General purpose integer factoring

Arjen K. Lenstra

EPFL IC LACAL, Station 14, CH-1015 Lausanne, Switzerland

Abstract

This chapter describes the developments since 1970 in general purpose integer factoring and highlights the contributions of Peter L. Montgomery.

This article appeared as Chapter 5 of the book *Topics in Computational Number Theory inspired by Peter L. Montgomery*, edited by Joppe W. Bos and Arjen K. Lenstra and published by Cambridge University Press. See `www.cambridge.org/9781107109353`. There are cross-references to two chapters of the same book: Chapter 6 *Polynomial selection for the number field sieve* by Thorsten Kleinjung and Chapter 7 *The block Lanczos algorithm* by Emmanuel Thomé.

Contents

5 General purpose integer factoring *page* 1

5.1 Introduction 1

5.2 General purpose factoring 2

5.2.1 Two-step approach 2

5.2.2 Smoothness and *L*-notation 4

5.2.3 Generic analysis 5

5.2.4 Smoothness testing 6

5.2.5 Finding dependencies 7

5.2.6 Filtering 8

5.2.7 Overall effort 11

5.3 Pre-sieving general purpose factoring 11

5.3.1 Dixon's random squares method 11

5.3.2 Continued fraction method 12

5.4 Linear and quadratic sieve 14

5.4.1 Linear sieve 14

5.4.2 Quadratic sieving: plain 17

5.4.3 Quadratic sieving: fancy 18

5.4.4 Multiple polynomial quadratic sieve 19

5.5 Number field sieve 23

5.5.1 Earlier methods to compute discrete logarithms 24

5.5.2 Special number field sieve 30

5.5.3 General number field sieve 37

5.5.4 Coppersmith's modifications 43

5.6 Provable methods 44

Bibliography 47

Subject index 55

5

General purpose integer factoring

Arjen K. Lenstra

5.1 Introduction

General purpose integer factoring refers to methods for integer factorization that do not take advantage of size-related properties of the unknown factors of the composite to be factored. Methods that do are special purpose methods, such as trial division, John Pollard's rho and $p-1$ methods [70, 69] and variants [94, 7], and Hendrik Lenstra's elliptic curve method [59]. Some of these methods are discussed in other chapters. The subject of this chapter is general purpose integer factoring.

In 1970 a new general purpose integer factoring record was set with the factorization of the seventh Fermat number $F_7 = 2^{2^7} + 1$, a number of 39 decimal digits [65]. It required about 90 minutes of computing time, accumulated over a period of seven weeks on an IBM 360/91 computer at the UCLA Campus Computing Network. In the late 1970s the 78-digit eighth Fermat number was *almost* factored by a general purpose method [83]. But its small, 16-digit factor was discovered in 1980 using Pollard's rho method [11], with general purpose factoring achieving 71 digits only in 1984. It leapt to 87 digits in 1987, a development that could for a large part be attributed to Peter Montgomery [16], requiring on the order of two months of computing time contributed by a modest local network in about a week of wall-clock time. It further jumped ahead to 100 digits in the fall of 1988, the first computation that harvested the Internet's computational resources [57].

Around that same time it became necessary to distinguish two different record-categories [54], a distinction that persists until the present day: records for general composites, such as cryptographic moduli [81], and records for composites with a special form, such as Fermat or Cunningham numbers [28, 12]. The first category record stands at 232 decimal digits, with the 2009 factorization of a 768-bit cryptographic modulus [44]; Montgomery's work was used

in several steps of this calculation. It required about two thousand core years of computation, accumulated over a period of almost three years on a wide variety of computers world-wide. The current record calculation of the second category is the shared factorization of $2^m - 1$ for $m \in \{1007, 1009, 1081, 1111, 1129, 1151, 1153, 1159, 1177, 1193, 1199\}$ which required about five thousand core years over a period of almost five years [22, 45].

In this chapter the various developments are described that contributed to the progress in general purpose integer factoring since 1970. At first this progress heavily relied on new algorithms, but this came mostly to a halt around 1992. Starting in the late 1980s the proliferation of (easily accessible) hardware, the end of which is still not in sight, played an increasingly important role. For a different perspective refer to [77].

Throughout this chapter n indicates the composite to be factored. It is assumed that n is odd and not a prime power. An integer k *divides* an integer m if m is an integer multiple of k; this is denoted by $k|m$.

5.2 General purpose factoring

At this point in time all general purpose factoring algorithms and their running time analyses work in more or less the same way, from a high-level point of view. The common framework is presented in this section.

General idea. Following Maurice Kraitchik's variation of Pierre de Fermat's method [47, 48], all general purpose factoring methods are *congruence of squares methods*. They all construct pairs of integers x, y such that $x^2 \equiv y^2 \bmod n$; from $n|x^2 - y^2 = (x - y)(x + y)$ it follows that $\gcd(x - y, n)$ is a non-trivial factor of n if $x \neq \pm y \bmod n$. Although most congruence of squares methods are entirely, or to some extent, deterministic, it is reasonable to assume (for some methods this can be proved) that the resulting pairs are more or less random, under the condition that $x^2 \equiv y^2 \bmod n$. This implies that if n has r distinct prime factors, then $x \equiv \pm y \bmod n$ with probability $\frac{2}{2^r}$. Thus, in practice a few pairs x, y will suffice to factor n.

5.2.1 Two-step approach

In [65] Michael Morrison and John Brillhart proposed a systematic, two-step approach to construct pairs x, y as above (see also [92]). Let $\omega \in \mathbf{Z}_{>0}$ be a small constant indicating the desired *oversquareness* (cf. below). (Refer to [85, 82] for *squfof*, an entirely different method to construct pairs x, y as above; it

is based on binary quadratic forms and particularly efficient for small values of n.)

Step one: relation collection. In the first step a finite set P of integers is selected depending on n, and a set V of $|P| + \omega$ *relations* is sought, where a relation is a pair (v, e_v) with $v \in \mathbf{Z}$ and $e_v = (e_{v,p})_{p \in P} \in \mathbf{Z}^{|P|}$ such that

$$w = \prod_{p \in P} p^{e_{v,p}} \quad \text{where} \quad w \equiv v^2 \bmod n. \tag{5.1}$$

The central distinguishing feature between the general purpose factoring algorithms described in this chapter is the method by which relations are obtained. For instance, and most elementary, it may simply involve selecting integers v and inspecting if $w \equiv v^2 \bmod n$ allows factorization over P. The set of representatives for the integers w may be explicitly chosen, for instance as $\{0, 1, \ldots, n-1\}$ or as $\{-\lfloor \frac{n}{2} \rfloor, \ldots, -1, 0, 1, \ldots, \lfloor \frac{n}{2} \rfloor\}$, or may be implicit in the way w is generated during the search. The set P is commonly referred to as the *factor base*.

Step two: linear algebra. Since the oversquareness ω is strictly positive, the vectors e_v resulting from the first step are linearly dependent. In the second step linear algebra is used to determine dependencies modulo 2 among the vectors e_v, i.e., non-empty subsets S of V such that all entries of the vector $\sum_{v \in S} e_v = (s_p)_{p \in P}$ are even. Each subset S gives rise to a pair

$$x = (\prod_{v \in S} v) \bmod n, \quad y = (\prod_{p \in P} p^{\frac{s_p}{2}}) \bmod n.$$

With Condition (5.1) it follows that $x^2 \equiv y^2 \bmod n$. At least ω independent subsets S can be determined, which should lead to at least ω independent chances of at least 50% to factor n.

Selection of *P*. Morrison and Brillhart chose P as the set consisting of the element -1 along with the primes up to some bound B. All noteworthy general purpose factoring methods since then follow the same two-step approach and essentially make the same choice for P. The choice of B involves a trade-off: with a small B only a few relations are needed, but they may be hard to find, whereas for larger B more relations are required that should, however, be easier to find. The linear algebra effort does not involve a trade-off: it increases some way or another as B increases. When the overall effort is minimized, the optimal choice for B may depend on either of the two separate efforts or both.

Morrison and Brillhart relied on experience to decide on a value for B that would minimize the relation collection effort for the n-value at hand, knowing from their experiments that the linear algebra step would be relatively easy as long as enough memory is available. Richard Schroeppel was the first, during

the mid 1970s, to analyse the trade-off and to derive an expression for the optimal asymptotic effort needed for the relation collection step [81, Section IX.A] – disregarding the linear algebra step as he had the same experience that it would involve only a relatively minor effort. Schroeppel's analysis required insight in the probability to find a relation, for which he relied on the result presented below, even though back then it had not been fully proved yet. The first to fully (though still heuristically, cf. Section 5.2.3) analyse the known general purpose integer factoring algorithms was Carl Pomerance in [74].

5.2.2 Smoothness and L-notation

An integer is called B-smooth if all its prime factors are at most B. Smoothness probabilities and the resulting asymptotic estimates are commonly expressed using the generalized L-notation. The L-notation was introduced by Pomerance in [74, Section 2] to (citing Pomerance) "streamline many arguments and to have the important magnitudes stand out", reasons that are still valid today. Following the generalization from [53, (3.16)], denote by $L_x[r,\psi]$ any function of x that is

$$e^{(\psi+o(1))(\log x)^r(\log\log x)^{1-r}}, \text{ for } x \to \infty,$$

where $r,\psi \in \mathbf{R}$ and $0 \leq r \leq 1$ and where logarithms are natural. For fixed $r,s,\psi,\beta \in \mathbf{R}_{>0}$ with $s < r \leq 1$, it follows from results by Nicolaas De Bruijn [30] and E. Rodney Canfield, Paul Erdös and Pomerance [15] (and citing [53]) that a random positive integer at most $L_x[r,\psi]$ is $L_x[s,\beta]$-smooth with probability $L_x[r-s, -\frac{\psi(r-s)}{\beta}]$, for $x \to \infty$.

With $\psi > 0$, the expression $L_x[0,\psi]$ is *polynomial* in $\log x$, whereas $L_x[1,\psi]$ is *exponential* in $\log x$; for $0 < r < 1$, the expression $L_x[r,\psi]$ is called *subexponential* in $\log x$. To illustrate the quote from Pomerance, if $r,s,\psi,\beta \in \mathbf{R}$ are fixed with $0 \leq s < r \leq 1$ and $\psi > 0$, then $L_x[r,\psi]L_x[s,\beta] = L_x[r,\psi]$: the factor $L_x[s,\beta]$ disappears in the $o(1)$ in $L_x[r,\psi]$. This includes the case $s = 0$ where $L_x[0,\beta] = (\log x)^{\beta+o(1)}$: factors that are fixed rational polynomials in $\log x$ disappear in $L_x[r,\psi]$ if $r,\psi > 0$. Thus, if $B = L_x[r,\psi]$ for $r,\psi > 0$, then the number $\pi(B)$ of primes at most B equals B. Note also that $L_x[r,\psi]L_x[r,\beta] = L_x[r,\psi+\beta]$ and $L_x[r,\psi] + L_x[r,\beta] = L_x[r,\max(\psi,\beta)]$.

Below all efforts expressed in terms of L_n are expected and asymptotic for $n \to \infty$ and, unless noted otherwise, heuristic. In the remainder of this chapter L is used for L_n.

5.2.3 Generic analysis

Let $S(x, B)$ denote the average effort to inspect a random positive integer at most x for B-smoothness, and let $M(m)$ denote the effort to find dependencies modulo 2 among the columns of an integer $m \times (m + \omega)$-matrix. Let $L[s, \beta]$ for $s, \beta \in \mathbf{R}_{>0}$ be the upper bound for the primes in P. For each general purpose factoring method that has been published, there are $r, \psi \in \mathbf{R}_{>0}$ such that the absolute value of each number that is inspected for smoothness (and that leads to a relation as in (5.1) on page 3 if it is $L[s, \beta]$-smooth) is bounded above by $L[r, \psi]$: for the number field sieve $r = \frac{2}{3}$ and for all earlier methods $r = 1$, as set forth in the sections below. Assuming that these absolute values behave as random positive integers at most $L[r, \psi]$, the overall factoring effort of almost all general purpose factoring methods can be expressed as

$$\underbrace{(\pi(L[s,\beta]) + \omega)}_{\substack{\text{number of}\\ \text{relations to}\\ \text{be collected}}} \cdot \underbrace{L[r - s, \tfrac{\psi(r-s)}{\beta}]}_{\substack{\text{inverse of}\\ \text{smoothness}\\ \text{probability}}} \cdot \underbrace{S(L[r,\psi], L[s,\beta])}_{\substack{\text{average effort to}\\ \text{inspect an integer}\\ \text{for smoothness}}} + \underbrace{M(\pi(L[s,\beta]))}_{\substack{\text{effort for}\\ \text{linear}\\ \text{algebra}}}; \quad (5.2)$$

see Section 5.4.1 on page 14 for the exception. For one of the general purpose methods presented below it can be proved that the numbers to be tested for smoothness behave, with respect to smoothness properties, as random positive integers at most $L[r, \psi]$. For that method, Expression (5.2) is the expected asymptotic factoring effort, for $n \to \infty$. For all other methods the smoothness assumption is a heuristic assumption that has, so far, been supported by empirical evidence. For those methods, Expression (5.2) is the heuristic expected asymptotic factoring effort, for $n \to \infty$.

Optimal choices for s, ψ and β depend on the general purpose factoring method used and on the smoothness testing and linear algebra methods used, and are derived in the sections below. But a few general observations can be made that simplify Expression (5.2). It follows from $r > 0$ and the factor $L[r - s, \frac{\psi(r-s)}{\beta}]$ in the first term of Expression (5.2) that the optimal s must be strictly positive. With $\beta > 0$ this implies that $\pi(L[s,\beta]) + \omega = L[s,\beta]$ and, again with Expression (5.2), that $s = \frac{r}{2}$ is optimal. As a result, the overall factoring effort becomes

$$L[\tfrac{r}{2}, \beta + \tfrac{\psi r}{2\beta}]\, S(L[r,\psi], L[\tfrac{r}{2},\beta]) \;+\; M(L[\tfrac{r}{2},\beta]). \quad (5.3)$$

Further optimization depends on the general purpose factoring method under consideration, but also on the rules of the game that one decides to play: for both $S(x, B)$ and $M(m)$ either the historically correct methods (i.e., methods available at the time the general purpose factoring method was developed) or the current best methods (i.e., as may have been developed later and thus pos-

sibly anachronistic) can be used, which may give rise to different outcomes. This is further discussed below.

5.2.4 Smoothness testing

Trial division consists of testing a non-zero integer w for B-smoothness by testing if p divides w for all primes $p \leq B$, while replacing w by $\frac{w}{p}$ and repeating the test for p if indeed $p|w$. If $w = 1$ at the end of this process, then the original w is B-smooth and may lead to a relation as in (5.1) on page 3. If $1 < w < B^2$ after trial division, then the original w is B-smooth except for one*largeprime*, i.e., a prime factor larger than B but less than B^2; such w-values may lead to a *large prime relation*, as further discussed in the sections below. The trial division effort for w is $O(B \log w)$ implying that $S(L[r,\psi], L[\frac{r}{2},\beta])$ would become $L[\frac{r}{2},\beta]$. Thus, if trial division is used as smoothness test, the first term of Expression (5.3) becomes $L[\frac{r}{2},\beta+\frac{\psi r}{2\beta}]L[\frac{r}{2},\beta] = L[\frac{r}{2}, 2\beta+\frac{\psi r}{2\beta}]$. Similarly, using Pollard's rho method [70] would lead to $S(L[r,\psi], L[\frac{r}{2},\beta]) = L[\frac{r}{2}, \frac{\beta}{2}]$.

The elliptic curve method of factorization may, heuristically, be expected to find a factor at most B of a positive integer w at effort $O((\log w)^2 L_B[\frac{1}{2},\sqrt{2}])$ [59] (see also [53, 4.3]). It follows that $S(L[r,\psi], L[\frac{r}{2},\beta])$ would become $L[\frac{r}{2}, 0]$. This implies that, if the elliptic curve method is used as smoothness test, the first term of Expression (5.3) becomes $L[\frac{r}{2},\beta + \frac{\psi r}{2\beta}]$. As shown by Pomerance in [75], the heuristic arguments can be removed from the analysis, resulting in a slightly modified elliptic curve-based smoothness test that works with high probability (see also Section 5.6 on page 44). Although with elliptic curve-based smoothness testing the $S(L[r,\psi], L[\frac{r}{2},\beta])$-contribution to Expression (5.3) conveniently vanishes in the $o(1)$, in practice its contribution would be considerable, in particular compared to *sieving* as discussed in the next paragraphs – if sieving can be applied.

Sieving amortizes the smoothness testing effort over all values that have to be tested for smoothness, and achieves the same effect on Expression (5.3) as elliptic curve-based smoothness testing. In practice, sieving is much preferred over using elliptic curves, but compared to the latter it has the disadvantage that it can not be applied in all circumstances: the set of values to be tested must be the set of values of a polynomial evaluated over sufficiently many integers in an arithmetic progression (such as a sufficiently large interval of consecutive integers).

Let I be an interval of consecutive integers such that the length $|I|$ of I is at least $L[\frac{r}{2},\beta]$ and let d be a small positive integer ($d \leq 2$ until Section 5.5 on page 23). Assume that there exists a degree d polynomial $f(X) \in \mathbf{Z}[X]$ such that $\{f(i) : i \in I\}$ is the set of values that has to be tested for $L[\frac{r}{2},\beta]$-smoothness

(more general arithmetic progressions are treated in a similar fashion). Sieving exploits the fact that if p divides $f(z)$ for some $z \in \mathbf{Z}$, then p divides $f(z + kp)$ for all $k \in \mathbf{Z}$. Thus, once all roots of f modulo p in $\{0, 1, \ldots, p-1\}$ have been determined, all roots of f modulo p in I are found at additional effort at most $\frac{d|I|}{p}$. The latter roots correspond to the subset of polynomial-values that are divisible by p. Root finding modulo a prime p takes (probabilistic) effort $(\log p)^t$ for some small constant t. Using $|I| \geq L[\frac{r}{2}, \beta]$, the overall effort of finding all prime divisors at most $L[\frac{r}{2}, \beta]$ of the values in $\{f(i) : i \in I\}$ to be tested for $L[\frac{r}{2}, \beta]$-smoothness is

$$\sum_{p \leq L[\frac{r}{2}, \beta]} ((\log p)^t + \tfrac{d|I|}{p}) = |I| \tag{5.4}$$

(where it is used that $\sum_{p \leq L[\frac{r}{2}, \beta]} \frac{1}{p} \approx \log\log(L[\frac{r}{2}, \beta])$). In the context of Expression (5.3) on page 5, the interval length $|I|$ and thus the overall effort over all $|I|$ values to be tested for smoothness would be $L[\frac{r}{2}, \beta + \frac{\psi r}{2\beta}]$. As a result, the average smoothness testing effort $S(L[r, \psi], L[\frac{r}{2}, \beta])$ becomes $L[\frac{r}{2}, 0]$, namely the quotient of the number $|I|$ of values to be tested for smoothness and the overall smoothness testing effort $|I|$, so that the first term of Expression (5.3) simplifies to $L[\frac{r}{2}, \beta + \frac{\psi r}{2\beta}]$.

In practice, sieving typically consists of adding a rough approximation of $\log_b p$, for some base $b \in \mathbf{R}_{>0}$, to all sieve locations $z + kp \in I$ for $k \in \mathbf{Z}$, after a root z of f modulo p has been computed. After doing this for all (prime, root) pairs, the locations $\ell \in I$ at which a total value has been accumulated that is close to a rough estimate of $\log_b f(\ell)$, are inspected more closely by attempting to factor $f(\ell)$ over P. Large prime relations can also easily be recognized when more locations (with smaller but still sufficiently large values) are inspected. The base b is chosen in such a way that a single byte per sieve location suffices to represent an approximation to $\log_b f(\ell)$ for all $\ell \in I$. Montgomery proposed to choose the initial values of the sieve locations so that a final non-negative value indicates that the location needs to be inspected for actual smoothness, because a four- or eight-byte mask can then be used to check four or eight sieve locations at a time for non-negativity. He also proposed to put a non-negative value right after the last location to be inspected, so that it suffices to check the termination condition for $\ell \in I$ only at locations containing non-negative values.

