ET PUBLISHING
今日電子

Internet和数据库加密

Cryptography for Internet and Database Applications

[美] Nick Galbreath 著
曾振宇 白克壮 尹 喆 等译

WILEY
電子工業出版社
Publishing House of Electronics Industry
www.phei.com.cn

Internet 和数据库加密

Cryptography for Internet and Database Applications

[美] Nick Galbreath 著

曾振宇 白克壮 尹 喆 等译

電子工業出版社

Publishing House of Electronics Industry

北京 · BEIJING

内 容 简 介

本书介绍了如何用Java的加密技术提高数据的安全性。重点介绍了存储、消息完整性和身份验证密码学。全书包括7章：第1～4章介绍了密码学的基础知识，第5章介绍了Java密码API，第6章介绍了短消息的编码和加密，第7章是前面6章内容的实际应用。

本书的特点是用简单实用的示例程序来阐述理论问题和实现方法，并为数据库和Web应用程序的安全提供了详细的Java语言实现指南和代码示例。全书内容翔实、清晰、概念性强、论述深入浅出，适合于从事密码系统研究的软件工程师阅读，也可作为大专院校相关专业的教学参考书。

Copyright ©2002 by Publishing House of Electronics Industry. Original English language edition copyright ©2002 by Wiley Publishing, Inc. All rights reserved including the right of reproduction in whole or in part in any form. This translation published by arrangement with Wiley Publishing, Inc.
本书中文简体专有翻译出版权由美国Wiley Publishing，Inc.授予电子工业出版社及其所属今日电子杂志社。未经许可，不得以任何手段和形式复制或抄袭本书内容。该专有出版权受法律保护，侵权必究。

著作权合同登记号　图字：01-2002-5162

图书在版编目（CIP）数据

Internet和数据库加密／（美）加尔布雷恩（Galbreath,N.）著；曾振宇等译.
—北京：电子工业出版社，2003.6
书名原文：Crytography for Internet and Database Applications
ISBN 7-5053-8760-X

Ⅰ.I...　Ⅱ.①加...②曾...　Ⅲ.①因特网－基本知识②因特网－安全技术　Ⅳ.TP393.4

中国版本图书馆CIP数据核字（2003）第040379号

责任编辑：王春宁
印　　刷：北京大中印刷厂
出版发行：电子工业出版社　www.phei.com.cn
　　北京市海淀区万寿路173信箱　邮编：100036
经　　销：各地新华书店
开　　本：787×980　1/16　　印张：22.75　字数：524千字
版　　次：2003年6月第1版　　2003年6月第1次印刷
定　　价：39.00元

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系调换。联系电话：（010）88211980　68279077

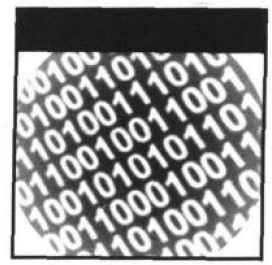
译 者 序

Java从版本1.1开始引入加密体系结构（JCA），它是访问和开发Java平台密码功能的框架，其中包括数字签名和消息摘要API。版本1.2对该结构进行了较大的扩展。Java密码扩展（JCE）是一组包，提供了加密框架及其实现、密钥生成和消息认证码（MAC）的算法，现已成为Java SDK 1.4的核心组成部分。

本书介绍了如何用Java的加密方法提高数据的安全性，重点介绍了存储、消息完整性和身份验证密码学。作者Nick Galbreach是一位著名的安全问题专家，曾长期从事敏感信息存储和转换的加密策略研究，相信对欲了解Java密码API的读者会有所帮助。

与纯粹的密码学书籍不同，本书用简明实用的示例代码来阐述理论问题和具体的实现方法，并为数据库和Web应用程序的安全提供了详细的Java语言实现指南和代码示例。全书包括7章。第1章介绍了基本的数值理论，这一章是理解全书内容的基础。第2章和第3章介绍了公开密钥和保密密钥密码学标准。第4章所介绍的随机数和随机数发生器是密码学的核心。第5章介绍了Java密码API,密码包中的类按字母顺序（而不是包顺序）列在了附录A中。第6章介绍了短消息的编码和加密，这对数据库的查找和索引很有意义。第7章是前面几章内容的实际运用，介绍了如何在应用程序中使用和实现数据编码和加密，内容包括应用程序和数据库设计、口令和记号的使用、密钥管理和日志等等。

