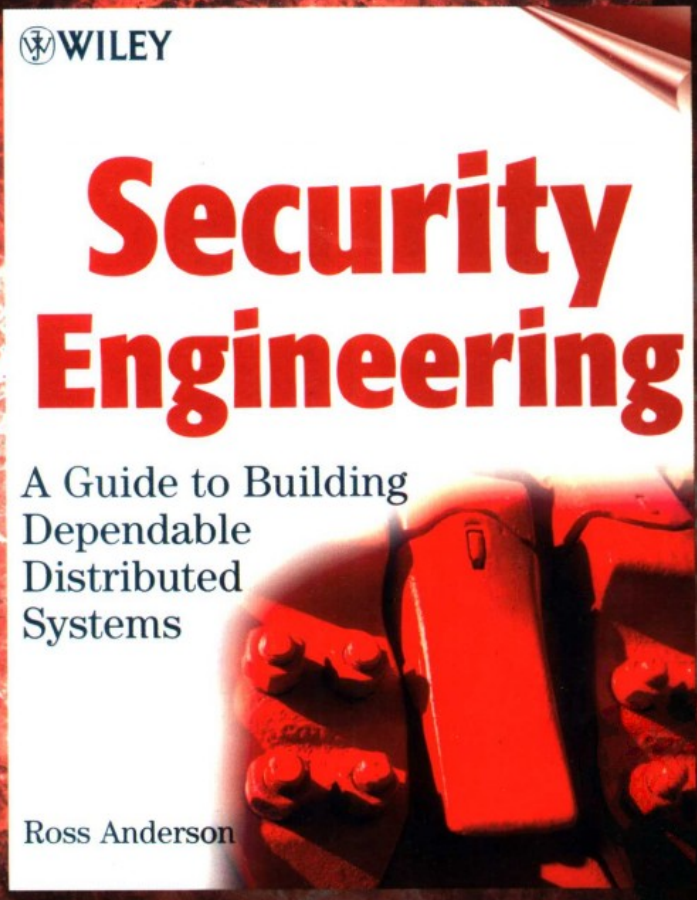
计 算 机 科 学 丛 书

信息安全工程

(英) Ross J. Anderson 著 蒋佳 刘新喜 等译



Security Engineering

A Guide to Building Dependable Distributed Systems

机械工业出版社
China Machine Press

“如果你正在考虑从事安全工程方面的工作，请读本书。这是所有已经出版的有关端到端安全设计和安全工程的著作中最好的一本。”

——Bruce Schneier（世界著名安全技术专家）

PC机和服务器的Internet安全问题现在已经非常令人忧虑，但是很快将会有比PC多得多的设备：移动电话、电冰箱、防盗警报器、心脏监视仪连在网上。面对无数的接入设备和空前复杂的信息系统，安全技术人员该怎样应对因此带来的风险呢？

一般讲述计算机安全的书往往把主要篇幅放在访问控制机制、密码算法和协议等局部内容上，本书则以前所未有的广度讨论了各种信息安全方面的问题。书中充满了奇闻轶事和战争故事，可读性强、观点新颖，并融入了大量最新的研究成果。

本书根据作者在剑桥大学讲授信息安全工程课程的讲义写成，凝聚了作者在安全技术领域20多年来的丰富实践经验和10多年教学经验，可以作为计算机和信息安全专业高校教材。对于从事各种类型信息系统安全的专业人士而言，也是极具价值的参考书。

本书内容：

- 安全工程基础，从协议、密码系统和访问控制到分布式系统的细节
- 生物测量学、安全印章、版权等等诸多安全保护机制深入阐释——本书提供的许多内容均是第一次以图书形式出版
- 各种不同系统可能遭受的各种形式的攻击，从银行、医疗系统到警报器、智能卡、移动通信以及电子商务，同时还提供了相应的防护措施
- 管理和策略问题，会正确处理计算机安全与法律和社会文化的相互影响

作者简介

Ross J. Anderson 任教于英国剑桥大学，并负责领导剑桥大学计算机实验室安全领域的研究。20世纪70年代毕业于剑桥大学，师从已故的ACM 会士 Roger Needham。此后在航空电子界、银行界、信息安全界工作多年，领导了数字水印等许多前沿技术的开发。1992年回到剑桥大学任教。他是世界公认的安全权威之一，在各种实际安全系统的设计和理论研究方面涉猎广泛，发表了众多学术论文。

封面制作 刘永青

ISBN 7-111-09587-1

9 787111 095873

HZ BOOKS

华章图书

网上购书：www.china-pub.com

北京市西城区百万庄南街1号 100037

读者服务热线：(010)68995259，68995264

读者服务信箱：hzedu@hzbook.com

http://www.hzbook.com

ISBN 7-111-09587-1/TP · 2226

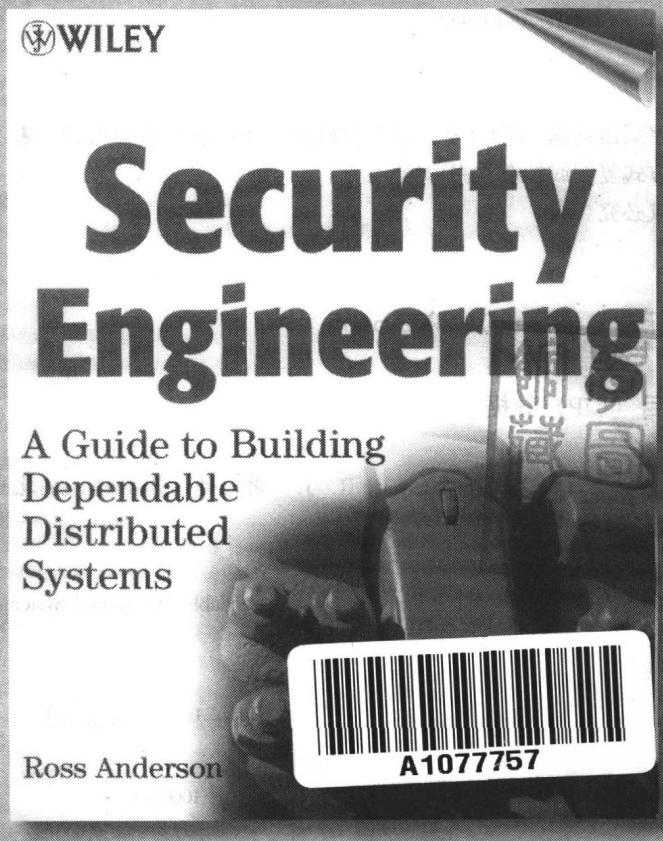
定价：49.00 元

计 算 机 科 学 丛 书

TP309
2A 213

信息安全工程

(英) Ross J. Anderson 著 蒋佳 刘新喜 等译



Security Engineering

A Guide to Building Dependable Distributed Systems

机械工业出版社
China Machine Press

H/J5.65/07

本书是当今世界安全工程领域的权威 Ross J. Anderson 的呕心之作，内容涵盖安全工程的方方面面。具体内容包括：基本安全工程概念描述，协议、加密、访问控制以及分布式系统的安全细节，生物测量学、安全印章、版权等诸多安全保护机制。本书提供的许多内容均是第一次以书籍形式出版；剖析了各种不同系统可能遭受的形式各异的攻击，从银行、医疗系统到报警系统、智能卡、移动通信以及电子商务，同时还提供了相应防护措施；本书还包括如何正确处理计算机安全与法律和社会文化相互影响的管理和策略问题。

本书内容丰富、表述准确、概念清晰、针对性极强，堪称安全工程方面的鸿篇巨著。适合所有对安全工程感兴趣的读者。

Ross J. Anderson: Security Engineering: A Guide to Building Dependable Distributed Systems (ISBN: 0-471-38922-6).

Authorized translation from the English language edition published by John Wiley & Sons, Inc.

Copyright © 2001by Ross J. Anderson.

All rights reserved.

本书中文简体字版由约翰-威利父子公司授权机械工业出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

版权所有，侵权必究。

本书版权登记号：图字：01-2002-5455

图书在版编目（CIP）数据

信息安全工程/ （英）安德森（Anderson, R. J.）著；孙彦妍等译. -北京：机械工业出版社，2003.8

（计算机科学丛书）

书名原文：Security Engineering: A Guide to Building Dependable Distributed Systems

ISBN 7-111-09587-1

Ⅰ.计… Ⅱ.①安…②孙… Ⅲ.信息系统-安全工程-理论 Ⅳ.C202

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：李 炎

北京瑞德印刷有限公司印刷·新华书店北京发行所发行

2003年8月第1版第1次印刷

787mm×1092mm 1/16 · 31.75印张

印数：0 001-5 000册

定价：49.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

出版者的话

文艺复兴以降，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；也正是这样的传统，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，还揭橥了学术的源变，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、从业人员较少的现状下，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图文信息有限公司较早意识到“出版要为教育服务”。自1998年开始，华章公司就将工作重点放在了遴选、移译国外优秀教材上。经过几年的不懈努力，我们与Prentice Hall，Addison-Wesley，McGraw-Hill，Morgan Kaufmann等世界著名出版公司建立了良好的合作关系，从它们现有的数百种教材中甄选出Tanenbaum，Stroustrup，Kernighan，Jim Gray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、研究及庋藏。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襄助，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，有的还专诚为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近百个品种，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，为进一步推广与发展打下了坚实的基础。

随着学科建设的初步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步入一个新的阶段。为此，华章公司将加大引进教材的力度，在“华章教育”的总规划之下出版三个系列的计算机教材：除“计算机科学丛书”之外，对影印版的教材，则单独开辟出“经典原版书库”；同时，引进全美通行的教学辅导书“Schaum's Outlines”系列组成“全美经典学习指导系列”。为了保证这三套丛书的权威性，同时也为了更好地为学校和老师们服务，华章公司聘请了中国科学院、北京大学、清华大学、国防科技大学、复旦大学、上海交通大学、南京大学、浙江大学、中国科技大学、哈尔滨工业大学、西安交通大学、中国人民大学、北京航空航天大学、北京邮电大学、中山大学、解放军理工大学、郑州大学、湖北工学院、中国国

家信息安全测评认证中心等国内重点大学和科研机构在计算机的各个领域的著名学者组成“专家指导委员会”，为我们提供选题意见和出版监督。

这三套丛书是响应教育部提出的使用外版教材的号召，为国内高校的计算机及相关专业的教学度身订造的。其中许多教材均已为M. I. T.，Stanford，U.C. Berkeley，C. M. U. 等世界名牌大学所采用。不仅涵盖了程序设计、数据结构、操作系统、计算机体系结构、数据库、编译原理、软件工程、图形学、通信与网络、离散数学等国内大学计算机专业普遍开设的核心课程，而且各具特色——有的出自语言设计者之手、有的历经三十年而不衰、有的已被全世界的几百所高校采用。在这些圆熟通博的名师大作的指引之下，读者必将在计算机科学的宫殿中由登堂而入室。

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，但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。教材的出版只是我们的后续服务的起点。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：

电子邮件：hzedu@hzbook.com
联系电话：（010）68995264
联系地址：北京市西城区百万庄南街1号
邮政编码：100037

专家指导委员会

（按姓氏笔画顺序）

尤晋元	王　珊	冯博琴	史忠植	史美林
石教英	吕　建	孙玉芳	吴世忠	吴时霖
张立昂	李伟琴	李师贤	李建中	杨冬青
邵维忠	陆丽娜	陆鑫达	陈向群	周伯生
周克定	周傲英	孟小峰	岳丽华	范　明
郑国梁	施伯乐	钟玉琢	唐世渭	袁崇义
高传善	梅　宏	程　旭	程时端	谢希仁
裘宗燕	戴　葵			

序

在和 Roger Needham 合写的一篇论文中，Ross Anderson 发明了一个说法“给撒旦的计算机编程”，用来表述计算机安全工程师们面临的问题。这使我对 Ross 一直充满期望，从那以后我也总是使用这个说法。

计算机编程本身并没有多难：埋头苦干，直到计算机能够按预定的方式工作。大型应用程序和操作系统更加复杂些，但是方法基本上相同。编写可靠的计算机程序就困难多了，因为程序需要面临各种随机发生的错误的考验，因为我们每天使用的实际上都是“墨菲计算机(Murphy's computer)”[⊖]。人们已经对设计可靠的软件进行了大量研究，很多关键任务（mission-critical）的应用软件为了冲破墨菲定律的宿命，进行了精心设计。

编写安全的计算机程序完全是另外一回事。安全的内容不仅包括保证程序正常工作、没有随机故障，还要日复一日地顶住聪明而恶毒的对手在最不可能的时间以最不可能的方式发出的攻击。真的是在给撒旦的计算机编程！

安全工程不同于其他任何一种编程。在我的书《网络信息安全的真相》、我的每月时事通讯《Crypto-Gram》，还有其他作品中我都曾反反复复强调这一点。Ross 也在本书的每一章里阐述了这个观点。这就是为什么如果你正在从事甚至只是想要从事安全工程的话一定要看这本书的原因。这是第一本，也是惟一一本全面讲述现代安全设计和工程学方面的书籍。

本书的出版非常及时。因特网的历史可以分为三次浪潮。第一波以大型机和终端为核心。那时计算机很昂贵、很稀少。第二波是从 1992 年至今，核心是个人计算机、浏览器和大型应用程序。第三波将从现在开始，你将看到现在网络上的、没联网的和非计算机的各种设备都连接到一起。到 2003 年，会有比计算机更多的移动电话连上因特网。几年之后，我们将看到很多电冰箱、心脏监听器、公路铁路自动售票机、警报器和电子仪表都能以 IP 协议相互通信。个人计算机将只是因特网的一小部分。

安全工程，尤其在第三次浪潮中，需要你转变思维方式。你不是去了解一个系统怎样工作，而是了解怎样让它不工作。你必须设想计算机内部有一个聪明的、不怀好意的敌人（还记得撒旦的计算机吗?），要时刻准备新的方法制服它。你必须考虑会使系统崩溃的各种可能，它们大多数是与设计本身无关的。你必须后向、颠倒、横向地考虑每件事情。你必须像外星人一样思维。

正如已故伟大的科幻小说家 John W.Campbell 所说：“外星人的思维像人类，但不是人类。”计算机安全与此非常相似。Ross 属于那些具有外星人思维的少数人之列，现在他把这种思维传授给了人类。祝你阅读愉快。

Bruce Schneier[⊖]

2001 年 1 月

⊖ 语出墨菲定律：“凡是可能出错的，必将出错。”所谓墨菲计算机就是肯定会出问题的计算机。——编者注

⊖ Bruce Schneier 是国际著名的安全专家和密码学家，名著《应用密码学》的作者。——编者注

前　言

多少年以来，人们使用锁、围墙、签名、印章、账簿和计量仪器来定义和保护自己的财产及隐私。这些方式已经受到了人类社会的认可，从各国通用的国际条约到人们的风俗习惯。

时代是飞速变化的。现在，从银行账目到不动产登记，大部分记录都是电子化的；并且随着购物转移到网络上，各种交易也日益电子化。同样重要但并不那么显著的是，很多日常系统也悄然自动化了。报警器再也不会吵醒邻居，而是悄悄地向警察局发送消息；学生再也不用向宿舍的洗衣机、甩干机投硬币，而是使用能在学校书店充值的智能卡（smartcard）；锁不再是简单的机械物，而是可以通过电子远程控制器或者刷卡来操纵；人们不再租借录像带，而是从人造卫星或者有线电视频道看电影。甚至连最小额的钞票也不再是简单地把油墨印在纸上，它可能带有数字水印，这样很多假币都可以被机器检验出来。

这些安全技术到底有多好呢？很不幸的是，诚实的回答是："一点也不像应该的那么好"。新系统经常很快就崩溃，同样的小错误在一个又一个应用软件上重复发生。要实现正确的安全设计，经常需要尝试四、五次，这些步骤实在是太多了。

媒体经常报道互联网上的安全漏洞；银行要阻止客户从提款机上"神不知鬼不觉地提款"（phantom withdrawal）；VISA 报道说网络信用卡交易产生的争议大幅增加；卫星电视公司追捕着复制智能卡的盗版者；执法部门试图用控制加密的法律监视网络空间。还有更糟糕的——设备的交互式特性。如果不小心按了手机的某个键，重播了上一个号码，这在原来可能只是小事——可是如果有人发明了一台机器，每一次它的号码被呼叫时就送出一罐软饮料。突然间你发现手机账单上有 50 瓶可乐时，谁来负责任？电话公司、手机制造商还是自动售货机的经营者？一旦影响你生活的绝大部分电子设备都连网了（微软预期到 2010 年会这样），"网络安全"对你来说意味着什么，你又怎样应付它？

系统除了会出故障之外，很多系统还不能很好地工作。医疗记录系统不允许医生们随意共享个人健康信息，但是仍然不能防止好奇的私家侦探。预算很高的军事系统禁止没有"绝密"许可的任何人获得情报数据，但是却常常设计成几乎每个人工作时都需要这个许可。设计旅客票务系统是为了防止乘客作弊，但是当联邦反垄断检查官分拆了铁路部门以后，他们仍然不能阻止新的铁路公司彼此欺骗。只要设计者们当初多了解一点前人做过的努力和失败，这些问题都可以预见到。

安全工程学是一门新学科，它从所有这些混乱中孕育而生。

尽管人们很了解大部分基础技术（密码学、软件可靠性、防篡改、安全印刷和审计等），但是却缺乏有效应用它们的知识和经验。由于各领域都从机械机制一下子转换到了电子机制，工程界还来不及吸取以前的经验教训。我们反复地看到人们在发明同样的四方车轮。

转型最可能成功的行业通常是那些可以从其他学科借鉴到技术的领域。例如，把军事上鉴别敌友的设备重新应用到银行现金机上，甚至应用到汽油预付费仪表上。因此，即使一个安全设计者在某个领域有高深的技术（不管他是研究密码的数学家，还是研究钞票印制油墨

的化学家），他在纵观整个学科领域时还是会很拘谨的。安全工程的本质在于了解系统的潜在威胁，然后采用合适的混合保护措施（既有技术上的，也有组织上的），来控制这些威胁。了解别人做过的工作，更重要的是了解经验教训，对进行决策是莫大的帮助，还可以节省很多开支。

本书的目的是，以21世纪的眼光对安全工程学进行完整的介绍。本书适用于以下四个方面：

- 作为介绍安全工程学的教材，你可以花几天时间从头到尾读完它。本书主要是给需要了解这门学科的IT专业人员准备的，也可以做大学一个学期的教程。
- 作为参考书，你可以寻找某类系统的操作概述。本书介绍的系统包括提款机、出租车仪表、雷达干扰发射机和匿名医疗记录数据库等。
- 介绍一些基础技术，例如密码学、访问控制、推理控制、防篡改和印章。由于篇幅有限，我没有深入讲解；但是对每门学科做了基本说明，给有兴趣的读者提供了阅读列表（还给研究生提供了一个公开研究问题表）。
- 作为一本有创新的科技作品，我尽力概括了安全工程学需要的基础原理，总结了人们建造系统时应该了解的经验教训。我多年来一直从事安全工作，对此印象深刻。例如，设计者若不知道对于序列密码的简单攻击，他设计的普通防空火力控制雷达很容易受到干扰；同时，由于印制钞票和设计版权标志的人不了解雷达领域熟知的技巧，以至于大部分数字水印遭受到了非常普遍的攻击。

我已尽力使这本书保持中立风格；安全工程的书必须如此，因为很多基本技术都是美国的，而很多应用程序都是欧洲的（假如你了解美国大学和研究实验室具有更多的资金，欧洲有更多样化的民族和市场，这就不足为奇了）。此外，欧洲很多成功的革新（从智能卡到GSM移动电话，再到按次计费的电视服务），穿越了大西洋，现在在美国也已非常盛行。科学和案例研究都是必要的。

本书来自于我在剑桥大学讲授的安全工程学教程。我重新整理了笔记，使其具有完整的体系，并增加了许多材料。本书适合于专业的安全管理者和顾问参考；适用于研究密码学的计算机科学教授；适用于要侦破电脑骗局的警探；也适用于为了规范加密和匿名而冥思苦想的策略制定者。最重要的是，面向Dilbert。我的读者主要是程序员或者工程师，他们要设计不管客户、经理和其他任何人怎样做都能长久运转的实用系统。

本书分为三个部分：

- 第一部分着眼于基本概念，由安全协议这个重要概念开始，继而介绍人机界面问题、访问控制、密码学和分布式系统问题。除了具备基本的计算机知识以外，这部分不需要特别的技术背景。第一部分基于我给二年级大学生开的《安全入门》课程。
- 第二部分更详细地介绍了许多重要的应用，比如军事通信、医疗记录系统、提款机、移动电话和付费电视。这些用来介绍更多的高级技术和概念。第二部分还从不同角度（比如公司、客户、罪犯、警察和间谍）考虑了信息安全问题。这些材料来源于我在安全方面开设的高级教程、研究工作和经验咨询。
- 第三部分讲解机构和策略问题：计算机安全怎样与法律、证据和公司策略相互影响；我们怎样才能有信心让系统按照意图正常工作；如何最好地管理整个安全工程产业。

我相信，在21世纪，建立一个面对恶意攻击时稳健工作的系统是工程师们面临的最重要、最有趣，也是最艰巨的任务之一。

Ross J. Anderson

2001年1月于英国剑桥

注：参加本书翻译工作的还有张兴凯、方建、刘辉、辛冉、周鹏、李家峻、郝丽杰、吴淑英、邓多多、李欣、成洁易、王政、桂黎明、方巍巍、孙淑芳。

目　录

出版者的话
专家指导委员会
序
前言

第一部分

第 1 章　什么是安全工程 …… 2
1.1　例一：银行 …… 2
1.2　例二：空军基地 …… 3
1.3　例三：医院 …… 4
1.4　例四：家庭 …… 5
1.5　定义 …… 5
1.6　小结 …… 8
第 2 章　协议 …… 9
2.1　偷听口令的风险 …… 9
2.2　谁去那里？简单的认证 …… 10
2.2.1　质询和响应 …… 12
2.2.2　米格战斗机中间人攻击 …… 13
2.2.3　反射攻击 …… 15
2.3　伪造消息 …… 16
2.4　环境变更 …… 16
2.5　选择协议攻击 …… 17
2.6　管理密钥 …… 18
2.6.1　基本密钥管理 …… 18
2.6.2　Needham-Schroeder 协议 …… 19
2.6.3　Kerberos …… 20
2.7　走向形式化 …… 21
2.7.1　一个典型的银行智能卡协议 …… 21
2.7.2　BAN 逻辑 …… 21
2.7.3　认证付费协议 …… 22
2.7.4　形式化认证的局限性 …… 23
2.8　小结 …… 24
研究问题 …… 24
参考资料 …… 24
第 3 章　口令 …… 25
3.1　基础 …… 25
3.2　实用心理问题 …… 26
3.2.1　社会工程 …… 26
3.2.2　可靠口令输入的困难 …… 27
3.2.3　记住口令的困难 …… 27
3.3　系统问题 …… 29
3.3.1　保护自己还是保护他人 …… 29
3.3.2　入侵检测问题 …… 30
3.3.3　可以培训用户吗 …… 30
3.3.4　日益缺乏数据安全 …… 31
3.4　口令的技术保护 …… 32
3.4.1　口令输入攻击 …… 32
3.4.2　口令存储攻击 …… 33
3.4.3　绝对限制 …… 35
3.5　小结 …… 35
研究问题 …… 36
参考资料 …… 36
第 4 章　访问控制 …… 37
4.1　引言 …… 37
4.2　操作系统访问控制 …… 38
4.2.1　组和角色 …… 39
4.2.2　访问控制列表 …… 39
4.2.3　Unix 操作系统安全 …… 40
4.2.4　Windows NT …… 41
4.2.5　权能 …… 42
4.2.6　Windows 2000 增加的新特性 …… 43
4.2.7　粒度 …… 44
4.2.8　沙盒和携带证据代码 …… 44
4.2.9　对象请求代理 …… 45
4.3　硬件保护 …… 45
4.3.1　Intel 80x86/Pentium 处理器 …… 46
4.3.2　ARM 处理器 …… 46
4.3.3　安全处理器 …… 47
4.3.4　其他处理器 …… 47
4.4　哪里出了问题 …… 47
4.4.1　击毁堆栈 …… 48
4.4.2　其他攻击技术 …… 48
4.4.3　用户界面失败 …… 49

