Introduction to Topics in computational number theory inspired by Peter L. Montgomery

Joppe W. Bos, Arjen K. Lenstra, Herman te Riele, and Daniel Shumow

NXP Semiconductors, Leuven, Belgium

EPFL, Lausanne, Switzerland

CWI, Amsterdam, Netherlands

Microsoft Research, Redmond, USA

Abstract

This article appeared as the introductory chapter 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`. It contains multiple cross-references to the other chapters of the same book.

Contents

1 Introduction *page* 1
- 1.1 Outline 1
- 1.2 Biographical sketch 1
- 1.3 Overview 5
- 1.4 Simultaneous inversion 8

Bibliography 11
Subject index 15

1

Introduction

Joppe W. Bos, Arjen K. Lenstra, Herman te Riele and Daniel Shumow

1.1 Outline

This introductory chapter collects some personal background on Peter L. Montgomery, both provided by himself and based on stories from his colleagues and collaborators. This is followed by a brief introduction to the other chapters, and by a description of Peter's simultaneous inversion method.

1.2 Biographical sketch

Peter was born in San Francisco, California, on September 25, 1947. He described his activities since his 1965 graduation from San Rafael High School (SRHS) in a letter for a 1995 high school reunion; this letter is copied in its entirety in Figure 1.1, and further commented on throughout this chapter.

At Berkeley, Peter's undergraduate advisor was Derrick H. Lehmer, an excellent match given Peter's research interests since high school. Ron Graham, one of Peter's coauthors on his first three papers (all on Ramsey theory and written shortly after his time in Berkeley), recalls [25] that Peter had the "breakthrough idea" for the first of these papers, with coauthor Bruce Rothschild remembering Peter "as an extremely clever guy who came up with surprising ideas" [51]. The papers in question ([22], [23] and [24]) were further coauthored by Paul Erdös, Joel Spencer, and Ernst Straus. After this auspicious start, Peter's next paper describes an algorithm that he developed at SDC to evaluate arbitrary Fortran boolean expressions in such a way that they result in a small number of instructions when compiled on a Control Data 7600 computer [35]. The paper is illustrative of the type of tinkering in which Peter excels.

About twenty years after his time in Berkeley, and by the time that Pe-

Name: Peter L. Montgomery

Activities since graduation in 1965

I went to UC Riverside in 1965 and to UC Berkeley in 1967. While at Berkeley, I lived in Cloyne Court, a student co-op, where I was vending machine manager for three years. I ranked among the top five participants in the 1967 William Lowell Putnam Mathematical Competition, the best ranking possible. I got a BA in Mathematics in 1969 and an MA in 1971.

In 1972, I went to work as a junior programmer for System Development Corporation (SDC) in Huntsville, AL. SDC had the support contract at an Army supercomputer center. They acquired a CDC 7600 one month after I arrived. I spent considerable time studying the 7600, and soon became the local guru.

I waited until age 17 before taking driver training at SRHS, and never liked to drive. The environmental movement blossomed while I was at Berkeley, and I vowed in 1972 never to drive again. But Huntsville lacked sidewalks between home and work. For one year, I walked with large placards ''WE NEED SIDEWALKS'' and ''WHERE'S MY LANE'', until the City Council voted to install sidewalks near schools and to fund bike lanes and paths. I served on the bikeways committee, and bought a bicycle after bike paths were installed along my route to work in 1979.

My relatives urged me to move closer to San Rafael. I transferred to SDC's Santa Monica, CA office in 1982, and remained there until it closed in 1989. Meanwhile the company name changed to Unisys.

Most assignments at Unisys made little use of my mathematical background, other than the ability to think logically. But I remained active in the mathematical community, attending at least one conference per year. The Santa Monica computers were accessible on weekends (unlike the Huntsville machines), and I did mathematical research during off-hours. My research focused on computational number theory, especially algorithms for integer factorization. This topic had interested me ever since SRHS teacher Thomas Place asked my 1962 trigonometry class about primes of the form 2^n - 1. I improved known algorithms, demonstrated my improvements by running jobs during idle time, and published the results.