5.2.5 Finding dependencies

For all general purpose factoring methods the matrices are *sparse*, i.e., the number of non-zero entries per column is at most of order $\log(L[r, \psi])$. Regular

Gaussian elimination hardly profits from the sparseness: usually, the sparsity-advantage is no longer noticeable after about a fifth of the pivots have been processed. As a result, with regular Gaussian elimination the term $M(L[\frac{r}{2}, \beta])$ in Expression (5.3) on page 5 becomes $L[\frac{r}{2}, 3\beta]$. If pivots are selected using *structured Gaussian elimination* [49, 78], the sparse original $m \times (m + \omega)$-matrix $\mathcal{M}$ over $\mathbf{Z}/2\mathbf{Z}$ can often easily be reduced to a dense $m' \times (m' + \omega)$-matrix $\mathcal{M}'$ over $\mathbf{Z}/2\mathbf{Z}$ with $m' \approx \frac{m}{3}$, in such a way that dependencies among the columns of $\mathcal{M}'$ (for instance determined using regular Gaussian elimination) lead to dependencies among the columns of $\mathcal{M}$. Although this combined approach does not change the term $M(L[\frac{r}{2}, \beta]) = L[\frac{r}{2}, 3\beta]$ in Expression (5.3), it was of great practical importance until the mid 1990s: compared to regular Gaussian elimination, it not just reduced the effort by a factor of approximately 3^3, it also reduced the storage requirement of m^2 bits by a factor of about 3^2. Volker Strassen's method [88] (applied to $\mathcal{M}$ or to $\mathcal{M}'$) reduces $M(L[\frac{r}{2}, \beta])$ in Expression (5.3) to $L[\frac{r}{2}, (\log_2 7)\beta]$; with the latest variants of the method by Don Coppersmith and Shmuel Winograd [25, 50, 5] it would become about $L[\frac{r}{2}, 2.373\beta]$.

Block versions of methods by Cornelius Lanczos [24, 67, 64] or Douglas Wiedemann [93, 23] (see [53, 2.19] for a high-level description) profit much more effectively from the sparseness of $\mathcal{M}$, because for both methods the effort is dominated by a sequence of $O(m)$ multiplications of the matrix $\mathcal{M}$ by a vector. Both methods find dependencies modulo 2 among the columns of $\mathcal{M}$ in $O(mW(\mathcal{M}))$ bit operations, where the weight $W(\mathcal{M})$ of $\mathcal{M}$ is the number of non-zero entries of $\mathcal{M}$. It follows that the term $M(L[\frac{r}{2}, \beta])$ in Expression (5.3) can be simplified to $L[\frac{r}{2}, 2\beta]$. Storage requirements are limited to storage of the original sparse matrix $\mathcal{M}$ and an m-dimensional vector over $(\mathbf{Z}/2\mathbf{Z})^k$, where the constant k is the *blocking factor* used. Refer to Chapter 7 on block Lanczos for more information on these methods. Montgomery contributed not just to block Lanczos, but also did a lot of work on a preprocessing step that is commonly used and that is generally referred to as *filtering*. The main ideas of this preprocessing step are described in the next section.

5.2.6 Filtering

Let the notation be as in the previous section. Filtering refers to a collection of methods that aim to transform the $m \times (m + \omega)$-matrix $\mathcal{M}$ into an $m' \times (m' + \omega')$-matrix $\mathcal{M}'$ for which $m'W(\mathcal{M}')$ is smaller than $mW(\mathcal{M})$ and such that dependencies modulo 2 among the columns of $\mathcal{M}'$ easily lead to dependencies modulo 2 among the columns of $\mathcal{M}$. The methods of the previous section can then profitably be applied to $\mathcal{M}'$ instead of $\mathcal{M}$: in practice a speedup of one

or more orders of magnitude may be expected. Background and more details about the material presented here can be found in [17, 18, 78]. Moduli different from 2, as required for the application in Section 5.5.1 on page 24, are handled in a slightly different but similar manner. Below, $\mathcal{M}$ refers to the $m \times (m + \omega)$-matrix in transition, with changing values for m and ω until $\mathcal{M} = \mathcal{M}'$, $m = m'$, and $\omega = \omega'$, for the final matrix $\mathcal{M}$.

Filtering proceeds by first removing duplicates of columns that correspond to identical relations (cf. Section 5.2.1 on page 2), next alternatedly removing singleton columns and sets of columns that are referred to as *cliques*, and finally combining the remaining columns in a *merge* step. These four steps are further described below. Note that, to avoid useless dependencies, duplicates must be removed irrespective of attempts to lower $mW(\mathcal{M})$.

Removing duplicates. In practice duplicate relations turn out to be unavoidable: lattice sieving with many distinct special q-primes will produce identical relations (cf. Section 5.5.2 on page 30), prematurely stopped relation collection jobs may have been restarted, or different relation collection methods may be used for the same factorization (cf. [19]).

Assuming canonical representations of relations, a few piped Unix commands remove duplicates at minimal human effort. The storage resources and time required may, however, become substantial. It is common to apply a hash function to each canonical representation, and to locate and further inspect the collisions. Appropriate hash functions are easily designed depending on the application. Refer to [17, Section 2.1] for an example of a hash function proposed by Montgomery that is injective for the relations as generated for the factorization reported in [19] (so that collisions correspond to duplicate relations).

Removing singletons. If there is a row in $\mathcal{M}$ that contains only a single entry that is non-zero modulo 2 (or another applicable modulus), then the column containing that non-zero entry can not occur in a dependency. Such columns, called *singletons*, can be removed from $\mathcal{M}$. This is easily done using a frequency table, but because each removal may generate one or more new singletons, several passes are normally required before all singletons have been removed. For large collections of relations each singleton-removal pass is quite time-consuming, with a quick drop in the number of removals during the later passes. Continuing until the very end may therefore not be worth the effort.

Removing cliques. To have a better chance to get a low $mW(\mathcal{M})$-value many more relations are collected than necessary to make the original matrix oversquare (even until the original ω is much larger than m). As mentioned in [78] (and as for instance used in the structured Gaussian elimination step of the factorization reported in [56]) the simplest approach is to remove the excess

columns in (decreasing) order of their number of odd entries, until the excess is deemed small enough. But [78] also suggests another approach, which is further pursued in [17, 18] following ideas of Montgomery. It has become the most common way to remove the excess columns until ω is reduced to approximately $\frac{m}{2}$.

The method is, in filtering context, referred to as *clique removal*, notwithstanding the non-standard definition of cliques. Consider the graph with vertex set corresponding to the set of relations with an edge connecting two vertices if there is a row in $\mathcal{M}$ such that the two corresponding relations are the only two relations that share a non-zero entry in that row. The components of this graph are called cliques in [17]. It follows that removal from $\mathcal{M}$ of a single relation in a clique triggers a chain of singletons in the same clique, so that the clique can be removed in its entirety. Fast recognition – and removal – of cliques is therefore an efficient way to deal with large amounts of interdependent excess columns, while also lowering the number of rows.

With $(e_{v,p})_{p\in P}$ denoting a column in $\mathcal{M}$ (i.e., a relation), and given a frequency table containing, for each row in $\mathcal{M}$, the total number o_p of odd entries in the row for p in $\mathcal{M}$, it is easy to compute the value $\sum_{p\in P:e_{v,p}\text{ odd}} 2^{-o_p}$ for each column. Cliques may now be removed by removing the columns for which the computed value is at least $\frac{1}{4}$. Because this may remove too many other columns as well, initially a cut-off larger than $\frac{1}{4}$ may be used, gradually lowering it until a targeted excess remains. The value will be at least $\frac{1}{2}$ for newly created singletons, so they have a good chance to be removed during a next round of clique removal with an updated frequency table.

Merging. If there is a $p \in P$ for which $o_p = 2$, it is advantageous to replace the two columns in which p occurs an odd number of times by their sum, because as a result m decreases by at least one and $W(\mathcal{M})$ decreases by at least two. This is called a *two-way merge*. After the two-way merges have been carried out, the process can be repeated for $\mathfrak{m}$-way merges, for increasing values of $\mathfrak{m}$, where an $\mathfrak{m}$-way merge replaces the $\mathfrak{m}$ columns sharing a particular p-value with $o_p = \mathfrak{m}$ by $\mathfrak{m} - 1$ independent, pair-wise sums among those $\mathfrak{m}$ columns. The least weight-increasing set of $\mathfrak{m}-1$ pairs is easily determined as a minimal spanning tree in the complete graph induced by the $\mathfrak{m}$ columns. The overall effect for larger $\mathfrak{m}$-values may, however, become counterproductive. Merges for larger $\mathfrak{m}$-values are therefore followed by removal of the heaviest columns, as long as the oversquareness ω remains large enough.

As a result of the merging step, the final matrix $\mathcal{M}$ is written as the product $\mathcal{M} = \widetilde{\mathcal{M}}\mathcal{T}$ of a pre-merger matrix $\widetilde{\mathcal{M}}$ and a merging-transformation matrix $\mathcal{T}$. Because in the preferred methods to find dependencies the same final matrix $\mathcal{M}$ is repeatedly multiplied by a (changing) vector (cf. Section 5.2.5 on page 7),

it is advantageous to use this representation $\widetilde{\mathcal{M}}\mathcal{T}$ of $\mathcal{M}$ if $W(\widetilde{\mathcal{M}}) + W(\mathcal{T}) < W(\mathcal{M})$; this approach was first used in [45].

5.2.7 Overall effort

With these insights (which may be anachronistic, depending on the context), Expression (5.3) on page 5 becomes

$$L[\tfrac{r}{2}, \max(\beta + \tfrac{\psi r}{2\beta}, 2\beta)]. \tag{5.5}$$

Optimization of Expression (5.5) still depends on the applicable r- and ψ-values, as further explained below. In more generality, the overall effort can be expressed as

$$L[\tfrac{r}{2}, \max((1+\sigma)\beta + \tfrac{\psi r}{2\beta}, \mu\beta)] \tag{5.6}$$

with σ representing the smoothness testing effort and μ the linear algebra exponent. The value for σ ranges from 0 (as in Expression (5.5)) for elliptic curve-based smoothness testing and for sieving (if applicable), to 1 for trial division (with $\sigma = \frac{1}{2}$ for Pollard's rho method). The linear algebra exponent μ ranges from 2 (as in Expression (5.5)) for the methods by Lanczos and Wiedemann, to 3 for Gaussian elimination (with $\mu = \log_2 7$ for Strassens's method and $\mu \approx 2.373$ for the Coppersmith-Winograd method). An overview of the results as of 1983 (some of which may have been improved since then due to more efficient auxiliary steps) is given in [74, Table on page 93], the most important ones of which are also presented below.

5.3 Pre-sieving general purpose factoring

Let the notation be as in Section 5.2 on page 2, with P the set of primes up to $L[\frac{r}{2}, \beta]$ for $r, \beta > 0$ to be specified below.

5.3.1 Dixon's random squares method

Not the earliest but conceptually the most straightforward general purpose factoring method that requires subexponential effort is John Dixon's *random squares method* [33]. It has never been proved to be practical, because by the time it was proposed more practical methods already existed. The random squares method selects at random integers $v \in \{1, 2, \ldots, n-1\}$ that have not been selected before, computes $v^2 \bmod n = w \in \{0, 1, \ldots, n-1\}$, assumes that $w \neq 0$ (because n can directly be factored if $w = 0$), uses trial division to

write $w = w' \prod_{p \in P} p^{e_{v,p}}$ with $w' \in \mathbf{Z}$ free of factors in the set of primes P, and if $w' = 1$ adjoins $(v, (e_{v,p})_{p \in P})$ to the set of relations. Once enough relations have been found, it uses Gaussian elimination to find dependencies.

Expression (5.6) with $\sigma = 1$ and $\mu = 3$ applies to Dixon's random squares method. The numbers w that are tested for smoothness can be bounded by $n = L[1, 1]$, so that $r = \psi = 1$ and the overall effort becomes:

$$L[\tfrac{1}{2}, \max(2\beta + \tfrac{1}{2\beta}, 3\beta)],$$

which is minimized for $\beta = \frac{1}{2}$ and becomes $L[\frac{1}{2}, 2]$. The effort of the matrix step is $L[\frac{1}{2}, \frac{3}{2}]$, which is dominated by the relation collection effort $L[\frac{1}{2}, 2]$. Values v for which $v < \sqrt{n}$ are useless, but as $v < \sqrt{n}$ with probability $n^{-\frac{1}{2}} = L[1, -\frac{1}{2}]$ this is unlikely to occur because only $L[\frac{1}{2}, \frac{3}{2}]$ values will be selected.

With faster smoothness testing and linear algebra methods (both anachronistic), the overall effort becomes (cf. Expression (5.5) with $r = \psi = 1$)

$$L[\tfrac{1}{2}, \max(\beta + \tfrac{1}{2\beta}, 2\beta)],$$

which is minimized for $\beta = \frac{1}{2}\sqrt{2}$ and becomes $L[\frac{1}{2}, \sqrt{2}]$. In this analysis both steps require the same effort in L-notation, i.e., disregarding everything that disappears in the $o(1)$-terms.

Using least absolute remainders for w (and adjoining -1 to P) does not change $L[\frac{1}{2}, 2]$ or $L[\frac{1}{2}, \sqrt{2}]$, but should make the method a bit faster in practice. Both these efforts are larger than the efforts required by the factoring methods considered in the remainder of this chapter. However, for Dixon's random squares method the analysis does not involve heuristics. See also Section 5.6 on page 44.

5.3.2 Continued fraction method

The *continued fraction method* (often referred to as CFRAC) developed by Morrison and Brillhart [65] represented the state of the art in general purpose integer factoring from 1970, when $F_7 = 2^{2^7} + 1$ was factored, until the mid 1970s. A special purpose hardware device (*the Georgia Cracker*) was built implementing it [80]. The continued fraction method was used to factor many Cunningham numbers [12]. It inspired the development of faster general purpose factoring methods, as further described in Section 5.4 below.

From the above analysis of Dixon's random squares method it follows that there are two main issues that would have to be addressed in order to get a more efficient factoring method: the speed of the smoothness test and the size of the integers w to be tested for smoothness. The second issue had already been

dealt with in the continued fraction methods, several years before Dixon proposed his random squares method. Although generating the w-values is more cumbersome in the continued fraction method than in Dixon's random squares method, this disadvantage is far outweighed by their much smaller size and thus substantially larger smoothness probability. As explained in detail in [65], and as follows from for instance [41, Theorem 164], the continued fraction expansion of $\sqrt{n}$ leads to a sequence of triples $(v_i, t_i, w_i) \in \mathbf{Z}^3$ for $i = 0, 1, 2, \ldots$ such that

$$v_i^2 - nt_i^2 = (-1)^i w_i \text{ where } 0 < w_i < 2\sqrt{n}. \tag{5.7}$$

It follows that for those i for which w_i is found to be smooth, the value v_i along with the vector of exponents of the factorization of w_i (including the sign) leads to a relation. With $r = 1$ and $\psi = \frac{1}{2}$ (as $0 < w_i < 2\sqrt{n} = L[1, \frac{1}{2}]$), trial division ($\sigma = 1$), and Gaussian elimination ($\mu = 3$), the overall effort from Expression (5.6) on page 11 becomes

$$L[\tfrac{1}{2}, \max(2\beta + \tfrac{1}{4\beta}), 3\beta],$$

which is minimized for $\beta = \frac{1}{4}\sqrt{2}$ and becomes $L[\frac{1}{2}, \sqrt{2}]$. The effort $L[\frac{1}{2}, \frac{3}{4}\sqrt{2}]$ of the matrix step is again dominated by the effort of relation collection, in accordance with Morrison's and Brillhart's practical experience. This asymptotic result was first, and informally, derived in the mid 1970s by Schroeppel: informal because the effort of the matrix step was not included in his argument; because the smoothness result used (as stated in Section 5.2.2 on page 4) had by then not been fully proved yet; because the w_i-values are chosen deterministically and can hardly be argued to behave as randomly selected positive integers at most $L[1, \frac{1}{2}]$; and finally because only primes p with Legendre symbol $(\frac{n}{p}) \in \{0, 1\}$ can occur in the factorizations of the w_i-values, thus requiring the later and more refined argument from [84, Theorem 74].

With $\sigma = 0$ and $\mu = 2$ as in Expression (5.5) on page 11 (anachronistic, because the required methods did not exist yet in 1970), and the customary heuristic handwaving, the effort is reduced to

$$L[\tfrac{1}{2}, \max(\beta + \tfrac{1}{4\beta}), 2\beta],$$

which is minimized for $\beta = \frac{1}{2}$ and becomes $L[\frac{1}{2}, 1]$ (with balanced efforts for the two steps).

Morrison and Brillhart describe how, depending on n, it is often advantageous to replace n by kn for a small positive *multiplier* $k \in \mathbf{Z}$, in order to boost the smoothness probabilities by aiming for more small primes with $(\frac{kn}{p}) = 1$ than with $(\frac{n}{p}) = 1$, or even to use several k-values (most likely leading to more primes that may occur). They also suggest to allow in the factorizations of the

w_i-values a large prime less than the square of the smoothness bound. As mentioned in Section 5.2 on page 2 such *large prime relations* can be recognized at no additional effort during trial division. A pair of large prime relations with the same large prime is easily transformed into a single regular relation (with, however, on average more non-zero entries in the exponent-vector and thus a less sparse matrix).

5.4 Linear and quadratic sieve

5.4.1 Linear sieve

Schroeppel found a way to replace trial division by sieving, as introduced in Section 5.2.4 on page 6, while keeping ψ almost as small as in the continued fraction method, namely $\psi = \frac{1}{2} + o(1)$. Despite a promising start the practical potential of his *linear sieve* was never conclusively shown: according to [83] its first attempted factorization – that of the eighth Fermat number $F_8 = 2^{2^8} + 1$ in 1980, and a tour de force at that time – was cut short during the first stage of the linear algebra step, because the factorization of F_8 was independently announced by others (and later reported in [11]). The linear sieve work on F_8 remains unpublished till the present day and was, at the time, only known to those who had been so fortunate to attend the single talk that Schroeppel ever gave about his linear sieve [83].

In the linear sieve the values tested for smoothness are

$$(i + [\sqrt{n}])(j + [\sqrt{n}]) - n = ij + (i + j)[\sqrt{n}] + [\sqrt{n}]^2 - n \tag{5.8}$$

for $i, j \in \mathbf{Z}$ of relatively small absolute value and with, say, $|i| \geq |j|$. Values as in Expression (5.8) have two advantages, and lead to one complication. The first advantage is that they are easier to generate than the w_i-values in Expression (5.7) as used in the continued fraction method, while having a comparable smoothness probability: because $[\sqrt{n}]^2 - n$ is of order $\sqrt{n}$, each value in Expression (5.8) is only of order $|i + j|\sqrt{n}$ if $|i|$ and $|j|$ are relatively small. More precisely, if $|i|$ and $|j|$ are bounded by $L[u_i, \gamma_i]$ and $L[u_j, \gamma_j]$, respectively, for some $u_i, u_j < 1$, then

$$\begin{aligned} |ij + (i + j)[\sqrt{n}] + [\sqrt{n}]^2 - n| &\leq L[u_i, \gamma_i]L[u_j, \gamma_j] + \\ &\quad (L[u_i, \gamma_i] + L[u_j, \gamma_j])L[1, \tfrac{1}{2}] + L[1, \tfrac{1}{2}] \\ &= L[1, \tfrac{1}{2}]. \end{aligned}$$

Thus, when expressed in L-notation the values generated by Expression (5.8) are of the same order $L[1, \frac{1}{2}]$ as the w_i-values in the continued fraction method,

and the smoothness disadvantage of $|i+j|\sqrt{n}$ compared to $2\sqrt{n}$ in the continued fraction method disappears in the $o(1)$. It follows that $r = 1$ and $\psi = \frac{1}{2}$.

The second advantage is that for fixed j Expression (5.8) is a linear polynomial in i. This implies that smoothness testing can be done using sieving. From the sieving analysis in Section 5.2.4 on page 6 it follows that as long as the length $L[u_i, \gamma_i]$ of the interval of i-values is at least as large as the smoothness bound $L[\frac{1}{2}, \beta]$, the sieving effort for a fixed j equals $L[u_i, \gamma_i]$. An overall sieving effort of $L[u_i, \gamma_i]L[u_j, \gamma_j]$ then follows.

