10.3. Information-theoretical efficiency

Consider a generator G that produces outputs with distribution G_{out} . To compute the output, the generator has access to several oracles $\mathcal{O}_1, \dots, \mathcal{O}_t$ which might perform random choices. All other operations in G should be deterministic. Denote by G_{in} the random variable carrying the results of the calls to the different oracles during a run of G . In abuse of language we will frequently call the entropy of G_{in} the *input entropy* of G .

With these definitions, the (information-theoretic) *efficiency* of G is given by

$$\eta(G) = \frac{H(G_{\text{out}})}{H(G_{\text{in}})}.$$

Obviously, for all generators the efficiency has values in the unit interval, i.e. $\eta(G) \in [0, 1]$.

Assume we have any generator G that produces elements from some set A by sampling uniformly from a set $B \supseteq A$ until the result lies in A . Such a generator has output entropy $H(G_{\text{out}}) = \log_2 \#A$ since the output of G will be uniformly selected from A . Let us consider the index entropy: Per sample G uses $\log_2 \#B$ bits of entropy. Since the expected number of samples is $\frac{\#B}{\#A}$, we obtain input entropy $\frac{\#B}{\#A} \cdot \log_2 \#B$ bits and thus the efficiency of G is

$$(10.3.1) \quad \eta(G) = \frac{\#A \log_2 \#A}{\#B \log_2 \#B}.$$

We can directly apply this result to Algorithm 10.1.2 for any $[c_1, c_2]$ -balanced notion \mathcal{A} of RSA integers with tolerance r : There, we sample uniformly from the banner \mathcal{B}_x until we end up with a pair of primes in $\mathcal{A}_x \subseteq \mathcal{B}_x$. By Lemma 8.3.5 the number of prime pairs in \mathcal{A}_x is for large x lower bounded by $\frac{\text{area}(\mathcal{A}_x)}{c_2^2 \ln^2 x}$. Thus, we obtain by (10.3.1) that Algorithm 10.1.2 has for large x an efficiency of at least

$$\frac{\text{area}(\mathcal{A}_x)(\log_2 \text{area}(\mathcal{A}_x) - \log_2 \ln^2 x - 2 \log_2 c_2 + 2)}{c_2^2 \ln^2 x \text{area}(\mathcal{B}_x) \log_2 \text{area}(\mathcal{B}_x)} \geq \frac{2 \text{area}(\mathcal{A}_x) \log_2 \text{area}(\mathcal{A}_x)}{\text{area}(\mathcal{B}_x) \log_2 \text{area}(\mathcal{B}_x)} \cdot \frac{1}{\ln^2 x}.$$

Consider now Algorithm 10.1.3. There we select a prime p first via rejection sampling and afterwards q such that the result is in the desired interval. Even though it is difficult to explicitly estimate in general the input and the output entropy of this generator, we can easily evaluate its efficiency. This is possible, since for large x the efficiency of both prime number generators used in the algorithm is $\frac{1}{\ln x}$, implying that the algorithm itself has for large x efficiency

$$\frac{1}{\ln^2 x}.$$

Also Algorithm 10.1.5 can be treated similarly: The only difference to Algorithm 10.1.3 is that the selection of the first prime is not done uniformly at random but following some distribution $F_{\mathcal{A}_x}$, see Section 10.1. We will not go into more details on these kinds of algorithms, since we will not need them for the analysis of the standards and implementations in the next section.

10.4. Impact on standards and implementations

In order to get an understanding of the common implementations, it is necessary to consult the main standard on RSA integers, namely the standard PKCS#1 (Jonsson & Kaliski 2003). However, one cannot find *any* requirements on the shape of RSA integers. Interestingly, they even allow more than two factors for an RSA modulus. Also the standard ISO 18033-2 (International Organization for Standards 2006) does not give any details besides the fact that it requires the RSA integer to be a product of two different primes of similar length. A more precise standard is set by the German Bundesnetzagentur (Wohlmacher 2009). They do not give a specific algorithm, but at least require that the prime factors are not too small and not too close to each other. We will now analyze several standards which give a concrete algorithm for generating an RSA integer. In particular, we consider the standard of the RSA foundation (RSA Laboratories 2000), the IEEE standard 1363 (IEEE working group 2000), the NIST standard FIPS 186-3 (NIST 2009) and the standard ANSI X9.44 (Accredited Standards Committee X9 2007) and the standard resulting from the European NESSIE project (NESSIE working group 2003).

10.4.1. RSA-OAEP. The RSA Laboratories (2000) describe the following variant:

ALGORITHM 10.4.1. Generating an RSA integer for RSA-OAEP and variants.

Input: A number of bits k , the public exponent e .

Output: An integer $n = pq$.

1. Pick p from $\left[\left\lfloor 2^{(k-1)/2} \right\rfloor + 1, \left\lfloor 2^{k/2} \right\rfloor - 1 \right] \cap \mathbb{P}$ such that $\gcd(e, p-1) = 1$.
2. Pick q from $\left[\left\lfloor 2^{(k-1)/2} \right\rfloor + 1, \left\lfloor 2^{k/2} \right\rfloor - 1 \right] \cap \mathbb{P}$ such that $\gcd(e, q-1) = 1$.
3. Return pq .

This will produce uniformly at random an integer from the interval $[2^{k-1} + 1, 2^k - 1]$ and no cutting off. The output entropy is thus maximal. So this corresponds to the notion $\mathcal{A}^{\text{FB}(2,0)}$ generated by Algorithm 10.1.5. The standard requires an expected number of $k \ln 2$ primality tests if the gcd condition is checked first. Otherwise the expected number of primality tests increases to $\frac{\varphi(e)}{\varphi_1(e)} \cdot k \ln 2$ (see (8.3.11)). We will in the following always mean by the above notation that the second condition is checked first and afterwards the number is tested for primality. For the security Theorem 8.6.3 applies.

10.4.2. IEEE. IEEE standard 1363-2000, Annex A.16.11 (IEEE working group 2000) introduces our algorithmic proposal:

ALGORITHM 10.4.2. Generating an RSA integer, IEEE 1363-2000.

Input: A number of bits k , the odd public exponent e .

Output: An integer $n = pq$.

1. Pick p from $\left[2^{\lfloor \frac{k-1}{2} \rfloor}, 2^{\lfloor \frac{k+1}{2} \rfloor} - 1 \right] \cap \mathbb{P}$ such that $\gcd(e, p-1) = 1$.
2. Pick q from $\left[\left\lfloor \frac{2^{k-1}}{p} + 1 \right\rfloor, \left\lfloor \frac{2^k}{p} \right\rfloor \right] \cap \mathbb{P}$ such that $\gcd(e, q-1) = 1$.
3. Return pq .

Since the resulting integers are in the interval $[2^{k-1}, 2^k - 1]$ this standard follows $\mathcal{A}^{\text{ALG}_2(2,1/2)}$ generated by a corrupted variant of Algorithm 10.1.5 using an expected number of $k \ln 2$ primality tests like the RSA-OAEP standard. The notion it implements is neither symmetric nor antisymmetric. The selection of the integers is *not* done in a uniform way, since the number of possible q for the largest possible p is roughly half of the corresponding number for the smallest possible p . Since the distribution of the outputs is close to uniform, we can use the techniques from Section 10.2 to estimate the output entropy to find that the entropy-loss is less than 0.69 bit. The (numerically approximated) values in Table 10.4.1 gave an actual entropy-loss of approximately 0.03 bit.

10.4.3. NIST. We will now analyze the standard FIPS 186-3 (NIST 2009). In Appendix B.3.1 of the standard one finds the following algorithm:

ALGORITHM 10.4.3. Generating an RSA integer, FIPS186-3.

Input: A number of bits k , a number of bits $\ell < k$, the odd public exponent $2^{16} < e < 2^{256}$.

Output: An integer $n = pq$.

1. Pick p from $[\sqrt{2}2^{k/2-1}, 2^{k/2} - 1] \cap \mathbb{P}$ such that $\gcd(e, p - 1) = 1$ and $p \pm 1$ has a prime factor with at least ℓ bits.
2. Pick q from $[\sqrt{2}2^{k/2-1}, 2^{k/2} - 1] \cap \mathbb{P}$ such that $\gcd(e, p - 1) = 1$ and $q \pm 1$ has a prime factor with at least ℓ bits and $|p - q| > 2^{k/2-100}$.
3. Return pq .

In the standard it is required that the primes p and q shall be either provable prime or at least probable primes. The (at least ℓ -bit) prime factors of $p \pm 1$ and $q \pm 1$ have to be provable primes. We observe that also in this standard a variant of the notion $\mathcal{A}^{\text{FB}(2,0)}$ generated by Algorithm 10.1.5 is used. The output entropy is thus maximal. However, we do not have any restriction on the parity of k , such that the value $k/2$ is not necessarily an integer. Another interesting point is the restriction on the prime factors of $p \pm 1, q \pm 1$. Our notions cannot directly handle such requirements, but we are confident that this can be achieved by appropriately modifying the densities in Lemma 8.3.5.

The standard requires an expected number of slightly more than $k \ln 2$ primality tests. It is thus slightly less efficient than the RSA-OAEP standard. For the security the remarks from the end of Section 8.6 apply.

10.4.4. ANSI. The ANSI X9.44 standard (Accredited Standards Committee X9 2007), formerly part of ANSI X9.31, requires strong primes for an RSA modulus. Unfortunately, we could not access ANSI X9.44 directly and are therefore referring to ANSI X9.31-1998. Section 4.1.2 of the standard requires that

- $p - 1, p + 1, q - 1, q + 1$ each should have prime factors p_1, p_2, q_1, q_2 that are randomly selected primes in the range 2^{100} to 2^{120} ,
- p and q shall be the first primes that meet the above, found in an appropriate interval, starting from a random point,
- p and q shall be different in at least one of their first 100 bits.

The additional restrictions are similar to the ones required by NIST. This procedure will have an output entropy that is close to maximal (see Section 10.2).

10.4.5. NESSIE. The European NESSIE project gives in its security report (NESSIE working group 2003) a very similar algorithm:

ALGORITHM 10.4.4. Generating an RSA integer, NESSIE standard.

Input: A number of bits k , the odd public exponent e .

Output: An integer $n = pq$.

1. Pick p from $\left[2^{k-1}, 2^k - 1\right] \cap \mathbb{P}$ such that $\gcd(e, p - 1) = 1$.
2. Pick q from $\left[2^{k-1}, 2^k - 1\right] \cap \mathbb{P}$ such that $\gcd(e, q - 1) = 1$.
3. Return pq .

The resulting integer n is selected uniformly at random from the interval $[2^{2k-2}, 2^{2k} - 1]$ and thus corresponds to the fixed bound notion $\mathcal{A}^{\text{FB}(4,0)}$ generated by Algorithm 10.1.5. The output entropy is thus maximal. Note the difference to the standard of the RSA foundation: Besides the fact, that in the standard of the RSA laboratories some sort of rounding is done, the security parameter k is treated differently: While for the RSA foundation the security parameter describes the (rough) length of the output, in the NESSIE proposal it denotes the size of the two prime factors. The standards requires an expected number of $2k \ln 2$ primality tests. It is thus as efficient as the RSA-OAEP standard. For the security Theorem 8.6.3 applies.

10.4.6. OpenSSL. We now turn to implementations: For `OpenSSL` (Cox *et al.* 2009), we refer to the file `rsa_gen.c`. Note that in the configuration the routine used for RSA integer generation can be changed, while the algorithm given below is the standard one. `OpenSSH` (de Raadt *et al.* 2009) uses the same library. Refer to the file `rsa.c`. We have the following algorithm:

ALGORITHM 10.4.5. Generating an RSA integer in `OpenSSL`.

Input: A number of bits k .

Output: An integer $n = pq$.

1. Pick p from $\left[2^{\lfloor \frac{k-1}{2} \rfloor}, 2^{\lfloor \frac{k+1}{2} \rfloor} - 1\right] \cap \mathbb{P}$.
2. Pick q from $\left[2^{\lfloor \frac{k-3}{2} \rfloor}, 2^{\lfloor \frac{k-1}{2} \rfloor} - 1\right] \cap \mathbb{P}$.
3. Return pq .

This is nothing but a rejection-sampling method of a notion similar to the fixed bound notion $\mathcal{A}^{\text{FB}(4,0)}$ generated by Algorithm 10.1.2. The output entropy is thus maximal. The result the algorithm produces is always in $[2^{k-2}, 2^k - 1]$. It is clear that this notion is antisymmetric and the factors are on average a factor 2 apart of each other. The implementation runs in an expected number of $k \ln 2$ primality tests. The public exponent e is afterwards selected such that $\gcd((p-1)(q-1), e) = 1$. It is thus slightly more efficient than the RSA-OAEP standard. For the security Theorem 8.6.3 applies.

10.4.7. Openswan. In the open source implementation `Openswan` of the IPsec protocol (Richardson *et al.* 2009) one finds a rejection-sampling method that is actually implementing the notion $\mathcal{A}^{\text{FB}(4,0)}$ generated by a variant of Algorithm 10.1.2. We refer to the function `rsasigkey` in the file `rsasigkey.c`:

ALGORITHM 10.4.6. Generating an RSA integer in `Openswan`.

Input: A number of bits k .

Output: An integer $n = pq$.

1. Pick p from $[2^{\lfloor \frac{k-2}{2} \rfloor}, 2^{\lfloor \frac{k}{2} \rfloor} - 1] \cap \mathbb{P}$.
2. Pick q from $[2^{\lfloor \frac{k-2}{2} \rfloor}, 2^{\lfloor \frac{k}{2} \rfloor} - 1] \cap \mathbb{P}$.
3. Return pq .