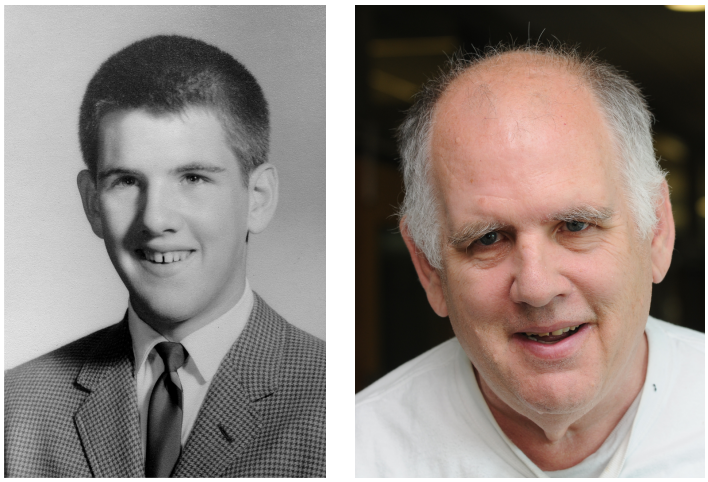
In 1987, I returned to graduate school at UCLA part time. After Unisys's 1989 closure, I got a 3-year U.S. Army fellowship and switched to full time. I got my PhD in Mathematics in 1992.

By this time, I was internationally famous in the mathematical community. For example, I had all expenses paid to a 1991 German cryptographic workshop. In 1992 I accepted a position at Oregon State University (OSU) in Corvallis. That ended in 1993 when a National Science Foundation grant was not renewed. Then I got an eight-month position at Centrum voor Wiskunde en Informatica (CWI) in Amsterdam, Holland, to continue the research begun in Oregon. In June, 1994, we overcame the last of several technical hurdles and factored some record-size numbers.

When I returned to graduate school, I intended my education to be a sabbatical, after which I would return to an industrial software position, preferably in California. But a recession intervened, and I got few interviews in 1991-1993 before accepting the OSU and CWI research positions. Since the CWI work ended, my only offer has been a research position in a New Jersey community lacking sidewalks. I live with my mother in Terra Linda.

Figure 1.1 Peter's letter for his 1995 high-school reunion

ter was already "internationally famous" (cf. Figure 1.1), David G. Cantor supervised Peter's PhD dissertation *An FFT extension of the elliptic curve method of factorization* [38]. Milos Ercegovac, who served on the PhD committee, related [21] that Cantor referred to Peter as "a genius UCLA never saw before", how Peter was brilliant in solving engineering problems related to computer arithmetic, and that, as a result of his incisive reviews, authors



(a) At age 15 (b) In 2009

Figure 1.2 Peter L. Montgomery (from Peter's collection and Wikipedia)

on several occasions asked if they could include this "marvelous" reviewer as a coauthor because Peter's comments invariably provided better solutions than those given in the submissions. Another member of the committee, Murray Schacher, recalled [52] how Peter "regularly swamped the system" in what may have been his first venture into large scale computing: verifying reducibility of $X^{8k} + X^m + 1 \in \mathbf{F}_2[X]$ for large k and $0 < m < 8k$. For this project Peter taught himself the Cantor-Zassenhaus method because the non-sparse $8k \times 8k$ matrix required by the usual Berlekamp algorithm surpassed the system's capacity [4, 11].

Peter did not accept the offer for a "research position in a New Jersey community lacking sidewalks" (cf. Figure 1.1). The offer was made after Peter's successful visits at the Bellcore research facility in Morristown, NJ, during the summers of 1994 and 1995. During his stays Peter worked with the second author's summer internship students Scott Contini and Phong Nguyen on implementing Peter's new methods that were useful for the number field sieve integer factorization algorithm (cf. Section 1.3 below): block Lanczos with Scott in 1994 and the next year the final square root method with Phong (who for his project had to learn about integer lattices and lattice basis reduction, and did so only very reluctantly – after a scolding by Ramarathnam "Venkie" Venkatesan). A memorable event was a lab outing where Peter beat Phong at bowling and afterwards bought ice cream for the whole party.

Figure 1.3 Front cover of [3] (© System Development Corporation, 1981).