4.4.4 为何这么多地方出现错误 …… 50
4.4.5 补救措施 …… 50
4.4.6 环境蠕变 …… 51
4.5 小结 …… 52
研究问题 …… 52
参考资料 …… 52
第5章 密码学 …… 54
5.1 引言 …… 54
5.2 历史背景 …… 55
5.2.1 早期序列密码：vigenère 表 …… 55
5.2.2 一次一密法 …… 56
5.2.3 早期的分组密码：Playfair 方法 …… 57
5.2.4 单向函数 …… 58
5.2.5 非对称基本加密方法 …… 59
5.3 随机预言模型 …… 60
5.3.1 随机函数：哈希函数 …… 61
5.3.2 随机序列生成器：序列密码 …… 63
5.3.3 随机置换：分组密码 …… 64
5.3.4 公钥加密和陷门单向置换 …… 65
5.3.5 数字签名 …… 66
5.4 对称加密方法 …… 67
5.4.1 SP 网络 …… 67
5.4.2 高级加密标准 …… 70
5.4.3 Feistel 加密 …… 71
5.5 操作模式 …… 74
5.5.1 电子密码本模式 …… 74
5.5.2 分组密码链接 …… 75
5.5.3 输出反馈 …… 75
5.5.4 计数器加密模式 …… 76
5.5.5 密码反馈模式 …… 76
5.5.6 消息验证码模式 …… 77
5.6 哈希函数 …… 77
5.6.1 基础加密的额外要求 …… 78
5.6.2 常用哈希函数及其应用 …… 79
5.7 非对称加密方法 …… 80
5.7.1 基于因数分解的加密 …… 80
5.7.2 基于离散对数的加密 …… 81
5.7.3 特殊目的的签名方法 …… 84
5.7.4 认证 …… 85
5.7.5 非对称加密方法的强度 …… 86
5.8 小结 …… 86
研究问题 …… 87
参考资料 …… 87
第6章 分布式系统 …… 89
6.1 并行 …… 89
6.1.1 使用陈旧的数据与呈扩散状态的花费 …… 89
6.1.2 通过锁定防止不一致的更新 …… 90
6.1.3 更新的顺序 …… 91
6.1.4 死锁 …… 91
6.1.5 不收敛的状态 …… 91
6.1.6 安全时间 …… 92
6.2 容错和故障恢复 …… 93
6.2.1 故障模型 …… 93
6.2.2 恢复什么 …… 94
6.2.3 冗余在什么层 …… 95
6.2.4 拒绝服务攻击 …… 96
6.3 命名 …… 96
6.3.1 分布式系统的命名观点 …… 97
6.3.2 哪里出了问题 …… 99
6.3.3 名字的类型 …… 102
6.4 小结 …… 102
研究问题 …… 103
参考资料 …… 103

第二部分

第7章 多级安全 …… 106
7.1 引言 …… 106
7.2 什么是安全策略模型 …… 106
7.3 Bell-LaPadula 安全策略模型 …… 107
7.3.1 密级和许可 …… 108
7.3.2 信息流控制 …… 109
7.3.3 Bell-LaPadula 模型的标准批判 …… 110
7.3.4 可选模式 …… 111
7.3.5 Biba 模型 …… 112
7.4 多级安全系统的几个例子 …… 113
7.4.1 SCOMP …… 113
7.4.2 Blacker …… 114
7.4.3 MLS Unix、CMW 和 Trusted Windowing …… 114
7.4.4 NRL 泵 …… 115
7.4.5 后勤系统 …… 115
7.4.6 紫色的 Penelope …… 116
7.4.7 未来的 MLS 系统 …… 116
7.5 哪里出了问题 …… 117
7.5.1 组合系统 …… 117

7.5.2 串联问题 …… 118
7.5.3 隐蔽通道 …… 118
7.5.4 病毒威胁 …… 119
7.5.5 Polyinstantiation …… 120
7.5.6 其他一些实际问题 …… 120
7.6 MLS更广泛的含义 …… 122
7.7 小结 …… 123
研究问题 …… 123
参考资料 …… 124
第8章 多边安全 …… 125
8.1 引言 …… 125
8.2 分割、长城和BMA模型 …… 126
8.2.1 分割和网格模型 …… 126
8.2.2 长城模型 …… 128
8.2.3 BMA模型 …… 129
8.2.4 比较分析 …… 133
8.3 推理控制 …… 133
8.3.1 在医学推理控制中的基本问题 …… 134
8.3.2 推理控制的其他应用程序 …… 134
8.3.3 推理控制理论 …… 135
8.3.4 一般方法的局限性 …… 139
8.3.5 缺陷保护的代价 …… 140
8.4 剩余问题 …… 141
8.5 小结 …… 142
研究问题 …… 142
参考资料 …… 143
第9章 银行业和簿记系统 …… 144
9.1 引言 …… 144
9.1.1 簿记的起源 …… 144
9.1.2 复式簿记 …… 145
9.2 银行电脑系统如何工作 …… 146
9.2.1 Clark-Wilson安全策略模型 …… 146
9.2.2 责任的分离 …… 147
9.2.3 哪里出了问题 …… 149
9.3 大规模支付系统 …… 151
9.3.1 全世界银行间金融电信协会(SWIFT) …… 151
9.3.2 哪里出了问题 …… 152
9.4 自动柜员机 …… 154
9.4.1 ATM的基础 …… 154
9.4.2 哪里出了问题 …… 156
9.4.3 实际应用 …… 158
9.5 小结 …… 159
研究问题 …… 160
参考资料 …… 160
第10章 监控系统 …… 161
10.1 引言 …… 161
10.2 报警器 …… 161
10.2.1 威胁模式 …… 162
10.2.2 为什么不能保护一幅画 …… 163
10.2.3 传感器失灵 …… 164
10.2.4 特征交互 …… 165
10.2.5 攻击通信系统 …… 166
10.2.6 经验教训 …… 168
10.3 预付费仪表 …… 169
10.3.1 需给电表 …… 170
10.3.2 系统如何工作 …… 171
10.3.3 什么地方会出错 …… 171
10.4 计程器、转速表以及卡车速度限制器 …… 173
10.4.1 哪里出了问题 …… 174
10.4.2 对策 …… 176
10.5 小结 …… 179
研究问题 …… 179
参考资料 …… 179
第11章 核武器的指挥与控制 …… 180
11.1 引言 …… 180
11.2 肯尼迪备忘录 …… 181
11.3 无条件安全认证码 …… 181
11.4 共享控制系统 …… 182
11.5 防篡改与指定行动链接 …… 184
11.6 条约验证 …… 185
11.7 哪里出了问题 …… 185
11.8 保密还是公开 …… 186
11.9 小结 …… 187
研究问题 …… 187
参考资料 …… 187
第12章 安全印刷和印章 …… 188
12.1 引言 …… 188
12.2 历史 …… 188
12.3 安全印刷 …… 189
12.3.1 威胁模型 …… 189
12.3.2 安全印刷技术 …… 190
12.4 包装和印章 …… 194

12.4.1 基片特性 …… 194
12.4.2 粘贴问题 …… 195
12.5 系统脆弱性 …… 195
12.5.1 威胁模型的特性 …… 196
12.5.2 员工的细心 …… 197
12.5.3 随机失败的效果 …… 197
12.5.4 材料控制 …… 197
12.5.5 不保护正确的事物 …… 198
12.5.6 检查的成本和性质 …… 198
12.6 评估方法论 …… 199
12.7 小结 …… 200
研究问题 …… 200
参考资料 …… 200
第 13 章 生物测量学 …… 201
13.1 引言 …… 201
13.2 手写签名 …… 201
13.3 面部识别 …… 203
13.4 指纹 …… 204
13.5 虹膜编码 …… 208
13.6 声音识别 …… 210
13.7 其他系统 …… 210
13.8 哪里出了问题 …… 211
13.9 小结 …… 213
研究问题 …… 213
参考资料 …… 213
第 14 章 物理防篡改 …… 214
14.1 引言 …… 214
14.2 历史 …… 214
14.3 高端物理安全处理器 …… 215
14.4 评估 …… 219
14.5 中级——安全处理器 …… 220
14.5.1 iButton …… 220
14.5.2 Dallas 5002 …… 221
14.5.3 Capstone/Clipper 芯片 …… 222
14.6 智能卡和微控制器 …… 223
14.6.1 体系结构 …… 224
14.6.2 安全的演化 …… 224
14.6.3 技术现状 …… 229
14.7 哪里出了问题 …… 230
14.7.1 体系结构错误 …… 231
14.7.2 模糊性和评估错误 …… 231
14.7.3 协议失败 …… 232
14.7.4 功能蠕变 …… 233
14.8 什么应该受到保护 …… 234
14.9 小结 …… 235
研究问题 …… 236
参考资料 …… 236
第 15 章 发射安全 …… 237
15.1 引言 …… 237
15.2 历史 …… 238
15.3 技术监视和对策 …… 239
15.4 被动攻击 …… 241
15.4.1 通过电源和信号电缆的泄露 …… 241
15.4.2 通过射频信号的泄露 …… 243
15.5 主动攻击 …… 245
15.5.1 风暴型病毒 …… 245
15.5.2 Nonstop …… 246
15.5.3 假信号脉冲 …… 246
15.5.4 差异故障分析 …… 246
15.5.5 联合攻击 …… 247
15.5.6 商业利用 …… 247
15.5.7 防御 …… 247
15.6 发射安全攻击有多严重 …… 248
15.6.1 政府 …… 248
15.6.2 商业 …… 248
15.7 小结 …… 249
研究问题 …… 249
参考资料 …… 249
第 16 章 电子战与信息战 …… 250
16.1 引言 …… 250
16.2 基础 …… 250
16.3 通信系统 …… 251
16.3.1 信号侦察技术 …… 252
16.3.2 通信攻击 …… 253
16.3.3 保护技术 …… 254
16.3.4 民用与军用的交互 …… 258
16.4 监视与目标探测 …… 259
16.4.1 雷达类型 …… 259
16.4.2 干扰技术 …… 259
16.4.3 高级雷达与反测量措施 …… 261
16.4.4 其他传感器与多传感器问题 …… 262
16.5 敌我识别系统（IFF） …… 262
16.6 定向能量武器 …… 263
16.7 信息战 …… 265
16.7.1 定义 …… 265
16.7.2 学说 …… 266

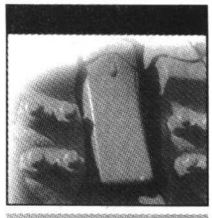
16.7.3 电子战中潜在的教训 …… 266
16.7.4 电子战与信息战的区别 …… 267
16.8 小结 …… 268
研究问题 …… 268
参考资料 …… 268
第 17 章 电信系统的安全 …… 269
17.1 引言 …… 269
17.2 电话盗打 …… 269
17.2.1 对仪表的攻击 …… 269
17.2.2 信号攻击 …… 271
17.2.3 攻击交换机与配置 …… 272
17.2.4 不安全的终端系统 …… 273
17.2.5 特征干扰 …… 274
17.3 移动电话 …… 275
17.3.1 移动电话复制 …… 275
17.3.2 GSM 系统结构 …… 276
17.3.3 通信安全机制 …… 276
17.3.4 下一代产品：3gpp …… 280
17.3.5 GSM 安全：成功或失败 …… 283
17.4 群体欺诈 …… 284
17.5 小结 …… 285
研究问题 …… 286
参考资料 …… 286
第 18 章 网络攻击与防御 …… 287
18.1 引言 …… 287
18.1.1 最普通的攻击手段 …… 287
18.1.2 技术问题：脚本小子和打包防御 …… 288
18.2 网络协议攻击 …… 289
18.2.1 局域网攻击 …… 290
18.2.2 使用因特网协议和机制的攻击 … 290
18.3 防御网络攻击 …… 293
18.3.1 配置管理 …… 293
18.3.2 防火墙 …… 294
18.3.3 防火墙的作用和局限性 …… 295
18.3.4 加密技术 …… 296
18.4 特洛伊、病毒和蠕虫 …… 297
18.4.1 早期的恶意代码 …… 298
18.4.2 因特网蠕虫 …… 298
18.4.3 病毒和蠕虫如何工作 …… 299
18.4.4 竞争 …… 300
18.4.5 近期历史 …… 300
18.4.6 防病毒措施 …… 301
18.5 入侵检测 …… 302
18.5.1 入侵检测类型 …… 302
18.5.2 入侵检测的普遍局限性 …… 303
18.5.3 检测网络攻击的特殊问题 …… 304
18.6 小结 …… 305
研究问题 …… 306
参考资料 …… 306
第 19 章 保护电子商务系统 …… 307
19.1 引言 …… 307
19.2 电子商务的电报史 …… 307
19.3 信用卡介绍 …… 309
19.3.1 欺骗行为 …… 309
19.3.2 伪造 …… 310
19.3.3 自动欺骗检测 …… 310
19.3.4 经济学 …… 311
19.4 在线信用卡欺骗：大肆宣传以及现实情况 …… 312
19.5 密码保护机制 …… 313
19.5.1 安全套接层 …… 313
19.5.2 安全电子事务（SET） …… 314
19.5.3 公共密钥基础设施（PKI） …… 316
19.5.4 电子数据交换（EDI）和 B2B 系统 …… 318
19.5.5 电子钱包和微支付 …… 319
19.6 网络经济 …… 320
19.7 具有竞争力的应用和公司间的冲突 … 321
19.8 还有什么其他容易出现的问题 …… 323
19.9 商家能做些什么 …… 323
19.10 小结 …… 324
研究问题 …… 324
参考资料 …… 325
第 20 章 版权和隐私保护 …… 326
20.1 引言 …… 326
20.2 版权 …… 327
20.2.1 软件 …… 328
20.2.2 书刊 …… 332
20.2.3 音频 …… 332
20.2.4 视频和付费电视 …… 334
20.2.5 DVD …… 340
20.3 信息隐藏 …… 343
20.3.1 DVD 标记概念 …… 343
20.3.2 常规信息隐藏技术 …… 344
20.3.3 对版权标记的攻击 …… 345

20.3.4 版权标记方案的应用 …… 348
20.4 隐私机制 …… 348
20.4.1 内容隐藏：PGP …… 349
20.4.2 内容否认——隐写术 …… 350
20.4.3 联合隐藏——remailer 和译解密码者 …… 351
20.4.4 联合拒绝——数字货币 …… 353
20.4.5 其他应用和问题 …… 354
20.5 小结 …… 358
研究问题 …… 359
参考资料 …… 359

第三部分

第 21 章 电子策略 …… 362
21.1 引言 …… 362
21.2 密码技术策略 …… 363
21.2.1 警方窃听的历史 …… 364
21.2.2 流量分析的历史 …… 365
21.2.3 对外国目标的通信情报 …… 367
21.2.4 密码策略的历史 …… 370
21.2.5 讨论 …… 373
21.3 版权 …… 376
21.3.1 数字千年版权法案 …… 378
21.3.2 即将出现的欧洲法令和 UCITA … 378
21.4 数据保护 …… 379
21.4.1 欧洲数据保护的历史 …… 380
21.4.2 欧洲和美国间的差异 …… 381
21.4.3 目前的趋势 …… 382
21.5 证据的问题 …… 383
21.5.1 证据的有效性 …… 383
21.5.2 证据的可靠性 …… 384
21.5.3 电子签名 …… 384
21.5.4 证据的负担 …… 386
21.6 其他公共部门的问题 …… 387
21.6.1 服务交付 …… 387
21.6.2 社会排挤和歧视 …… 388
21.6.3 税收保护 …… 388
21.6.4 选举 …… 388
21.7 小结 …… 389
研究问题 …… 389
参考资料 …… 389
第 22 章 管理问题 …… 391
22.1 引言 …… 391
22.2 管理安全项目 …… 391
22.2.1 三家超市的故事 …… 392
22.2.2 平衡风险和报酬 …… 393
22.2.3 组织问题 …… 393
22.3 方法论 …… 397
22.3.1 自顶向下设计 …… 397
22.3.2 反复设计 …… 399
22.3.3 来自安全关键型系统的教训 …… 400
22.4 安全需求工程 …… 403
22.4.1 管理需求的发展 …… 404
22.4.2 管理项目需求 …… 408
22.4.3 并行处理 …… 409
22.5 风险管理 …… 410
22.6 经济问题 …… 412
22.7 小结 …… 413
研究问题 …… 413
参考资料 …… 414
第 23 章 系统评估与保证 …… 415
23.1 引言 …… 415
23.2 保证 …… 415
23.2.1 不正当的经济动机 …… 415
23.2.2 项目保证 …… 417
23.2.3 处理保证 …… 418
23.2.4 保证增长 …… 420
23.2.5 进化和安全保证 …… 422
23.3 评估 …… 422
23.3.1 信赖方的评估 …… 423
23.3.2 通用准则 …… 425
23.3.3 什么容易出现问题 …… 427
23.4 前面的路 …… 430
23.4.1 半开放设计 …… 430
23.4.2 开放源代码 …… 431
23.4.3 Penetrate-and-Patch、CERT 和 bugtraq …… 432
23.4.4 教育 …… 433
23.5 小结 …… 433
研究问题 …… 433
参考资料 …… 434
结束语 …… 435
参考文献 …… 437

第一部分

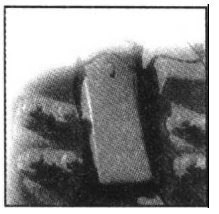


本书这一部分讲述安全工程的基础。第 1 章概述了四种环境下的安全分布式系统：银行、空军基地、医院和家庭。第 2 章讨论安全协议——整个学科的心脏：它们指定了系统内的成员(不管是人、电脑还是其他电子设备)怎样相互通信。第 3 章讨论口令和类似的机制，介绍了一种非常简单的广泛应用在计算机身份认证方面的安全协议，并介绍了建立安全系统所需要的基础。

接下来两章关于访问控制和密码学。即使客户端（假设是电话、个人电脑或其他）得到了服务器的认证——通过口令，或者更加精确的协议——仍然需要控制机制来决定它在服务器上能读写哪些数据和执行哪些操作。最简单的方法就是，先在单一集中系统（访问控制）的上下文中检查，然后再考虑采用多服务器(有可能是在不同的域中)的分布式方法如何实现，这里采用密钥的技术就是密码学。密码学是关于编码和加密的艺术（也是一门科学）。它不仅仅是一种防止别人偷听机密消息的技术。现在密码学已经广泛应用在认证和管理中："确保信源到信宿的安全"[535]。

第一部分的最后一章介绍分布式系统。在这个领域人们感兴趣的研究有并行控制、容错性和命名。由于系统要防止恶意的和随机的失败，上面的研究都需要采用各种新技术。使用过期数据(重复交易和使用别人留下的信用卡)是一个严峻的问题，人们在各种系统中使用不同的名字（邮箱地址、信用卡号码和用户名等）也一样值得关注。很多系统都是因为没有恰当的处理这些问题而遭受失败。

这些章的大部分内容是标准的教学材料，打算采用教学的形式，而不是写成百科全书，因此我没有像本书其余部分那样做太多引证。不过我希望，专家们也能从中发现一些有价值的案例。



第 1 章　什么是安全工程

扭曲的人性，犹如扭曲的木材，成就不出正直的东西。

——康德

世界是不完美的，不管在线还是离线；因此不要过分苛求在线的表现。

——Esther Dyson

安全工程研究如何建立能够面对恶意攻击、错误和灾难的可靠系统。这门学科集中研究如何设计、实现并测试完整的系统和应用现有系统作为背景环境时所需要的工具、程序和方法。

安全工程需要多学科的专业知识——从密码学和计算机安全，到硬件抗干扰和形式化方法，再到应用心理学、组织方法、审计方法和法律知识。系统工程的技术，如商务过程分析、软件工程、评估和测试，都是很重要的；但是这些还不够，因为它们只是针对错误和灾难，而不考虑恶意攻击。

许多安全系统有严格的保障要求。核安全控制系统失败，会危害人类生活和环境；提款机等银行系统失败，会严重破坏经济生活秩序；医疗记录系统失败，会危及个人隐私；付费电视系统失败，会使该行业无法继续生存；汽车防盗系统失败，会使犯罪活动轻易得手。甚至认为系统比实际更脆弱的想法，都会阻碍经济的发展，如人们往往不愿在网上支付信用卡。

一般认为，软件工程是要确保某些事情能够发生（“约翰能看这份文件”），而安全工程则确保某些事情不能发生（“外国政府不能看这份文件”）。实际当中更加复杂。系统对安全的需要是不一样的。普通系统只需要用户认证、交易诚实可信、容错性、消息保密和信息隐藏这其中的几种。但是很多系统都失败了，因为设计者保护的目标错了，或者目标正确但是方法不对。

为了理解各种系统的安全需要，下面快速地看一下四个领域：银行、空军基地、医院和家庭。在给出安全工程师采取保护的具体例子后，就可以理解一些相关的定义了。

1.1　例一：银行

银行要运行大量的对安全要求很苛刻的计算机系统：

- 银行业务的核心通常是一个分支簿记系统。它要保存客户的主账目文件和记载每日业务的分类账目。对该系统的威胁主要是由银行内部员工引起的；每年都有 1% 的银行员工被解雇，其中大部分是因为诈骗（平均额只有几千美元）。主要防卫方法是几百年来一直在完善的簿记程序。例如，账目上的每次借贷都必须有一个等额的相反记录与之匹配；因此钱只可以在银行内流通，既不会增加也不会减少。除此之外，大额转账需要两、三个人许可。还有一些警报系统用来监控反常的交易，员工需要定期休假，休假期间不能踏入银行或者访问银行系统。

- 银行的公共形象是自动柜员机 ATM。基于客户卡和个人身份号码的认证交易——以这种方式同时防御内部和外部攻击——要比看起来困难得多！坏人（或者银行员工）发现了银行系统的漏洞，结果导致很多地方都出现了“梦幻提款”。ATM 是首次在商业上大规模使用密码学，还帮助建立了很多加密标准。
- 在银行系统的后面有很多价值昂贵的消息系统。这些系统用来在本土移动大额钱；进行安全交易；签发信用证和保证书等等。攻击这些系统是老练的白领罪犯的梦想。防卫方法是综合采用簿记程序、访问控制和加密技术。
- 大部分银行部门还有一个大型保险箱或保险库，它的报警器直接与保安公司的控制中心保持通信。采用加密技术防止盗贼伪造通信，在行窃时使警报器发出“一切正常”的信号。
- 近几年来，很多银行连接了因特网，有了自己的网站和设施，允许客户在线处理账目。银行还发行了信用卡允许客户在线购物，然后再从商户那里取得物品。为了保护这种交易，银行采用了标准的因特网安全技术，包括对 Web 浏览器进行 SSL/TLS 加密，或安装防火墙来阻止黑客通过攻击网站服务器从而进入后台主簿记系统。

本书将在后面的章节中讲述这些应用。银行计算机安全无疑是非常重要的。直到最近，银行仍然是计算机安全产品在非军用行业的主要市场，因此对于制定安全标准有着不可估量的影响。再者，即使银行的技术不被国际标准采用，仍然可以在其他行业中得到广泛的应用。最初为银行屋顶设计的警报器现在已用于珠宝店和普通家庭；甚至还装在超市里，用来防止商店员工为得到处理食品而蓄意破坏冷藏柜。

1.2　例二：空军基地

军事系统也是推动技术发展的重要力量。20 年来，它们激励政府向计算机安全领域的学术研究作了很多投资。同银行一样，军事系统不是单一的而是多样化的：

- 一些最复杂的装置比如电子战系统，其目的是在防止己方雷达被阻塞的同时努力阻塞敌方的雷达。信息战这个领域非常具有指导性，因为几十年来，资金充裕的科研实验室已经研制出先进的对抗和反对抗技术等——其伪装策略的深度和广度都是其他任何领域所没有的。作战时使用这些设备可以带来前所未有的洞察力。电子战的主流拒绝服务攻击，现在已经开始在网上出现，政府也开始谈及“信息战”，所以军事上的经验会很有用。
- 军事通信系统有一些特别的要求。仅仅把消息译成密码是不够的：敌人如果看到了用其他人的密钥加密的通信，就可以很容易地找到发送方的位置并进行攻击。小概率窃听（LPI）无线电连接是此类问题的一个答案，这里使用了很多技巧，比如正应用在版权标志上的广谱调制。
- 军事机构有一些最大的后勤和存货管理系统，这些系统有一些特殊的安全要求。例如，可能会在每个安全级别上都设置一个单独的储备管理系统：对喷气机燃料和靴子、鞋油之类的东西采用普通系统，对有可能泄漏战术意图的储备和装置采取二级保密系统（这很像企业给股东和税务人员准备不同的账本，可怜的审计员也遇到类似的问题）。还有带有更高级的保护要求的智能系统和命令系统。一般的规则是，敏感信息不能流入下一级安全限制里去。因此你可以从秘密仓库系统把文件复制到顶

级秘密仓库系统，但是不能反过来。同一规则也适用于通过窃听器收集数据的智能系统：信息必须从调查目标流向智能分析者，但是目标却不能知道在窃听谁的通信。管理带有信息流向限制的多个系统是一个很困难的问题，这激发了人们研究的兴趣。

- 核武器出现后经历了两代人，保护核武器的问题已经引出了很多有趣的安全技术。其中有电子认证系统，防止没有国家命令机构许可的人使用核武器；有封印和警报系统；还有识别人员的高精度的生物认证方法，比如虹膜识别。

