Deterministic Integer Factorization Algorithms

N. A. Carella

Abstract: A new integer deterministic factorization algorithm, rated at arithmetic operations to $O(N^{1/6+\varepsilon})$ arithmetic operations, is presented in this note. Equivalently, given the least $(\log N)/6$ bits of a factor of the balanced integer $N = pq$, where $p$ and $q$ are primes, the algorithm factors the integer in polynomial time $O(\log(N)^c)$, with $c \geq 0$ constant, and $\varepsilon > 0$ an arbitrarily small number. It improves the current deterministic factorization algorithm, rated at arithmetic operations to $O(N^{1/5+\varepsilon})$ arithmetic operations.

Contents

1 Introduction 1
2 Deterministic Exponential Time $O(N^{1/4})$ 2
3 Deterministic Exponential Time $O(N^{1/5})$ 5
4 Basic Systems Of Polynomials Equations 5
5 Deterministic Exponential Time $O(N^{1/6})$ 6

1 Introduction

Let $\mathcal{A} = \{N = pq : p^{1+\delta} < q < ap^{1+\delta}\}$ and $\mathcal{B} = \{N = pq : p < q < ap\}$, where $p$ and $q$ are primes, and $a > 1$, and $\delta > 0$ are constants, be the subsets of unbalanced integers, and balanced integers respectively. The factorization of a balanced integer has the worst time complexity, while the factorization of an unbalanced integer does not have the worst time complexity. Thus, the time complexity of integer factorization is measured by the time complexity of factoring balanced integers. This article discusses a new deterministic integer factorization algorithm acting on the subset of balanced integers in details. The running time complexity of this algorithm is deterministic exponential time $O(N^{1/6+\varepsilon})$, where $\varepsilon > 0$ is a small number. This improves the current performances of deterministic integer factorization algorithms rated at the running time complexities $O(N^{1/4+\varepsilon})$. The standard references [6], [7], [18], [24], [25], [26], [28] and others, provide extensive details on the theory of integer
The innovation here involves a technique for generating multivariables polynomial equation \( f(x, y) = (a_1x + a_0)(b_1y + b_0) \in \mathbb{Z}[x, y] \) for the integer factorization problem, the earlier work in [3] use a simpler polynomial \( f_0(x, y) = (x + a_0)(y + b_0) \in \mathbb{Z}[x, y] \). The main result is the following.

**Theorem 1.1.** Any integer \( N \) can be decomposed into its prime factors in deterministic time complexity \( O(N^{1/6+\varepsilon}) \), where \( \varepsilon > 0 \), is an arbitrary small constant.

Equivalently, given the least \( \log N/6 \) bits of a factor of \( N = pq \), \( p \) and \( q \) primes, the algorithm factors the integer in polynomial time \( O(\log(N)^c) \), with \( c \geq 0 \) constant. This is an improvement of the Coppersmith algorithm, which requires the least \( \log N/4 \) bits, see Theorem 2.2.

The proof of this result follows from Theorem 5.1 in Section 5. Section 2 has a survey of the current deterministic integer factorization algorithms, and a new analysis of the Fermat Method. Section 4 covers some information on polynomials equations.

## 2 Deterministic Exponential Time \( O(N^{1/4}) \)

There are many integer factorization algorithms of deterministic exponential time complexities \( O(N^{1/4+\varepsilon}) \), where \( \varepsilon > 0 \) is an arbitrary small number, in the literature, see [7, p. 238], [6], [23], et cetera.

Integers \( N = pq \) composed of nearly equal factors \( p \) and \( q = p + d \), where \( d \) is small, can be factored by completing the square \( N + d^2/4 = x^2 \). This produces the factor \( p = \sqrt{N + d^2/4} - d/2 \). For example, a factor of a product of any twin primes pairs is \( p = \sqrt{N + 1} - 1 \). This is one of the earliest and most important integer factorization algorithms. It is usually called the Fermat factoring method, and uses an equivalent formulation based on the difference of squares equation \( 4N = x^2 - y^2 \). The Fermat factoring method effectively handles any integer \( N \geq 1 \) with a pair of factors in the ranges

\[
N^{1/2} - N^{1/4} \log^c N < p < N^{1/2} \quad \text{and} \quad N^{1/2} < q < N^{1/2} + N^{1/4} \log^c N, \tag{1}
\]

with \( c \geq 0 \) constant. A brute force search for the prime factors \( p \) and \( q \) of the integer \( N = pq \) in the ranges (1) has exponential time complexity of approximately \( O(N^{1/4} \log^c N) \) arithmetic operations, since

\[
q = \frac{N}{[N^{1/2}] + x} \quad \text{for some integer} \quad 0 \leq x \leq N^{1/4} \log^c N. \tag{2}
\]