In 1998 Peter accepted a position with the *Security and Cryptography group* at Microsoft Research (MSR) in Redmond, WA. He retired in 2014. His work at MSR was largely focused on pure research (cf. [30], [16], [41], [19], [20], and [9]). However, Peter began implementing libraries for asymmetric encryption for use in products as well. Peter's first contribution was to write Diffie–Hellman key exchange and the Digital Signature Algorithm in the static bignum library. This was included in the default Microsoft Windows CAPI 1.0 library dssenh.dll, that shipped in Windows 2000. A later version of this library, msbignum, also included RSA as well as elliptic curve cryptography. In 2006, with Windows Vista, msbignum became the basis of the API for the Cryptographic Next Generation and underlied in the following decade all cryptography running on Windows at Microsoft. Peter's role may have been unique at MSR at that time: over the course of the organization's existence there have been very few teams at MSR, let alone individual researchers, delivering entire functional software libraries.

During his time at MSR, Peter continued collaborations with researchers world-wide and was a regular long-term guest at various universities and research institutes. The first three authors of this chapter have fond memories of Peter's visits (and are not the only ones who have lugged around Peter's considerable suitcase when regular cabs were not up to the task [25]) – as Milos Ercegovac put it: a modest man with a sense of humor, a gentle chuckle when people make silly technical comments [21]. Numerous papers were the result of his visits: several record factorizations (140 digits [14], 512 bits [15], 768 bits [27, 28], and Mersenne numbers [8]); security estimates [5]; elliptic curve discrete logarithm cryptanalysis using Playstation 3 game consoles [7, 6]; improved stage two for Pollard's $p \pm 1$ method [42]; better elliptic curve factoring parameter selection [2]; and results on the period of Bell numbers [43].

While discussing his work at SDC in the early software industry Peter liked to proudly show a copy of "The System Builders: The Story of SDC" [3] with his name displayed on the front cover, about an inch underneath the title (cf. Figure 1.3). Peter never showed a similar pride in his more celebrated algorithmic and mathematical results – possibly because his work as a software engineer greatly motivated his mathematical research and he invented new approaches just to make his implementations more efficient. This is illustrated by Peter's explanation how he got the inspiration for his paper *Modular multiplication without trial division* [36], arguably one of his most widely-used results: when implementing textbook multiprecision multiplication in assembler on a PDP series computer, he noticed there were unused registers and wanted to find a way to exploit them. He managed to do so by interleaving the multiplication with the modular reduction.

When Peter began working, at SDC, he programmed using punch cards on time-shared main frame computers. By the time he had retired from MSR, he had implemented and optimized arithmetic code that was running on the mobile processors in smartphones. Peter worked as a programmer from the earliest stages of software engineering when computers needed to be held in their own buildings until the era of ubiquitous computing where people carry computing devices in their pockets. Both his algorithmic and mathematical contributions and his work as a programmer were instrumental to these massive advances that spanned his career – a career of more than forty years, impressive by the standards of the software industry.

1.3 Overview

The subsequent chapters of this book are on active research areas that were initiated by Peter or to which he has made substantial contributions. This section contains an overview of the subjects that are treated.

As mentioned above, Peter's 1985 method for modular multiplication without trial division [36] was invented almost serendipitously when Peter tried to put otherwise idle registers to good use. Because it allows simple implementation both in software and in hardware, *Montgomery multiplication* (as the method is now commonly referred to) has found broad application. Its software and hardware aspects are covered in chapters 2 and 3, respectively.

Peter's contributions as described and expanded upon in chapters 4 through 8 were all inspired by integer factorization, his main research interest since high school. In [37, Section 10.3.1] Peter describes how his original implementation of the elliptic curve method of integer factorization [33] uses the Weierstrass

equation with affine coordinates, and how the modular inversion required for the point addition can be made faster if several curves are run at once (cf. Section 1.4 on page 8). He then writes "The author later discovered an alternative parametrization that requires no inversions" and continues with the description of what is now known as *Montgomery curves* and the *Montgomery ladder*. The former is an elliptic curve parametrization that allows fast computation of the sum of two points if their difference is known and that does not require modular inversion. The latter refers to the addition chain that can then be used to compute any scalar multiple of a point. Initially these results were relevant solely to those implementing the elliptic curve integer factorization method, but the later growing importance of elliptic curve cryptography increased the interest in Peter's and yet other curve parametrizations and addition chains. Chapter 4 describes the developments in this field that have taken place since the publication of [37].