其他安全工程师可以从这些技术中学到很多知识。例如，早期有人借鉴广谱无线电的思想，把版权标志插入到数字语音和视频中，但是却抵抗不了同步破坏，现在有很多广谱系统仍然面临这个问题。另外一个例子来自军需品管理，系统的规则是："不要把炸药和雷管装在同一辆卡车上。"这种技术可以推广到更多的领域，比如为了满足卫生要求，禁止把生肉和熟肉放在一起。

1.3　例三：医院

我们继续从食物卫生的角度来谈一下医疗保健。医院里使用的是非常标准的簿记系统或者类似的系统，但是也有一些有趣的保护要求，大部分是与病人的安全和隐私有关的。

- 由于医院采用了基于网络的技术，所以引入了一些新的安全问题。例如，由于药物目录等参考书移到了网络上，医生必须保证那些人命关天的数据（例如每人的剂量）和有关部门公布的完全一样，没有被有意无意地改动过。几年以后，这些安全问题也会影响到其他的 Web 系统。还有另外一个例子，医生需要从家里或者车上用笔记本电脑访问含有病人个人档案的网页，这需要合适的电子认证和加密工具。
- 病人记录系统不应该让所有的员工都能看到每个病人的记录，否则就会造成隐私纠纷。系统应该履行这样的规则："护士可以看到 90 天内在本科室护理过的任何病人的记录。"传统的计算机安全机制很难做到这一点，因为角色会变化（护士从一个科室调到另一个科室）；另外系统之间存在依赖性（病人记录系统依赖于人事系统的访问控制决策，所以人事系统失败了就会牵连到安全、隐私，或者两者都有）。这种应用正激起人们去研究基于角色的访问控制。
- 病人记录通常是匿名的，只作为研究用，但是也很难保证安全。只对病人的姓名加密一般是不够的，比如"查看 1966 年 9 月 15 日锁骨骨折的 59 岁男性病人所有记录"，一般就足以查到已知在大学当运动员时受伤的一位政治家的记录。但是如果记录使用了不恰当的匿名，就必须遵循更严厉的规则来处理数据，这样就增加了医疗研究的费用。
- 新技术可能会引入未知的风险。医院管理员需要用备份程序来处理用电量和电话服务等，但是医疗手段越来越依赖网络，通常没有记录文件。例如，临床科室开始使用网上药品数据库；不再开药品处方；也不再通知额外计划组（contingency planning team）。所以对网络服务进行攻击（例如病毒和分布式的拒绝服务攻击）会给医疗带来严重的后果。

本书会在后面更详细地讲解医疗系统安全。比起银行 IT 和军事系统，这是一个新兴领域，但是在所有的发达国家里，医疗保健在 GNP 里占的比例比银行 IT 和军事系统都大，而且由于医院也开始越来越多地采用 IT，医疗系统也变得重要起来。

1.4 例四：家庭

你可能想像不到普通的家庭也要用到安全的分布式系统。可以看看下面的实例：

- 很多人使用着前面介绍过的一些系统。你通过基于网络的电子银行系统付账；今后你会在网上拥有加密的医疗记录。你的警报器没有动不动就吵醒邻居，而是每隔几分钟向保安公司发送加密的“一切正常”信号。
- 你的车装有防盗锁，向钥匙链上的无线应答器发送加密的质询信号，启动车之前应答器必须正确回应。小偷如果想出售赃车，必须给车新装一个发动机控制器，从而增加了销赃的难度，你因此节省了很多保险费。然而，劫车的次数越来越多了，罪犯狗急跳墙，只有持枪劫车了。
- 早期的移动电话很容易被坏人“克隆”。用户可能忽然间发现账单上多了几百甚至几千美元。现在的 GSM 数字移动电话采用与车锁类似的一种加密质询响应协议进行网络认证。
- 只要你交费，卫星电视的机顶盒就会解译电影；而 DVD 播放器采用基于密码学和版权标志的复制控制机制来限制复制光盘（或者限制在指定的地区以外播放光盘）。
- 在很多国家里，没有信用卡的家庭交电费和煤气费时可以使用预付费仪表，用完智能卡额度后可以在本地商店充值。在很多大学，学生使用复印机、洗衣机甚至喝饮料时也使用类似的手段交费。

你可能已经使用了很多带有保护措施或者电子机制的系统。在未来的几十年里，这样的系统会迅速增加。可是根据以往的经验，很多系统设计得很糟糕。必要的技术还没有广泛传播开来。

本书的目的就是让人们设计更好的系统。为此，工程师和程序员需要了解现有的系统怎样工作，以往哪些系统失败了。土木工程师从倒塌的桥中学到的经验教训远远超过了屹立百年的桥；在安全工程中这个道理也同样成立。

1.5 定义

安全工程中的很多术语都比较直观，但是也有些术语容易引起误解，甚至产生争议。尽管在相关的章节中对技术术语还有更详细的定义，在这里还是要指出容易出问题的地方。

我们首先要阐明系统的含义。所谓系统指的是：

1）一种产品或者组件，例如加密协议、智能卡或者 PC 硬件。

2）再加上操作系统、通信系统和其他东西，构成一个组织的基本结构。

3）再加上一个或者几个应用（账目、工资表、设计等等）。

4）上面的任何一种或者全部，再加上 IT 部门的员工。

5）上面的任何一种或者全部，再加上内部用户和管理层。

6）上面的任何一种或者全部，再加上客户和其他外部用户。

7）上面的任何一种或者全部，再加上周围环境，包括媒体、竞争者、管理机构和政治家。

混淆定义会滋生错误和漏洞。广义地讲，厂商和评估者主要关心第一种（有时也包括第二种）定义，而企业则会关注第六种（有时是第五种）定义。轻视人力因素，从而忽视了可

用性和可靠性，是引起安全故障的一个主要原因，所以我们一般采用第六或第七种定义。当我们缩小范围时，可以从上下文中找到准确的含义。

另一类错误的原因是分不清楚使用者是谁及他们想验证什么。在安全和加密行当里有一个传统，就是通过连续取首字母来辨认安全协议里的当事人，所以我们可以看到诸如此类的声明："Alice 向 Bob 证实了自己"。这样使得事件更容易读懂，但是常常不够精确。我们是指 Alice 向 Bob 证明她的名字确实是 Alice，还是她证明自己已经获得了某种信任？这种证实是由 Alice 本人来做的，还是 Alice 委托智能卡或者软件工具来做的？如果是后者，我们能否肯定那是 Alice，而不是向 Alice 借卡用的 Cherie，或者把 Alice 的卡偷来的 David，甚至黑了她电脑的 Eve 吗？

这里我用*主体*代表客观存在的人（人类、外星人……），它可以是手术员、当事人或者牺牲者。我用*人*既代表客观存在的人，也代表像公司、政府这样的法人。

*当事人*是一个参与了安全系统的实体。这个实体可以是主语、人、角色或者设备，比如 PC 机、智能卡或读卡终端。当事人也可以是一个通信信道（根据需要可以是端口号或者密钥）。当事人还可以是其他当事人的混合体，例如组（Alice 或 Bob）、联合体（Alice 和 Bob 一起做事）、复合角色（Alice 是 Bob 的经理）和委托（Alice 缺席时由 Bob 代替）。注意组和角色是不一样的。*组*（group）指的是一组当事人，而*角色*（role）是由不同人承担的职责（例如"美国军舰尼米兹号上的值班员"、"冰岛医疗协会的临时主席"）。当事人可以进行多层提取。例如，"Alice 缺席时由 Bob 代替"可以理解成"Alice 缺席时 Bob 用自己的智能卡代替她"，或者是"Alice 缺席时 Bob 使用 Alice 的智能卡"。当需要考虑细节的时候，本书会更具体的讲解。

身份（identity）这个词的含义是有争议的。我在这里指的是当两个当事人代表着同一个人或者同一个设备时，他们的名字之间的关联关系。例如，知道"Alice 是 Bob 的经理"中的 Bob 和"Bob 是 Charlie 的经理"、"Bob 作为部门经理和 David 共同签署了银行草案"中的 Bob 是同一个人，这是很重要的。*身份*常常被滥用为仅仅指的是"名字"，这样的惯用语有"用户名"和"居民身份证"。在不会引起歧义的地方，为了简便本书还会如此使用习惯用语。

信任（trust）和*可信任*（trustworthy）的定义常常被混淆。下面的例子可以看出两者的差别：如果有人在巴尔的摩华盛顿国际机场的洗手间中看到一位国家安全局的工作人员向某国外交官出卖重要资料（假设他的行为是未经授权的），我们可以说他"受到了信任，但不可信任"。在这以后，本书会使用国家安全局的定义，如果信任系统或者组件失败了就会破坏安全策略，而可信任系统或者组件则不会。

但要注意信任还有很多种定义。英国军事上的观点是强调可审查和不安全（fail-secure）的特性：信任系统元素就是"工作时通过外部观察看不出来它是否完整"。其他的定义一般是看一个系统是否被批准了权力：信任系统可能是"如果在我值班时它被黑掉了，我不会被解雇的系统"，或者甚至是"我可以买保险的系统"。本书不会用这两个定义。当所指的系统是不易失败的、被批准了的或者买了保险的系统时，本书会直接这样说。

保密（confidentiality）、*隐私*（privacy）和*秘密*（secrecy）的定义也非常容易混淆。这些术语显然有关联，但是不完全相同。如果我的邻居砍掉了栅栏上的藤蔓，结果他的小孩能看到我的花园并戏弄我的狗，这侵犯了我的隐私而不是机密。对前任老板的事务守口如瓶，这

是保护机密而不是隐私。

本书要使用以下的词语：

- 秘密是一个技术术语，指的是用来限制访问信息的当事人数量的机制的效果，比如加密或者计算机访问控制。
- 保密包含了你为其他人或者组织保守秘密的职责。
- 稳私是你保护个人秘密的能力或权利；它可推广到防止别人侵犯个人空间的能力或权利（同一个词，在不同的国家里差别很大）。隐私可以推广到家庭，但不能用在公司一类的法人上。

举个例子，医院的病人具有隐私权，医生、护士和其他员工都有保护病人隐私的职责。在商业事务上，医院没有隐私权，但是那些在医院工作的并有利害关系的员工有保护秘密的责任。简而言之，隐私是关系到个体利益的秘密，而保密是关系到一个机构利益的秘密。

更复杂的事情在于，把消息的内容设为秘密常常是不够的。例如，很多国家的法律都规定治疗性病是秘密，然而如果有人看到你向性病诊所发送密文的话，很容易得出结论：你在那里看病。所以，也必须保护元数据，如消息的来源或目标。在隐私（或者机密）问题上，匿名也是一个和秘密同样重要的因素。甚至还有更复杂的事情，有些作者把我讲的秘密说成消息内容保密，匿名说成消息来源（或目标）保密。

确实性（authenticity）和完整性（integrity）也有微妙的差别。在安全协议的学术讨论里，确实指的是完整性并且有时效的：你在和一个真实的当事人建立通话，而不是重发前面的消息。在银行协议里有类似的思想。如果一个国家的银行法律指明支票在六个月内有效，那么七个月没有兑付现金的支票是完整的（假设没有更改过），只是不再有效了（银行在这个问题上不使用确实这个词）。军事上使用确实性技术来识别当事人和他们发出的命令，而把完整性应用到存储数据上。因此，我们会谈论到电子战威胁数据库的完整性（它没有被敌人或者墨菲破坏），但是谈论将军指令的确实性（这和学术上的使用方法有交叉）。还有一些奇怪的用法，如人们谈论的确实性复制，就是敌方电子作战员发来的欺骗指令；这里确实性指的是复制和存储行为。与之类似，在作案现场警官会把一张伪造的支票装进证物包里，称之为保护证据的确实性。

在这里还要最后讲一些术语，用来描述我们所要达到的目标。脆弱性（vulnerability）是系统或者环境的一个特性，如果和内部、外部威胁（threat）联手的话，会导致安全失败（security failure），这违背了系统安全策略。安全策略（security policy）是系统保护策略的一种简洁描述（例如，“每次借贷都必须有一个等额的相反的记录与之匹配，超过$1000的业务必须有两个经理许可”）。安全目标是更加详细的说明，它陈述了在不同产品中实现安全策略的具体方法——加密和数字签名机制、访问控制、审计日志等等——并作为评价设计者和实现者工作好坏的准绳。在这两者之间还有一个保护框架（protection profile），它很像安全目标，但是写法上与设备无关，可以在不同的产品之间、同一产品的不同版本之间进行比较和评估。本书将在第7章和第23章详细阐述安全策略、安全目标和保护框架。一般来讲，保护这个词指的是确实性或者完整性等特性，给出了抽象的定义，我们要在普通系统的上下文中思考它而不是在具体实现中。

最后要说的就是，安全工程中容易混淆的词语从本质上来说有点政治原因。安全这个词用得太多了，对不同的人常常指不同的东西。对于公司，它可能是指能够监控员工们收发邮

件和上网浏览，而对于员工，安全则可能是指能够自由使用邮件和网络而不被监控到。

1.6　小结

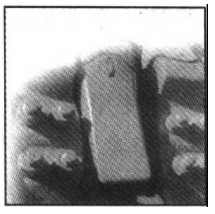
我想起了刘易斯·卡罗尔[⊖]的一段话：

> Humpty Dumpty 很轻蔑地说："当我用词的时候，它表达的就是我要表达的意思——不多也不少。"艾丽斯说道："问题是，你是否能用语言表达这么多不同的东西。"Humpty Dumpty 说："问题是，哪个词表达得最好——那是关键。"

对安全工程师来说，对普通语言在不同的应用中的细微差别敏感一些，能够从中总结出安全策略和目标是很重要的。有些客户有时想侥幸逃脱什么，可能不太方便说出自己的想法，但是一般情况下，想设计稳健的安全系统就必须十分明确保护目标。

⊖　《艾丽斯漫游奇境》的作者。

第2章 协 议



聪明的后果是无法预见的。

——Christopher Stracher

如果安全工程有一个统一的主题的话，那就是对安全协议的研究。本书并不以安全协议的正式定义开始，而是给出一个粗略的表述，然后使用一些例子来增进理解。由于这是一本工程书，还会给出几个协议失败的例子。

安全系统通常包括很多当事人，例如人、公司、计算机和磁卡机，它们通过各种渠道通信，包括电话、电子邮件、无线电、红外线和像银行卡和车票这样的能携带数据的物理装置。安全协议是管理这些通信的规则，它们特意设计可以让系统逃过恶意攻击，比如电话诈骗、敌方政府阻塞无线通信或者制造假火车票。如果要防御所有可能攻击的话，代价通常太昂贵了，所以协议一般都是在假设某些特定危险的情况下设计的。因此，评估一个协议的优劣需要回答两个问题。首先，危险模型是现实的吗？其次，协议能对付这些危险吗？

协议可以是非常简单的，例如想进一栋楼就需要先在门禁上刷卡识别；协议也可能很复杂。全球提款机网络使用几十个协议，规定提款机怎样与客户交互信息、怎样与操作它的银行会话、银行怎样与网络通信、两家银行怎样转账、在不同的当事人之间怎样建立密钥，还有传送哪种报警消息（比如吞卡的指令），这些协议必须在庞大的复杂系统中同时运转。

有些系统表面上看起来没有什么毛病，却常常会被发现有严重的缺点。举个例子，过去很多银行用只有中央计算机和提款机知道的密钥对客户的 PIN 加密，同时把它写到卡的磁条上。这个想法是为了让提款机在本地识别 PIN，节省了通信量，还可以在提款机脱机时启动一些有限的服务。这套系统使用了很多年都平安无事。后来一名从事门禁系统工作的程序员发现，他把妻子的银行账户号码换成自己的，就可以改变他自己的银行卡的磁条。他就可以使用改过的卡和自己的 PIN 从妻子的账户提款。他意识到这个方法可以从任何账户里窃取钱财，于是在几年内偷了几十万。倒霉的银行只好花费几百万更换系统。

所以人们应该系统地看待安全协议及其失败的原因。安全协议应用广泛，通常又设计得很糟糕，本书将列举一些不同的应用实例。

2.1 偷听口令的风险

口令仍然是计算机安全所依赖的基础，因为这是计算机系统验证人类用户身份的主要机制。口令还可以以 PIN 的形式用在很多嵌入式系统中，比如提款机、移动电话还有报警器。口令引起了不少问题，比如人们需要选择一个不容易猜到的口令，还得记住系统随机产生的口令。

本书在下一章中讲述口令的“人界面”（human interface）问题。现在，我们讨论在嵌入式系统中使用口令的局限性。上世纪 90 年代中期以前，用来打开车库或者打开车门的遥控

器是一个典型的应用。这些原始的遥控器仅仅发射 16 位序列号，这些序列号同时也发挥着口令的作用。

使用“抢夺者”（grabber）是一种很平常的攻击，这种设备能够记录信号码，随后重播。“抢夺者”好像产自台湾，在 1995 年左右进入市场。盗贼可以潜伏在停车场内，记录下来打开车门的信号，一旦车主离开就重发这个信号把车门打开。

应付这个问题的一种对策，就是在开车和锁车时分别使用不同的信号。但是这样效果也不理想。首先，贼可以潜伏在你的住所旁，早晨记录下你的开车号码；然后夜间再把你的车偷走。其次，16 位的口令太短了。在 20 世纪 90 年代中期，有一些设备可以对所有可能的口令依次试验。平均试验 2^{15} 次就可以发现口令，如果每秒钟 10 次的话，还不到一个小时就能完成。如果在停车场里的贼身边带上一百个设备，那么不到一分钟就能得手了。

另外一个对策就是把口令的长度加倍，从 16 位变成 32 位。制造商自豪地做广告“超过 40 亿种代码口令”，但是这只能表示他们并没有真正地理解问题。每一辆车还是只有一个口令（或者两个），尽管猜起来是不切实际的，但是贼仍然频频得手。

采用序列号作为口令有一个更大的弱点，就是很多人都能够访问它。比如汽车的例子，可能所有的经销商的员工或许还有政府机动车登记处都有访问权。有些报警器也使用序列号作为主口令；这样更糟糕，因为序列号会在定购单、提货单、发票和其他一切正常的书面文件上显示出来。

采用简单的口令有时候恰如其分，甚至在兼顾使用序列号的时候。举个例子，我在游泳池的月票就只有一个条形码。我肯定能用复印机和层压机伪造出一个，但是转门处有门童，并且服务员也认识“老顾客”，所以没有必要采用更昂贵的安全手段。我上班的实验室的卡式钥匙更难伪造，因为它使用了红外条码。这样也对，我们还有更昂贵的设备放置在有更多的锁的房间里。本书将在第 3 章详细讨论口令。至于像汽车那种很多人想要偷的东西，就需要更好的保护技术了。因此就谈到了加密认证协议。

2.2　谁去那里？简单的认证

在停车楼里用户用来抬起横竿的红外线通行卡就是认证设备的一个简单的例子。它首先发送序列号，然后再发送一个认证块，其中包括和前面相同的序列号和一个随机数，它们都已经用该设备独有的密钥加密过了。

以后再讨论怎样加密和密文应该具备的特性；这里用符号 $\{X\}_{KT}$ 简单地表示用密钥 K 对消息 X 加密。于是汽车上的通行卡和停车场之间的协议可以写成如下的形式：

$$T \rightarrow G:\ T,\ \{T,\ N\}_{KT}$$

这是标准的协议工程符号，开始的时候可能不太明白，所以我们慢慢的讲解。

汽车内的通行卡发送它的名字 T，后面是加密的 T，随后是 N，这里 N 代表“一次性使用的数”或者 nonce。使用 nonce 是为了向接受者保证消息是“新鲜”的，也就是说，不是重发攻击者观察到的旧消息。认证过程很简单：停车场服务器读到 T，获得相应的密钥 KT，把余下的消息解密，核对包含 T 的纯文本，最后收到从没使用过的 nonce：N。

有两处容易产生混淆的地方。首先，在冒号左边 T 代表一个当事人（用户的通行卡），而在冒号右边 T 代表通行卡的名字（也就是，序列号）。其次，在本书开始讨论攻击协议的时候，我们发现从通行卡 T 发往停车场 G 的消息实际上被不速之客 F 截获了，不久后又发

回来。所以这个符号是不合适的，但是它已经深入人心、很难更改了。专业人员一般只把冒号左边的 $T \rightarrow G$ 简单地当成是协议设计者头脑中的一种示意。

nonce 可以指能够保证消息有时效的任何东西。根据上下文，nonce 可以是随机数、序列号，或者第三方发出的随机质询。三者之间有微小的差别，比如它们抵抗各种重发攻击的能力不同（本书将在后面讲解）。但是在低成本的系统中，前两种占优势，因为它们可以更廉价地建立一个单工通信信道。

这种设备的密钥管理也很简单。一般的停车场通行卡密钥 *KT* 就是把序列号在一个通用主密钥 *KM* 下加密，只有中央服务器知道 *KM*：

$$KT = \{T\}_{KM}$$

这就是密钥多样化（key diversification）。它给出了实现访问通行卡的一个简单方法，并在以智能卡为基础的系统中得到了广泛的应用。但是仍然有很多地方会出现错误。

至少两个生产商都犯了一个错误，他们只检查 nonce 与上次发送的不同，这样的话，假设有两个有效码 *A* 和 *B*，那么序列 *ABABAB*…就被译为有效码。对于车锁，小偷只要重播上一次的码就可以把门打开。另外还有一个预付费仪表的例子。英国有一百多万个家庭，加上发展中国家还有几百万的家庭，都安装了先进的电表和煤气表，这样，把加密卡牌买回家插到表里就可以使用相应数量的电和煤气。在南非有一种广泛使用的电表只检查解密命令中的 *nonce* 是否与上次相同。这样的话，顾客买两个小额电票，再反复地交替使用，就可以充电到最高值 [39]⊖。

使用随机数还是计数器，这个问题并不简单 [195]。使用随机数的话，锁必须记住前面很多个码。还要面临随从攻击（valet attack），如代客泊车的服务员，他有通行卡的临时访问权限，可以记录下很多访问码，随后重播它们偷走你的车。

使用计数器的问题是怎样保持同步。一片钥匙可能用在多把锁上，被口袋中的东西挤压了也可能激活（有一次我把试验卡牌带回家，竟被我的狗咬到激活了）。如果计数器增加了几百甚至几千以后，要想办法给它复原。如果在一定条件下让锁向钥匙“学习”，与它保持同步，这个问题就解决了；但是细节一般都设计得不够周到。普通的产品使用 16 位计数器，只要译码以后计数器的值比最后一次使用时增加了不到 16 就可以访问。有时钥匙在别处使用了（或者被宠物抓到）16 次以上，为了处理这种情况，倘若比起上次有效使用时增加的值是 17 ~ 32 767，锁就打开第二个区段（计数器采用循环方式，65 535 的下一个数就是 0）。这引发了重播攻击，因为只需要六个访问码——比如 0、1、20 000、20 001、40 000和 40 001 就可以攻破系统。

所以即使设计一个简单的卡牌认证机制也不容易。有很多种攻击方法并不“违背”加密规则。当加密认证机制迅速增加以后，这类攻击会变得很常见。

举个可能引起争议的例子是附件控制（accessory control）。一般的游戏控制台都要安装质询/响应协议，如果不交许可费的话就不能使用游戏卡或者其他附件。这个做法正在推广。据一位认证芯片的厂商说，有些打印机公司已经开始在打印机中嵌入认证装置，保证它使用原装的色粉盒。如果装上竞争对手的产品，打印机的分辨率就会从 1200 dpi 降到 300 dpi。移动电话的一大部分利润来自于电池，认证协议可以识别竞争对手的产品，并让它消耗得更快。

⊖ 请参阅本书所附“参考书目”中的 [39]。

我猜过不久，墨盒和电池厂家就会想出对策，推出更好的产品，如同不断面世的先进盗车工具一样。

2.2.1 质询和响应

最现代的汽车门锁使用了更复杂的双工协议，一般称为质询/响应协议。把车钥匙插入驾驶锁时，发动机管理设备向钥匙发出一个短程无线质询信号，内容是 n 位的随机数。钥匙把质询信号加密，计算出一个响应信号。这样，用 E 表示发动机控制器，T 表示钥匙上的应答器，K 表示在应答器和发动机控制器之间共享的密钥，N 表示随机质询，协议可以写成下面的形式：

$$E \rightarrow T: N$$

$$T \rightarrow E: \{T, N\}_K$$

这样仍然不够安全。在一个系统中，由发动机管理设备产生的随机数相对是可预测的，因此贼有可能向车主口袋中的钥匙发送质询信号，如果过关，就得到下一次的质询信号。

实际上，大多数应用广泛加密的软件产品——包括 Kerberos、Netscape 和 PGP——都不时地被破解，原因是它们的随机数发生器还不够随机 [340, 256]。各个系统上固定使用的随机数发生器是不同的。使用放射性衰变的方法可以建立硬件随机数发生器，但是因为破坏环境它没有得到普及。可以应用到像 PC 这样的大系统中的随机数来源有多种，例如空气湍流给硬盘旋转速度带来的微小变化 [225]。实际的 PC 系统处于多种随机性环境中，比如网络流量和按键时间，以及源于内部系统的随机性 [363]，这些因素的结合方式是很关键的 [447]。但是在通常的嵌入式系统中（比如车锁），随机质询信号是对计数器加密产生的，其密钥被保留在设备中，不用于其他任何用途。

