Computational Number Theory and Modern Cryptography

Song Y. Yan, North China University of Technology, P.R. China and Harvard University, USA

Computational number theory and modern cryptography are two of the most important and fundamental research fields in information security. In this book, Song Y. Yang combines knowledge of these two critical fields, providing a unified view of the relationships between computational number theory and cryptography. The author takes an innovative approach, presenting mathematical ideas first, thereupon treating cryptography as an immediate application of the mathematical concepts. The book also presents topics from number theory, which are relevant for applications in public-key cryptography, as well as modern topics, such as coding and lattice based cryptography for post-quantum cryptography. The author further covers the current research and applications for common cryptographic algorithms, describing the mathematical problems behind these applications in a manner accessible to computer scientists and engineers.

Computational Number Theory and Modern Cryptography is ideal for graduate and advanced undergraduate students in computer science, communications engineering, cryptography and mathematics. Computer scientists, practicing cryptographers, and other professionals involved in various security schemes will also find this book to be a helpful reference.

Makes mathematical problems accessible to computer scientists and engineers by showing their immediate application

Presents topics from number theory relevant for public-key cryptography applications

Covers modern topics such as coding and lattice based cryptography for post-quantum cryptography

Starts with the basics, then goes into applications and areas of active research

Geared at a global audience; classroom tested in North America, Europe, and Asia

Includes exercises in every chapter

Instructor resources available on the book’s Companion Website

www.wiley.com/go/yan/cryptography
COMPUTATIONAL NUMBER THEORY AND MODERN CRYPTOGRAPHY
INFORMATION SECURITY SERIES

The Wiley-HEP Information Security Series systematically introduces the fundamentals of information security design and application. The goals of the Series are:

- to provide fundamental and emerging theories and techniques to stimulate more research in cryptology, algorithms, protocols, and architectures
- to inspire professionals to understand the issues behind important security problems and the ideas behind the solutions
- to give references and suggestions for additional reading and further study

The Series is a joint project between Wiley and Higher Education Press (HEP) of China. Publications consist of advanced textbooks for graduate students as well as researcher and practitioner references covering the key areas, including but not limited to:

- Modern Cryptography
- Cryptographic Protocols and Network Security Protocols
- Computer Architecture and Security
- Database Security
- Multimedia Security
- Computer Forensics
- Intrusion Detection

LEAD EDITORS

Song Y. Yan  London, UK
Moti Yung  Columbia University, USA
John Rief  Duke University, USA

EDITORIAL BOARD

Liz Bacon  University of Greenwich, UK
Kefei Chen  Shanghai Jiaotong University, China
Matthew Franklin  University of California, USA
Dieter Gollmann  Hamburg University of Technology, Germany
Yongfei Han  Beijing University of Technology, China
ONETS Wireless & Internet Security Tech. Co., Ltd. Singapore
Kwangjo Kim  KAIST-ICC, Korea
David Naccache  Ecole Normale Supérieure, France
Dingyi Pei  Guangzhou University, China
Peter Wild  University of London, UK
COMPUTATIONAL NUMBER THEORY AND MODERN CRYPTOGRAPHY

Song Y. Yan
College of Sciences
North China University of Technology
Beijing, China

&

Department of Mathematics
Harvard University
Cambridge, USA
# CONTENTS

About the Author ix
Preface xi
Acknowledgments xiii

## Part I  Preliminaries

1 Introduction 3
   1.1 What is Number Theory? 3
   1.2 What is Computation Theory? 9
   1.3 What is Computational Number Theory? 15
   1.4 What is Modern Cryptography? 29
   1.5 Bibliographic Notes and Further Reading 32
   References 32

2 Fundamentals 35
   2.1 Basic Algebraic Structures 35
   2.2 Divisibility Theory 46
   2.3 Arithmetic Functions 75
   2.4 Congruence Theory 89
   2.5 Primitive Roots 131
   2.6 Elliptic Curves 141
   2.7 Bibliographic Notes and Further Reading 154
   References 155

## Part II  Computational Number Theory

3 Primality Testing 159
   3.1 Basic Tests 159
   3.2 Miller–Rabin Test 168
   3.3 Elliptic Curve Tests 173
   3.4 AKS Test 178
   3.5 Bibliographic Notes and Further Reading 187
   References 188
# Contents

4 Integer Factorization 191

4.1 Basic Concepts 191
4.2 Trial Divisions Factoring 194
4.3 $\rho$ and $p - 1$ Methods 198
4.4 Elliptic Curve Method 205
4.5 Continued Fraction Method 209
4.6 Quadratic Sieve 214
4.7 Number Field Sieve 219
4.8 Bibliographic Notes and Further Reading 231
References 232

5 Discrete Logarithms 235

5.1 Basic Concepts 235
5.2 Baby-Step Giant-Step Method 237
5.3 Pohlig–Hellman Method 240
5.4 Index Calculus 246
5.5 Elliptic Curve Discrete Logarithms 251
5.6 Bibliographic Notes and Further Reading 260
References 261

