HZ BOOKS
华章教育

计 算 机 科 学 丛 书

Apress

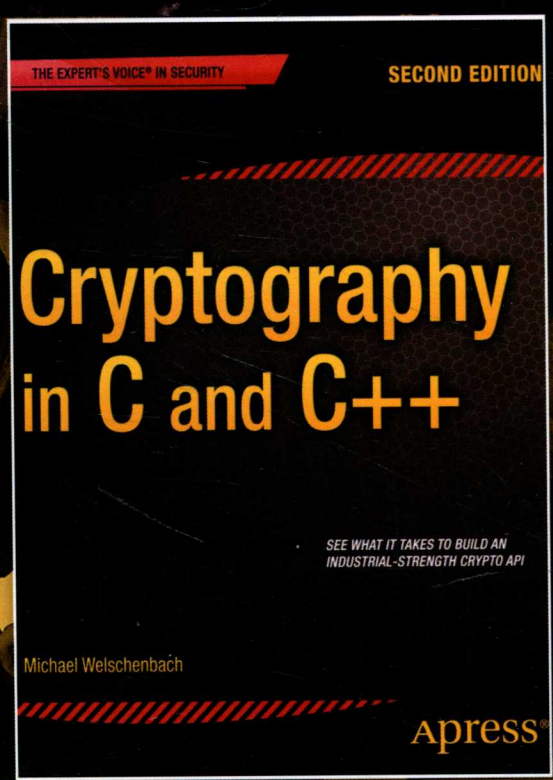
原书第2版

密码学

C/C++语言实现

[德] 迈克尔·威尔森巴赫（Michael Welschenbach） 著
杜瑞颖 何琨 周顺淦 译

Cryptography in C and C++
Second Edition



机械工业出版社
China Machine Press

计 算 机 科 学 丛 书

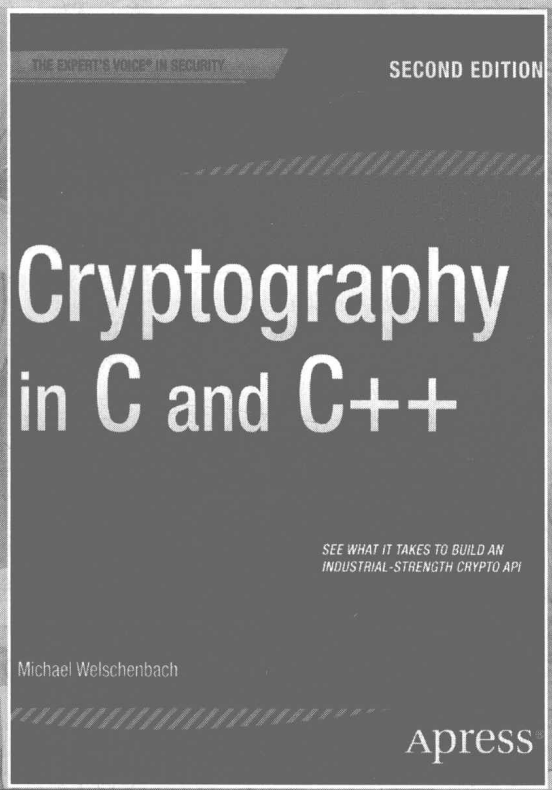
原书第2版

密码学

C/C++语言实现

[德] 迈克尔·威尔森巴赫（Michael Welschenbach） 著
杜瑞颖 何琨 周顺淦 译

Cryptography in C and C++
Second Edition



机械工业出版社
China Machine Press

图书在版编目（CIP）数据

密码学：C/C++ 语言实现（原书第 2 版）/（德）威尔森巴赫（Welschenbach, M.）著；杜瑞颖，何琨，周顺淦译 . —北京：机械工业出版社，2015.10
（计算机科学丛书）
书名原文：Cryptography in C and C++, Second Edition

ISBN 978-7-111-51733-7

I. 密… II. ①威… ②杜… ③何… ④周… III. 密码算法 – C 语言 – 程序设计 IV. ① TN918.1 ② TP312

中国版本图书馆 CIP 数据核字（2015）第 233102 号

本书版权登记号：图字：01-2013-9167

本书主要阐述如何使用 C 和 C++ 语言实现密码学算法，包括编写专家级的密码所需要掌握的知识和技术，以及如何安全并高效地实现密码学算法。第 2 版包括了许多全新内容，同时对原有内容进行了修改和完善，使之涵盖密码学领域的最新技术进展。作为一本密码学的书籍，本书叙述了一个重要的对称加密算法 AES 的理论及实现，还完整地实现了一个重要的非对称密码系统——RSA 加密和 RSA 签名。作为一本算法实现的书籍，本书严格遵循软件开发原则，详细描述了设计思想及错误处理方法，并对所有函数进行了广泛测试。