质询/响应协议不仅仅应用在锁上。很多大机构，包括美国的银行、电话公司和国防部门，都给员工发密码发生器，让他们登录公司的计算机系统 [808]。这些密码发生器看起来像计算器，甚至可以当计算器用，但是它们有下面的功能。当要登录到网络上某台机器的时候，登录屏幕给出一个大约七位数的随机质询。你把这个数键入到密码发生器中，同时键入大约四位数的 PIN。密码发生器就会用和公司安全服务器共享的密钥对这十一位数加密，然后显示出结果的前七位数。你把这七位数作为密码输入（见图 2-1）。如果你有含恰当密钥的密码发生器、PIN 输入正确、密码结果输入无误的话，公司的计算机系统就会让你进入。但是如果你没有和已知 PIN 相对应的密码发生器，就不太可能成功登录。

我们用 S 代表服务器，P 代表密码发生器，PIN 代表用户启动密码发生器的身份识别码，U 代表用户，N 代表随机 nonce：

$$S \rightarrow U: N$$

$$U \rightarrow P: N, PIN$$

$$P \rightarrow U: \{N, PIN\}_K$$

$$U \rightarrow S: \{N, PIN\}_K$$

（关于主要的质询/响应产品的详细描述，参见 [15] 第 211 页。）

质询/响应协议里的加密算法不需要可逆，这样的话，可以使用一个“单向函数”或者“哈希加密函数”来实现，它不像加密算法那样容易受到出口限制（其技术特性参见第 5 章“密码学”）。

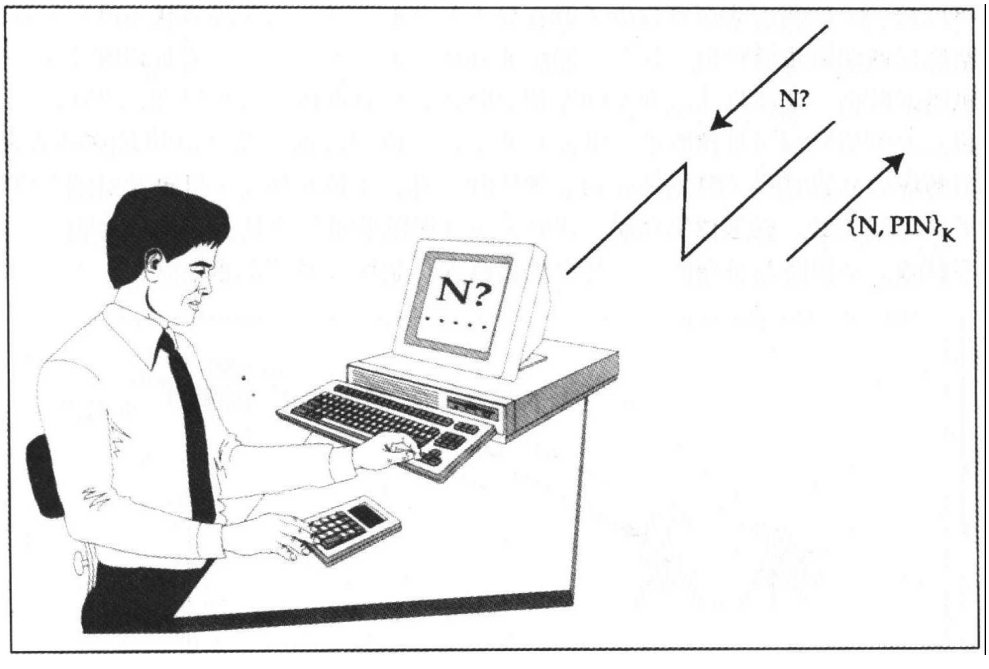


图 2-1 使用密码发生器

2.2.2 米格战斗机中间人攻击

有一次针对质询/响应系统有名的攻击，它在南非的和平进程中起了重要作用。

在 20 世纪 30 和 40 年代，战斗机的飞行速度一直在加快，同时还发明了喷气式发动机、雷达和火箭技术，这使得防空部门很难区别出敌方和己方的飞机。于是出现了误伤的严重情况，这驱使人们去研制敌我识别（IFF）系统。这些早期的系统最先在二战中派上用场，被雷达侦测到的飞机播送一个确认号码，表明自己是己方的。1952 年这些系统应用到民用航空上，让空中交通管制员识别飞机，同时由于担心技术推广以后会丧失安全性，美国空军开始研究怎样在系统中插入加密保护。现在，一般的空军防御系统用雷达信号发出随机质询，友方的飞机拥有设备和密钥，能正确响应并确认自己的身份。

美国空军使用的 IFF 系统叫做 Mode XII，并且正在研制以用于地面部队。但是南非空军（SAAF）受到制裁，不能得到西方军事援助，不得不设计自己的系统。

在 20 世纪 80 年代后期，南非军队在纳米比亚北部和安哥拉南部展开了战争。战争目的是要维护纳米比亚的白人统治，并给安哥拉强加一个傀儡政府（UNITA）。因为南非防御部队的大部分士兵都是从少量的白人人口中征募来的，所以减少伤亡是至关重要的。于是大部分南非部队都留在纳米比亚维护治安，而与北方作战的任务就交给了 UNITA 部队。南非空军的任务是双重的：轰炸安哥拉的目标给 UNITA 提供战术支持，还要确保安哥拉及其盟国古巴不会回来攻击纳米比亚。

然而，古巴空军突然冲破了南非的空防，并且轰炸了位于纳米比亚北部的一个南非军营，炸死了很多白人士兵。这证明南非已经丧失了空军优势，比勒陀利亚政府由此决定退出纳米比亚。这件事成为南非由少数白人统治向民主政治过渡进程中的重大历史事件。

几年以后，一位前南非军官告诉了我古巴人获胜的经过。几架米格战斗机在安哥拉南部的南非空防区的北部边沿徘徊，这时一架南非黑羚羊轰炸机正在轰炸安哥拉的目标。随后米格战斗机迅速转向，公然地飞越南非空军的空防区，空防系统于是发出 IFF 的质询信号。米格战斗机把质询信号向安哥拉防空火炮阵地重播，阵地再向南非轰炸机传送这些质询信号，轰炸机的响应信号就这样实时地传送回了米格战斗机，于是米格战斗机再发射这个信号就可以通行了（见图 2-2）。据这位军官说，这件事对比勒陀利亚的军队参谋部触动很大。不仅被黑人对手打败，而且还是被智取了，这与他们白人优越的世界观大相径庭。

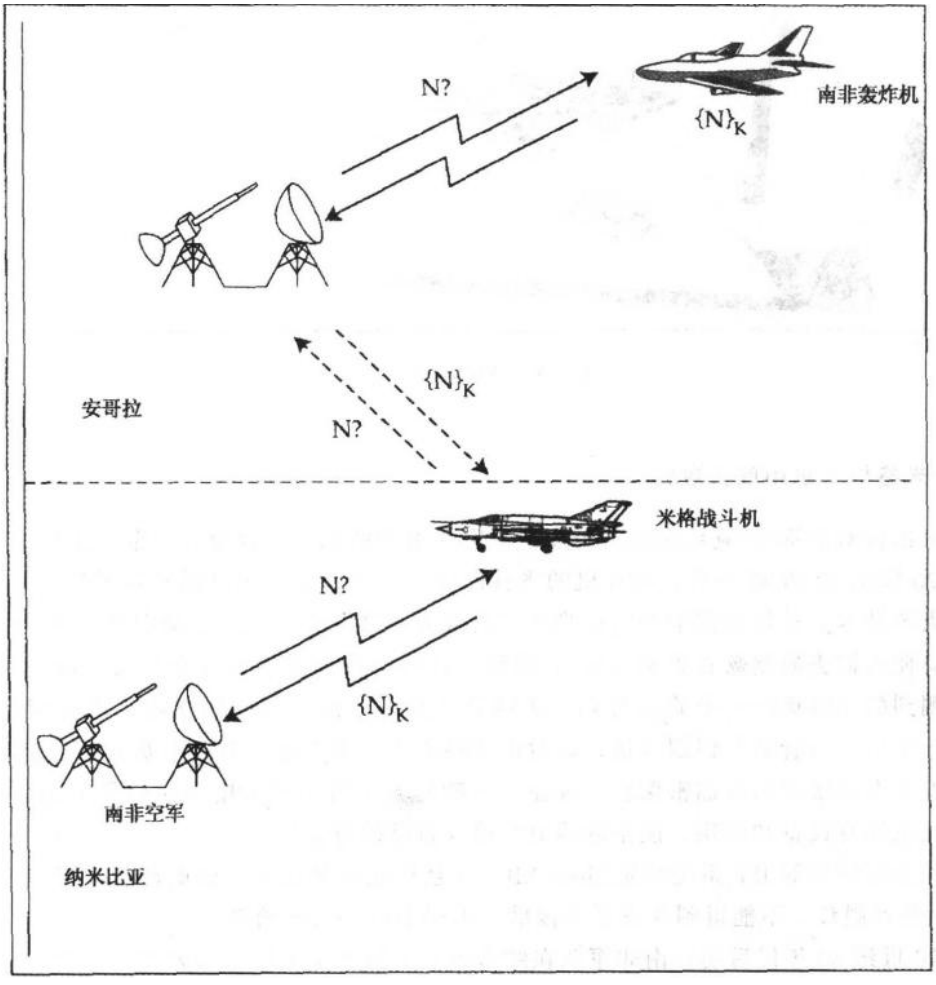


图 2-2　米格战斗机中间人攻击

我没有单独向安哥拉或者古巴方面证实这段历史。但是所提到的基本技术至少和二战一样古老，它阐明了加密技术里的中间人攻击的基本思想。这种攻击在从付费电视到因特网安全协议的各种应用中会反复地用到。中间人攻击甚至还应用到在线游戏上。像数学家 John Conway 曾经说的那样，用邮件下棋击败一个大师是很容易的：只要同时和两位大师比赛，一个执白棋，另外一个执黑棋，实际上让他们之间下棋。

在很多情况下，可以使用中间人攻击但是并不经济。比如偷车的情况，派一位同谋者跟

着司机，把无线质询信号转播到你那里帮你开锁，当然有可能把车偷走。但是偷或抢司机的钥匙会更简单。

2.2.3 反射攻击

在交互认证中，当两个当事人都必须识别对方时，出现了很多有趣的问题。设想一下，为了防止防空炮手攻击己方飞机而设计出的简单质询/响应 IFF 系统，也必须在轰炸机上安装一个。现在假设空军在每架飞机上安装了一个炮手质询设备，并把它连接到指挥射击的雷达上。敌方轰炸机可能把质询信号反射给我方战斗机，获得了正确的响应后，再把它作为自己的响应反射回来：

$$F \rightarrow B: N$$

$$B \rightarrow F: N$$

$$F \rightarrow B: \{N\}_K$$

$$B \rightarrow F: \{N\}_K$$

所以要把质询系统和响应器结合起来。把这两个装置连在一起并共用一列质询，这样还是不够，因为敌人如果被两架飞机攻击，会把其中一架飞机向另外一架飞机发出的质询反射回来。同样，在空战中，人们也不愿意人为地从“进攻”转换为“防守”。

有很多方法可以阻止反射攻击。在很多情况下，进行认证交换时只要包含两方的名字就足够了。在前面的例子中，对于质询：

$$F \rightarrow B: N$$

己方的轰炸机回送这样的响应：

$$B \rightarrow F: \{B, N\}_K$$

这样一来，可以检测到反射的响应 $\{F, N\}$（或者还有战斗机驾驶员发出的 $\{F', N\}$）。

这是个对 IFF 的简化说明，但是仍然可以用来阐明认证协议中的各种信任假设。如果你发出了一个质询信号 N，并在 20 毫秒内收到了响应 $\{N\}_K$，那么，由于光在 20 毫秒内可以传播 3730 英里，你就可以知道在 2000 英里内有人拥有密钥 K。但是你所知道的也就只有这些。如果你确信这个响应不是用自己的设备计算出来的，那你就知道在 2000 英里内还有其他人拥有密钥 K。如果你进一步假设，密钥 K 的所有副本都掌握在足以信任的设备上，并且你看到了 $\{B, N\}_K$，你就可以推断出带有呼号 B 的战斗机位于 2000 英里以内。对信任假设及其后果有一个清楚的理解是设计安全协议的核心。

到现在为止，你可能认为本书已经全面地讲解了 IFF 协议设计的各个方面。但是我忽略了一个最重要的问题——也是早期设计 IFF 系统时没有预料到的一个问题。由于雷达的返回信号比较弱，战斗机上的 IFF 发射器发射的信号往往比返回信号传送得更远。盟军是在二战后期认识到这一点的。1944 年 1 月，盟军破解了 Enigma 密码后得知，德军一直采用在两倍正常雷达测距外向美英轰炸机发送质询信号的办法，绘制盟军飞机的进攻线路图。所以很多现代系统都不仅识别响应信号，也要识别质询信号。例如，北约模式 XII 中，采用 32 位的加密质询，对每一次提问都产生一个不同的有效质询信号，一般是每秒钟 250 次。理论上，在敌方领域内并不需要切断信号，但是实际上，敌方记录下有效的质询信号并重播它们，可

以作为一次攻击。

IFF 还有很多方面与协议没有太大的关系，比如中立引起的困难，密集运转环境中的错误率，怎样处理设备失败，怎样管理密钥和怎样应付多国联合，这些情况在海湾战争的“沙漠风暴”行动中都出现了。在第 16 章还会讲 IFF。至于现在，质询欺骗问题强调了一个关键点：安全协议的正确性依赖于对需求的假设。协议如果阻止了一种攻击（被己方击毙），但是又暴露在另外的更有危害的攻击之下（被敌方击毙），那么这样做就弊大于利。事实上，在二战中质询欺骗问题实在是太严重了，因此很多专家提倡宁可废弃掉整个 IFF，也不愿冒险让几百架轰炸机的编队中可能有一个飞行员忽视命令而打开了 IFF。

2.3 伪造消息

有一种中间人攻击常常被划分成单独的一类攻击。这就是攻击者不仅反射认证信息，还以某种手段修改消息内容。在本章开始有一个例子：伪造离线使用的 ATM 卡窃取钱财。实际上，在电话线（或者主机）关掉时，磁卡是担任银行主机和提款机之间储存和转送数据的通信渠道。

再举一个例子，不法的出租车司机在计价器和变速箱传感器的连线上接入一个脉冲发生器。当推进杆转动时，传感器就发出脉冲信号，让计价器计算出租车所走的路程。伪造设备插入了多余的脉冲，使出租车看起来走了更远的路。本书将在第 10 章“监控系统”的 10.4 节以更多的篇幅讨论这种攻击。

然而，前面也看到了，伪造应用层消息的攻击和重播攻击是有区别的。这些攻击不只局限于低级系统，像门锁那样，记录和重发一个固定的密码就被击溃了。用于国际电话和数据通信的国际通信卫星具有稳健的机制，能够防止一个指令被接收两次——否则，攻击者重复同样的操作会使卫星耗尽燃料 [617]。

另外一个例子是密钥日志攻击，它击败了很多付费电视系统（它还有一个别名叫延迟数据传输或者 DDT）。一般的付费电视设备有一个对视频信号解密的解码器和一个产生解码密钥的客户智能卡。这些密钥每秒钟用单向加密函数重复计算好几次，该加密函数应用在信号中出现的各种“权限控制消息”中。这种系统经过了精细的设计（以后本书会讨论一些更复杂的攻击），但是有一种非常简单的攻击总能奏效。如果在智能卡和各个解码器之间传送的消息都是一样的（通常情况下如此），那么订户可以记录下他们的卡向解码器发出的全部密钥的日志，再把它们邮递到网络上。有的人没交订金但是有加密的视频节目录像，他就可以从网络上下载密钥日志，再用它来解译磁带。

为了防止 DDT 攻击而更换付费电视的协议很困难。装好的设备基础庞大，很多明显的对策又对合法用户有负面影响（比如禁止对节目录像）。付费电视公司一般都忽视这种攻击，因为只有沉迷于电视的人才会用专门的硬件适配器把 PC 机连上卫星电视解码器；而真正想威胁到这种收入实在是太困难了。

2.4 环境变更

导致协议失败的一个普遍原因是环境变更，这样原来的假设不再成立，安全协议就无法应付新的威胁。

伦敦交通部门使用的票务系统是一个很好的例子。在 20 世纪 80 年代早期，乘客发明

了各种诡计来少买车费。例如，一个乘客长途往返于郊区和城区之间，他可能会买两张更便宜的短程季度票——一张是从郊区到附近的车站，另一张是从目的地到其他城区车站。这使得他们能够通过关卡；偶然在途中遇到查票员时，他们谎称是郊区车站的售票机坏了。

作了一笔大额投资改造以后，系统具有了防止这些诡计的全面特性：所有的关卡都自动化了，车票可以保留状态，修改法律规定对无票的乘客施行现场罚款。

但是之后大环境也变化了，私有化使交通部门被分化成几十家私营的铁路和公路公司。有些新的运营公司开始互相欺骗，而系统对此毫无办法！例如，每卖掉一张当日通票，收入由各个汽车、火车和地铁运营商分配，而计算的规则是依靠售票的地点。突然间，铁路公司想要通过预售火车票来保持最大的营业收入。混乱和诉讼随之而来。

运输系统的问题早就出现了。20世纪70年代中期人们在意大利的法沙谷滑雪场已经观察到了这个问题。在那里，游人买一张月票，可以登上山谷中运送滑雪者上山的所有缆车。有人看到某个索道的服务员拿着一摞卡，每过一个游客就用一张卡过一下读卡机。原来各个索道部门的收入是依靠运送客人的数量来分配的。所以每个部门都尽可能地增加自己的数字[730]。

在提款机诈骗领域还有一个相关的例子。在1993和1994年，荷兰遭受了梦幻提款风潮，当时报刊上有很多争论，一方面银行声称其系统是安全的，而另一方面很多人向报纸写信声称受害。最后，银行在谴责声中积极调查了那些事件，注意到很多受害者都在乌得勒支附近的某个汽车加油站使用过银行卡。加油站被置于警察监视下，随后一名员工被逮捕。原来他从读卡机和控制它的PC机之间接了个分线；他的分线记录了客户卡磁条上的详细内容，而他则偷看到客户的PIN [19]。

为什么这个系统设计得这么糟糕？原来，在20世纪80年代早期，当IBM和VISA这样的组织制定管理磁条卡和PIN的标准时，工程师们已经作了两种假设。第一种假设就是磁条的内容不保密（包括卡号、版本号和有效期限），而PIN是保密的[548]。（使用的类比就是，磁条是持卡人的名字，而PIN是密码。本书后面还会多讲一些关于命名的微妙之处。）第二种假设就是银行卡设备只会用在可信任的环境里，比如性能稳定的自动柜员机或者银行出纳员。这样，显然只需要在PIN输入到服务器途中对PIN加密，而磁条上的数据可以从磁卡机公然传送。

1993年这两种假设都改变了。有一段时间盛行制造假卡，主要集中在20世纪80年代后期远东地区，这促使银行在磁条上加入认证码。而银行卡产业取得了商业成功，致使各国银行把借贷卡从ATM机扩展应用到各种商店的终端上。这两种环境变换联合在一起，破坏了最初的系统设计：人们不是把含有公开数据的磁条卡插入一个信任的机器上，而是把需要依赖安全数据的卡插入一个不能信任的机器上。这些变换逐渐发生，在这么长的时间里，业界没有看出即将要发生的问题。

2.5 选择协议攻击

有些人在努力宣传“多功能智能卡”的想法，这种认证设备可以在大范围的交易中使用，用户不再需要随身携带几十张卡和密钥了。

这又引入了许多有意思的新风险。假设你用卡签署银行交易，平常的办法就是用卡对交易数据计算出一个数字签名。实际上，为了节省计算量，签名是用交易中看起来像是随机的

20字节摘要计算的（本书将在第5章讲怎样计算摘要）。现在假设这张卡还可以用在其他应用场合。黑手党会怎样设计协议来攻击它呢？

这里有一个例子。目前访问色情站点的人经常被问及“年龄证明”，一般要向该站点或者年龄核对服务提交一个信用卡号码。如果信用卡具有数字签名的功能，那么色情站点理所当然地会要求顾客在一个随机质询上签名作为年龄证明。于是色情站点就可以上演黑手党中间人攻击，如图2-3所示。犯罪者会等待轻信的顾客访问站点，然后从经销商那里订一些可以转售的商品（例如金币），扮演硬币经销商的顾客。当金币经销商向他们发送用来签名的交易摘要时，他们把摘要以随机质询的形式转发给等待的顾客。顾客签名了，黑手党获得了金币，当数以千计的人们在月底忽然抱怨卡上的巨大开销时，色情站点已经销声匿迹了——还有金币也一起消失了［446］。

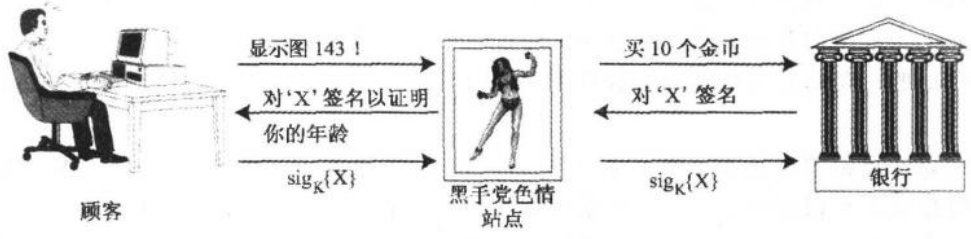


图2-3　黑手党中间人攻击

这是乌得勒支犯罪的一个极端的派生方法。由此得出几点教训：在多个应用场合使用密钥或者其他认证机制是危险的；让其他人破解自己的应用安全则是彻底的傻瓜。

2.6　管理密钥

本书目前讨论的安全协议的例子，大部分是关于认证一个当事人的名字或者应用数据，例如驱动出租车计价器的脉冲。还有一种重要的更深层的认证协议：用来管理密钥的协议。到目前为止，这种协议在支持运转的后台中广泛应用。为了管理提款机和银行互相通信用的密钥研制了很多技术。但是现在，像付费电视这样的系统还是使用密钥管理来控制对系统的直接访问。

认证协议还可以用在分布式计算机系统上，进行一般的密钥管理，由此变得非常重要。Kerberos就是得到广泛应用的第一种这样的系统，它的一个派生系统用在Windows 2000中。现在要为理解Kerberos奠定基础。

2.6.1　基本密钥管理

密钥分配协议的基本思想就是，在两个当事人要进行通信时，他们会找一个信任的第三方作介绍。

本书谈论过，在认证协议的文化里，传统上是给当事人起个人性化的名字，以避免陷入太多的代数符号。因此本书把两个通信当事人称为Alice和Bob，信任的第三方为Sam。但是请不要假定本书正在谈论人类的当事人。Alice和Bob很可能是程序，而Sam可能是服务器；Alice可能是出租车计价器里的程序，Bob是变速箱传感器的程序，而Sam是出租车监察站的计算机。

总之，简单的认证协议运行如下：

1）Alice 首先呼叫 Sam，申请与 Bob 进行通信的密钥。

2）Sam 回应，向 Alice 发送一对证书。每个证书都包括一个密钥的副本，第一个加密证书只有 Alice 能读懂，第二个加密证书只有 Bob 能读懂。

3）然后 Alice 呼叫 Bob，提交第二个证书作为自己的介绍。两者都用和 Sam 共享的密钥对相应的证书解密，然后才可以得到新的密钥。Alice 现在可以用这个密钥向 Bob 发送密文，并且接收 Bob 返回的消息。

本书提到过，重播攻击是认证协议面临的一个典型的问题，因此为了让 Alice 和 Bob 证实他们的证书是新鲜的，Sam 会在每个证书上打上一个时间戳。如果证书永远也不作废的话，处理那些权利已经失效的用户时会产生严重的问题。

使用协议符号，可以把这个过程描述如下：

$$A \rightarrow S: A, B$$

$$S \rightarrow A: \{A, B, K_{AB}, T\}_{K_{AS}}, \{A, B, K_{AB}, T\}_{K_{BS}}$$

$$A \rightarrow B: \{A, B, K_{AB}, T\}_{K_{BS}}, \{M\}_{K_{AB}}$$

把符号扩展一下，Alice 呼叫 Sam，说她想和 Bob 通话。Sam 构造一个会话密钥消息，包括 Alice 的名字、Bob 的名字、他们使用的密钥和一个时间戳。Sam 用和 Alice 共享的密钥对它加密，再用和 Bob 共享的密钥加密。他把两个密文都发给 Alice。Alice 从加密的密文中重新获得密钥，并把为 Bob 加密的密文传送给他。现在 Alice 可以用这个密钥给 Bob 发任何消息了。