In parallel with his work on special purpose integer factoring methods in [37], Peter also heavily influenced the two most important methods of the more generic integer factorization algorithms that are described in Chapter 5: the *quadratic sieve* and the *number field sieve*, two methods that were initially plagued by severe performance problems [48, 31]. Although Peter was not the first to propose a practical version of the quadratic sieve, his *multiple polynomial quadratic sieve* (as published by Robert Silverman in [54] and described in Chapter 5), represented the state of the art in integer factorization from the mid 1980s until the early 1990s. It was later enhanced by *self-initialization* [50], but by that time the number field sieve had taken over as the record-breaking factoring method, a position it holds to the present day. Peter played a prominent role in that development, contributing important ideas to all steps of the number field sieve: polynomial selection, (early) sieving, building a matrix, processing the matrix, and the final square root calculation. A brief summary follows, with Chapter 5 describing in more detail how the steps fit together.

In the first step of the number field sieve two or more polynomials must be found that have a root in common modulo the number to be factored. Though theoretically such polynomials are easily found, small modifications may trigger orders of magnitude reductions of the resulting factoring effort. Finding good polynomials has become a subject in its own right since Peter's early contributions (many of which were first published in Brian Murphy's 1999 PhD dissertation [45]). Chapter 6 is entirely devoted to the current state of the art of polynomial selection for the number field sieve.

The number field sieve (and many other sieving-based integer factorization methods) relies on the ability to quickly recognize which values in an arith-

metic progression have many small factors. Chapter 5 briefly mentions some of Peter's many tricks which are now part of the standard repertoire to speed up this so-called *sieving* step. His improvements to the earliest *line sieving* software for the number field sieve by the second author were used for numerous integer factorizations in the 1990s [10, 32]. For most composites, however, line sieving has since then been surpassed by *lattice sieving* [47].

The sieving step results in a large, very sparse matrix. In the matrix step dependencies among the columns of this matrix must be found. Given its sparsity, it is advantageous to first transform the matrix into an equivalent matrix that is smaller but less sparse. Originally, when regular Gaussian elimination was still used for the matrix step, a set of ad hoc methods referred to as structured Gaussian elimination [29, 49] was used to find a transformed matrix with a smallest possible number of rows, but without paying attention to its density. The more sophisticated tools that became available for the matrix step in the mid 1990s required an adaption of the matrix transformation method that would minimize the product of the number of rows of the matrix and its number of non-zero entries. These days this is referred to as *filtering*. The details of the various filtering steps proposed by Peter (and based on the earlier work in [29, 49]) can be found in [12, 13] and are sketched in Chapter 5.

The above "more sophisticated tools" are the *block Wiedemann* and *block Lanczos* methods, based on the classical Wiedemann and Lanczos methods. They were almost simultaneously and independently developed by Don Coppersmith and Peter, respectively, and have comparable performance [17, 40]. Though both methods allow efficient distributed implementation, for block Lanczos all processors must be able to communicate quickly (i.e., a single cluster with a fast interconnection network), whereas block Wiedemann can be made to run efficiently on a modest number of disjoint clusters. Almost since its inception in the mid 1990s, block Lanczos became the method of choice for all academic factoring projects. Around 2005 it became more convenient for these applications to use a number of separate clusters for the matrix step, which implied a switch to block Wiedemann [27, 28]. Chapter 7 describes the block Lanczos method in detail.

For the number field sieve each column dependency produced by the matrix step requires a square root calculation in the number fields involved before a factorization of the targeted composite may be derived. In [39] Peter presented an iterative approach that cleverly exploits the special form of the squared algebraic numbers, thereby replacing Jean-Marc Couveignes' much slower and more restrictive method [18]. Peter's method is used, essentially unmodified, until the present day, and sketched in Chapter 5.