本书可以作为高等院校信息技术相关专业高年级本科生或研究生的教材，也是信息技术从业人员极佳的参考书。

出版发行：机械工业出版社（北京市西城区百万庄大街 22 号　邮政编码：100037）
责任编辑：盛思源　　责任校对：董纪丽
印　　刷：中国电影出版社印刷厂　　版　　次：2015 年 10 月第 1 版第 1 次印刷
开　　本：185mm × 260mm　1/16　　印　　张：19.5
书　　号：ISBN 978-7-111-51733-7　　定　　价：69.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88378991　88361066　　投稿热线：（010）88379604
购书热线：（010）68326294　88379649　68995259　　读者信箱：hzjsj@hzbook.com

文艺复兴以来，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的优势，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭示了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短的现状下，美国等发达国家在其计算机科学发展的几十年间积淀和发展的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起到积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章公司较早意识到“出版要为教育服务”。自 1998 年开始，我们就将工作重点放在了遴选、移译国外优秀教材上。经过多年的不懈努力，我们与 Pearson，McGraw-Hill，Elsevier，MIT，John Wiley & Sons，Cengage 等世界著名出版公司建立了良好的合作关系，从他们现有的数百种教材中甄选出 Andrew S. Tanenbaum，Bjarne Stroustrup，Brain W. Kernighan，Dennis Ritchie，Jim Gray，Afred V. Aho，John E. Hopcroft，Jeffrey D. Ullman，Abraham Silberschatz，William Stallings，Donald E. Knuth，John L. Hennessy，Larry L. Peterson 等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及珍藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力相助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专门为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近两百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍。其影印版“经典原版书库”作为姊妹篇也被越来越多实施双语教学的学校所采用。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证。随着计算机科学与技术专业学科建设的不断完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都将步入一个新的阶段，我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方法如下：

华章网站：www. hzbook. com
电子邮件：hzjsj@hzbook. com
联系电话：(010)88379604
联系地址：北京市西城区百万庄南街 1 号
邮政编码：100037

华章科技图书出版中心

译者序

Cryptography in C and C++, Second Edition

密码学是一门有着悠久历史的科学，在人类历史中扮演着重要的角色。然而，密码学又是一门神秘的学问，许多人知道它，却不了解它；许多人研究它，却不能公开讨论它。早期的密码学研究作为各国的军事机密而讳莫如深，直到20世纪中叶，这一状况才开始改变。随着众多杰出科学家的介入，密码学由一门依靠设计者经验的编码艺术，转变为一门严谨的科学。尤其是公钥密码学的诞生，使得密码学具备了保密通信之外的更为广泛的功能，也使得密码学成为一门蓬勃发展的新兴学科。

密码技术从诞生之初就以保护军事机密和商业机密为目标。我们的日常生活也离不开密码技术，如无线网络接入认证、安全电子邮件系统等。因此，我们不仅要分析密码学的理论基础，更要研究密码学的具体实现。本书就是这样一本专注于密码学实现的书籍，这种独特的视角使其在众多密码学理论书籍中脱颖而出。

本书讨论了如何使用C语言和C++语言实现密码学算法。这一过程远比想象的复杂。公钥密码学以大整数计算为基础，而这些大整数远远超过了C语言和C++语言对整数的处理范围，因此实现密码学算法必须先实现大整数计算。在大整数计算的基础上，本书介绍了大量密码学算法的实现，并确保它们既高效又安全。

作为一本密码学的书籍，本书叙述了一个重要的对称加密算法AES的理论及实现，也完整地实现了一个重要的非对称密码系统——RSA加密和RSA签名。作为一本算法实现的书籍，本书严格遵循软件开发的原则，详细描述了设计思想及错误处理方法，并对所有函数进行了广泛的测试。

对于以实现真正实用的密码学算法为目的，并了解相关理论基础的读者而言，本书是一本极佳的读本。

由于译者的外语水平及专业知识有限，所以在翻译中难免有错误或不妥之处，请读者理解并指正。

译者

2015年8月

密码学是一门有着两千多年历史的古老艺术。只要信息保密的需求一直存在，那么保护秘密的尝试就会一直进行。然而直到最近三十年，密码学才发展成为一门科学，并且为我们提供日常生活中所需的安全保障。无论是自动柜员机、蜂窝电话、因特网商务，还是汽车上的计算机点火锁，密码学都暗藏其中。更重要的是，没有密码学这些应用都将无法工作！