Note that here the notion *is* actually symmetric. However, still the uniformly at random selected integer pq will not always have the same length. The implementation runs in an expected number of $k \ln 2$ primality tests and output entropy is maximal. Again the public exponent e is selected afterwards such that $\gcd((p-1)(q-1), e) = 1$. It is thus as efficient as the RSA-OAEP standard. For the security Theorem 8.6.3 applies.

10.4.8. GnuPG. Also `GnuPG` (Skala *et al.* 2009) uses rejection-sampling of the fixed bound notion $\mathcal{A}^{\text{FB}(2,1)}$ generated by a variant of Algorithm 10.1.2, implying that the entropy of its output distribution is maximal.

ALGORITHM 10.4.7. Generating an RSA integer in `GnuPG`.

Input: A number of bits k .

Output: An integer $n = pq$.

1. Repeat 2–3

2. Pick p from $\left[2^{\lfloor \frac{k-1}{2} \rfloor}, 2^{\lfloor \frac{k+1}{2} \rfloor} - 1\right] \cap \mathbb{P}$.
3. Pick q from $\left[2^{\lfloor \frac{k-1}{2} \rfloor}, 2^{\lfloor \frac{k+1}{2} \rfloor} - 1\right] \cap \mathbb{P}$.
4. Until $\text{len}(pq) = 2 \lceil k/2 \rceil$
5. Return pq .

We refer here to the file `rsa.c`. The algorithm is given in the function `generate_std` and produces always integers with either k or $k + 1$ bits depending on the parity of k . Note that the generation procedure indeed first selects primes before checking the validity of the range. This is of course a waste of resources, see Section 10.1.

The implementation runs in an expected number of roughly $2.589 \cdot (k + 1) \ln 2$ primality tests. It is thus less efficient than the RSA OAEP standards. Like in the other, so far considered implementations, the public exponent e is afterwards selected such that $\text{gcd}((p - 1)(q - 1), e) = 1$. For the security Theorem 8.6.3 applies.

10.4.9. GNU Crypto. The GNU Crypto library (Free Software Foundation 2009) generates RSA integers the following way. Refer here in the file `RSAPairGenerator.java` to the function `generate`.

ALGORITHM 10.4.8. Generating an RSA integer in GNU Crypto.

Input: A number of bits k .

Output: An integer $n = pq$.

1. Pick p from $\left[2^{\lfloor \frac{k-1}{2} \rfloor}, 2^{\lfloor \frac{k+1}{2} \rfloor} - 1\right] \cap \mathbb{P}$.
2. Repeat
3. Pick q from $\left[2^{\lfloor \frac{k-1}{2} \rfloor}, 2^{\lfloor \frac{k+1}{2} \rfloor} - 1\right]$.
4. Until $\text{len}(pq) = k$ and $q \in \mathbb{P}$.
5. Return pq .

Also here the notion $\mathcal{A}^{\text{FB}(2,1)}$ is used, but the generated integers will not be uniformly distributed, since for a larger p we have much less choices for q . Since the distribution of the outputs is not close to uniform, we could only compute the entropy for real-world parameter choices numerically (see Table 10.4.1). For all choices the loss was less than 0.63 bit. The implementation is as efficient as the RSA-OAEP standard.

The Free Software Foundation provides `Gnu Classpath`, which generates RSA integers exactly like the GNU Crypto library, i.e. following $\mathcal{A}^{\text{FB}(2,1)}$. We refer to the file `RSAPairGenerator.java`. As in the other, so far

considered implementations, the public exponent e is afterwards randomly selected such that $\gcd((p-1)(q-1), e) = 1$. Like in the IEEE 1363-2000 and the ANSI X9.44 standard this does not impose practical security risks, but it does not meet the requirement of uniform selection of the generated integers.

10.4.10. Summary. It is striking to observe that not *a single* analyzed implementation follows one of the standards described above. The only standards all implementations are compliant to are the standards PKCS#1 and ISO 18033-2, which themselves does not specify anything related to the integer generation routine. We found that also the requirements from the algorithm catalog of the German Bundesnetzagentur (Wohlmacher 2009) are not met in a single considered implementation, since it is never checked whether the selected primes are too close to each other. The implementation that almost meets the requirements is the implementation of OpenSSL. Interestingly there are standards and implementations around that generate integers non-uniformly. Prominent examples are the IEEE and the ANSI standards and the implementation of the Gnu Crypto library. This does not impose practical security issues, but it violates the condition of uniform selection.








Standard	Entropy (entropy loss/norm. efficiency)			Notion
	$k = 768$	$k = 1024$	$k = 2048$	
Implementation				
uniform $n = pq$	762.59	1018.24	2041.39	—
PKCS#1	Undefined	—	—	—
ISO 18033-2	Undefined	—	—	—
ANSI X9.44	Undefined	—	—	—
FIPS 186-3	$\lesssim 747.34$ ($\gtrsim 0\%_0/1$)	$\lesssim 1002.51$ ($\gtrsim 0\%_0/1$)	$\lesssim 2024.51$ ($\gtrsim 0\%_0/1$)	
RSA-OAEP	747.34 (0% ₀ /1)	1002.51 (0% ₀ /1)	2024.51 (0% ₀ /1)	
IEEE 1363-2000	749.33 (0.04% ₀ /1)	1004.50 (0.03% ₀ /1)	2026.50 (0.01% ₀ /1)	
NESSIE	749.89 (0% ₀ /1)	1005.06 (0% ₀ /1)	2027.06 (0% ₀ /1)	
GNU Crypto	747.89 (0.84% ₀ /1)	1003.06 (0.62% ₀ /1)	2025.06 (0.31% ₀ /1)	
GnuPG	748.52 (0% ₀ /0.771)	1003.69 (0% ₀ /0.772)	2025.69 (0% ₀ /0.772)	
OpenSSL/OpenSwan	749.89 (0% ₀ /1)	1005.06 (0% ₀ /1)	2027.06 (0% ₀ /1)	

Table 10.4.1: Overview of various standards and implementations. As explained in the text, the entropy of the standards is slightly smaller than the values given due to the fixed public exponent e . Additionally there is a small entropy loss for the standard FIPS 186-3 due to the fact that it requires strong primes. In the table we have multiplied the efficiency of the implementations by the factor $\ln^2 x$ and the efficiency of the standards by the factor $\frac{\varphi_1(e)}{\varphi(e)} \cdot \ln^2 x$ (due to the preselected public exponent e) to obtain a normalized efficiency.

Chapter 11

Future work and open problems

There are still many open problems on various aspects of the topics presented in the thesis. Further research could focus either on the more number theoretic work or the applied implementation issues. We describe here a few ideas.

First of all one should be able to tighten the number theoretic part in Chapter 6: It would be nice to have asymptotic formulas for the count of $[B, C]$ -grained integers that specify the constant in the main term explicitly. The same improvement should be possible for the count of RSA integers in Chapter 8 and generalized RSA integers in Chapter 9. More specifically the technique that could lead to deeper insight in these issues is to think again about the evaluation of integrals of the form

$$\iint \frac{1}{\ln p \ln q} dq dp.$$

Though there is no hope for an elementary solution to such kind of integrals (since it would readily lead to elementary expressions defining the logarithmic integral) it should be possible to tackle these integrals in such a way that the desired constant pops out.

Also, additional requirements on the prime factors (like being strong primes) need further investigation. To advance in this direction one would either have to know some explicit counting results of such kind of prime numbers (which seem out of reach at the moment) or somehow circumvent the necessity of such counts.

Another possible field of research is to obtain even deeper insight in the intermediate steps of the General Number Field Sieve:

One could try to use the observation that the numbers entering the cofactorization step that correspond to promising candidates are exactly those numbers that are $[B, C]$ -grained and simultaneously of a certain length. This in turn could be used to specify the runtime bound of the Elliptic Curve Method 3.5.9 such that there is as little waste of runtime as possible. Also, if one would manage to explicitly describe the *input distribution* of cofactorization step in the General Number Field Sieve (studied experimentally in Chapter 7) one might heuristically obtain better runtime bound of the General Number Field Sieve for a given input.

Related to this issue are further improvements in the search of efficient formulas for differential addition on special elliptic curves. This would also lead to faster implementations of the General Number Field Sieve.

A completely different topic for further work is the study of the different standards on various topics in cryptography. It should be possible to define appropriate “notions” for many cryptographic objects that might lead to very general insight about the properties of these objects. Yet, it is not obvious at all which kind of results such an approach could yield.

Bibliography

The numbers in brackets at the end of a reference are the pages on which it is cited. Names of authors and titles are usually given in the same form as on the article or book.

ACCREDITED STANDARDS COMMITTEE X9 (2007). ANSI X9.44-2007: Public Key Cryptography Using Reversible Algorithms for the Financial Services Industry: Transport of Symmetric Algorithm Keys Using RSA. Technical report, American National Standards Institute, American Bankers Association. [142, 186–188]

LEONARD MAX ADLEMAN (1979). A Subexponential Algorithm for the Discrete Logarithm Problem with Applications. In *Proceedings of the 20th Annual IEEE Symposium on Foundations of Computer Science*, San Juan PR, 55–60. [56]

MANINDRA AGRAWAL, NEERAJ KAYAL & NITIN SAXENA (2004). PRIMES is in P. *Annals of Mathematics* **160**(2), 781–793. URL <http://annals.math.princeton.edu/issues/2004/Sept2004/Agrawal.pdf>. [49–50]

WILLIAM ROBERT ALFORD, ANDREW GRANVILLE & CARL POMERANCE (1994a). On the difficulty of finding reliable witnesses. In *Algorithmic Number Theory, First International Symposium, ANTS-I*, Ithaca, NY, USA, LEONARD MAX ADLEMAN & MING-DEH HUANG, editors, number 877 in Lecture Notes in Computer Science, 1–16. Springer-Verlag, Berlin, Heidelberg. ISBN 978-3-540-58691-3. ISSN 0302-9743. URL <http://dx.doi.org/10.1007/3-540-58691-1>. [42]

WILLIAM ROBERT ALFORD, ANDREW GRANVILLE & CARL POMERANCE (1994b). There are infinitely many Carmichael numbers. *Annals of Mathematics* **140**, 703–722. [42]

ROBERTO M. AVANZI, HENRI COHEN, CHRISTOPHE DOCHE, GERHARD FREY, TANJA LANGE, KIM NGUYEN & FREDERIK VERCAUTEREN (2006). *Handbook of Elliptic and Hyperelliptic Curve Cryptography*. Discrete Mathematics and its Applications. Chapman & Hall/CRC. ISBN 1-58488-518-1. [71]

ERIC BACH & JEFFREY SHALLIT (1996). *Algorithmic Number Theory, Vol.1: Efficient Algorithms*. MIT Press, Cambridge MA. [49]

CLAUDE GASPAR BACHET DE MÉZIRIAC (1612). *Problèmes plaisans et délectables, qui se font par les nombres*. Pierre Rigaud, Lyon.

RICHARD BELLMAN (1957). *Dynamic Programming*. Princeton University Text. [132]

JACOB BERNOULLI (1713). *Ars conjectandi, opus posthumum, Accedit Tractatus de seriebus infinitis, et epistola gallicé scripta de ludo pilae reticularis*. Basileae, impensis Thurnisiorum, Basel.

DANIEL JULIUS BERNSTEIN, PETER BIRKNER, MARC JOYE, TANJA LANGE & CHRISTIANE PETERS (2008a). Twisted Edwards Curves. In *Progress in Cryptology: Proceedings of AFRICACRYPT 2008*, Casablanca, Morocco, SERGE VAUDENAY, editor, volume 5023 of *Lecture Notes in Computer Science*, 389–405. URL http://dx.doi.org/10.1007/978-3-540-68164-9_26. [69]

DANIEL JULIUS BERNSTEIN, PETER BIRKNER, TANJA LANGE & CHRISTIANE PETERS (2008b). ECM using Edwards curves URL <http://cr.ypt.to/factorization/eecm-20080120.pdf>. [69]

DANIEL JULIUS BERNSTEIN & TANJA LANGE (2007a). Faster addition and doubling on elliptic curves. In *Advances in Cryptology: Proceedings of ASIACRYPT 2007*, Kuching, Sarawak, Malaysia, KAORU KUROSAWA, editor, volume 4833 of *Lecture Notes in Computer Science*, 29–50. URL <http://dx.doi.org/10.1007/978-3-540-76900-2>. [69, 72]

DANIEL JULIUS BERNSTEIN & TANJA LANGE (2007b). Inverted Edwards Coordinates. In *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes, 17th International Symposium, AA ECC-17, Bangalore, India, December 16-20, 2007, Proceedings*, SERDAR BOZTAS & HSIAO-FENG LU, editors, volume 4851 of *Lecture Notes in Computer Science*, 20–27. URL <http://dx.doi.org/10.1007/978-3-540-77224-8>. [69, 72]

DANIEL JULIUS BERNSTEIN & TANJA LANGE (2011). Explicit Formulas Database (EFD). URL <http://www.hyperelliptic.org/EFD/>. [64, 70]

ÉTIENNE BÉZOUT (1766). *Cours de Mathématiques à l'usage des Gardes du Pavillon et de la Marine*. J . B . G . Musier, Paris.

JACQUES PHILIPPE MARIE BINET (1730). *Miscellanea Analytica*. J. Tonson and J. Watts, London.