Returning to special purpose integer factorization, in the last paragraph of

his $p-1$ paper [46] John Pollard mentions "we would theoretically use the fast Fourier transform on Step 2", concluding his paper with "I cannot say at what point this becomes a practical proposition". In [37, Section 4] Peter refers to Pollard's remark, and further pursues the idea for Pollard's $p-1$ method, first with Silverman in [44] and later with Alexander Kruppa in [42]. For the elliptic curve factoring method it is the subject of Peter's PhD dissertation [38]. Peter's work on this subject is presented in Chapter 8.

In Chapter 9 Peter's contributions to cryptographic pairings are outlined. Given Peter's simultaneous inversion method (described below in Section 1.4) and his fast modular inversion it became attractive, contrary to conventional wisdom, to replace projective coordinates by affine ones for applications involving multiple or parallelized pairings, products of pairings, or pairings at high security levels [19, 20, 30]. Moreover, Peter coauthored a paper introducing algorithms to compute the squared pairing, which has the advantage, compared to Victor Miller's algorithm, that its computation cannot fail [34].

1.4 Simultaneous inversion

For completeness, this section describes Peter Montgomery's simultaneous inversion method, a popular method that did not lead to any follow-up work and that is therefore not treated in the later chapters.

At a computational number theory conference held in August 1985 at Humboldt State University in Arcata, CA, Peter gave a presentation about his experience using the elliptic curve integer factorization method ([33], informally announced in February 1985). He proposed to maximize the overall throughput of the method as opposed to minimizing the latency per curve, thus exploiting the method's inherent parallelism. The idea is simple: at a modest overhead the cost of the modular inversion required for each elliptic curve group operation in the Weierstrass model can be shared among any number of simultaneously processed curves, assuming all curves target the same number to be factored. It has become known as *simultaneous inversion*, and was later briefly mentioned in [37, Section 10.3.1]:

> When working over several curves, the program used a scheme similar to (1.2) to do all the inversions at once, since $(1/x) = y(1/xy)$ and $(1/y) = x(1/xy)$. This reduces the asymptotic cost of an inversion to that of 3 multiplications.

The "scheme similar to (1.2)" suggests that Peter took his inspiration from John Pollard's observation that the cost of a number of greatest common divisor calculations with the same to-be-factored modulus can be reduced to that

same number of multiplications modulo the same modulus, plus a single gcd calculation. In the inversion context, there may have been a historical precedent for Peter's trick, with Richard Schroeppel recalling he may have seen a similar method to avoid floating divides in pre-1970s Fortran programs. But even if it is not entirely original, Peter's insight cleverly adapts an old idea to a regime where numerics are exact [53]. These days, simultaneous inversion is, for instance, used in record computations of elliptic curve discrete logarithms (e.g., [6] and the record attempt [1]; it was not used in [55]).

Let $z_1, z_2, \ldots, z_k \in \mathbf{Z}/n\mathbf{Z}$ for some modulus n be the elements that must be inverted; in the rare case that at least one of the inverses does not exist, simultaneous inversion fails, and an appropriate alternative approach is used. Working in $\mathbf{Z}/n\mathbf{Z}$ and with $w_0 = 1$, first for $i = 1, 2, \ldots, k$ in succession calculate and store $w_i = w_{i-1}z_i$. Next, calculate $w = w_k^{-1}$. Finally, for $i = k, k-1, \ldots, 1$ in succession first compute z_i^{-1} as ww_{i-1} and next replace w by wz_i (so that $w = w_{i-1}^{-1}$). The overall cost is $3(k-1)$ multiplications, a single inversion, and storage for k temporary values, all in $\mathbf{Z}/n\mathbf{Z}$. Given that for relevant modulus sizes and software implementation of arithmetic in $\mathbf{Z}/n\mathbf{Z}$ inversion is at least five times costlier than multiplication, the resulting speed up can be significant. One should be aware, though, that in hardware the difference can be made smaller [26].

