Refereed. Will be printed in J. P. Buhler and P. Stevenhagen, *Algorithmic number theory: lattices, number fields, curves and cryptography.*

REDUCING LATTICE BASES TO FIND SMALL-HEIGHT VALUES OF UNIVARIATE POLYNOMIALS

DANIEL J. BERNSTEIN

Abstract. This paper generalizes several previous results on finding divisors in residue classes (Lenstra, Konyagin, Pomerance, Coppersmith, Howgrave-Graham, Nagaraj), finding divisors in intervals (Rivest, Shamir, Coppersmith, Howgrave-Graham), finding modular roots (Hastad, Vallée, Girault, Toffin, Coppersmith, Howgrave-Graham), finding high-power divisors (Boneh, Durfee, Howgrave-Graham), and finding codeword errors beyond half distance (Sudan, Guruswami, Goldreich, Ron, Boneh) into a unified algorithm that, given f and g, finds all rational numbers r such that $f(r)$ and $g(r)$ both have small height.

1. Introduction

Consider the fraction $(r^3-s)/n$, where n is a large integer with no known factors. Usually there is no cancellation between the numerator r^3-s and the denominator n. In other words, the height of $(r^3-s)/n$ is usually $\max\{\left|r^3-s\right|, n\}$. Here the **height** of a rational number m/n is, by definition, $\max\{|m|, |n|\}/\gcd\{m,n\}$.

However, if r is a cube root of s modulo n, then one can remove n from both the numerator and denominator. In other words, the height of $(r^3-s)/n$ is only $\max\{\left|(r^3-s)/n\right|, 1\}$. The problem of finding a cube root of s modulo n can thus be viewed as the problem of finding small-height values of the polynomial $(x^3-s)/n$.

Many other useful properties of numbers r can be recast in the form "$f(r)$ has small height" for various polynomials f. For example, the problem of factoring n can be viewed as the problem of finding all r such that r/n has small height.

There is a surprisingly fast method, using lattice-basis reduction, to find all numbers r such that *both* r and $f(r)$ have small height. This paper presents a very general statement of the method (see Theorem 2.3); asymptotically optimal parameters (see Section 3); and an exposition of various applications of the method (see Sections 4, 5, and 6). The theorems and algorithms can easily be switched from $\mathbf{Q}$ to the rational function field $\mathbf{F}_q(t)$ over a finite field $\mathbf{F}_q$, although better algorithms are often available in the function-field case.

I have made no attempt to cover analogous methods for higher-degree global fields or for polynomials in more variables. There are several papers on small-height values of bivariate polynomials, but each application seems to pose a new optimization problem. I will leave it to future writers to unify the literature on this topic.

Date: 2004.02.10. Permanent ID of this document: `82f82c041b7e2bdce94a5e1f94511773`.

2000 *Mathematics Subject Classification.* Primary 11Y16. Secondary 94B35.

The author was supported by the National Science Foundation under grant DMS–0140542, and by the Alfred P. Sloan Foundation.

History. The following table fits previous results into the framework of Theorem 2.3. Notation: f is the polynomial with useful small-height values; d is the degree of f; m is the lattice rank; k is the highest f exponent used in defining the lattice. Results improve primarily as m increases, secondarily as k increases.

Find	$f(r)$	k	m	Notes
divisors, in $u + v\mathbf{Z}$, of n	$(r + uw)/n$ where $wv \in 1 + n\mathbf{Z}$	1	3	1984 Lenstra [22], for proving primality
divisors, in an interval, of n	$(r + w)/n$ for one w	1	3	1986 Rivest Shamir [27], for breaking cryptosystems; independent of [22]
roots of $p(x)$ mod n	$p(r)/n$	1	$d+1$	1988 Håstad [16, Section 3]; first use of nonlinear f; independently: 1989 Vallée Girault Toffin [30] (using the dual lattice; more difficult)
roots of $p(x)$ mod n	$p(r)/n$	big	big	1996 Coppersmith [7] (using dual), for breaking cryptosystems; first use of increasing m; first use of increasing k; simplified: 1997 Howgrave-Graham [17] (explicitly avoiding dual)
divisors, in an interval, of n	$(r + w)/n$	big	big	1996 Coppersmith [8] (in a much more complicated way); simplified: 1997 Howgrave-Graham [17]
divisors, in $1 + v\mathbf{Z}$, of n	$(r + w)/n$	2	5	1997 Konyagin Pomerance [20, Algorithm 3.2]; independent of [7]
divisors, in $u + v\mathbf{Z}$, of n	$(r + uw)/n$	big	big	1998 Coppersmith Howgrave-Graham Nagaraj [18, Section 5.5]
large values of $\gcd\{x + w, n\}$	$(r + w)/n$	1	big	1999 Goldreich Ron Sudan [12] (using dual), for error correction; previous function-field version: 1997 Sudan [29]; independent of [7]
high-power divisors, in an interval, of n	$(r + w)^d/n$	big	big	1999 Boneh Durfee Howgrave-Graham [6]
large values of $\gcd\{x + w, n\}$	$(r + w)/n$	big	big	2000 Boneh [4], for error correction; independently: 2001 Howgrave-Graham [19, Section 3]; previous function-field version: 1999 Guruswami Sudan [15]
large values of $\gcd\{p(x), n\}$	$p(r)/n$	big	big	2000 Boneh [4, Section 4]

It was recognized in [17] and [6] that "$r + w$ divides n" and "$(r + w)^d$ divides n" could be handled by the same technique as "$p(r)$ is divisible by n." Meanwhile, "$\gcd\{r + w, n\}$ is large" appeared independently in [12]. A unified "$\gcd\{p(r), n\}$ is large" algorithm finally appeared, with insufficient fanfare, in [4, Section 4].

2. The general method

This section explains how to find all rational numbers r such that $f(r)$ and $g(r)$ simultaneously have small height. Here $f, g \in \mathbf{Q}[x]$ are polynomials, each of positive degree, each with positive leading coefficient. Write $d = \deg f$, and assume for simplicity that $\deg g = 1$.

Theorem 2.2 below gives a more precise definition of "small height." The height bound depends on two integer parameters $k \geq 1$ and $m \geq dk+1$. A typical special case is $k = 1$ and $m = 2d$. See Section 3 for further comments on the choice of k and m.