最近三十年的密码学历史是一段非同寻常的成功故事。最重要的事件无疑是 20 世纪 70 年代中期公钥密码学的发现。这是一场真正的革命：现在我们知道那些以前不敢想象的事物是可能的。Diffie 和 Hellman 第一个公开表达了安全通信必须能自发地发生的愿景。在此之前，发送者和接收者必须首先通过保密通信建立一个共同的密钥。Diffie 和 Hellman 提出了一个天真的问题：人们是否可以在不共享一个共同密钥的情况下进行保密通信？他们的想法是不使用其他人不知道的私钥加密信息。这个想法标志着公钥密码学的诞生。随着几年后 RSA 算法的出现，这一愿景不再只是一个大胆的猜测。

数学与计算机科学富有成效的合作使现代密码学成为可能。数学为算法的创造和分析提供了基础。没有数学，尤其是没有数论，公钥密码学就无法实现。数学提供了用于算法运算的成果。

若要实现密码学算法，则需要能够支持大整数计算的程序：这些算法不能仅在理论上发挥作用，还必须按现实世界的规范执行。这是计算机科学的任务所在。

本书区别于所有其他相关书籍的地方在于，本书阐明了数学和计算之间的关系。我没有看到任何其他的密码学书籍能在充分地呈现数学基础的同时，还能提供大量的实际应用，并且使所有内容都清晰易读。

本书是作者对其专业知识的精彩呈现。他了解理论，并且能清晰地表达它们；他了解应用，并且能展现许多程序来实现它们；他了解许多东西，但没有表现出无所不知的样子；他清晰地提出他的论据，以便读者能获得一个清晰的理解。简而言之，这是一本出色的书。

祝福作者！并祝福你，本书的读者！

Albrecht Beutelspacher

英文第2版前言

Cryptography in C and C++, Second Edition

当必须与数字打交道时，我宁愿把自己塞进地洞中，这样就看不见任何东西。如果我抬起头，看见大海、一棵树或者一个女人(即使只是一个老妇人)，如果将所有的结果和数字都化作一阵轻烟。它们长出翅膀飞走了，我只能去追赶。

——*Nikos Kazanzakis*，《*Zorba the Greek*》

本书的英文第2版又经历了一次修订与扩充。我们完全重写了随机数生成器这一章，并且大幅修订了素性检验这一节。Agrawal、Kayal 和 Saxena 在素性检验方面的最新成果——曾经在2002年引起轰动的《PRIMES is in P》，也涵盖在内。我们重新安排了 Rijndael/AES 这一章的位置，以便达到更好的效果。同时也指出 Rijndael 的标准化作为高级加密标准(Advanced Encryption Standard)被美国国家标准与技术研究院(National Institute of Standards and Technology，NIST)列为官方标准。

与本书之前的版本不同，英文第2版不再包含一张有程序源代码的光盘。相应地，源代码可以从 www. apress. com 的 Downloads 处下载。

感谢出版社和译者，他们使这本书有了中文、韩文、波兰文和俄文的版本，同时他们的仔细阅读也为该版本的质量做出了贡献。

再一次感谢 David Kramer 付出智慧与汗水将本书翻译为英文，还要感谢 Apress 的 Gary Cornell 愿意出版这本英文第2版。

最后，感谢 Springer Science 出版社，尤其是 Hermann Engesser、Dorothea Glausinger 和 Ulrike Sricker，感谢他们的愉快合作。

数学是一门被误解甚至被中伤的学科。它不是在小学里我们被灌输的蛮力计算，也不是关于清算的科学。数学家不会把他们的时间花在想出更聪明的乘法方法、更快的加法方法以及更好的开立方方法上。

——*Paul Hoffman*，《*The Man Who Loved Only Numbers*》

英文第1版翻译自德文第2版。德文第2版在许多方面对德文第1版进行了修订和扩充，增加了一些密码学算法实例，如Rabin和El Gamal函数，并在RSA函数的实现中采用了RIPEMD-160散列函数以及PKCS#1格式。同时，还讨论了导致程序缺陷的可能错误来源。许多文字部分都进行了扩充、澄清以及错误更正。另外，强化了讲授策略，因此有些程序的源代码与书中的描述存在某些细节上的区别。并不是所有技术细节都同样重要，快速有效的代码也不总是清晰易读、引人注意。