2.6.2 Needham-Schroeder 协议

很多事情都会出错，本书将在后面讲大量的例子。现在，将一个历史上著名的例子介绍给读者。很多现有的密钥分配协议是从 1978 年 Roger Needham 和 Mike Schroeder 发明的协议继承下来的 [589]。这个协议和刚才提到的很相似，但是采用 nonce 而不是时间戳。它的过程如下：

消息 1 $A \rightarrow S: A, B, N_A$

消息 2 $S \rightarrow A: \{N_A, B, K_{AB}, \{K_{AB}, A\}_{K_{BS}}\}_{K_{AS}}$

消息 3 $A \rightarrow B: \{K_{AB}, A\}_{K_{BS}}$

消息 4 $B \rightarrow A: \{N_B\}_{K_{AB}}$

消息 5 $A \rightarrow B: \{N_B - 1\}_{K_{AB}}$

这里，Alice 发起请求，告诉 Sam：“我是 Alice；我想和 Bob 通话，我的随机 nonce 是 N_A。”Sam 给她提供一个会话密钥，用他和 Alice 共享的密钥进行加密。密文中也包括 Alice 的 nonce，因此她可以断定这不是一次重发。Sam 还给她一个证书让她把密钥转给 Bob。Alice 把证书发给 Bob，然后 Bob 做一次质询/响应来检验她在场而且是警觉的。

这个协议有一个微小的问题：Bob 必须假定他从 Sam（经由 Alice）那里收到的 K_{AB} 是新鲜的。这样做是不必要的：Alice 在第 2 和第 3 步之间可能等待了一年。在很多应用程序里，这个问题并不重要，甚至还可以帮助 Alice 在服务器失败的情况下保留密钥。但是如果一个对

手，如 Charlie，掌握了 Alice 的密钥 K_{AS}，他可以使用这个密钥和其他很多当事人建立会话。

举个例子，假设 Alice 也申请并收到了和 Dorothy 通信的密钥，Charlie 偷了她的密钥以后，假冒 Alice 向 Sam 发送消息，并获得了和 Freddie 和 Ginger 通信的密钥。他还有可能观察到 Alice 和 Dorothy 通过协议交换的消息 2，也就是说，当 Sam 发给 Alice 一个与 Dorothy 通话的密钥时，是在 K_{AS} 下加密的，而现在这个密钥已经有危险了。因此现在 Charlie 可以向 Dorothy、Freddie 和 Ginger 扮演 Alice。当 Alice 发觉她的密钥被偷了，或许是通过和 Dorothy 比较消息记录发现的，她就必须让 Sam 和每个她曾发放过密钥的人取得联系，并通知他们原来的密钥失效了。她本人无法做到这一点，因为她对 Freddie 和 Ginger 一无所知。换句话说，撤销权利是一个问题：Sam 可能必须保留做过事情的全部记录，这些记录永远都在增长，除非当事人的名字在将来某个固定的时间作废。

20 多年以后，这个例子还在安全协议学界引发争议。片面的观点认为 Needham 和 Schroeder 就是做错了；Susan Pancho 和 Dieter Gollmann 的观点（我比较同情这个观点）则认为，这又是一个改变假设而导致协议失败的例子 [345，600]。1978 年计算机还是一个友善、文雅的世界，此后计算机安全关注把坏人挡在系统之外，但是现在却必须把坏人假想为系统的用户。Needham-Schroeder 的论文显然假定所有的当事人就扮演自己的角色，攻击者都来自于外界 [589]。在这种假设下，协议是合理的。

2.6.3 Kerberos

从 Needham-Schroeder 协议派生出来的一个重要的协议用在了 Kerberos 中，这是一个分布式访问控制系统，起源于 MIT，现在是 Windows 2000 中默认的认证选项 [735]。Kerberos 不只是一个信任的第三方，而是有两种角色：一种是让用户登录的认证服务器，还有一种是给用户发放票证来允许他们访问各种资源（比如文件）的服务器。这样就能允许分级访问管理。举个例子，在大学里通过宿舍楼管理学生，而通过院系来管理文件服务器；在公司里，人事部门把用户注册到薪资系统，而部门管理员负责管理服务器和打印机之类的资源。

首先，Alice 使用密码登录到认证服务器上。她 PC 机上的客户端软件从服务器取回一张票，它是在这个密码下加密的并且包含一个会话密钥 K_{AS}。假设她的密码是正确的，现在她可以控制 K_{AS}；为了访问发行票证的服务器 S 控制的资源 B，要执行以下的协议。协议产生一个密钥 K_{AB}，它带有时间戳 T_S 和生存期 L，用来认证 Alice 随后和资源 B 间的通信：

$$A \rightarrow S:\ A,\ B$$

$$S \rightarrow A:\ \{T_S,\ L,\ K_{AB},\ B,\ \{T_S,\ L,\ K_{AB},\ A\}_{K_{BS}}\}_{K_{AS}}$$

$$A \rightarrow B:\ \{T_S,\ L,\ K_{AB},\ A\}_{K_{BS}},\ \{A,\ T_A\}_{K_{AB}}$$

$$B \rightarrow A:\ \{T_A + 1\}_{K_{AB}}$$

翻译出来就是：Alice 向发行票证的服务器要求访问 B。如果这个要求是允许的，那么服务器创建一个含有合适密钥 K_{AB} 的票 $\{T_S,\ L,\ K_{AB},\ A\}_{K_{BS}}$，发放给 Alice 使用。Alice 还获得一个她能读懂的该密钥的副本，也就是用 K_{AS} 加密过的。现在她向资源发送一个时间戳 T_A 来核实这把密钥，通过给发送回的时间戳加 1 确认它的存在性（这是一个惯例，表明所要访问的资源能够正确解密票证，并且提取出密钥 K_{AB}）。

通过引入时间戳而不是随机的 nonce，Needham-Schroeder 的脆弱性已经得到了巩固。但是，正如生活中的情形一样，没有免费的安全。现在又出现了新的脆弱性，就是各种客户端和服务器的时钟可能不同步，甚至被更复杂的攻击故意弄得不同步。

2.7 走向形式化

在本书讨论到的协议中遇见一些小困难，还有很多方法的保护效果依赖于非常狭窄（通常也不明显）的初始假设，这使得研究人员要在密钥分配协议中采用形式化方法。起初这个实践的目的是为了判断一个协议的对错。要么使用正确的协议，要么禁止攻击。最近，已经扩展到阐明协议的各种基本假设。

有多种方法可以说明协议的正确性。最著名的方法是一种信任逻辑，叫做 *BAN 逻辑*，它是以发明者 Mike Burrows、Martín Abadi 和 Roger Needham 的名字命名的 [148]。它在已知某些消息和时间戳等条件时，推理出当事人应该信任什么。第二种方法是*随机预言模型*，将在第 5 章提到，有很多数学家热衷于钻研这门学科的终极理论。这种方法看起来没有信任逻辑那样富有表现力，但是可以让协议的性能依赖于所采用的加密算法的性能。最后，很多研究人员采用了主流形式化方法，例如 CSP 和 Lotos。

有历史记录表明，有些用形式化方法证明无误的协议中存在缺陷。下面的小节提供了一个典型的例子。

2.7.1 一个典型的银行智能卡协议

这个系统现在被称为 COPAC，是 VISA 在通信条件差的国家中使用的一种电子钱包系统 [35]。它是最早投入使用的、使用形式化技术来设计和验证基本协议族的金融系统，并且是 BAN 逻辑的变体。与其类似的一个协议现在还应用在“黄金银行”里，这是德国银行向客户发行的一种电子钱包。

交易是在顾客的智能卡和商店的智能卡之间进行的。顾客给商店一张电子支票，内含两个认证码，一个被网络核对，另一个被顾客的银行核对。这个协议可以简要描述如下：

$$C \rightarrow R: \{C, N_C\}_K$$
$$R \rightarrow C: \{R, N_R, C, N_C\}_K$$
$$C \rightarrow R: \{C, N_C, R, N_R, X\}_K$$

翻译过来就是：顾客和零售商人共享一个密钥 K。顾客卡使用这个密钥对含有其账号 C 和交易序列号 N_C 的消息加密。零售商确认自己的名字 R、交易序列号 N_R 和刚刚从顾客那里收到的信息。现在顾客发出电子支票 X 和至今为止协议交换的全部数据。可以把电子支票归类为一种需要顾客和零售商的名字和账号的付费设备（在每条消息中都重复前面的所有数据，这样做的目的是防止坏人用剪切/粘贴的方法伪造消息）。

2.7.2 BAN 逻辑

BAN 逻辑提供了一种形式化方法，来推理加密协议中当事人的信任所在。其基本思想就是，如果一条消息是用相应的密钥加密的，并且是新鲜的（就是在协议的当前运行期间产生的），它就是可信的。还有进一步的假设，包括当事人只坚持他们信任的陈述，并且有一些当事人是某些陈述的权威者。这可以用如下的符号形象化描述：

- $A|\equiv X$：A 相信 X，或者更精确地说，A 有资格相信 X。
- $A|\sim X$：A 曾经说过 X（没有暗示是否在近期说过）。
- $A|\Rightarrow X$：A 具有 X 的权威；换句话说，A 是 X 的权威者，可以信任。
- $A\triangleleft X$：A 看见 X；就是说，有人向 A 发送了一条含有 X 的消息，A 可以读这条消息并重复它。
- $\#X$：X 是新鲜的；就是说，包含一个当前的时间戳，或者有信息表明该消息是在协议的当前运行期间由相应的当事人发出的。
- $\{X\}_K$：用密钥 K 加密以后的 X，本章其余部分与此相同。
- $A\leftrightarrow^K B$：A 和 B 共享密钥 K；换句话说，就是他们用来通信的密钥。

其他的符号，例如关于公共密钥操作和密码的符号，在此不涉及。

使用这些符号时有一些基本规则，包括：

消息含义规则。规定，如果 A 看见了一条用 K 加密的消息，并且 K 是 A 与 B 通信的良好密钥，那么，A 就相信这条消息是 B 曾经说过的（本书假设每个当事人都能够认出或者忽略自己的消息）。记为：

$$\frac{A|\equiv A\leftrightarrow^K B, A\triangleleft\{X\}_K}{A|\equiv B|\sim X}$$

Nonce 认证规则。规定，如果一位当事人曾经说过一条消息，并且这条消息是新鲜的，那么这位当事人仍然相信它。记为：

$$\frac{A|\equiv \#X, A|\equiv B|\sim X}{A|\equiv B|\equiv X}$$

裁定规则。规定，如果一位当事人相信某件事情，并且是这件事情的权威，那么，这位当事人应该受到信任。记为：

$$\frac{A|\equiv B|\Rightarrow X, A|\equiv B|\equiv X}{A|\equiv X}$$

在这些表示法中，上面的语句是条件；下面的是结果。很多进一步的规则涉及到消息处理中的更加机械的方面。例如，倘若一位当事人有合适的密钥，他（她）看到一条语句时就看到了其中的各个部分；如果公式中有一部分是新鲜的，那么整个公式就肯定是新鲜的。

2.7.3 认证付费协议

假设只有那些能够如实遵守协议的值得信赖的当事人拥有密钥 K，那么形式化认证就很直截了当。办法就是从想得到的结果出发并倒推。本例中想要证明零售商信任这张支票；也就是证明 $R|\equiv X$（从这个角度可以认为支票和密钥的语法是类似的，即当且仅当支票是真实并且新鲜时才有效）。

现在要用裁定规则推导出 $R|\equiv X$，需要的条件为 $R|\equiv C|\Rightarrow X$（R 相信 C 有裁定 X 的资格）和 $R|\equiv C|\equiv X$（R 认为 C 相信 X）。

第一个条件符合硬件约束，即只有 C 才能发出诸如 $\{C,\cdots\}_K$ 的文本。

第二个条件 $R|\equiv C|\equiv X$ 必须用 nonce 认证规则推导，需要的条件为 $\#X$（X 是新鲜的）

和 $R|\equiv C|\sim X$（R 相信 C 发出的 X）。

#X 是由出现在包括序列号 N_R 的 $\{C, N_C, R, N_R, X\}_K$ 中的 X 推导出来的，而 $R|\equiv C|\sim X$ 是从硬件约束中得到的。

证明的原理很简洁。如果想要了解认证的逻辑细节，应该查找原文，并参阅本章末尾推荐的更多读物。

2.7.4 形式化认证的局限性

由于形式化方法迫使设计者把每样事情都表示得很清楚，这样就必须面临更费力的设计，否则有可能粗制滥造，因而在安全协议设计期间寻找漏洞时，它是一种非常好的方法。但是它们也有局限性。

第一个问题是所作的外部假设。例如，假设没有经过授权使用钥匙的人是拿不到钥匙的。实际上，这个假设并不是总成立。尽管 COPAC 钱包协议是在防篡改的智能卡上执行的，但是其软件会存在漏洞；并且任何情况下所提供的防篡改性都不可能是完整的（本书第 14 章“物理防篡改”解释这个问题）。因此系统作了很多让步机制来检测和抵制假卡，例如“影子账号”，它跟踪每张卡上应该有的余额并在每次交易完成后刷新。系统还列出了热卡黑名单发送到各个终端上，可以防备被偷的卡，也可以用来对付假卡。

第二，协议的理想化经常存在很多问题。把 BAN 逻辑用在使用公钥加密的协议上有一个著名的例子。消息含义规则里有一个方案只能用于数字签名，却被人错误地用在了解码中，这无疑导致了在一个有缺陷的协议下进行认证。还有一个例子是在早期版本的 COPAC 系统中发现的漏洞。系统中的密钥 K 实际上由两个密钥组成：首先用一个“交易密钥”加密，这个密钥是多变的（就是说，每张卡都有自己的变化），然后再用一个“银行密钥”加密，这个密钥是不变的。第一次加密是通过网络操作的，第二次是在发行该卡的银行完成的。这样做的理由是为了实现双重控制，以保证如果攻击者成功地从一张卡上找出了密钥，他只能伪造那一张卡，而无法伪造其他的卡（这样就可以击溃热卡机制）。但是由于银行的密钥是不变的，任何攻击者只要解开了一张卡就知道了这个密钥。这意味着攻击者可以解开加密的外部包装；在某些环境下，有可能进行消息重发（在有人发现和利用这个漏洞之前，后来的版本中银行密钥就是多变的了）。

在这个例子中形式化认证方法并没有出现失误，因为没有试图对多样机制进行确认。但是它确实阐述了安全工程中的一个普遍问题——脆弱性出现在两种保护技术的衔接处。本例中有三种技术：硬件抗干扰性、认证协议和影子账号/热卡黑名单机制。不同的保护技术通常用于不同专家的领域里，而他们并不完全了解别人所作的假设（这也是安全工程师需要本书的原因之一：帮助各学科专家去了解彼此的工具并进行更有效的沟通）。

出于以上原因，人们已经研究了其他方法来确保认证协议的设计，其中包括协议鲁棒性思想。正如结构化编程技术的目标是确保系统化地设计软件并且不疏漏任何要点，鲁棒性协议设计的目标主要是为了清楚直接。鲁棒性原则规定一个协议的阐述应该依据它的内容，而不是上下文；每个要点（比如当事人的名字）都应该在消息中清楚地指出来。还有一些问题涉及到序列号、时间戳和随机质询提供的新鲜性以及加密的方式。如果协议中采取了公钥加密或者数字签名机制，还有更多的技术鲁棒性问题。

2.8 小结

密码并不总是一种足够安全的保护措施，尤其是当它们必须在公开的通信渠道中多次使用的时候。简单的认证协议，不论是单工（例如使用随机 nonce）还是双工（质询/响应），在很多情况下比较适用，已经在各种系统中派上用场，从遥控汽车门锁到军事 IFF 系统，还有分布式计算机系统中的认证。

设计一个有效的安全协议是很困难的。它们要解决很多潜在的问题，其中包括中间人攻击、修改攻击、反射攻击和重发攻击。这些威胁可能会与实现的脆弱性相互影响，比如不良的随机数发生器。使用数学技术来鉴定协议的正确性可能有帮助，但是并不能抓住所有的漏洞。一些致命的失败是由于设计协议的环境悄然改变，以至于它的保护不再奏效。

研究问题

过去几年里，有人觉得协议已经“妥”了，大家应该转向新的研究课题。其实新的协议应用中带有一大堆新的错误和有待发现的攻击漏洞，它们的出现已经反复证明这些人的观点是错误的。在 20 世纪 90 年代早期，密钥管理协议是研究的焦点；90 年代中期，人们忙于提出电子商务中的各种建议；到了 90 年代末期，关于在因特网上保护版权提出了整个一系列的机制，使人们有了新的研究目标。

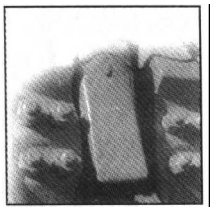
是继续开发出不完善的协议让别人来攻击，还是设法首先找到一个设计正确协议的方法论？形式化方法（还有其他数学方法，比如随机预言模型）的真正用途和局限是什么？

对于一个系统来说，鲁棒性协议通常规定每件事情都要说明并核查（当事人的名字、角色、安全策略陈述、协议版本、时间、日期、序列号、安全内容，甚至祖母家厨房洗菜盆的制造者），而系统工程规定一个好的规范不应该过多地约束实现者，怎样缓解这两者之间的关系？

参考资料

安全协议方面的研究论文在学术界随处可见。介绍性的论文可能主要有 Needham-Schroeder 最初的论文［589］；Burrows-Abadi-Needham 认证逻辑［148］；Martín Abadi 与 Roger Needham，还有 Roger Needham 与我写的关于协议鲁棒性的论文［2，47］。还有 Roger 和我写的一篇调查报告，提出“给撒旦的计算机编程”（Bruce Schneier 在序中提到）作为安全协议设计的比喻［48］。在［449］中，分析了一个用三种不同的形式化方法实现的有缺陷的安全协议。除此之外，安全协议研讨会的论文集［183，184］给现在的研究提供了向导；在各种会议上也出现了很多论文。

第3章 口 令



人类没有能力安全地保管高质量的密钥，它们进行加密操作的速度和准确度也让人难以接受（这些密钥还非常巨大、维护起来昂贵、难于管理并且污染环境。继续制造和开发这样的设备是令人惊奇的事情。但是这些早就无所不在，我们必须围绕它们的局限性来设计自己的协议）。

——Kaufman、Perlman 和 Speciner［444］

保管老式的访问控制卡，例如金属钥匙，是一种常识性的问题。但是对于保护计算机系统的设备，常识往往不够用。人机隔阂在很多上下文中导致了安全问题，从直观的系统管理到用户不正确地管理安全产品（加密软件）的方法［803］（本书不使用流行的说法“人机界面”：“隔阂”可能更合适些）。然而，大多数问题都出在相对易于分析和讨论的简单的上下文中——口令管理。

除了那些“显然”的口令，比如用来登录计算机的口令和激活银行卡的 PIN，还有很多其他的东西（或者某些东西的组合）具有口令的效用。最人尽皆知的是社会保险号和母亲婚前的姓名，很多机构用这些来识别。举个例子，AT&T 的无线服务合同规定，一个人如果知道了你的名字、住址、电话号码和社会保险号的后四位数字，他就有权改动你的账号；合同还声明对缺乏隐私概不负责［201］。

猜测或者从或多或少的公共资源中找到这些数据是非常容易的事情，因此产生了大规模的身份盗窃行当［285］。罪犯取得了信用卡、移动电话和你名下的其他东西，抢夺这些资产，并让你收拾残局。在美国，每年大约有 50 万人遭遇这种欺诈行为。

口令是安全工程目前面临的最大的应用问题之一。它是很多信息安全建立的基础（通常都不太可靠）。如果频繁使用（这样口令就很牢靠地保留在记忆中）并有连续的上下文（这样不同的口令不会在你的记忆里互相干扰），记住口令是可能的事情。当人们要为数量繁多而又不常访问的网站选择口令时，上面两个条件都不能满足。于是当他们在越来越多的电子系统中成为当事人时，就一次又一次地使用相同的口令。不仅局外人可能猜测到，而且其他系统里的成员也可能采取攻击。

3.1 基础

在一般的系统中，用户必须向客户端（可以是 PC 机、移动电话、ATM 机或者其他）证实自己的身份，接下来客户端要向一个或者多个服务器或者服务（例如电子银行系统或者电话公司）证实自己的身份。像第 2 章“协议”中所阐述的，电子设备彼此认证或多或少是一个可以管理的问题（至少理论上可行），而向设备认证一个人比较困难。

基本上有三种方法可以做到。第一种是人掌握设备的物理控制——像遥控汽车门锁钥匙、PDA 甚至笔记本电脑。第二种是他（她）提交所知道的某些东西，例如口令。第三种是

使用生物识别，例如指纹或者虹膜的形状。这些选择一般概括为“你拥有的东西、你知道的东西和你本身是什么”。由于成本的原因，大部分系统选择第二种。即使是使用一种物理卡片/设备，比如一部手持式口令生成器，通常也要使用口令来锁定它。

所以口令和管理口令问题是现实世界的一个严重的问题。本书首先着眼于人的问题，然后阐述各种攻击模型，最后谈论技术攻击和防御。所有这些问题都很重要，因此只偏重其中一个就会导致糟糕的设计。

3.2 实用心理问题

人们关注的问题基本上有三点：

- 用户把口令透露给第三方而破坏了系统安全，是无意的、故意的还是受骗的结果？
- 用户正确输入口令的概率是不是足够高？
- 用户会记忆口令，还是会把口令写下来或者选择一个攻击者容易猜到的口令？

3.2.1 社会工程

信息保密性面临的一个最严重的实际威胁就是，攻击者通过讲述一些似是而非的假话，可以从有权访问的人那里直接得到口令。这种攻击名为社会工程，本书将在第 8 章专门处理医疗系统时讲述，因为它是医疗隐私目前面临的主要威胁。有一种常见的攻击就是保险调查员，他向医院或者医生的办公室打电话，假扮成护理所要调查目标的一名急诊医师。这个技术，在英国称为“闲聊”，在美国称为“托辞”，被普遍地用来从银行、保险公司和其他掌管个人信息的公司中得到信息。有些人以此为生 [261]。

假的托辞电话经常能够得到口令。有人以总经理人事助理的身份给系统管理员打电话，前一两次询问一些小事情；一旦管理员相信了她的话，她就会找借口说急需高级口令。除非公司具有严密的策略，否则这种攻击非常容易得逞。例如有一次系统的实验调查，向悉尼大学 336 名计算机科学专业的学生发出电子邮件，声称发现了一次可疑的闯入而需要“验证”口令数据库，要求他们填写自己的口令。其中 138 名学生填写了有效的口令。一些人有点怀疑：30 人返回了看似正确而实际上无效的口令，而 200 多人在没有正式提醒的情况下更改了口令。但是他们当中几乎没人向管理机构报告这份电子邮件 [354]。

为了堵住这个漏洞，有家公司的策略规定：“每台机器的根口令要长到不易记忆，由系统随机选择至少 16 位字母和数字字符；这个口令应该写在一张纸上并装进信封，保留在机器所在的房间里；任何情况下都不要通过电话泄漏或者在网络上使用它。它可能仅在它控制的机器的控制台上输入。”如果整个组织都严格地履行这样的规则，那么对根口令进行托辞攻击就显而易见，不太可能成功了。

还有另外一种方法，就是 NSA 所采用的，内部和外部电话使用不同颜色，彼此不互相连接，规定当某个房间的外部电话接通时，就不可以在这个房间中讨论机密资料，更不用说在电话中谈论了。有一种不太极端的方法（我的实验室里采用），就是内部和外部打来的电话使用不同的震铃声音。这样只要有警报系统管理员就行。像第 2 章讨论的口令生成器这样的物理认证设备效果更好，但是通常比较昂贵，并与遗留系统不兼容，或者与某些策略（不管是否合理）相悖。

3.2.2 可靠口令输入的困难

第二个关于人的问题是，如果口令太长或者太复杂，用户可能就不容易正确地输入。长的随机口令会让输入者头晕，如果他们要进行的操作非常着急的话，可能会产生安全问题或者其他麻烦。

在加密访问码的应用中，这点比较重要。人们说出预订号，可以入住旅馆房间或者租到汽车。飞机票检票也采取这种形式，营业员给乘客一个号码而不是登机牌，以便他们在登机口交验。当号码变得很长时，错误率必然会增加。

南非做了一次有趣的研究，其背景是某些地区采用预付费电表售电，顾客没有信用等级甚至也没有地址。顾客有一个电表，向销售代理交钱，然后得到一个或者几个打印在收据上的20位数。他把收据带回家，在电表的键盘上输入这些数。这些数是一些加密的指令，可以分配电量、改变资费等等；电表译码并依此进行操作。