A complication arises from the fact that a smooth $w = (i+[\sqrt{n}])(j+[\sqrt{n}])-n$ generated by Expression (5.8) leads to

$$w = \prod_{p \in P} p^{e_{i,j,p}} \quad \text{where} \quad w \equiv (i + [\sqrt{n}])(j + [\sqrt{n}]) \bmod n \tag{5.9}$$

which does not conform to Condition (5.1) on page 3. This is easily fixed by taking $i = j$, an idea that was discarded by Schroeppel for reasons set forth below [83]. Schroeppel fixed it in another manner, namely by adjoining to P the $(i + [\sqrt{n}])$- and $(j + [\sqrt{n}])$-values with $|i| \leq L[u_i, \gamma_i]$ and $|j| \leq L[u_j, \gamma_j]$. With $e_{i,j,i+[\sqrt{n}]} = e_{i,j,j+[\sqrt{n}]} = -1$, this turns (5.9) into

$$\frac{w}{(i+[\sqrt{n}])(j+[\sqrt{n}])} = \prod_{p \in P} p^{e_{i,j,p}} \quad \text{where} \quad \frac{w}{(i+[\sqrt{n}])(j+[\sqrt{n}])} = 1 \bmod n,$$

which is of the required form. As a consequence, however, the cardinality of P, and thus the number of relations to be found, increases from $L[\frac{1}{2}, \beta]$ to $L[\frac{1}{2}, \beta] + \max(L[u_i, \gamma_i], L[u_j, \gamma_j])$. To find these relations over a search space of $L[u_i, \gamma_i]L[u_j, \gamma_j]$ elements, it must be the case that

$$L[u_i, \gamma_i]L[u_j, \gamma_j] \geq (L[\tfrac{1}{2}, \beta] + \max(L[u_i, \gamma_i], L[u_j, \gamma_j]))L[\tfrac{1}{2}, \tfrac{1}{4\beta}] \tag{5.10}$$

because, as shown above, the values to be tested for smoothness are of order $L[1, \frac{1}{2}]$ and are thus heuristically assumed to be $L[\frac{1}{2}, \beta]$-smooth with probability $L[\frac{1}{2}, -\frac{1}{4\beta}]$. It follows that the optimal u_i and u_j satisfy $\max(u_i, u_j) = \frac{1}{2}$. If $u_i \neq u_j$ then it must be the case that $\gamma_i \geq \gamma_i + \frac{1}{4\beta}$ or that $\gamma_j \geq \gamma_j + \frac{1}{4\beta}$ (because of Condition (5.10)), which is impossible. Thus $u_i = u_j = \frac{1}{2}$, simplifying Condition (5.10) to

$$L[\tfrac{1}{2}, \gamma_i + \gamma_j] \geq L[\tfrac{1}{2}, \max(\beta, \gamma_i, \gamma_j) + \tfrac{1}{4\beta}]$$

and thus

$$\gamma_i + \gamma_j \geq \max(\beta, \gamma_i, \gamma_j) + \tfrac{1}{4\beta}. \tag{5.11}$$

The relation effort is bounded from below by $L[u_i, \gamma_i]L[u_j, \gamma_j] = L[\frac{1}{2}, \gamma_i + \gamma_j]$ (as argued above), attaining this lower bound if $\gamma_i \geq \beta$, and the linear

algebra effort is $L[\frac{1}{2}, \mu \max(\beta, \gamma_i, \gamma_j)]$. Because $\beta + \frac{1}{4\beta} \geq 1$ (reaching its minimal value 1 for $\beta = \frac{1}{2}$) it follows from Condition (5.11) that $\gamma_i + \gamma_j \geq 1$ and thus that $\max(\gamma_i, \gamma_j) \geq \frac{1}{2}$, so that the efforts are bounded from below by $L[\frac{1}{2}, 1]$ and $L[\frac{1}{2}, \frac{\mu}{2}]$, respectively. The minima are achieved for $\beta = \gamma_i = \gamma_j = \frac{1}{2}$, which is optimal.

It is impossible to lower the overall effort (thus, if $\mu > 2$, the effort of the linear algebra step) by balancing the two efforts involved: for $\beta = \frac{1}{2}$ this is obvious, and if $\beta \neq \frac{1}{2}$, then $\beta + \frac{1}{4\beta} > 1$ and thus $\max(\gamma_i, \gamma_j) > \frac{1}{2}$ and the linear algebra effort becomes larger than $L[\frac{1}{2}, \frac{\mu}{2}]$. With (not anachronistic) $\mu = \log_2 7$ due to Strassen's method, the overall effort of Schroeppel's linear sieve narrowly beats the continued fraction method's $L[\frac{1}{2}, \sqrt{2}]$ because $\frac{1}{2} \log_2 7 \approx 1.404 < 1.414 \approx \sqrt{2}$ (see also [74, Table on page 93]).

The resulting optimal (but heuristic, expected and asymptotic) relation collection effort $L[\frac{1}{2}, 1]$ is the factoring effort that was cited in [81, Section IX.A], neglecting the dominating term $L[\frac{1}{2}, \frac{\mu}{2}]$ for the linear algebra step. At the time this was somewhat optimistic but also understandable because experience with the continued fraction method had shown that the linear algebra effort was consistently negligible compared to the relation collection effort. For the purposes of [81], the optimism was later justified by the development of faster linear algebra methods (with $\mu = 2$), and then turned out to be too pessimistic due to the number field sieve.

Variant with $i = j$. As mentioned above, Schroeppel considered taking $i = j$ but rejected this idea, even though (5.9) with $i = j$ would directly conform to Condition (5.1) on page 3 without adjoining the $(i + [\sqrt{n}])$- and $(j + [\sqrt{n}])$-values to P (while also effectively reducing the size of P by a factor of two, as shown below). Schroeppel argued that, with i and j independently bounded by $L[\frac{1}{2}, \frac{1}{2}]$, Expression (5.8) on page 14 generates a total of $L[\frac{1}{2}, 1]$ values that are all bounded by $L[\frac{1}{2}, \frac{1}{2}]\sqrt{n}$ in absolute value [83]. To generate the same number of values with $i = j$, the bound on i becomes $L[\frac{1}{2}, 1]$, resulting in a bound of $L[\frac{1}{2}, 1]\sqrt{n}$ on the absolute values generated by Expression (5.8) on page 14. In L-notation, $L[\frac{1}{2}, \frac{1}{2}]\sqrt{n}$ and $L[\frac{1}{2}, 1]\sqrt{n}$ are both equal to $L[1, \frac{1}{2}]$, and both lead to smoothness probability $L[\frac{1}{2}, -\frac{1}{4\beta}]$. But in practice the choice $i = j$ leads to noticeably lower smoothness probabilities. The latter effect was perceived to be worse than having to generate about twice as many relations, because it would result in an overall slowdown of the relation collection step. The more cumbersome linear algebra step that Schroeppel had to deal with by allowing $i \neq j$ was considered to be a mere nuisance because, so far, the matrix effort had been futile compared to the relation collection effort. New developments, however, and to some extent Schroeppel's own analysis and experience, proved

him wrong, because it turned out that even with $i = j$ the i-values can be kept small, as further shown in Section 5.4.3 on the next page and Section 5.4.4 on page 19. Schroeppel also reported [83] that he initially rejected the use of a multiplier as had been used in the continued fraction method, but later reconsidered, and that he allowed two large primes per relation, about a decade before that was independently done in [58].

5.4.2 Quadratic sieving: plain

Unfazed by the issue pointed out by Schroeppel, Pomerance proposed using Schroeppel's linear sieve with $i = j$. He called it *quadratic sieve* because, similar to Schroeppel's linear sieve, it uses a sieve to locate smooth values of the quadratic polynomial

$$(i + [\sqrt{n}])^2 - n = i^2 + 2i[\sqrt{n}] + [\sqrt{n}]^2 - n \tag{5.12}$$

(cf. Section 5.2.4 on page 6). Pomerance's description [74] is the first paper containing careful and accessible explanations and thorough analyses of general purpose factoring methods and their variants, setting an example for later publications and turning the subject into a more serious scientific endeavor.

Initial results obtained by the quadratic sieve were not stellar, with [37] reporting a 47-digit factorization; this may be compared to Schroeppel's 78-digit linear sieve effort that was aborted during the linear algebra step, and which had, at the time, garnered little or no attention. It took several additional contributions – notably by Jim Davis, Diane Holdridge and Gus Simmons, by Montgomery, and by Pomerance, Jeffrey Smith and Randy Tuler – to turn the quadratic sieve into the state of the art in general purpose integer factoring, a position it held until 1994. These developments are described below.

An advantage of quadratic sieve over linear sieve is the simplified analysis and, for $\mu > 2$, its better overall heuristic asymptotic effort. Because $i = j$, the generic analysis from Section 5.2.3 on page 5 applies with $r = 1$, $\psi = \frac{1}{2}$, and $\sigma = 0$ in Expression (5.6) on page 11. More precisely, redoing the linear sieve effort analysis with $i = j$, the original cardinality $L[\frac{1}{2},\beta]$ of P and an i-interval of length $L[u,\gamma]$ for some u and γ with $0 < u < 1$ and $\gamma > 0$, Condition (5.10) on page 15 simplifies to

$$L[u,\gamma] \geq L[\tfrac{1}{2},\beta + \tfrac{1}{4\beta}] \tag{5.13}$$

because the values to be tested for smoothness, in absolute value bounded by $L[u,\gamma]L[1,\frac{1}{2}] = L[1,\frac{1}{2}]$ (since $u < 1$), are heuristically assumed to be $L[\frac{1}{2},\beta]$-smooth with probability $L[\frac{1}{2},-\frac{1}{4\beta}]$. It follows from Condition (5.13) that $L[u,\gamma] \geq L[\frac{1}{2},\beta]$ so that, with the sieving analysis from Section 5.2.4

on page 6, the relation collection effort becomes $L[u, \gamma]$. Minimizing the sum of the relation collection effort $L[u, \gamma]$ and the linear algebra effort $L[\frac{1}{2}, \mu\beta]$ under Condition (5.13), first leads to $u = \frac{1}{2}$ and then with $\gamma = \beta + \frac{1}{4\beta}$ to overall effort $L[\frac{1}{2}, \max(\beta + \frac{1}{4\beta}, \mu\beta)]$ (as indeed in Expression (5.6) with $r = 1$, $\psi = \frac{1}{2}$, and $\sigma = 0$), which depends on μ. For $\mu = 3$ it results in $\beta = \frac{1}{4}\sqrt{2}$ and $L[\frac{1}{2}, \frac{3}{4}\sqrt{2}] = L[\frac{1}{2}, 1.061]$ and for $\mu = \log_2 7$ in $\beta = 0.372$ and $L[\frac{1}{2}, 1.044]$. For $\mu = 2$ it results in $\beta = \frac{1}{2}$ and reaches its minimal value $L[\frac{1}{2}, 1]$. In all cases the efforts of the two steps are balanced.

If p divides $(i + [\sqrt{n}])^2 - n$ then $n \equiv (i + [\sqrt{n}])^2 \bmod p$, so that n is a square modulo p. It follows that only primes p with Legendre symbol $(\frac{n}{p}) = 1$ can occur in the factorizations of values generated by Expression (5.12), as in the continued fraction method. Following Morrison and Brillhart, the use of a suitable multiplier is therefore recommended. Also, the condition $(\frac{n}{p}) = 1$ effectively halves the size of P, making the quadratic sieve linear algebra step in practice yet again easier compared to linear sieve. Because Expression (5.12) is a quadratic polynomial in i, finding the roots modulo the primes in P is more cumbersome than for the linear polynomials in the linear sieve; this issue is further discussed on page 22 in Section 5.4.4. The growth of the polynomial values behaves according to Schroeppel's prediction and has a noticeably counterproductive effect compared to linear sieve. In the remainder of this section it is shown how this problem was overcome. When expressed in the L-notation, all variants presented below require the same effort: the speedups, though practically worthwhile, all disappear in the $o(1)$.

5.4.3 Quadratic sieving: fancy

Davis, Holdridge and Simmons in [29] were the first who managed to avoid a single large sieving interval and the resulting growth of the values to be tested for smoothness. Their method, referred to by the authors as *quadratic sieving: fancy*, proved to be more effective than the plain quadratic sieve as used in [37]. In 1984 it was used to set a 71-digit factorization record: on a Cray X-MP mainframe computer the relation collection took 8.75 hours, followed by 45 minutes for the linear algebra.

Assume that as a result of regular sieving over $i \in I$ with the polynomial in Expression (5.12) on page 17 a number of large prime relations (cf. Section 5.2.4 on page 6) has been found, each involving a single prime larger than the smoothness bound, but smaller than its square. For each such large prime

relation, corresponding to an equation of the form

$$(i_q + [\sqrt{n}])^2 - n = q(\prod_{p \in P} p^{e_{i_q,p}}) \tag{5.14}$$

involving a large prime q and an integer i_q with $i_q \in I$, Davis, Holdridge and Simmons use a sieve over $i \in I'$ to find smooth values of the quadratic integer polynomial

$$\frac{(qi+i_q+[\sqrt{n}])^2-n}{q} = qi^2 + 2i(i_q + [\sqrt{n}]) + \frac{(i_q+[\sqrt{n}])^2-n}{q}. \tag{5.15}$$

Each new smooth value thus found corresponds to a new large prime relation involving the large prime q, and can be combined with large prime relation (5.14) to produce a regular relation (which is, however, less sparse). If $I' \subseteq I$, the values of the polynomial in Expression (5.15) may be assumed to have smoothness probabilities comparable to or better than the values encountered during the sieve using the polynomial in Expression (5.12).

The advantage compared to the plain quadratic sieve is that a new sieve can be used for each large prime relation found as a result of the sieve using Expression (5.12). In particular, both I and I' can be chosen considerably smaller than the single large sieving interval used in the plain quadratic sieve. In [29] it is reported that with judicious choices for I and I' composites of approximately 64 digits could be factored at the same effort as approximately 56-digit ones using the original method.

5.4.4 Multiple polynomial quadratic sieve

In the mid 1980s, and independent of Davis, Holdridge and Simmons, Montgomery invented another way to keep the polynomial values in quadratic sieve relatively small. His method, now known as the *multiple polynomial quadratic sieve*, was published in [87] and quickly became the general purpose factoring method of choice. It allows straightforward *embarrassingly parallel* implementation, making it perfectly suitable to use the idle time of the networks of desktop computers that were emerging around that time. Indeed, in [87] the multiple polynomial quadratic sieve was used to factor an 81-digit composite on a local network, the first general purpose factorization surpassing Schroeppel's aborted 78-digit F_8-attempt, later reaching 87 digits as further described by Thomas Caron and Robert Silverman in [16]. This was quickly and independently followed by the first scientific distributed Internet computation that the author is aware of, reaching for the first time a 100-digit general purpose factorization, as described by the author and Mark Manasse in [57]. The

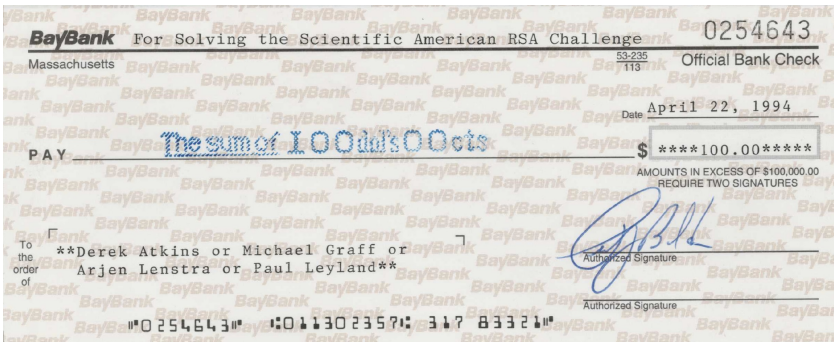

Figure 5.1 Proof of the factorization of the Scientific American challenge.

independent, non-networked but nevertheless widely parallelized implementation described in [4] trailed this development by a few unfortunate months; it was particularly challenging because it required daily, campus-wide floppy-disk collection and distribution [76].

Probably the most prominent result obtained using the multiple polynomial quadratic sieve was the 1994 factorization of the 129-digit challenge published in the August 1977 issue of Scientific American, solved in [6] by Derek Atkins, Michael Graff, the author and Paul Leyland (cf. Figure 5.1). It used the software from [57] and did not take advantage of the somewhat faster *self-initializing* method described at the end of this section, because improving the software was not found to be worth the effort: around that time the much more promising method from Section 5.5.3 on page 37 was about to become practical and competitive with the quadratic sieve (refer to [38, Section 5] for a direct comparison). Indeed, in [6] the 129-digit quadratic sieve factorization was already referred to as "probably the last gasp of an elderly workhorse". Two other old workhorses that were used for the last time for a record factorization were the combination of structured and regular Gaussian elimination (cf. Section 5.2.5 on page 7) and the 16 384-core massively parallel MasPar supercomputer: half a day on a desktop to reduce the original sparse bit-matrix $\mathcal{M}$ with $m \approx 525\,000$ to a dense matrix $\mathcal{M}'$ with $m' \approx 188\,000$, followed by two days of regular Gaussian elimination on the MasPar to find dependencies among the columns of $\mathcal{M}'$.

Montgomery showed how to construct a virtually limitless supply of integer polynomials f as in Expression (5.12) on page 17 and Expression (5.15) by focusing on their two crucial properties. The first of these is that they have a non-zero discriminant that is zero modulo n: this ensures that a smooth value

leads to a relation as in Condition (5.1) on page 3. The second is that, for any arbitrarily selected $\gamma \geq \beta$, the polynomial values must be bounded by $L[\frac{1}{2}, \gamma]\sqrt{n}$ over the sieving interval of length $L[\frac{1}{2}, \gamma]$. This guarantees not just the usual $L[\frac{1}{2}, \beta]$-smoothness probability $L[\frac{1}{2}, -\frac{1}{4\beta}]$ but also makes it possible to sieve with many polynomials over short sieving intervals: in theory $L[\frac{1}{2}, \beta + \frac{1}{4\beta} - \gamma]$ polynomials, each over an interval of length $L[\frac{1}{2}, \gamma]$. In Expressions (5.12) and (5.15) this is achieved for polynomials of a specific form and specific γ-values, but there are many degrees of freedom that can be exploited, as shown below.

Following Montgomery's construction as described in [87] and [53, 4.16], consider values $f(i)$ of the quadratic integer polynomial

$$f(X) = a^2X^2 + bX + c \tag{5.16}$$

for integers i with $|i| \leq L[\frac{1}{2}, \gamma]$, with integers a, b, c such that the discriminant $\Delta = b^2 - 4a^2c$ of f is a small odd multiple of n. It follows that

$$f(i) \equiv (ai + \tfrac{b}{2a})^2 \bmod n,$$

so that each $L[\frac{1}{2}, \beta]$-smooth $f(i)$ leads to a relation as in Condition (5.1) on page 3. To bound $f(i)$ by $L[\frac{1}{2}, \gamma]\sqrt{n}$ for $|i| \leq L[\frac{1}{2}, \gamma]$, the leading coefficient a^2 of f must be of order $\frac{\sqrt{n}}{L[\frac{1}{2}, \gamma]}$. Furthermore, to maximize the probability that $f(i)$ is divisible by primes at most $L[\frac{1}{2}, \gamma]$, the leading coefficient a^2 must be free of prime factors at most $L[\frac{1}{2}, \gamma]$. These two conditions are satisfied if a is chosen as a prime number such that $a^2 \approx \frac{\sqrt{\Delta}}{L[\frac{1}{2}, \gamma]}$. To make sure that a solution to $b^2 \equiv \Delta \bmod 4a^2$ is easy to find as well, a is chosen such that $a \equiv 3 \bmod 4$ and with Legendre symbol $(\frac{\Delta}{a})$ equal to one: it follows that $\tilde{b} = \Delta^{\frac{a+1}{4}} \bmod a$ satisfies $\tilde{b}^2 \equiv \Delta \bmod a$, after which $\tilde{b}$ is lifted to b with $b^2 \equiv \Delta \bmod 4a^2$ by first solving $(\tilde{b} + ka)^2 \equiv \Delta \bmod a^2$ for k, which leads to

$$k = \tfrac{\Delta - \tilde{b}^2}{a}((2\tilde{b})^{-1} \bmod a) \bmod a$$

(known as Hensel's lemma), and defining $b = \tilde{b} + ka$ or $b = \tilde{b} + ka - a^2$ depending on which of the two is the proper, odd choice. The resulting b and $c = \frac{b^2 - \Delta}{4a^2}$ are of order $\frac{\sqrt{\Delta}}{L[\frac{1}{2}, \gamma]}$ and $L[\frac{1}{2}, \gamma]\sqrt{\Delta}$, respectively, as required.

The number of suitable a-values, and thus of suitable polynomials $f(X)$ as in (5.16), is of order $n^{\frac{1}{4}+o(1)}$. It is therefore a simple matter to parallelize the sieving effort for the multiple polynomial quadratic sieve: for any n-value for which it is worthwhile to parallelize the factoring effort, disjoint intervals containing an adequate supply of a-values can be farmed out to any realistic number of sieving clients, with the resulting relations collected at a central location.

This is how [87, 16] and, independently but later and on a larger scale, [57] worked.

Further improvements. Selecting the best value for γ involves a trade-off because smaller γ-values result in higher smoothness probabilities (of, on average, smaller $f(i)$-values), but also in more frequent sieve initialization, i.e., computing the roots of the polynomial $f(X)$ modulo all primes $p \leq L[\frac{1}{2}, \beta]$ with $(\frac{n}{p}) = 1$. This requires the relatively costly calculation of a^{-1} modulo all those primes. As shown in [79], Montgomery's construction of a and b allows a further generalization where a single a, chosen as the product of ℓ distinct primes from a collection of κ (much smaller) primes, gives rise to $2^{\ell-1}$ distinct b-values. In this so-called *self-initializing quadratic sieve*, due to Pomerance, Smith, and Tuler, the $2^{\ell-1}$ polynomials resulting from each of the $\binom{\kappa}{\ell}$ choices for a, can be ordered in such a way that the roots modulo p for the current polynomial lead with a few additions modulo p to the roots of the next polynomial. In this way the costly inversions can essentially be amortized over $\binom{\kappa}{\ell}2^{\ell-1}$ polynomials, leading to a speedup of about a factor of two over the multiple polynomial quadratic sieve [20]. Refer to [79], [4] and [68] for details and to [43] for a recent improvement.