谈到效率，在附录D中将程序的运行时间与GNU多精度库中的特定函数进行了比较。在比较中，FLINT/C指数运算表现不俗。作为补充，附录F提供了一些算术和数论包以供参考。

软件扩充了一些函数，并在一些地方进行了大量完善工作，移除了一些错误和不精确的地方。软件开发了额外的测试函数，并扩充了现有的测试函数。软件还实现了一种安全模式，通过重写，删除了函数中与安全性密切相关的变量。附录中明确地引用了所有的C和C++函数并加以说明。

由于当前编译器支持的C++标准并不统一，所以FLINT/C包的C++模块被设计为在传统的形式为xxxxx.h的C++头文件和新的ANSI头文件中都可以使用。出于同样的原因，在使用new()运算符时将检查是否返回一个null指针。这类错误处理没有使用ANSI标准异常，但能在当前的编译器下工作。而遵从标准的方法，即new()通过throw()生成一个错误，并不总是可行的。

虽然本书专注于非对称密码学的基本原理，但是由于Rijndael最近被美国国家标准与技术研究院(NIST)提名为高级加密标准(AES)，所以将这一算法的描述放到最后一章(第11章)。感谢Apress的Gary Cornell提出这个主题，并使我相信它值得成为本书的一部分。感谢Vincent Rijmen、Antoon Bosselaers、Paulo Barreto和Brian Gladman允许我们在本书的源代码中包含他们的Rijndael实现。

感谢所有第1版的读者，尤其是那些指出错误、给出评论或提出改进意见的人。我们非常愿意与他们交流。和往常一样，作者承担所有仍然留在本书或软件中的错误，以及任何新增错误的责任。

由衷地感谢Apress的Gary Cornell以及Springer-Verlag的Hermann Engesser、Dorothea Glaunsinger和Ulrike Stricker，感谢他们的无私奉献和友好合作。

感谢我的译者David Kramer，他以卓越的专业知识以及孜孜不倦的奉献精神提出了大量宝贵的意见，这些内容也融入了本书的德文版中。

警告

在使用包含在本书中的程序前，请参考相关软件和计算机的产品指南与技术说明。作者和出版社都不承担任何由于不正确地使用本书程序所带来的损失。可下载的源代码中的程序受到版权保护，未经出版社允许不能复制。

德文第1版前言

Cryptography in C and C++, Second Edition

数学是科学的皇后。数论是数学的皇后。通常，她屈尊帮助天文学与其他自然科学，但在任何情况下，她都是最重要的。

——*Carl Friedrich Gauss*

本书专注于整数算术及其在计算机程序中的应用，不过，为什么我们需要这样一本密码学的书？这与计算机科学一般所牵涉的重要问题比起来是否是一个微不足道的主题呢？只要我们把自己封闭在那些可以用编程语言标准数字类型表示的数的范围内，算术就是一件非常简单的事，那些熟悉的算术运算伴随着熟悉的符号＋、－、/、＊在程序中自然地出现。

但是，当我们需要长度远远大于16位或32位的结果时，情况就变得有趣了。即使是基本的算术运算也无法在这些数上实现，我们需要投入大量精力解决那些以前从来不是问题的问题。任何研究数论尤其是现代密码学课题的人，无论是专业人士还是业余爱好者，都熟知这样的情况：我们在学校里学到的算术技术都要重新思考，并且我们会发现自己有时候要解决难以置信的复杂过程。

那些想要在这些领域开发程序但不想从头开始的读者将发现，本书包含的一系列用于大整数计算的函数可以作为C和C++的扩展。我们并不讨论那些解释“它的工作原理是什么”的“小儿科”的例子。我们提供一套完整的函数和方法，它们满足行业的稳定与性能需求，并有着坚实的理论基础。

在理论和实践之间建立连接是本书的目标，即填补理论文献与实际编程问题之间的缝隙。在前面的几章中，我们逐步研究大的自然数的基本计算原理、有限环和域中的算术，以及一些更复杂的初等数论函数，并阐述将这些原理应用于现代密码学的各种可能性。我们对数学基本原理的解释足以帮助读者理解本书给出的程序，对于那些想要深入了解的人，我们提供了大量的参考文献。我们将开发的函数组织到一起并进行大量测试，最终形成一个有用且全面的编程接口。