Bibliography

[1] D. V. Bailey, L. Batina, D. J. Bernstein, P. Birkner, J. W. Bos, H.-C. Chen, C.-M. Cheng, G. van Damme, G. de Meulenaer, L. J. D. Perez, J. Fan, T. Güneysu, F. Gurkaynak, T. Kleinjung, T. Lange, N. Mentens, R. Niederhagen, C. Paar, F. Regazzoni, P. Schwabe, L. Uhsadel, A. V. Herrewege, and B.-Y. Yang. Breaking ECC2K-130. Cryptology ePrint Archive, Report 2009/541, 2009. `http://eprint.iacr.org/2009/541`. (Cited on page 9.)

[2] R. Barbulescu, J. W. Bos, C. Bouvier, T. Kleinjung, and P. L. Montgomery. Finding ECM-friendly curves through a study of Galois properties. *The Open Book Series – Proceedings of the Tenth Algorithmic Number Theory Symposium*, pages 63–86, 2013. (Cited on page 4.)

[3] C. Baum. *The System Builders: The Story of SDC*. System Development Corporation, 1981. (Cited on pages 4 and 5.)

[4] E. R. Berlekamp. *Algebraic coding theory*. McGraw-Hill, 1968. (Cited on page 3.)

[5] J. W. Bos, M. E. Kaihara, T. Kleinjung, A. K. Lenstra, and P. L. Montgomery. On the security of 1024-bit RSA and 160-bit elliptic curve cryptography. Cryptology ePrint Archive, Report 2009/389, 2009. `http://eprint.iacr.org/`. (Cited on page 4.)

[6] J. W. Bos, M. E. Kaihara, T. Kleinjung, A. K. Lenstra, and P. L. Montgomery. Solving a 112-bit prime elliptic curve discrete logarithm problem on game consoles using sloppy reduction. *International Journal of Applied Cryptography*, 2(3):212–228, 2012. (Cited on pages 4 and 9.)

[7] J. W. Bos, M. E. Kaihara, and P. L. Montgomery. Pollard rho on the PlayStation 3. In *Special-purpose Hardware for Attacking Cryptographic Systems – SHARCS 2009*, pages 35–50, 2009. `http://www.hyperelliptic.org/tanja/SHARCS/record2.pdf`. (Cited on page 4.)

[8] J. W. Bos, T. Kleinjung, A. K. Lenstra, and P. L. Montgomery. Efficient SIMD arithmetic modulo a Mersenne number. In E. Antelo, D. Hough, and P. Ienne, editors, *IEEE Symposium on Computer Arithmetic – ARITH-20*, pages 213–221. IEEE Computer Society, 2011. (Cited on page 4.)

[9] J. W. Bos, P. L. Montgomery, D. Shumow, and G. M. Zaverucha. Montgomery multiplication using vector instructions. 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 471–489. Springer, Heidelberg, Aug. 2014. (Cited on page 4.)

[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 7.)

[11] D. G. Cantor and H. Zassenhaus. A new algorithm for factoring polynomials over finite fields. *Mathematics of Computation*, 36:587–592, 1981. (Cited on page 3.)

[12] 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 page 7.)

[13] S. Cavallar. *On the number field sieve integer factorisation algorithm*. PhD thesis, Leiden University, 2002. (Cited on page 7.)

[14] S. Cavallar, B. Dodson, A. K. Lenstra, P. C. Leyland, W. M. Lioen, P. L. Montgomery, B. Murphy, H. te Riele, and P. Zimmermann. Factorization of RSA-140 using the number field sieve. In K.-Y. Lam, E. Okamoto, and C. Xing, editors, *Advances in Cryptology – ASIACRYPT'99*, volume 1716 of *Lecture Notes in Computer Science*, pages 195–207. Springer, Heidelberg, Nov. 1999. (Cited on page 4.)

[15] 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 page 4.)

[16] M. Ciet, M. Joye, K. Lauter, and P. L. Montgomery. Trading inversions for multiplications in elliptic curve cryptography. *Des. Codes Cryptography*, 39(2):189–206, 2006. (Cited on page 4.)

[17] D. Coppersmith. Solving homogeneous linear equations over GF(2) via block Wiedemann algorithm. *Mathematics of Computation*, 62(205):333–350, 1994. (Cited on page 7.)

[18] J.-M. Couveignes. Computing a square root for the number field sieve. pages 95–102 in [31], 1992. (Cited on page 7.)