The Fermat factoring method miraculously reduces the running time complexity from exponential time complexity \( O(N^{1/4} \log(N)^c) \) to polynomial time complexity \( O(\log^c N) \). This is accomplished by computing a solution \( U = p + q \), and \( V = q - p \) of the equation \( 4N = U^2 - V^2 \), instead of directly and independently computing the prime factors \( p \) and \( q \). The basic structure of this method is attributed to Fermat,
but it is not clear if the time complexity analysis was known before modern time. Statement (i) is well known, [25, p. 148], [12, p. 19], and other sources. But statement (ii) seems to be new. Here the symbol \([x]\) denotes the largest integer function.

**Theorem 2.1.** Suppose that an integer \(N = pq\) has a pairs of prime factors \(p\) and \(q\). Then, the followings hold.

(i) If the factors \(p\) and \(q\) satisfy

\[
|N^{1/2} - p| = O(N^{1/4} \log^c N) \quad \text{and} \quad |N^{1/2} - q| = O(N^{1/4} \log^c N),
\]

then it can be factored in deterministic polynomial time complexity \(O(\log(N)^c)\), with \(c \geq 0\) constant.

(ii) If the factors \(p\) and \(q\) satisfy

\[
|N^{1/2} - p| = O(N^{1/2} \log^c N) \quad \text{and} \quad |N^{1/2} - q| = O(N^{1/2} \log^c N),
\]

then it can be factored in deterministic exponential time complexity \(O(N^{1/4} \log(N)^{2c})\), with \(c \geq 0\) constant.

**Proof.** Case (i). To prove this claim, consider the approximations

\[
p = \lfloor N^{1/2} \rfloor + x \quad \text{and} \quad q = \lfloor N^{1/2} \rfloor + y,
\]

where \(0 \leq |x|, |y| = O(\log(N)^c)\), and \(c \geq 0\) is a constant. The search for a solution \(z = x + y\) of the equation \(4N = V^2 - V^2\), where \(U = p + q\) and \(V = q - p\), start with the sequence of approximations

\[
U_z = \lfloor N^{1/2} \rfloor + x + \lfloor N^{1/2} \rfloor + y = 2\lfloor N^{1/2} \rfloor + z,
\]

where \(z = 0, \pm 1, \pm 2, \ldots\). The approximate number of cycles required to determine a solution \((U_z, V_z)\) is at most

\[
p + q - \left(2N^{1/2} + z + \frac{N}{2N^{1/2} + z}\right) \leq \frac{(N^{1/2} - p)^2}{p} = O(\log^{2c} N)
\]

cycles. Therefore, the time complexity of the algorithm is at most \(O(\log^{2c} N)\) arithmetic operations.

Case (ii). To prove this claim, consider the approximations

\[
p = \lfloor N^{1/2} \rfloor + \lfloor N^{1/4} \rfloor x_0 + x \quad \text{and} \quad q = \lfloor N^{1/2} \rfloor + \lfloor N^{1/4} \rfloor y_0 + y,
\]

where

1. \(0 \leq |x_0|, |y_0| = O(N^{1/4})\),
2. \(0 \leq |x|, |y| = O(\log(N)^c)\),
and \(c \geq 0\) is a constant. The search for a solution \(z = x + y\) of the equation
\(4N = V^2 - U^2\), where \(U = p + q\) and \(V = q - p\), start with the sequence of approximations
\[
U_z = [N^{1/2}] + [N^{1/4}]x_0 + x + [N^{1/2}] + [N^{1/4}]y_0 + y
\]
\[
= 2[N^{1/2}] + 2[N^{1/4}]z_0 + z,
\]
where \(0 \leq |x_0| = O(N^{1/4})\) is a given parameter, and \(z = 0, \pm 1, \pm 2, \ldots\). Given the correct parameter \(z\), the approximate number of cycles required to determine a solution \((U_z, V_z)\) is at most
\[
p + q - (2N^{1/2} + [N^{1/4}]z_0 + z) \leq \frac{(N^{1/2} + [N^{1/4}]z_0 - p)^2}{p}
\]
\[
= O(\log^{2c} N)
\]
cycles. Since the correct parameter \(|z_0| = O(N^{1/4})\), the time complexity of the algorithm is at most \(O(N^{1/4} \log^{2c} N)\) arithmetic operations.