我们从大数的表示开始，并在随后几章中探讨计算的基础。对于大数的加法、减法、乘法和除法，我们编写了强大的基本函数。基于这些函数，我们解释剩余类中的模算术，并在库函数中实现了相应的运算。我们划分了单独的一章专注于耗时的幂运算过程，该章设计并实现了一些针对模算术中应用的特定算法。

在经过大数输入与输出、不同基数之间的转换等准备后，我们使用这些基本的算术函数研究初等数论算法，然后从计算大数的最大公约数开始开发程序。接着我们转向研究计算Legendre和Jacobi符号、在有限环上求逆和计算平方根等问题，并熟悉中国剩余定理及其应用。

接下来，我们讨论识别大素数的原理的细节，并编写了强大的多阶段素性检验程序。

随后的一章致力于大随机数的生成，开发了密码学中使用的位生成器，并测试了其统计特征。

在第一部分的最后，我们测试了算术以及其他功能。我们从算术的数学规则中导出特殊的测试方法，并考虑了高效的外部工具的实现。

第二部分的主题是逐步构建 C++ 类 LINT(Large INTegers)。在此过程中，我们将第一部分的 C 函数嵌入面向对象编程语言 C++ 的语法与语义中。我们特别关注使用灵活的流函数和操控器对 LINT 对象进行格式化输入和输出，以及基于异常的错误处理。用 C++ 表示的算法的优雅令人印象深刻，尤其是标准类型和 LINT 对象的边界变得模糊，使得在实现算法时语法较为接近，并且更清晰和透明。

最后，我们通过实现用于加密和数字签名的扩展 RSA 密码系统展示如何应用我们开发的方法。在这个过程中，我们解释最具代表性的非对称密码系统 RSA 的理论及其操作。根据 C++ 编程语言的面向对象原则，我们在一个自包含的例子里开发了一个可扩展的内核。

我们以对软件库进一步可能的扩展的讨论作为结束。作为最后的一个要点，我们给出 4 个用于乘法和除法的 80x86 汇编语言的函数，它们能改进软件的效率。附录 D 包含了使用和不使用汇编器情况下的典型计算时间的表格。

衷心地欢迎本书的所有读者加入我们，或者根据个人兴趣专注于某些章节，并试用给出的函数。作者用“我们”指代自己及读者，希望这一点不会引起误解。他希望借此鼓励他们在数学和计算机科学的前沿领域发挥积极的作用，并从本书中受益。至于软件，我们鼓励读者通过新的实现优化一个或多个函数的范围或速度。

感谢 Springer-Verlag，特别是 Hermann Engesser、Dorothea Glaunsinger 和 Ulrike Stricker，他们对本书的出版抱有兴趣，并开展了友好积极的合作。本书手稿由 Jörn Garbers、Josef von Helden、Brigitte Nebelung、Johannes Ueberberg 和 Helga Welschenbach 审阅。衷心地感谢他们至关重要的建议与改进，以及他们的细心与耐心。尽管我们付出了努力，但本书或软件中仍可能存在错误，作者将独自承担责任。非常感谢我的朋友及同事 Robert Hammelrath、Franz-Peter Heider、Detlef Kraus 和 Brigitte Nebelung，在多年的合作中，他们对数学与计算机科学之间关联的洞察对我影响深远。

目 录

Cryptography in C and C++, Second Edition

第一部分

Cryptography in C and C++, Second Edition

算术与数论：C实现

算术和整个数学艺术的重要性是显而易见的，几乎所有的创造都离不开精确的数字和度量。同样，如果没有度量和比例，也没有独立存在的艺术。

——*Adam Ries*，《*Book of Calculation*》，*1574*

操纵数字的排印规则，事实上，也就是数字的运算规则。

——*D. R. Hofstadter*，

《*Godel, Escher, Bach: An Eternal Golden Braid*》

人类的大脑将不再因为需要计算而感受到负担！天才的人们拥有思考的能力而非只是书写数字。

——*StenNadolny*，《*The Discovery of Slowness*》

（*Ralph Freedman* 译）

第1章

Cryptography in C and C++, Second Edition

绪　论

上帝创造了整数，剩下的就是人类的工作了。

——Leopold Kronecker(1823—1891)

看着“零”时，你什么都看不到，但是透过它你可以看到这个世界。

——Robert Kaplan，《The Nothing That Is: A Natural History of Zero》

不管是否愿意，理解现代密码学必将涉足数论，即自然数的研究。数论是整个数学学科中最引人入胜的领域之一。但是，我们不必为了密码学的应用而深入浩瀚的数学海洋、挖掘晦涩的数学宝藏，我们的目标比较适中。当然，对密码学涉及的数论进行任何程度的深入研究都不为过，在密码学这个领域确实有很多著名的数学家做出了非常重要的贡献。