JACQUES PHILIPPE MARIE BINET (1843). Mémoire sur l'intégration des équations linéaires aux différences finies d'un ordre quelconque, à coefficients variables. *Comptes Rendus de l'Académie des Sciences Paris* **17**, 559–567. [29]

JOHANNES BLÖMER & ALEXANDER MAY (2003). New Partial Key Exposure Attacks on RSA. In *Advances in Cryptology - CRYPTO 2003*, DAN BONEH, editor, volume 2729 of *Lecture Notes in Computer Science*, 27–43. Springer-Verlag, Berlin, Heidelberg. ISBN 978-3-540-40674-7. URL <http://dx.doi.org/10.1007/b1181710.1007/b11817>. [172]

DAN BONEH, GLENN DURFEE & NICK HOWGRAVE-GRAHAM (1999). Factoring $N = p^r q$ for Large r . In *Advances in Cryptology: Proceedings of CRYPTO '99*, Santa Barbara, CA, MATTHEW WIENER, editor, number 1666 in Lecture Notes in Computer Science, 326–237. Springer-Verlag, Berlin, Heidelberg. ISBN 3-540-66347-9. ISSN 0302-9743. URL http://dx.doi.org/10.1007/3-540-48405-1_21. [172]

JØRGEN BRANDT & IVAN DAMGÅRD (1993). On Generation of Probable Primes by Incremental Search. In *Advances in Cryptology: Proceedings of CRYPTO '92*, Santa Barbara, CA, E. F. BRICKELL, editor, number 740 in Lecture Notes in Computer Science, 358–370. Springer-Verlag, Berlin. ISSN 0302-9743. URL http://dx.doi.org/10.1007/3-540-48071-4_26. [2, 142, 181–183]

ÉRIC BRIER & MARC JOYE (2002). Weierstraß Elliptic Curves and Side-Channel Attacks. In *Public Key Cryptography*, DAVID NACCACHE & PASCAL PAILLIER, editors, number 2274 in Lecture Notes in Computer Science, 183–194. Springer-Verlag, Berlin, Heidelberg. ISBN 3-540-43168-3. ISSN 0302-9743. URL http://dx.doi.org/10.1007/3-540-45664-3_24. [76]

JOHN BRILLHART, DERRICK HENRY LEHMER & JOHN LEWIS SELFRIDGE (1975). New Primality Criteria and Factorizations of $2^m \pm 1$. *Mathematics of Computation* **29**(130), 620–647. [45]

NICOLAAS GOVERT DE BRUIJN (1951). The asymptotic behaviour of a function occurring in the theory of primes. *Journal of the Indian Mathematical Society n.s.* **15**, 25–32. [25]

JOSEPH P. BUHLER, HENDRIK WILLEM LENSTRA, JR. & CARL POMERANCE (1993). Factoring integers with the number field sieve. In Lenstra & Lenstra (1993), 50–94. [54–55]

Александр Адольфович. Бухштаб (1937). Асимптотическая оценка одной общей теоретикочисловой функции. Математический сборник **2** (44)(6), 1239–1246. URL <http://mi.mathnet.ru/msb5649>. [26, 86]

EARL RODNEY CANFIELD, PAUL ERDŐS & CARL POMERANCE (1983). On a problem of Oppenheim concerning ‘Factorisatio Numerorum’. *Journal of Number Theory* **17**, 1–28. [25, 52]

ROBERT DANIEL CARMICHAEL (1909/10). Note on a new number theory function. *Bulletin of the American Mathematical Society* **16**, 232–238.

JOHN WILLIAM SCOTT CASSELS (1966). Diophantine equations with special reference to elliptic curves. *Journal of the London Mathematical Society* **41**, 193–291. [60]

WOUTER CASTRYCK, STEVEN D. GALBRAITH & REZA REZAEIAN FARASHAHI (2008). Efficient arithmetic on elliptic curves using a mixed Edwards-Montgomery representation. Cryptology ePrint Archive, Report 2008/218. URL <http://eprint.iacr.org/2008/218.pdf>. [3, 70–71, 74]

Пафну́тий Льво́вич Чебы́шёв (1852). Mémoire sur les nombres premiers. *Journal de Mathématiques Pures et Appliquées, I série* **17**, 366–390. *Mémoires présentées à l'Académie Impériale des sciences de St.-Pétersbourg par divers savants* **6** (1854), 17–33. *Œuvres I*, eds. A. MARKOFF and N. SONIN, 1899, reprint by Chelsea Publishing Co., New York, 49–70. [9]

CLAY MATHEMATICS INSTITUTE (2000). WWW. URL http://www.claymath.org/prize_problems/p_vs_np.htm. [5, 82]

CLIFFORD C. COCKS (1973). A note on 'non-secret encryption'. CESG Memo. URL <http://www.cesg.gov.uk/publications/media/notense.pdf>. Last download 12 May 2009. [141]

HENRI COHEN & GERHARD FREY (2006). *Handbook of Elliptic and Hyperelliptic Curve Cryptography*. Discrete Mathematics and its Applications. Chapman & Hall/CRC. ISBN 1-58488-518-1. With the help of Roberto M. Avanzi, Christophe Doche, Tanja Lange, Kim Nguyen, and Frederik Vercauteren. [64]

MARK J. COX, RALF ENGELSCHALL, STEPHEN HENSON & BEN LAURIE (2009). OpenSSL 0.9.8j. Open source implementation. URL <http://www.openssl.org/>. Refer to `openssl-0.9.8j.tar.gz`. Last download 21 April 2009. [156, 189]

RICHARD CRANDALL & CARL POMERANCE (2005). *Prime numbers – A computational perspective*. Springer-Verlag, 2nd edition. ISBN 0-387-25282-7. [5, 23, 39–42, 45, 50, 55–57, 62, 67, 141]

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>. [2, 87, 142, 146, 156–157, 160, 165]

MAX DEURING (1941). Die Typen der Multiplikatorenringe elliptischer Funktionenkörper. *Abhandlungen aus dem Mathematischen Seminar der Hamburgischen Universität* **14**, 197–272. [61]

KARL DICKMAN (1930). On the frequency of numbers containing prime factors of a certain relative magnitude. *Arkiv für Matematik, Astronomi och Fysik* **22A**(10), 1–14. [24]

WHITFIELD DIFFIE & MARTIN EDWARD HELLMAN (1976). New directions in cryptography. *IEEE Transactions on Information Theory* **IT-22**(6), 644–654. [81–82]

JOHANN PETER GUSTAV LEJEUNE DIRICHLET (1837). Beweis des Satzes, dass jede unbegrenzte arithmetische Progression, deren erstes Glied und Differenz ganze Zahlen ohne gemeinschaftlichen Factor sind, unendlich viele Primzahlen enthält. *Abhandlungen der Königlich Preussischen Akademie der Wissenschaften* 45–81. *Werke*, Erster Band, ed. L. KRONECKER, 1889, 315–342. Reprint by Chelsea Publishing Co., 1969. [11, 155]

- PIERRE DUSART (1998). *Autour de la fonction qui compte le nombre de nombres premiers*. Thèse de doctorat, Université de Limoges. URL http://www.unilim.fr/laco/theses/1998/T1998_01.html. [10–11, 88, 116, 127]
- HAROLD MORTIMER EDWARDS, JR. (1974). *Riemann's Zeta Function*. Pure and applied mathematics. Academic Press, New York. Republished 2001 by Dover Publications, Inc., ISBN 978-0-486-41740-0. [5, 15–18]
- HAROLD MORTIMER EDWARDS, JR. (2007). A Normal Form for Elliptic Curves. *Bulletin of the American Mathematical Society* **44**(3), 393–422. [3, 57, 69–73, 76–79]
- JAMES HENRY ELLIS (1970). The possibility of secure non-secret digital encryption. URL <http://cryptocellar.web.cern.ch/cryptocellar/cesg/possnse.pdf>. Last download 12 May 2009. [141]
- PAUL ERDŐS (1950). On almost primes. *The American Mathematical Monthly* **57**, 404–407. [40]
- LEONHARD PAUL EULER (1730). De progressionibus transcendentibus seu quarum termini generales algebraice dari nequeunt. *Commentarii academiae scientiarum Petropolitanae* **5**, 36–57. Eneström 19. *Opera Omnia*, series 1, volume 14, 1–24.
- LEONHARD PAUL EULER (1737). Variarum observationum circa series infinitas. *Commentarii academiae scientiarum Petropolitanae* **9**, 160–188. Eneström 72. *Opera Omnia*, series 1, volume 14, 217–244. [12]
- LEONHARD PAUL EULER (1741). Theorematum Quorundam ad Numeros Primos Spectantium Demonstratio. *Novi commentarii academiae scientiarum imperialis Petropolitanae* **8**, 141–146. Eneström 54. *Opera Omnia*, series 1, volume 2, B. G. Teubner, Leipzig, 1915, 33–37. [35]
- LEONHARD PAUL EULER (1755). *Institutiones calculi integralis*, volume 2. Academiae scientiarum Petropolitanae, St. Petersburg, chapter 5 and 6. Eneström 212. *Opera Omnia*, series 1, volume 10, 1–24. [16]
- LEONHARD PAUL EULER (1760/61). Theoremata arithmetica nova methodo demonstrata. *Novi commentarii academiae scientiarum imperialis Petropolitanae* **8**, 74–104. Summarium ibidem 15–18. Eneström 271. *Opera Omnia*, series 1, volume 2, B. G. Teubner, Leipzig, 1915, 531–555. [32]
- LEONHARD PAUL EULER (1761). Theoremata circa residua ex divisione potestatum relicta. *Novi commentarii academiae scientiarum imperialis Petropolitanae* **7**, 49–82. Eneström 262. *Opera Omnia*, series 1, volume 2, B. G. Teubner, Leipzig, 1915, 493–518. [35]
- PIERRE DE FERMAT (1640). Letter to Bessy. In *Œuvres de Fermat*, PAUL TANNERY & CHARLES HENRY, editors, volume 2, Correspondance, 206–212. Gauthier-Villars, Paris, 1894.
- PIERRE DE FERMAT (1647). Letter to Mersenne. In *Œuvres de Fermat*, PAUL TANNERY & CHARLES HENRY, editors, volume 2, Correspondance, 253–256. Gauthier-Villars, Paris, 1894.

М. М. Артюхов (1966/67). Некоторые критерии простоты чисел, связанные с малой теоремой Ферма (Certain criteria for the primality of numbers connected with Fermat's little theorem). *Acta Arithmetica* **12**, 355–364. [45]

KEVIN FORD (2002a). Vinogradov's integral and bounds for the Riemann zeta function. *Proceedings of the London Mathematical Society (3)* **85**, 565–633. URL <http://dx.doi.org/10.1112/S0024611502013655>. [9, 88, 114]

KEVIN FORD (2002b). Zero-free regions for the Riemann zeta function. In *Number Theory for the Millenium (Urbana, IL, 2000)*, M. A. BENNETT, BRUCE C. BERNDT, N. BOSTON, H. G. DIAMOND, ADOLF J. HILDEBRAND & W. PHILIPP, editors, volume II, 25–56. A. K. Peters. ISBN 978-1568811468. URL <http://www.math.uiuc.edu/~ford/wwwpapers/zeros.pdf>. [114]

JENS FRANKE & THORSTEN KLEINJUNG (2005). RSA 640. URL <http://www.crypto-world.com/announcements/rsa640.txt>. [1, 129]

JENS FRANKE & THORSTEN KLEINJUNG (2006). Continued Fractions and Lattice Sieving. *unpublished* URL <http://www.math.uni-bonn.de/people/thor/frac.ps>. [2]

FREE SOFTWARE FOUNDATION (2009). GNU Crypto. Open source implementation. URL <http://www.gnu.org/software/gnu-crypto/>. Refer to `gnu-crypto-2.0.1.tar.bz2`. Last download 21 April 2009. [156, 191]

STEVEN D. GALBRAITH (2011). *Mathematics of Public Key Cryptography*. Cambridge University Press. [82–83]

JOACHIM VON ZUR GATHEN & JÜRGEN GERHARD (1999). *Modern Computer Algebra*. Cambridge University Press, Cambridge, UK, First edition. ISBN 0-521-64176-4. URL <http://cosec.bit.uni-bonn.de/science/mca/>. Other available editions: Second edition 2003, Chinese edition, Japanese translation. [142]

JOACHIM VON ZUR GATHEN, TIM GÜNEYSU, ANTON KARGL, DANIEL LOEBENBERGER, CHRISTOF PAAR & JENS PUTZKA (2007). Faktorisierung großer Zahlen: Hardware für Elliptische Kurven Faktorisierung. Technical report, HGI Bochum, b-it Bonn & Siemens AG München. [134]

PIERRICK GAUDRY & DAVID LUBICZ (2009). The arithmetic of characteristic 2 Kummer surfaces and of elliptic Kummer lines. *Finite Fields and Their Applications* **15**(2), 246–260. ISSN 1071-5797. [3, 70–71]

JOHANN CARL FRIEDRICH GAUSS (1801). *Disquisitiones Arithmeticae*. Gerh. Fleischer Iun., Leipzig. English translation by ARTHUR A. CLARKE, Springer-Verlag, New York, 1986. [36]

JOHANN CARL FRIEDRICH GAUSS (1849). Brief an Encke, 24. Dezember 1849. In *Werke* II, Handschriftlicher Nachlass, 444–447. Königliche Gesellschaft der Wissenschaften, Göttingen, 1863. Reprinted by Georg Olms Verlag, Hildesheim New York, 1973. [9]

ODED GOLDBREICH (2001). *Foundations of Cryptography*, volume I: Basic Tools. Cambridge University Press, Cambridge. ISBN 0-521-79172-3. [167]

LOUIS GOUBIN & MITSURU MATSUI (editors) (2006). *Cryptographic Hardware and Embedded Systems, Workshop, CHES'06*, Yokohama, Japan, volume 4249 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, Heidelberg. ISBN 978-3-540-46559-1. ISSN 0302-9743. URL <http://dx.doi.org/10.1007/11894063>.