Algorithms that compute multiples of
\[
p + q, \quad q - p, \quad ap + bq, \quad aq - bp,
\]
are the topic of current research in integer factorization and related topics, consult [20], [13], et alii. The Pollard factoring method, and the elliptic curve factoring method, and a few other algorithms are based on the direct or indirect calculations of multiples of \(p + q = N + 1 - \varphi(N)\), or \(q - p = \sqrt{(p + q)^2 - 4N}\). It should be noted that multiples of \(p^2 + q^2 = N^2 + 1 - \varphi_2(N)\), are also effective.

Another related, and recently discovered integer factorization algorithm in this class is the following.

**Theorem 2.2.** ([3]) If the least (or most) significant \((\log N)/4\) bits of a prime factor \(p\) or \(q\) of the integer \(N = pq, p < q < 2p\), are given, then it can be decomposed in deterministic polynomial time complexity \(O(\log(N)^c)\), \(c > 0\) constant.

An improved version of this result is given in Theorem 1.1. The Fermat factoring algorithm (Theorem 2.1), and the Coppersmith factoring algorithm (Theorem 2.2), are equivalent integer factorization algorithms of the same running time complexity \(O(N^{1/4} \log(N)^c)\). Moreover, both have equivalent proofs based on lattice reduction methods. Both of these algorithms are practical for small integers, for example, \(N = O(2^{200})\) or thereabouts. Some improvement to the Fermat method is reported in [10], and [22], and experimental data for the Coppersmith factoring algorithm are compiled in [5], [11], [14], and many other similar references.

Another class of algorithm, based on efficient evaluations of high degree polynomials, is stated below. The author of this paper also have a survey of current literature on this class of integer factorization algorithms.

**Theorem 2.3.** ([4]) There exists a deterministic algorithm that computes the prime factorization of a positive integer \(N\) in \(O(N^{1/4} \log(N) \log \log(N)^{-1/2})\) bit operations.
3 Deterministic Exponential Time $O(N^{1/5})$

The fastest, deterministic, and unconditionally proven integer factorization algorithms in the literature have running time complexities $O(N^{1/5+\varepsilon})$. This is a very recent development. The previous algorithm of the same complexity was conditional on the GRH. This conditional integer factorization algorithm is based on the class number of numbers fields, see [7, p. 251] for some details.

**Theorem 3.1.** ([13]) There is a deterministic integer factorization algorithm achieving $O(N^{1/5+\varepsilon})$ arithmetic operations.

4 Basic Systems Of Polynomials Equations

The applications of lattice reduction methods to the theory of polynomial equations and its applications to cryptography are considered in fine details in [27], [3], [15], [2], [5], [8], [17], [19] and others.

Comprehensive introductions to lattice reduction methods and its applications to polynomials equations are given in [9, Chapter 2], [16, Chapter 3], [21, Chapter 3], and similar sources. The evolving analysis on a few specific polynomial equations of three variables is given in [1].

Employing lattice reduction methods, several results for the polynomials

$$f(x, y) = \sum_{0 \leq i,j \leq d} a_{i,j} x^i y^j$$ (10)

and

$$f(x, y, z) = \sum_{0 \leq i,j,k \leq d} a_{i,j,k} x^i y^j z^k$$ (11)

of two and three variables respectively, have been unconditionally proven. A relevant result from the theory of polynomials equations is included in this Section.

**Theorem 4.1.** ([3]) Let $f(x, y) \in \mathbb{Z}[x, y]$ be an irreducible polynomial of maximum degree $\deg(f) = d$ in each variable, and let $(x_0, y_0)$ be a root of $f(x, y) = 0$, such that $0 \leq |x_0| \leq X$, $0 \leq |y_0| \leq Y$. The height of the polynomial $f(xX, yY)$ is defined by

$$W = ||f(xX, yY)||_\infty = \max\{|a_{i,j}X^iY^j| : 0 \leq i, j \leq d\}.$$ (12)

(i) If $XY < W^{2/(3d)}$, then the roots $(x_0, y_0)$ can be determined in deterministic polynomial time $O(\log^c N)$, $c > 0$ constant.

(ii) If $XY < W^{1/d}$, and the total degree of the polynomial satisfies $0 = i + j = d$, then the roots $(x_0, y_0)$ can be determined in deterministic polynomial time $O(\log^c N)$, $c > 0$ constant.