An additional speedup of a similar order of magnitude can be obtained by allowing more than a single large prime per relation, as shown for the multiple polynomial quadratic sieve in [58] (re-inventing what Schroeppel had already used for linear sieve, but had never published) and in [61].

Though relevant and of some interest when they occurred, with hindsight all developments since the continued fraction method sketched above were only rather modest improvements of its basic idea. Probably the biggest single contributions were Schroeppel's informal first analysis of the smoothness bound trade-off and his introduction of sieving, followed by Pomerance's influential more formal treatment of the subject in [74]. The constant $\mathfrak{c}$ in the factoring effort estimate $L[\frac{1}{2}, \mathfrak{c}]$ slowly decreased over time, but got stuck at $\mathfrak{c} = 1$: as further shown below, (failed) attempts to further reduce $\mathfrak{c}$ were not sufficiently ambitious by targeting the wrong constant in $L[\frac{1}{2}, 1]$. An example is the *cubic sieve algorithm*, a never realized extension of the approach from [24] (see also [53, Section 4.E]).

As follows from Expression (5.6) on page 11, no general purpose factoring method that is based on the two-step approach from Section 5.2.1 on page 2 can improve on $L[\frac{1}{2}, \mathfrak{c}]$ for positive $\mathfrak{c}$ as long as the numbers to be tested for smoothness are of order $L[1, \psi] = n^{\psi+o(1)}$ for positive ψ. It took a new idea (or, actually, a sequence of new ideas) to replace this constant power $n^{\psi+o(1)}$ of n by a vanishingly small power of n: more precisely, by $L[\frac{2}{3}, \psi] = n^{o(1)}$, which

then results in overall effort $L[\frac{1}{3}, \mathfrak{c}]$ for some positive $\mathfrak{c}$ (cf. Expression (5.6)). This is further explained in the next section.

5.5 Number field sieve

While the polishing efforts described in the previous section were underway, an independent development took place that started as a cottage industry (cf. Figure 5.2) but that quickly took center stage. Triggered by the factorizations of the seventh Fermat number F_7 in 1970 and the eighth Fermat number F_8 in 1980 (cf. sections 5.3.2 on page 12 and 5.4.1 on page 14), and rightly concluding from [57] that the ninth Fermat number F_9 would be out of reach of general purpose factoring methods for the foreseeable future unless a breakthrough would occur, Pollard designed, in 1988, a new factorization method specifically targeted at Fermat numbers. After using it to factor F_7 on his 8-bit Philips P2012 computer (with 64K random access memory and two 640K disk drives), he sent a description of his method (later published as [72]) to Andrew Odlyzko, accompanied by a letter, dated August 31, 1988, with Richard Brent, John Brillhart, Hendrik Lenstra, Claus Schnorr, and Hiromi Suyama in copy:

```
For a 40-digit number the time is perhaps a little
longer than QS on my computer. With larger numbers,
for those able to attempt them, it may have an
advantage over QS.
...
(Perhaps I am talking nonsense?).
...
If F9 is still unfactored, then it might be a
candidate for this kind of method eventually?
I would be grateful for any comments.
```

Pollard was, of course, known for **not** talking nonsense, but Odlyzko did not take the bait. Lenstra, however, did. This led not only to the factorization of F_9 in 1990 – to Pollard's great numerological relief – but more importantly to the development of the number field sieve integer factoring method, the current state of the art in general purpose integer factorization. As sketched below, Montgomery later played an active role in the auxiliary steps that turned the number field sieve into a practical factoring method.

Pollard's original method, *factoring with cubic integers* as described in [72], applied only to integers of a special form. It led to the factorization method in [55] which was called the *number field sieve* (cf. [3]) and which was more general than Pollard's method because it could use quartic, quintic, etc. instead

Figure 5.2 Tidmarsh Cottage, the birthplace of the number field sieve.

of just cubic integers, but which still only applied to composites of a special form. This restriction was removed in [14], at which point the original number field sieve became known as the *special number field sieve*, and the new method from [14] as the *general number field sieve*. At this point in time, the "general" is dropped most of the time. In this section the various historical developments before and after Pollard's method from [72] and as collected in [54] are described.

5.5.1 Earlier methods to compute discrete logarithms

Compared to earlier general purpose integer factorization methods, Pollard's method in [72] introduced two main new ingredients: factorization into prime ideals of certain elements of an algebraic number field of degree three (or higher), and homomorphically mapping such elements to integers modulo n to get two distinct factorizations that are identical modulo n. Both ingredients had already been used for quadratic fields by Coppersmith, Odlyzko and Schroeppel in their *Gaussian integer method* from [24] to compute discrete logarithms over prime fields, a method that is related to Taher ElGamal's method from [35] to compute discrete logarithms over quadratic extensions of prime fields using prime ideal factorizations. The latter method was a generalization of an ear-

lier method to compute discrete logarithms over prime fields [92, 1] (and as mentioned in [71, Section 1]), which in turn was based on the same two-step approach to integer factorization described in Section 5.2.1 on page 2. The developments from [92, 1] via [35] to [24] that would ultimately lead to [72] and [54] are described below.

Discrete logarithms over prime fields. Let q be a prime number and let g be a generator of the multiplicative group $\mathbf{F}_q^\times$ of the finite field $\mathbf{F}_q$ of q elements. The *discrete logarithm* of $h \in \langle g \rangle$ with respect to g, denoted $\log_g h$, is the $x \in \mathbf{Z}/(q-1)\mathbf{Z}$ such that $g^x = h$. As shown in [92], the two-step approach to integer factorization from Section 5.2.1 can also be used to compute discrete logarithms, with Leonard Adleman in [1] being the first to use Schroeppel's approach to analyse that the required effort is subexponential in $\log q$ (cf. Section 5.2.2 on page 4). If in Dixon's random squares method from Section 5.3.1 on page 11 the values w are selected as $w = g^\chi \in \mathbf{F}_q$ for random exponents $\chi \in \mathbf{Z}/(q-1)\mathbf{Z}$ (and identifying $\mathbf{F}_q$ in the canonical manner with the set of integers $\{0, 1, \ldots, q-1\}$), a relation $w = \prod_{p \in P} p^{e_{\chi,p}}$ leads to the identity

$$\chi = (\sum_{p \in P} e_{\chi,p} \log_g p) \bmod (q-1).$$

With $|P|$ linearly independent relations the values $\log_g p$ for $p \in P$ can be found using linear algebra modulo $q-1$, after which, for each h for which $\log_g h$ must be calculated, values $\tau \in \mathbf{Z}/(q-1)\mathbf{Z}$ are randomly selected until $hg^\tau = \prod_{p \in P} p^{e_{\tau,p}}$ so that

$$\log_g h = (\sum_{p \in P} e_{\tau,p} \log_g p) - \tau \bmod (q-1).$$

Discrete logarithms over general finite fields. The above method works because of the canonical identification between the elements of $\mathbf{F}_q$ and the elements of the set of integers $\{0, 1, \ldots, q-1\}$. This makes it possible to embed $\mathbf{F}_q$ into the integers while transferring smoothness-related properties of the set of integers $\{0, 1, \ldots, q-1\}$ to the corresponding elements of $\mathbf{F}_q$. Given this simple approach, it is a natural question to ask what happens if prime fields are replaced by extension fields. In [67] it is shown that the same approach works again for fixed constant field characteristic q with the extension degree d going to infinity: with $f(X) \in \mathbf{F}_q[X]$ irreducible of degree d, the extension field $\mathbf{F}_{q^d}$ is isomorphic to $(\mathbf{F}_q[X])/(f)$, which is naturally embedded in $\mathbf{F}_q[X]$. An extension field element can thus be defined to be smooth if the corresponding polynomial of degree at most $d-1$ in $\mathbf{F}_q[X]$ factors into polynomials in $\mathbf{F}_q[X]$ of sufficiently small degrees. The required effort is of the form $L_q[\frac{1}{2}, \mathfrak{c}]$ for constant $\mathfrak{c} \in \mathbf{R}_{>0}$, i.e., subexponential in d (see [67] and also [53, 3.9-3.12]).

More refined methods exploit the considerable degree of freedom in the representation of field elements if the characteristic is fixed (but $d \to \infty$). Coppersmith, in his 1984 paper [21], was the first to achieve $L[\frac{1}{3}, c]$. After almost three decades this line of research was picked up again, resulting in a sequence of dramatic further improvements [39, 42, 8, 40].

Discrete logarithms over quadratic extension fields. Naively doing the same for $d = 2$ to compute discrete logarithms in $\mathbf{F}_{q^2}$, with prime q, fails because the elements of $\mathbf{F}_{q^2}$ would be identified with polynomials of degree at most one in $\mathbf{F}_q[X]$, via the isomorphism between $\mathbf{F}_{q^2}$ and $(\mathbf{F}_q[X])/(f)$. With the above definition of smoothness of polynomials, all elements are smooth and the algorithm becomes meaningless. ElGamal in [35] showed how to fix this. First he uses the same isomorphism $\mathbf{F}_{q^2} \simeq (\mathbf{F}_q[X])/(f)$ for the calculation of the w-values, which will be degree one polynomials in X over $\mathbf{F}_q$. In these polynomials he replaces X and $\mathbf{F}_q$ by α and $\mathbf{Z}$, respectively, where α is a zero of f regarded as an irreducible polynomial in $\mathbf{Z}[X]$ (with the usual canonical map between $\mathbf{F}_q$ and the set of integers $\{0, 1, \ldots, q-1\}$, and where irreducibility over $\mathbf{Z}$ follows from irreducibility modulo q). This results in w-values in $\mathbf{Z}[\alpha]$, the smoothness of which is then defined in terms of a smooth prime ideal factorization in the algebraic number field $\mathbf{Q}(\alpha) \simeq \mathbf{Q}[X]/(f)$.

Prime ideal factorization. As described in [35, Appendix C] for quadratic number fields and for higher degree number fields in [55], [14] and [56], prime ideal factorizations in $\mathbf{Q}(\alpha)$ lead to a myriad of issues. The present informal description is loosely based on [35, Appendix C] and [55, Sections 2, 3, 5] to cover both the present discrete logarithm application and the number field sieve in Section 5.5 on page 23. For simplicity it is assumed that $\mathbf{Z}[\alpha]$ is a unique factorization domain; for the more general case refer to [35, Appendix C], [55, Section 3] and [14].

Assume that f is monic and of degree d as above. Because the field $\mathbf{F}_{q^d}$ is isomorphic to $(\mathbf{F}_q[X])/(f)$, the generator g of the multiplicative group $\mathbf{F}_{q^d}^\times$ of $\mathbf{F}_{q^d}$ can be represented as a polynomial $g(X) \in (\mathbf{F}_q[X])/(f)$ of degree at most $d-1$. For random $\chi \in \mathbf{Z}/(q^d-1)\mathbf{Z}$, the element $w = g^\chi \in \mathbf{F}_{q^d}^\times$ is calculated as $g(X)^\chi \in (\mathbf{F}_q[X])/(f)$, which results in a polynomial $w(X) = (\mathbf{F}_q[X])/(f)$ of degree at most $d-1$. For the present purpose $d = 2$, so that the polynomial $w(X)$ has degree at most one. Although in sections 5.5.2, 5.5.3, and 5.5.4 below more general d-values are used, the different construction that is used there also leads to polynomials $w(X) \in (\mathbf{F}_q[X])/(f)$ of degree at most one. The resulting polynomial $w(X)$ can thus be written as $a - bX \in (\mathbf{F}_q[X])/(f)$. This polynomial is interpreted as $a - b\alpha \in \mathbf{Z}[\alpha]$, tested for smoothness in $\mathbf{Z}[\alpha]$, and if smooth written as a product over $\mathbf{Z}[\alpha]$ of *prime elements* in $\mathbf{Z}[\alpha]$. From this product a relation then follows. The test for smoothness is straightforward: $a - b\alpha$ is B-

smooth if and only if its norm $\mathbf{N}(a - b\alpha) = b^d f(\frac{a}{b}) \in \mathbf{Z}$ is B-smooth (note that the norm is a degree d integer polynomial that is homogeneous in a and b). The remaining steps are more involved, as briefly described in the next paragraphs.

Integer factors that a and b may have in common are easily dealt with in the usual manner. Therefore let $\gcd(a, b) = 1$ from now on, and let $\mathbf{N}(a - b\alpha)$ (and thus $a - b\alpha$) be B-smooth. One could now hope that if $\mathbf{N}(a - b\alpha) = \prod_{p \in P} p^{e_{a,b,p}}$ for some set of primes P, then $a - b\alpha = \prod_{\mathfrak{p} \in \mathfrak{P}} \mathfrak{p}^{e_{a,b,\mathfrak{p}}}$ with $\mathfrak{P}$ denoting a set of prime elements in $\mathbf{Z}[\alpha]$ that corresponds one way or another to P. This is indeed the case if "prime element" means *prime ideal*; the equality is interpreted as the factorization of the ideal $(a - b\alpha)$ into prime ideals; and if a final issue is addressed: generally speaking a prime ideal is not uniquely identified by its norm, so ambiguities have to be resolved. The latter is easily done too: because $\mathbf{N}(a - b\alpha) = b^d f(\frac{a}{b})$, a prime p divides $\mathbf{N}(a - b\alpha)$ if and only if $\frac{a}{b} \bmod p$ is a root of f modulo p. Therefore, it suffices to define

$$\mathfrak{P} = \{(p, z) : p \text{ prime}, \ p \leq B, \ z \in \mathbf{Z}, \ 0 \leq z < p, \ f(z) \equiv 0 \bmod p\}$$

and to rewrite the above factorization of $\mathbf{N}(a - b\alpha)$ as

$$\mathbf{N}(a - b\alpha) = \prod_{(p,z) \in \mathfrak{P}} p^{e_{a,b,p,z}}$$

where $e_{a,b,p,z} = 0$ if $a \neq bz \bmod p$. Note that for $d = 2$ at most two pairs in $\mathfrak{P}$ share the same prime. After identifying each pair $(p, z) \in \mathfrak{P}$ with the prime ideal $\mathfrak{p}$ generated by p and $z - \alpha$, the prime factorization of $\mathbf{N}(a - b\alpha)$ over $\mathfrak{P}$ corresponds to the prime ideal factorization

$$(a - b\alpha) = \prod_{\mathfrak{p} \in \mathfrak{P}} \mathfrak{p}^{e_{a,b,\mathfrak{p}}} \tag{5.17}$$

of the ideal $(a - b\alpha)$. These prime ideals $\mathfrak{p}$ are *first degree prime ideals* and are the only prime ideals that can occur in the prime ideal factorization of ideals of the form $(a - b\alpha)$.

Two more issues must be addressed to turn Equation (5.17) into a factorization of the element $a - b\alpha \in \mathbf{Z}[\alpha]$ that holds over $\mathbf{Z}[\alpha] = (\mathbf{Z}[X])/(f)$ and that can thus be turned into a factorization in $(\mathbf{F}_q[X])/(f) \simeq \mathbf{F}_{q^d}$. The latter is required for ElGamal's method to compute discrete logarithms in $\mathbf{F}_{q^2}$ and for the early version of the special number field sieve – later it turned out that the prime ideal factorization in Equation (5.17) suffices (thanks to two other additional ideas, mentioned on pages 39 and 41 in Section 5.5.3).

Factoring $a - b\alpha$ over $\mathbf{Z}[\alpha]$. As is, in Equation (5.17), the ideal $\mathfrak{p}$ generated by p and $z - \alpha$ does not contribute in a useful or meaningful fashion to a factorization of $a - b\alpha$ over $\mathbf{Z}[\alpha]$, because $\mathfrak{p}$ can not be interpreted as an element

of $\mathbf{Z}[\alpha]$. In the context of [35] and [55] this can be fixed by determining, for each $\mathfrak{p} = (p, z - \alpha) \in \mathfrak{P}$, an element $\mathfrak{g}_\mathfrak{p} \in \mathbf{Z}[\alpha]$ that generates the same ideal as $\mathfrak{p}$: this is the case if the norm $\mathbf{N}(\mathfrak{g}_\mathfrak{p})$ of $\mathfrak{g}_\mathfrak{p}$ equals p and $\mathfrak{g}_\mathfrak{p}$ regarded as a polynomial of degree at most $d - 1$ has a root z modulo p. Here $\mathbf{N}(\mathfrak{g}_\mathfrak{p})$ is as above if $d = 2$ (in which case $\mathfrak{g}_\mathfrak{p}$ is a polynomial in α of degree at most one); in general $\mathbf{N}(\mathfrak{g}_\mathfrak{p})$ is a degree d integer polynomial that depends on f and that is homogeneous in the d coefficients of $\mathfrak{g}_\mathfrak{p}$ (see also [55, 3.6]). In [35, Lemma 4] and [55, Section 3] a search-process is described that determines $\mathfrak{g}_\mathfrak{p}$ for all $\mathfrak{p} \in \mathfrak{P}$ and that is efficient for the polynomials at hand in [35] and [55]. Essentially, degree $d - 1$ integer polynomials with relatively small coefficients are inspected until all generators $\mathfrak{g}_\mathfrak{p}$ have been found. Once all found, the prime ideal factorization (5.17) of the ideal $(a - b\alpha)$ can be rewritten as

$$(a - b\alpha) = \prod_{\mathfrak{p} \in \mathfrak{P}} (\mathfrak{g}_\mathfrak{p})^{e_{a,b,\mathfrak{p}}}. \tag{5.18}$$

Even though $\mathfrak{g}_\mathfrak{p} \in \mathbf{Z}[\alpha]$, Equation (5.18) may not yet be the factorization of $a - b\alpha$ over $\mathbf{Z}[\alpha]$ that is aimed for, because the prime ideal generators $\mathfrak{g}_\mathfrak{p}$ are not unique. In principle, any choice for $\mathfrak{g}_\mathfrak{p}$ is as good as any other one, but different choices would lead to different factorizations of $a - b\alpha$, which can not be correct. This final issue is resolved by finding the *unit contribution*: if $\mathfrak{g}_\mathfrak{p}$ and $\bar{\mathfrak{g}}_\mathfrak{p}$ are distinct but generate the same prime ideal, then their quotient is a polynomial $\mathfrak{u} \neq 1$ in $\mathbf{Z}[\alpha]$ of norm equal to one. Such a $\mathfrak{u}$ is a *unit*. In more generality, given a choice of prime ideal generators $\mathfrak{g}_\mathfrak{p}$ for all $\mathfrak{p} \in \mathfrak{P}$, the quotient $\mathfrak{u}_{a,b}$ of $a - b\alpha$ and $\prod_{\mathfrak{p} \in \mathfrak{P}} (\mathfrak{g}_\mathfrak{p})^{e_{a,b,\mathfrak{p}}}$ satisfying Equation (5.18) is called the unit contribution. To be able to deal with the unit contributions, each much be written as a product over a fixed set of units. This is done as follows.

During the search for the prime ideals generators $\mathfrak{g}_\mathfrak{p}$, a minimal finite set $\mathfrak{U} \subset \mathbf{Z}[\alpha]$ of units can be determined that multiplicatively generates all units, directly by keeping the polynomials of norm equal to one or by considering quotients of two generators that have, in absolute value, the same norm [55, Section 3]. Once $\mathfrak{U}$ has been determined, integers $e_{a,b,\mathfrak{u}}$ can be found such that $\mathfrak{u}_{a,b} = \prod_{\mathfrak{u} \in \mathfrak{U}} \mathfrak{u}^{e_{a,b,\mathfrak{u}}}$. This can be done using table look-up or using (much faster) complex embeddings as described in [55, Section 5]. As a result it is found that

$$a - b\alpha = \Big(\prod_{\mathfrak{u} \in \mathfrak{U}} \mathfrak{u}^{e_{a,b,\mathfrak{u}}} \Big) \Big(\prod_{\mathfrak{p} \in \mathfrak{P}} \mathfrak{g}_\mathfrak{p}^{e_{a,b,\mathfrak{p}}} \Big) \tag{5.19}$$

holds over $\mathbf{Z}[\alpha] = (\mathbf{Z}[X])/(f)$.