ANDREW GRANVILLE (2008). Smooth numbers: computational number theory and beyond. In *Algorithmic Number Theory: Lattices, Number Fields, Curves and Cryptography*, JOSEPH P. BUHLER & PETER STEVENHAGEN, editors, number 44 in Mathematical Sciences Research Institute Publications, 69–82. Cambridge University Press, New York. ISBN 978-0-521-80854-5. URL <http://www.math.leidenuniv.nl/~psh/ANTproc/09andrew.pdf>. [2, 23, 86]

TIM GÜNEYSU, CHRISTOF PAAR, GERD PFEIFFER & MANFRED SCHIMMLER (2008). Enhancing COPACOBANA for Advanced Applications in Cryptography and Cryptanalysis. In *International Conference on Field Programmable Logic and Applications, 2008 (FPL 2008)*, 675–678. IEEE Computer Society Press, Heidelberg, Germany. URL <http://dx.doi.org/10.1109/FPL.2008.4630037>. [1, 130]

JACQUES SALOMON HADAMARD (1896). Sur la distribution des zéros de la fonction $\zeta(s)$ et ses conséquences arithmétiques. *Bulletin de la Société mathématique de France* **24**, 199–220. [9]

YASUFUMI HASHIMOTO (2009). On asymptotic behavior of composite integers $n = pq$. *Journal of Math-for-industry* **1**(2009A-6), 45–49. [156]

HELMUT HASSE (1933). Beweis des Analogons der Riemannsches Vermutung für die Artinschen und F. K. Schmidtschen Kongruenzzetafunktionen in gewissen elliptischen Fällen. Vorläufige Mitteilung. *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse* **42**, 253–262. [61]

DAVID HILBERT (1900). Mathematische Probleme. *Nachrichten von der Königlichen Gesellschaft der Wissenschaften zu Göttingen* 253–297. URL http://www-gdz.sub.uni-goettingen.de/cgi-bin/digbib.cgi?PPN252457811_1900. *Archiv für Mathematik und Physik*, 3. Reihe **1** (1901), 44–63 and 213–237. English translation: Mathematical Problems, *Bulletin of the American Mathematical Society* **8** (1902), 437–479.

ADOLF J. HILDEBRAND (1985). Integers free of large prime factors and the Riemann hypothesis. *Mathematika* **31**(2), 258–271. [25]

ADOLF J. HILDEBRAND (1986). On the Number of positive integers $\leq x$ and free of Prime factors $> y$. *Journal of Number Theory* **22**(3), 289–307. ISSN 0022-314X. URL [http://dx.doi.org/10.1016/0022-314X\(86\)90013-2](http://dx.doi.org/10.1016/0022-314X(86)90013-2). [25]

ADOLF J. HILDEBRAND & GÉRALD TENENBAUM (1993). Integers without large prime factors. *Journal de Théorie des Nombres de Bordeaux* **5**(2), 411–484. [23]

- IEEE WORKING GROUP (2000). IEEE 1363-2000: Standard Specifications For Public Key Cryptography. IEEE standard, IEEE, New York, NY 10017, USA. URL <http://grouper.ieee.org/groups/1363/P1363/>. [142, 156, 186–187]
- INTERNATIONAL ORGANIZATION FOR STANDARDS (2006). ISO/IEC 18033-2, Encryption algorithms — Part 2: Asymmetric ciphers. Technical report, International Organization for Standards. [142, 186]
- CARL GUSTAV JACOB JACOBI (1837). Über die Kreisteilung und ihre Anwendung auf die Zahlentheorie. *Monatsberichte der Königlich Preussischen Akademie der Wissenschaften, Berlin* 127–136.
- QIN JIUSHAO (1247). *Shushu Chiuchang (Mathematical Treatise in Nine Sections)*.
- JAKOB JONSSON & BURT KALISKI (2003). Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1. URL <http://tools.ietf.org/html/rfc3447>. RFC 3447. [142, 186]
- MARC JOYE & PASCAL PAILLIER (2006). Fast Generation of Prime Numbers on Portable Devices: An Update. In Goubin & Matsui (2006), 160–173. URL http://dx.doi.org/10.1007/11894063_13. [2, 142, 183]
- BENJAMIN JUSTUS (2009). On integers with two prime factors. *Albanian Journal of Mathematics* 3(4), 189–197. [159–160]
- BENJAMIN JUSTUS & DANIEL LOEBENBERGER (2010). Differential Addition in Generalized Edwards Coordinates. In *Proceedings of the 5th International Workshop on Security*, Kobe, Japan, November 2010, ISAO ECHIZEN, NOBORU KUNIHIRO & RYOICHI SASAKI, editors, volume 6434 of *Lecture Notes in Computer Science*, 316–325. Springer-Verlag, Berlin, Heidelberg. ISBN 978-3-642-16824-6. ISSN 0302-9743. URL http://dx.doi.org/10.1007/978-3-642-16825-3_21. [69, 72, 75–78]
- WOLFGANG KILLMANN & WERNER SCHINDLER (2008). A Design for a Physical RNG with Robust Entropy Estimators. In *CHES 2008*, ELISABETH OSWALD & PANKAJ ROHATGI, editors, number 5154 in LNCS, 146–163. ISBN 978-3-540-85052-6. URL http://dx.doi.org/10.1007/978-3-540-85053-3_10. [183]
- THORSTEN KLEINJUNG (2006). On Polynomial Selection for the General Number Field Sieve. *Mathematics of Computation* 75(256), 2037–2047. URL <http://dx.doi.org/10.1090/S0025-5718-06-01870-9>. [55]
- DONALD ERVIN KNUTH (1998). *The Art of Computer Programming, vol. 2, Seminumerical Algorithms*. Addison-Wesley, Reading MA, 3rd edition. ISBN 0-201-89684-2. First edition 1969. [30, 34, 180]
- NEAL KOBLITZ (1987). Elliptic Curve Cryptosystems. *Mathematics of Computation* 48(177), 203–209. [58]
- NIELS FABIAN HELGE VON KOCH (1901). Sur la distribution des nombres premiers. *Acta Mathematica* 24(1), 159–182. URL <http://dx.doi.org/10.1007/BF02403071>. [11, 88]

KONRAD KÖNIGSBERGER (2001). *Analysis I*. Springer, Berlin, Heidelberg, 5th edition. ISBN 3-540-41282-4. [14]

ALWIN REINHOLD KORSELT (1899). Problème chinois. *L'Intermédiaire des Mathématiciens* **6**, 143.

MAURICE KRAÏTCHIK (1922). *Théorie des Nombres*. Gauthier-Villars et Cie., Paris. [56]

ANDREY V. KULSHA (2008). Values of $\pi(x)$ and $\Delta(x)$ for different x 's. Web-page. URL <http://www.primefan.ru/stuff/primes/table.html>. Last visited 2 February 2009. [88]

SANDEEP KUMAR, CHRISTOF PAAR, JAN PELZL, GERD PFEIFFER & MANFRED SCHIMMLER (2006). Breaking Ciphers with COPACOBANA - A Cost-Optimized Parallel Code Breaker. In Goubin & Matsui (2006), 101–118. URL http://dx.doi.org/10.1007/11894063_9. [130]

JOSEPH LOUIS DE LAGRANGE (1770/71). Réflexions sur la résolution algébrique des équations. *Œuvres complètes* **3**, 205–421. Nouveaux mémoires de l'Académie royale des sciences et belles-lettres de Berlin. [32]

GABRIEL LAMÉ (1844). Note sur la limite du nombre des divisions dans la recherche du plus grand commun diviseur entre deux nombres entiers. *Comptes Rendus de l'Académie des Sciences Paris* **19**, 867–870. [29]

EDMUND GEORG HERMANN LANDAU (1909). *Handbuch der Lehre von der Verteilung der Primzahlen*, volume 1. B.G. Teubner, Leipzig, 1st edition. [23]

ADRIEN MARIE LEGENDRE (1798, An VI). *Essai sur la théorie des nombres*. Duprat, Paris.

ADRIEN-MARIE LEGENDRE (1830). *Théorie des Nombres*, volume 2. Firmin Didot frères, Paris, 4th edition.

ARJEN KLAAS LENSTRA & HENDRIK WILLEM LENSTRA, JR. (editors) (1993). *The development of the number field sieve*, number 1554 in Lecture Notes in Mathematics. Springer-Verlag, Berlin. [55]

HENDRIK WILLEM LENSTRA, JR. (1987). Factoring integers with elliptic curves. *Annals of Mathematics* **126**, 649–673. [57, 64]

XIAN-JIN LI (1997). The positivity of a sequence of numbers and the Riemann hypothesis. *Journal of Number Theory* **65**, 325–333. [40]

JOHN EDENSOR LITTLEWOOD (1912). Quelques conséquences de l'hypothèse que la fonction $\zeta(s)$ n'a pas de zéros dans le demi-plan $\Re(s) > \frac{1}{2}$. *Comptes Rendus des Séances de l'Académie des Sciences* **154**, 263–266. [19]

JOHN EDENSOR LITTLEWOOD (1914). Sur la distribution des nombres premiers. *Comptes Rendus des Séances de l'Académie des Sciences* **158**, 1869–1872. [10]

- DANIEL LOEBENBERGER & MICHAEL NÜSKEN (2010). Coarse-grained integers. *e-print arXiv:1003.2165v1* URL <http://arxiv.org/abs/1003.2165>. [85, 158, 174]
- DANIEL LOEBENBERGER & MICHAEL NÜSKEN (2011a). Analyzing standards for RSA integers. In *AFRICACRYPT 2011*, ABDERRAHMANE NITAJ & DAVID POINTCHEVAL, editors, volume 6737 of *Lecture Notes in Computer Science*, 260–277. Springer. ISBN 978-3-642-21968-9. ISSN 0302-9743. URL http://dx.doi.org/10.1007/978-3-642-21969-6_16. [141, 155–164, 167, 177]
- DANIEL LOEBENBERGER & MICHAEL NÜSKEN (2011b). Analyzing standards for RSA integers — full version. *Submitted to Journal of Cryptology* URL <http://arxiv.org/abs/1104.4356>. [177]
- DANIEL LOEBENBERGER & JENS PUTZKA (2009). Optimization strategies for hardware-based cofactorization. In *Selected Areas in Cryptography*, MICHAEL J. JACOBSON, VINCENT RIJMEN & REI SAFAVI-NAINI, editors, volume 5867 of *Lecture Notes in Computer Science*, 170–181. Berlin, Heidelberg. URL http://dx.doi.org/10.1007/978-3-642-05445-7_11. [129]
- FRANÇOIS ÉDOUARD ANATOLE LUCAS (1878). Théorie des fonctions numériques simplement périodiques. *American Journal of Mathematics* **1**, I: 184–240, II: 289–321.
- UELI M. MAURER (1995). Fast Generation of Prime Numbers and Secure Public-Key Cryptographic Parameters. *Journal of Cryptology* **8**(3), 123–155. URL <http://dx.doi.org/10.1007/BF00202269>. [145]
- 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/>. [142]
- FRANZ MERTENS (1897). Über eine zahlentheoretische Function. *Sitzungsberichte der Akademie der Wissenschaften, Wien, Mathematisch-Naturwissenschaftliche Classe* **106**, 761–830.
- GARY LEE MILLER (1975). Riemann’s Hypothesis and Tests for Primality. In *Proceedings of the Seventh Annual ACM Symposium on Theory of Computing*, Albuquerque NM, 234–239. ACM Press. [45–47]
- GARY LEE MILLER (1976). Riemann’s Hypothesis and Tests for Primality. *Journal of Computer and System Sciences* **13**, 300–317. [22]
- VICTOR SAUL MILLER (1986). Use of Elliptic Curves in Cryptography. In *Advances in Cryptology: Proceedings of CRYPTO ’85*, Santa Barbara, CA, HUGH C. WILLIAMS, editor, number 218 in *Lecture Notes in Computer Science*, 417–426. Springer-Verlag, Berlin. ISSN 0302-9743. [58]
- AUGUST FERDINAND MÖBIUS (1832). Über eine besondere Art von Umkehrung der Reihen. *Journal für die reine und angewandte Mathematik* **9**, 105–123.
- LOUIS MONIER (1980). Evaluation and comparison of two efficient probabilistic primality testing algorithms. *Theoretical Computer Science* **12**, 97–108. [47]

PETER LAWRENCE MONTGOMERY (1987). Speeding the Pollard and Elliptic Curve Methods of Factorization. *Mathematics of Computation* **48**(177), 243–264. [3, 63–64, 69–71]

MICHAEL A. MORRISON & JOHN BRILLHART (1975). A Method of Factoring and the Factorization of F_7 . *Mathematics of Computation* **29**(129), 183–205. [52]

NESSIE WORKING GROUP (2003). NESSIE D20 - NESSIE security report. Technical report, NESSIE. [142, 156, 186, 189]

JOHN VON NEUMANN (1951). Various techniques used in connection with random digits. Monte Carlo methods. *National Bureau of Standards, Applied Mathematics Series* **12**, 36–38. [178]

SIR ISAAC NEWTON (1671). *The Method of Fluxions and Infinite Series with its Application to the Geometry of Curve-Lines*. Henry Woodfall, London.

