一窥软件背后世界之究竟

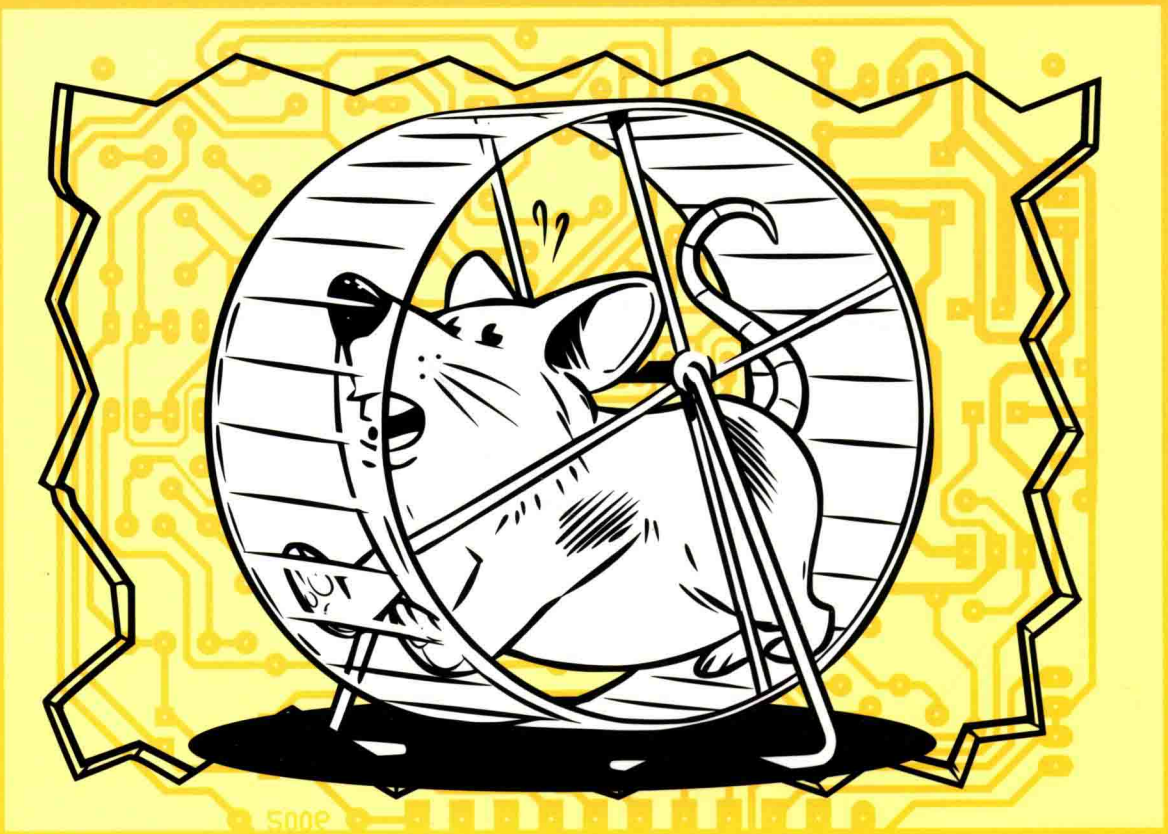


# 软件的奥秘

## ——加密、密码、压缩、搜索是如何工作的

## The Magic Behind Encryption, CGI, Search Engines, and Other Everyday Technologies

[美] V. Anton Spraul 著　　解福祥 译



中国工信出版集团　人民邮电出版社 POSTS & TELECOM PRESS


# 软件的奥秘

## ——加密、密码、压缩、搜索是如何工作的

## The Magic Behind Encryption, CGI, Search Engines, and Other Everyday Technologies

[美] V. Anton Spraul 著　解福祥 译

人 民 邮 电 出 版 社
北 京

**图书在版编目（CIP）数据**

软件的奥秘 : 加密、密码、压缩、搜索是如何工作的 / (美) 斯普劳 (V. Anton Spraul) 著 ; 解福祥译. — 北京 : 人民邮电出版社, 2017.9
ISBN 978-7-115-46199-5