Wrapping up discrete logarithms over quadratic extension fields. Returning, for $d = 2$, to where $a - b\alpha$ came from, namely from $g^\chi = a - bX \in (\mathbf{F}_q[X])/(f) \simeq \mathbf{F}_{q^2}$ where g generates $\mathbf{F}_{q^2}^\times$ and χ is chosen at random from

$\mathbf{Z}/(q^2-1)\mathbf{Z}$, Equation (5.19) is the relation that follows from the smoothness of $a - b\alpha$. With X substituted for α (also in all $\mathfrak{u} \in \mathfrak{U}$ and in $\mathfrak{g}_\mathfrak{p}$ for all $\mathfrak{p} \in \mathfrak{P}$) it holds for integer polynomials modulo the polynomial $f(X) \in \mathbf{Z}[X]$. It thus holds modulo q too and, with all polynomials interpreted as elements of $\mathbf{F}_{q^2}$, leads to

$$\chi = \log_g(a - bX) = (\sum_{\mathfrak{u}\in\mathfrak{U}} e_{a,b,\mathfrak{u}} \log_g \mathfrak{u} + \sum_{\mathfrak{p}\in\mathfrak{P}} e_{a,b,\mathfrak{p}} \log_g \mathfrak{g}_\mathfrak{p}) \bmod (q^2-1).$$

As usual, with $|\mathfrak{U}| + |\mathfrak{P}|$ relations the discrete logarithms of all $\mathfrak{u} \in \mathfrak{U}$ and all $\mathfrak{g}_\mathfrak{p}$ for $\mathfrak{p} \in \mathfrak{P}$ can be found using linear algebra modulo $q^2 - 1$. Individual discrete logarithms can then be calculated as described above. In [35] ElGamal has shown that the required effort is subexponential in $\log q$.

Gaussian integer method. Returning to the computation of discrete logarithms over prime fields $\mathbf{F}_q$, in [24] Coppersmith, Odlyzko and Schroeppel show how the ideas of Schroeppel's linear sieve can be used to substantially speed up the basic algorithm described in one of the earliest paragraphs of this section. One of their methods combines Gaussian integers (similar to ElGamal's method sketched above, with $d = 2$) with a homomorphism between the set $\mathbf{Z}[i]$ of Gaussian integers and the ring $\mathbf{Z}/q\mathbf{Z}$ of integers modulo q to find two distinct factorizations that are the same modulo q. This combination can be interpreted as the *degree two version* of what was later used in the number field sieve for general degrees, and is briefly described below.

Let $f(X) = X^2 - t \in \mathbf{Z}[X]$, where $|t|$ is small, $t < 0$, and t is a quadratic residue modulo q. With α such that $f(\alpha) = 0$, the same assumption as above is made that $\mathbf{Z}[\alpha]$ is a unique factorization domain (unnecessarily limiting the choice of t to just nine possibilities for the present simplified description). With $m \in \mathbf{Z}$ such that $m^2 \equiv t \bmod q$, the mapping φ from $\mathbf{Z}[\alpha]$ to $\mathbf{Z}/q\mathbf{Z}$ that maps $a - b\alpha$ to $a - bm \bmod q$ is a ring homomorphism because $f(\alpha) = 0 \equiv f(m) \bmod q$. It follows that if the integer $a - bm$ is B-smooth and the Gaussian integer $a - b\alpha$ is smooth as in Equation (5.19), and where a and b are coprime as usual, then

$$\prod_{p\le B} p^{e_{a,b,p}} = a - bm \equiv \varphi(a - b\alpha) \bmod q$$

and

$$\varphi(a - b\alpha) = \varphi\Big(\Big(\prod_{\mathfrak{u}\in\mathfrak{U}} \mathfrak{u}^{e_{a,b,\mathfrak{u}}}\Big)\Big(\prod_{\mathfrak{p}\in\mathfrak{P}} \mathfrak{g}_\mathfrak{p}^{e_{a,b,\mathfrak{p}}}\Big)\Big) = \Big(\prod_{\mathfrak{u}\in\mathfrak{U}} \varphi(\mathfrak{u})^{e_{a,b,\mathfrak{u}}}\Big)\Big(\prod_{\mathfrak{p}\in\mathfrak{P}} \varphi(\mathfrak{g}_\mathfrak{p})^{e_{a,b,\mathfrak{p}}}\Big).$$

This leads to the relation

$$\prod_{p\le B} p^{e_{a,b,p}} \equiv \Big(\prod_{\mathfrak{u}\in\mathfrak{U}} \varphi(\mathfrak{u})^{e_{a,b,\mathfrak{u}}}\Big)\Big(\prod_{\mathfrak{p}\in\mathfrak{P}} \varphi(\mathfrak{g}_\mathfrak{p})^{e_{a,b,\mathfrak{p}}}\Big) \bmod q \tag{5.20}$$

and thus to the identity

$$\sum_{p\le B} e_{a,b,p} \log_g p \equiv \Big(\sum_{\mathfrak{u}\in\mathfrak{U}} e_{a,b,\mathfrak{u}} \log_g \varphi(\mathfrak{u}) + \sum_{\mathfrak{p}\in\mathfrak{P}} e_{a,b,\mathfrak{p}} \log_g \varphi(\mathfrak{g}_\mathfrak{p}) \Big) \bmod (q-1)$$

between the discrete logarithms of a specific set of elements of $\mathbf{F}_q^\times \simeq (\mathbf{Z}/q\mathbf{Z})^\times$. With sufficiently many identities of this sort, all these discrete logarithms can be determined – assuming $\log_g g = 1$ is among them – after which individual logarithms can be found in, more or less, the customary fashion.

In [24] integers y, z of order $\sqrt{q}$ with $\frac{y}{z} \equiv m \bmod q$ are determined (by interrupting the iteration of the extended Euclidean calculation of $m^{-1} \bmod q$ approximately halfway) to replace $a - bm$ of approximate order q by $z(a - bm) \equiv az - by \bmod q$ of approximate order $\sqrt{q}$. This considerably increases the smoothness probabilities, at the cost of introducing an additional factor z on the right-hand side of Relation (5.20) (and an additional term $\log_g z$ on the right-hand side of the ensuing identity modulo $q-1$), but the basic idea remains the same. After this modification, the overall required effort becomes $L_q[\frac{1}{2}, 1]$, using, for the first time, the Lanczos method and thus $\mu = 2$ (cf. Section 5.2.5 on page 7 and Section 5.2.7 on page 11) for the linear algebra step. The method has for a long time been competitive with later number field sieve based discrete logarithm methods [91].

Interestingly, with $|a|, |b| \le L_q[\frac{1}{2}, \frac{1}{2}]$, the smoothness probabilities of the integers and of the algebraic integers are unbalanced: for the integers $az - by$ the probability is $L_q[\frac{1}{2}, -\frac{1}{2}]$, for the algebraic integers the smoothness is bounded below by a positive constant [24]. If a larger degree polynomial $f(X)$ is used, then the probabilities can be better balanced. This is precisely what happens in the number field sieve.

5.5.2 Special number field sieve

Pollard in [72] showed how the ideas from Section 5.5.1 on page 24 can be used with degree $d > 2$ to factor numbers of the form $x^3 - t$, for small $|t|$, while using a unique factorization domain $\mathbf{Z}[\alpha]$ for his example factorization of F_7. Elsewhere (cf. his letter quoted above in Section 5.5 on page 23) he mentioned that non-unique factorization and degree higher than three `seem possible and not too difficult`. This was indeed shown to be the case in the follow-up paper [55]. Using $d = 5$ and a rough first implementation of a generalized version of Pollard's method, several previously unfactored composites from the Cunningham tables [28, 12] were factored. Many of these numbers were at the time out of reach of the multiple polynomial quadratic sieve or its faster self-initializing variant. Several cases were encountered where $\mathbf{Z}[\alpha]$ was

not a unique factorization domain and had to be replaced by the ring of integers of $\mathbf{Q}(\alpha)$ to get unique factorization. These early experiments culminated in the summer of 1990 in the factorization of F_9, reported by the author, Lenstra, Manasse and Pollard in [56]. Since then much more refined implementations have been used to obtain a string of factorization records for Cunningham numbers, most recently the shared factorizations from [45] mentioned in the introduction (which also uses one of the ideas from [22], described in Section 5.5.4 on page 43). Pollard's method, now known as the special number field sieve, is at this point in time still the state of the art for the factorization of Cunningham and other numbers of a similar special form.

Relations in the special number field sieve. A relation in the original variants of the number field sieve [72, 55] is a higher degree variation of the homomorphic equivalence (5.20) on page 29 encountered in the Gaussian integer method in Section 5.5.1 on page 24, with the prime q replaced by the composite n to be factored: when Relation (5.20) is divided by its left-hand side, an equation similar to the one in Condition (5.1) on page 3 is obtained (with $w = v = 1$):

$$1 \equiv \Big(\prod_{p \le B} p^{-e_{a,b,p}}\Big)\Big(\prod_{\mathfrak{u} \in \mathfrak{U}} \varphi(\mathfrak{u})^{e_{a,b,\mathfrak{u}}}\Big)\Big(\prod_{\mathfrak{p} \in \mathfrak{P}} \varphi(\mathfrak{g}_\mathfrak{p})^{e_{a,b,\mathfrak{p}}}\Big) \bmod n.$$

With $\pi(B) + |\mathfrak{U}| + |\mathfrak{P}| + \omega$ such equations (with ω the oversquareness as in Section 5.2.1 on page 2) the composite n can most likely be factored.

As set forth in Section 5.5.1, Relation (5.20) requires simultaneous smoothness, for coprime integers a, b, of the integers $a - bm$ and $\mathbf{N}(a - b\alpha) = b^d f(\frac{a}{b})$ where $f(m) \equiv 0 \bmod n$ for an irreducible, degree d polynomial $f(X) \in \mathbf{Z}[X]$ with $f(\alpha) = 0$. Because $|a|$ and $|b|$ will be relatively small, the smoothness probabilities depend on the size of m, the degree d, and the sizes of the coefficients of $f(X)$. In the Gaussian integer method the size-issue was addressed by replacing m by $\frac{y}{z}$ for smaller y and z, and by taking $f(X) = X^2 - t$ for small $|t|$. In the special number field sieve it is done by considering a similar polynomial of degree larger than two, and by considering only specific n-values.

Special numbers. Pollard in [72] targeted composites of the form $x^3 - k$ for small $|k|$. In [55] this was generalized to $x^D - k$ for small positive integers D and $|k|$. Examples of such composites are the Cunningham numbers, with factorizations tabulated in [28, 12] and to the present day the subject of intense computations. Given a composite $n = x^D - k$ and a targeted degree d, a polynomial $f(X) = X^d - t \in \mathbf{Z}[X]$ and integer m with $f(m) \equiv 0 \bmod n$ are easily found by taking the smallest integer ℓ such that $\ell d \ge D$ and putting $t = kx^{\ell d - D}$ and $m = x^\ell$.

The choice of d leads to a trade-off between the smoothness probabilities of $a - bm$ and $b^d f(\frac{a}{b})$, with larger d leading to smaller $a - bm$ but faster growth

of $|b^d f(\frac{a}{b})|$ for larger a- and b-values. This trade-off is analyzed further below. It leads to a d-value that grows as a function of n and smoothness probabilities of $a - bm$ and $b^d f(\frac{a}{b})$ that are better balanced than in the Gaussian integers method.

With $f(X) = X^d - t$, deriving Relation (5.20) on page 29 from a pair $(a - bm, b^d f(\frac{a}{b}))$ of smooth integers works as described in Section 5.5.1 on page 24, assuming that $\mathbf{Z}[\alpha]$ is a unique factorization domain. If the latter is not the case the general number field sieve approach (cf. below) may be used while still exploiting the favorable smoothness probabilities resulting from the polynomial $X^d - t$ (compared to the polynomials that normally occur in the general number field sieve). Alternatively, as suggested in [55, Section 3] and as done for several of the factorizations obtained in [55], the search-based approach from Section 5.5.1 can still be used, but with $\mathbf{Z}[\alpha]$ replaced by $\frac{1}{c}\mathbf{Z}[\alpha]$ for an appropriately chosen small $c \in \mathbf{Z}_{>1}$ (and with φ redefined as $\varphi(\frac{a-b\alpha}{c}) = (a - bm)(c^{-1} \bmod n) \bmod n$).

Finding relations. Pairs of coprime integers a, b (with $b \geq 0$) such that $a - bm$ and $b^d f(\frac{a}{b}) = a^d - tb^d$ are both smooth are normally found using a two-stage sieving process. Commonly, everything related to $a - bm$ is referred to as the *rational side*, and everything related to $b^d f(\frac{a}{b})$ as the *algebraic side*. With the notation from Section 5.2.3 on page 5 let $L[\frac{r}{2}, \beta]$ be the smoothness bound (without loss of generality shared for the smoothness of $a - bm$ and of $b^d f(\frac{a}{b})$), where $r > 0$ and $\beta > 0$ are specified in the analysis below. Furthermore, assume that coprime pairs of integers a, b with $|a| \leq L[\frac{r}{2}, \gamma_a]$ and $0 \leq b \leq L[\frac{r}{2}, \gamma_b]$ must be considered, with, without loss of generality, $\gamma_a \geq \gamma_b$ and $\gamma_a + \gamma_b = 2\beta$ (cf. Section 5.2.3 and the analysis below). Under this standard assumption on the size $L[\frac{r}{2}, \gamma_a + \gamma_b] = L[\frac{r}{2}, 2\beta]$ of the search space versus the smoothness bound $L[\frac{r}{2}, \beta]$, the sieving effort equals $L[\frac{r}{2}, 2\beta]$ for both methods sketched below.

Line sieving. The first method to find relations, as used by the earliest implementations, is *line sieving*. Pollard in [72] used it in a first stage to locate pairs of coprime integers a, b for which $a - bm$ is smooth, and then, in the second stage, used trial division to inspect the corresponding values of $b^d f(\frac{a}{b})$ for smoothness. In [55] line sieving was used in both stages, i.e., first to find pairs of coprime integers a, b for which $a - bm$ is (likely to be) smooth and next to find the pairs for which $b^d f(\frac{a}{b})$ is smooth as well. This gave the number field sieve its name. In line sieving, for $b = 0, 1, 2, \ldots, L[\frac{r}{2}, \gamma_b]$ in succession the entire *line* of a-values with $|a| \leq L[\frac{r}{2}, \gamma_a]$ is sieved. This is similar to Schroeppel's linear sieve where consecutively for each fixed j-value the interval of i-values is processed.

The elementary line siever from [55] was used for, among others, the factorization of F_9 reported in [56] and, after considerable improvements by Mont-

gomery, for many other factorizations (see for instance [10]). The order of the two sieving stages in the special number field sieve is explained by the fact that on average the absolute values $|a - bm|$ on the rational side are larger than the absolute values $|b^d f(\frac{a}{b})|$ on the algebraic side: compared to the reverse order of sieving, fewer candidate locations resulting from the first sieve over the $|a - bm|$-values remain to be inspected after the second sieve over the $|b^d f(\frac{a}{b})|$-values. In the general number field sieve $|b^d f(\frac{a}{b})|$ is on average larger than $|a - bm|$ so that it becomes more efficient to reverse the order of the sieving stages: there pairs of integers a, b for which $b^d f(\frac{a}{b})$ is likely to be smooth are located first, and next the resulting set of pairs is further restricted to pairs for which $a - bm$ is smooth as well.

Lattice sieving. Unless γ_b is small, it is more efficient to use *lattice sieving*, as suggested by Pollard in [73]. Lattice sieving was used for the first time in [38] (for the general number field sieve), and has from that time on been used for all record factorizations obtained using the special or the general number field sieve (for the record reported in [19] both line sieving and lattice sieving were used). In lattice sieving relation collection is split up, not according to disjoint lines as in line sieving, but into non-disjoint subtasks specified by (prime, root) pairs (q, z) for primes q relatively close to the smoothness bound $L[\frac{r}{2}, \beta]$. The prime q is often referred to as *special prime* or *special q-prime*. This has nothing to do with the "special" in special number field sieve; the prime q, however, is reminiscent of how Davis, Holdridge and Simmons managed to get quadratic sieve to work, cf. Section 5.4.3 on page 18. In subtask (q, z) relations are sought specified by pairs of coprime integers a, b for which $\frac{a}{b} \equiv z \bmod q$. Because a relation given by a pair of integers a, b may be found by different subtasks (q, z) and $(\bar{q}, \bar{z})$ if $\frac{a}{b} \equiv z \bmod q$ and $\frac{a}{b} \equiv \bar{z} \bmod \bar{q}$, duplicates among the relations must be removed. There are on the order of $L[\frac{r}{2}, \beta]$ (prime, root) pairs for which the prime is close to $L[\frac{r}{2}, \beta]$, each of which is processed (typically in parallel, in ranges of sequential q-values) until enough distinct relations have been found. It follows that per (prime, root) pair effort at most $L[\frac{r}{2}, \beta]$ may be spent. How this is achieved is described below.

Let (q, z) with q of order $L[\frac{r}{2}, \beta]$ be a fixed (prime, root) pair. In the special number field sieve (q, z) is typically chosen such that $z \equiv m \bmod q$ and subtask (q, z) results in pairs of coprime integers a, b with q dividing $a - bm$; in the general number field sieve (q, z) is chosen such that $f(z) \equiv 0 \bmod q$ so that q divides $b^d f(\frac{a}{b})$ for the pairs a, b resulting from subtask (q, z). Without loss of generality, assume that $z \equiv m \bmod q$ and that $\gamma_a = \gamma_b = \beta$. The pairs of integers a, b for which q divides $a - bm$ form an index-q sublattice of $\mathbf{Z}^2$ with basis $\{\binom{q}{0}, \binom{z}{1}\}$ over $\mathbf{Z}$: in subtask (q, z) only elements of this sublattice are considered. Intersecting the sublattice with the original rectangular search

space $\{(a,b) : |a| \leq L[\frac{r}{2},\beta], 0 \leq b \leq L[\frac{r}{2},\beta]\}$ results in a search space for subtask (q,z) that consists of approximately $\frac{2L[\frac{r}{2},2\beta]}{q} = L[\frac{r}{2},\beta]$ elements. This intersection is not calculated precisely, but only approximated in the sense that a subtask search space is defined that should be approximately as effective as the actual intersection: first a reduced basis $\{u,v\} \subset \mathbf{Z}^2$ of the original basis $\{\binom{q}{0},\binom{z}{1}\}$ is found (i.e., the vectors u and v should have entries that are in absolute value close to $\sqrt{q}$) after which the intersection is approximated as $\{\binom{a}{b} = iu + jv \in \mathbf{Z}^2 : |i| \leq L[\frac{r}{2},\frac{\beta}{2}], 0 \leq j \leq L[\frac{r}{2},\frac{\beta}{2}]\}$. The subtask search space is then defined as the rectangle $\{(i,j) : |i| \leq L[\frac{r}{2},\frac{\beta}{2}], 0 \leq j \leq L[\frac{r}{2},\frac{\beta}{2}]\}$ in the (i,j)-plane, with each pair (i,j) identified with the a,b pair $\binom{a}{b} = iu + jv$ with $\frac{a}{b} \equiv z \bmod q$.

Sieving by vectors. The above new rectangle of size $2L[\frac{r}{2},\frac{\beta}{2}] \times L[\frac{r}{2},\frac{\beta}{2}] = L[\frac{r}{2},\beta]$ in the (i,j)-plane must be sieved with all $L[\frac{r}{2},\beta]$ distinct (prime, root) pairs (p,z_p) while spending effort $L[\frac{r}{2},\beta]$, i.e., proportional to the size of the subtask search space. This implies that line sieving can not be used because it would consider all $L[\frac{r}{2},\frac{\beta}{2}]$ consecutive j-values (i.e., all lines in the new rectangle; cf. [13]) and it would do so for for each of the $L[\frac{r}{2},\beta]$ (prime, root) pairs (p,z_p). Thus, line sieving would take effort at least $L[\frac{r}{2},\frac{3}{2}\beta]$, which is more than $L[\frac{r}{2},\beta]$. Instead, in the (i,j)-plane sieving with a (prime, root) pair (p,z_p) must be done in such a way that it takes effort at most $\frac{L[\frac{r}{2},\beta]}{p}$; summation over all (prime, root) pairs (p,z_p) then results in an upper bound $L[\frac{r}{2},\beta]$ on the total sieving effort (cf. Equation (5.4) on page 7).

As above, the points to be visited per pair (p,z_p) belong to an index-p sublattice of the (i,j)-plane. Those among them that belong to the new rectangle in the (i,j)-plane are located in a manner similar to how that new rectangle was defined: first a suitably reduced basis is determined for the index-p sublattice induced by (p,z_p) in the (i,j)-plane, after which the intersection with the rectangle can be determined. Pollard in [73] refers to this approach as *sieving by vectors* and poses the problem how to quickly generate the points in the intersection. It was done crudely but fairly effectively in [9, 38] by considering small linear combinations of the vectors spanning the reduced bases; refer to [36], however, for the solution to Pollard's problem.