[19] K. Eisenträger, K. Lauter, and P. L. Montgomery. Fast elliptic curve arithmetic and improved Weil pairing evaluation. In M. Joye, editor, *Topics in Cryptology – CT-RSA 2003*, volume 2612 of *Lecture Notes in Computer Science*, pages 343–354. Springer, Heidelberg, Apr. 2003. (Cited on pages 4 and 8.)

[20] K. Eisenträger, K. E. Lauter, and P. L. Montgomery. Improved Weil and Tate pairings for elliptic and hyperelliptic curves. In D. A. Buell, editor, *Algorithmic Number Theory, 6th International Symposium, ANTS-VI, Burlington, VT, USA, June 13-18, 2004, Proceedings*, volume 3076 of *Lecture Notes in Computer Science*, pages 169–183. Springer, 2004. (Cited on pages 4 and 8.)

[21] M. Ercegovac, November 2015. Private communication. (Cited on pages 2 and 4.)

[22] P. Erdös, R. L. Graham, P. L. Montgomery, B. K. Rothschild, J. Spencer, and E. G. Strauss. Euclidean Ramsey theorems, I. *Journal of Combinatorial Theory, Series A*, 14(3):341–363, 1973. (Cited on page 1.)

[23] P. Erdös, R. L. Graham, P. L. Montgomery, B. K. Rothschild, J. Spencer, and E. G. Strauss. Euclidean Ramsey theorems, II. In A. Hajnal, R. Rado, and V. T. Sós, editors, *Colloquia Mathematica Societatis János Bolyai, 10*, volume I of *Infinite and Finite Sets*, pages 529–557. North-Holland, Amsterdam-London, 1975. (Cited on page 1.)

[24] P. Erdös, R. L. Graham, P. L. Montgomery, B. K. Rothschild, J. Spencer, and E. G. Strauss. Euclidean Ramsey theorems, III. In A. Hajnal, R. Rado, and V. T. Sós, editors, *Colloquia Mathematica Societatis János Bolyai, 10*, volume I of *Infinite and Finite Sets*, pages 559–583. North-Holland, Amsterdam-London, 1975. (Cited on page 1.)

[25] R. Graham, November 2015. Private communication. (Cited on pages 1 and 4.)

[26] M. E. Kaihara and N. Takagi. A hardware algorithm for modular multiplication/division. *IEEE Transactions on Computers*, 54(1):12–21, 2005. (Cited on page 9.)

[27] 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 4 and 7.)

[28] T. Kleinjung, J. W. Bos, A. K. Lenstra, D. A. Osvik, K. Aoki, S. Contini, J. Franke, E. Thomé, P. Jermini, M. Thiémard, P. Leyland, P. L. Montgomery, A. Timofeev, and H. Stockinger. A heterogeneous computing environment to solve the 768-bit RSA challenge. *Cluster Computing*, (15):53–68, 2012. (Cited on pages 4 and 7.)

[29] 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 7.)

[30] K. Lauter, P. L. Montgomery, and M. Naehrig. An analysis of affine coordinates for pairing computation. In M. Joye, A. Miyaji, and A. Otsuka, editors, *PAIRING 2010: 4th International Conference on Pairing-based Cryptography*, volume 6487 of *Lecture Notes in Computer Science*, pages 1–20. Springer, Heidelberg, Dec. 2010. (Cited on pages 4 and 8.)

[31] 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 6, 12, 13, and 14.)

[32] A. K. Lenstra, H. W. Lenstra Jr., M. S. Manasse, and J. M. Pollard. The number field sieve. pages 11–42 in [31], 1989. (Cited on page 7.)

[33] H. W. Lenstra Jr. Factoring integers with elliptic curves. *Annals of Mathematics*, 126(3):649–673, 1987. (Cited on pages 5 and 8.)

[34] V. S. Miller. The Weil pairing, and its efficient calculation. *Journal of Cryptology*, 17(4):235–261, Sept. 2004. (Cited on page 8.)

[35] P. L. Montgomery. Evaluation of boolean expressions on one's complement machines. *SIGPLAN Notices*, 13:60–72, 1978. (Cited on page 1.)