The lattice. Define $L \subset \mathbf{Q}[x]$ as the $\mathbf{Z}$-module

$$\begin{array}{llllllllll}
 & \mathbf{Z} & + & \mathbf{Z}g & + & \mathbf{Z}g^2 & + & \cdots & + & \mathbf{Z}g^{d-1} \\
+ & \mathbf{Z}f & + & \mathbf{Z}gf & + & \mathbf{Z}g^2 f & + & \cdots & + & \mathbf{Z}g^{d-1}f \\
+ & \mathbf{Z}f^2 & + & \mathbf{Z}gf^2 & + & \mathbf{Z}g^2 f^2 & + & \cdots & + & \mathbf{Z}g^{d-1}f^2 \\
+ & \cdots & & & & & & & & \\
+ & \mathbf{Z}f^{k-1} & + & \mathbf{Z}gf^{k-1} & + & \mathbf{Z}g^2 f^{k-1} & + & \cdots & + & \mathbf{Z}g^{d-1}f^{k-1} \\
+ & \mathbf{Z}f^k & + & \cdots & + & \mathbf{Z}g^{m-dk-1}f^k. & & & &
\end{array}$$

For example, if $k = 1$ and $m = d+1$, then $L = \mathbf{Z} + \mathbf{Z}g + \mathbf{Z}g^2 + \cdots + \mathbf{Z}g^{d-1} + \mathbf{Z}f$; if $k = 1$ and $m = 2d$, then $L = \mathbf{Z} + \mathbf{Z}g + \mathbf{Z}g^2 + \cdots + \mathbf{Z}g^{d-1} + \mathbf{Z}f + \cdots + \mathbf{Z}g^{d-1}f$.

The basis elements $1, g, \ldots, g^{d-1}, f, \ldots, g^{m-dk-1}f^k$ have degrees $0, 1, 2, \ldots, m-1$ respectively. Thus L is a lattice of rank m under the usual coefficient-vector metric on $\mathbf{Q}[x]$, namely $\varphi \mapsto |\varphi| = \sqrt{\varphi_0^2 + \varphi_1^2 + \varphi_2^2 + \cdots}$, where $\varphi = \varphi_0 + \varphi_1 x + \varphi_2 x^2 + \cdots$.

The basis elements have leading coefficients $1, g_1, g_1^2, \ldots, g_1^{m-dk-1} f_d^k$, where g_1 is the leading coefficient of g and f_d is the leading coefficient of f. Thus

$$\begin{aligned}
\det L &= g_1^{kd(d-1)/2+(m-dk)(m-dk-1)/2} f_d^{dk(k-1)/2+k(m-dk)} \\
&= g_1^{m(m-1)/2} (g_1^d/f_d)^{dk(k+1)/2-mk}.
\end{aligned}$$

For example, if $k = 1$ and $m = 2d$, then $\det L = g_1^{d(d-1)} f_d^d = g_1^{d(2d-1)} (g_1^d/f_d)^{-d}$.

Theorem 2.1. *Let d, k, m be positive integers with $m \geq dk+1$. Let $f \in \mathbf{Q}[x]$ be a polynomial of degree d with leading coefficient $f_d > 0$. Let $g \in \mathbf{Q}[x]$ be a polynomial of degree 1 with leading coefficient $g_1 > 0$. Define L as above. If $\varphi \in L$, $r \in \mathbf{Q}$, and $\gcd\{1, f(r)\}^k \gcd\{1, g(r)\}^{\max\{d-1, m-dk-1\}} > \left|(1, r, \ldots, r^{m-1})\right| |\varphi|$, then $\varphi(r) = 0$.*

For example, if $k = 1$, $m = 2d$, $\varphi \in L$, $r \in \mathbf{Q}$, and $\gcd\{1, f(r)\} \gcd\{1, g(r)\}^{d-1} > \left|(1, r, \ldots, r^{2d-1})\right| |\varphi|$, then $\varphi(r) = 0$.

The reader should interpret $\gcd\{1, f(r)\} > \cdots$ as "$f(r)$ has small denominator"; $\gcd\{1, g(r)\} > \cdots$ as "$g(r)$ has small denominator"; and $\left|(1, r, \ldots, r^{m-1})\right| < \cdots$ as "$f(r)$ and $g(r)$ have small numerators." Theorem 2.1 can thus be summarized as "$\varphi(r) = 0$ if $f(r)$ and $g(r)$ both have small height."

Proof. $|\varphi(r)| \leq \left|(1, r, \ldots, r^{m-1})\right| |\varphi| < \gcd\{1, f(r)\}^k \gcd\{1, g(r)\}^{\max\{d-1, m-dk-1\}}$. But $\varphi \in \mathbf{Z} + \mathbf{Z}g + \cdots + \mathbf{Z}g^{d-1}f^{k-1} + \mathbf{Z}f^k + \cdots + \mathbf{Z}g^{m-dk-1}f^k$ by definition of L, so $\varphi(r) \in \mathbf{Z} + \mathbf{Z}g(r) + \cdots + \mathbf{Z}g(r)^{d-1}f(r)^{k-1} + \mathbf{Z}f(r)^k + \cdots + \mathbf{Z}g(r)^{m-dk-1}f(r)^k \subseteq \mathbf{Z}\gcd\{1, f(r)\}^k \gcd\{1, g(r)\}^{\max\{d-1, m-dk-1\}}$. Thus $\varphi(r) = 0$. $\square$

Theorem 2.2. *Let d, k, m be positive integers with $m \ge dk + 1$. Let $f \in \mathbf{Q}[x]$ be a polynomial of degree d with leading coefficient $f_d > 0$. Let $g \in \mathbf{Q}[x]$ be a polynomial of degree 1 with leading coefficient $g_1 > 0$. Define L as above. Let $\varphi \in L$ be a nonzero vector such that $|\varphi| \le 2^{(m-1)/2}(\det L)^{1/m}$. If $r \in \mathbf{Q}$ and*

$$\gcd\{1, f(r)\}^k \gcd\{1, g(r)\}^{\max\{d-1, m-dk-1\}} > \left|(1, r, \dots, r^{m-1})\right| (2g_1)^{(m-1)/2} (g_1^d/f_d)^{dk(k+1)/2m-k}$$

then $\varphi(r) = 0$.

Proof. $(\det L)^{1/m} = g_1^{(m-1)/2}(g_1^d/f_d)^{dk(k+1)/2m-k}$. Apply Theorem 2.1. □

For example, if $k = 1$ and $m = 2d$, then $\varphi(r) = 0$ for every $r \in \mathbf{Q}$ such that $\gcd\{1, f(r)\} \gcd\{1, g(r)\}^{d-1} > \left|(1, r, \dots, r^{2d-1})\right| (2g_1)^{d-1/2}(g_1^d/f_d)^{-1/2}$.

Theorem 2.3. *Let d, k, m be positive integers with $m \ge dk + 1$. Let $f \in \mathbf{Q}[x]$ be a polynomial of degree d with leading coefficient $f_d > 0$. Let $g \in \mathbf{Q}[x]$ be a polynomial of degree 1 with leading coefficient $g_1 > 0$. Define L as above. Let $\varphi \in L$ be a nonzero vector such that $|\varphi| \le 2^{(m-1)/2}(\det L)^{1/m}$. Define $\gamma = m^{1/2k}(2g_1)^{(m-1)/2k}(g_1^d/f_d)^{d(k+1)/2m-1}$. If $r \in \mathbf{Q}$, $|r| \le 1$, $\gcd\{1, f(r)\} > \gamma$, and $g(r) \in \mathbf{Z}$, then $\varphi(r) = 0$.*

For example, if $k = 1$ and $m = 2d$, then $\varphi(r) = 0$ for every $r \in \mathbf{Q}$ such that $|r| \le 1$, $g(r) \in \mathbf{Z}$, and $\gcd\{1, f(r)\} > \gamma$, where $\gamma = m^{1/2}(2g_1)^{d-1/2}(g_1^d/f_d)^{-1/2}$.