Part III Modern Cryptography

6 Secret-Key Cryptography 265

6.1 Cryptography and Cryptanalysis 265
6.2 Classic Secret-Key Cryptography 277
6.3 Modern Secret-Key Cryptography 285
6.4 Bibliographic Notes and Further Reading 291
References 291

7 Integer Factorization Based Cryptography 293

7.1 RSA Cryptography 293
7.2 Cryptanalysis of RSA 302
7.3 Rabin Cryptography 319
7.4 Residuosity Based Cryptography 326
7.5 Zero-Knowledge Proof 331
7.6 Bibliographic Notes and Further Reading 335
References 335

8 Discrete Logarithm Based Cryptography 337

8.1 Diffie–Hellman–Merkle Key-Exchange Protocol 337
8.2 ElGamal Cryptography 342
8.3 Massey–Omura Cryptography 344
8.4 DLP-Based Digital Signatures 348
8.5 Bibliographic Notes and Further Reading 351
References 351
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>Elliptic Curve Discrete Logarithm Based Cryptography</td>
<td>353</td>
</tr>
<tr>
<td>9.1</td>
<td>Basic Ideas</td>
<td>353</td>
</tr>
<tr>
<td>9.2</td>
<td>Elliptic Curve Diffie–Hellman–Merkle Key Exchange Scheme</td>
<td>356</td>
</tr>
<tr>
<td>9.3</td>
<td>Elliptic Curve Massey–Omura Cryptography</td>
<td>360</td>
</tr>
<tr>
<td>9.4</td>
<td>Elliptic Curve ElGamal Cryptography</td>
<td>365</td>
</tr>
<tr>
<td>9.5</td>
<td>Elliptic Curve RSA Cryptosystem</td>
<td>370</td>
</tr>
<tr>
<td>9.6</td>
<td>Menezes–Vanstone Elliptic Curve Cryptography</td>
<td>371</td>
</tr>
<tr>
<td>9.7</td>
<td>Elliptic Curve DSA</td>
<td>373</td>
</tr>
<tr>
<td>9.8</td>
<td>Bibliographic Notes and Further Reading</td>
<td>374</td>
</tr>
<tr>
<td></td>
<td>References</td>
<td>375</td>
</tr>
<tr>
<td>**</td>
<td><strong>Part IV  Quantum Resistant Cryptography</strong></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Quantum Computational Number Theory</td>
<td>379</td>
</tr>
<tr>
<td>10.1</td>
<td>Quantum Algorithms for Order Finding</td>
<td>379</td>
</tr>
<tr>
<td>10.2</td>
<td>Quantum Algorithms for Integer Factorization</td>
<td>385</td>
</tr>
<tr>
<td>10.3</td>
<td>Quantum Algorithms for Discrete Logarithms</td>
<td>390</td>
</tr>
<tr>
<td>10.4</td>
<td>Quantum Algorithms for Elliptic Curve Discrete Logarithms</td>
<td>393</td>
</tr>
<tr>
<td>10.5</td>
<td>Bibliographic Notes and Further Reading</td>
<td>397</td>
</tr>
<tr>
<td></td>
<td>References</td>
<td>397</td>
</tr>
<tr>
<td>11</td>
<td>Quantum Resistant Cryptography</td>
<td>401</td>
</tr>
<tr>
<td>11.1</td>
<td>Coding-Based Cryptography</td>
<td>401</td>
</tr>
<tr>
<td>11.2</td>
<td>Lattice-Based Cryptography</td>
<td>403</td>
</tr>
<tr>
<td>11.3</td>
<td>Quantum Cryptography</td>
<td>404</td>
</tr>
<tr>
<td>11.4</td>
<td>DNA Biological Cryptography</td>
<td>406</td>
</tr>
<tr>
<td>11.5</td>
<td>Bibliographic Notes and Further Reading</td>
<td>409</td>
</tr>
<tr>
<td></td>
<td>References</td>
<td>410</td>
</tr>
<tr>
<td>**</td>
<td><strong>Index</strong></td>
<td>413</td>
</tr>
</tbody>
</table>
ABOUT THE AUTHOR

Professor Song Y. Yan majored in both Computer Science and Mathematics, and obtained a PhD in Number Theory in the Department of Mathematics at the University of York, England. His current research interests include Computational Number Theory, Computational Complexity Theory, Algebraic Coding Theory, Public-Key Cryptography and Information/Network Security. He published, among others, the following five well-received and popular books in computational number theory and public-key cryptography:


Song can be reached by email address songyuanyan@gmail.com anytime.
PREFACE

The book is about number theory and modern cryptography. More specifically, it is about computational number theory and modern public-key cryptography based on number theory. It consists of four parts. The first part, consisting of two chapters, provides some preliminaries. Chapter 1 provides some basic concepts of number theory, computation theory, computational number theory, and modern public-key cryptography based on number theory. In chapter 2, a complete introduction to some basic concepts and results in abstract algebra and elementary number theory is given.