Ⅰ. ①软… Ⅱ. ①斯… ②解… Ⅲ. ①软件工程 Ⅳ. ①TP311.5

中国版本图书馆CIP数据核字(2017)第183673号

## 版权声明



◆ 著 [美] V. Anton Spraul
译 解福祥
责任编辑 陈冀康
责任印制 焦志炜

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 http://www.ptpress.com.cn
北京鑫正大印刷有限公司印刷

◆ 开本：800×1000 1/16
印张：12.5
字数：277 千字 2017 年 9 月第 1 版
印数：1 – 2 400 册 2017 年 9 月北京第 1 次印刷

著作权合同登记号 图字：01-2016-1242 号

定价：49.00 元

**读者服务热线：(010)81055410 印装质量热线：(010)81055316**
**反盗版热线：(010)81055315**
**广告经营许可证：京东工商广登字 20170147 号**

# 内容提要

软件已经成为人们日常生活与工作中常见的辅助工具，但是对于软件的工作原理，很多人却不是非常了解。

本书对软件的工作原理进行了解析，让读者对常用软件的工作原理有一个大致的了解。内容涉及数据如何加密、密码如何使用和保护、如何创建计算机图像、如何压缩和存储视频、如何搜索数据、程序如何解决同样的问题而不会引发冲突以及如何找出最佳路径等方面。

本书适合从事软件开发工作的专业技术人员，以及对软件工作原理感兴趣的读者。

## 作者简介

V. Anton Spraul 已经为来自世界各地的学生讲授了 15 年以上的入门编程和计算机科学。同时他也是《Think Like a Programmer》（No Starch 出版社）和《Computer Science Made Simple》（Boardway 出版社）这两本书的作者。

## 技术审阅者简介

Randall Hyde 是《The Art of Assembly Language》和《Write Great Code》这两本书（No Starch 出版社）的作者，同时他还是《The Waite Group's Microsoft Macro Assembler 6.0 Bible》这本书的合著者。Hyde 在加州大学河滨分校（University of California, Riverside）讲授汇编语言已有 10 多年。在过去的 12 年中，他一直在为核反应堆控制台编写软件。

# 致谢

本书是在一群才华横溢的编辑们的塑造和指导下才得以出版，他们分别是Alison Law、Greg Poulos、Seph Kramer、Hayley Baker、Randall Hyde、Rachel Monaghan 和 No Starch 出版社的“Big Fish”以及 Bill Pollock。除了全体编辑工作人员之外，我还要对在 No Starch 共事过的每个人的支持和帮助表示衷心的感谢。

其中对我帮助最大的是 Mary Beth 和 Madeline，她们是最棒的妻子和女儿，没有她们的爱与支持，这本书是不可能成形的。

# 前　言

科幻小说家 Arthur C. Clarke 曾写道：“任何足够先进的技术都与魔法无异”。如果我们不知道事物背后的原理，倒不妨用超自然力量来解释。不过若是按照这个标准来说的话，那我们无异于生活在一个充满魔法的时代。

今天软件已经融入了我们的生活，例如在线交易、电影特效以及流式视频播放等这样的日常事务。我们逐渐淡忘了在过去的生活中不但没有 Google 搜索，甚至连计划一次汽车旅行的路线时，都不得不先打开一份烦琐的地图。

但很少有人知道这些软件背后的工作原理。与过去的很多创新不同，对软件来说，你无法把它拆开来一窥究竟。所有的一切都发生在计算机芯片内，无论那些计算设备正在执行一项了不起的任务，还是压根就没有启动，这些在我们看来并没有什么两样。想要了解程序的工作原理，似乎需要花费多年的学习成为一名编程人员才可以。因此，难怪许多人会认为软件超出了普通大众的理解，而只有那些技术精英才能掌握其中的奥秘。但实际上这种想法是不对的。

## 本书适用对象

其实任何人都能了解软件的工作原理，所需的只是好奇心而已。无论你只是对技术突然感兴趣，还是一名成长中的编程人员，抑或是二者之间，本书都适合你。

本书涵盖了在软件中常用的一些技术，并且没有任何代码，因此无须事先学习计算机的运行原理。为了达到此效果，我简化了一些技术和细节，但这并不代表本书只是些高度概述，相反，本书含有丰富的细节讲述，使你能从中学到真正的“干货”，从而真正了解这些程序背后的工作原理。