本书主要由曾振宇、白克壮、尹喆、纪宁、梅开、文达、杨开开、成文、颜先杰、任映梅翻译，参与翻译的还有苏泳民、单力、汪梓鸣、魏莲方、戚叶等，梁敏、史荣光、李昕怡对本书进行了全面的审校。不当和疏漏之处敬请广大读者批评指正。



前言

此书是专为那些接触过一点或没接触过密码学的软件工程师编写的。在此，作者尽量避免写成一本密码式的天书。这类书皆容易陷入两种误区：百科全书式的和记述式的或者纯粹是API式的描述。作者的目的是通过对密码学的整体介绍，试图在两者之间架起桥梁，同时提供应用实例和用法来达到这一目的。另外，许多书都只是大量地关注公开密钥技术，而以作者的经验，公开密钥密码系统最常用的是第三方应用程序（虚拟专用网、电子邮件）或现行的协议（如SSL和SSH）。

虽然在C和C++语言中有许多优秀的密码库，然而Java扮演着参考书目的角色，因为：

- ◆ 在新型商务和服务器应用程序中非常流行。
- ◆ 自动管理内存，消除了各种类型的错误（堆栈破碎和缓冲区溢出）。
- ◆ 它提供了标准密码API，虽然不完美，但这是我们能够得到的“通用”API。

Java密码API分散在几个包中。附录A按字母顺序列出了这些类，而不是像通常的做法那样按包列出这些类。作者发现这比在不同的包间翻来翻去有用多了。作者试图对示例源代码加以限制，使它相对简单或能说明要点或本身就是实用程序。更复杂（也可能更有用）的示例没有被采用，这是因为对一个安全应用程序

此为试读，需要完整PDF请访问：www.ertongbook.com

来说，错误和异常处理是相当烦琐的，并且实际上没有增加多少价值。作者一直不喜欢将CD-ROM中的内容大量写进书中，同样，也从没发现源代码章节对说明问题特别有用。相关的Web站点（www.wiley.com/compbooks.galbreath）提供了全部的源代码和更多的示例。

由于时间和篇幅的限制，许多主题不能讨论。特别是：

- ◆ XML密钥管理、加密和签名模式。
- ◆ 秘密项目，包括秘密“密封”、Gramm-Leech-Biley 1999法案和HIPPA保密条例。
- ◆ 具有更多细节性的数据库技巧和技术。

希望在后续版本的书中会详细论述这些主题。

绪 论

本书的目的是针对Java密码学及其应用，向系统程序员做一个全面的介绍。与诸多课本不同，此书重点不在“传输”（让两方秘密通信）密码学上。对于大多数应用程序来说，那些需求可由SSL或SSH协议处理，或者由第三方应用程序解决。本书把重点放在存储、消息的完整性和身份验证密码学上（即所谓的“单点”技术，它在客户应用程序中用得更普遍）。除了纯密码学外，本书还介绍了许多使该密码学有用的“辅助”技术。

第1章论述了基本逻辑和数值运算，对两者的介绍都局限在理论层面上，并有Java数值模型的规范。对许多读者来说，这可能是个温习，但我发现许多系统程序员恰恰很少接触此类底层细节。他们或者从没学过这方面内容，或者更可能是因为它不是每天工作常用的，所以被遗忘了。那些对C和C++语言更熟悉的读者会发现，第1章对于翻译Java代码非常有用。

接下来的两章介绍科学、数学以及秘钥和公钥密码学的标准。在这里读者将会学到为什么算法可工作以及它们之间的种种运算变换。虽然数学不是日常编程任务常用的，但目的是让读者熟悉常出现的可能影响信息判断的词汇。

第4章讨论随机数和随机数生成器。虽然从技术上讲不是密码学，但随机数是它的本质。那些开发游戏系统的读者会发现，此章内容非常有用。算法示例是用Java编写的，但也很容易用其他编程语言实现。

第5章将介绍Java密码API。Java SDK已经带有一些很好的文档，但是我想

ETS1941107

把它们整理一下，并将JCA和JCE合并成协调的形式。同样，当编程需要时，我制作了附录A——Java密码类参考。附录中的类按字母顺序（而不是包）排列。可以发现，这种组织方式使读者不必总是在各个包间翻来翻去，并且可以更好地理解API。