Speedup obtained by lattice sieving. When using lattice sieving with special q-primes between B_0 and B_1 close to the smoothness bound $L[\frac{r}{2},\beta]$, a fraction $\approx \log\frac{\log B_1}{\log B_0}$ of the original search space is considered. The precise values depend on how much sieving (or over-sieving) one decides to do, but normally speaking the fraction will be considerably less than one and far outweighs the overhead inherent in sieving by vectors (as the latter requires a basis reduction step for each (prime, root) pair (p,z_p) that must be sieved with). Another nega-

tive effect is that relations for which $a - bm$ is $(B_0 - 1)$-smooth will be missed. Overall, however, for large composites lattice sieving is to be preferred to line sieving. It should be noted that when sieving the values that have a special prime q as a fixed divisor, sieving is normally restricted to (prime, root) pair (p, z_p) for which the prime p is less than q.

Free relations. With P and $\mathfrak{P}$ as in Section 5.5.1 on page 24, if during the construction of $\mathfrak{P}$ a prime $p \in P$ is encountered such that $f(X)$ splits into linear factors modulo p, a *free relation* is obtained. Let $f(X) = \prod_z (X - z)^{e_z} \bmod p$ for distinct integers $z \in \{0, 1, \ldots, p-1\}$ and strictly positive integers e_z. For each of these integers z, define $\mathfrak{p}_{p,z}$ as the first degree prime ideal of norm p generated by p and $z - \alpha$. Then the ideal generated by p equals the product of the ideals $\mathfrak{p}_{p,z}^{e_z}$. With $\mathfrak{P}_p$ the set containing all these ideals $\mathfrak{p}_{p,z}$ it follows that $(p) = \prod_{\mathfrak{p}_{p,z} \in \mathfrak{P}_p} \mathfrak{p}_{p,z}^{e_z}$ which is, with $a = p$, $b = 0$, and $e_{p,0,\mathfrak{p}_{p,z}} = e_z$, an equation of the same form as Equation (5.17) on page 27. This leads to a useful relation because $p \in P$ and $\mathfrak{P}_p \subset \mathfrak{P}$. The fraction of relations that thus comes for free is inversely proportional to the degree of the splitting field of $f(X)$.

Heuristic asymptotic analysis of the special number field sieve. In the analysis below the second argument used in the L-notation introduced in Section 5.2.2 on page 4 often involves an $o(1)$-term, for $D \to \infty$ where $n = x^D - k$; this term is silently ignored. Let $r, \psi_r \in \mathbf{R}_{>0}$ be such that $\max(|a - bm|, |b^d f(\frac{a}{b})|) \leq L[r, \psi_r]$, and let $s, \beta \in \mathbf{R}_{>0}$ be such that the largest of the two smoothness bounds is upper bounded by $L[s, \beta]$ (zero arguments can be seen not to work). Thus, it suffices to find $L[s,\beta] + L[s,\beta] = L[s,\beta]$ coprime (a, b) pairs that satisfy the smoothness requirements. Furthermore, dependencies must be found in a sparse $L[s,\beta] \times L[s,\beta]$-matrix, at cost $L[s, 2\beta]$ (cf. Section 5.2.5 on page 7).

With the smoothness probabilities from Section 5.2.2 and heuristically assuming that the values $a - bm$ and $b^d f(\frac{a}{b})$ behave as independent random integers, it is expected that to find a single satisfactory coprime (a, b) pair, it suffices to consider $L[r - s, \psi_s]$ random pairs, for some $\psi_s \in \mathbf{R}_{>0}$. Because $L[s,\beta]$ pairs suffice, at most $L[s,\beta]L[r - s, \psi_s]$ pairs have to be inspected, which is minimized for $s = \frac{r}{2}$ (cf. this repeats the argument given just before Expression (5.3) on page 5).

The trade-off between the smoothness probabilities of $a - bm$ and $b^d f(\frac{a}{b})$ now determines the values for r and the degree d. It follows from $\max(|a - bm|, |b^d f(\frac{a}{b})|) \leq L[r, \psi_r]$ that $m \leq L[r, \psi]$ for some $\psi \in \mathbf{R}_{>0}$. With $m \approx n^{\frac{1}{d}} = e^{\frac{1}{d} \log n}$, this bound on m implies that $d \approx \delta(\frac{\log n}{\log \log n})^{1-r}$, where $\delta = \frac{1}{\psi}$. With a search space that contains $L[\frac{r}{2}, \beta + \psi_s]$ pairs (a, b) and given the symmetry of $|a|$ and b in $|b^d f(\frac{a}{b})|$, both $|a|$ and b may be upper bounded by $L[\frac{r}{2}, \gamma]$ for

some $\gamma \geq \frac{\beta+\psi_s}{2}$, so that $\max(|a|, b)^d = L[1-\frac{r}{2}, \gamma\delta]$. Balancing the upper bounds for $|a-bm|$ and $|b^d f(\frac{a}{b})|$, leads to the optimal choice $r = 1-\frac{r}{2}$, and thus $r = \frac{2}{3}$.

In terms of the L-notation, no savings can be obtained when different smoothness bounds are used for $|a-bm|$ and $|b^d f(\frac{a}{b})|$, so let $L[\frac{1}{3}, \beta]$ be the smoothness bound for both. With $|a|$ and b both bounded by $L[\frac{1}{3}, \gamma]$ and m by $L[\frac{2}{3}, \psi]$ and heuristically assuming random behavior and independence, the values $a-bm$ and $|b^d f(\frac{a}{b})|$ are both $L[\frac{1}{3}, \beta]$-smooth with probability $L[\frac{1}{3}, -\frac{\psi}{3\beta}]L[\frac{1}{3}, -\frac{\gamma\delta}{3\beta}]$, so that a total of $L[\frac{1}{3}, \beta + \frac{\psi+\gamma\delta}{3\beta}]$ pairs must be inspected to find $L[\frac{1}{3}, \beta]$ satisfactory ones (and $\psi_s = \frac{\psi+\gamma\delta}{3\beta}$). This is minimized when $3\beta^2 = \psi + \gamma\delta$ and thus results, with effort $L[\frac{1}{3}, 0]$ per smoothness test (cf. Section 5.2.4 on page 6), in effort $L[\frac{1}{3}, 2\max(\beta, \gamma)]$ to find the required (a, b) pairs. Taking $\gamma = \beta$ and noting that this satisfies all the above boundary conditions, and including the cost $L[\frac{1}{3}, 2\beta]$ to find the dependencies, it follows that the overall effort is $L[\frac{1}{3}, 2\beta]$, which remains to be minimized under the condition $3\psi\beta^2 - \beta - \psi^2 = 0$. The single positive root $\beta = \frac{1}{6\psi}(1 + \sqrt{1+12\psi^3})$ attains its minimal value $\beta = (\frac{2}{3})^{\frac{2}{3}}$ for $\psi = (\frac{2}{3})^{\frac{1}{3}}$ (and thus $\delta = (\frac{3}{2})^{\frac{1}{3}}$). The resulting overall effort is $L[\frac{1}{3}, (\frac{32}{9})^{\frac{1}{3}}]$.

More general polynomials with constant coefficients. The above analysis of the relation collection and linear algebra effort applies for $n \to \infty$ as long as the absolute values of the coefficients of the polynomial $f(X)$ are bounded by a constant. Even though the algorithm as described in this section may not apply to such more general polynomials (because the search for generators of the first degree prime ideals may fail) one nevertheless says that the special number field sieve applies to n-values that admit polynomials with coefficients bounded by a constant. For these somewhat more general n-values the search for generators may be replaced by the more general approach used for the general number field sieve and described in Section 5.5.3 below.

Large prime relations. Relations involving large primes play a much more prominent role in the number field sieve than in earlier general purpose factoring methods, because large primes can relatively easily be found on the rational side (i.e., large primes dividing $a-bm$) and on the algebraic side (i.e., large primes dividing $b^d f(\frac{a}{b})$). Depending on the number of large primes allowed, the number of pairs to be inspected after the sieving may increase considerably, resulting in relatively costly cofactor processing (for which other factoring algorithms, including the elliptic curve method and quadratic sieve, turn out to be useful). The presence of large primes also complicates the linear algebra step (cf. the discussion in Section 5.2.6 on page 8 on filtering and Montgomery's contributions to it) and even deciding if enough relations have been collected becomes a more cumbersome process. Overall, however, usage of large primes leads to a considerable speedup (which, as usual, disappears in the $o(1)$ in the

L-notation). Refer to [34] for the earliest results (which were, back then, found to be rather surprising) and to [62, 45] for the most recent ones.

5.5.3 General number field sieve

Though it had not escaped at least one of the authors of [55] that, if a number of obstructions are ignored, an approach and analysis similar to the special number field sieve could apply to arbitrary composites, Joe Buhler and Pomerance were the first who dared to publicly suggest this. Their optimism turned out to be justified: after several obstacles had been resolved, the *general number field sieve* became a reality in the early 1990s. As a result (as shown below and with the usual vigorous handwaving) the expected general purpose factoring effort was reduced, quite spectacularly, from $L[\frac{1}{2}, 1]$ to $L[\frac{1}{3}, (\frac{64}{9})^{\frac{1}{3}}] \approx L[\frac{1}{3}, 1.9223]$, for $n \to \infty$. This is a bit worse than the special number field sieve's $L[\frac{1}{3}, (\frac{32}{9})^{\frac{1}{3}}] \approx L[\frac{1}{3}, 1.5263]$ but not overly so. Given the proven practicality of the special number field sieve, some expected that its generalization would soon turn out to be practical as well – and quite possibly replace quadratic sieve as the best practical general purpose factoring method.

Despite the encouraging remarks in [14, Section 1], this expectation was not generally shared. Initial experiments were indeed hardly encouraging. In [9] a 66-digit general number was factored (using lattice sieving, cf. Section 5.5.2 on page 30) in a few hours on a MasPar supercomputer (cf. Section 5.4.4 on page 19), where quadratic sieve took only a few minutes. This compares very poorly to the performance of the special number field sieve, which had been used to obtain the record factorization of F_9, an achievement that was far beyond the capacity of quadratic sieve. Neither were the results reported in [13] competitive, but it is not clear if sieving by vectors was used in the lattice sieving from [13] (as required to get the right performance). The first more encouraging estimate appeared in [32, Section 1], confirmed by an experiment reported in [38, Section 5] suggesting that the 129-digit quadratic sieve factorization reported in [6], at that point in time the state of the art in general purpose factoring, could have been achieved at about a third of the effort using the general number field sieve.

In 1996 the general number field sieve finally replaced quadratic sieve as the state-of-the-art general purpose factoring method for non-special numbers as well: the factorization of a 130-digit general composite took an effort that was, according to [27], "a fraction of what was spent on the previous record" (of the 129-digit composite in [6]), and used the advantageous effect, as had already been reported in [34], of the use of multiple large primes on both the rational

and algebraic side. Probably the most prominent factorization achieved with the general number field sieve is still the 1999 factorization of a 512-bit cryptographic modulus, in [19]. For 512-bit numbers it thus took almost a decade to close the gap between "special" and "general". The latter factorization required a 500-fold larger effort than the former, so this gap was not entirely closed by Moore's law. The current general number field sieve factorization record stands at 768 bits [44][1]. There is a 400-fold effort gap between the current special number field sieve record (which stands at about 1200 bits) and a general 1024-bit composite (the factorization of which could have practical implications). Actually closing this gap using current methods would result in a power-bill that can not – or hardly – be justified by the importance of the resulting factorization: it would be preferable to have a significantly improved method before embarking on a general 1024-bit factorization. Unfortunately, however, factoring developments over the last two decades have been disappointing. Thus, it seems there comes no end to the number field sieve's "day in the sun" [14, Section 1]: true progress in general purpose factoring has come to a standstill since the publication of [14] and [22]. The sole exception is [86], but as it relies on the as yet uncertain realization of quantum computing it has no practical implications, yet.

Polynomial selection. Finding a suitable polynomial for arbitrary n is easy; finding a good polynomial is much harder and figuring out how to actually use it to factor n is yet another story (part of which is told below). Indeed, for any composite n and $d \in \mathbf{Z}_{>0}$, any integer m close to but less than $n^{\frac{1}{d}}$ may be chosen, after which $f(X)$ may be defined as $\sum_{i=0}^{d} f_i X^i$ where $n = \sum_{i=0}^{d} f_i m^i$ is the base m representation of n (i.e., $f_i \in \mathbf{Z}$ and $0 \leq f_i < m$, for $0 \leq i \leq d$). If luck has it that the resulting $f(X)$ is not irreducible (this has not happened yet in practice), a factorization of n may follow right away. The order $m \approx n^{\frac{1}{d}}$ estimate of the coefficients of $f(X)$ (as used in the analysis below) gives only a rough impression of the relative performance of a particular choice. Initially mostly due to the efforts by Montgomery, selecting and distinguishing more effective parameters for the number field sieve has grown into an active area of research, to which Chapter 6 is devoted.

No matter how carefully a polynomial $f(X)$ has been selected, however, the rough estimate $m \approx n^{\frac{1}{d}}$ for its coefficients is inescapable, generally speaking. This leads, unavoidably, to a rather ill-behaved number field $\mathbf{Q}(\alpha)$ where the approach sketched in Section 5.5.1 on page 24 meets with a number of obstructions that looked, in the late 1980s, hard to overcome. For instance, although obtaining the prime ideal factorization of the ideal $(a - b\alpha)$ as in Equation (5.17)

[1] The current general number field sieve record for the computation of discrete logarithms over prime fields also stands at 768 bits [46]

on page 27 is still possible, turning it into Equation (5.19) on page 28 (as required to obtain Relation (5.20) on page 29) requires finding generators in $\mathbf{Z}[\alpha]$ (or in $\frac{1}{c}\mathbf{Z}[\alpha]$, for some integer c) for the units and the prime ideals in $\mathfrak{P}$. For general number fields – as may be expected given a defining polynomial with coefficients of order $n^{\frac{1}{d}}$ – it is not even feasible to write down such generators [55, Section 9], let alone find them (and the primitive search described in Section 5.5.1 would most certainly be inadequate).

While joint efforts were underway to remove the obstructions, which seemed possible but cumbersome [14], Leonard Adleman proposed an elegant and deceptively simple solution in [2]. This led to the approach sketched below.

Relations in the general number field sieve. In the general number field sieve relations are given by coprime pairs of integers a, b for which the integers $a - bm$ and $b^d f(\frac{a}{b})$ are both smooth, just as in the special number field sieve. In the latter, relations are turned into identities modulo n between two products by applying φ to the relations themselves. Sufficiently many modular identities can then be combined into a single identity modulo n between two squares: an integer square on the left-hand side with on the right-hand side the square of the product of a (large) number of φ-values of elements of $\mathbf{Z}[\alpha]$. This approach requires turning Equation (5.17), for each pair a, b under consideration, into something with a right-hand side to which φ can be applied, such as Equation (5.19). As mentioned above and as shown in [55] that works if the polynomial $f(X)$ defining the number field has a particularly nice form, but as elaborated upon in [14] (and mentioned above) it is problematic for general $f(X)$. As discussed in [14] there are several ways to overcome this problem, the most convenient one of which is using Adleman's quadratic characters.

Remark. More general descriptions of the general number field sieve no longer refer to a rational side (for $a - bm$) and an algebraic side (for $b^d f(\frac{a}{b})$) but replace $a - bm$ by $b^{\bar{d}} g(\frac{a}{b})$ for a polynomial $g(X) \in \mathbf{Z}[X]$ of degree $\bar{d} \geq 1$ that has modulo n a root m in common with $f(X)$. The methods described in this chapter apply to this more general situation as well. Refer to Chapter 6 for a discussion on more general pairs of polynomials.

Quadratic characters. In [2] Adleman proposed to construct the above identity modulo n between two squares in a different manner: an integer square on the left-hand side, as above, but on the right-hand side the square of the φ-value of an element of $\mathbf{Z}[\alpha]$. In this way the application of φ is postponed as long as possible, and everything "on the right-hand side" stays in $\mathbf{Z}[\alpha]$ until the last moment. To get this to work in a naive fashion, sets S of pairs of coprime integers a, b would have to be found such that $\prod_{(a,b)\in S}(a - bm) \in \mathbf{Z}$ is the square of some $x \in \mathbf{Z}$, and such that $\eta = \prod_{(a,b)\in S}(a - b\alpha) \in \mathbf{Z}[\alpha]$ is a square so that $\sqrt{\eta} \in \mathbf{Z}[\alpha]$ can be computed; the required modular identity

$x^2 \equiv y^2 \bmod n$ would then follow with $y = \varphi(\sqrt{\eta})$. Unfortunately, this does not work, due to the fourth obstruction listed in [14, Section 6], namely that $\sqrt{\eta}$ does not necessarily belong to $\mathbf{Z}[\alpha]$. But, as also shown in [14, Section 6], this can easily be fixed: with $f'(X)$ the derivative of the polynomial $f(X)$ it is the case that $f'(\alpha)\sqrt{\eta}$ belongs to $\mathbf{Z}[\alpha]$ and the modular identity becomes $(f'(m)x)^2 \equiv \varphi(f'(\alpha)\sqrt{\eta})^2 \bmod n$.

The condition on $\prod_{(a,b)\in S}(a - bm)$ is equivalent to a dependency modulo 2 among the exponent vectors of the factorizations of the smooth values $a - bm$, as usual. The condition on $\eta = \prod_{(a,b)\in S}(a - b\alpha)$ is only slightly more involved. In the first place, if $f'(\alpha)^2\eta$ is a square in $\mathbf{Z}[\alpha]$ the sum of exponent vectors $(e_{a,b,\mathfrak{p}})_{\mathfrak{p}\in\mathfrak{P}}$ as in Equation (5.17) on page 27 is a vector with all even entries. This condition is equivalent to the usual dependency modulo 2 among the vectors $(e_{a,b,\mathfrak{p}})_{\mathfrak{p}\in\mathfrak{P}}$ (which is a stronger condition than just $\prod_{(a,b)\in S} b^d f(\frac{a}{b})$ being a square). Furthermore, if $f'(\alpha)^2\eta$ is a square in $\mathbf{Z}[\alpha]$ then $\prod_{(a,b)\in S}(a - bz_q)$ is a square modulo q, for any prime, root pair (q, z_q) with q prime and $f(z_q) \equiv 0 \bmod q$ [14, Section 8]. But, these are only necessary conditions for $f'(\alpha)^2\eta$ to be a square in $\mathbf{Z}[\alpha]$.

As shown in [2], an effective version of the converse is true too: $f'(\alpha)^2\eta$ is most likely a square in $\mathbf{Z}[\alpha]$ if the vectors $(e_{a,b,\mathfrak{p}})_{\mathfrak{p}\in\mathfrak{P}}$ are dependent modulo 2, and if $\prod_{(a,b)\in S}(a - bz_\mathfrak{q})$ is a square modulo $\mathfrak{q}$ for sufficiently many $(\mathfrak{q}, z_\mathfrak{q})$ pairs as above for which $f'(z_\mathfrak{q}) \not\equiv 0 \bmod \mathfrak{q}$ and for which the first degree prime ideal generated by $\mathfrak{q}$ and $z_\mathfrak{q} - \alpha$ does not belong to $\mathfrak{P}$. Refer to [14, Section 8] for the number of prime, root pairs that suffices in theory; in practice one commonly uses 64 or 128 pairs. To enforce the condition that $\prod_{(a,b)\in S}(a - bz_\mathfrak{q})$ is a square modulo $\mathfrak{q}$, for each $(\mathfrak{q}, z_\mathfrak{q})$-pair each vector $(e_{a,b,\mathfrak{p}})_{\mathfrak{p}\in\mathfrak{P}}$ includes an additional bit with value zero if $a - bz_\mathfrak{q}$ is a square modulo q and with value one otherwise. It remains to compute $f'(\alpha)\sqrt{\eta} \in \mathbf{Z}[\alpha]$.

Computing square roots in the number field sieve. Let $\eta = \prod_{(a,b)\in S}(a - b\alpha) \in \mathbf{Z}[\alpha]$ be an element of known smooth norm for which it is known that $f'(\alpha)^2\eta$ is the square of an element of $\mathbf{Z}[\alpha]$. Several methods have been proposed to compute the latter element $f'(\alpha)\sqrt{\eta} \in \mathbf{Z}[\alpha]$ or just $\varphi(f'(\alpha)\sqrt{\eta}) \in \mathbf{Z}/n\mathbf{Z}$ (which would suffice for the present application); refer to [89] for a recent discussion.

A direct approach would be to calculate the quadratic polynomial $X^2 - f'(\alpha)^2\eta \in \mathbf{Z}[\alpha][X]$ and to factor it over $\mathbf{Q}(\alpha)$ using a standard (polynomial-time) method to do so; refer to [14, Section 9] and [89] for references and an extensive discussion of this method. Back in the early 1990s it was deemed to be infeasible, but at this point in time it enjoys renewed interest, simply because these days symbolic algebra packages seem to be able to handle the resulting problems (involving rather large coefficients) without too much trouble. If it works, it is certainly quite convenient.