## 章节内容

计算机在现代生活中无处不在，我能拿出来讲的主题无穷无尽，这里我只挑选了那些对我们的生活最有影响以及解释起来最有趣的主题。

- **第 1 章：加密**使我们可以混淆数据以便只有我们自己能够访问数据。当你锁定你的手机或是通过密码保护.zip 文件时，你就是在使用加密技术。我们将在本章中看到在现代加密软件中是如何组合使用那些不同的加密技术的。
- **第 2 章：密码**是指我们在加密数据时，或是在远程系统上识别自己时所用到的密钥。在本章中你将会了解到那些加密技术是如何使用密码的，并且还会学到一些必要的手段去保护密码的安全。
- **第 3 章：网络安全**是我们在进行安全的在线购物时，或是访问自己的账户时需要用到的技术。对传输数据的加密需要用到一种不同的加密技术——公钥加密。此外你还将了解到一个安全的网络会话如何利用前 3 章中涵盖的所有技术。
- **第 4 章：电影 CGI** 是纯粹的软件魔法，它用数学的概念创造了电影的整个世界。在本章中你将了解到软件是如何接管传统的 cel 动画的，此外还能学到在用软件制作整个电影场景时背后所涉及的核心技术。
- **第 5 章：游戏图形**之所以令人印象深刻，不仅在于它们的视觉效果，还在于它们如何能在仅仅几分之一秒的时间内就创造出了这些效果。我们将在本章探讨在游戏开发过程中会用到的一些技巧，并讲述它们是如何在没有充足的时间去利用上一章介绍的那些技术时，还能产生惊人的画面效果的。
- **第 6 章：数据压缩**可以缩减数据，以便我们可以更充分地利用存储和带宽。我们将在本章中探讨缩减数据的最佳方法，并且还会了解到如何将这些技术组合起来去压缩蓝光光盘和网络流中的高清视频。
- **第 7 章：搜索**是指即时查找数据，可以是搜索自己计算机上的文件，也可以是搜索整个网络。我们将在本章中探讨如何组织数据以满足快速搜索，以及如何通过网络搜索返回最有用的结果。
- **第 8 章：并发**使得多个程序可以共享数据。若没有并发技术，那多人视频游戏将不可能实现，在线银行系统一次只能处理一位客户的需求。此外我们还将讨论一些可以使不同处理器访问相同数据并且不会相互影响的方法。
- **第 9 章：地图路径**就是我们从地图网站或车载导航仪中获得的那些即时路线。你将在本章中了解到软件世界对地图的定义，以及一些用于找出最佳路径的特殊搜索技术。

## 寄语

我认为分享这些知识很重要。我们不应该生活在一个我们不了解的世界中，如果不了解软件背后的工作原理，就不可能了解当今的世界。Clarke 的话可以看做一种警告，那些了解技术原理的人可以去糊弄那些不了解的人。例如，一家公司可能会声称其用户登录信息的失窃对其客户不会造成什么影响。果真如此吗？为何呢？读完本书后，你就将会知道此类问题的答案。

除此之外，关于为何要了解软件的工作原理还有一个更好的理由：因为这些技术实在是太酷了。而且我认为最好的魔法技巧在你了解其秘诀后会让你觉得更加神奇。继续读下去，你就会明白我的意思。

# 目　录

# 第1章

# 加密

我们每天都在依赖软件来保护我们的数据，但大部分人对这种保护原理知之甚少。为何浏览器角落的“锁”图标意味着输入信用卡卡号是安全的？手机设置密码后是如何保护内部数据的？是什么真正阻止了其他人登录你的网络账号？

计算机安全是一门保护数据的科学。从某种程度来说，计算机安全相当于通过技术来解决由技术带来的问题。不久以前，大部分数据并没有以数字化形式存储，办公室有文件柜，床底下有放照片的鞋盒。当然，那时你不能方便地与世界各地的朋友们分享你的照片，也不能用手机核查银行存款余额，但是，也没人可以窃取你的私人数据，除非去偷走它本身。而如今，你的私人数据不仅可以被远程窃取，甚至在银行打电话问你为何买了数千元的礼品卡之前，你可能都不知道它们已经被窃取了。