The detailed heuristic analysis of the lattices for $f(x, y, z) = c_0 xy + c_1 x + c_2 y + c_3 z + c_4 \in \mathbb{Z}[x, y, z]$, and a few other polynomials of three variables and related applications, appear in [1], [11, p. 8], and [16, p. 66]. Practical applications also appear in [16] and [14, p. 11].
5 Deterministic Exponential Time $O(N^{1/6})$

For a pair of fixed parameters $0 < \alpha < 1 < \beta$, let $p$ and $q$ be prime numbers such that $\sqrt{\alpha N} < p < \sqrt{N}$, and $\sqrt{N} < q < \sqrt{\beta N}$, let $\gamma \sqrt{N} = (\sqrt{\alpha N} + \sqrt{\beta N})/2$ be the arithmetic mean of the interval $[\sqrt{\alpha N}, \sqrt{\beta N}]$. The Fermat factoring method, (Theorem 2.1) and Coppersmith factoring method (Theorem 2.2) assume that the prime factors of the integer $N = pq$ are centered at the symmetric center $\sqrt{N}$ of the interval $[\sqrt{\alpha N}, \sqrt{\beta N}]$. Shifting the symmetric center $\sqrt{N}$ to the arithmetic mean center $\gamma \sqrt{N}$ of the factors, or to a pair of distinct centers $\sqrt{\alpha N}$ and $\sqrt{\beta N}$, with $\alpha, \beta \neq 1$, can be used to derive various multivariable polynomials, which reduce the time complexities of both the Fermat factoring method, and Coppersmith factoring method, respectively.

**Theorem 5.1.** Given the least (or most) significant $(\log N)/6$ bits of a prime factor $p$ or $q$ of a large integer $N = pq$, $p < q < 2p$, the integer $N$ can be decomposed in deterministic polynomial time $O(\log^c N)$, with $c > 0$ constant.

**Proof.** Suppose that $N = pq$ has balanced prime factors such that $\sqrt{N/2} < p < \sqrt{N} < q < \sqrt{2N}$ centered at the symmetric center $\sqrt{N}$ of the interval $[\sqrt{N/2}, \sqrt{2N}]$, and the least significant $(\log N)/6$ bits $x_0$ of the prime factor $p$ are given. Here the symbol $[x]$ denotes the largest integer function. Here the symbol $[x]$ denotes the largest integer function.

Let $B = [N^{1/6}]$, and let $B^3 = [N^{1/2}]$. Assume the integer (or prime number) $B > x_0$ satisfies the condition $\gcd(B, x_0) = 1$. Now, expand the factors as $B$-adic integers

$$p = x_1B^3 + B^2x_3 + Bx_2 + x_0 = x_1B^3 + B(Bx_3 + x_2) + x_0,$$

and

$$q = y_1B^3 + B^2y_3 + By_2 + y_0 = y_1B^3 + B(By_3 + y_2) + y_0,$$

where the variables have the following dynamic ranges.

1. $0 \leq |x_i|, |y_i| < N^{1/6}$, for $i \in \{0, 1, 2, 3\}$,
2. $0 \leq |Bx_3 + x_2| = x \leq N^{1/3}$,
3. $0 \leq |By_3 + y_2| = y \leq N^{1/3}$,
4. $x_1y_1 \leq \log N$.

Lines 2 and 3 show the changes of variables $x = Bx_3 + x_2$ and $y = By_3 + y_2$. The last condition is arises from $x_1y_1B^6 = x_1y_1[N^{1/6}]^6 \leq N$, for balanced factors $p < q < 2p$, the small variables $x_1, y_1 \in \{1, 2\}$ work well.

Proceed to use the given least significant $(\log N)/6$ bits $x_0 < B$ of the prime factor $p$ to compute $y_0 < B$ via the congruence equation