The first method to be used in practice (in [9]) was due to Jean-Marc Couveignes [26]. It requires d to be odd, is based on the use of Chinese remaindering, and produces $\varphi(f'(\alpha)\sqrt{\eta}) \in \mathbf{Z}/n\mathbf{Z}$. Let Q be a product of distinct primes such that $\frac{Q}{2}$ bounds the absolute values of the integer coefficients of $f'(\alpha)\sqrt{\eta}$ and such that $f(X)$ remains irreducible modulo each q dividing Q. Here it is assumed that such a Q can be found, but see [26] and [14, Section 9]. For any prime q dividing Q it is the case that $(\mathbf{Z}/q\mathbf{Z})[X]/(f(X))$ is isomorphic to the finite field $\mathbf{F}_{q^d}$ of q^d elements, so that $\pm f'(\alpha)\sqrt{\eta} \bmod q$ can easily be computed in $\mathbf{F}_{q^d}$. The root $f'(\alpha)\sqrt{\eta} \in \mathbf{Z}[\alpha]$ can then be computed by combining (using Chinese remaindering) the roots modulo all primes q dividing Q, where the sign-ambiguity (i.e., which of the two choices modulo q to use) is resolved using norm-calculations and the fact that d is odd. As shown in [26] (and used in [9]) the calculation of $f'(\alpha)\sqrt{\eta} \in \mathbf{Z}[\alpha]$ (with huge coefficients, in absolute value only bounded by $\frac{Q}{2}$) can be avoided in a neat way and $\varphi(f'(\alpha)\sqrt{\eta}) \in \mathbf{Z}/n\mathbf{Z}$ can be calculated directly without requiring arithmetic with numbers larger than n^2. Although it is conceptually quite simple and allows (to a large extent) parallelization, the disadvantage of Couveignes' method is that it works only for odd d and, more importantly, that the effort involved grows quadratically with the number of primes dividing Q.

Montgomery's square root method. Both disadvantages were addressed by the method proposed in 1994 by Montgomery in [63]. Since its initial development it has not led to any new insights or algorithms, because it is perfectly satisfactory as is: currently it is still the method of choice for practical applications (but see also the combination of the direct approach and Couveignes' method in [89, Section 4]).

A proper description of the method can be found in Montgomery's own paper [63] and in [66]. Here the following rough description suffices.

Let $\eta = \prod_{(a,b)\in S}(a - b\alpha) \in \mathbf{Z}[\alpha]$ be such that $f'(\alpha)\sqrt{\eta} \in \mathbf{Z}[\alpha]$, as above, and thus the ideal generated by η equals a product $\prod_{\mathfrak{p}\in\mathfrak{P}} \mathfrak{p}^{2e_\mathfrak{p}}$ (for integers $e_\mathfrak{p}$) of squared first degree prime ideals, with $\mathfrak{P}$ as in Section 5.5.1 on page 24. Montgomery's square root is an iterative process that builds the desired square root in $\mathbf{Z}[\alpha]$ by patiently – and measurably – chipping away parts of η. Initialize the square-root-to-be $\varsigma \in \mathbf{Z}[\alpha]$ as one. The basic idea is to remove a product of some of the squared prime ideals from η, find a generator in $\mathbf{Z}[\alpha]$ of an ideal contained in the product of the (non-squared) ideals, to multiply the square-root-to-be ς by that generator, and to iterate until η has become small enough to further compute its square root directly. This works, except that the newly found generator may contain a factor not contained in the product of ideals, so per iteration η may have to be corrected by the square of the inverse of that spurious factor. To make this correction step less cumbersome, η is con-

structed in a different way (though equivalent from the point of view of the linear algebra), namely as a quotient of two similar products as above, with approximately equal norms in the numerator and the denominator:

$$\eta = \frac{\prod_{(a,b)\in S_{\text{num}}}(a - b\alpha)}{\prod_{(a,b)\in S_{\text{den}}}(a - b\alpha)}$$

which leads to

$$\eta = \frac{\prod_{\mathfrak{p}\in\mathfrak{P}_{\text{num}}} \mathfrak{p}^{2e_{\mathfrak{p}}}}{\prod_{\mathfrak{p}\in\mathfrak{P}_{\text{den}}} \mathfrak{p}^{2e_{\mathfrak{p}}}}$$

with $\mathfrak{P}_{\text{num}} \cup \mathfrak{P}_{\text{den}} = \mathfrak{P}$. With $\varsigma_{\text{num}}, \varsigma_{\text{den}} \in \mathbf{Z}[\alpha]$ and a spurious factor $s \in \mathbf{Z}$, all with initial value equal to one, this leads to the following slightly more precise description. If the norm of the ideal $\prod_{\mathfrak{p}\in\mathfrak{P}_{\text{num}}} \mathfrak{p}^{2e_{\mathfrak{p}}}$ is small enough, then compute the square root ς directly and replace ς_{num} by $\varsigma_{\text{num}}\varsigma$. Otherwise, let $\mathfrak{P}'$ be a subset of $\mathfrak{P}_{\text{num}}$ such that the ideal $\mathfrak{I} = \prod_{\mathfrak{p}\in\mathfrak{P}'} \mathfrak{p}^{e_{\mathfrak{p}}}$ has a norm in some targeted interval and such that $\mathfrak{p}_s \in \mathfrak{P}'$ if $s \neq 1$. Identifying $\mathfrak{I}$ with a lattice (for which a basis is easily constructed given the (prime, root) generators of the first degree prime ideals in $\mathfrak{P}'$), a short vector ς in the lattice is found (using, for instance a basis reduction algorithm [52]). The short vector ς can be interpreted as an element of $\mathbf{Z}[\alpha]$ and the ideal (ς) is contained in the ideal $\mathfrak{I}$. To check equality of those two ideals, the spurious factor s is replaced by the quotient of the norms of the ideals (ς) and $\mathfrak{I}$. If $s \neq 1$, then the proper ideal $\mathfrak{p}_s$ of norm s is located (i.e., $\mathfrak{p}_s\mathfrak{I} = (\varsigma)$) and $\mathfrak{P}_{\text{den}}$ is replaced by $\mathfrak{P}_{\text{den}} \cup \mathfrak{p}_s$, with $e_{\mathfrak{p}_s} = 1$. Finally, $\mathfrak{P}_{\text{num}}$ is replaced by $\mathfrak{P}_{\text{num}} - \mathfrak{P}'$ and ς_{num} is replaced by $\varsigma_{\text{num}}\varsigma$. Once ς_{num} has been updated, repeat the process with the roles of $(\mathfrak{P}_{\text{num}}, \varsigma_{\text{num}})$ and $(\mathfrak{P}_{\text{den}}, \varsigma_{\text{den}})$ reversed.

The targeted interval for the norm of $\mathfrak{I}$ (i.e., the choice of $\mathfrak{P}'$) is probably best determined empirically. It has been proved that per iteration the loss (i.e., the spurious factor s) is relatively small compared to the gain (i.e., the norm of $\mathfrak{I}$), and the method requires effort roughly proportional to the size of $\mathfrak{P}$.

Heuristic asymptotic analysis of the general number field sieve. The analysis of the general number field sieve proceeds along the same lines as the analysis of the special number field sieve. The main difference occurs when bounding $|b^d f(\frac{a}{b})|$, which is here bounded by $(d+1)m \max(|a|, b)^d = L[\frac{2}{3}, \psi]L[\frac{1}{3}, \gamma]^d = L[\frac{2}{3}, \psi + \gamma\delta]$. It follows that a total of $L[\frac{1}{3}, \beta + \frac{2\psi+\gamma\delta}{3\beta}]$ pairs (a, b) must be inspected, which is minimized when $3\beta^2 = 2\psi + \gamma\delta$. With $\gamma = \beta$ this leads to the modified quadratic equation $3\psi\beta^2 - \beta - 2\psi^2 = 0$ with a single positive root $\beta = \frac{1}{6\psi}(1 + \sqrt{1 + 24\psi^3})$ which attains its minimal value $\beta = (\frac{8}{9})^{\frac{1}{3}}$ for $\psi = (\frac{1}{3})^{\frac{1}{3}}$. The overall effort becomes $L[\frac{1}{3}, (\frac{64}{9})^{\frac{1}{3}}]$ with $\delta = 3^{\frac{1}{3}}$.

5.5.4 Coppersmith's modifications

Two variants of the general number field sieve were proposed by Coppersmith in [22]. At this point in time (and in the public domain) neither of these methods has proved to be practical yet, though an obvious adaptation of Coppersmith's second method to special numbers was shown to be practical [45].

Using more number fields per composite. The first method lowers the general number field effort from $L[\frac{1}{3}, (\frac{64}{9})^{\frac{1}{3}}] \approx L[\frac{1}{3}, 1.9223]$ to $L[\frac{1}{3}, \frac{(92+26\sqrt{13})^{\frac{1}{3}}}{3}] \approx L[\frac{1}{3}, 1.9019]$ using a conceptually straighforward idea. For any $m \approx n^{\frac{1}{d}}$ and degree d polynomial $f(X) \in \mathbf{Z}[X]$ with $f(m) \equiv 0 \bmod n$ (with coefficients of order m) many similar degree d polynomials can easily be constructed, for instance by adding multiples in $\mathbf{Z}[X]$ of $X - m$ to $f(X)$. Assume that λ such polynomials have been selected, giving rise to λ distinct algebraic numbers fields, say $\mathbf{Q}(\alpha_1)$, $\mathbf{Q}(\alpha_2)$, ..., $\mathbf{Q}(\alpha_\lambda)$. For any coprime pair a, b of integers for which $a - bm$ is smooth there are λ (assumed to be) independent chances for one of the ideals $(a - b\alpha_i)$ to be smooth (as in Equation (5.17) on page 27, with $\mathfrak{P}$ replaced by $\mathfrak{P}_i$). Per smooth $a - bm$, as many distinct vectors will result as there are distinct smooth prime ideals, where the vectors are $(|P| + \sum_{i=1}^{\lambda} |\mathfrak{P}_i|)$-dimensional: $|P|$ coordinates for the exponents on the rational side plus $|\mathfrak{P}_i|$ coordinates for each of the λ number fields, where per relation only one of the λ latter parts contains non-zero entries.

In [22], the optimal λ-value is derived (as $\approx L[\frac{1}{3}, 0.1250]$) along with the smoothness bounds $|P| \approx L[\frac{1}{3}, \frac{1.9019}{2}]$ and $|\mathfrak{P}_i| = \frac{|P|}{\lambda}$ for $1 \leq i \leq \lambda$, which then leads to the effort cited above. Sieving can be used on the rational side to find the pairs for which $a - bm$ is smooth. With $\lambda > 1$ it follows from the relative sizes of the rational and algebraic smoothness bounds that on the algebraic sides sieving has to be replaced by elliptic curve-based smoothness testing (cf. Section 5.2.4 on page 6).

Factorization factory. This method exploits the idea that distinct composites may share a database of smooth values on the rational side. Actually creating such a database could have severe implications because it would reduce the individual factoring effort to $L[\frac{1}{3}, (\frac{20+8\sqrt{6}}{9})^{\frac{1}{3}}] \approx L[\frac{1}{3}, 1.6386]$, which is getting close to the effort required by the special number field sieve. The catch is that this low effort can only be achieved after a preparatory effort $L[\frac{1}{3}, (\frac{12+5\sqrt{6}}{3})^{\frac{1}{3}}] \approx L[\frac{1}{3}, 2.0069]$ to build the database, and that it requires an amount of permanent storage that is proportional to the individual factoring effort: as it refers to storage, this can only be interpreted as staggering.

As above, pairs a, b for which $a - bm$ is smooth may be used for different polynomials, but in the present case the polynomials are targeted at different composites to be factored. Let $m = 2^{\frac{N}{d}}$ for some targeted bit size N, and sup-

pose that in a preparatory sieving step a sufficiently large set S of coprime pairs a, b has been collected for which $a - bm$ is smooth. Any N-bit composite n (which does not have to be known before the preparatory step is carried out) can then be factored by first finding pairs a, b in S for which $b^d f(\frac{a}{b})$ is smooth (for a polynomial $f(X) \in \mathbf{Z}[X]$ with $f(m) \equiv 0 \bmod n$, constructed in the usual manner), after which the linear algebra and square root steps can be carried out in the usual manner. As mentioned in [22] (and analyzed in detail in [31]), optimization leads to matching relatively small rational and algebraic smoothness bounds (both $\approx L[\frac{1}{3}, 0.8193]$, proportional to the square root of the individual factoring effort), but a relatively large rational sieving rectangle (of size $\approx L[\frac{1}{3}, 2.0069]$) to allow collection of sufficiently many smooth values on the rational side. As above, sieving can not be used on the algebraic side.

Refer to [45] for a limited scale application of the factorization factory idea where the roles of the rational and algebraic sides are reversed: two examples are presented of a single special polynomial $f(X)$ that is shared by several Mersenne numbers (for a number of different roots per polynomial).

5.6 Provable methods

This chapter is concluded with a brief description of the relatively poor state of the art in general purpose factoring algorithms that allow a rigorous analysis. None of the rigorous methods below has ever been proved practical.

No general purpose factoring method is known for which the expected asymptotic effort is provably of the form $L[r, \mathfrak{c}]$ for $r < \frac{1}{2}$ (and constant $\mathfrak{c} \in \mathbf{R}_{>0}$). All methods below require effort $L[\frac{1}{2}, \mathfrak{c}]$, for various constants $\mathfrak{c} \in \mathbf{R}_{>0}$, and they all rely on Pomerance's rigorous version of the elliptic curve-based smoothness test from [75] mentioned in Section 5.2.4 on page 6.

So far, the only general purpose factoring method in this chapter for which the analysis does not involve heuristic arguments is Dixon's random squares method from Section 5.3.1 on page 11 with a provable expected asymptotic factoring effort $L[\frac{1}{2}, \sqrt{2}]$. Brigitte Vallée has shown in [90] how to improve Dixon's random squares method by still choosing the random integers v almost uniformly but such that the least absolute remainder $v^2 \bmod n$ is of order only $n^{\frac{2}{3}}$. This results in a provable expected factoring effort $L[\frac{1}{2}, \sqrt{\frac{4}{3}}]$ (cf. Section 5.2.7 on page 11).

Further lowering the effort seems to require using the approach initiated by Martin Seysen in [84]. It replaces Dixon's random integers v by random quadratic forms of negative discriminant $\Delta = -n$, while still using the familiar two-step approach from Section 5.2.1 on page 2. As informally shown in [53,

sections 2.C and 4.10-4.14] smooth forms can be combined (using linear algebra) to produce ambiguous forms, and thereby most likely a factorization of $|\Delta| = n$. Because smoothness of the forms used depends on smoothness of integers of order $\sqrt{n}$, this leads to factoring effort $L[\frac{1}{2}, 1]$ in the usual manner (cf. Section 5.2.7). The generalized Riemann hypothesis can be used to ensure that sufficiently many small primes p exist for which $(\frac{\Delta}{p}) = 1$ (the only ones that can occur in smooth forms), which then leads to a rigorous but conditional effort $L[\frac{1}{2}, 1]$ [51] (see also [53, 4.14]). The dependence on the Riemann hypothesis was removed by Lenstra and Pomerance, in [60]. This is, a quarter of a century later, still the state of the art in provable general purpose factoring.

Acknowledgements

The author thanks Scott Contini, Robert Granger, Herman te Riele, Thorsten Kleinjung, and Richard Schroeppel for their contributions and comments.

Bibliography

[1] L. M. Adleman. A subexponential algorithm for the discrete logarithm problem with applications to cryptography. In *Proceedings of the 20th Annual Symposium on Foundations of Computer Science*, SFCS '79, pages 55–60, Washington, DC, USA, 1979. IEEE Computer Society. (Cited on page 25.)

[2] L. M. Adleman. Factoring numbers using singular integers. In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing, May 5-8, 1991, New Orleans, Louisiana, USA*, pages 64–71, 1991. (Cited on pages 39 and 40.)

[3] L. M. Adleman. The story of sneakers, the movie and Len Adleman the mathematician. URL: `http://www.usc.edu/dept/molecular-science/fm-sneakers.htm`, 1991. (Cited on page 23.)

[4] W. R. Alford and C. Pomerance. Implementing the self-initializing quadratic sieve on a distributed network. In A. J. van der Poorten, I. Shparlinski, and H. G. Zimmer, editors, *Number theoretic and algebraic methods in computer science (Moscow 1993)*, pages 163–174. World Scientific, 1995. (Cited on pages 20 and 22.)

[5] A. Ambainis, Y. Filmus, and F. Le Gall. Fast matrix multiplication: Limitations of the Coppersmith-Winograd method. In R. A. Servedio and R. Rubinfeld, editors, *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 585–593. ACM, 2015. (Cited on page 8.)

[6] D. Atkins, M. Graff, A. K. Lenstra, and P. C. Leyland. The magic words are squeamish ossifrage. In J. Pieprzyk and R. Safavi-Naini, editors, *Advances in Cryptology – ASIACRYPT'94*, volume 917 of *Lecture Notes in Computer Science*, pages 263–277. Springer, Heidelberg, Nov. / Dec. 1995. (Cited on pages 20 and 37.)

[7] E. Bach and J. Shallit. Factoring with cyclotomic polynomials. *Mathematics of Computation*, 52:201–219, 1989. (Cited on page 1.)

[8] R. Barbulescu, P. Gaudry, A. Joux, and E. Thomé. A heuristic quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic. In P. Q. Nguyen and E. Oswald, editors, *Advances in Cryptology – EUROCRYPT 2014*, volume 8441 of *Lecture Notes in Computer Science*, pages 1–16. Springer, Heidelberg, May 2014. (Cited on page 26.)

[9] D. J. Bernstein and A. K. Lenstra. A general number field sieve implementation. pages 103–126 in [54], 1992. (Cited on pages 34, 37, and 41.)

[10] R. P. Brent, P. L. Montgomery, H. J. J. te Riele, H. Boender, M. Elkenbracht-Huizing, R. Silverman, and T. Sosnowski. Factorizations of $a^n \pm 1$, $13 \leq a < 100$: Update 2, 1996. (Cited on page 33.)

[11] R. P. Brent and J. M. Pollard. Factorization of the eighth Fermat number. *Mathematics of Computation*, 36(154):627–630, 1981. (Cited on pages 1 and 14.)

[12] J. Brillhart, D. H. Lehmer, J. L. Selfridge, B. Tuckerman, and S. S. Wagstaff Jr. *Factorizations of $b^n \pm 1$, $b = 2, 3, 5, 6, 7, 10, 11, 12$ Up to High Powers*, volume 22 of *Contemporary Mathematics*. American Mathematical Society, First edition, 1983, Second edition, 1988, Third edition, 2002. Electronic book available at: `http://homes.cerias.purdue.edu/~ssw/cun/index.html`, 1983. (Cited on pages 1, 12, 30, and 31.)

[13] J. Buchmann, J. Loho, and J. Zayer. An implementation of the general number field sieve. In D. R. Stinson, editor, *Advances in Cryptology – CRYPTO'93*, volume 773 of *Lecture Notes in Computer Science*, pages 159–165. Springer, Heidelberg, Aug. 1994. (Cited on pages 34 and 37.)

[14] J. P. Buhler, H. W. Lenstra Jr., and C. Pomerance. Factoring integers with the number field sieve. pages 50–94 in [54], 1992. (Cited on pages 24, 26, 37, 38, 39, 40, and 41.)

[15] E. Canfield, P. Erdös, and C. Pomerance. On a problem of Oppenheim concerning "Factorisatio Numerorum". *J. Number Theory*, 17:1–28, 1983. (Cited on page 4.)

[16] T. R. Caron and R. D. Silverman. Parallel implementation of the quadratic sieve. *J. Supercomput.*, 1:273–290, 1988. (Cited on pages 1, 19, and 22.)

[17] S. Cavallar. Strategies in filtering in the number field sieve. In W. Bosma, editor, *ANTS*, volume 1838 of *Lecture Notes in Computer Science*, pages 209–231. Springer, 2000. (Cited on pages 9 and 10.)

[18] S. Cavallar. *On the number field sieve integer factorisation algorithm*. PhD thesis, Leiden University, 2002. (Cited on pages 9 and 10.)

[19] S. Cavallar, B. Dodson, A. K. Lenstra, W. M. Lioen, P. L. Montgomery, B. Murphy, H. te Riele, K. Aardal, J. Gilchrist, G. Guillerm, P. C. Leyland, J. Marchand, F. Morain, A. Muffett, C. Putnam, C. Putnam, and P. Zimmermann. Factorization of a 512-bit RSA modulus. In B. Preneel, editor, *Advances in Cryptology – EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 1–18. Springer, Heidelberg, May 2000. (Cited on pages 9, 33, and 38.)