在本书的前 3 章中，我们将讨论计算机安全背后最重要的一些概念。在本章中，我们先来讨论加密。加密使我们能够加密数据，并且只有我们自己才能解密数据。接下来的两章讨论的技术都是以加密为基础，并且需要依赖完整的安全套件，因此加密才是计算机安全的核心。

## 1.1 加密目标

想象一下你计算机上的文件，可能是文本、照片、电子表格、音频或者视频，你希望能够访问这些文件但又对其他人保密，其实这就是计算机安全的基本问题。为了使这些文件保密，你可以将其加密（encryption）成一种不可读的新格式，直到把它们解密（decryption）回原有格式。原有文件是明文（plaintext，即便不是文本文件），加密后的文件是密文（ciphertext）。

攻击者（attacker）是指那些企图擅自对密文进行解密的人。加密的目标就是创建一个密文，在便于授权用户解密的同时，又使得攻击者几乎不可能解密，这“几乎”是许多安全研究人员头痛的来源。就像没有什么锁是绝对牢不可破的一样，加密也绝非不能被解密。在足够多的时间和足够强的计算能力的前提下，任何加密方案在理论上都是可以破解的。因此计算机安全的目标，就是加大攻击者解密的难度，使解密所需的计算资源远超攻击者所拥有的资源，以致攻击行为在实际上不可能成功。

与其一头扎进错综复杂的基于软件的加密中去，不如先从一些简单的例子开始。尽管多年来加密强度有了很大的改进，但这些经典技术是一切加密的基础。稍后，你将看到在现代数字化加密方案中，这些技术是如何结合使用的。

## 1.2 换位法：相同的数据，不同的顺序

加密数据的一个最简单的方法叫作换位（transposition），简单来说就是“改变位置”。换位是我和我的朋友们在上小学传纸条时所用到的一种加密技术。由于纸条会经过一些不能信任的人，所以必须使除我们之外的其他人无法理解这些纸条。

为了使信息保密，我们使用了一个简单、易恢复的方法来重新排列字母的顺序。假设我需要分享的重要情报是 CATHY LIKES KEITH。为了加密信息，我复制了明文的每第 3 字母（忽略任何空格）。第一遍处理完消息后，我复制了 5 个字母，如图 1-1 所示。