[36] P. L. Montgomery. Modular multiplication without trial division. *Mathematics of Computation*, 44(170):519–521, April 1985. (Cited on page 5.)

[37] P. L. Montgomery. Speeding the Pollard and elliptic curve methods of factorization. *Mathematics of Computation*, 48(177):243–264, 1987. (Cited on pages 5, 6, and 8.)

[38] P. L. Montgomery. *An FFT extension of the elliptic curve method of factorization*. PhD thesis, University of California, 1992. (Cited on pages 2 and 8.)

[39] 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 7.)

[40] 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 7.)

[41] P. L. Montgomery. Five, six, and seven-term Karatsuba-like formulae. *IEEE Transactions on Computers*, 54(3):362–369, 2005. (Cited on page 4.)

[42] P. L. Montgomery and A. Kruppa. Improved stage 2 to p±1 factoring algorithms. In A. J. van der Poorten and A. Stein, editors, *Algorithmic Number Theory – ANTS-VIII*, volume 5011 of *Lecture Notes in Computer Science*, pages 180–195. Springer, 2008. (Cited on pages 4 and 8.)

[43] P. L. Montgomery, S. Nahm, and S. S. Wagstaff Jr. The period of the bell numbers modulo a prime. *Mathematics of Computation*, 79(271):1793–1800, 2010. (Cited on page 4.)

[44] P. L. Montgomery and R. D. Silverman. An FFT extension to the $p - 1$ factoring algorithm. *Mathematics of Computation*, 54(190):839–854, 1990. (Cited on page 8.)

[45] B. A. Murphy. *Polynomial selection for the number field sieve integer factorisation algorithm*. PhD thesis, Australian National University, 1999. (Cited on page 6.)

[46] J. M. Pollard. Theorems on factorization and primality testing. *Proceedings of the Cambridge Philosophical Society*, 76:521–528, 1974. (Cited on page 8.)

[47] J. M. Pollard. The lattice sieve. pages 43–49 in [31], 1990. (Cited on page 7.)

[48] 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 page 6.)

[49] 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 page 7.)

[50] 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 6.)

[51] B. Rothschild, November 2015. Private communication. (Cited on page 1.)

[52] M. Schacher, November 2015. Private communication. (Cited on page 3.)

[53] R. Schroeppel, April 2015. Private communication. (Cited on page 9.)

[54] R. D. Silverman. The multiple polynomial quadratic sieve. *Mathematics of Computation*, 48:329–339, 1987. (Cited on page 6.)

[55] E. Wenger and P. Wolfger. Solving the discrete logarithm of a 113-bit Koblitz curve with an FPGA cluster. In A. Joux and A. M. Youssef, editors, *SAC 2014: 21st Annual International Workshop on Selected Areas in Cryptography*, volume 8781 of *Lecture Notes in Computer Science*, pages 363–379. Springer, Heidelberg, Aug. 2014. (Cited on page 9.)

Subject index

addition chain, 6

cryptographic pairing, 8

Diffie–Hellman key exchange, 4
digital signature, 4

elliptic curve
  cryptography, 4
elliptic curve method of factorization, 6

fast Fourier transform extension, 2, 8
  elliptic curve method, 2
  $p - 1$ and $p + 1$ method, 8
FFT, *see* fast Fourier transform
filtering, 7

Gaussian elimination, 7

integer factoring, 5

Lanczos algorithm, 7
latency, 8

Montgomery
  curve, *see* Montgomery curve
  ladder, *see* Montgomery ladder
  multiplication, *see* Montgomery multiplication
Montgomery curve, 6
Montgomery ladder, 6
Montgomery multiplication, 5
msbignum, 4

number field sieve, 6
  polynomial selection, *see* polynomial selection
  square root computation, *see* square root computation for the number field sieve

$p - 1$ method, 8
polynomial selection, 6

quadratic sieve, 6
  multiple polynomial, 6
  self-initializing, 6

Ramsey theory, 1

sieving, 6
  lattice sieving, 7
  line sieving, 7
simultaneous inversion, 8
square root computation for the number field sieve, 6

throughput, 8

Wiedemann algorithm, 7