NIST (2009). FIPS 186-3: Digital Signature Standard (DSS). Technical report, Information Technology Laboratory, National Institute of Standards and Technology. [142, 156, 186–188]

MICHAEL NÜSKEN (2006-2011). Private communication. [85, 99–103, 116–119]

JOSEPH OESTERLÉ (1979). Versions effectives du théorème de Chebotarev sous l'hypothèse de Riemann généralisée. *Société Mathématique de France, Astérisque* **61**, 165–167. [20]

TATSUAKI OKAMOTO & SHIGENORI UCHIYAMA (1998). A new public-key cryptosystem as secure as factoring. In *Advances in Cryptology: Proceedings of EUROCRYPT 1998*, Helsinki, Finland, KAISA TELLERVO NYBERG, editor, number 1403 in Lecture Notes in Computer Science, 308–318. Springer-Verlag, Berlin, Heidelberg. ISBN 3-540-64518-7. ISSN 0302-9743. URL <http://dx.doi.org/10.1007/BFb0054135>. [171]

KATSUYUKI OKEYA & KOUICHI SAKURAI (2001). Efficient elliptic curve cryptosystem from a scalar multiplication algorithm with recovery of the y-coordinate on a Montgomery-form elliptic curve. In *Cryptographic Hardware and Embedded Systems, Workshop, CHES'01*, Paris, France, ÇEMAL K. KOÇ, DAVID NACCACHE & CHRISTOF PAAR, editors, number 2162 in Lecture Notes in Computer Science, 126–141. Springer-Verlag, Berlin, Heidelberg. ISBN 3-540-42521-7. ISSN 0302-9743. URL <http://dx.doi.org/>. [76]

JÁNOS PINTZ (1984). On the remainder term of the prime number formula and the zeros of Riemann's zeta-function. In *Number Theory Noordwijkerhout 1983*, HENDRIK JAGER, editor, number 1068 in Lecture Notes in Mathematics. Springer-Verlag, Berlin, Heidelberg. ISBN 978-3-540-13356-8. ISSN 0075-8434 (Print) 1617-9692 (Online). URL <http://dx.doi.org/10.1007/BFb0099452>.

LEONARDO PISANO (1202). *Liber Abaci*.

JOHN MICHAEL POLLARD (1974). Theorems on factorization and primality testing. *Proceedings of the Cambridge Philosophical Society* **76**, 521–528. [57, 64, 67]

JOHN MICHAEL POLLARD (1978). Monte Carlo Methods for Index Computation (mod p). *Mathematics of Computation* **32**(143), 918–924. [56]

JOHN MICHAEL POLLARD (1988). Factoring with cubic integers. In Lenstra & Lenstra (1993), 4–10. URL <http://dx.doi.org/10.1007/BFb0091536>. [52–55]

CARL POMERANCE (1985). The quadratic sieve factoring algorithm. In *Advances in Cryptology: Proceedings of EUROCRYPT 1984*, Paris, France, T. BETH, N. COT & I. INGEMARSSON, editors, number 209 in Lecture Notes in Computer Science, 169–182. Springer-Verlag, Berlin. ISSN 0302-9743. [51]

CARL POMERANCE (1996). A tale of two sieves. *Notices of the American Mathematical Society* **43**(12), 1437–1485. [51]

CARL POMERANCE, JOHN LEWIS SELFRIDGE & SAMUEL STANDFIELD WAGSTAFF, JR. (1980). The pseudoprimes to $25 \cdot 10^9$. *Mathematics of Computation* **35**, 1003–1025. [46]

THEO DE RAADT, NIELS PROVOS, MARKUS FRIEDL, BOB BECK, AARON CAMPBELL & DUG SONG (2009). OpenSSH 2.1.1. Open source implementation. URL <http://www.openssh.org/>. Refer to `openssh-2.1.1p4.tar.gz`. Last download 21 April 2009. [189]

MICHAEL OSER RABIN (1980). Probabilistic Algorithms for Testing Primality. *Journal of Number Theory* **12**, 128–138. [45–47]

V. RAMASWAMI (1949). The number of positive integers $\leq x$ and free of prime divisors $> x^c$, and a problem of S. S. Pillai. *Duke Mathematical Journal* **16**, 99–109. [24]

MICHAEL RICHARDSON, PAUL WOUTERS, ANTONY ANTONY, KEN BANTOFT, BART TROJANOWSKI, HERBERT XU, DAVID MCCULLOUGH, D. HUGH REDELMEIER, ANDREAS STEFFEN, DR{WHO} ON FREENODE, JACCO DE LEEUW, MATHIEU LAFON, NATE CARLSON, STEPHEN BEVAN, TUOMO SOINI, MATTHEW GALGOCI, MILOSLAV TRMAC, AVESH AGARWAL, HIREN JOSHI CYBEROAM, SHINGO YAMAWAKI & WILLY AT W.ODS.ORG (2009). Openswan 2.6.20. Open source implementation. URL <http://www.openswan.org/>. Refer to `openswan-2.6.20.tar.gz`. Last download 21 April 2009. [190]

GEORG FRIEDRICH BERNHARD RIEMANN (1859). Ueber die Anzahl der Primzahlen unter einer gegebenen Grösse. *Monatsberichte der Berliner Akademie* 145–153. *Gesammelte Mathematische Werke*, ed. HEINRICH WEBER, Teubner Verlag, Leipzig, 1892, 177–185.

RONALD LINN RIVEST, ADI SHAMIR & LEONARD MAX ADLEMAN (1977). A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Technical Report MIT/LCS/TM-82, Massachusetts Institute of Technology, Laboratory for Computer Science, Cambridge, Massachusetts. [141]

- RONALD LINN RIVEST, ADI SHAMIR & LEONARD MAX ADLEMAN (1978). A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM* **21**(2), 120–126. [82, 141]
- HERBERT ROBBINS (1955). A Remark on Stirling’s Formula. *The American Mathematical Monthly* **62**(1), 26–29. URL <http://www.jstor.org/stable/2308012>. [105]
- JOHN BARKLEY ROSSER & LOWELL SCHOENFELD (1962). Approximate formulas for some functions of prime numbers. *Illinois Journal of Mathematics* **6**, 64–94. [88–90, 116]
- JOHN BARKLEY ROSSER & LOWELL SCHOENFELD (1975). Sharper Bounds for the Chebyshev Functions $\vartheta(x)$ and $\psi(x)$. *Mathematics of Computation* **29**(129), 243–269. URL <http://www.jstor.org/stable/2005479>. [88]
- RSA LABORATORIES (2000). RSAES-OAEP Encryption Scheme. Algorithm specification and supporting documentation, RSA Security Inc., Bedford, MA 01730 USA. URL ftp://ftp.rsasecurity.com/pub/rsalabs/rsa_algorithm/rsa-oaep_spec.pdf. [142, 156, 186]
- RSA LABORATORIES (2007). The RSA Challenge Numbers. [129]
- YANNICK SAOUTER, XAVIER GOURDON & PATRICK DEMICHEL (2011). An improved lower bound for the de Bruijn-Newman constant. *Mathematics of Computation* **80**, 2281–2287. [18]
- WERNER SCHINDLER (2008a). Evaluation Criteria for Physical Random Number Generators. Technical report, Bundesamt für Sicherheit in der Informationstechnik (BSI), Bonn. [182]
- WERNER SCHINDLER (2008b). Random Number Generators for Cryptographic Applications. Technical report, Bundesamt für Sicherheit in der Informationstechnik (BSI), Bonn. [182]
- BRUCE SCHNEIER (1996). *Applied cryptography: protocols, algorithms, and source code in C*. John Wiley & Sons, New York, 2nd edition. ISBN 0-471-12845-7, 0-471-11709-9, XXIII, 758. [142]
- LOWELL SCHOENFELD (1976). Sharper bounds for the Chebyshev functions $\vartheta(x)$ and $\psi(x)$. II. *Mathematics of Computation* **30**(134), 337–360. [11, 88]
- RENÉ SCHOOF (1995). Counting points on elliptic curves over finite fields. *Journal de Théorie des Nombres de Bordeaux* **7**, 219–254. [61–62]
- DANIEL SHANKS (1969). Class number, a theory of factorization, and genera. In *Number Theory Institute 1969*, number 20 in Proceedings of Symposia in Pure Mathematics, 415–440. American Mathematical Society. [56]
- VICTOR SHOUP (1997). Lower Bounds for Discrete Logarithms and Related Problems. In *Advances in Cryptology: Proceedings of EUROCRYPT 1997*, Konstanz, Germany, SPRINGER-VERLAG, editor, number 1233 in Lecture Notes in

Computer Science, 256–266. Rüşchlikon, Switzerland. ISSN 0302-9743. URL <http://www.shoup.net/papers/>. [56]

TOMÁS OLIVEIRA E SILVA (2003). Fast implementation of the segmented sieve of Eratosthenes. WWW. URL http://www.ieeta.pt/~tos/software/prime_sieve.html. Simple implementation of the segmented sieve of Eratosthenes, released under the version 2 (or any later version) of the GNU general public license. Last visited 4 February 2009. [88]

JOSEPH HILLEL SILVERMAN (1986). *The Arithmetic of Elliptic Curves*, volume 106 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 2nd edition. [58]

MATTHEW SKALA, MICHAEL ROTH, NIKLAS HERNAEUS, RÉMI GUYOMARCH & WERNER KOCH (2009). GnuPG. Open source implementation. URL <http://www.gnupg.org/>. Refer to `gnupg-2.0.9.tar.bz2`. Last download 21 April 2009. [156, 190]

ROBERT MARTIN SOLOVAY & VOLKER STRASSEN (1977). A fast Monte-Carlo test for primality. *SIAM Journal on Computing* **6**(1), 84–85. Erratum in **7** (1978), p. 118. [45]

JACQUES CHARLES FRANÇOIS STURM (1835). Mémoire sur la résolution des équations numériques. *Mémoires présentés par divers savants à l'Académie des Sciences de l'Institut de France* **6**, 273–318. [108]

TSUYOSHI TAKAGI (1998). Fast RSA-type cryptosystem modulo p^kq . In *Advances in Cryptology: Proceedings of CRYPTO '98*, Santa Barbara, CA, H. KRAWCZYK, editor, number 1462 in *Lecture Notes in Computer Science*, 318–326. Springer-Verlag, Berlin, Heidelberg. ISBN 3-540-64892-5. ISSN 0302-9743. URL <http://dx.doi.org/10.1007/BFb0055715>. [171]

CHARLES-JEAN ÉTIENNE GUSTAVE NICOLAS, BARON DE LA VALLÉE POUSSIN (1896). Recherches analytiques sur la théorie des nombres premiers. *Annales de la Société Scientifique de Bruxelles* **20**, 183–256 and 281–397. [9]

ARNOLD WALFISZ (1936). Zur additiven Zahlentheorie. II. *Mathematische Zeitschriften* **40**(1), 592–608. URL <http://dx.doi.org/10.1007/BF01218882>.

ARNOLD WALFISZ (1963). *Weylsche Exponentialsummen in der neueren Zahlentheorie*. Number XV in *Mathematische Forschungsberichte*. VEB Deutscher Verlag der Wissenschaften, Berlin, 231 pages. [9]

LAWRENCE CLINTON WASHINGTON (2003). *Elliptic Curves — Number Theory and Cryptography*. Discrete Mathematics and its Applications. CRC Press, Boca Raton, FL, USA. ISBN 1-58488-365-0. [61, 65–66]

BENNE M. M. DE WEGER (2007–2011). Private communication. [2, 156]

KARL WEISTRASS (1895a). *Mathematische Werke – Erster Band*. Mayer & Müller. [58]

KARL WEISTRASS (1895b). *Mathematische Werke – Zweiter Band*. Mayer & Müller. [58]

ANDREW WILES (1995). Modular elliptic curves and Fermat's Last Theorem. *Annals of Mathematics* **142**, 443–551. [58]

HERBERT SAUL WILF (1994). *generatingfunctionology*. Academic Press, 2nd edition. URL <http://www.math.upenn.edu/~wilf/DownldGF.html>. First edition 1990. [122]

PETRA WOHLMACHER (2009). Bekanntmachung zur elektronischen Signatur nach dem Signaturgesetz und der Signaturverordnung (Übersicht über geeignete Algorithmen). *Bundesanzeiger* **2009**(13), 346–350. ISSN 0344-7634. Preprint at <http://www.bundesnetzagentur.de/media/archive/14953.pdf>. [142, 145, 186, 192]

SUN ZI (ca 473 A.D.). *Sunzi suanjing* (*Sun Zi's Mathematical Manual*).

Players

The numbers in brackets at the end of a person entry point to the pages on which they are cited. For all of the images a thorough copyright research was performed. If you feel that in one of the pictures we missed corresponding author information, please do not hesitate to contact us.