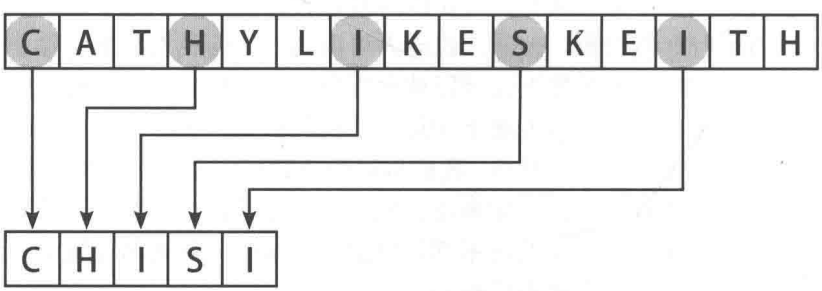



图 1-1　示例信息的第一遍换位处理

在到达信息尾部后，我从头开始并继续从剩下的字母中选出每第 3 个字母，经过第二遍处理后的结果如图 1-2 所示。

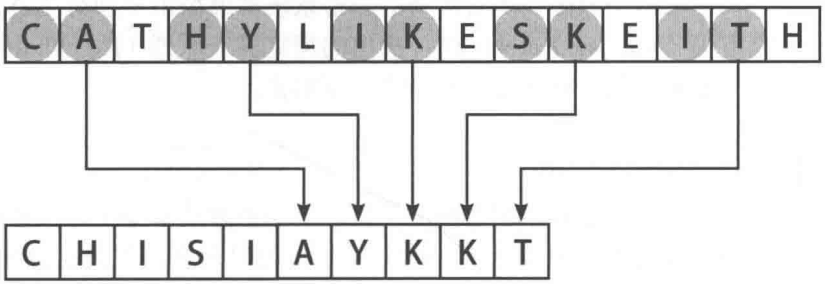



图 1-2　第二遍换位处理

最后，我复制了剩下的所有字母，如图 1-3 所示。

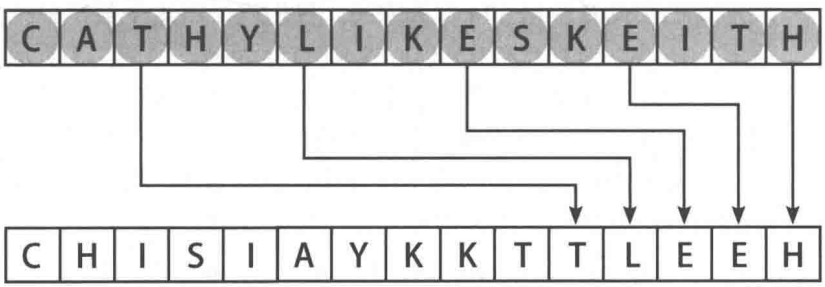



图 1-3　最后一遍换位处理

最终生成的密文是 CHISIAYKKTTLEEH。我的朋友们通过逆转换位的过程，就能够读懂这条信息。逆转处理的第一步如图 1-4 所示，将所有的字母逆转回原位即可展现明文。

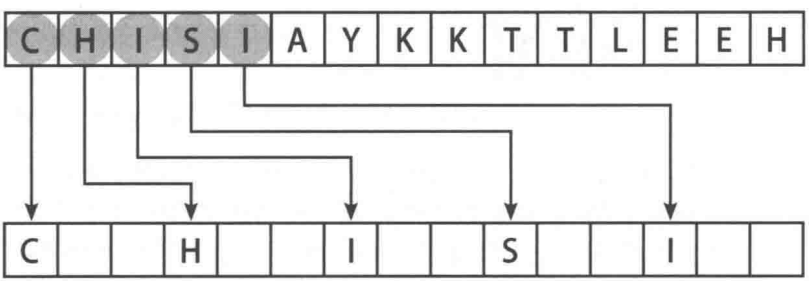



图 1-4　解密时逆转换位的第一遍处理

这个基本的换位方法使用起来很有趣，但是它的加密性非常弱。最大的问题在于泄密——某个朋友将加密方法泄露给了圈外的人。一旦发生这种情况，即使发送的是加密过的消息也将不再安全，加密本身变得没有任何意义。可悲的是泄密是不

可避免的，不仅只有小学生会泄密，每个加密方法都很容易被泄漏出去。使用一个特定加密方法的人越多，就越有可能泄漏。

出于这个原因，所有优秀的加密系统都遵循由早期荷兰密码学家奥古斯特·柯克霍夫（Auguste Kerckhoffs）提出的柯克霍夫原则（Kerckhoffs' principle）：数据的安全不应依赖于加密方法本身来保证加密。

### 1.2.1 密钥

这就引发了一个明显的问题，如果加密方法本身不是保密的，那我们该如何安全地加密数据？答案在于加密方法本身可以公开，但是必须使用密钥（cipher key 或 key）去保证不同数据对应的加密结果也不同。为了理解什么是密钥，先来解释一个更常见的换位方法。

在该方法中，发送任何消息之前，发送方和接收方共享一个密码。假设我和我的朋友们都同意使用 374 作为密码。我们将使用这个数字来改变生成密文时的换位模式。还是以消息 CATHY LIKES KEITH 为例，如图 1-5 所示。我们密码中的数字决定了应该将明文中的哪一个字母复制到密文。因为密码的第一个数字是 3，所以明文的第 3 个字母 T 就成为了密文的第 1 字母。而密码的下一个数字是 7，所以下一个要复制的是 T 后面的第 7 个字母，也就是 S。然后，是 S 后面的第 4 个字母。经过这样的处理后，密文的前 3 个字母是 TST。