第6章介绍短消息编码。像第4章一样，从技术上说，它不是密码学范畴，但当它涉及到密码时总是个问题。在此，读者可以学到如何将二进制数据（也许加密了）转换成各种ASCII码形式；当将数据嵌入URL、生成密码和密码记号时，这很重要。

第7章将所有内容综合在一起，并讨论许多主题：应用程序和数据库设计、密码和记号的使用、密钥管理和日志。

本书使用了大量的源代码例子，可以在Web站点www.modp.com查到这些源代码。本书中许多例子需要稍加修改以后才能作为产品使用，读者也肯定愿意按自己的需要修改错误。某些情况下，如果代码太长也不很明了，那么我不在书中列出，而是让读者到Web站点查看。我发现一页一页地印源代码也不是特别有意义。

遗憾的是，由于时间和篇幅的限制，许多重要主题不能一一包括。数字签名也只是提了一下，虽然它值得写得更多些，但我发现，对于基本应用程序来说，数字签名不是很有用。或者因为大多数人不愿使用数字签名或已经存在信任关系，或者因为人们更愿意用散列或MAC进行加密。最后，当我确实需要使用数字签名时，通常可以用第三方应用程序来处理它。还有其他的主题也没有得到应有的论述，包括用来创建包含加密数据文档的新XML和SAML标准、数据库中的嵌入式密码（如最新的Oracle数据库可做到这一点）。密钥管理和数据库设计也可多加论述，也许将来的版本会加以补救。

目　录

此为试读，需要完整PDF请访问：www.ertongbook.com

第1章 位和字节

在进入密码算子细节前，我们先复习一下位和数值基数的一些基本内容。对于使用操作系统、底层协议、嵌入式系统的读者，本章的内容可能是复习一下以前所学的知识；对于其他每天都与信息技术和基本软件开发打交道，而不直接与字节信息打交道的人来说，这些信息是为理解本书余下部分所包含信息的主要基础。

1.1 一般运算

大多数密码功能直接作用于数字的机器表示。接下来的内容是关于数值如何以不同的进制表示、电脑如何用一组位的形式表示数、如何直接操作位。这部分是用计算机和语言类方式表达的，下节将着重论述 Java 模型。

1.1.1 数的基数

在日常操作中，我们用十进制表示数，数字是从 0~9；这样给出一个十进制数字串 $d_n d_{n-1} \cdots d_2 d_1 d_0$，数值是：

$$10^n d_n + 10^{n-1} d_{n-1} + \cdots + 10^2 d_2 + 10 d_1 + d_0$$

推而广之可将基数换成任何其他数x，它有x个不同的数字，数值可表示为：

$$x^n d_n + x^{n-1} d_{n-1} + \cdots + x^2 d_2 + x d_1 + d_0$$

在计算机领域，最主要的基数是2，或者说数值的二进制表示，即每个数字或位不是0就是1。十进制数30可用二进制表示成11110或16+8+4+2。十六进制（或者叫基数16）也经常用，数字是0~9以及分别代表10~15的A，B，C，D，E和F。数30可以用十六进制表示成1E或16+14。二进制、十进制和十六进制数位间关系列在表1.1中。

表1.1 二进制、十进制和十六进制表示法

二进制	十进制	十六进制
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	10	A
1011	11	B
1100	12	C
1101	13	D
1110	14	E
1111	15	F

使用不同进制时，数的基数也许是模糊的。例如，99是二进制还是十六进制（=9 × 16+9）？在这种情况下，通常在十六进制数前加0x或者只加x（例如：99写成0x99或x99）。二进制数也可能发生同样的问题，101是十六进制101还是二进制101（4+1=5）？为避免二进制数与其他进制数混淆，通常在数字串后加个下标2，例如101_2。

任何数都可以用另一个正的基数b表示出来；但是转换不是很容易，我们将在较后的章节中讨论一般的基数转换。但是，如果知道一个基数是另一个基数的幂，那么两者之间的转换就容易多了。例如，十进制数1234用基数10可表示为：1 × 1000+2 × 100+3 × 10+4。但是，1234也可以看成是基数为100的两个“数字”（12）和（34），值为12 × 100+34 × 1。数相同，值也相同，只是位被重组了。这