[20] S. Contini. Factoring integers with the self-initializing quadratic sieve. Masters Thesis, U. Georgia, 1997. (Cited on page 22.)

[21] D. Coppersmith. Fast evaluation of logarithms in fields of characteristic two. *IEEE Transactions on Information Theory*, 30:587–594, 1984. (Cited on page 26.)

[22] D. Coppersmith. Modifications to the number field sieve. *Journal of Cryptology*, 6(3):169–180, 1993. (Cited on pages 2, 31, 38, 43, and 44.)

[23] D. Coppersmith. Solving homogeneous linear equations over GF(2) via block Wiedemann algorithm. *Mathematics of Computation*, 62(205):333–350, 1994. (Cited on page 8.)

[24] D. Coppersmith, A. M. Odlyzko, and R. Schroeppel. Discrete logarithms in GF(p). *Algorithmica*, 1(1):1–15, 1986. (Cited on pages 8, 22, 24, 25, 29, and 30.)

[25] D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions. *J. Symbolic Comput.*, 9:251–280, 1990. (Cited on page 8.)

[26] J.-M. Couveignes. Computing a square root for the number field sieve. pages 95–102 in [54], 1992. (Cited on page 41.)

[27] J. Cowie, B. Dodson, R. M. Elkenbracht-Huizing, A. K. Lenstra, P. L. Montgomery, and J. Zayer. A world wide number field sieve factoring record: On to 512 bits. In K. Kim and T. Matsumoto, editors, *Advances in Cryptology – ASIACRYPT'96*, volume 1163 of *Lecture Notes in Computer Science*, pages 382–394. Springer, Heidelberg, Nov. 1996. (Cited on page 37.)

[28] A. J. C. Cunningham and H. J. Woodall. Factorizations of $y^n \pm 1$, $y = 2, 3, 5, 6, 7, 10, 11, 12$ up to high powers. Frances Hodgson, London, 1925. (Cited on pages 1, 30, and 31.)

[29] J. A. Davis, D. B. Holdridge, and G. J. Simmons. Status report on factoring (at the Sandia national laboratories). In T. Beth, N. Cot, and I. Ingemarsson, editors, *Advances in Cryptology – EUROCRYPT'84*, volume 209 of *Lecture Notes in Computer Science*, pages 183–215. Springer, Heidelberg, Apr. 1985. (Cited on pages 18 and 19.)

[30] N. De Bruijn. On the number of positive integers $\leq x$ and free of prime factors $> y$, ii. *Indag. Math.*, 38:239–247, 1966. (Cited on page 4.)

[31] M. Delcourt, T. Kleinjung, and A. K. Lenstra. Analyses of number field sieve variants. manuscript in preparation, 2016, 2016. (Cited on page 44.)

[32] T. F. Denny, B. Dodson, A. K. Lenstra, and M. S. Manasse. On the factorization of RSA-120. In D. R. Stinson, editor, *Advances in Cryptology – CRYPTO'93*, volume 773 of *Lecture Notes in Computer Science*, pages 166–174. Springer, Heidelberg, Aug. 1994. (Cited on page 37.)

[33] J. D. Dixon. Asymptotically fast factorization of integers. *Mathematics of Computation*, 36(153):255–260, 1981. (Cited on page 11.)

[34] B. Dodson and A. K. Lenstra. NFS with four large primes: An explosive experiment. In D. Coppersmith, editor, *Advances in Cryptology – CRYPTO'95*, volume 963 of *Lecture Notes in Computer Science*, pages 372–385. Springer, Heidelberg, Aug. 1995. (Cited on page 37.)

[35] T. ElGamal. A subexponential-time algorithm for computing discrete logarithms over GF(p^2). *IEEE Transactions on Information Theory*, 31:473–481, 1985. (Cited on pages 24, 25, 26, 28, and 29.)

[36] J. Franke and T. Kleinjung. Continued fractions and lattice sieving. In *Special-purpose Hardware for Attacking Cryptographic Systems – SHARCS 2005*, 2005. `http://www.hyperelliptic.org/tanja/SHARCS/talks/FrankeKleinjung.pdf`. (Cited on page 34.)

[37] J. L. Gerver. Factoring large integers with a quadratic sieve. *Mathematics of Computation*, 41:287–294, 1983. (Cited on pages 17 and 18.)

[38] R. Golliver, A. K. Lenstra, and K. McCurley. Lattice sieving and trial division. In *Algorithmic Number Theory Symposium – ANTS'94*, volume 877 of *LNCS*, pages 18–27, 1994. (Cited on pages 20, 33, 34, and 37.)

[39] F. Göloglu, R. Granger, G. McGuire, and J. Zumbrägel. On the function field sieve and the impact of higher splitting probabilities — application to discrete logarithms in $\mathbb{F}_{2^{1971}}$ and $\mathbb{F}_{2^{3164}}$. In R. Canetti and J. A. Garay, editors, *Advances in Cryptology – CRYPTO 2013, Part II*, volume 8043 of *Lecture Notes in Computer Science*, pages 109–128. Springer, Heidelberg, Aug. 2013. (Cited on page 26.)

[40] R. Granger, T. Kleinjung, and J. Zumbrägel. On the discrete logarithm problem in finite fields of fixed characteristic. Available from `http://arxiv.org/abs/1507.01495`, 6th July 2015. (Cited on page 26.)

[41] G. H. Hardy and E. M. Wright. *An introduction to the theory of numbers*. Oxford Univ. Press, 4th edition, 1960. (Cited on page 13.)

[42] A. Joux. A new index calculus algorithm with complexity $L(1/4 + o(1))$ in small characteristic. In T. Lange, K. Lauter, and P. Lisonek, editors, *SAC 2013: 20th Annual International Workshop on Selected Areas in Cryptography*, volume 8282 of *Lecture Notes in Computer Science*, pages 355–379. Springer, Heidelberg, Aug. 2014. (Cited on page 26.)

[43] T. Kleinjung. Quadratic sieving. *Mathematics of Computation*, 85:1861–1873, 2016. (Cited on page 22.)

[44] T. Kleinjung, K. Aoki, J. Franke, A. K. Lenstra, E. Thomé, J. W. Bos, P. Gaudry, A. Kruppa, P. L. Montgomery, D. A. Osvik, H. J. J. te Riele, A. Timofeev, and P. Zimmermann. Factorization of a 768-bit RSA modulus. In T. Rabin, editor, *Advances in Cryptology – CRYPTO 2010*, volume 6223 of *Lecture Notes in Computer Science*, pages 333–350. Springer, Heidelberg, Aug. 2010. (Cited on pages 1 and 38.)

[45] T. Kleinjung, J. W. Bos, and A. K. Lenstra. Mersenne factorization factory. In P. Sarkar and T. Iwata, editors, *Advances in Cryptology – ASIACRYPT 2014, Part I*, volume 8873 of *Lecture Notes in Computer Science*, pages 358–377. Springer, Heidelberg, Dec. 2014. (Cited on pages 2, 11, 31, 37, 43, and 44.)

[46] T. Kleinjung, C. Diem, A. K. Lenstra, C. Priplata, and C. Stahlke. Computation of a 768-bit prime field discrete logarithm. In J.-S. Coron and J. Nielsen, editors, *Eurocrypt 2017, Part I*, volume 10210 of *Lecture Notes in Computer Science*, pages 185–201. Springer, Heidelberg, 2017. (Cited on page 38.)

[47] M. Kraitchik. *Théorie des nombres, Tome II*. Gauthiers-Villars, Paris, 1926. (Cited on page 2.)

[48] M. Kraitchik. *Recherches sur le théorie des nombres, Tome II*. Gauthiers-Villars, Paris, 1929. (Cited on page 2.)

[49] B. A. LaMacchia and A. M. Odlyzko. Solving large sparse linear systems over finite fields. In A. J. Menezes and S. A. Vanstone, editors, *Advances in Cryptology – CRYPTO'90*, volume 537 of *Lecture Notes in Computer Science*, pages 109–133. Springer, Heidelberg, Aug. 1991. (Cited on page 8.)

[50] F. Le Gall. Powers of tensors and fast matrix multiplication. In *Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation*, ISSAC '14, pages 296–303, New York, NY, USA, 2014. ACM. (Cited on page 8.)

[51] A. K. Lenstra. Fast and rigorous factorization under the generalized Riemann hypothesis. *Indagationes Mathematicae*, 50:443–454, 1988. (Cited on page 45.)

[52] A. K. Lenstra, H. W. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, 1982. (Cited on page 42.)

[53] A. K. Lenstra and H. W. Lenstra Jr. Algorithms in number theory. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science (Volume A: Algorithms and Complexity)*, pages 673–715. Elsevier and MIT Press, 1990. (Cited on pages 4, 6, 8, 21, 22, 25, 44, and 45.)

[54] A. K. Lenstra and H. W. Lenstra Jr. *The Development of the Number Field Sieve*, volume 1554 of *Lecture Notes in Mathematics*. Springer-Verlag, 1993. (Cited on pages 1, 24, 25, 47, 48, 50, and 51.)

[55] A. K. Lenstra, H. W. Lenstra Jr., M. S. Manasse, and J. M. Pollard. The number field sieve. pages 11–42 in [54], 1989. (Cited on pages 23, 26, 28, 30, 31, 32, 37, and 39.)

[56] A. K. Lenstra, H. W. Lenstra Jr., M. S. Manasse, and J. M. Pollard. The factorization of the ninth Fermat number. *Mathematics of Computation*, 61(203):319–349, 1993. (Cited on pages 9, 26, 31, and 32.)

[57] A. K. Lenstra and M. S. Manasse. Factoring by electronic mail. In J.-J. Quisquater and J. Vandewalle, editors, *Advances in Cryptology – EUROCRYPT'89*, volume

434 of *Lecture Notes in Computer Science*, pages 355–371. Springer, Heidelberg, Apr. 1990. (Cited on pages 1, 19, 20, 22, and 23.)

[58] A. K. Lenstra and M. S. Manasse. Factoring with two large primes. *Mathematics of Computation*, 63:785–798, 1994. (Cited on pages 17 and 22.)

[59] H. W. Lenstra Jr. Factoring integers with elliptic curves. *Annals of Mathematics*, 126(3):649–673, 1987. (Cited on pages 1 and 6.)

[60] H. W. Lenstra Jr. and C. Pomerance. A rigorous time bound for factoring integers. *Journal of the American Mathematical Society*, 5:483–516, 1992. (Cited on page 45.)

[61] P. C. Leyland, A. K. Lenstra, B. Dodson, A. Muffett, and S. S. Wagstaff Jr. MPQS with three large primes. In C. Fieker and D. R. Kohel, editors, *Algorithmic Number Theory, 5th International Symposium, ANTS-V*, volume 2369 of *Lecture Notes in Computer Science*, pages 446–460. Springer, 2002. (Cited on page 22.)

[62] A. Miele, J. W. Bos, T. Kleinjung, and A. K. Lenstra. Cofactorization on graphics processing units. In L. Batina and M. Robshaw, editors, *Cryptographic Hardware and Embedded Systems – CHES 2014*, volume 8731 of *Lecture Notes in Computer Science*, pages 335–352. Springer, Heidelberg, Sept. 2014. (Cited on page 37.)

[63] P. L. Montgomery. Square roots of products of algebraic numbers. *Mathematics of Computation 1943-1993: A Half-Century of Computational Mathematics*, 48:567–571, 1994. (Cited on page 41.)

[64] P. L. Montgomery. A block Lanczos algorithm for finding dependencies over GF(2). In L. C. Guillou and J.-J. Quisquater, editors, *Advances in Cryptology – EUROCRYPT'95*, volume 921 of *Lecture Notes in Computer Science*, pages 106–120. Springer, Heidelberg, May 1995. (Cited on page 8.)

[65] M. A. Morrison and J. Brillhart. A method of factoring and the factorization of F_7. *Mathematics of Computation*, 29(129):183–205, 1975. (Cited on pages 1, 2, 12, and 13.)

[66] P. Q. Nguyen. A Montgomery-like square root for the number field sieve. In J. Buhler, editor, *ANTS*, volume 1423 of *Lecture Notes in Computer Science*, pages 151–168. Springer, 1998. (Cited on page 41.)

[67] A. M. Odlyzko. Discrete logarithms in finite fields and their cryptographic significance. In T. Beth, N. Cot, and I. Ingemarsson, editors, *Advances in Cryptology – EUROCRYPT'84*, volume 209 of *Lecture Notes in Computer Science*, pages 224–314. Springer, Heidelberg, Apr. 1985. (Cited on pages 8 and 25.)

[68] R. Peralta. A quadratic sieve on the n-dimensional cube. In E. F. Brickell, editor, *Advances in Cryptology – CRYPTO'92*, volume 740 of *Lecture Notes in Computer Science*, pages 324–332. Springer, Heidelberg, Aug. 1993. (Cited on page 22.)

[69] J. M. Pollard. Theorems on factorization and primality testing. *Proceedings of the Cambridge Philosophical Society*, 76:521–528, 1974. (Cited on page 1.)

[70] J. M. Pollard. A Monte Carlo method for factorization. *BIT Numerical Mathematics*, 15(3):331–334, 1975. (Cited on pages 1 and 6.)

[71] J. M. Pollard. Monte Carlo methods for index computation (mod p). *Mathematics of Computation*, 32(143):918–924, 1978. (Cited on page 25.)

[72] J. M. Pollard. Factoring with cubic integers. pages 4–10 in [54], 1988. (Cited on pages 23, 24, 25, 30, 31, and 32.)

[73] J. M. Pollard. The lattice sieve. pages 43–49 in [54], 1990. (Cited on pages 33 and 34.)

[74] C. Pomerance. Analysis and comparison of some integer factoring algorithms. In J. Hendrik W. Lenstra and R. Tijdeman, editors, *Computational methods in number theory I*, volume 154 of *Mathematical Centre Tracts*, pages 89–139, Amsterdam, 1982. Mathematisch Centrum. (Cited on pages 4, 11, 16, 17, and 22.)

[75] C. Pomerance. Fast, rigorous factorization and discrete logarithm algorithms. pages 119–143 in [**?**], 1987. (Cited on pages 6 and 44.)

[76] C. Pomerance, October 1988. Private communication. (Cited on page 20.)

[77] C. Pomerance. A tale of two sieves. *Notices of the AMS*, 43(12):1473–1485, December 1996. (Cited on page 2.)

[78] C. Pomerance and J. W. Smith. Reduction of huge, sparse matrices over finite fields via created catastrophes. *Experiment. Math.*, 1:89–94, 1992. (Cited on pages 8, 9, and 10.)

[79] C. Pomerance, J. W. Smith, and R. Tuler. A pipeline architecture for factoring large integers with the quadratic sieve algorithm. *SIAM j. Comput.*, 17:387–403, 1988. (Cited on page 22.)

[80] C. Pomerance, J. W. Smith, and S. S. Wagstaff. New ideas for factoring large integers. In D. Chaum, editor, *Advances in Cryptology – CRYPTO'83*, pages 81–85. Plenum Press, New York, USA, 1983. (Cited on page 12.)

[81] R. L. Rivest, A. Shamir, and L. M. Adleman. A method for obtaining digital signature and public-key cryptosystems. *Communications of the Association for Computing Machinery*, 21(2):120–126, 1978. (Cited on pages 1, 4, and 16.)

[82] R. J. Schoof. Quadratic fields and factorization. In J. Hendrik W. Lenstra and R. Tijdeman, editors, *Computational methods in number theory II*, volume 155 of *Mathematical Centre Tracts*, pages 235–286, Amsterdam, 1982. Mathematisch Centrum. (Cited on page 2.)

[83] R. Schroeppel, April 2015. Private communication. (Cited on pages 1, 14, 15, 16, and 17.)

[84] M. Seysen. A probabilistic factorization algorithm with quadratic forms of negative discriminant. *Mathematics of Computation*, 48:757–780, 1987. (Cited on pages 13 and 44.)

[85] D. Shanks. Class number, a theory of factorization, and genera. In D. J. Lewis, editor, *Symposia in Pure Mathematics*, volume 20, pages 415–440. American Mathematical Society, 1971. (Cited on page 2.)

[86] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997. (Cited on page 38.)

[87] R. D. Silverman. The multiple polynomial quadratic sieve. *Mathematics of Computation*, 48:329–339, 1987. (Cited on pages 19, 21, and 22.)

[88] V. Strassen. Gaussian elimination is not optimal. *Numer. Math.*, 13:354–356, 1969. (Cited on page 8.)

[89] E. Thomé. Square root algorithms for the number field sieve. In F. Özbudak and F. Rodríguez-Henríquez, editors, *WAIFI*, volume 7369 of *Lecture Notes in Computer Science*, pages 208–224. Springer, 2012. (Cited on pages 40 and 41.)

[90] B. Vallée. Generation of elements with small modular squares and provably fast integer factoring algorithms. *Mathematics of Computation*, 56:823–849, 1991. (Cited on page 44.)

[91] D. Weber. Computing discrete logarithms with quadratic number rings. In *EUROCRYPT'98*, pages 171–183, 1998. (Cited on page 30.)

[92] A. E. Western and J. C. P. Miller. *Tables of indices and primitive roots*. Royal Society Mathematical Tables, vol 9, Cambridge University Press, 1968. (Cited on pages 2 and 25.)

[93] D. H. Wiedemann. Solving sparse linear equations over finite fields. *IEEE Trans. Inform. Theory*, IT–32(1):54–62, Jan. 1986. (Cited on page 8.)

[94] H. C. Williams. A $p + 1$ method of factoring. *Mathematics of Computation*, 39(159):225–234, 1982. (Cited on page 1.)

Subject index

algebraic side, 32

block Lanczos algorithm, *see* Lanczos algorithm
block Wiedemann algorithm, *see* Wiedemann algorithm

CFRAC, *see* continued fraction method
characteristic, 25
Chinese remainder theorem, 41
congruence of squares, 2
  finding dependencies, 7
  generic analysis, 5
  linear algebra, 3, 7
  relation collection, 3, 6
  smoothness testing, *see* smoothness testing
continued fraction method, 12–14
  multiplier, 13
Coppersmith-Winograd method, 8, 11
cubic sieve, 22
Cunningham number, 1, 12, 31

discrete logarithm, 25–30
Dixon's random squares method, 11–12

elliptic curve method of factorization, 1, 6
embarrassing parallelism, 1, 19

factor base, 3
factoring with cubic integers, 23
factorization factory, 2, 43
Fermat number, 1, 14, 23
filtering, 8–11
finding dependencies, 7
first degree prime ideal, 27
free relation, 35

Gaussian elimination, 8, 9, 20
Gaussian integer method, 24, 29
general number field sieve, *see* number field sieve
Georgia Cracker, 12

integer factoring, 1
Internet computation, 1, 19

L-notation, 4
Lanczos algorithm, 8
large prime relation, 6, 14, 17, 18, 22, 36–38
linear algebra step, 3
linear sieve, 14–17
  multiplier, 17
look-up table, 28

matrix multiplication exponent, 8, 11
Mersenne number, 2, 44
Morrison-Brillhart approach, 2
multiple polynomial quadratic sieve, *see* quadratic sieve, multiple polynomial
multiplier, 13, 17, 18

number field sieve, 23, 37–42, 44
  Coppersmith's variant, 43
  heuristic analysis, 42, 43
  quadratic character, 39
  relation, 39

oversquareness, 2, 3

$p - 1$ method, 1
$p + 1$ method, 1
polynomial selection, 38
prime ideal factorization, 26
provable integer factoring, 11–12, 44

quadratic sieve, 17–23
  fancy, 18–19
  multiple polynomial, 19–22
  multiplier, 18
  plain, 17–18
  self-initializing, 20, 22

rational side, 32

record calculation
  discrete logarithm
    extension field, 26
    prime field, 38
  integer factoring
    continued fraction method, 1, 12
    number field sieve, 2, 38
    quadratic sieve, 1, 18, 20
    special number field sieve, 2, 23, 31
relation, 3
relation collection step, 3
rho method, 1
RSA challenge
  RSA-512, 38
  RSA-768, 2, 38
sieving, 6–7
  by vectors, 34
  lattice sieving, 9, 33–35
  line sieving, 32
smoothness
  integers, 4
  polynomials, 25
smoothness testing, 6
  elliptic curve method of factorization, 6
  sieving, *see* sieving
  trial division, 6
sparseness, 7
special q-prime, 9, 19, 33
special number, 31
special number field sieve, 24, 30–37
  finding relations, 32–35
  heuristic analysis, 35
  relation, 31
square root computation for the number field sieve, 40–42
  Couveignes' method, 41
  direct method, 40, 41
  Montgomery's method, 41–42
squfof, 2
Strassen's method, 8, 11, 16
trial division, 1, 6
unit contribution, 28
Wiedemann algorithm, 8