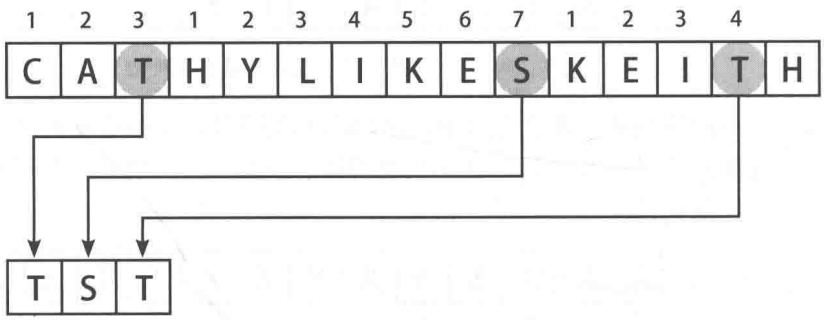



图 1-5　使用密钥 374 来进行第一遍换位处理

图 1-6 展示了接下来的两个字母是如何被复制到密文的。接着上次的位置开始（图中圈 1 位置），往后数 3 个位置，若到达尾部则回到明文头部并继续进行，因此选到了 A 作为密文的第 4 个字母。而下一个要复制的字母是 A 后面数 7 个位置，且需要跳过我们已经复制过的字母，所以是 K。继续这样的处理直到所有明文的字母都被换位。

这个密码 374，就是我们说的密钥。其他人即使截获了该信息，甚至知道我们使用的是换位加密法，但在没有密钥的情况下，仍然无法解密信息。密钥可以定期更改，以防止泄密。

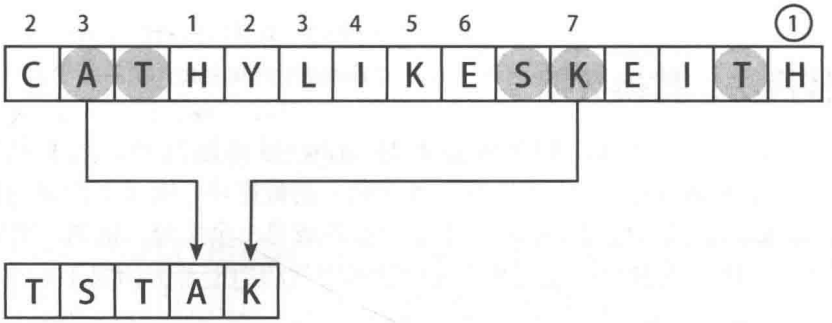



图 1-6　使用密钥 374 进行第二遍换位

### 1.2.2　对加密的攻击

即使没有密钥，攻击者仍然可以通过其他手段去尝试恢复明文。加密后的数据可以通过暴力攻击（brute force），尝试所有的可能性以找出密文所使用的加密方法。对于一条使用了换位加密的信息来说，暴力攻击将检测密文的所有排列。由于暴力攻击几乎总会被使用，所以攻击者用来找出明文所需要尝试的次数，是一个很好的判断加密强度的基准。在我们的示例中，消息 CATHY LIKES KEITH 大约有 400 亿种排列。

这是一个巨大的数字，所以聪明的攻击者将会使用一些常识以便更快地恢复明文，而不是一味地蛮用暴力攻击。如果攻击者可以假设明文都是英文，那么在测试之前就可以排除掉很大一部分的排列。例如，攻击者可以假设明文不会以字母 HT 开始，因为没有英文单词会是以这些字母开头，这就有 10 亿的排列不需要攻击者去检测了。

如果攻击者有了信息中单词的一些线索，那就可以更机智地破译出明文。在我们的示例中，攻击者可能猜到消息中会包含同班同学的名字，所以他们可以先找出密文中的字母能够组成哪些名字，然后再确定剩下的字母可以组成哪些单词。

这种猜测明文内容的手段被称为 cribs。最强的一种 crib 是已知明文攻击（known-plaintext attack）。想要采取这种攻击方式的话，攻击者必须能够获知明文 A、对应的密文 A 以及与密文 A 使用了相同密钥加密的密文 B。尽管这种情况听起来不太可能，但它确实会发生。人们经常会使一些他们认为不再是机密的文档处于无保护状态，而并没有意识到这些文档也可能会被用于协助攻击其他文档。已知明文攻击很强大，当你同时拥有明文和密文时，要找出其中的换位模式将会很容易。

对已知明文攻击最好的防御，就是形成良好的安全实践，例如定期更改密码。不过即使有最好的安全实践，攻击者也总是能够获得明文内容的一些线索（这就是为什么他们会如此有兴趣去读那些无保护文档的原因）。在许多情况下，他们将知道大多数的明文，并且可能还会获知明文——密文对。所以一个良好的加密系统应该使 cribs 和已知明文对攻击者无用。