Proof. $\gamma^k = m^{1/2}(2g_1)^{(m-1)/2}(g_1^d/f_d)^{dk(k+1)/2m-k}$; $\left|(1, r, \dots, r^{m-1})\right| \le m^{1/2}$; and $\gcd\{1, g(r)\} = 1$. Apply Theorem 2.2. □

Theorem 2.4. *Let d, k, m be positive integers with $m \ge dk + 1$. Let $f \in \mathbf{Q}[x]$ be a polynomial of degree d with leading coefficient $f_d > 0$. Let $g \in \mathbf{Q}[x]$ be a polynomial of degree 1 with leading coefficient $g_1 > 0$. Define L as above. Let $\varphi \in L$ be a nonzero vector such that $|\varphi| \le 2^{(m-1)/2}(\det L)^{1/m}$. Assume that $g_1 < (g_1^d/f_d)^{2k/(m-1)-dk(k+1)/m(m-1)}/2m^{1/(m-1)}$. If $r \in \mathbf{Q}$, $|r| \le 1$, $f(r) \in \mathbf{Z}$, and $g(r) \in \mathbf{Z}$, then $\varphi(r) = 0$.*

For example, if $k = 1$ and $m = 2d$, then $\varphi(r) = 0$ for every $r \in \mathbf{Q}$ such that $|r| \le 1$, $f(r) \in \mathbf{Z}$, and $g(r) \in \mathbf{Z}$, provided that $g_1 < (g_1^d/f_d)^{1/(2d-1)}/2(2d)^{1/(2d-1)}$.

Proof. By assumption $m^{1/2k}(2g_1)^{(m-1)/2k}(g_1^d/f_d)^{d(k+1)/2m-1} < 1 = \gcd\{1, f(r)\}$. Apply Theorem 2.3. □

Computation. It is easy to compute the rational numbers r identified in Theorems 2.2, 2.3, and 2.4:

- Feed the basis vectors $1, g, \dots, g^{d-1}, f, \dots, g^{m-dk-1}f^k$ of L to a lattice-basis-reduction algorithm, such as the Lenstra-Lenstra-Lovasz algorithm, to obtain a nonzero vector $\varphi \in L$ such that $|\varphi| \le 2^{(m-1)/2}(\det L)^{1/m}$. See [21] and Chapter **??** of this book. The theorems now state that all desired numbers r are roots of φ.
- Compute the rational roots of φ, by approximating the real (or 2-adic) roots of φ to high precision. See, e.g., [23]. By construction φ has degree at most $m - 1$, so it has at most $m - 1$ roots.
- Check each root r to see whether it satisfies the stated conditions.

Each step is reasonably fast if f, g, k, and m are reasonably small.

3. Parameter choice and other optimizations

This section discusses the choice of k amd m in Section 2, and other ways to speed up the computation of the desired numbers r.

Parameter choice for Theorem 2.3. Theorem 2.3 assumes that $\gcd\{1, f(r)\} > \gamma$, where $\gamma = m^{1/2k}(2g_1)^{(m-1)/2k}(g_1^d/f_d)^{d(k+1)/2m-1}$. How small can one make this lower bound γ?

Assume that g_1 and $1/f_d$ exceed 1. Theorem 3.1 then says that γ is smaller than $\beta = m^{1/2k}(2g_1)^{\alpha d(1+1/2k)} f_d/g_1^d$, where $\alpha = \sqrt{1 + (\log(1/f_d))/\log((2g_1)^d)}$, if m is chosen as $\lceil \alpha d(k+1) \rceil$. This choice of m approximately balances the factors $(2g_1)^{(m-1)/2k}$ and $(g_1^d/f_d)^{d(k+1)/2m}$ in Theorem 2.3. Note that $\alpha \geq 1$, so $m \geq dk+d$. Note also that m is not difficult to compute: comparing $\alpha d(k+1)$ to an integer boils down to comparing integer powers of f_d and $2g_1$.

As k increases (slowing down the computation), β converges to $(2g_1)^{\alpha d} f_d/g_1^d$, which is very close to a lower bound on γ. The quantity $(2g_1)^{\alpha d}$ is the doubly-geometric average of $(2g_1)^d$ and $(2g_1)^d/f_d$.

For comparison: If $k = 1$, the optimal choice of m is approximately $\sqrt{2}\alpha d$ for large αd, with $\gamma \approx (2g_1)^{\sqrt{2}\alpha d} f_d/g_1^d$. Allowing larger k thus changes the exponent of $2g_1$ by a factor of approximately $\sqrt{2}$. This is exactly the $\sqrt{2}$ improvement from [29] to [15], and from [12] to [4].

Theorem 3.1. *Let d, k be positive integers. Let $f \in \mathbf{Q}[x]$ be a polynomial of degree d with leading coefficient $f_d > 0$. Let $g \in \mathbf{Q}[x]$ be a polynomial of degree 1 with leading coefficient $g_1 > 0$. Assume that $g_1 \geq 1$ and $1/f_d \geq 1$. Define $\alpha = \sqrt{1 + (\log(1/f_d))/\log((2g_1)^d)}$, $m = \lceil \alpha d(k+1) \rceil$, $\beta = m^{1/2k}(2g_1)^{\alpha d(1+1/2k)} f_d/g_1^d$, and L as above. Let $\varphi \in L$ be a nonzero vector such that $|\varphi| \leq 2^{(m-1)/2}(\det L)^{1/m}$. If $r \in \mathbf{Q}$, $|r| \leq 1$, $\gcd\{1, f(r)\} \geq \beta$, and $g(r) \in \mathbf{Z}$, then $\varphi(r) = 0$.*

Proof. First $m - 1 \leq \alpha d(k+1)$ so $(2g_1)^{(m-1)/2k} \leq (2g_1)^{\alpha d(k+1)/2k}$. Second $1/m \leq 1/\alpha d(k+1)$ so $(g_1^d/f_d)^{d(k+1)/2m} \leq (g_1^d/f_d)^{1/2\alpha} < ((2g_1)^d/f_d)^{1/2\alpha} = (2g_1)^{\alpha d/2}$ by choice of α. Thus $m^{1/2k}(2g_1)^{(m-1)/2k}(g_1^d/f_d)^{d(k+1)/2m} f_d/g_1^d < \beta$. Apply Theorem 2.3. □

Parameter choice for Theorem 2.4. Theorem 2.4 assumes that g_1 is smaller than $(g_1^d/f_d)^{2k/(m-1)-dk(k+1)/m(m-1)}/2m^{1/(m-1)}$. How large can one make this exponent $2k/(m-1) - dk(k+1)/m(m-1)$?