The second part is on computational number theory. There are three chapters in this part. Chapter 3 deals with algorithms for primality testing, with an emphasis on the Miller-Rabin test, the elliptic curve test, and the AKS test. Chapter 4 treats with algorithms for integer factorization, including the currently fastest factoring algorithm NFS (Number Field Sieve), and the elliptic curve factoring algorithm ECM (Elliptic Curve Method). Chapter 5 discusses various modern algorithms for discrete logarithms and for elliptic curve discrete logarithms. It is well-known now that primality testing can be done in polynomial-time on a digital computer, however, integer factorization and discrete logarithms still cannot be performed in polynomial-time. From a computational complexity point of view, primality testing is feasible (tractable, easy) on a digital computer, whereas integer factorization and discrete logarithms are infeasible (intractable, hard, difficult). Of course, no-one has yet been able to prove that the integer factorization and the discrete logarithm problems must be infeasible on a digital computer.

Building on the results in the first two parts, the third part of the book studies the modern cryptographic schemes and protocols whose security relies exactly on the infeasibility of the integer factorization and discrete logarithm problems. There are four chapters in this part. Chapter 6 presents some basic concepts and ideas of secret-key cryptography. Chapter 7 studies the integer factoring based public-key cryptography, including, among others, the most famous and widely used RSA cryptography, the Rabin cryptosystem, the probabilistic encryption and the zero-knowledge proof protocols. Chapter 8 studies the discrete logarithm based cryptography, including the DHM key-exchange protocol (the world’s first public-key system), the ElGamal cryptosystem, and the US Government’s Digital Signature Standard (DSS). Chapter 9 discusses various cryptographic systems and digital signature schemes based on the infeasibility of the elliptic curve discrete logarithm problem, some of them are just the elliptic curve analogues of the ordinary public-key cryptography such as elliptic curve DHM, elliptic curve ElGamal, elliptic curve RSA, and elliptic curve DSA/DSS.
It is interesting to note that although integer factorization and discrete logarithms cannot be solved in polynomial-time on a classical digital computer, they all can be solved in polynomial-time on a quantum computer, provided that a practical quantum computer with several thousand quantum bits can be built. So, the last part of the book is on quantum computational number theory and quantum-computing resistant cryptography. More specifically, in Chapter 10, we shall study efficient quantum algorithms for solving the Integer Factorization Problem (IFP), the Discrete Logarithm Problem (DLP) and the Elliptic Curve Discrete Logarithm Problem (ECDLP). Since IFP, DLP and ECDLP can be solved efficiently on a quantum computer, the IFP, DLP and ECDLP based cryptographic systems and protocols can be broken efficiently on a quantum computer. However, there are many infeasible problems such as the coding-based problems and the lattice-based problems that cannot be solved in polynomial-time even on a quantum computer. That is, a quantum computer is basically a special type of computing device using a different computing paradigm, it is only suitable or good for some special problems such as the IFP, DLP and ECDLP problems. Thus, in chapter 11, the last chapter of the book, we shall discuss some quantum-computing resistant cryptographic systems, including the coding-based and lattice-based cryptographic systems, that resist all known quantum attacks. Note that quantum-computing resistant cryptography is still classic cryptography, but quantum resistant. We shall, however, also introduce a truly quantum cryptographic scheme, based on ideas of quantum mechanics and some DNA cryptographic schemes based on idea of DNA molecular computation.

The materials presented in the book are based on the author’s many years teaching and research experience in the field, and also based on the author’s other books published in the past ten years or so, particularly the following three books, all by Springer:


The book is suited as a text for final year undergraduate or first year postgraduate courses in computational number theory and modern cryptography, or as a basic research reference in the field.

Corrections, comments and suggestions from readers are very welcomed and can be sent via email to songyuanyan@gmail.com.

Song Y. Yan
London, England
June 2012
ACKNOWLEDGMENTS

The author would like to thank the editors at Wiley and HEP, particularly Hongying Chen, Shelley Chow, James Murphy, Clarissa Lim, and Shalini Sharma, for their encouragement, assistance, and proof-reading. Special thanks must also be given to the three anonymous referees for their very helpful and constructive comments and suggestions.

The work was supported in part by the Royal Society London, the Royal Academy of Engineering London, the Recruitment Program of Global Experts of Hubei Province, the Funding Project for Academic Human Resources Development in Institutions of Higher Learning under the Jurisdiction of the Beijing Municipality (PHR/IHLB), the Massachusetts Institute of Technology and Harvard University.
In this part, we shall first explain what are number theory, computation theory, computational number theory, and modern (number-theoretic) cryptography are. The relationship between them may be shown in the following figure:

Then we shall present an introduction to the elementary theory of numbers from an algebraic perspective (see the following figure), that shall be used throughout the book.
1

Introduction

In this chapter, we present some basic concepts and ideas of number theory, computation theory, computational number theory, and modern (number-theoretic) cryptography. More specifically, we shall try to answer the following typical questions in the field:

- What is number theory?
- What is computation theory?
- What is computational number theory?
- What is modern (number-theoretic) cryptography?