当引入这个电表的时候，人们担心三分之一的人口是文盲，在输入号码的过程中会不知所措，所以这个电表不会得到推广应用。但是结果表明文盲并不是问题，人们即使不识字也会输入号码。(正如一名工程师所说："每人都会使用电话。") 输入错误是一个更大的问题，但是把20位数字印成两排，第一排三组、第二排两组四位数号码，已经解决了这个问题[39]。

美国核武器发射号码是一种截然不同的应用。它只包括12位十进制数字。如果一旦要使用，操作员将处于极度的压力下，也许会使用临时的或者荒废的通信渠道。试验表明12位数字是能在这种环境中可靠传输的最大值。

3.2.3 记住口令的困难

对口令最多的抱怨就是它难于记忆 [146, 823]。12到20位数还可以从电报或者电表票据上抄写一遍，但是当需要记忆口令的时候，顾客不是选择了别人容易猜到的数字就是把口令写下来，都很不安全。

问题并不局限于计算机访问。例如，法国一家连锁旅店实行完全无人值班的服务。人们找到旅店，在收银机上划信用卡，取得一张收据，上面印有数字访问码，然后就可以打开房门了。为了降低成本，房间里没有卫生间，这样客人必须要使用公共厕所。通常失败的例子都是客人上完厕所以后就忘记了他的访问码。除非他随身携带了收据，不然他只有睡在厕所的地板上，直到第二天早晨员工来。

与口令记忆有关的问题可以分成两类讨论：设计错误和操作错误。

3.2.3.1 设计错误

试图设计系统而让口令变得易于记忆，经常会产生严重的设计错误——尤其是那些没有技巧的人匆忙建立的电子商务系统。举一个具有教育意义、同时也是很重要的例子，即为什么不能向顾客询问"你母亲的婚前姓名"。很多银行、政府部门和其他机构以这种方式确认顾客的身份。这有两个十分严重的问题：首先，母亲的婚前姓名很容易被贼掌握，他可以向周围的人询问、查询出生和婚姻登记或者使用在线家族数据库；其次，即使你决定从现在开始让你母亲的婚前姓名变为Yngstrom（或者甚至是yGt5r4ad），而不用Smith了，仍然有问题，因为你提供了假数据，可能违反了信用卡协议，还会使保险合同无效。

此外，询问婚前姓名所作的假设并不是在所有国家都成立（冰岛人没有姓，并且很多其他国家的妇女出嫁后不改名字）。一般可能没有人准备改这个口令，所以如果贼知道了口令，你就必须停止账户并重新开一个账户。最后，你会被很多机构问及这个口令，其中任何一个部门都有可能存在不老实的员工。你必须一直向银行说“Yngstrom”，向电话公司说“Jones”，向旅游部门说“Geraghty”等等；但是各公司之间的数据广泛地共享，因此你可能很快就把系统搞昏头了（更不用说你自己）。

有些电子商务网站在设计时考虑得稍微周到一点，会询问一个口令而不是婚前姓名。但是现在向人们要求口令的应用程序数目已经超出了人的记忆能力。所以顾客要么把口令写下来（不顾忌警告），要么在很多不同的地方使用相同的口令。这样，用这个口令在设计好的电子银行系统中认证顾客的身份，很有可能被黑手党色情站点截获。

顾客面临的风险不仅仅是身份盗窃和诈骗所带来的直接损失。恶劣设计的口令机制会摧毁顾客的信誉，并导致丧失合法权益。例如，如果一个贼成功地伪造了你的 ATM 卡，然后支取你的银行账户，银行就会询问你是否曾经和别的人或者公司共用了个人身份证号码。如果你承认在移动电话上也使用相同的号码，银行会说是你使用电话时太粗心被别人听到了，或者归咎于电话公司的人。在两种情况下，你都无法找到人并且控告他们。

有些机构努力寻找其他的安全信息。我的银行向顾客询问最近一次从账户支走的支票的数额。理论上，这是一个好系统：优点是即使有人危及到顾客的口令——比如在顾客做电话交易时无意中听到——系统安全也会或多或少地自动恢复。但是实现细节却需要注意。起初引入这个系统时，我心存疑虑，刚给一家厂商开过一张支票，他应该有机会假扮我。我认为向顾客询问最近三次支票数额会更加安全。但是我真正遇到的是另外一个问题。把支票簿交给会计做年度审计以后，就忘记了这个数额，无法通过电话认证自己的身份，即便查余额也必须亲自到银行去。

寻找其他解决方法更是经常触礁。一家银行向顾客发信警告他们不要写下 PIN，而是提供了一个与众不同的纸卡，认为可以靠以下的方式隐藏 PIN：假设 PIN 为 2256，选择一个四字母的单词，比如 blue，在卡上的第 2、2、5、6 列分别写下这四个字母，如图 3-1 所示。然后把其他空格写上随机字母。

1	2	3	4	5	6	7	8	9	0
	b								
	l								
				u					
					e				

图 3-1　一个糟糕的银行 PIN 记忆系统

这显然是一个糟糕的想法。即使随机字母没有采取稍微不同的写法，快速检查一下也可以看出 4 × 10 的随机字母矩阵可以提供 24 个单词（除非底下一行有“s”，那样可以达到 40 到 50 个单词）。所以贼三次就能猜测到 PIN 的几率从需要 3000 个样本缩减到只需 8 个样本。

有些银行允许顾客选择自己的 PIN。相信有三分之一的顾客会使用出生日期，这种情况下，防止贼猜对的几率是稍微大于 100（如果贼认识受骗者的话会更小一些）。即使可以接受这个风险，PIN 很可能还会和家人共用的移动电话 PIN 设的一样。为了分析这个问题，本书不得不在后面考虑很多不同的威胁模型，将在下一节中讲述。

3.2.3.2　操作问题

机构应该制订和执行什么规则来支持所用的口令机制，想不通这个问题导致了一些非常引人入胜的案例。英国在 20 世纪 80 年代后期有一个重要的案子，就是 R 诉 Gold 和 Schi-

freen。被告在一次展览会上发现一台电脑终端贴着个便条，上面记有 Prestel 开发系统（英国电信使用的早期公共邮件服务）的电话号码。他们后来拨打电话，发现欢迎屏幕上显示有最强大的维护口令。经过在系统上试验这些口令，发现可以使用！他们进一步掌管了爱丁堡公爵的电子邮件账户，“从”他那里向不喜欢的人发信件，宣布授予他骑士身份。这个案子给当权者带来了巨大的震惊，当公诉人在当时的法律下败诉以后，国会就通过了英国的第一部计算机犯罪法律。

把管理员口令装入信封贴在工作站侧面，如果办公室经常有人看管或者上锁，在某些环境下也可行。能够得到口令的人都是那些可以直接接触到这台机器的人。但是如果采用这样的策略，就必须注意确保人们理解这样做的原因，从而不会把口令轻易地留在登录屏幕上。但是让人们在投入使用的系统使用和开发环境中一样的管理员口令，就不值得原谅了。

一个常犯的类似的错误就是，没有重设某些系统服务的默认口令。例如，在 20 世纪 80 年代，一个非常畅销的拨号软件默认用户名 999999 和口令 9999，还默认管理员名字为 777777，口令为 7777。大部分使用该软件的站点都没有换口令，结果很多网站都被黑掉了。未能换掉设备的默认口令已经影响到各种计算机、某些加密设备甚至还有移动电话（很多用户从来不考虑改变安装 PIN “0000”）。

3.3 系统问题

了解了用户心理以后，本书接下来要了解攻击者的心理。正如只能在特定的威胁模型下才能谈安全协议的健全性一样，本书只针对要防范的攻击来判断给定口令方案的健全性。广泛地讲，有：

针对某个账户攻击。入侵者试图猜测某个特殊用户的口令。他也许想猜测比尔·盖茨的银行账户，或者办公室里冤家对头的登录口令，目的是直接搞破坏。

试图突破一个系统中的任何账户。入侵者试图以一个系统中的任何用户身份登录。这样可以直接窃取服务（比如免费享受电话卡服务），或者作为进一步攻击的跳板。

试图突破任何系统的任何账户。入侵者想在某个域的任何系统中有一个账户，是哪个域并不重要。例如，一般十几岁黑客的目的就是要找个地方收藏盗版软件或者色情图片，或者找一个平台对其他系统发出匿名攻击；有经验的人攻击目标会带来更严重的威胁。想寻找军事机密的间谍最初会努力攻击“.mil”域名内的任何计算机，而进入微软内部网络的私家侦探的任务可能只是需要登录“microsoft.com”里的某些机器。

拒绝服务攻击。攻击者可能希望阻止合法用户使用系统。攻击的目标可能是一个特殊账户（例如取消某人的信用卡来骚扰他）或者整个系统。

这样进行分类是有用的，当选择或者设计口令系统时，可以帮助你想到相关的问题。然而，还有些问题会关系到假想攻击和能采用的对策的类别。

3.3.1 保护自己还是保护他人

首先，系统需要对用户保护到何种程度？有些系统不能花别人的钱来享受服务，比如移动电话和提款机系统。我们假定攻击者就是系统中的一个合法用户。因此肯定要（至少是应该）在设计时考虑周到，当知道一位用户的口令以后，不会危及其他可以确认的账户的安全，从而采用多边安全（本书将在第 8 章以更多的篇幅讨论）。用户选择了容易被猜出的口

令只会损害到自己，因此增加口令强度的做法可以理解。（记住人们选择的口令通常很容易被配偶猜到［146］，所以应该多考虑一下，比如配偶受骗后会怎样报复。这是一个再普通不过的问题。）

但是对于很多系统，即使只有一个敌人入侵也是个噩耗。Unix 和 Windows 这样的操作系统，设计时考虑到要防止一个用户被其他用户偶然干扰，但是却没有进一步防止其他用户可能采取的恶意攻击。这些系统有很多知名的脆弱性漏洞，被不时地公布在网络上。竞争对手如果在一个共享计算机系统中取得了一个账户，他通常很快就可以当上系统管理员，并可以在那里为所欲为。典型的开拓路径就是从外部人员到普通用户再到管理员，其中第一步比较艰难。因此让用户随意选择自己喜欢的口令可能不是一件好事情。特别在军事系统中，一般都给用户分配随机口令，以保证最小的口令强度（本书以后还会说明）。

3.3.2　入侵检测问题

第二个问题是关于口令系统和入侵检测系统相互作用的方式。像银行这样的机构通常规定，输入三次错误口令以后，终端和用户账号就被冻结了；然后要与管理员取得联系，重新激活它们。在军事系统中这样是很危险的，因为敌人进入网络以后会采用大量的虚假登录请求，从而产生拒绝服务攻击；如果给出一个机器上所有用户名字的列表，很有可能使服务完全崩溃。

不只是要在军事系统中注意这一点。电话卡是另外一个例子，它们通常有一个前缀，后面是要入账的本地电话号码和四位数的 PIN。电话公司扫描电话卡的号码，把具有多个 PIN 号码的卡取消。这并没有防止人们取消别人的卡（同时也不能防止诈骗犯猜测到有效的卡号，因为他可以用一个 PIN 试验所有的本地电话号码）。

入侵检测系统的设计因目标不同而大大地不同。有简单的阈值报警，账户在三次失败登录以后就被关闭；还有更加复杂的分布式系统，用来对付入侵者在很多账号上试验一个口令，或者在每台机器上试验一个账号，等等。入侵检测会在第 18 章中详细讲解；这里，仅仅指出口令和入侵检测之间相互影响的事实。

3.3.3　可以培训用户吗

第三个问题是用户能否培训和规范？在公司或者军事环境里，甚至在大学中，可以控制用户数量。可以教他们如何选择好的口令；如果他们选择糟糕的口令可以给出负面评价；可以给他们发随机口令，并规定如果他们把口令写下来就必须像保护数据一样保护它（因此“绝密”口令应该装进信封，放进保险箱，没人时把房间上锁，并且楼内要有警卫巡逻）。要保证只有可靠的人才能靠近口令所在的终端。夜间可以派警卫检查，保证周围没有留下写着口令的纸条。可以推行整理桌面的做法，这样就不会在一堆纸张中疏漏什么。

我和同事研究了培训用户的好处［815］。写这本书的时候，我找不到任何实用心理学在这方面的试验记录（就是找足够多的人做随机试验，得出统计结果）。最相近的一次研究是调查各种口令的回忆率、遗忘率和猜测率［146］；这是有价值的，但是并不能告诉我们给用户各种建议所带来的实际（而不是“可能”）影响。因此，我从一年级学生中选择了三组志愿者（大约 100 人）。

- 给红（控制）组平常的建议（口令至少选择六位字符，其中包括一个非字母字符）。

- 让绿组的人想出一个短语，从中选择一些字母组成一个口令。这样，“It's 12 noon and I am hungry”组成口令可能就是I'S12&IAH。
- 让黄组的人从交给他们的表中随机选择八位字符（字母或者数字），写下来，等他们记住口令以后就把该表格毁掉。

我做试验的目的是为了验证红组的口令比绿组的口令易于猜测，而绿组的口令要比黄组的口令易于猜测；同时验证黄组的口令最难记忆（或者被迫重新设定口令的次数更多），接下来依次是绿组和红组。然而试验结果却并非如此。

红组中30%的人选择了能被破解软件（本书后文再介绍）猜测到的口令，其他两组大约有10%。因此短语和随机口令的效果看来是一样的。我查看口令重新设定率时，发现三组并没有明显的差别。我向学生询问他们是否觉得口令难记（或者是否写下来了），黄组的问题尤其比另外两组多；但是红组和绿组之间没有大的差别。

因此可以得出如下的结论：

- 对于遵从口令设置要求的用户来说，采用以记忆短语为基础的口令在两个方面都有最好的性能：它们既像直接选择的口令一样好记，又像随机口令一样难猜。
- 仅仅采用记忆口令或者随机口令并没有太大帮助，因为问题在于用户服从。大多数用户（可能占1/3）根本不听从口令设置要求。

因此，当集中分配口令时，采取随机口令从军事角度是一个较好的策略，它的价值在于口令是集中分配的（强制用户服从），而不是随意分配的（和记忆口令的效果是一样的）。

然而，至少有两种情况不宜采用集中式口令分配。第一种情况是用户拥有系统无法访问的某类资源访问权限。这里用到数字签名作为证据，而用户的数字签名密钥是靠口令保护的，如果使用集中式口令分配，系统管理员就有权得到签名密钥并且伪造消息，从而毁坏了签名的证明价值。

第二种更加微妙的情况是向公众提供服务的系统。不管你是通过专用终端提供服务，比如提款机和移动电话，还是通过网络向标准PC机提供服务，都无法指望能培训和规范用户；如果你尝试这样做的话，会让法官觉得你的合同条款不合情理。

或许理想的解决办法是指导用户去选择记忆口令，并且安装一个口令破解程序作为口令过滤器；用户如果选择了猜测列表中的口令，就告诉他重试一遍。不管怎么说，确实还需要对这方面的问题作更多的心理调查。

3.3.4 日益缺乏数据安全

第四个问题是一个真正困难的问题：用户是不是在其他系统上使用同样的口令而导致威胁？

可以自由选择口令或者PIN的人通常在很多系统上使用同样的口令，这样便于记忆。如果某些系统不允许顾客更改PIN的话，有些人就会把PIN写下来。虽然可以在合同上禁止这样做，但是他们仍照做不误。有些人干脆就换了别的地方做生意（如果可以选择的话，我宁愿选择根本不需要口令的网站，不管该网站是可以选择不要口令还是它本身就不需要口令）。

用来识别身份和授权行为的良好安全信息非常缺乏。曾尝试采用“多功能”智能卡来解决这个问题，但已经引起了争论，焦点是谁的标志在前面和由谁控制邮件列表。如果采用现有的设备（比如移动电话）进行授权交易的话，在基础设施上就只需较少的花费。即使在

几年以后，每人都拥有一部第三代移动电话，能够用它进行银行交易，并且可以接收含有基于网络的电子商务交易授权码的加密消息，仍然会有很多技术问题需要解决。这些问题包括前面提到的选择协议攻击，防止程序之间互相干扰的困难，本书后面即将讨论此类问题。

从商业和法律事务中还会引发更严重的问题，比如，如果某公司控制了个人的信用卡数据或者出卖历史数据，或者两者兼有，会怎么样呢？本书将在第19章至第21章讨论这类问题。

3.4 口令的技术保护

很多攻击都可以恢复别人的口令。有些攻击以口令输入机制作为目标，而有些则着眼于口令存储。

3.4.1 口令输入攻击

口令入口的防护一般很薄弱。

3.4.1.1 界面设计

有时考虑不周的界面设计会产生问题。例如，某些常用提款机的模型就是在人头部的高度放置一个垂直键盘，这样小偷很容易看到顾客输入PIN，然后再偷走顾客手袋中的钱包。键盘的高度是按照设计者的高度为标准的，但是女人和有些国家的男人的身高会比其矮几英寸，这样输入PIN时就非常容易暴露。比较讽刺的是，有一种提款机还迫使顾客从一个窄缝中盯住屏幕，以“保护客户隐私”。你的收支是隐秘了，但是PIN并没有隐秘！

很多付费电话也有类似的问题，在某些地点，比如美国大型火车站和飞机场，获得电话卡的详细数据的肩部偷窥已经成为地方病。为了防范起见，我在公共场合输入卡号和PIN时，一般都用身体或者另外一只手覆盖住拨号的手——但是系统并未设计成假设所有的顾客都会这样做。

3.4.1.2 窃听

留心口令输入可以防止坏人在机场电话旁透过肩膀偷看你使用电话卡，但是却不能防止所有的窃听攻击。例如，饭店经理可能会使用交换设备记录你在房间内电话上键入的号码；如果你不使用电话卡的话，他偷钱的风险就小了。

很多联网的计算机系统在局域网内按明文方式发送口令，让服务器检查；如果有人在局域网的某台机器上设计了一个程序或者附上自己的探测设备，就能取得口令。这也是微软在Windows 2000中采用Kerberos认证协议的原因之一——不在网络上传送明文口令（Windows NT 4使用一种专有认证协议）。

3.4.1.3 需要可信路径

你登录的机器也许带有恶意。在公共机房的无人看管机器上，就可能运行着一个简单的攻击程序；它看起来就和普通的登录屏幕一样，提示输入用户名和口令。当毫无防备的用户照此输入以后，攻击程序会把口令保存在系统的某个地方，然后回答“对不起，口令错误”并退出，之后再调用真正的口令程序。用户会认为他第一次输入错误，而不会想太多。这就是为什么Windows NT具有一个“安全注意顺序”（secure attention sequence），名为ctrl-alt-del，保证给你真正的口令提示。能够保证用户与真正的系统交谈的工具就是可信路径。

如果整个终端设备都是假的，很显然所有的措施都失去了意义。我有一次在公共机房抓到一个学生，他通过安装篡改的键盘来获取口令。如果攻击者准备采用更加高级的口令窃取手段的话，完成所有的 ctrl-alt-del 序列能够使软件设计更为简单隐蔽。

还有几个放置虚假提款机的案例。1993 年康涅狄格州有一个著名的案件，罪犯竟然买了真正的提款机（凭借信用），安装在购物场所，然后从使用它的的银行顾客那里收集 PIN 和卡数据 [19]。同年，伦敦的罪犯也延续了这个想法，并扩大化，完整建立了虚假的银行支行 [405]。还有其他的案件涉及到改装提款机，在真正提款机前面放置一个伪造的前端，甚至在 ATM 设备入口处替换了用卡控制的门锁。某些国家的销售点使用带有 PIN 的卡，这样使得此类攻击就更容易奏效。

3.4.1.4 击败口令重试计数器的技术

很多小孩子发现，只要松开自行车暗码锁的每个环，在几分钟内就可以解开它。同样的想法也适用于很多计算机系统。PDP-10-TENEX 操作系统检查口令时，每次检查一个字符，只要一个有错误就马上停止。这导致了计时攻击，攻击者把一个猜测口令反复放在内存中合适位置，在文件访问请求时检查它，等着看多长时间以后被拒绝 [493]。第一个字符错误马上就会通知，第二个字符错误会花稍微长的时间通知，第三个字符错误的话会更长，依此类推。就有可能依次猜测到每个字符，相对于从 A 个字符的表中找出 N 个字符的口令平均要猜测 $(A^N)/2$ 次，这个方法只要猜测 $A^{N/2}$ 次（记住，30 年以后，你现在建立的系统可能就是有报道价值的安全失败例子）。

关于遥控汽车门锁设备有一个类似的例子：只要钥匙链发出错误的字节，接收器的报警红灯就亮。

还有其他的方法能让口令重试的限制失败。对于有些智能卡，通过输入每个可能的值，并查看卡的耗电量，如果输入错误就重置，这样就能检验出顾客的 PIN。其原因在于，错误的 PIN 导致 PIN 重试计数器减一，而把结果写到含有该计数器的 EEPROM 存储器中要耗费几毫安的电流，这样在计数器写完之前可以及时检测到并重置卡 [478]。

3.4.2 口令存储攻击

口令存储的地方一般漏洞也比较多。20 世纪 80 年代某操作系统更新中有一个十分可怕的漏洞：用户如果输入了错误的口令并被告知“对不起，口令错误”，只要敲击回车键就能返回系统。很快就有人发现了这点，并提供了一个补丁，但在德国大约有一百台美国政府的系统使用盗版软件，而没有得到补丁，结果攻击者闯入这些系统并偷取了数据，据传他们把数据卖给了苏联国家安全委员会。

另外一个可怕的程序错误给一家英国银行带来了打击，他们给所有的顾客错误地分发相同的 PIN。由于对 PIN 的处理过程控制谨慎，银行中没有人能得到其他人的 PIN，因此直到数以千计的卡收放出去以后才发现了这个漏洞。

3.4.2.1 通过审计追踪攻击

在对失败登录尝试进行日志的系统中，日志中通常包括大量的口令，因为用户常常把“用户名，口令”顺序搞反。如果日志没有好好地保护，攻击就变得很容易。当人们看到一个不存在的用户“e5gv，8yp”的错误登录记录时，99% 的可能是系统中一个有效用户的口令。

3.4.2.2　单向加密

对于有些系统，口令存储也是一个问题。把口令存储为纯文本文件是危险的。麻省理工学院的兼容时间共享系统 ctss（Multics 的前身）中，曾经有一个人编辑当天的消息，而另外一个人编辑口令文件。由于系统的 bug，两位编辑的临时文件交换了，结果登录的每个人都得到了一份口令文件副本！

经历这样的事故以后，通常用单向加密算法对口令加密以保护口令，这归功于 Roger Needham 和 Mike Guy 的创新。口令输入以后，通过一个单向函数，只有和以前存储的数据相匹配时，用户才能登录。

然而，有时候不能使用单向加密来保护含有安全信息的文件，例如这些信息需要以某种方式处理。GSM 移动电话是一个经典的例子，每个用户在主地址注册数据库（home location register database）中有一个加密密钥。由于这把密钥用于计算质询响应对，通过无线方式认证用户，并采用明文方式（本书将在第 17 章讨论这种设计决策的理由和其他替代方案）。

3.4.2.3　口令破解

有些使用加密口令文件的系统把自己变成全世界可读的（Unix 是最基本的例子——由于一个设计错误，但是现在已经太深入人心而无法更改）。这意味着对手得到了文件以后，可以使用字典脱机破解口令；他把字典中的数值加密并和文件中的数据比较（这种行为称为*字典攻击*，或者更通俗的*口令破解*）。NT 稍微好一点，但是口令文件仍然可以被了解其用途的用户得到，这些口令也许能够在其他旧版本和出于兼容原因而使用旧的脆弱的保护机制的系统中通过（例如 Netware 或者 NT 的早期版本）。

人们总是在自己的设备上使用配偶的名字、单个字母甚至用一个空字符再敲回车键作为口令。因此有些系统要求最小口令长度，甚至把用户输入的口令放到字典中检验，看是否为糟糕的口令。尽管如此，设计一个口令质量增强机制要比想像中困难得多。Grampp 和 Morris 在关于 Unix 安全的经典论文［350］中提到，当软件上实现了强制口令长度不低于 6 并且至少含有一个非字母以后，他们把 20 个最常用的女性名字列出来，每个名字后面加一个数字。用这 200 个口令，他们检查了几十台机器，每台机器上都至少使用了其中的一个口令。

有一个报告指出，当用户被迫改变口令并且不能使用前面输入的几个口令时，他们就迅速搜索历史列表，返回到喜欢的口令。禁止用户在 15 天之内更改口令，这意味着没有管理员的帮助就不能换掉危险的口令［603］。以我的亲身经历看来，坚决要求使用字母数字口令并且每个月强迫改变口令会导致人们选择 julia03 代表三月，julia04 代表四月，依此类推。因此我不认为经常改变口令是一件好事情。

Klein 作了一次著名的研究，他收集了 25 000 个存储在加密口令文件中的 Unix 口令，并运行破解软件来猜测它们［460］。他发现 21%～25%的口令能够被猜到，猜测率依赖于所作的努力。其中字典单词占 7.4%，普通名字占 4%，用户名和账户名合起来占 2.7%，还有其他一些较小概率的选择，比如科幻单词（占 0.4%）和运动术语（占 0.2%）。一些口令是直接从字典查到的；其他的使用模板。例如，重构用户名和账户名组合的算法认为 klone 属于 Daniel V. Klein 这位用户，就试验例如 klone、klone1、klone123、dvk、dvkdvk、leinad、neilk 和 DvkkvD 等等口令。