Theorem 3.2 chooses $m = dk + d$, achieving exponent $k/(dk + d - 1)$, which is reasonably close to optimal. As k increases (slowing down the computation), the exponent converges to $1/d$.

Theorem 3.2. *Let d, k be positive integers. Let $f \in \mathbf{Q}[x]$ be a polynomial of degree d with leading coefficient $f_d > 0$. Let $g \in \mathbf{Q}[x]$ be a polynomial of degree 1 with leading coefficient $g_1 > 0$. Define $m = dk + d$ and $L = \mathbf{Z} + \mathbf{Z}g + \cdots + \mathbf{Z}g^{d-1} + \mathbf{Z}f + \mathbf{Z}gf + \cdots + \mathbf{Z}g^{d-1}f + \cdots + \mathbf{Z}f^k + \mathbf{Z}gf^k + \cdots + \mathbf{Z}g^{d-1}f^k$. Let $\varphi \in L$ be a nonzero vector such that $|\varphi| \leq 2^{(m-1)/2}(\det L)^{1/m}$. Assume that $g_1 < (g_1^d/f_d)^{k/(m-1)}/2m^{1/(m-1)}$. If $r \in \mathbf{Q}$, $|r| \leq 1$, $f(r) \in \mathbf{Z}$, and $g(r) \in \mathbf{Z}$, then $\varphi(r) = 0$.*

Proof. $d(k+1)/m = 1$ so $2k/(m-1) - dk(k+1)/m(m-1) = 2k/(m-1) - k/(m-1) = k/(m-1)$. Apply Theorem 2.4. □

Combining Theorem 3.2 with brute force. Theorem 3.2, applied to f and g, finds all rational numbers $r \in [-1,1]$ with $f(r), g(r) \in \mathbf{Z}$. The same theorem, applied to $f(x+2)$ and $g(x+2)$, finds all rational numbers $r \in [1,3]$ with $f(r), g(r) \in \mathbf{Z}$. With c such computations, involving c lattices of rank $m = dk + d$, one can cover an r interval of length $2c$.

One can view Theorem 3.2 as searching the rationals r with $g(r) \in \mathbf{Z}$, to see which ones also have $f(r) \in \mathbf{Z}$. In an interval of length $2c$, there are approximately $2cg_1 < c(g_1^d/f_d)^{k/(dk+d-1)}$ rationals r with $g(r) \in \mathbf{Z}$, so the number of r's searched per unit time is approximately $(g_1^d/f_d)^{k/(dk+d-1)}$ divided by the time to handle a lattice of rank $dk+d$. Given f and g, one can choose k to (approximately) maximize this ratio. This idea appears in [7].

Another way to expand the number of r's searched is to perform several rational-root calculations after each lattice-basis reduction, searching for roots of shifts of φ. For example, the roots of $\varphi-2, \varphi-1, \varphi, \varphi+1, \varphi+2$ include all $r \in \mathbf{Q}$ such that $|r| \le 1$, $f(r) \in \mathbf{Z}$, and $g(r) \in \mathbf{Z}$, provided that $g_1 < 3(g_1^d/f_d)^{k/(m-1)}/2m^{1/(m-1)}$; note the 3 here. I learned this idea from Hendrik Lenstra.

Smaller improvements. The choice of m in Theorem 3.1 is often suboptimal. It is better to have the computer run through all pairs (k,m), in increasing order of the r computation time, until finding a pair (k,m) where the bound in Theorem 2.3 is satisfactory. Similar comments apply to Theorem 3.2.

I quoted lattice-basis reduction in Section 2 as producing nonzero vectors $\varphi \in L$ such that $|\varphi|$ is at most $2^{(m-1)/2}(\det L)^{1/m}$. Slower reduction algorithms can shrink the factor $2^{(m-1)/2}$; even without this extra work, lattice-basis reduction often produces a vector φ with $|\varphi| < (\det L)^{1/m}$. Bounds that depend on φ, as in Theorem 2.1, are slightly better than bounds that depend solely on $\det L$.

In Theorems 2.3, 2.4, 3.1, and 3.2, the lattice L can be replaced by a slightly smaller lattice, namely $\mathbf{Z}+\mathbf{Z}g+\mathbf{Z}g(g-1)/2+\mathbf{Z}g(g-1)(g-2)/6+\cdots$. The point is that $g(r)(g(r)-1)/2$ etc. are integers if $g(r)$ is an integer. This idea was published in [10], with credit to Howgrave-Graham and Lenstra independently.

A few years earlier, Howgrave-Graham in [18, Section 4.5.2] had made the similar observation that f could often be replaced by $f/d!$, after suitable tweaking of the coefficients of f.

Another slight improvement is to change the metric used to define the lattice, replacing $1, x, x^2, \ldots, x^{m-1}$ with Chebyshev polynomials. This idea was published by Coppersmith in [10, page 24], with partial credit (of unclear scope) to Boneh.

4. Example: roots mod n given their high bits

Theorem 4.1 explains how to search an interval $[-H, H]$ for integer roots of an integer polynomial p modulo n, if H is not too large. For example, with $p = (x+t)^3 - s$, Theorem 4.1 explains how to search $[t-H, t+H]$ for cube roots of s modulo n, if H is not too large.

Theorem 4.1. *Let d, k, n be positive integers. Let $p \in \mathbf{Z}[x]$ be a monic polynomial of degree d. Define $m = dk + d$. Let H be a positive integer smaller than $n^{k/(m-1)}/2m^{1/(m-1)}$. Define $f = p(Hx)/n \in \mathbf{Q}[x]$, $g = Hx \in \mathbf{Q}[x]$, and $L = \mathbf{Z}+\mathbf{Z}g+\cdots+\mathbf{Z}g^{d-1}+\mathbf{Z}f+\mathbf{Z}gf+\cdots+\mathbf{Z}g^{d-1}f+\cdots+\mathbf{Z}f^k+\mathbf{Z}gf^k+\cdots+\mathbf{Z}g^{d-1}f^k$. Let $\varphi \in L$ be a nonzero vector such that $|\varphi| \le 2^{(m-1)/2}(\det L)^{1/m}$. If $s \in \mathbf{Z}$, $p(s) \in n\mathbf{Z}$, and $|s| \le H$, then $\varphi(s/H) = 0$.*

Proof. Define $r = s/H$. By hypothesis $r \in \mathbf{Q}$, $|r| \le 1$, $f(r) = p(s)/n \in \mathbf{Z}$, $g(r) = s \in \mathbf{Z}$, and $g_1 = H < n^{k/(m-1)}/2m^{1/(m-1)} = (g_1^d/f_d)^{k/(m-1)}/2m^{1/(m-1)}$. Apply Theorem 3.2. □

As k increases, the exponent $k/(m-1)$ converges to $1/d$. One can push H up to (and slightly beyond) $n^{1/d}$ by combining Theorem 4.1 with brute force, as discussed in Section 3. For example, one can find cube roots of s modulo n in any interval of length about $n^{1/3}$. This generalizes the obvious fact that one can quickly compute r from $r^3 \bmod n$ if $0 \le r < n^{1/3}$.