些特点对底为 2 的数很重要，一个二进制数字串，将其分为每 4 位一组，这样易于转换成十六进制数，计算成十六进制值为：

```
10011100 = 1001 1100 = 9C
```

1.1.2　位和字节

位是计算机可赖以工作的最小信息单位，它可采用两个值“1”和“0”，虽然根据上下文有时用“是”和“否”。在多数现代计算机中，我们不直接处理“位”，而是一串位，有时称做一个“字”，最小的一个字是一个“字节”。今天，一个字节默认为 8 位，但技术上它可以是 4~10 个位。除了旧的或实验用 CPU 系统还在用外，奇数值在实际应用中已无人用了。为了精确，许多标准用“8 位组”（octet）来表示 8 位。但是我们更愿意用通常的“字节”（byte）。现代 CPU 一般在大得多的字长上运行：术语“32 位微处理器”也就是说，在一个时钟周期内，CPU 基本上在32 位字上运行。当然也可在8位字上运行，但不意味着运行得更快。许多CPU 也有特殊的说明，可在更大的字上工作，如 SSE 和有类似指令的多媒体系统，还有 PowerPC 上的矢量处理。

每个 0~255 的整数都可用基数为 2 的一个字节来表示。位 n 表示值 2^n，字节的值就是这些位的和。一个数的表示可用位展开，位 0 在右，位 7 在左：

$$(b_7b_6b_5b_4b_3b_2b_1b_0) = 2^7b_7 + 2^6b_6 + 2^5b_5 + 2^4b_4 + 2^3b_3 + 2^2b_2 + 2^1b_1 + 2^0b_0$$

或用十进制表示：

$$(b_7b_6b_5b_4b_3b_2b_1b_0) = 128b_7 + 64b_6 + 32b_5 + 16b_4 + 8b_3 + 4b_2 + 2b_1 + b_0$$

例如：00110111 = 32 + 16 + 4 + 2 + 1 = 55

左边的位称为“最有意义的位”，因为它们对整个数值的贡献最大。同样，右边的位叫“意义最小的位”。这种排列方式叫做“大尾”（Big-Endian），我们将在后面讨论。

1.1.3　有符号字节

负数可以用多种方式表示。最简单的是将一个位取反来代表这个数的符号，不管是正还是负。0~6 的位表示数，位 7 代表正负。这样可表示 -127~127 的数，但有两个零值：一个正零（+0）和一个负零（-0）。虽然有两个零值很奇怪，但工作起来很方便。较大的问题是何时发生溢出，如：127 + 2 在无符号运算时为 129（或 1000001），而在有符号运算时，其值却是 -1。

最常用的表示法是“2的补码”。给定x，它的负数是将所有的位反转（1变成0，0变成1）并加1，或处理成-1-x（值不变）。例如，在表1.2中，1+127值是-128。虽然这种方法有些怪，但有很多好处。微处理器只能给一个加循环编码，而与之互补的循环可作为加和减两种运算（事实上，许多CPU在做减法时处理原值的补码）。另外的好处体现在发生转换时，即将一个字节转换成较大的字时（见下节）。

表1.2 2的补码表示法

无符号值	有符号值	十六进制表示	二进制表示
0	0	00	00000000
1	1	01	00000001
2	2	02	00000010
126	126	7d	01111110
127	127	7f	011111111
128	-128	80	10000000
129	-127	81	10000001
130	-126	82	10000010
253.	-3	fd	111111101
254	-2	fe	111111110
255	-1	ff	111111111

1.1.4 位运算符

常用的数学运算功能，如加法和乘法，将字解释成数值，并执行合适的运算。其他运算直接作用于位而不考虑它们的数值表示。这些运算就是位运算或逻辑运算。下节中的例子虽然是用8位字节表示的，但实际上它们可扩展到任何字长。

如表1.3所示，根据上下文是编程还是书面表示，运算用不同的符号来表示。需要时，本书将采用编程符号，除了XOR，因为脱字符号（^）在一些系统中用来表示指数。

表1.3 位逻辑运算和符号

运算	C风格表示法	C风格的自赋值	书面表示
非a	~a		¬a
a与b	a & b	a & = b	a ^ b
a或b	a | b	a | =b	a ˇ b
a异或b	a^b	a^= b	a⊕b

此为试读，需要完整PDF请访问：www.ertongbook.com