1.1 What is Number Theory?

Number theory is concerned mainly with the study of the properties (e.g., the divisibility) of the integers

\[ \mathbb{Z} = \{ \ldots, -3, -2, -1, 0, 1, 2, 3, \ldots \}, \]

particularly the positive integers

\[ \mathbb{Z}^+ = \{ 1, 2, 3, \ldots \}. \]

For example, in divisibility theory, all positive integers can be classified into three classes:

1. Unit: 1.
2. Prime numbers: 2, 3, 5, 7, 11, 13, 17, 19, \ldots.
3. Composite numbers: 4, 6, 8, 9, 10, 12, 14, 15, \ldots.

Recall that a positive integer \( n > 1 \) is called a prime number, if its only divisors are 1 and \( n \), otherwise, it is a composite number. 1 is neither prime number nor composite number. Prime numbers play a central role in number theory, as any positive integer \( n > 1 \) can be written uniquely into the following standard prime factorization form:

\[ n = p_1^{a_1} p_2^{a_2} \cdots p_k^{a_k} \quad (1.1) \]
where \( p_1 < p_2 < \cdots < p_k \) are primes and \( \alpha_1, \alpha_2, \cdots, \alpha_k \) positive integers. Although prime numbers have been studied for more than 2000 years, there are still many open problems about their distribution. Let us investigate some of the most interesting problems about prime numbers.

1. The distribution of prime numbers.

Euclid proved 2000 years ago in his *Elements* that there were infinitely many prime numbers. That is, the sequence of prime numbers

\[
2, 3, 5, 7, 11, 13, 17, 19, \ldots
\]

is endless. For example, 2, 3, 5 are the first three prime numbers, whereas \( 2^{43112609} - 1 \) is the largest prime number to date, it has 12978189 digits and was found on 23 August 2008. Let \( \pi(x) \) denote the prime numbers up to \( x \) (Table 1.1 gives some values of \( \pi(x) \) for some large \( x \)), then Euclid’s theorem of infinitude of primes actually says that

\[
\pi(x) \to \infty, \quad \text{as } x \to \infty.
\]

A much better result about the distribution of prime numbers is the Prime Number theorem, stating that

\[
\pi(x) \sim x / \log x.
\]  
\[(1.2)\]

In other words,

\[
\lim_{x \to \infty} \frac{\pi(x)}{x / \log x} = 1.
\]  
\[(1.3)\]

Note that the log is the natural logarithm \( \log_e \) (normally denoted by \( \ln \)), where \( e = 2.7182818 \ldots \). However, if the Riemann Hypothesis [3] is true, then there is a refinement of the Prime Number theorem

\[
\pi(x) = \int_{2}^{x} \frac{dt}{\log t} + \mathcal{O} \left( x e^{-c \sqrt{\log x}} \right)
\]  
\[(1.4)\]
to the effect that
\[ \pi(x) = \int_2^x \frac{dt}{\log t} + O(\sqrt{x} \log x). \] (1.5)

Of course we do not know if the Riemann Hypothesis is true. Whether or not the Riemann Hypothesis is true is one of the most important open problems in mathematics, and in fact it is one of the seven Millennium Prize Problems proposed by the Clay Mathematics Institute in Boston in 2000, each with a one million US dollars prize [4]. The Riemann hypothesis states that all the nontrivial (complex) zeros \( \rho \) of the \( \zeta \) function
\[ \zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s}, \quad s = \sigma + it, \quad \sigma, \ t \in \mathbb{R}, \ i = \sqrt{-1} \] (1.6)

lying in the critical strip \( 0 < \text{Re}(s) < 1 \) must lie on the critical line \( \text{Re}(s) = \frac{1}{2} \), that is, \( \rho = \frac{1}{2} + it \), where \( \rho \) denotes a nontrivial zero of \( \zeta(s) \). Riemann calculated the first five nontrivial zeros of \( \zeta(s) \) and found that they all lie on the critical line (see Figure 1.1), he then conjectured that all the nontrivial zeros of \( \zeta(s) \) are on the critical line.

![Figure 1.1 Riemann hypothesis](image-url)
Table 1.2  Ten large twin prime pairs