CLAUDE GASPARD BACHET DE MÉZIRIAC (1581–1638). *9 October 1581, Bourg-en-Bresse. †26 February 1638. URL http://de.wikipedia.org/wiki/Claude_Gaspard_Bachet_de_Méziriac. [30]



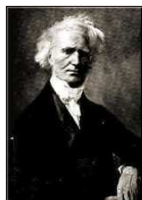
JACOB BERNOULLI, also JAMES BERNOULLI or JACQUES BERNOULLI (1654–1705). *27 December 1654, Basel, Switzerland. †16 August 1705, Basel, Switzerland. URL http://en.wikipedia.org/wiki/Jacob_Bernoulli. [15–16]



BERNARD FRÉNICLE DE BESSY (ca. 1605–1675). *?/?/, Paris, France. †?/?/. URL http://en.wikipedia.org/wiki/Bernard_Frénicle_de_Bessy. [35]



ÉTIENNE BÉZOUT (1730–1783). *31 March 1730, Nemours, Seine-et-Marne. †27 September 1783, Basses-Loges (close to Fontainebleau). URL http://en.wikipedia.org/wiki/Etienne_Bezout. [6, 30–31]



JACQUES PHILIPPE MARIE BINET (1786–1856). *2 February 16786, Rennes, France. †12 May 1856, Paris, France. URL http://en.wikipedia.org/wiki/Jacques_Philippe_Marie_Binet. [28–30]



Александр Адольфович Бухштаб, also ALEKSANDR ADOLFOVICH BUHŠTAV (1905–1990). *4 October 1905, Stavropol, Russia. †27 February 1990. URL <http://genealogy.math.ndsu.nodak.edu/id.php?id=29696>. Image from URL <http://mi.mathnet.ru/umn4571>. [25–26]



ROBERT DANIEL CARMICHAEL (1879–1967). *1 March 1879, Goodwater, Alabama, USA. †2 May 1967. URL http://en.wikipedia.org/wiki/Robert_Daniel_Carmichael. Image from URL <http://m1.ikiwq.com/img/x1/HpiUAWHZWhRdwPAqAdmAa.jpg>. [42–46]



Пафнύтий Львóвич Чебышёв, also PAFNUTY LVOVICH CHEBYSHEV (1821–1894). *16 May 1821, Borovsk, Kaluga, Russian Empire. †8 December 1894, Saint Petersburg, Russian Empire. URL http://en.wikipedia.org/wiki/Pafnuty_Chebyshev. [9]



KARL DICKMAN (ca. 1862–1940). *?/?/? †?/?/?. URL <http://www.math.kth.se/matstat/fofu/reports/PoiDir.pdf>. [24–25]



Διόφαντος ὁ Ἀλεξανδρεύς, also DIOPHANTUS OF ALEXANDRIA (ca. 200–284). *?/?/? †?/?/?. URL <http://en.wikipedia.org/wiki/Diophantus>. [58]



JOHANN PETER GUSTAV LEJEUNE DIRICHLET (1805–1859). *13 February 1805, Düren, French Empire. †5 May 1859, Göttingen, Hanover. URL <http://en.wikipedia.org/wiki/Dirichlet>. [11, 20–22]



JOHANN FRANZ ENCKE (1791–1865). *23 September 1791, Hamburg, Germany. †26 August 1865, Spandau, Germany. URL http://en.wikipedia.org/wiki/Johann_Franz_Encke. [8]



Ερατοσθένης ὁ Κυρηναῖος, also ERATOSTHENES OF CYRENE (276–194 BC). *?/?/, Cyrene, Greece. †?/?/, Alexandria, Egypt. URL <http://en.wikipedia.org/wiki/Eratosthenes>. [7]



Εὐκλείδης ὁ Ἀλεξάνδρεια, also EUCLID OF ALEXANDRIA (365–300 BC). *?/?/, presumably Alexandria or Athens. †?/?/. URL <http://en.wikipedia.org/wiki/Euclid>. Image: The School of Athens, Fresco, Stanza della Segnatura, Palazzi Pontifici, Vatican. [6–7, 27–33, 65–66, 82]



LEONHARD PAUL EULER (1707–1783). *15 April 1707, Basel, Switzerland. †18 September 1783, St. Petersburg. URL http://en.wikipedia.org/wiki/Leonhard_Paul_Euler. Portrait by Johann Georg Brucker, 1756. [7–8, 11–13, 17–21, 32, 35–36, 43–46, 82]



PIERRE DE FERMAT (1601/1607/1608–1665). *17 August 1601, Beaumont-de-Lomagne, France. †12 February 1665, Beaumont-de-Lomagne, France. URL http://en.wikipedia.org/wiki/Pierre_de_Fermat. Portrait by an unknown artist. [35, 40–47, 50–51, 58]



LEONARDO FIBONACCI, also LEONARDO OF PISA, LEONARDO PISANO BIGOLLO, LEONARDO BONACCI or FIBONACCI (ca. 1170–1250). *?/?/, Pisa, Italy. †?/?/, Pisa, Italy. URL <http://en.wikipedia.org/wiki/Fibonacci>. Portrait by an unknown artist. [28–30]



FERDINAND GEORG FROBENIUS (1849–1917). *26 October 1849, Berlin. †3 August 1917, Charlottenburg, heute ein Ortsteil von Berlin. URL http://en.wikipedia.org/wiki/Ferdinand_Georg_Frobenius. Image from URL http://owpdb.mfo.de/detail?photo_id=10587. Bildarchiv des Mathematischen Forschungsinstituts Oberwolfach. [62]



JOHANN CARL FRIEDRICH GAUSS, also CAROLUS FRIDERICUS GAUSS (1777–1855). *30 April 1777, Braunschweig, Electorate of Brunswick-Lüneburg, Holy Roman Empire. †23 February 1855, Göttingen, Kingdom of Hanover. URL <http://en.wikipedia.org/wiki/Gauss>. Portrait by Gottlieb Biermann, 1887, copied from a painting of Christian Albrecht Jensen, 1840. [8–10, 18, 36–37]



JACQUES SALOMON HADAMARD (1865–1963). *8 December 1865, Versailles, France. †17 October 1963, Paris, France. URL <http://en.wikipedia.org/wiki/Hadamard>. [9]



HELMUT HASSE (1898–1979). *25 August 1898, Kassel. †26 December 1979, Ahrensburg bei Hamburg. URL http://en.wikipedia.org/wiki/Helmut_Hasse. Bildarchiv des Mathematischen Forschungsinstituts Oberwolfach. [61, 66]



DAVID HILBERT (1862–1943). *23 January 1862, Königsberg or Wehlau (today Znamensk, Kaliningrad Oblast), Province of Prussia. †14 February 1943, Göttingen, Germany. URL http://en.wikipedia.org/wiki/David_hilbert. [5]



CARL GUSTAV JACOB JACOBI (1804–1851). *10 December 1804, Potsdam, Kingdom of Prussia. †18 February 1851, Berlin, Kingdom of Prussia. URL http://en.wikipedia.org/wiki/Carl_Gustav_Jakob_Jacobi. [35–37, 43–45]



QIN JIUSHAO (1202–1261). *?/?/, Puzhou (Anyue), Szechwan province, China. †?/?/, Meizhou (now Meixian), Guangtong province, China. URL http://www-history.mcs.st-andrews.ac.uk/Biographies/Qin_Jiushao.html. Image from URL http://www.confuciusonline.com/wp-content/uploads/auto_save_image/2010/10/073110E16.jpg. [34]



NIELS FABIAN HELGE VON KOCH (1870–1924). *15 January 1870, Stockholm, Sweden. †11 March 1924, Stockholm, Sweden. URL http://en.wikipedia.org/wiki/Helge_von_Koch. [20]



ALWIN REINHOLD KORSELT (1864–1947). *17 March 1864, Mittelherwigsdorf, Germany. †4 February 1947, Plauen, Germany. URL http://de.wikipedia.org/wiki/Alwin_Reinhold_Korselt. [42–44]



JOSEPH LOUIS DE LAGRANGE (1736–1813). *25 January 1736, Turin, Sardinia. †10 April 1813, Paris, France. URL <http://en.wikipedia.org/wiki/Lagrange>. [31]



EDMUND GEORG HERMANN LANDAU (1877–1938). *14 February 1877, Berlin, Germany. †19 February 1938, Berlin, Germany. URL http://en.wikipedia.org/wiki/Edmund_Landau. [22–23]



ADRIEN-MARIE LEGENDRE (1752–1833). *18 September 1752, Paris, France. †10 January 1833, Paris, France. URL http://en.wikipedia.org/wiki/Adrien-Marie_Legendre. Watercolor caricature by Julien-Leopold Boilly, 1820. [9, 34–36, 55, 61]



FRANÇOIS ÉDOUARD ANATOLE LUCAS (1842–1891). *4 April 1842, Amiens, France. †3 October 1891, Paris, France. URL http://en.wikipedia.org/wiki/Édouard_Lucas. [7]



ROBERT HJALMAR MELLIN (1854–1933). *19 June 1854, Liminka, Finland. †5 April 1933, Helsinki, Finland. URL http://en.wikipedia.org/wiki/Hjalmar_Mellin. [19]



MARIN MERSENNE, also MARIN MERSENNUS or LE PÈRE MERSENNE (1588–1648). *8 September 1588, Oizé, Maine (present day Sarthe), France. †1 September 1648, Paris, France. URL http://en.wikipedia.org/wiki/Marin_Mersenne. Portrait by Philippe de Champaigne. [51]



FRANZ MERTENS (1840–1927). *20 March 1840, Środa, Prussia. †5 March 1927, Vienna, Austria. URL http://en.wikipedia.org/wiki/Franz_Mertens. Image from URL <http://www-history.mcs.st-andrews.ac.uk/PictDisplay/Mertens.html>. [19]



AUGUST FERDINAND MÖBIUS (1790–1868). *17 November 1790, Schulpforta, Saxony-Anhalt, Germany. †16 August 1868, Leipzig, Germany. URL http://en.wikipedia.org/wiki/August_Ferdinand_Möbius. Portrait by Adolf Neumann. [17–19]



ABRAHAM DE MOIVRE (1667–1754). *26 May 1667, Vitry-le-François, Champagne, France. †27 November 1754, London, England. URL http://en.wikipedia.org/wiki/Abraham_de_Moivre. Potrait by an unknown artist. [28]



SIR ISAAC NEWTON (1643–1727 (greg.)). *4 January 1643, Woolsthorpe-by-Colsterworth in Lincolnshire, England. †31 March 1727, Kensington, England. URL http://de.wikipedia.org/wiki/Isaac_Newton. Portrait by Godfrey Kneller, 1689. [37–38]



GEORG FRIEDRICH BERNHARD RIEMANN (1826–1866). *17 September 1826, Breselenz, Kingdom of Hanover. †20 July 1866, Selasca, Kingdom of Italy. URL http://en.wikipedia.org/wiki/Bernhard_Riemann. [5, 11–22, 25, 47, 70, 87–88, 91, 114, 147, 150, 157–163, 172–175]



LOWELL SCHOENFELD (1920–2002). *1 April 1920. †6 February 2002. URL http://en.wikipedia.org/wiki/Lowell_Schoenfeld. Image from URL http://owpdb.mfo.de/detail?photo_id=5935. Bildarchiv des Mathematischen Forschungsinstituts Oberwolfach. [20]



JOHN LEWIS SELFRIDGE (1927–2010). *17 February 1927, Ketchikan, Alaska, USA. †31 October 2010, DeKalb, Illinois, USA. URL http://en.wikipedia.org/wiki/John_Selfridge. [45]



CARL LUDWIG SIEGEL (1896–1981). *31 December 1896, Berlin, Germany. †4 April 1981, Göttingen, Germany. URL http://en.wikipedia.org/wiki/Carl_Ludwig_Siegel. Image from URL http://owpdb.mfo.de/detail?photo_id=3840. Bildarchiv des Mathematischen Forschungsinstituts Oberwolfach. [11]



CHARLES-JEAN ÉTIENNE GUSTAVE NICOLAS, BARON DE LA VALLÉE POUSSIN (1866–1962). *14 August 1866, Leuven, Belgium. †2 March 1962, Watermael-Boitsfort, Brussels, Bergium. URL http://en.wikipedia.org/wiki/Charles_Jean_de_la_Vallée-Poussin. Portrait by Charles Levieux. [9]



ARNOLD WALFISZ (1892–1962). *2 June 1892, Warsaw, Congress Poland, Russian Empire. †29 May 1962, Tbilisi, Georgia, Soviet Union. URL http://en.wikipedia.org/wiki/Arnold_Walfisz. [11]

*Bundhof*

KARL THEODOR WILHELM WEIERSTRASS (1815–1897). *31 October 1815, Ostenfelde, Westphalia, Germany. †19 February 1897, Berlin, Germany. URL http://en.wikipedia.org/wiki/Karl_Weierstrass. Portrait by an unknown artist. [58–61, 70–73]



SUN ZI (ca. 400–460). *?/?/, China. †?/?/, China. URL http://www-groups.dcs.st-and.ac.uk/~history/Biographies/Sun_Zi.html. [34]

Index

A page number is underlined (for example: 123) when it represents the definition or the main source of information about the index entry. For several key words that appear frequently only ranges of pages or the most important occurrences are indexed.