有一些公开的程序（Unix 使用 crack，Windows 使用 LOphtcrack［481］）可以做这种搜索。系统管理员可以用它们来查找系统中糟糕的口令。坏人也可以同样方便地用它们来检索得到

的口令副本文件。因此口令破解是需要引起注意的事情，尤其是当系统中包括任何运行 Unix 或者 Linux 的机器时。习惯上用 crack 这样的程序过滤用户选择的口令；另外一种情况是，使用了解语言统计特性的常规程序并且拒绝掉那些很可能被别人随机选择的口令 [98, 220]；还有一种情况是采用恰当的编码方案把上面两种方法结合起来 [725]。

3.4.3 绝对限制

不管口令管理得多好，操作系统或者其他平台一般对建立其上的系统都有一些绝对限制。例如，Unix 系统把口令的长度限制为八位字符（你可以输入更多位口令，但是第九位和其后的字符被忽略）。试验所有可能的口令要进行的尝试——加密的行话为“总耗次数”——是 96^8 或者大约是 2^{52}；平均每次搜索的次数为这个数字的二分之一。现在，条件好的政府部门（或者组织有序的黑客集团，他们使用分散在因特网上的 PC 机）已经可以破解 Unix 标准口令文件中的任何加密过的口令。

这激发人们采取更多的技术防御措施来对付口令破解，包括影子口令（shadow password）口令，也就是，把加密的口令隐藏在一个私有文件（最新的 Unices）里，使用一个模糊机制进行加密（Novell），或者在加密中应用秘密密钥（MVS）。这些机制的强度有所不同。

由于以上原因，军事系统的管理员一般宁愿随机分配口令。这样也可以估算出猜测口令攻击的概率并进行管理。例如，假设 L 是口令最大生存期，R 是登录请求率，S 是口令空间的大小，那么一个口令在生存期内被猜中的概率为：

$$P = LR/S$$

这个公式摘自美国国防部的口令管理指南 [242]。它存在两个问题。首先（比较次要），口令空间可能会用完，这时 $LR/S > 1$，P 不符合概率的定义。第二（更严重些），攻击者对猜测口令的兴趣远不如访问账户的兴趣。所以应该考虑用户的数量。如果一个大型防御网络有一百万个可能的口令和一百万个用户，任何账户只要三次口令错误就报警，那么可能的攻击方法就是使用一个口令对所有的账户都登录一次。这样，真正的重点在于，在最大可能的口令猜中率下，系统生存期内口令空间被用完的概率。

举个具体的例子，英国政府系统趋向用一个固定模板来分配随机口令，这个模板由辅音 C、元音 V 和数字 N 组成，易于记忆，形如 CVCNCVCN（举例，fuR5xEb8）。如果口令不是大小写敏感的，猜测概率仅为 $21^4.5^2.10^2$，或者大约为 2^{29}。所以假设一个攻击者每秒能猜 100 次口令——网络的几百台机器上可能分布着10,000 个账户，为了不至于引起报警——那么他大概需要 5 百万秒，或者两个月时间，才能进入网络。

在商务系统中，你可以采取这样的策略，在一定次数的口令登录错误以后就封锁账户。如果阈值设定为一个月内三次错误猜测的话，那么对于 10 000 个账户，最大猜中率为每月30 000 个口令，当猜测发生在不被人察觉的前提下，猜中率更低。但是军事系统的设计者们不愿意采用账户封锁的方法，因为那样会引来拒绝服务攻击。本书在 3.3.2 节中提到，敌人如果进入了网络并且输入了错误的口令，能够使整个系统的账户全部冻结起来。

3.5 小结

口令管理是许多安全系统中最重要，也是最困难的设计问题。由于人们在越来越多的系统中拥有账户，他们重复使用口令的方式暴露了系统严重的脆弱性。即使用户在受控的环境

中进行操作，问题也决不简单。

脱机猜测口令的本领使得攻击者能够或多或少威胁到系统中的某些账户，除非口令是集中分配的，或者用户选择口令时经过了过滤。在这里，人们也有可能阻止脱机猜测，比如保持口令文件的保密性。但是人们购买像 Unix 这样的系统一般都是因为它有强大的软件基础，对于用户要使用的软件，要改变其口令机制是很困难的。

设计一个口令系统时所提出的典型问题不仅有人们是否重复使用口令，还有他们是否需要彼此保护，用户是否可以培训和规范，以及在一定错误猜测后账户是否应该被冻结。你还必须考虑到，攻击者是针对一个特定的账户，还是随意闯入一个机器或者一个网络上的任何账户；另外考虑一下技术保护的焦点问题，比如口令是否会被恶意软件、虚假终端或者网络偷听器窃取到。

研究问题

本书已经提到了有关本章内容发表的经验调查是很少的。过去做得很少，而且现在还有很多问题有待研究。例如，使用户服从一个口令策略的最好方式是什么？有一些极端的解决方案——比如给每个用户发放一张随机口令列表，其中每个口令只能使用一次——这显然起作用。但是在有些应用中这些极端的手段不合理时该怎么办？

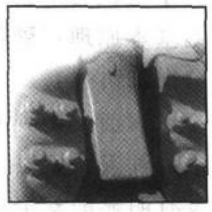
还有另外一个问题，它跨越了安全协议设计的界限，就是能否设计出更好的交互式口令系统。在学术界有各种可视化方案和记忆方案，还有一些早期产品：有一个系统向用户提交九个面孔，其中只有一个面孔是用户认识的；他们必须在一排面孔中几次找出正确面孔才能登录 [223]。另一方案是给出一张表，让用户动脑筋做一次秘密的计算。设计这样的方案是非常简单的；评估它们却非常困难，因为要涉及密码学、心理学和系统工程的知识。

日益常用的机制就是，提问多条安全信息，而不是一条。电话中心可能不仅仅问你母亲的婚前姓名、口令和最后一次交易的数额，还会问及狗的昵称和你喜欢的颜色。其基本思想就是，尽管攻击者可能发现一些事情，但是了解你知道的所有事情会困难得多。再者，需要用应用心理学的工具仔细评估此类方案的可用性和有效性。

参考资料

有关口令的学科尽管很重要，却并没有太多文献。Bob Morris 和 Ken Thompson 的论文 [561]，Fred Grampp 和 Bob Morris 的论文 [350]，还有 Dan Klein 的论文 [460]，都是经典之作。DoD 方针也有重要的影响 [242]。

第4章 访问控制



回顾早期的分时系统，我们的系统人员把用户和他们编写的代码当作致命的敌人。我们仿佛是暴力贫民窟中的警察。

——Roger Needham

微软把有效的安全措施具体化为标准，但是高超的应对手法层出不穷。安全系统有一个适得其反的坏毛病，无疑会引发重大的问题。

——Rick Maybury

4.1 引言

访问控制是计算机安全的传统重心。它是安全工程和计算机科学的交点。它的功能是控制当事人（人、程序、机器……）访问系统中的资源——哪个文件可以读，哪个程序可以执行，怎样与其他当事人共享数据，等等。

注意 本章比前面的章节需要更多计算机科学知识背景，但是我尽量将其减少到最小。

访问控制有很多工作级别，见图4-1，描述如下：

1）用户在应用层看到的访问控制机制具有非常丰富、复杂的安全策略。现代化的在线商务会在几十个不同的角色中给职员分配一个角色，每个角色都会在系统中初始化几百种可能交易的一些子集。有些交易（比如顾客进行信用卡交易）可能需要第三方的在线认证，而其他交易（比如还款）可能需要双重控制。

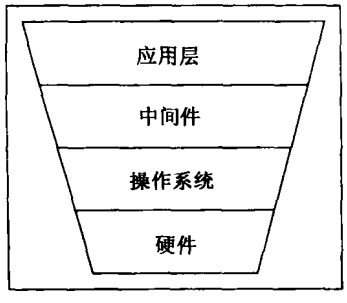


图4-1 系统中的各级访问控制

2）应用层可以写到中间件的顶部，如数据库管理系统或者簿记包，这可以增强很多保护属性。例如，簿记软件可以保证交易中一定数额的借款必须在其他账目上有一笔等额贷款记录。

3）中间件使用基本操作系统提供的设备。由于它由文件和来自底层组件的通信端口等资源构成，需要给其提供访问控制。

4）最后，操作系统的访问控制还依赖于处理器或者相关的内存管理硬件提供的硬件特性。它们控制着一个程序可以访问哪些内存地址。

随着我们从硬件发展到操作系统和中间件再到应用层，控制渐渐变得复杂、不可靠。实际上大多数计算机诈骗都是因为员工偶然发现他们可以投机取巧地利用某些应用代码，或者

在系统认为不会发生某些事情时却偏偏做出此类事件。但是本书在这一章将集中讲述访问控制基本原理：硬件和操作系统的访问控制（应用层控制有不同的原理，本书将在本书第二部分详细讲述）。

和前面讲到的各部分内容一样，访问控制只在以安全为目标的情况下才有意义，用来表达一种安全策略。当谈论到运行单用户操作系统的 PC 机时，比如 DOS 和 Win95/98，它们并没有明显的安全策略：每个进程都可以修改任何数据，这使得本书处于稍微不利的境地。尽管如此，人们还是有明确的保护目标；你绝对不希望一个压缩打包程序摧毁硬盘。所以制定清楚的安全策略是好的，尤其当产品支持一些看来应提供保护的特征时，比如登录 ID。

本书后面会提到一项保护技术——沙盒，关于病毒以及相关内容的讨论见 18.4 节。下面，将会讨论一下支持多进程隔离的系统的保护机制。首先讨论操作系统机制，因为它们的保护需求是硬件保护系统设计的驱动之源。

4.2　操作系统访问控制

操作系统提供的访问控制一般使用口令或者 Kerberos 机制对当事人进行认证，然后调度他们访问文件、通信端口和其他系统资源。

其效果可以用一个访问许可矩阵作为模型，列表示文件，行表示用户。r 表示可以进行读操作，w 表示可以进行写操作，x 表示允许执行一个程序，“-”表示没有任何访问权，如图 4-2 所示。

用户	操作系统	账户程序	账户数据	审计追踪
Sam	rwx	rwx	rw	r
Alice	x	x	rw	-
Bob	rx	r	r	r

图 4-2　原始的访问控制矩阵

在这个经过简化的例子中，Sam 是系统管理员，有完全的访问权（审计追踪例外，他只有读的权限）。Alice 是一位经理，需要运行操作系统和应用程序，但是只能通过批准的接口——她不可以篡改接口。她也需要读写数据。Bob 是审计员，可以读任何资源。

一般情况下这样做足够了，但是簿记系统比较特殊，不能这样做。我们需要保证交易组织有序——就是每次借款都有一笔贷款和它匹配——所以不能让 Alice 不受约束地向账户文件写数据。我们还希望 Sam 也没有这样的权利；所以向账户文件写数据的操作都是由账户程序完成的。访问许可情况如图 4-3 所示（还存在有一个间接的脆弱性漏洞，就是 Sam 有可能在未经授权的情况下自行覆盖账户程序，本书到第 9 章才会讨论这种情况）。

用户	操作系统	账户程序	账户数据	审计追踪
Sam	rwx	rwx	r	r
Alice	rx	x	-	-
账户程序	rx	r	rw	w
Bob	rx	r	r	r

图 4-3　簿记系统的访问控制矩阵

还可以采用另一方式阐述这个策略，就是用户、程序和文件三重访问。一般情况下，我们对程序的关注远不如对保护域的关心，它是一系列进程或者线程，享有同一资源的共享访问权限（尽管任何时候文件的开放或者调度优先权可能不一样）。

访问控制矩阵（二维或者三维）可以用来实现保护机制，也可以只是建立保护机制的模型。但是它们的缩放效果不好。例如，一家银行有50 000名员工和300个应用程序，那么访问控制矩阵就有15 000 000个元素。这太巨大了，不方便。它不仅有性能问题，还（可能）会导致管理员发生错误。通常我们需要一种更简洁的方法来存储和管理信息。主要有两种方法，采用组或者角色来同步管理大量用户的特权，或者按照列（访问控制列表）、行（能力，有时称为“票据”）或证书来存储访问控制矩阵[662，804]。

4.2.1 组和角色

考察大型机构的时候，通常会发现大多数员工可以划分到少数类别中去。一家银行可能有40到50个这样的类别：出纳员、总出纳、部门会计、部门经理，等等。其余的人（比如安全经理和主要外汇商……），需要单独定义访问权限，数量只有几十人。

所以我们想要预先定义少量组，或者功能角色，可以把员工分配进去。有些人把组和角色两个单词换着用，在很多系统上是可以这样做；但是严谨的定义如下，组是一组当事人的列表，角色是一些固定的访问许可权限，假定由一个或者多个当事人在一段时间内通过一些规定的程序使用。角色的一个经典例子就是一艘轮船上值班的官员。在任何时候都肯定有一个值班者，换班时官员根据规定的程序换成另外的人。实际上，在大多数军事应用中，这个值班者代表实际的角色而不是个体。

组和角色可以结合起来。海上所有船只上正在值班的官员就是一个角色组。在银行业，剑桥支行的经理可以通过经理组的成员和剑桥支行的代理经理的角色假设这两个条件得到其特权。经理 组表达出在某个机构中的级别（或许还有薪水级别），而代理经理的角色可能包括经理、副经理和支行会计都生病时代行职务的助理会计。

是否需要仔细对待这个区别是应用中面临的问题。在一艘军舰上，如果所有的官员都牺牲了，甚至可以让一位有能力的水兵来值班。在一家银行里，可能会有这样的策略：“一千万美元以上的交易必须有两位职员批准，一位是经理级别职员，另一位是助理会计级别职员。”有时会出现弊端，助理会计本身就是经理，在大额交易时就必须找地区总部签第二个名字。

直到最近，才出现对组和角色的支持，但是并没有得到广泛使用。开发者要么在应用程序代码中实现这种功能，要么添加在自定义中间件中（在20世纪80年代，我参与了两个银行项目，那里的组支持是手写代码实现的，作为大型机操作系统的扩展）。Windows 2000（Win2k）带有广泛的组支持，而学术研究人员已经开始研制基于角色的访问控制（RBAC），本书将在第7章讨论。现在只有等待，看这两种方法哪个会在应用发展中占主导地位。

4.2.2 访问控制列表

简化访问权管理的另外一个方法是，每次存储访问控制矩阵一列，同时还包括那一列所指定的资源。这称为访问控制列表，又名ACL。在本章前面的第一个例子中，文件3（账户

文件，即表中账户数据）的 ACL，如图 4-4 所示。

用户	账户数据
Sam	rw
Alice	rw
Bob	r

图 4-4　访问控制列表（ACL）

ACL 作为管理安全状况的方法有很多优缺点。可以分为 ACL 的一般属性和个别实现中的特殊属性。

ACL 广泛应用在用户管理自己文件安全的环境里，比如在大学和实验室里常见的 Unix 系统。当访问控制策略集中设置的时候，适合于面向数据保护的环境里；而不太适合于用户人数众多并且经常变化的环境，或者用户在某段时间把权限委托给其他用户来运行一个特定程序的环境。ACL 易于实现，但是作为一种运行时安全检查方法却不是很有效，因为操作系统知道哪个用户正在运行特定的程序，却不知道哪个文件调用后被授权访问过。操作系统必须在每次访问文件时检查 ACL，或者用其他的方式跟踪激活的访问权。

最后，把访问规则分配到 ACL 中，要找出用户有权访问的所有文件会很乏味。例如，调用一位刚刚解雇的员工的访问权，一般必须取消其密码或者靠其他认证机制才能完成。检查整个系统也是一个乏味的事情，比如确认不存在一个可以让任何人都有权写入的文件。这需要检查上百万个用户文件中的 ACL。

下面举两个 ACL 的重要例子：在 Unix 和 NT 上实现 ACL。

4.2.3　Unix 操作系统安全

在 Unix 和由其演变而来的 Linux 操作系统中，文件不可以拥有任意访问控制列表，但是会给资源拥有者、组和全世界（任意用户）赋予简单的 rwx 属性。这些属性允许文件可以读、写和执行。通常的访问控制列表显示一个标志，表明文件是否为一个目录；然后再为全世界（任意用户）、组和拥有者分别标出 r、w 和 x；这样它就具有拥有者的名字和组名了。一个带有全部标志的目录，其 ACL 为：

```
drwxrwxrwx Alice Accounts
```

在根据本章第一个例子所给出的图 4-4 中，文件 3 的 ACL 应该为：

```
-rw-r----- Alice Accounts
```

这里的文件不是一个目录；文件拥有者可以读写它；组成员可以读，但是不能写；非组成员没有访问的权限；文件拥有者是 Alice；组是 Accounts。

在 Unix 系统中，启动（操作系统内核）时就有控制权的程序被当作超级用户运行，可以不受限制地访问整个机器。其他所有程序都作为一般用户，由超级用户分配访问权。访问决策是在程序中相关的用户 ID 基础上得到的。但是，如果用户 ID 为 0（根用户），访问控制决策就是“yes”。因此根用户可以做任何想做的事情——访问任何文件、成为任何用户或者其他。除此之外，还有一些事情只有根用户才能做，比如启动某些通信进程。通常系统管理员可以得到根用户 ID。

这意味着（带有典型 Unix 色彩）系统管理员可以做任何事情，所以我们很难实现一个审计追踪，让其中包含他无法修改的文件。这不仅意味着在本例中 Sam 可以修补账户，如果他被错误地指控修改文件就很难进行辩护，还意味着如果黑客成为系统管理员就可以清除所有的入侵证据。平常的防御措施是把系统日志发送到一个加锁房间里的打印机上，或者——

如果数据太多——记录到其他机器上并由别人来管理。

伯克利系列产品，包括 FreeBSD，可以在某种程度上弥补这个问题。用户、系统或者两者只能添加文件、不能修改和删除它。在启动过程中，如果用户设置了一个足够安全的级别，以后文件就不可以被覆盖或者移动，即使根用户也无权这样做。各种军事系统更难实现责任分离。不过，防止根用户威胁数据的一个最简单、最普通的方法是，把数据保存在一个独立的服务器上。

第二，ACL 只包含用户名字，没有程序名字，这样就不能直接实现三重访问控制（用户、程序和文件）。Unix 提供了一个间接的替代方法：suid 和 sgid 文件属性。

程序的拥有者可以记为 suid。这样他就可以行使拥有者的特权，而不是调用它的用户的特权；sgid 在组中的作用是一样的。于是，为了得到图 4-3 所需要的功能，创建一个用户“account-package”来拥有文件 2（账户包），使文件属性为 suid，并把文件放到 Alice 可以访问的一个目录下。特定的用户就可以获得账户程序的访问控制属性了。

可以这样看待访问控制问题，就是本来应该用三维模型——三者（用户、程序和数据）——却用二维机制来实现了。这个机制没有三维模型直观，于是人们在实现时经常犯错误。程序员有时是因为懒散，有时是因为面临紧迫的交验期限；所以他们经常把程序做成 suid root，这样它可以做任何事情。

这种行为导致了一些非常令人震惊的安全漏洞。访问控制决策的责任从操作系统环境转移到了应用程序上，而且大部分程序员都不太有经验，并不能仔细地检查他们所做的每件事情。特别是，调用 suid root 程序的人将会控制它所在的环境，并可以巧妙地利用这点，从而导致系统保护失败。

第三，ACL 不善于表示变化的状态。这样，管理有状态的访问规则，比如双重控制，会变得很困难；人们要么在应用层实现它，要么再次使用 suid/sgid。另外，ACL 很难追踪一个用户可能打开过的文件（当在系统中调用他们的权限时，可能想这样做）。

第四，Unix ACL 只给一个用户命名：旧版本允许一个进程同时只能拥有一个组 ID，并强制它使用特权程序访问其他组；新的 Unix 系统在用户所在的全部组中提供同一进程。这甚至更加没有表现力了。理论上，ACL 和 su 机制经常用于实现想要的效果。实际中，程序员一般懒得指出怎样做，所以代码中设计所提供的权利要高于实际需要。

4.2.4 Windows NT

使用基于访问控制列表的保护机制的另外一类重要的操作系统就是 Windows NT。NT 目前的版本（版本 5，或者 Win2k）相当复杂，所以追溯从前的版本比较有帮助（如果你需要把 NT4 升级到 Win2k 的话，就更有帮助了）。

NT4 的保护实现与 Unix 很像，看来是从那里得到的启发，因此先简单地谈一下其主要的创新之处。

首先，NT4 不只具有读、写和执行属性，还有成为拥有者、改变权限和删除等单独属性，这意味着 NT4 支持更加灵活的授权行为。这些属性既可以用于组，也可以用于用户，组权限与 Unix 系统中的 sgid 程序具有相同的效果。属性不像 Unix 中只是简单的“有”和“无”，而是多值的：可以设置为 AccessDenied、AccessAllowed 或者 SystemAudit。按照这个顺序进行解析。如果在相关用户或组的 ACL 中看到 AccessDenied（不允许访问），就不用考虑任

何矛盾的 AccessAllowed 标志。

更多语法带来的好处就是，你可以安排各种事情，对于日常配置任务就不用授予全部管理员特权了，比如安装打印机（尽管很少安装）。

其次，可以把用户和资源分配到不同管理员的域中，并且在域之间实行单向或者双向信任。在一家典型的大公司里，你可以把所有用户放到人事部门管辖的一个域里，而把资源，诸如服务器和打印机，放到部门控制的资源域中；个人工作站甚至可以由其用户来管理。一般是这样安排的：部门资源域信任用户域，但是不能反过来——这样腐败或者粗心的部门管理员就不能在自己所属域以外做坏事。接下来，个人工作站信任部门（但是不能反过来），这样用户可以在本地权限下执行任务（安装很多软件包都需要这个权限）。管理员拥有至高权限（所以无法创建真正抗干扰的审计追踪，除非使用一次性写存储设备），但是通过合理的组织，他们能做出的损害是有限的。用来管理所有此类问题并在用户界面上隐藏 ACL 细节的数据结构，称为“注册表”。

在超大规模的机构中设计 NT 体系结构的问题有：命名问题（将在后面探讨），当事人数目增加（恶意地）时域扩展的方式，还要限制其他域的用户不能成为管理员（否则会在本地组和全局组之间产生复杂的交互作用）。

NT 有个特性就是“everyone”是个当事人，这不是默认的控制，也不是失去控制，因此“remove everyone”表示防止一个文件被所有人访问。把“everyone”设置为“no access”，可以迅速将资源锁定。这样，很自然地想到了权能。

4.2.5 权能

管理访问控制矩阵的另外一种方法是按行存储。这称为权能。在图 4-2 的例子中，Bob 的权能如图 4-5 所示。

用户	操作系统	账户程序	会计数据	审计追踪
Bob	rx	r	r	r

图 4-5　权能

对于 ACL 来说，权能的强弱或多或少是与之对立的。运行时安全检查更加有效了，我们可以毫不费事地如此授权：Bob 可以创建一个证明，说“这是我的权能，我从上午 9 点到下午 1 点给 David 授予读文件 4 的权限；签名 Bob”。但另一方面，改变文件的状态突然变得更加棘手了，因为很难找出哪个用户有访问权限。当调查一次事故或者为一个案件准备证据时，会很费力。

在 20 世纪 70 年代很多次的试验都实现了权能，看起来很像文件密码；用户会得到难以猜测的位串，用于读、写还有其他授予的权能。结果发现这种方式能够给出综合的保护[804]。另外也很容易发现，操作系统内几乎全部程序都可以在用户模式下运行，而不是以超级用户出现，所以操作系统的 bug 对安全的要求并不严格（实际上，很多操作系统的 bug 都违背了安全原则，这样使得调试操作系统比较容易）。

IBM AS/400 系列的系统采用了基于权能的保护，并获得了商业上的成功。现在权能正以公钥证书的形式复苏着。本书将在第 5 章讲述公钥加密机制，并在 19.5 节给出基于证书的

系统的具体细节，比如 SSL/TLS。在这里，将把公钥证书看作是经过官方签名的凭证，它宣布特定密钥持有者可以是一个特定的人、某个集团的一员或者特权持有者。

为了阐述基于证书的权能的用处，这里有一个医院的例子。如果我们实现了一个规则“护士可以访问她病房里的所有病人，或者最近 90 天以内看护过的病人的记录”，这样，对于病人记录系统中的每次访问控制决策都需要引用管理系统，来查找哪位护士和哪位病人当时是在哪个病房。这表示，管理系统失败了就会比以前更加直接地危及病人的安全，这显然不是一件好事。事情可以简化为：给护士发一些证书，赋予权力让她们去访问当前病房的有关文件。在我所在大学的医院里，已经开始使用这样一个系统。