For comparison: $(k, m) = (1, d+1)$ works with H up to $n^{2/d(d+1)}/2(d+1)^{1/d}$; i.e., about $n^{1/6}$ if $d = 3$.

History. As indicated in Section 1, the $n^{2/d(d+1)}$ result was first published by Håstad, and the $n^{1/d}$ result was first published by Coppersmith. Both authors used their results to break various naive forms of the RSA cryptosystem.

The results also have a positive application to cryptography: viz., compressing RSA (or Rabin) signatures. Instead of transmitting a cube root (or square root) of s modulo n, one can transmit the top 2/3 (or 1/2) of the bits of the root. However, this application is now obsolete, because Bleichenbacher has proposed a different compression mechanism allowing substantially faster decompression and verification: compress the cube root to an integer v such that the remainder $v^3 s \bmod n$ is a cube in $\mathbf{Z}$.

A numerical example. Define $n = 2844847044114666594769924451263$ and $p = (x + 1249180057712313741000000000000)^2 - 1982518464324230691670577165029$. The goal here is to find a square root of 1982518464324230691670577165029 modulo n close to 1249180057712313741000000000000.

Choose $k = 2$ and $H = 10^{12}/2$. Define $d = \deg p = 2$ and $m = dk + d = 6$. Then $m(2H)^{m-1} = 6 \cdot 10^{60} < n^2$ so $H < n^{k/(m-1)}/2m^{1/(m-1)}$. Define $f = p(Hx)/n$, $g = Hx$, and $L = \mathbf{Z} + \mathbf{Z}g + \mathbf{Z}f + \mathbf{Z}gf + \mathbf{Z}f^2 + \mathbf{Z}gf^2$.

Reduce the basis $1, g, f, gf, f^2, gf^2$ to find a nonzero vector in L of length at most $2^{(m-1)/2}(\det L)^{1/m} = 2^{5/2}H^{5/2}/n \approx 0.352$: for example, the vector $\varphi = 3gf^2 - 14990160692547764892644746695414f^2 + 16455550604884219114654409906953gf - 70731079160202264042139668259422536394926 0f + (\cdots)g + (\cdots)1 =$

$$
\begin{aligned}
&(93750000000000000000000000000000000000000000000000000000000/n^2)x^5\\
&-(40296668463375000000000000000000000000000000000000000000000/n^2)x^4\\
&-(8521407708770506207316875000000000000000000000000000000000/n^2)x^3\\
&+(7549559148957274134432151119009658000000000000000000000000/n^2)x^2\\
&+(85256085569824577108175046901012420957509825119500000000000/n^2)x\\
&-(73391645786690147620682490399407175727933183364776412308271/n^2)1,
\end{aligned}
$$

of length approximately 0.019.

The only rational root of φ is $372834385559/H$. Check that $p(372834385559)$ is a multiple of n, i.e., that 1249180057712313741372834385559 is a square root of 1982518464324230691670577165029 modulo n.

Theorem 4.1 guaranteed that this procedure would find every integer root of p modulo n in the interval $[-H, H]$. (Theorem 2.1 guaranteed an even wider interval after $|\varphi|$ turned out to be noticeably smaller than $2^{(m-1)/2}(\det L)^{1/m}$.) This is much faster than separately checking each of the $10^{12} + 1$ integers in this interval.

5. Example: constrained divisors of n

Theorem 5.1 explains how to search for small integers s such that

- $u + s$ divides n; or, more generally,
- $u + vs$ divides n, where v is coprime to n; or, more generally,
- $(u + vs)^d$ divides n, where v is coprime to n.

For example, by choosing $d = 1$ and choosing v as a large power of 2, one can search for divisors of n having specified low bits.

Theorem 5.1. *Let d, k, n, u, v, w, H be positive integers such that $vw - 1 \in n\mathbf{Z}$ and $n \ge H^d$. Define $\alpha = \sqrt{(\log 2^d n)/\log 2^d H^d}$, $m = \lceil \alpha d(k+1) \rceil$, $f = (uw + Hx)^d/n \in \mathbf{Q}[x]$, $g = Hx \in \mathbf{Q}[x]$, $\lambda = m^{1/2kd}(2H)^{\alpha(1+1/2k)}$, and L as above. Let $\varphi \in L$ be a nonzero vector such that $|\varphi| \le 2^{(m-1)/2}(\det L)^{1/m}$. If $s \in \mathbf{Z}$, $|s| \le H$, $u + vs \ge \lambda$, and $n \in (u + vs)^d\mathbf{Z}$, then $\varphi(s/H) = 0$.*

The polynomial $(uw + Hx)^d/n$ used here is better than $(u + vHx)^d/n$ when $v > 1$: it has a smaller leading coefficient, so it produces a smaller lattice L.

Proof. By hypothesis $u + vs \ge \lambda > 0$. Note that $u + vs$ divides $uw + s$. Indeed, $u + vs$ divides $(u + vs)w = uw + s + (vw - 1)s$; but $u + vs$ also divides $(u + vs)^d$, hence n, hence $vw - 1$.

Define $r = s/H$. Then $f(r) = (uw + s)^d/n$. The numerator $(uw + s)^d$ and the denominator n are both divisible by $(u + vs)^d$, so $\gcd\{1, f(r)\} \ge (u + vs)^d/n \ge \lambda^d/n = m^{1/2k}(2H)^{\alpha d(1+1/2k)}/n$.

By hypothesis $g_1 = H \ge 1$; $1/f_d = n/H^d \ge 1$; $\alpha = \sqrt{1 + \log(1/f_d)/\log((2g_1)^d)}$; $r \in \mathbf{Q}$; $|r| = |s|/H \le 1$; $\gcd\{1, f(r)\} \ge m^{1/2k}(2g_1)^{\alpha d(1+1/2k)} f_d/g_1^d$; and $g(r) = s \in \mathbf{Z}$. Apply Theorem 3.1. □

The main limitation in Theorem 5.1 is the condition $u + vs \ge \lambda$. To search the arithmetic progression $u - vH, u - v(H-1), \ldots, u + v(H-1), u + vH$, one must ensure that the smallest entry $u - vH$ exceeds λ, where λ^d is approximately the doubly-geometric average of n and H^d. In other words, if the smallest entry $u - vH$ is about $n^{1/d\alpha}$, then the number of entries is at most about $n^{1/d\alpha^2}$.