$$(x_1B^3 + Bx + x_0)(y_1B^3 + By + y_0) - N \equiv 0 \mod B$$

(15)
By the initial conditions gcd(B, x₀) = 1 on the integer B, this congruence has a unique solution
\[ y₀ \equiv N \cdot x₀^{-1} \mod B. \] (16)
Next, expanding the product \( N = (x₁B³ + Bx + x₀)(y₁B³ + By + y₀) \) yields the corresponding polynomial
\[
f(x, y) = (x₁B³ + Bx + x₀)(y₁B³ + By + y₀) - N
= B²xy + (y₁B³ + y₀)Bx + (x₁B³ + x₀)By
+ (x₁B³ + x₀)(y₁B³ + y₀) - N
= c₃xy + c₂x + c₁y + c₀.
\]
The coefficients are:
5. \( c₀ = (x₁B³ + x₀)(y₁B³ + y₀) - N, \)
6. \( c₁ = x₁B³ + x₀, \)
7. \( c₂ = y₁B³ + y₀, \)
8. \( c₃ = B². \)
To demonstrate that \( f(x, y) \in \mathbb{Z}[x, y] \) is an irreducible polynomial, consider the factorization
\[
f(x, y) = c₃xy + c₂x + c₁y + c₀ = (a₁x + a₀)(b₁y + b₀) \] (18)
into linear factors. Matching coefficients yields
9. \( a₀b₀ = c₀ = (x₁B³ + x₀)(y₁B³ + y₀) - N, \)
10. \( a₀b₁ = c₁ = x₁B³ + x₀, \)
11. \( a₁b₀ = c₂ = y₁B³ + y₀, \)
12. \( a₁b₁ = c₃ = B². \)
But, the constant terms
\[
a₀b₀ = \frac{(a₀b₁) \cdot (a₁b₀)}{a₁b₁} = \frac{(x₁B³ + x₀)(y₁B³ + y₀)}{B²} \neq c₀
\] (19)
do not agree. Therefore, this is an irreducible polynomial over the integers.
By construction, the polynomial \( f(x, y) \in \mathbb{Z}[x, y] \) has a small root \((x, y) = (x₄, y₄)\) such that \( p = x₁B³ + Bx₄ + x₀ \) is a factor of \( N \).
To estimate the upper bounds of the solutions $x$ and $y$, it is sufficient to estimate the height $W = ||f(yXxX)||_{\infty} \leq \sqrt{f(yY xX)}$. Suppose that $X = Y = N^{1/3}$, and $B$ is prime. Then,

$$\gcd(c_0, c_1, c_2, c_3) = 1,$$

(20)

where $c_0 \equiv c_3 \equiv 0 \bmod B$, but $c_1, c_2 \not\equiv 0 \bmod B$. Thus, it follows that the height is given by

$$||f(yXxX)||_{\infty} = \max\{c_3XY, c_2X, c_1Y, c_0\} = N,$$

(21)

for example, $c_3XY = B^{2}XY = N^{1/3}XY = N$.

Since $XY \leq W^{2/3} = N^{2/3}$, where $\deg(f) = d = 1$ is the total degree of the polynomial, using lattice reduction methods, the small integer roots $x_1 \leq N^{1/3}$ and $x_1 \leq N^{1/3}$ can be determined in deterministic polynomial time, see Theorem 4.1. ■

**Note 1.** Simple modification of Theorem 5.1 can be used to handle all the other factorizations of nonbalanced integers $N = pq$ with $p$ and $q$ primes such that $(\alpha N)^{1/2 - \delta} < p < N^{1/2} < q < (\beta N)^{1/2 + \delta}$ with $\delta > 0$. For example, if $(\alpha N)^{1/3} < p < N^{1/3}$ and $N^{1/3} < q < (\beta N)^{2/3}$, where $0 < \alpha < \beta$ are small constants. Let $B = N^{1/6}$, and write the prime factors in the form

$$p = x_1B^2 + Bx + x_0 \quad \text{and} \quad q = y_1B^4 + By + y_0,$$

(22)

where $0 \leq |x| \leq N^{1/12}$, $0 \leq |y| \leq N^{7/12}$, $0 \leq |x_0|, |y_0| \leq N^{1/6}$, and $x_1y_1[N^{1/12}]^{12} \leq N$. Now proceed as before, but making the necessary changes as needed. At most a few changes of parameter $\delta \in \{0, 1/6, 1/4, \ldots \}$ are required to cover all possible prime factorizations.

**References**

[1] Bauer, Aurelie; Joux, Antoine. Toward a Rigorous Variation of Coppersmith’s Algorithm on Three Variables, Deterministic Integer Factorization Algorithms. EUROCRYPT 2007, LNCS 4515, pp. 361-378, 2007.

[2] Blomer, Johannes; May, Alexander. A Tool Kit for Finding Small Roots of Bivariate Polynomials over the Integers. EUROCRYPT 2005: 251-267.

[3] Coppersmith, Don. Small solutions to polynomial equations, and low exponent RSA vulnerabilities. J. Cryptology 10 (1997), no. 4, 233-260.