人们对数论的研究历史源远流长。古希腊数学家和哲学家 Pythagoreans 及他的学派早在公元前六世纪就已经对整数之间的关系进行了较为深入的探索并获得了一些重要的结论，例如著名的 Pythagoreans 定理[⊖]，该定理几乎在每一所中学的课本上都会出现。当时由于对宗教的信仰，他们认为所有的数都应该和自然数相对应。不久，他们就发现自己处在一个自相矛盾的境地，因为他们发现了像$\sqrt{2}$这样不能表示成两个整数的商的无理数。这个发现使得 Pythagoreans 学派的世界观发生了混乱，以至于他们抵制无理数的相关知识，当然，这只是人类历史上经常重演的一次次徒劳无益的行为之一。

我们今天用来保障因特网中通信安全最常用的加密算法就与两个最早的数论算法紧密相关，这两个算法分别出自古希腊数学家 Euclid(公元前三世纪)和 Eratosthenes(公元前 267 年～公元前 195 年)之手。“Euclid 算法”和“Eratosthenes 筛法”这两个算法与我们现在的工作紧密相连，我们将在 10.1 节和 10.5 节中分别介绍其理论和应用。

Pierre de Fermat(1601—1665)、Leonhard Euler(1707—1783)、Adrien Marie Legendre(1752—1833)、Carl Friedrich Gauss(1777—1855)和 Ernst Eduard Kummer(1810—1893)等人被认为是现代数论最重要的奠基者。他们的工作形成了这个领域发展的基础，尤其是对密码学这样有趣的应用领域的基础性贡献，例如非对称加密的产生和数字签名的生成(参见第 17 章)。如果不是限于篇幅，我们还想再提一些在这个领域做出重要贡献的数学家，他们一直为数论的发展发挥着极其重要的作用。我非常推崇 Simon Singh 所著的《Fermat's Last Theorem》一书。

考虑到我们在孩提时代已经学会了计数并将一些事实认为理所当然，比如 2 加 2 等于 4，我们必须构建一些抽象的概念进行一些理论判断的假设。例如，集合论帮助我们从“(几乎)没有”出发去理解自然数的存在和运算。“几乎没有”就是空集$\varnothing := \{\}$，即这个集合中没有任何元素。当我们把空集对应到自然数 0 时，我们就可以通过如下方法构建加法的集合。0 的后继者0^+可以对应为集合$0^+ := \{0\} = \{\varnothing\}$，这个集合包含唯一的元素，这个元素即为空集。我们将 0 的后继者命名为 1，于是我们可以给出 1 的后继者$1^+ :=$

⊖ 即勾股定理。——译者注

$\{\varnothing, \{\varnothing\}\}$，它包含0和1两个元素，我们将其定义为2。于是这些集合就定义了我们熟知的自然数0、1、2。

上述集合的构建方式，即为每一个 x 给出其后继者 $x^+ := x \cup \{x\}$，这种方式可以用来产生更多的数。于是，除了0以外，每个数都可以用该方法产生，即它本身是一个包含前继者的集合，只有0没有前继者。为了保证这个产生过程无限地继续下去，集合论规定了一个称为*无穷性公理*的法则：即存在一个集合，它包含0和其中每一个元素的后继者。

根据假设的*后继集合*(以0开始且包含所有后继者的集合)存在性，集合论给出了一个最小的后继集合$\mathbb{N}$的存在性，该集合是所有后继集合的子集。这个最小且唯一的后继集合$\mathbb{N}$称为*自然数集*，这里我们将0也作为元素包含其中[⊖]。

自然数可以用Giuseppe Peano(1858—1932)提出的公理来描述，这种方式也更符合我们直观上对自然数的理解：

1) 两个不相等自然数的后继者也不相等：对于所有的 n，$m \in \mathbb{N}$，若 $n \neq m$，则 $n^+ \neq m^+$。

2) 除0以外，所有自然数都有后继者：$\mathbb{N}^+ = \mathbb{N} \setminus \{0\}$。

3) *完全归纳法*：如果 $S \in \mathbb{N}$，$0 \in S$，且任给 $n \in S$，总有 $n^+ \in S$，则 $S = \mathbb{N}$。