In particular, say $d = 1$, and say we are searching for divisors around $n^{1/2}$ in a specified arithmetic progression. Then the number of entries searched is at most about $n^{1/4}$. This bound is tight: a sufficiently large choice of k will achieve $n^{1/4-\epsilon}$ for any desired $\epsilon > 0$.

Consider, for example, a divisor between $n^{0.49}$ and $n^{0.50}$ in a specified arithmetic progression $u + v\mathbf{Z}$, where $v > n^{0.27}$. Select $d = 1$, $k = 24$, and $H \approx (2n)^{0.23}/2$; then $\alpha \approx 2.08514$ and $\lambda \approx n^{0.48957}$. Select u in the arithmetic progression slightly above $\lambda + vH$. Theorem 5.1 then searches the arithmetic progression from $u - vH$ through $u + vH$; if n is large then $[n^{0.49}, n^{0.50}] \subseteq [u - vH, u + vH]$. One can cover other ranges of divisors by varying H.

History. As indicated in Section 1, results of this type were developed in two contexts independently. The first context is proving primality of n: the Adleman-Pomerance-Rumely method in [3] exhibits some arithmetic progressions and proves, using factors of unit groups of extensions of $\mathbf{Z}/n$, that every divisor of n is in one of those progressions. The second context is factoring an RSA public key n given part of the secret key: for example, finding a divisor of n given the low bits of the divisor.

In the first context, Lenstra in [22] showed how to find all divisors of n in an arithmetic progression $u + v\mathbf{Z}$ with $\lg v > (1/3)\lg n$. Konyagin and Pomerance in [20, Algorithm 3.2] improved $(1/3)\lg n$ to $0.3\lg n$, in the special case $u = 1$. This $0.3\lg n$ result, for any u, follows from Theorem 2.3 with $m = 5$ and $k = 2$; I have not checked whether the resulting algorithm is equivalent to the Konyagin-Pomerance algorithm.

In the second context, Rivest and Shamir in [27] gave a heuristic outline of a method to find a divisor of n given about $(1/3)\lg n$ high bits of the divisor. Coppersmith in [8] proved that a much more complicated bivariate algorithm would find a divisor of n given $(0.25 + \epsilon)\lg n$ high bits of the divisor. Howgrave-Graham in [17] achieved $(0.25+\epsilon)\lg n$ with the simpler algorithm shown here. Each of these authors commented that the method also applied to low bits, but they did not generalize to other arithmetic progressions.

These two threads in the literature were finally combined in [18, Section 5.5]: Coppersmith, Howgrave-Graham, and Nagaraj improved the Konyagin-Pomerance $0.3\lg n$ to $(0.25 + \epsilon)\lg n$.

Boneh, Durfee, and Howgrave-Graham in [6] pointed out, at least for $v = 1$, the further generalization from divisors $u + vs$ to divisors $(u + vs)^d$. As d increases, the allowable range of H shrinks, but the range of interesting divisors shrinks more quickly. At an extreme, for d larger than about $\sqrt{\lg n}$, this method finds d-power divisors of n more quickly than the elliptic-curve method.

A numerical example. Define $d = 2$, $u = 1814430925000000$, $v = 1$, $w = 1$, and $n = 3767375198243112483228974667456105955144630367$. The goal here is to find a divisor p^2 of n, given that $p \approx 1814430925000000$.

Choose $k = 2$ and $H = 10^6$. Define $\alpha = \sqrt{(\log 4n)/\log 4H^2} \approx 1.91424$ and $m = \lceil \alpha d(k+1) \rceil = 12$. Then $u - H \geq \lambda$ where $\lambda = m^{1/2kd}(2H)^{\alpha(1+1/2k)}$. Define $f = (uw + Hx)^d/n = (u + Hx)^2/n$, $g = Hx$, and $L = \mathbf{Z} + \mathbf{Z}g + \mathbf{Z}f + \mathbf{Z}gf + \mathbf{Z}f^2 + \mathbf{Z}gf^2 + \mathbf{Z}g^2f^2 + \mathbf{Z}g^3f^2 + \mathbf{Z}g^4f^2 + \mathbf{Z}g^5f^2 + \mathbf{Z}g^6f^2 + \mathbf{Z}g^7f^2$.

Reduce the basis $1, g, f, gf, f^2, gf^2, g^2f^2, g^3f^2, g^4f^2, g^5f^2, g^6f^2, g^7f^2$ to find a nonzero vector in L of length at most $2^{(m-1)/2}(\det L)^{1/m}$: for example, the vector

$$\begin{aligned}
&865428592905169853673115657973973290925440337012446696387011830651 6f^2 \\
&\quad - 60501094449047328939676706095029782423264573493203 54f \\
&\quad - 27255412018787295847722163555072174417628911011368 05gf^2 \\
&\quad - 1321737599339233171981104958040247284 \\
&\quad - 666887822947220831282660069477245533 2gf \\
&\quad + 751073287899629272340418092672916546g^2f^2 \\
&\quad - 832523980748052892274g \\
&\quad - 165577708623278785839g^3f^2 \\
&\quad + 22814g^4f^2,
\end{aligned}$$

of length approximately $2.3 \cdot 10^{-38}$. The only rational root of this polynomial is $339897/H$. Check that 1814430925339897^2 is a divisor of n.

Theorem 5.1 guaranteed that this procedure would find all divisors $(u+s)^2$ of n with $-H \leq s \leq H$. In fact, Theorem 2.3 guaranteed that $k = 2$ and $m = 7$ would have done the same job, and that $k = 1$ and $m = 5$ would have worked for the smaller interval $-450000 \leq s \leq 450000$.

6. Example: codeword errors past half the minimum distance

Fix a positive integer H. Fix finitely many distinct primes $p_1, p_2, \ldots$. Assume that the product $n = p_1 p_2 \cdots$ is much larger than H. The **residue representation** of an integer $s \in [-H, H]$ is, by definition, the vector $(s \bmod p_1, s \bmod p_2, \ldots)$.

If $s' \neq s$ then there must be many differences between the residue representations of s and s'. Define the **distance** between s and s' as the sum of $\lg p_i$ for all i such that $s \bmod p_i \neq s' \bmod p_i$. Then the distance between s and s' is exactly $\lg n - \lg \gcd\{s' - s, n\}$, which is at least $\lg n - \lg 2H$ since $\gcd\{s' - s, n\} \leq 2H$.

Thus the residue representation can tolerate some errors. For any vector v, there is at most one s whose representation has distance $< (\lg n - \lg 2H)/2$ from v.

Theorem 6.1 explains how to efficiently recover s from a vector at any distance up to about $\lg n - \sqrt{(\lg 2n) \lg 2H}$. One first interpolates the vector into an integer $u \in \{0, 1, \ldots, n-1\}$, and then finds s such that $\gcd\{u - s, n\}$ is large. Of course, for distances above $(\lg n - \lg 2H)/2$, there might be several possibilities for s; Theorem 6.1 finds them all.