<table>
<thead>
<tr>
<th>Rank</th>
<th>Twin primes</th>
<th>Digits</th>
<th>Discovery date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>65516468355 · 2^{333333} ± 1</td>
<td>100355</td>
<td>Aug 2009</td>
</tr>
<tr>
<td>2</td>
<td>2003663613 · 2^{195000} ± 1</td>
<td>58711</td>
<td>Jan 2007</td>
</tr>
<tr>
<td>3</td>
<td>194772106074315 · 2^{171960} ± 1</td>
<td>51780</td>
<td>Jun 2007</td>
</tr>
<tr>
<td>4</td>
<td>100314512544015 · 2^{171960} ± 1</td>
<td>51780</td>
<td>Jun 2006</td>
</tr>
<tr>
<td>5</td>
<td>16869987339975 · 2^{171960} ± 1</td>
<td>51779</td>
<td>Sep 2005</td>
</tr>
<tr>
<td>6</td>
<td>33218925 · 2^{169690} ± 1</td>
<td>51090</td>
<td>Sep 2002</td>
</tr>
<tr>
<td>7</td>
<td>22835841624 · 7^{54321} ± 1</td>
<td>45917</td>
<td>Nov 2010</td>
</tr>
<tr>
<td>8</td>
<td>12378188145 · 2^{140002} ± 1</td>
<td>42155</td>
<td>Dec 2010</td>
</tr>
<tr>
<td>9</td>
<td>23272426305 · 2^{140001} ± 1</td>
<td>42155</td>
<td>Dec 2010</td>
</tr>
<tr>
<td>10</td>
<td>8151728061 · 2^{125987} ± 1</td>
<td>37936</td>
<td>May 2010</td>
</tr>
</tbody>
</table>

2. The distribution of twin prime numbers.
Twin prime numbers are of the form \( n \pm 1 \), where both numbers are prime. For example, \((3, 5), (5, 7), (11, 13)\) are the first three smallest twin prime pairs, whereas the largest twin primes so far are 65516468355 · 2^{333333} ± 1, discovered in August 2009, both numbers having 100355 digits. Table 1.2 gives 10 large twin prime pairs. Let \( \pi_2(x) \) be the number of twin primes up to \( x \) (Table 1.3 gives some values of \( \pi_2(x) \) for different \( x \)), then the twin prime conjecture states that

\[
\pi_2(x) \to \infty, \quad \text{as } x \to \infty.
\]

If the probability of a random integer \( x \) and the integer \( x + 2 \) being prime were statistically independent, then it would follow from the prime number theorem that

\[
\pi_2(x) \sim \frac{x}{(\log x)^2}, \quad (1.7)
\]

or more precisely,

\[
\pi_2(x) \sim c \frac{x}{(\log x)^2}, \quad (1.8)
\]

with

\[
c = 2 \prod_{p \geq 3} \left(1 - \frac{1}{(p - 1)^2}\right). \quad (1.9)
\]

Table 1.3  \( \pi_2(x) \) for some large values

<table>
<thead>
<tr>
<th>( x )</th>
<th>( 10^6 )</th>
<th>( 10^7 )</th>
<th>( 10^8 )</th>
<th>( 10^9 )</th>
<th>( 10^{10} )</th>
<th>( 10^{11} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \pi_2(x) )</td>
<td>8169</td>
<td>58980</td>
<td>440312</td>
<td>3424506</td>
<td>27412679</td>
<td>224376048</td>
</tr>
</tbody>
</table>
As these probabilities are not independent, so Hardy and Littlewood conjectured that

\[
\pi_2(x) = 2 \prod_{p \geq 3} \frac{p(p - 2)}{(p - 1)^2} \int_2^x \frac{dt}{(\log t)^2} \approx 1.320323632 \int_2^x \frac{dt}{(\log t)^2}.
\]

The infinite product in the above formula is the twin prime constant; this constant was estimated to be approximately 0.6601618158… Using very complicated arguments based on sieve methods, in his work on the Goldbach conjecture, the Chinese mathematician Chen showed that there are infinitely many pairs of integers \((n, n + 2)\), with \(n\) prime and \(n + 2\) a product of at most two primes. The famous Goldbach conjecture states that every even number greater than 4 is the sum of two odd prime numbers. It was conjectured by Goldbach in a letter to Euler in 1742. It remains unsolved to this day. The best result for this conjecture is due to Chen, who announced it in 1966, but the full proof was not given until 1973 due to the chaotic Cultural Revolution, that every sufficiently large even number is the sum of one prime number and the product of at most two prime numbers, that is, \(E = p_1 + p_2 p_3\), where \(E\) is a sufficiently large even number and \(p_1, p_2, p_3\) are prime numbers. As a consequence, there are infinitely many such twin numbers \((p_1, p_1 + 2 = p_2 p_3)\). Extensions relating to the twin prime numbers have also been considered. For example, are there infinitely many triplet primes \((p, q, r)\) with \(q = p + 2\) and \(r = p + 6\)? The first five triplets of this form are as follows: \((5, 7, 11)\), \((11, 13, 17)\), \((17, 19, 23)\), \((41, 43, 47)\), \((101, 103, 107)\). The triplet prime problem is much harder than the twin prime problem. It is amusing to note that there is only one triplet prime \((p, q, r)\) with \(q = p + 2\) and \(r = p + 4\). That is, \((3, 5, 7)\). The Riemann Hypothesis, the Twin Prime Problem, and the Goldbach conjecture form the famous Hilbert’s 8th Problem.