- Accredited Standards Committee X9 142,
186, 188, 197
addition chain 65, 67, 79
Adleman, Leonard Max 56, 82, 141, 197,
208, 209
AfricaCrypt 141, 177
Agarwal, Avesh 190, 208
Agrawal, Manindra 49, 50, 197
AKS test *see* primality test, AKS
Alford, William Robert 42, 197
algebraic integer 53–55
algorithmic number theory 24–67
Alice 81, 82
Ammon, Liselotte xi
analytic number theory 8, 12–22, 27
ancient world 5, 57
ANSI
 X9.31 188
 X9.44 142, 186, 188–189, 192, 193
Antony, Antony 190, 208
asymptote 26
at w.ods.org, willy *see* w.ods.org
Avanzi, Roberto M. 71, 197

Baby-step-giant-step 56
Bach, Eric 49, 197
Bachet de Méziriac, Claude Gaspar ... 30, 197,
213
balanced
 function 96
 notion *see* notion, balanced
banner 178, 178, 181, 185
Bantoft, Ken 190, 208
Beck, Bob 189, 208
Bellman, Richard 132, 198
Bennett, M. A. 202
Bergmann, Herta xi
Berndt, Bruce C. 202
Bernoulli, Jacob (1654–1705) ... 15, 16, 198, 213
 numbers 15, 16
Bernstein, Daniel Julius 64, 69, 70, 72, 198
de Bessy, Bernard Frénicle (ca. 1605–1675)
 35, 213

Beth, T. 208
Bevan, Stephen 190, 208
Bézout, Étienne (1730–1783) 6, 30, 31, 198,
213
 identity 6, 31
Binet, Jacques Philippe Marie (1786–1856)
 28–30, 198, 214
 formula 29, 30
Birkner, Peter 69, 198
Blömer, Johannes 172, 198
Bob 81, 82
Bochum 1
Boneh, Dan 172, 198, 199
Boston, N. 202
Boztas, Serdar 198
Brandt, Jørgen 2, 142, 181, 183, 199
Brickell, E. F. 199
Brier, Éric 76, 199
Brillhart, John 45, 52, 199, 207
de Bruijn, Nicolaas Govert 25, 199
BSI *see* Bundesamt für Sicherheit in der
 Informationstechnik
Buhler, Joseph P. 54, 55, 199, 203
Бухштаб, Александр Адольфович (1905–1990)
 25, 26, 86, 199, 214
Bundesamt für Sicherheit in der
 Informationstechnik 1
Bundesnetzagentur 142, 145, 186, 192

Calgary 129
Campbell, Aaron 189, 208
Canada 129
Canfield, Earl Rodney 25, 52, 199
Carlson, Nate 190, 208
Carmichael, Robert Daniel (1879–1967) ... 42,
44, 46, 199, 214
 number 42, 42, 44, 46
Cassels, John William Scott 60, 199
Castryc, Wouter 3, 70, 71, 74, 199
Чебышёв, Пафнутий Львович (1821–1894)
 9, 200, 214
Cheng, Y. 114
China 34

- chinese remainder theorem [33](#), 33–34, 62, 155
- chip 130–132, 135–137
- Clay Mathematics Institute 5, 82, 200
- cluster 129–134, 136–138
- coarse-grained integer 3, [85](#), 85–127
- Cocks, Clifford C. 141, 200
- cofactorization . . . ix, 1, 3, [54](#), 129–131, 134, 196
- Cohen, Henri 64, 71, 197, 200
- complex
- analysis 13
 - coloring [13](#), 14, 15
 - number 14, 21
 - plot [13](#), 14, 15
 - root 53–55
 - value 13, 38
 - variable 13
- complexity 27, 56
- compositeness test *see* primality test
- COPACOBANA 1, 130, 136, 137
- cosec xi
- Cot, N. 208
- Cox, Mark J. 156, 189, 200
- Crandall, Richard 5, 23, 39, 41, 42, 45, 50, 55, 57, 62, 67, 141, 200
- critical
- line 18, [18](#), 19
 - strip 17, [17](#), 18
- cryptanalysis ix, 60, 81
- crypto system
- Okamoto-Uchiyama 4, 171
 - RSA 82–83, 87, 141, 154
- cryptography . . . ix, 24, 37, 60, 81–84, 172, 196
- Cyberoam, Hiren Joshi 190, 208
- Dakar 141, 177
- Damgård, Ivan 2, 142, 181, 183, 199
- de Bessy, Bernard Frénicle (ca. 1605–1675) *see* Bessy
- de Bruijn, Nicolaas Govert *see* Bruijn
- de Fermat, Pierre (1601/1607/1608–1665) *see* Fermat
- de Lagrange, Joseph Louis (1736–1813) *see* Lagrange
- de Leeuw, Jacco *see* Leeuw
- de Moivre, Abraham (1667–1754) *see* Moivre
- de Raadt, Theo *see* Raadt
- de Weger, Benne M. M. *see* Weger
- Decker, Andreas 2, 87, 142, 146, 156, 157, 160, 165, 200
- decryption 81, 82
- Demichel, Patrick 18, 209
- density *see* distribution, cumulative
- derivative *see* distribution
- Detrey, Jérémie xi
- Deuring, Max 61, 200
- Diamond, H. G. 202
- Dickman, Karl (ca. 1862–1940) 24, 25, 200, 214
- ϱ -function 24, [24](#), 25
 - rho function 24
- Diffie-Hellman problem *see* problem, Diffie-Hellman
- Diffie, Whitfield 81, 82, 200
- Διόφαντος ὁ Ἀλεξανδρεύς (ca. 200–284) 58, 214
- Arithmetica 58
- Dirac delta 96, [96](#), 97
- Dirichlet, Johann Peter Gustav Lejeune (1805–1859) 11, 20–22, 155, 200, 214
- L -function 20, 21, [21](#)
 - character [20](#), 21
 - theorem [11](#), 155
- discrete logarithm 56, 57, 82
- problem 56, [56](#), 82
- Disquisitiones Arithmeticae 36, 37
- distribution 179, 180, 182–186
- close to uniform [184](#), 187
 - cumulative 179–182
 - output 182, 190
 - uniform 183, 184
- division with remainder 51
- Doche, Christophe 71, 197
- Duisburg 1
- Durfee, Glenn 172, 199
- Dusart, Pierre 10, 11, 88, 116, 127, 201
- dynamic programming 132, 137
- Echizen, Isao 204
- Edwards, Harold Mortimer, Jr. 2, 3, 5, 15–18, 57, 69–73, 76, 77, 79, 201
- form *see* elliptic curve, Edwards form
- efficiency 4, 185–186
- elementary function 24
- elliptic curve 2, 3, 32, 51, 56, 57, [57](#), 58–67
- addition formula 59, 63, 70, 71, 74, 75
 - addition law 59, [59](#), 60, 62, 63, 65, 66, 72–77
 - affine coordinates 77, 79
 - affine point 62, [62](#)
 - arithmetic 57–60, 62–64, 69–79
 - doubling formula 59, 63, 70–74
 - Edwards form . . . 3, 69–72, [72](#), 73, 76, 77, 79
 - endomorphism ring 62
 - method *see* factorization algorithm, Elliptic Curve Method
 - order 60–62
 - parametrization 69–72, 79
 - point at infinity 59, [59](#), 60, 62, 63
 - point counting 60–62
 - projective coordinates [62](#), 63, 64, 72, 74, 76, 79
 - projective point 62
 - scalar multiplication 63, 71, 79
 - tripling formula 71, 72, 75, 79
 - Weierstraß form 60, 73

- Weierstraß form 59, 59, 70
elliptic pseudocurve 65, 66
Ellis, James Henry 141, 201
Encke, Johann Franz (1791–1865) 8, 215
encryption 81, 82
Engelschall, Ralf 156, 189, 200
entropy 4, 142, 143, 146, 181–186
 input 185, 185
 output 4, 182, 184–190
Ἐρατοσθένης ὁ Κυρηναῖος 7, 215
Erdős, Paul 25, 40, 52, 199, 201
Essen 1
Euclid of Alexandria *see* Εὐκλείδης ὁ
 Ἀλεξάνδρεια (365–300 BC)
Εὐκλείδης ὁ Ἀλεξάνδρεια (365–300 BC) 6, 7,
 27–31, 33, 65, 66, 82, 215
 algorithm 27, 28–31, 33
 Elements 6, 7, 27
 extended algorithm 31, 31, 65, 66, 82
Euler, Leonhard Paul (1707–1783) 7, 8,
 11–13, 16, 17, 19–21, 32, 35, 36, 43, 44, 46,
 82, 201, 215
 φ -function 11, 12, 20, 44, 82
 criterion 35, 35, 36, 43
 probable prime ... *see* probable prime, Euler
 product formula 12, 12, 13, 17, 19, 21
 pseudoprime *see* pseudoprime, Euler
 expected
 runtime *see* runtime, expected
 value 48
Explicit-Formulas Database 64, 70, 71,
 73–75, 79
exponent vector 52, 52, 53
exponentiation 31–33
 fast algorithm 32, 33, 39, 41, 44, 46, 82
factorial function 13
 approximation *see* Stirling’s formula
factorization algorithm 6, 22, 27, 51–67
 Elliptic Curve Method ... 1, 2, 57, 65, 64–67,
 83, 85, 129–131, 134, 135, 137, 139, 196
 General Number Field Sieve ix, 1, 3,
 54–55, 83, 85, 87, 125, 127, 129, 130, 133,
 134, 196
 Quadratic Sieve 51–52, 55, 56, 83
 Special Number Field Sieve 52–54
 trial division 7, 83
factorization problem ... ix, 6, 6, 7, 27, 51, 56,
 82, 84, 167
Farashahi, Reza Rezaeian ... 3, 70, 71, 74, 199
de Fermat, Pierre (1601/1607/1608–1665)
 35, 40–45, 47, 50, 51, 58, 201, 215
 little theorem 35, 35, 40
 probable prime *see* probable prime,
 Fermat
 pseudoprime *see* pseudoprime, Fermat
 test *see* primality test, Fermat
Fibonacci, Leonardo (ca. 1170–1250) ... 28–30,
 215
 number 28, 28, 29, 30
field extension 33, 61
finite field 32, 56, 60, 61, 82
FIPS 186-3 142, 156, 186, 188, 193
M. M. Артюхов 45, 202
Ford, Kevin 9, 88, 114, 202
FPGA 130, 134, 135
Franke, Jens 1, 2, 129, 202
Free Software Foundation 191
Free Software Foundation 156, 191, 202
 on Freenode, Dr{Who} 190, 208
Frey, Gerhard 64, 71, 197, 200
Friedl, Markus 189, 208
Frobenius, Ferdinand Georg (1849–1917) ... 62,
 216
 endomorphism 62
fundamental problem of arithmetic
 *see* factorization problem
fundamental theorem of arithmetic ... 5, 6, 7,
 12
Galbraith, Steven D. 3, 70, 71, 74, 82, 83,
 199, 202
Galgoci, Matthew 190, 208
gamma function 14, 14
 von zur Gathen, Joachim ... xi, 134, 142, 202
Gaudry, Pierrick 3, 70, 71, 202
Gauß, Johann Carl Friedrich (1777–1855)
 8–10, 18, 36, 37, 202, 216
General Number Field Sieve
 *see* factorization algorithm, General
 Number Field Sieve
generalized RSA integer 171–175, 195
generator 181–186
Gerhard, Jürgen 142, 202
Gnu Classpath 191
Gnu Crypto 156, 183, 191–193
GnuPG 156, 178, 190–191, 193
golden
 ratio 28, 28, 29, 30
 spiral 30
Goldreich, Oded 167, 203
Goubin, Louis 203–205
Gourdon, Xavier 18, 209
Goy, Denise xi
grained
 integer 2, 26, 81, 85, 84–127, 139,
 141–169, 171–175
grained integer 195
Granville, Andrew 2, 23, 42, 86, 197, 203
greatest common divisor 27, 27–31, 43
greedy 133
Güneysu, Tim 1, 130, 134, 202, 203
Guyomarch, Rémi 156, 190, 210
Hadamard, Jacques Salomon (1865–1963) ... 9,
 203, 216
Hashimoto, Yasufumi 156, 203
Hasse, Helmut (1898–1979) ... 61, 66, 203, 216