The simplest case $k = 1$, $m = 2$ of Theorem 6.1 finds all s with $\gcd\{u - s, n\} > (4Hn)^{1/2}$, i.e., with distance smaller than $(\lg n - \lg 4H)/2$. There is at most one such s.

Theorem 6.1. *Let k, n, u, H be positive integers such that $n \geq H$. Define $\alpha = \sqrt{(\log 2n)/\log 2H}$, $m = \lceil \alpha(k+1) \rceil$, $\lambda = m^{1/2k}(2H)^{\alpha(1+1/2k)}$, $f = (Hx - u)/n \in \mathbf{Q}[x]$, $g = Hx \in \mathbf{Q}[x]$, $d = 1$, and L as above. Let $\varphi \in L$ be a nonzero vector such that $|\varphi| \leq 2^{(m-1)/2}(\det L)^{1/m}$. If $s \in \mathbf{Z}$, $|s| \leq H$, and $\gcd\{u - s, n\} \geq \lambda$, then $\varphi(s/H) = 0$.*

Compare to the case $v = 1$, $w = 1$, $d = 1$ of Theorem 5.1.

Proof. Define $r = s/H$. By hypothesis $g_1 = H \geq 1$; $1/f_d = n/H \geq 1$; $\alpha = \sqrt{1 + \log(1/f_d)/\log(2g_1)}$; $r \in \mathbf{Q}$; $|r| = |s|/H \leq 1$; $g(r) = s \in \mathbf{Z}$; and $f(r) = (s - u)/n$, so $\gcd\{1, f(r)\} \geq \lambda/n = m^{1/2k}(2g_1)^{\alpha(1+1/2k)} f_d/g_1$. Apply Theorem 3.1. $\square$

History. The rational-function-field version of the simple case $k = 1$, $m = 2$ is the "Berlekamp-Massey algorithm" for decoding "Reed-Solomon codes."

The fact that one can efficiently correct larger errors was pointed out in the function-field case by Sudan in [29], and in the number-field case by Goldreich, Ron, and Sudan in [12]. These results are tantamount to optimizing m in Theorem 2.3 with $k = 1$. The $\sqrt{2}$ improvement from larger k's was pointed out in the function-field case by Guruswami and Sudan in [15], and in the number-field case by Boneh in [4].

Algorithms that may produce several values of s are often called "list decoding" algorithms. Of course, the resulting list is most useful when it has just one value of s.

A numerical example. Define $H = 1000000$, $n = 101 \cdot 103 \cdot 107 \cdot 109 \cdot 113 \cdot 127 \cdot 131 \cdot 137 \cdot 139 \cdot 149 \cdot 151 \cdot 157 \cdot 163 \cdot 167 \cdot 173 \cdot 179 \cdot 181 \cdot 191 \cdot 193 \cdot 197 \cdot 199$, and $u = 4765345845193600442153574482968114946568482 07$. The goal here is to find every $s \in [-H, H]$ with residue representation close to $(u \bmod 101, \ldots, u \bmod 199) = (94, 43, 17, 71, 103, 77, 64, 25, 114, 9, 106, 16, 62, 134, 75, 13, 155, 26, 138, 21, 105)$.

Choose $k = 3$. Define $\alpha = \sqrt{(\log 2n)/\log 2H} \approx 2.697$ and $m = \lceil \alpha(k+1) \rceil = 9$. Define $f = (Hx - u)/n$, $g = Hx$, and $L = \mathbf{Z} + \mathbf{Z}f + \mathbf{Z}f^2 + \mathbf{Z}f^3 + \mathbf{Z}gf^3 + \mathbf{Z}g^2f^3 + \mathbf{Z}g^3f^3 + \mathbf{Z}g^4f^3 + \mathbf{Z}g^5f^3$.

Reduce the basis $1, f, f^2, f^3, gf^3, g^2f^3, g^3f^3, g^4f^3, g^5f^3$ to find a nonzero vector in L of length at most $2^{(m-1)/2}(\det L)^{1/m}$: for example, the vector

$$\begin{aligned}
&(25587000000000000000000000000000000000000000000000000/n^3)x^8\\
&- (11721491973000000000000000000000000000000000000000000/n^3)x^7\\
&- (16962979598492916000000000000000000000000000000000000/n^3)x^6\\
&- (9080495056404508812155000000000000000000000000000000/n^3)x^5\\
&- (16268872584260106363071226779000000000000000000000000/n^3)x^4\\
&- (4786092732625488403021583597543361000000000000000000/n^3)x^3\\
&- (68525665600669610580610714527465995863869000000000000/n^3)x^2\\
&- (4866470374300829151096400546244449180155160401000000/n^3)x\\
&+ (19654220351564720341671319570621613333314080770830407/n^3)1.
\end{aligned}$$

The only rational root of this polynomial is s/H where $s = 476511$. The vector $(94, 33, 40, 72, 103, 7, 64, 25, 19, 9, 106, 16, 62, 60, 69, 13, 119, 157, 187, 165, 105)$ is the residue representation of s; the distance from s to u is approximately 79.41.

Theorem 6.1 guaranteed that this procedure would find every s within distance $\lg n - \lg \lambda \approx 79.8887$ of u; here $\lambda = m^{1/2k}(2H)^{\alpha(1+1/2k)}$. Even better, Theorem 2.3 guaranteed that this procedure would find every s within distance $-\lg \gamma \approx 83.38$ of u; here $\gamma = m^{1/2k}(2H)^{(m-1)/2k}n^{d(k+1)/2m-1}$. Both bounds are far above $(\lg n - \lg 2H)/2 \approx 45.16$.

References

[1] —, *Annual ACM symposium on theory of computing: proceedings of the 31st symposium (STOC '99) held in Atlanta, GA, May 1–4, 1999*, Association for Computing Machinery, New York, 1999. ISBN 1–58113–067–8. MR 2001f:68004.

[2] —, *Proceedings of the 32nd annual ACM symposium on theory of computing*, Association for Computing Machinery, New York, 2000. ISBN 1–58113–184–4.

[3] Leonard M. Adleman, Carl Pomerance, Robert S. Rumely, *On distinguishing prime numbers from composite numbers*, Annals of Mathematics **117** (1983), 173–206. ISSN 0003–486X. MR 84e:10008.

[4] Dan Boneh, *Finding smooth integers in short intervals using CRT decoding*, in [2] (2000), 265–272; see also newer version [5].

[5] Dan Boneh, *Finding smooth integers in short intervals using CRT decoding*, Journal of Computer and System Sciences **64** (2002), 768–784; see also older version [4]. ISSN 0022–0000. MR 1 912 302. Available from `http://crypto.stanford.edu/~dabo/abstracts/CRTdecode.html`.