完全归纳法给出了我们感兴趣的自然数的运算。作为基础运算的加法和减法可以如下定义递归。首先定义**加法**：

任意给定自然数 $n \in \mathbb{N}$，存在一个从$\mathbb{N}$到$\mathbb{N}$映射的函数 s_n，满足：

1) $s_n(0) = n$；

2) 对于任意给定的自然数 $x \in \mathbb{N}$，有 $s_n(x^+) = (s_n(x))^+$。

函数 $s_n(x)$的值就称为 n 和 x 的和 $n + x$。

然而，对于所有自然数 $n \in \mathbb{N}$，需要证明是否存在这样的函数 s_n，因为自然数的无穷性并不是一个先验的假设。根据上文提到的Peano第三公理(参见[Halm]的第11～13章)，将该加法的存在性证明规约到完全归纳法的基本规则。同时，可以用类似的方法来定义**乘法**：

任意给定自然数 $n \in \mathbb{N}$，存在一个从$\mathbb{N}$到$\mathbb{N}$映射的函数 p_n，满足：

1) $p_n(0) = 0$；

2) 对于任意给定的自然数 $x \in \mathbb{N}$，有 $p_n(x^+) = p_n(x) + n$。

函数 $p_n(x)$的值就称为 n 和 x 的积 $n \cdot x$。

不出所料，自然数的乘法就是用加法来定义的。通过如上对自然数加法和乘法的定义，该定义可以通过重复对 x 进行完全归纳来证明(根据Peano第三公理)，我们可以得出结合律、交换律和分配率(参见[Halm]的第13章)等熟知的运算律。尽管我们经常不假思索地频繁使用这些运算律，但是当检验我们编写的FLINT函数库(参见第13章和第18章)时，我们还是要尽可能地对其运用自如。

用类似的方法我们可以获得**幂运算**的定义，鉴于该运算在后续章节中的重要性，我们在这里给出其定义形式。

任意给定自然数 $n \in \mathbb{N}$，存在一个从$\mathbb{N}$到$\mathbb{N}$映射的函数 e_n，满足：

1) $e_n(0) = 1$；

2) 对于任意给定的自然数 $x \in \mathbb{N}$，有 $e_n(x^+) = e_n(x) \cdot n$。

⊖ 根据标准DIN 5473，0属于自然数，这本身并非一个毫无争议的选择。但是，从计算机科学的角度，以0作为计数的开始比1更符合实际，这也标志0作为加法运算(加法一致性)(即任意自然数与0相加等于其本身。——译者注)中“中立元素”的重要性。

函数 $e_n(x)$ 的值就称为 n 的 x 次幂 n^x。运用完全归纳法，我们可以证明**幂律**（参见第 6 章）：

$$n^x n^y = n^{x+y},\quad n^x \cdot m^x = (n \cdot m)^x,\quad (n^x)^y = n^{xy}$$

除了计算操作，在自然数集合 $\mathbb{N}$ 上还定义了“$<$”的顺序关系以便允许对任意给定的两个元素（n，$m \in \mathbb{N}$）进行比较。尽管这个事实值得我们从集合论的观点进行高度关注，但是实际上我们已能非常清晰地理解这种关系并在我们的日常生活中经常使用。

既然我们从建立一个空集作为唯一的基础构件开始定义自然数，那么接下来我们将考虑一些实质性的问题。尽管数论主要考虑自然数、整数以及它们的性质，而不过多地关注其他内容，但是，我们至少应鸟瞰数学分支的“细胞分裂”，这种“分裂”不仅产生了数论，还包括我们后面所涉及的运算及其规则。

关于软件

本书中描述的软件包含在一个完整的软件包中，这个软件包是我们经常引用的函数库。我们将这个库命名为 FLINT/C，它是“数论和密码学中的大整数函数”（functions for large integers in number theory and cryptography）的首字母缩写。

FLINT/C 库包含的模块如表 1-1～表 1-5 所示，这些模块的源代码在 www.apress.com 中可以找到。

表 1-1　目录 flint/src 中用 C 实现的算术与数论

flint.h	使用 flint.c 的头文件
flint.c	用 C 描述的算术和数论函数
kmul.{h,c}	Karatsuba 乘法和开方函数
ripemd.{h,c}	散列函数 RIPEMD-60 的实现
sha{1,256}.{h,c}	散列函数 SHA-1、SHA-256 的实现
entropy.c	生成作为伪随机序列初始值的熵
random.{h,c}	生成伪随机数
aes.{h,c}	高级加密算法（AES）的实现