3. The distribution of arithmetic progressions of prime numbers.

An arithmetic progression of prime numbers is defined to be the sequence of primes satisfying:

\[
p, p + d, p + 2d, \ldots, p + (k - 1)d
\]

where \(p\) is the first term, \(d\) the common difference, and \(p + (k - 1)d\) the last term of the sequence. For example, the following are some sequences of the arithmetic progression of primes:

\[
\begin{array}{cccc}
3 & 5 & 7 \\
5 & 11 & 17 & 23 \\
5 & 11 & 17 & 23 & 29 \\
\end{array}
\]

The longest arithmetic progression of primes is the following sequence with 23 terms: 56211383760397 + k·44546738095860 with \(k = 0, 1, \ldots, 22\). Thanks to Green and Tao who proved in 2007 that there are arbitrary long arithmetic progressions of primes (i.e., \(k\) can be any arbitrary large natural number), which enabled, among others, Tao to receive a Field Prize in 2006, the equivalent to a Nobel Prize for Mathematics. However, their result is not about consecutive primes; we still do not know
if there are arbitrary long arithmetic progressions of consecutive primes, although
Chowa proved in 1944 that there exists an infinity of three consecutive primes of arith-
etic progressions. Note that an arithmetic progression of consecutive primes is a se-
quency of consecutive primes in the progression. In 1967, Jones, Lal, and Blundon
found an arithmetic progression of five consecutive primes $10^{10} + 24493 + 30k$ with
$k = 0, 1, 2, 3, 4$. In the same year, Lander and Parkin discovered six in an arithmetic
progression $121174811 + 30k$ with $k = 0, 1, 2, 3, 4, 5$. The longest arithmetic progres-
sion of consecutive primes, discovered by Manfred Toplic in 1998, is $507618446770482 \cdot
193# + x^{77} + 210k$, where $193#$ is the product of all primes $\leq 193$, that is, $193# =
2 \cdot 3 \cdot 5 \cdot 7 \cdots 193$, $x^{77}$ is a 77-digit number $545382416838875826681897035901
10659057865934764604873840781923513421103495579$ and $k = 0, 1, 2, \cdots, 9$.

It should be noted that problems in number theory are easy to state, because they are mainly
concerned with integers with which we are very familiar, but often very hard to solve!

**Problems for Section 1.1**

1. Show that there are infinitely many prime numbers.
2. Prove or disprove there are infinitely many twin prime numbers.
3. Are there infinitely many triple prime numbers of the form $p, p + 2, p + 4$, where
   $p, p + 2, p + 4$ are all prime numbers? For example, $3, 5, 7$ are such triple prime
   numbers.
4. Are there infinitely many triple prime numbers of the form $p, p + 2, p + 6$, where
   $p, p + 2, p + 6$ are all prime numbers? For example, $5, 7, 11$ are such triple prime
   numbers.
5. (Prime Number Theorem) Show that
   $\lim_{x \to \infty} \frac{\pi(x)}{x / \log x} = 1$.
6. The Riemann $\zeta$-function is defined as follows:
   $$\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s}$$
   where $s = \sigma + it$ is a complex number. Riemann conjectured that all zeroes of $\zeta(s)$ in
   the critical strip $0 \leq \sigma \leq 1$ must lie on the critical line $\sigma = \frac{1}{2}$. That is,
   $$\zeta \left( \frac{1}{2} + it \right) = 0.$$
   Prove or disprove the Riemann Hypothesis.
7. Andrew Beal in 1993 conjectured that the equation $x^a + y^b = z^c$ has no positive integer
   solutions in $x, y, z, a, b, c$, where $a, b, c \geq 3$ and $\gcd(x, y) = \gcd(y, z) = \gcd(x, z) = 1$. Beal
   has offered $100,000 for a proof or a disproof of this conjecture.
8. Prove or disprove the Goldbach conjecture that any even number greater than 6 is the sum of two odd prime numbers.

9. A positive integer \( n \) is perfect if \( \sigma(n) = 2n \), where \( \sigma(n) \) is the sum of all divisors of \( n \).
   For example, 6 is perfect since \( \sigma(6) = 1 + 2 + 3 + 6 = 2 \cdot 6 = 12 \). Show \( n \) is perfect if and only if \( n = 2^{p-1}(2^p - 1) \), where \( 2^p - 1 \) is a Mersenne prime.

10. All known perfect numbers are even perfect. Recent research shows that if there exists an odd perfect number, it must be greater than \( 10^{300} \) and must have at least 29 prime factors (not necessarily distinct). Prove or disprove that there exists at least one odd perfect number.

11. Show that there are arbitrary long arithmetic progressions of prime numbers

\[ p, p + d, p + 2d, \ldots, p + (k - 1)d \]

where \( p \) is the first term, \( d \) the common difference, and \( p + (k - 1)d \) the last term of the sequence, and furthermore, all the terms in the sequence are prime numbers and \( k \) can be any arbitrary large positive integer.