[4] Costa, Edgar; Harvey, David. Faster Deterministic Integer Factorization. Mathematics Of Computation, May 7, 2013.

[5] Coron, Jean-Sebastien. Finding small roots of bivariate integer polynomial equations: a direct approach. Advances in cryptology-CRYPTO 2007, 379-394.

[6] Cohen, Henri. A course in computational algebraic number theory. Graduate Texts in Mathematics, 138. Springer-Verlag, Berlin, 1993.
[7] Pomerance, C.; Crandall, R. *Primes Numbers: A Computational Perspective.* Springer-Verlag, 2006.

[8] Coron, Jean-Sebastien; Kirichenko, Alexey; Tibouchi, Mehdi. *A Note on the Bivariate Coppersmith Theorem.* Journal of Cryptology, April 2013, Volume 26, Issue 2, pp 246-250.

[9] Durfee, Glenn. *CRYPTANALYSIS OF RSA USING ALGEBRAIC AND LATTICE METHODS.* Thesis, Stanford University, 2002.

[10] Erra, Robert; Grenier, Christophe. *The Fermat factorization method revisited.* 2009, epreprint.org

[11] Ernst, M; Jochemsz, E.; May, A.; de Weger, B. *Partial key exposure attacks on RSA up to full size exponents.* Lecture Notes in Computer Science 3494, 371-387, EUROCRYPT 2005.

[12] de Weger, Benne. *Cryptanalysis of RSA with small prime difference.* Appl. Algebra Engrg. Comm. Comput. 13 (2002), no. 1, 17-28.

[13] David Harvey, Markus Hittmeir. *A log-log speedup for exponent one-fifth deterministic integer factorisation.* http://arxiv.org/abs/2105.11105.

[14] Hinek, M. Jason. *On the security of multi-prime RSA.* J. Math. Cryptol. 2 (2008), no. 2, 117-147.

[15] Howgrave-Graham, Nicholas. *Finding small roots of univariate modular equations revisited.* Cryptography and coding (Cirencester, 1997), 131-142, Lecture Notes in Computer Science, 1355, Springer, Berlin, 1997.

[16] Jochemsz, Ellen; May, Alexander. *A Polynomial Time Attack on RSA with Private CRT-Exponents Smaller Than N 0.073.* CRYPTO 2007: 395-411.

[17] Jochemsz, Ellen. *Cryptanalysis of RSA variants using small roots of polynomials.* TU Eindhoven, PhD Thesis, October 4, 2007.

[18] Lenstra, A. *Integers Factoring.* Designs, Codes, and Cryptography, 19, 101-128 (2000).

[19] Lenstra, Hendrik W., Jr. *Lattices.* Algorithmic number theory: lattices, number fields, curves and cryptography, 127-181, Math. Sci. Res. Inst. Publ., 44, Cambridge Univ. Press, Cambridge, 2008.

[20] Lehman, R. S. *Factoring large integers.* Math. Comp. 28 (1974), 637-646.

[21] May, Alexander. *New RSA Vulnerabilities Using Lattice Reduction Methods.* Thesis, University of Pader, 2003.

[22] McKee, James. *Speeding Fermat’s factoring method.* Math. Comp. 68 (1999), no. 228, 1729-1737.
[23] McKee, James; Pinch, Richard. *Old and new deterministic factoring algorithms*. Algorithmic number theory (Talence, 1996), LNCS 1122, 217-224.

[24] Menezes, Alfred J.; van Oorschot, Paul C.; Vanstone, Scott A. *Handbook of applied cryptography*. CRC Press, Boca Raton, FL, 1997.

[25] Riesel, Hans. *Prime numbers and computer methods for factorization*. Second edition. Progress in Mathematics, 126. Birkhauser Boston, Inc., Boston, MA, 1994.

[26] Shoup, Victor. *A computational introduction to number theory and algebra*. Cambridge University Press, Cambridge, 2005.

[27] Vallee, Brigitte; Girault, Marc; Toffin, Philippe. *How to guess lth roots modulo n by reducing lattice bases*. Applied algebra, algebraic algorithms and error-correcting codes (Rome, 1988), 427-442, Lecture Notes in Comput. Sci., 357, Springer, Berlin, 1989.

[28] Williams, Hugh C. *Edouard Lucas and primality testing*. Canadian Mathematical Society Series of Monographs and Advanced Texts, 22. A Wiley-Interscience Publication. New York, 1998.