表 1-2　目录 flint/src/asm 中用 80x86 汇编器（参见第 19 章）描述的算术模块

mult.{s,asm}	乘法，用来代替 flint.c 中的 C 函数 mult()
umul.{s,asm}	乘法，用来代替 C 函数 umul()
sqr.{s,asm}	平方，用来代替 C 函数 sqr()
div.{s,asm}	除法，用来代替 C 函数 div_l()

表 1-3　目录 flint/test 和目录 flint/test/testvals 中的测试（参见 13.2 节和第 18 章）

testxxx.c[pp]	用 C 和 C++ 描述的测试程序
xxx.txt	AES 的测试向量

表 1-4　目录 flint/lib 和目录 flint/lib/dll 中用 80x86 汇编器描述的函数库

flinta.lib	用目标模块格式（OMF）描述的汇编函数库
flintavc.lib	用通用对象文件格式（COFF）描述的汇编函数库
flinta.a	OS/2 下 emx/gcc 的汇编函数静态库
libflint.a	Linuc 下汇编函数静态库
flint.dll	MS VC/C++ 下 FLINT/C 的动态链接库（DLL）
flint.lib	flint.dll 的链接库

表 1-5 flint/rsa 目录中 RSA 的实现(参见第 17 章)

rsakey.h	RSA 类的头文件
rsakey.cpp	RSA 类 RSAkey 和 RSApub 的实现
rsademo.cpp	RSA 类及其函数的应用例子

FLINT/C 软件的组件列表可以在源代码中的 readme.doc 中找到。该软件已经在如下的平台中用开发工具进行了测试:

- Linux、SunOS 4.1 和 Sun Solaris 下的 GNU gcc
- OS/2 Warp、DOS 和 Windows(9x、NT)下的 GUN/EMX gcc
- Windows(9x、NT、2000、XP)下的 Borland BCC32
- Windows(9x、NT、2000、XP)下的 lcc-win32
- Windows(9x、NT、2000、XP)下的 Cygnus cygwin B20
- OS/2 Warp 和 Windows(9x、NT、2000、XP)下的 IBM VisualAge
- DOS、OS/2 Warp 和 Windows(9x、NT)下的 Microsoft C
- Windows(9x、NT、2000、XP)下的 Microsoft Visual C/C++
- DOS、OS/2 Warp 和 Windows(3.1、9x、NT、XP)下的 Watcom C/C++
- Windows(2000、XP)下的 OpenWatcom C/C++

汇编程序通过 Microsoft MASM[㊀]、Watcom WASM 或 GNU 汇编器 GSA 可以进行转换。它们已转化成对象模型格式(OMF)和通用对象文件格式(COFF)库的格式,以及 LINUX 静态库的格式,并包含在可下载的源代码中。在已定义的 FLINT_ASM 宏和链接函数库的情况下,上述格式的代码可以代替 C 函数。

以 GNU 的编译器 gcc 为例,一次典型的编译器调用类似于如下的过程(到源程序目录的路径未知时):

```
gcc -O2 -o rsademo rsademo.cpp rsakey.cpp flintpp.cpp
   randompp.cpp flint.c aes.c ripemd.c sha256.c entropy.c
   random.c -lstdc++
```

在编译中定义宏 FLINTPP_ANSI 时,需要使用遵循美国国家标准协会(ANSI)标准编写的 C++ 头文件;否则,使用传统的头文件 xxxxx.h。

不同的计算机平台编译程序开关可能会稍有不同,但是通常为了达到最佳性能都需要开启速度优化程序。同时,由于栈的使用,不同的环境和应用也可能需要做相应的调整[㊁]。考虑到特定应用对栈大小的要求,读者可以参考第 6 章中对幂运算函数的一些建议。当然,如果使用动态栈分配或用动态寄存器(参见第 9 章)实现幂运算函数时,对栈的要求可以适当放宽。

```
extern int __FLINT_API add_l(CLINT, CLINT, CLINT);
extern USHORT __FLINT_API_DATA smallprimes[];
```

以及汇编函数

```
extern int __FLINT_API_A div_l (CLINT, CLINT, CLINT, CLINT);
```

中用宏定义的 C 函数和常量:

__FLINT_API　　C 函数的修饰符

㊀ 调用:ml /Cx /c /Gd< filename> 。

㊁ 除了 DOS 系统外,现代计算机都设置了虚拟内存,所以读者不必担心这一点,尤其是使用 UNIX 或 Linux 系统的用户。