- Hellman, Martin Edward 81, 82, 200
Henry, Charles 201
Henson, Stephen 156, 189, 200
Hernaeus, Niklas 156, 190, 210
Hielscher, Martin xi
Hilbert, David (1862–1943) 5, 203, 216
Hildebrand, Adolf J. 23, 25, 202, 203
homogeneous
 form of an elliptic curve 62
 polynomial 54
Howgrave-Graham, Nick 172, 199
Huang, Ming-Deh 197
hypercube 97
hyperplane 97
- IEEE 1363-2000 142, 156, 162, 183,
 186–187, 192, 193
IEEE working group 142, 156, 186, 187, 204
index calculus 56–57, 82
Ingemarsson, I. 208
input entropy *see* entropy, input
integer
 coarse-grained *see* coarse-grained integer
 generalized RSA *see* generalized RSA
 integer
 grained *see* grained, integer
 rough *see* rough, integer
 RSA *see* RSA, integer
 smooth *see* smooth, integer
International Organization for Standards
 142, 186, 204
inverse transform sampling 180, 181
IPsec 190
ISO 18033-2 142, 186, 192, 193
IWSEC 69
- Jacobi, Carl Gustav Jacob (1804–1851)
 35–37, 43–45, 204, 216
 symbol 35–37, 43–45
Jacobson, Michael J. 206
Jager, Hendrik 207
Japan 69
Jiushao, Qin (1202–1261) 34, 204, 217
Jonsson, Jakob 142, 186, 204
Joye, Marc 2, 69, 76, 142, 183, 198, 199, 204
Justus, Benjamin 69, 72, 75, 77, 78, 159, 160,
 204
- Kaliski, Burt 142, 186, 204
Kargl, Anton 134, 202
Kayal, Neeraj 49, 50, 197
key
 private 81–83
 public 81–83
 server 81
key exchange 81, 82
Kiel 1
Killmann, Wolfgang 183, 204
Kleinjung, Thorsten 1, 2, 55, 129, 202, 204
- Knuth, Donald Ervin 30, 34, 180, 204
Kobe 69
Koblitz, Neal 58, 204
Koç, Çemal K. 207
von Koch, Niels Fabian Helge 11, 20, 88,
 204, 217
Koch, Werner 156, 190, 210
Königsberger, Konrad 14, 205
Korselt, Alwin Reinhold (1864–1947) 42, 44,
 205, 217
 criterion 42, 42, 44
Kraitchik, Maurice 56, 205
Krawczyk, H. 210
Kulsha, Andrey V. 88, 205
Kumar, Sandeep 130, 205
Kunihiro, Noboru 204
Kurosawa, Kaoru 198
- Lafon, Mathieu 190, 208
de Lagrange, Joseph Louis (1736–1813) 31,
 32, 205, 217
 theorem 31
Lamé, Gabriel 29, 205
Landau, Edmund Georg Hermann (1877–1938)
 22, 23, 205, 217
Lange, Tanja 64, 69–72, 197, 198
Laurie, Ben 156, 189, 200
law of quadratic reciprocity *see* quadratic,
 reciprocity
de Leeuw, Jacco 190, 208
Legendre, Adrien-Marie (1752–1833) 9,
 34–36, 55, 61, 205, 217
 symbol 34, 36, 55, 61
Lehmer, Derrick Henry 45, 199
Lenstra, Arjen Klaas 55, 199, 205, 208
Lenstra, Hendrik Willem, Jr. 54, 55, 57, 64,
 199, 205, 208
 Elliptic Curve Method *see* factorization
 algorithm, Elliptic Curve Method
lexicographically less 29
Li, Xian-Jin 40, 205
Liber abbaci 28
linear algebra 56
Littlewood, John Edensor 10, 19, 205
Loebenberger, Daniel 69, 72, 75, 77, 78, 85,
 129, 134, 141, 155, 157–160, 162–164, 167,
 174, 177, 202, 204, 206
logarithmic integral 8, 8, 9, 10, 94
Lu, Hsiao-Feng 198
Lubicz, David 3, 70, 71, 202
Lucas, François Édouard Anatole (1842–1891)
 7, 206, 218
- Matsui, Mitsuru 203–205
Maurer, Ueli M. 145, 206
May, Alexander 172, 198
McCullough, David 190, 208
Mellin, Robert Hjalmar (1854–1933) 19, 218
Menezes, Alfred J. 142, 206

- Mersenne, Marin 51, 218
Mertens, Franz (1840–1927) 19, 206, 218
Meyn, Helmut xi
millenium problem ... *see* problem, millennium
Miller test *see* primality test, Miller
Miller, Gary Lee 22, 45, 47, 206
Miller-Rabin
 primality test *see* primality test,
 Miller-Rabin
Miller, Victor Saul 58, 206
Möbius, August Ferdinand (1790–1868) 17,
 19, 206, 218
 function 17, 19
de Moivre, Abraham (1667–1754) 28, 218
Monier, Louis 47, 206
Montgomery, Peter Lawrence 3, 63, 64,
 69–71, 207
Moree, Pieter 2, 87, 142, 146, 156, 157, 159,
 160, 165, 200
Morrison, Michael A. 52, 207
multiplication 33, 39, 63, 69–71, 73–76, 78,
 79

Naccache, David 199, 207
NESSIE 142, 156, 189, 193
NESSIE working group 142, 156, 186, 189,
 207
von Neumann, John 178, 207
Newton, Sir Isaac (1643–1727 (greg.)) 37,
 38, 207, 219
Nguyen, Kim 71, 197
NIST 142, 156, 186, 188, 207
Nitaj, Abderrahmane 206
notion 143, 143, 144–152, 154, 156–169,
 178–181, 183, 185, 187–191, 196
 algorithmically inspired 160–162
 antisymmetric 144, 181, 187, 190
 balanced ... 144, 145, 146, 150, 151, 157, 159,
 161, 163, 164, 167, 173, 178, 180
 fixed bound 158–160, 163–165, 189, 190
 generalized 171–175
 graph-bounded 148, 148, 149, 150, 178,
 185
 monotone 148, 148, 149, 150, 152, 154,
 157, 159, 161, 163, 164, 167, 180, 183
 number theoretically inspired 156–158
 piece-wise monotone 164, 167
 piece-wise monotone 148
 symmetric 144, 181, 187, 190
number theory ix, 3, 5–26
Nüsken, Michael xi, 85, 99–101, 103,
 116–119, 141, 155, 157–160, 162–164, 167,
 174, 177, 206, 207
Nyberg, Kaisa Tellervo 207

Oesterlé, Joseph 20, 207
Okamoto, Tatsuaki 171, 207
Okeya, Katsuyuki 76, 207
Oliveira Coelho, Cláudia xi

Oliveira e Silva, Tomás *see* Silva
on Freenode, Dr{Who} *see* Freenode
van Oorschot, Paul C. 142, 206
OpenSSH 189
OpenSSL 156, 189–190, 192, 193
Openswan 190, 193
optimization 133, 134, 136, 139
oracle 185
order
 element 32, 50, 57
 elliptic curve *see* elliptic curve, order
 group 32, 56, 60, 62, 66, 67
Oswald, Elisabeth 204
output distribution 142
output entropy *see* entropy, output

Paar, Christof ... 1, 130, 134, 202, 203, 205, 207
Paillier, Pascal 2, 142, 183, 199, 204
partition 131, 131, 132, 133, 136, 138
 coarsening 131, 131
 refinement 131, 133
Pelzl, Jan 130, 205
perfect power 37–39, 50, 51, 65
perfect power test 39, 39, 50
permutation 120, 121
Peters, Christiane 69, 198
Pfab, Fotini xi
Pfeiffer, Gerd 1, 130, 203, 205
Philipp, W. 202
Pintz, János 207
Pisano, Leonardo 207
PKCS#1 142, 186, 192, 193
Pointcheval, David 206
Pollard, John Michael ... 52, 53, 55–57, 64, 67,
 207, 208
 ($p - 1$)-method 57, 64, 67
 g -method 56
 Special Number Field Sieve
 *see* factorization algorithm, Special
 Number Field Sieve
polynomial
 homogeneous *see* homogeneous,
 polynomial
Pomerance, Carl ... 5, 23, 25, 39, 41, 42, 45, 46,
 50–52, 54, 55, 57, 62, 67, 141, 197, 199, 200,
 208
primality test 7, 7, 21, 27, 37, 39–50
 AKS 50, 49–51, 178
 Fermat 40, 40–42, 44, 45, 47
 Miller 47, 47, 178
 Miller-Rabin 46, 45–49, 178
 randomized 47, 48
 Solovay-Strassen 44, 43–45, 177
 strong *see* primality test, Miller-Rabin
prime 5–26
 counting function 8, 10, 17, 18
 distribution 5–12
 factorization 5, 6, 36, 37, 42, 55
 generator 48, 49

- number theorem . . . 8, 9, 9, 10, 19, 20, 87, 90, 125, 127, 160
- sum approximation 147, 150
- test *see* primality test
- private key *see* key, private
- probable prime 40, 40
 - Euler 43, 43, 44
 - Fermat 40, 40, 43
 - strong 45, 46–49
- problem
 - Diffie-Hellman 82
 - discrete logarithm . . . *see* discrete logarithm, problem
 - factorization *see* factorization problem
 - millennium 5, 82
 - RSA 83
- Provos, Niels 189, 208
- pseudoprime 40
 - Euler 43, 43, 46
 - Fermat 40, 40, 41–43
 - strong 46, 46
- public key *see* key, public
- Putzka, Jens 129, 134, 202, 206
- quadratic
 - reciprocity 36, 36, 37
 - residue 34, 34–40
 - sieve *see* factorization algorithm, Quadratic Sieve, 52
- de Raadt, Theo 189, 208
- Rabin, Michael Oser 45, 47, 208
- Raekow, Yona xi
- Ramaswami, V. 24, 208
- random compositeness test *see* primality test, randomized
- Redelmeier, D. Hugh 190, 208
- Richardson, Michael 190, 208
- Riemann, Georg Friedrich Bernhard (1826–1866) 5, 11, 13–22, 25, 47, 70, 87, 88, 91, 114, 147, 150, 157, 159, 160, 162, 163, 172–175, 208, 219
 - ϑ -function 3, 70
 - extended hypothesis 20, 21, 21, 22, 47
 - hypothesis 5, 11, 13, 18, 18–22, 25, 87, 88, 91, 114, 147, 150, 157, 159, 160, 162, 163, 172–175
 - prime count approximation 18
- Rijmen, Vincent 206
- ring homomorphism 53
- Rivest, Ronald Linn 82, 141, 208, 209
- Robbins, Herbert 105, 209
- Rohatgi, Pankaj 204
- Rosser, John Barkley 88, 90, 116, 209
- Roth, Michael 156, 190, 210
- rough
 - integer 2, 25, 25–26, 83–87
- RSA
 - crypto system *see* crypto system, RSA
 - fast variants 4, 171
 - foundation 142, 156, 186, 189
 - integer 2, 87, 141–169, 171, 177–180, 183, 185, 186, 189, 191, 195
 - notion *see* notion
 - OAEP 186–187, 191, 193
 - problem *see* problem, RSA
 - RSA Laboratories 129, 142, 156, 186, 209
 - runtime 1, 28, 29, 31, 33, 38, 39, 41, 44, 47, 48, 50, 57, 67, 131–135, 137–139
 - average 30
 - expected 48
 - heuristic 52, 55, 67
- SAC 129
- Safavi-Naini, Rei 206
- sage xi
- Sakurai, Kouichi 76, 207
- Saouter, Yannick 18, 209
- Sasaki, Ryoichi 204
- Saxena, Nitin 49, 50, 197
- Schimmler, Manfred 1, 130, 203, 205
- Schindler, Werner 182, 183, 204, 209
- Schneier, Bruce 142, 209
- Schoenfeld, Lowell (1920–2002) 11, 20, 88, 90, 116, 209, 219
- Schoof, René 61, 62, 209
- secret 81, 82
 - common 81, 82
 - pre-shared 81
- security ix, 82–84
- Selfridge, John Lewis (1927–2010) 45, 46, 199, 208, 219
- semi-prime 22–23
- Senegal 141, 177
- Shallit, Jeffrey 49, 197
- Shamir, Adi 82, 141, 208, 209
- Shanks, Daniel 56, 209
- Shoup, Victor 56, 209
- Siegel, Carl Ludwig (1896–1981) 11, 219
- Siemens AG 1
- sieve
 - general number field *see* factorization algorithm, General Number Field Sieve
 - quadratic *see* factorization algorithm, Quadratic Sieve
 - segmented 88
- signature 81
- Oliveira e Silva, Tomás 88, 210
- Silverman, Joseph Hillel 58, 210
- Skala, Matthew 156, 190, 210
- slope 59, 65–67
- smooth
 - curve 57, 58
 - element of a number field 54
 - function 91, 97
 - group order 57, 66
 - integer 2, 23, 23–25, 52, 54, 57, 66, 83–87
 - value of a polynomial 54

- Soini, Tuomo 190, 208
 Solovay, Robert Martin 45, 210
 Solovay-Strassen test *see* primality test,
 Solovay-Strassen
 Song, Dug 189, 208
 sorting 120, 120, 121, 122
 Springer-Verlag 209
 squaring 63, 69–71, 73–76, 78, 79
 standard 4, 186–192, 196
 statistical analysis 134
 Steffen, Andreas 190, 208
 Stevenhagen, Peter 203
 Stirling's formula 105, 106
 Strassen, Volker 45, 210
 strong
 primality test *see* primality test,
 Miller-Rabin
 prime 57, 169, 195
 probable prime *see* probable prime,
 strong
 pseudoprime *see* pseudoprime, strong
 Sturm, Jacques Charles François 108, 210
 subgroup 32, 44
- Takagi, Tsuyoshi 171, 210
 Tannery, Paul 201
 Tenenbaum, Gérald 23, 203
 trial division *see* factorization algorithm,
 trial division
- Trmac, Miloslav 190, 208
 Trojanowski, Bart 190, 208
 type 120, 121
- Uchiyama, Shigenori 171, 207
- de la Vallée Poussin, Charles-Jean Étienne
 Gustave Nicolas, Baron (1866–1962) 9,
 210, 219
- van Oorschot, Paul C. *see* Oorschot
 Vanstone, Scott A. 142, 206
 Vaudenay, Serge 198
 Vercauteren, Frederik 71, 197
 Virtex4 XC4VSX35 130, 134, 135
 von Koch, Niels Fabian Helge *see* Koch
 von Neumann, John *see* Neumann
- Wagstaff, Samuel Standfield, Jr. 46, 208
 Walfisz, Arnold (1892–1962) 9, 11, 210, 219
 Washington, Lawrence Clinton 61, 65, 66,
 210
- de Weger, Benne M. M. 2, 156, 210
 Weierstraß, Karl Theodor Wilhelm
 (1815–1897) 58–61, 70, 71, 73, 220
 form *see* elliptic curve, Weierstraß form
 parameters 61
 Weistraß, Karl 58, 210, 211
 Wiener, Matthew 199
 Wiles, Andrew 58, 211
 Wilf, Herbert Saul 122, 211
- Williams, Hugh C. 206
 witness 47, 47, 48
 at w.ods.org, willy 190, 208
 Wohlmacher, Petra 142, 145, 186, 192, 211
 Wouters, Paul 190, 208
- Xu, Herbert 190, 208
- Yamawaki, Shingo 190, 208
- zeta constants 16, 16, 17
 zeta function 12, 15, 13–21
 trivial zeros 15, 16, 17
 Zi, Sun 34, 211, 220
 Ziegler, Konstantin xi