12. Prove or disprove that there are arbitrary long arithmetic progressions of consecutive prime numbers.

### 1.2 What is Computation Theory?

Computation theory, or the theory of computation, is a branch that deals with whether and how efficiently problems can be solved on a model of computation, using an algorithm. It may be divided into two main branches: Computability theory and computational complexity theory. Generally speaking, computability theory deals with what a computer can or cannot do theoretically (i.e., without any restrictions), whereas complexity theory deals with what computer can or cannot do practically (with e.g., time or space limitations). Feasibility or infeasibility theory is a subfield of complexity theory, which concerns itself with what a computer can or cannot do efficiently in polynomial-time. A reasonable model of computation is the Turing machine, first studied by the great British logician and mathematician Alan Turing in 1936, we shall first introduce the basic concepts of Turing machines, then discuss complexity, feasibility, and infeasibility theories based on Turing machines.

**Definition 1.1** A standard multitape Turing machine, \( M \) (see Figure 1.2), is an algebraic system defined by

\[ M = (Q, \Sigma, \Gamma, \delta, q_0, \Box, F) \]  

(1.12)

where

1. \( Q \) is a finite set of internal states;
2. \( \Sigma \) is a finite set of symbols called the input alphabet. We assume that \( \Sigma \subseteq \Gamma - \{ \Box \} \);
3. \( \Gamma \) is a finite set of symbols called the tape alphabet;
4. \( \delta \) is the transition function, which is defined by
   
   (i) if \( M \) is a deterministic Turing machine (DTM), then
   
   \[
   \delta : Q \times \Gamma^k \rightarrow Q \times \Gamma^k \times \{L, R\}^k
   \]  

   (1.13)

   (ii) if \( M \) is a nondeterministic Turing machine (NDTM), then
   
   \[
   \delta : Q \times \Gamma^k \rightarrow 2^{Q \times \Gamma^k \times \{L, R\}^k}
   \]  

   (1.14)

   where \( L \) and \( R \) specify the movement of the read-write head left or right. When \( k = 1 \), it is just a standard one-tape Turing machine;

5. \( \square \in \Gamma \) is a special symbol called the blank;

6. \( q_0 \in Q \) is the initial state;

7. \( F \subseteq Q \) is the set of final states.

Thus, Turing machines provide us with the simplest possible abstract model of computation for modern digital (even quantum) computers.

Any effectively computable function can be computed by a Turing machine, and there is no effective procedure that a Turing machine cannot perform. This leads naturally to the following famous Church–Turing thesis, named after Alonzo Church (1903–1995) and Alan Turing (1912–1954):

**The Church–Turing thesis**: Any effectively computable function can be computed by a Turing machine.

The Church–Turing thesis thus provides us with a powerful tool to distinguish what is computation and what is not computation, what function is computable and what function...
introduction

1.3 Probabilistic $k$-tape ($k \geq 1$) Turing machine

is not computable, and more generally, what computers can do and what computers cannot
do. From a computer science and particularly a cryptographic point of view, we are not
just interested in what computers can do, but in what computers can do efficiently. That is,
in cryptography we are more interested in practical computable rather than just theoretical
computable; this leads to the Cook–Karp thesis.

**Definition 1.2** A probabilistic Turing machine is a type of nondeterministic Turing machine
with distinct states called coin-tossing states. For each coin-tossing state, the finite control
unit specifies two possible legal next states. The computation of a probabilistic Turing
machine is deterministic except that in coin-tossing states the machine tosses an unbiased
coin to decide between the two possible legal next states.

A probabilistic Turing machine can be viewed as a randomized Turing machine, as
described in Figure 1.3. The first tape, holding input, is just the same as conventional
multitape Turing machine. The second tape is referred to as random tape, containing ran-
domly and independently chosen bits, with probability $1/2$ of a 0 and the same probability
$1/2$ of a 1. The third and subsequent tapes are used, if needed, as scratch tapes by the
Turing machine.

**Definition 1.3** $\mathcal{P}$ is the class of problems solvable in polynomial-time by a deterministic
Turing machine (DTM). Problems in this class are classified to be tractable (feasible) and
eyasy to solve on a computer. For example, additions of any two integers, no matter how big
they are, can be performed in polynomial-time, and hence are is in $\mathcal{P}$.

**Definition 1.4** $\mathcal{NP}$ is the class of problems solvable in polynomial-time on a nondeter-
ministic Turing machine (NDTM). Problems in this class are classified to be intractable
Computational Number Theory and Modern Cryptography

(infeasible) and hard to solve on a computer. For example, the Traveling Salesman Problem (TSP) is in \( \mathcal{NP} \), and hence it is hard to solve.

In terms of formal languages, we may also say that \( \mathcal{P} \) is the class of languages where the membership in the class can be decided in polynomial-time, whereas \( \mathcal{NP} \) is the class of languages where the membership in the class can be verified in polynomial-time. It seems that the power of polynomial-time verifiable is greater than that of polynomial-time decidable, but no proof has been given to support this statement (see Figure 1.4). The question of whether or not \( \mathcal{P} = \mathcal{NP} \) is one of the greatest unsolved problems in computer science and mathematics, and in fact it is one of the seven Millennium Prize Problems proposed by the Clay Mathematics Institute in Boston in 2000, each with one-million US dollars.