[6] Dan Boneh, Glenn Durfee, Nick Howgrave-Graham, *Factoring $N = p^rq$ for large r*, in [31] (1999), 326–337. Available from `http://crypto.stanford.edu/~dabo/abstracts/prq.html`.

[7] Don Coppersmith, *Finding a small root of a univariate modular equation*, in [24] (1996), 155–165; see also newer version [9]. MR 97h:94008.

[8] Don Coppersmith, *Finding a small root of a bivariate integer equation; factoring with high bits known*, in [24] (1996), 178–189; see also newer version [9]. MR 97h:94009.

[9] Don Coppersmith, *Small solutions to polynomial equations, and low exponent RSA vulnerabilities*, Journal of Cryptology **10** (1997), 233–260; see also older version [7] and [8]. ISSN 0933–2790. MR 99b:94027.

[10] Don Coppersmith, *Finding small solutions to small degree polynomials*, in [28] (2001), 20–31. MR 2003f:11034.

[11] Michael Darnell (editor), *Cryptography and coding: proceedings of the 6th IMA International Conference held at the Royal Agricultural College, Cirencester, December 17–19, 1997*, Lecture Notes in Computer Science, 1355, Springer-Verlag, 1997. ISBN 3–540–63927–6. MR 99g:94019.

[12] Oded Goldreich, Dana Ron, Madhu Sudan, *Chinese remaindering with errors*, in [1] (1999), 225–234; see also newer version [13]. MR 2001i:68050. Available from `http://theory.lcs.mit.edu/~madhu/papers.html`.

[13] Oded Goldreich, Dana Ron, Madhu Sudan, *Chinese remaindering with errors*, IEEE Transactions on Information Theory **46** (2000), 1330–1338; see also older version [12]. ISSN 0018–9448. MR 2001k:11005. Available from `http://theory.lcs.mit.edu/~madhu/papers.html`.

[14] Ronald L. Graham, Jaroslav Nešetřil (editors), *The mathematics of Paul Erdős. I*, Algorithms and Combinatorics, 13, Springer-Verlag, Berlin, 1997. ISBN 3–540–61032–4. MR 97f:00032.

[15] Venkatesan Guruswami, Madhu Sudan, *Improved decoding of Reed-Solomon and algebraic-geometry codes*, IEEE Transactions on Information Theory **45** (1999), 1757–1767. ISSN 0018–9448. MR 2000j:94033. Available from `http://theory.lcs.mit.edu/~madhu/bib.html`.

[16] Johan Håstad, *Solving simultaneous modular equations of low degree*, SIAM Journal on Computing **17** (1988), 336–341. ISSN 0097–5397. MR 89e:68049. Available from `http://www.nada.kth.se/~johanh/papers.html`.

[17] Nicholas Howgrave-Graham, *Finding small roots of univariate modular equations revisited*, in [11] (1997), 131–142. MR 99j:94049.

[18] Nicholas Howgrave-Graham, *Computational mathematics inspired by RSA*, Ph.D. thesis, 1998. Available from `http://dimacs.rutgers.edu/~dieter/Seminar/Papers/nick-thesis.ps`.

[19] Nicholas Howgrave-Graham, *Approximate integer common divisors*, in [28] (2001), 51–66. MR 2003h:11160.

[20] Sergei Konyagin, Carl Pomerance, *On primes recognizable in deterministic polynomial time*, in [14] (1997), 176–198. MR 98a:11184. Available from `http://cr.yp.to/bib/entries.html#1997/konyagin`.

[21] Arjen K. Lenstra, Hendrik W. Lenstra, Jr., László Lovász, *Factoring polynomials with rational coefficients*, Mathematische Annalen **261** (1982), 515–534. ISSN 0025–5831. MR 84a:12002. Available from `http://cr.yp.to/bib/entries.html#1982/lenstra-lll`.

[22] Hendrik W. Lenstra, Jr., *Divisors in residue classes*, Mathematics of Computation **42** (1984), 331–340. ISSN 0025–5718. MR 85b:11118. Available from `http://www.jstor.org/sici?sici=0025-5718(198401)42:165<331:DIRC>2.0.CO;2-6`.

[23] Rüdiger Loos, *Computing rational zeros of integral polynomials by p-adic expansion*, SIAM Journal on Computing **12** (1983), 286–293. ISSN 0097–5397. MR 85b:11123.

[24] Ueli M. Maurer (editor), *Advances in cryptology—EUROCRYPT '96: Proceedings of the Fifteenth International Conference on the Theory and Application of Cryptographic Techniques held in Saragossa, May 12–16, 1996*, Lecture Notes in Computer Science, 1070, Springer-Verlag, Berlin, 1996. ISBN 3–540–61186–X. MR 97g:94002.

[25] Teo Mora (editor), *Applied algebra, algebraic algorithms and error-correcting codes: proceedings of the sixth international conference (AAECC-6) held in Rome, July 4–8, 1988*, Lecture Notes in Computer Science, 357, Springer-Verlag, Berlin, 1989. ISBN 3–540–51083–4. MR 90d:94002.

[26] Franz Pichler (editor), *Advances in cryptology—EUROCRYPT '85: proceedings of a workshop on the theory and application of cryptographic techniques (EUROCRYPT '85) held in Linz, April 1985*, Lectures Notes in Computer Science, 219, Springer-Verlag, 1986. ISBN 3–540–16468–5. MR 87d:94003.

[27] Ronald L. Rivest, Adi Shamir, *Efficient factoring based on partial information*, in [26] (1986), 31–34. MR 851 581.

[28] Joseph H. Silverman (editor), *Cryptography and lattices: proceedings of the 1st International Conference (CaLC 2001) held in Providence, RI, March 29–30, 2001*, Lecture Notes in Computer Science, 2146, Springer-Verlag, Berlin, 2001. ISBN 3–540–42488–1. MR 2002m:11002.

[29] Madhu Sudan, *Decoding of Reed Solomon codes beyond the error-correction bound*, Journal of Complexity **13** (1997), 180–193. ISSN 0885–064X. MR 98f:94024. Available from `http://theory.lcs.mit.edu/~madhu/bib.html`.

[30] Brigitte Vallée, Marc Girault, Philippe Toffin, *How to guess ℓth roots modulo n by reducing lattice bases*, in [25] (1989), 427–442. MR 90k:11168. Available from `http://cr.yp.to/bib/entries.html#1989/vallee`.

[31] Michael Wiener (editor), *Advances in cryptology—CRYPTO '99*, Lecture Notes in Computer Science, 1666, Springer-Verlag, Berlin, 1999. ISBN 3–5540–66347–9. MR 2000h:94003.

Department of Mathematics, Statistics, and Computer Science (M/C 249), The University of Illinois at Chicago, Chicago, IL 60607–7045

E-mail address: `djb@cr.yp.to`