**Definition 1.5** \( \mathcal{EXP} \) is the class of problems solvable by a deterministic Turing machine (DTM) in time bounded by \( 2^{\text{poly}(n)} \).

**Definition 1.6** A function \( f \) is polynomial-time computable if for any input \( w \), \( f(w) \) will halt on a Turing machine in polynomial-time. A language \( A \) is polynomial-time reducible to a language \( B \), denoted by \( A \leq_p B \), if there exists a polynomial-time computable function such that for every input \( w \),

\[
    w \in A \iff f(w) \in B.
\]

The function \( f \) is called the polynomial-time reduction of \( A \) to \( B \).
Definition 1.7  A language/problem $L$ is $\mathcal{NP}$-complete, denoted by $\mathcal{NP}_c$, if it satisfies the following two conditions:

1. $L \in \mathcal{NP}$,
2. $\forall A \in \mathcal{NP}, A \leq_p L$.

Definition 1.8  A problem $D$ is $\mathcal{NP}$-hard, denoted by $\mathcal{NP}_h$, if it satisfies the following condition:

$$\forall A \in \mathcal{NP}, A \leq_p D$$

where $D$ may be in $\mathcal{NP}$, or may not be in $\mathcal{NP}$. Thus, $\mathcal{NP}$-hard means at least as hard as any $\mathcal{NP}$-problem, although it might, in fact, be harder.

Definition 1.9  $\mathcal{RP}$ is the class of problems solvable in expected polynomial-time with one-sided error by a probabilistic (randomized) Turing machine (PTM). By “one-sided error” we mean that the machine will answer “yes” when the answer is “yes” with a probability of error $< 1/2$, and will answer “no” when the answer is “no” with zero probability of error.

Definition 1.10  $\mathcal{ZPP}$ is the class of problems solvable in expected polynomial-time with zero error on a probabilistic Turing machine (PTM). It is defined by $\mathcal{ZPP} = \mathcal{RP} \cap \text{co-}\mathcal{RP}$, where co-$\mathcal{RP}$ is the complement of $\mathcal{RP}$. By “zero error” we mean that the machine will answer “yes” when the answer is “yes” (with zero probability of error), and will answer “no” when the answer is “no” (also with zero probability of error). But note that the machine may also answer “?”, which means that the machine does not know if the answer is “yes” or “no.” However, it is guaranteed that in at most half of simulation cases the machine will answer “?.” $\mathcal{ZPP}$ is usually referred to as an elite class, because it also equals to the class of problems that can be solved by randomized algorithms that always give the correct answer and run in expected polynomial-time.

Definition 1.11  $\mathcal{BPP}$ is the class of problems solvable in expected polynomial-time with two-sided error on a probabilistic Turing machine (PTM), in which the answer always has probability at least $\frac{1}{2} + \delta$, for some fixed $\delta > 0$ of being correct. The “B” in $\mathcal{BPP}$ stands for “bounded away the error probability from $\frac{1}{2}$”; for example, the error probability could be $\frac{1}{3}$.

It is widely believed, although no proof has been given, that problems in $\mathcal{P}$ are computationally tractable, whereas problems not in (beyond) $\mathcal{P}$ are computationally intractable. This is the famous Cook–Karp thesis, named after Stephen Cook and Richard Karp:

The Cook–Karp thesis. Any computationally tractable problem can be computed by a Turing machine in deterministic polynomial-time.
Thus, problems in $\mathcal{P}$ are tractable whereas problems in $NP$ are intractable. However, there is not a clear cut line between the two types of problems. This is exactly the $P$ versus $NP$ problem, mentioned earlier.

Similarly, one can define the classes of problems of $P$-Space, $NP$-Space, $P$-Space Complete, and $P$-Space Hard. We shall use $NP$ to denote the set of $NP$-Complete problems, $PSC$ the set of $P$-Space Complete problems, $NPH$ the set of $NP$-Hard problems, and $PSH$ the set of $P$-Space Hard problems. The relationships among the classes $P, NP, NPC, PSC, NPH, PSH$, and $EXP$ may be described as in Figure 1.5. 

It is clear that a time class is included in the corresponding space class since one unit is needed for the space by one square. Although it is not known whether or not $P = NP$, it is known that $PSPACE = NPSPACE$. It is generally believed that

$$P \subseteq ZPP \subseteq RP \subseteq \left( \frac{BPP}{NP} \right) \subseteq PSPACE \subseteq EXP.$$  

Besides the proper inclusion $P \subset EXP$, it is not known whether any of the other inclusions in the above hierarchy is proper. Note that the relationship of $BPP$ and $NP$ is not known, although it is believed that $NP \not\subseteq BPP$. 

**Figure 1.5** Conjectured relationships among classes $P$, $NP$ and $NPC$, etc.