有一点需要记住的就是，由于公钥证书往往被认为是“加密技术”而不是“访问控制”，它们实现访问控制策略和结构通常考虑得不够周全。人们必须从 20 世纪 70 年代的权能系统上渐渐得出经验教训（这是一条艰难的路程）。总之，加密技术和访问控制之间的界限是一条断层线，人们容易在这里出错。专家们总是来自于不同的背景，而产品则来源于不同的供应商。

4.2.6 Windows 2000 增加的新特性

从主机访问控制产品到研究性系统，很多系统都合并了 ACL 和权能两种方法，试图达到最佳效果。但是权能最重要的应用是 Win2K 系统。

Win2K 用两种方法增加权能，可以覆盖或者弥补 NT4 中的 ACL。首先，通过配置文件的方法，把用户或组放入优良者名单或者黑名单中（在 NT4 中也可能有有限的黑名单）。安全策略是由组制定的，而不是整个系统使用一个策略。组倾向于集中配置管理和控制（组策略覆盖了单个配置文件）的基本方法。组策略与站点、域和组成单元等有关系，所以它能够靠命名对付一些复杂的问题。策略可以通过标准工具创建，也可以通过编码定义（微软已经宣布，将在一个标准规划中透露组策略数据）。组是在活动目录中定义的，活动目录是一个面向对象的数据库，它在一个分层名称空间的域内把用户、组、机器和组成单元组织起来，给它们做索引，这样可以搜索到任何属性。还有针对个体资源的细粒度（finer-grained）访问控制列表。

本书已经提到，Win2K 用 Kerberos 作为网络用户认证的主要方法[⊖]。它封装在安全支持提供者接口（Security Support Provider Interface，SSPI）里，让管理者可以插入其他认证设备。由此可以得到 Win2K 增加权能的第二种方法：在很多应用程序中，人们多选用 SSL/TLS 公钥协议，它在网络上应用非常广泛，它的基础是公钥证书。这些证书的管理在活动目录范围之外提供了一个面向权能的访问控制层（本书将在 19.5 节讨论 SSL/TLS）。

但 Win2K 的这些方式有各种向下兼容（backward-compatibility）问题。例如，Win2K 中带有完整加密认证的高安全配置与 NT4 不兼容。这是因为活动目录可以和 NT4 的注册表并存，但是注册表无法读它。因此，直到所有重要应用程序已经迁移到 Win2K 上，Win2K 的高安全特征才能在大机构中部署。

⊖ 实际上，Win2K 中的 Kerberos 所有权是不同的，随着证书格式的不同，它将阻止 Win2K 客户使用 Unix 的 Kerberos 基础结构。有关格式不同的文档只有在不用于兼容性实现时才公布。Microsoft 的目标是让所有人都安装 Win2K 的 Kerberos 服务器。这在当今开放系统社区中引起一片抗议声。

Win2K 提供了比市场中以前出售的任何系统都要丰富和灵活的访问控制工具。但是它也有设计局限。首先实现与组有不同要求的角色，在一些应用程序中会很棘手；SSL 证书显然可以做到这一点，但是需要一个外部管理基础设施。其次，Windows（在大部分版本中）仍然是单用户操作系统，这意味着某一时间只有一个人可以操作 PC 机。这样，如果我想要在 PC 机上使用一个无特权的、无足轻重的用户，来访问一个不可信任的可能带有恶意代码的网站，我就必须退出系统并重新登录，或者使用其他有难度的技术，一般很少有用户愿意费心这样做。因此在浏览网页时，用户并没有享受到操作系统期望中应具有的保护性能。

4.2.7 粒度

现在访问控制系统面临的一个实际问题是粒度。由于操作系统是同文件打交道的，通常文件是访问控制机制能处理的最小对象。举例说明，确保银行顾客在提款机上只能看到自己的收支情况而无法看到别人的，这种粒度属于应用层机制。

但是事情不仅仅如此简单。很多应用程序是用数据库工具建立的，这些工具引发了很多问题，不管是在 MVS 上运行 DB2 还是在 Unix 上运行 Oracle，情况都比较相似。所有的应用数据都捆绑在一个文件中，操作系统要么批准、要么拒绝用户访问该文件。所以，如果你用数据库工具来开发部门账目系统，那么你可能必须在操作系统层管理一个访问机制，而在数据库或者应用层管理另外一个。这导致了很多现实的问题。例如，操作系统和数据库系统的管理可能由不同的部门执行，它们彼此不通信息；IT 部门通常在用户的压力下提交性能粗糙的产品，使各种访问控制系统看似整体工作，但是实际上存在严重的漏洞。

另外一个粒度问题是单点登录。尽管计算机经理竭尽全力地工作，许多大公司还是积累了各种结构的系统，因此用户不得不在这些不同的系统上一次又一次地登录；结果，管理的花销就逐步增加了。很多机构想让每个员工一次登录到网络上的全部机器。一种初步的解决方法就是在 PC 机上添加一个主机菜单，允许登录，并在脚本中隐藏必要的用户 id 和密码。更加复杂的解决方法包括一个单独的安全服务器，所有登录都必须通过它，或者用一张智能卡实现不同系统的多认证协议。工程上很难正确地实现这种方案。不管选择何种方法，最好系统的安全会很容易成为最坏的。

4.2.8 沙盒和携带证据代码

实现访问控制的另一种方法是采用软件沙盒。用户想要运行从网络上下载的代码，作为 applet（Java 小程序）。他们关心的是，这个小程序可能会做一些讨厌的事情，比如把他们的文件列出来并且用邮件发给某个软件公司。

为了解决这个问题，Java 设计者给这些代码提供了一个“沙盒”——一个受限的环境，不让它访问本地硬盘（或者只能临时访问有限的目录），并且只允许与自己的主机进行通信。用一个解释程序来执行代码，就可以满足这些安全目标了，比如用 Java 虚拟机（JVM），它只有有限的访问权限 [346]。Java 还可用于智能卡，但是（至少以目前实现的水平看）JVM 是在卡外部有效的编译器，这引发了怎样把输出代码可靠地装入卡中的问题。

携带证据代码是一种代替的方法。这里，要执行的代码必须持有一个证明，保证它不会做违反本地安全策略的事情。这种方法不像解释程序那样导致速度低下，但是不得不在执行代码之前，信任一个简短的程序，由它去核对下载的程序所提供的证明。在 JVM 上消耗巨

大的管理费用是没有必要的［585］。

对于支持恰当的超级用户级限制的结构来说，这两者都不是太全面的替代方法。

4.2.9 对象请求代理

人们曾经对面向对象的软件开发很感兴趣，因为它有降低软件维护费用的潜在优势。一个对象包括捆绑在一起的代码和数据，只有通过指定的外部可视方法才能访问它。这样就有潜力作出更有力、更灵活的访问控制。现在正在进行很多研究，目标是研发一个与现行操作系统和硬件独立的统一安全接口。

其概念是把安全功能建立在对象请求代理（object request broker，ORB）的基础上，ORB是一种在对象之间通信的软件组件。很多研究集中在通用对象请求代理结构（Common Object Request Broker Architecture，CORBA）上，它是为面向对象系统设计的一种工业标准尝试。此类做法最重要的方面在于，ORB是在保护域中采取的一种控制呼叫的措施。这个方法看起来很有前途，但是仍然处于开发之中（［112］是一本关于CORBA安全的书）。

4.3 硬件保护

大多数访问控制系统不仅要控制用户的行为，还要限制程序的权限。在大部分系统中，用户要么写程序，要么下载并安装程序。程序中可能有漏洞甚至会有恶意代码。

保护问题就是防止一个程序干扰其他程序。限制问题通常定义为，防止程序通过未授权的通道与外部通信。这些定义带来了几种改进。目标是要防止活动干扰，比如内存重写，还有防止一个程序直接读取其他程序的内存。这就是商业操作系统要达到的目的。军事系统还要尽力保护元数据——关于其他数据、主题或者进程的数据——这样一来，举个例子，用户就无法发现其他用户登录到哪个系统或者他们正在运行什么进程。在一些应用中，比如处理人口调查数据，限制措施允许一个程序读取数据，但是除了一些有限的查询结果以外，不允许泄露其他信息，这部分内容将在第7章讲述。

除了使用沙盒技术以外（对于一般程序环境，它的限制太多了），在一个处理器中解决限制问题，意味着至少要防止一个程序覆盖其他程序的代码和数据。系统中一般有共享的内存区域，供交互通信使用；但是必须防止程序受到偶然的或者恶意的修改，它们所访问的内存也要采取类似的保护措施。

这一般意味着硬件访问控制必须和处理器的内存管理功能结合起来。段编址是一种典型的机制。内存是靠两个寄存器寻址的，其中段寄存器指向一个内存段，而地址寄存器指向段内的位置。段寄存器是由操作系统控制的，通常是由基准监控器这个特殊元件控制的，它是连接访问控制机制和硬件的桥梁。

但这种技术具体实现起来比处理器本身还要复杂。早期的IBM主机有一个双态CPU：机器分为授权状态和非授权状态。在第二种状态下，程序被限制在操作系统分配的内存段内。在第一种状态时，程序可以随意改变段寄存器。授权程序是从一个授权库中加载的。

只要有合理的授权库，任何访问控制策略都可以在此基础上实现，但是不会总有效；系统安全依赖于把糟糕的（恶意的或者有漏洞）代码清除出授权库。后来的处理器中提供了更加复杂的硬件机制。在20世纪60年代麻省理工学院开发出对Unix发展有激励作用的Multics操作系统，引入了保护环机制，它表达了特权分极的思想：第0环程序可以访问整个硬盘，

超级用户在第2环运行，用户代码在各种低特权的环里使用［687］。它的特性在某种程度上被后来的处理器采用了，比如在80286以后的Intel生产的主要处理器。

在连接硬件和软件安全机制时，存在很多普遍的问题。例如，经常出现低特权进程需要调用更高级的程序的情况，比如应用代码调用设备驱动程序。需要小心设计这样的机制，否则就会出现安全漏洞。例如，IBM主机操作系统MVS有一个漏洞，如果一个程序正在执行一个普通任务和一个授权任务，它可以使普通任务也得到授权［493］。另外，用参数还是用数值来调用不同特权级别，性能会有很大的差别［687］。

4.3.1　Intel 80x86/Pentium处理器

早期的Intel处理器，比如在早期PC机上使用的8088/8086，系统模式和用户模式之间没有区别，因此就没有任何保护——任何运行的程序都可以控制整个机器。80286增加了受保护的段编址和环，所以那是Intel第一次运行正确的操作系统。80386具有内置的虚拟内存和足够大的内存段（4 Gb），一般把机器看作是32位寻址（flat-address）机器。486和Pentium系列芯片增加了更多性能（超高速缓冲存储器、无序执行和MMX）。Pentium 3增加了一个新的安全特性——处理器序列号。此举导致了隐私提倡者的强烈抗议，他们害怕序列号会用于各种“违法控制”目的，所以将来的Pentium产品显然会废弃这种方法（但是识别一台PC机仍然很简单，因为在磁盘控制器和其他部件中还有很多偷窥程序可以读取到的序列号）。

很多机制都支持保护环。当前的特权级别只能由第0环（内核）进程更改。过程不能直接访问低级环里的对象；但是有一些门控，允许在不同的特权级别执行代码，管理所支持的基础设施，比如给不同特权级别准备多个堆栈段和异常处理（更多信息详见［404］）。

Pentium的后继体系结构，如IA-64，在本书编写的时候还没有问世。根据最新信息，它的内存管理基于把每个进程的虚拟地址空间分割成几个区域，由区域的识别器来识别所属进程的转换集合，并提供一个惟一的中间虚拟地址。这是为了避免高速缓冲存储器和翻译后备缓冲器的冲突问题。区域还在进程之间提供有效的共享区。像Pentium IA-64就有四个保护环［382］。

4.3.2　ARM处理器

ARM是32位处理器的核心，最常见的是将许可发放给嵌入式系统的第三方厂商。最初的ARM（支持Acorn Risc Machine）是最早的商用RISC设计。其现在的后继产品非常重要，因为它们已经在各种安全敏感的应用中形成一个统一体系，从移动电话到美国政府用来保护秘密数据的Capstone芯片。快速的乘法和累加指令，还有耗电量低，使得ARM在公钥加密和信号处理的嵌入式应用中具有强大的吸引力（标准参考［325］）。

ARM是作为处理器的核心许可的，芯片设计者可以把它放入到自己的产品中，加上很多可选的附加软件。基本的核心包括给用户和系统进程使用的单独的寄存器单元，还有软件中断机制，它可以把处理器从超级用户模式转换控制，跳到某个固定地址的进程。核心不包括内存管理，所以基于ARM的设计可以把硬件保护严格按照用户的需求来定做。系统控制协处理器可以协助实现这点。它支持具有相似的访问权（因而共用同样的转换表）但是彼此保护的进程域。这样可以快速执行上下文切换程序。ARM CPU芯片的标准产品，在模型600

以后，具有这种内置的内存支持。

Amulet 版本使用自定时逻辑。取消时钟以后，节省了能量并减少了 RF 干扰，但是却需要在主处理器上引入硬件保护措施，比如寄存器锁定，这样可以管理不同硬件进程之间的竞争。这是在主流处理器设计中正在重新使用的操作系统中保护技术的一个有趣的例子。

4.3.3 安全处理器

有些现代智能卡是基于 ARM 处理器的，前面的特性描述同样适用（尽管其内存的局限性意味着只能采取基本的硬件保护）。但是大部分正在使用的微处理器智能卡只有 8 位处理器。其中有些带有内存管理例程，当在前几次指令中向寄存器中输入密码以后，管理例程使某些地址变为只读的。这样做的目标就是，与卡有利害关系的各类对象——也许是卡制造商、OEM、网络或者银行——可能都把秘密存储到了卡上，但是能够互相保护。这可能是软件的事情；但是有些卡使用小型硬件访问控制矩阵来执行这种保护。

还有其他各种针对加密技术和访问控制的专门硬件安全支持。在银行用来处理 ATM PIN 的加密设备中，有些具有授权状态特征，要打印 PIN 时必须设置这种状态（通过两个控制台密码或者一个物理密钥）。这样，当进行这种工作时，可以允许变换超级用户控制。军事上也采用类似的设备分配密钥。本书将在第 14 章“物理防篡改”中详细讨论加密处理器。

4.3.4 其他处理器

在 20 世纪 70 年代，有些研究系统用硬件实现了非常广泛的安全检查，从 Multics 到各种权能系统。有些系统具有一个警戒地址（fence address），是硬件中的一个边界，低于这个值时只有操作系统有访问权。最近很多人研究服务质量（QoS）问题，寻找一种能够保证进程不会过多占用 CPU 从而导致其他进程阻塞的方法。这样的机制已经开始在商业中引入（微软已经允诺把“服务质量技术”用于“Win2K 时间帧”中）。这个特性与访问控制和保护之间的交互是将来需要关注的事情之一。

4.4 哪里出了问题

通用的操作系统是很庞大而且复杂的，比如 Unix/Linux 和 Windows，所以会有很多的 bug。它们在非常广泛的系统中使用，因此每天都被成百万的用户在各种多变的环境中测试其性能。结果，发现了很多 bug 并且报告出来。感谢网络，让知识广泛而迅速地传播。这样，在任何时候，网络上都可能传播着几十种已知的安全缺陷，并散布着其攻击脚本。直到最近，这个问题才被限制了。银行业主机使用不太普及的操作系统，而军事上使用定制“多级安全”操作系统，局外人根本不可用。但现今，这两种行业因为成本的缘故又不得不采用通用操作系统，所以公开攻击脚本可能会潜在破坏一大批系统。

通常攻击者的目标是在系统中获得一个正式账户，然后成为系统管理员，以便完全地掌控系统。操作系统中令人吃惊的众多 bug 能使用户转为根用户。这些缺陷可以按照很多方式分类，比如按照编程错误的类型、按照引入开发进程的阶段或者按照系统中哪一层出错等[493]。失败可能不是由技术实现引起的，而是在高级设计中产生的。用户界面可能会误导人们错误地管理访问权限，或者做出其他愚蠢的事情导致访问控制不起作用（例子见 4.4.3 节）。

总之，我们在系统中建立的保护机制越高级，系统就会越复杂，从而更加依赖于其他软件，也就更趋向于给人类带来错误，因此，可能变得更加不可靠。

4.4.1 击毁堆栈

在计算机应急响应小组（Computer Emergency Response Team，CERT）公告和安全邮件列表里给出的针对操作系统的技术攻击报告中，即使不是大多数的话，也有很多涉及到内存重写攻击，通俗地讲，就是击毁堆栈（见图 4-6）。

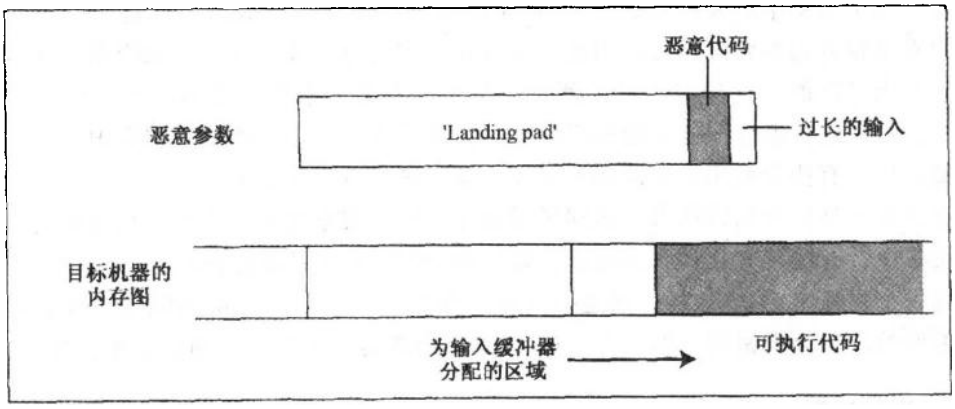


图 4-6　击毁堆栈攻击

程序员经常不注意检查参数的大小。Unix 的 finger 命令就是一个经典的脆弱性例子。实现这条命令时一般接受任何大小的参数，而程序只给这个参数分配了 256 字节。结果，攻击者在命令中使用多于 256 字节的参数，它的尾部字节就无法被 CPU 处理了。

常见的攻击技术有：给参数的尾部字节安排一个“着陆垫”、长空间的 no-operation（NOP）命令或者其他不改变控制流的注册表命令，目的就是捕捉处理器是否执行了其中任何一种操作。着陆垫向处理器传递攻击代码，比如创建一个根账户但是没有密码，或者直接用管理特权启动一个命令解释程序（shell），诸如此类。

CERT 和 bugtraq 日常汇报的很多攻击都是由这类方案变化而来。实在没有任何借口让问题继续存在下去，因为一代人都已经了解了它。在 20 世纪 60 年代早期，大部分分时系统都遭受到了这种攻击，并且修补了这类漏洞［349］。在 20 世纪 70 年代早期，系统发展公司（System Development Corporation，SDC）的深入分析表明，“异常参数”问题仍然是一种频繁使用的攻击策略［503］。1982 年，Intel 的 80286 处理器引入了显式参数核对指令——校验读、写和长度——但是为了防止体系结构依赖性，大部分软件设计者都省略了这类指令的使用。1988 年，大量 Unix 计算机同时被“因特网蠕虫”攻击以致崩溃了，它使用了刚刚提到的 finger 脆弱性，于是使大众媒体意识到了内存重写攻击［724］。然而程序员仍然不校验参数的大小，人们继续发现了很多漏洞。攻击甚至不仅仅局限于联网的计算机系统：发送比程序员预料中更长的消息至少可以击毁一张智能卡。

近来的调查报告把内存重写攻击描述成“时代的攻击”［207］。

4.4.2 其他攻击技术

在内存攻击以后，接下来就要讲解竞争条件。这里所攻击的事务一般分为两个或者更多

个阶段进行，某人有可能在验证访问权阶段之后更改事务。

例如，用来创建一个目录的 Unix 命令 mkdir，要分为两个步骤进行：分配存储空间，然后把拥有者身份交给用户。由于两个步骤是分离的，用户可以在后台初始化一个 mkdir；如果在中止之前只完成了第一步，可以用第二步把一个新建立的目录替换为密码文件的链接。然后初始化进程会继续，并把密码文件的拥有者身份交给用户。目录/tmp 用来放临时文件，也常常被这样滥用；窍门就是，等待一个特权用户运行程序在这里写文件，然后把它改成别处其他文件的符号链接——当特权用户的应用程序试图删除临时文件时，链接就会被清除。

还有其他各种 bug 能够让用户伪装成根用户状态并控制系统。比如，PDP-10 TENEX 操作系统有一个 bug，程序地址溢出时会写到程序状态字的下一位，而该位是特权模式位；这意味着程序溢出会使其处于超级用户状态。还有另外一个例子，有些 Unix 系统实现时有一个特性，如果一个用户在打开最多数目的文件时，执行 su 指令，su 不能打开密码文件，给用户以根用户状态作为响应。

还有很多 bug 允许拒绝服务攻击。例如，Multics 对当时打开的文件数目有全局限制，但是没有本地限制。用户可以耗尽本地资源并且锁定系统，这样甚至管理员都无法登录[493]。直到 20 世纪 90 年代后期，大部分因特网协议实现分配固定数量的缓冲空间，来处理 TCP/IP 连接初始化用到的 SYN 包。其结果导致了 SYN 洪流攻击（SYN flooding attack）。攻击者通过发送大量 SYN 包，可以耗尽可用的缓冲空间，阻止机器接受任何新连接。现在使用 syncookies 可以解决这个问题，见第二部分第 18 章。

4.4.3 用户界面失败

特洛伊木马是最早出现的攻击之一，这个程序诱使管理员运行它，如果运行了就会产生不良后果。人们设计游戏时经常检查玩家是否为系统管理员，如果是，就用一个已知的密码建立另外一个管理员账户。

还有另外一个恶作剧，就是设计一个与系统中常用的实用程序具有同样名字的程序，比如 ls 命令可以列出一个 Unix 目录下的所有文件，设计名为 ls 的程序就可以在调用真正的实用程序之前滥用管理员特权（如果有特权的话）。下一步就是向管理员投诉，说这个目录出了错误。当管理员进入目录并且键入 ls 查看有哪些文件时，系统就被攻击了。解决方法很简单：让管理员的路径（PATH）变量（一个目录列表，当调用命令时用来搜索相应名字的程序）中不包含“.”（代表当前目录的符号）。Unix 的近期版本默认这个功能；但是对于毫无戒备的人来说，它仍然是一个不必要的陷阱。

或许从遭受危险的系统数目这个角度来看，用户界面失败最严重的例子就是 Windows NT。在这个操作系统中，用户必须是系统管理员才可以安装任何东西。这可能是十分有用的，作为一个配置选项，可以防止银行部门的员工在午饭时间用 PC 机运行电子游戏并且带来病毒。然而，大部分环境控制得没有这样严格，因为人们需要安装软件才能完成工作。实际上，成百上千的人们拥有管理员特权但是并不是必需的，这样很容易遭受攻击，如恶意代码简单地弹出一个对话框告诉他们去做一些事情。微软对这个问题的答复就是前面提到过的单向信任机制，它可以配置系统，让人们管理自己的机器而又不具备太多的权力去危害公司的其他 IT 资源。然而，需要小心谨慎才能正确地实现它。另外它在有些应用中并不能提供保护，比如像网络服务器这样的应用程序，必须以根用户身份运行，对外界是可视

的，并且它所包含的一些软件 bug 使得其有可能被攻击者控制。

另外举一个有可能产生争议的用户界面失败的例子，是在使用各种活动上下文（比如 ActiveX 控件）时产生的。这是一种威胁，因为用户没有清楚直观的方法去控制它们，它们会引发严重的攻击。即使是 Java，它的一切假设的安全性能因为其不谨慎的实现而遭受到了很多攻击 [226]。但是，很多人（还有公司）不愿意放弃活动上下文所能提供的种种好处。

4.4.4 为何这么多地方出现错误

我们已经提到了操作系统安全设计所面临的基本问题：它们的产品种类众多因此容易有 bug，同时受到大批用户的并行测试，其中有些人会公开他们的发现却不向产品供应商汇报。另外还存在结构上的问题。

导致系统安全失败的最严重的原因是“内核膨胀”。在 Unix 环境下，所有的设备驱动程序、文件系统等等都必须位于内核当中。Windows 2000 内核包括大量智能卡、读卡机和类似设备的驱动程序，其中很多程序是由设备供应商编写的。因此大量代码得到了信任，被放入安全环境里。软件公司让这么多供应商闯入系统中，不管是故意的还是偶然的，实际上都不是一个好主意。有些其他系统，比如 MVS，引入了降低实用程序所需的信任级别的机制。然而，在大部分常用的操作系统中，实现它的方法极其少而且相对不标准。

甚至还有更加严重的情况存在，应用程序开发人员往往把程序作为根用户运行。这样做可能很简单，因为它避免了许可问题。在低级特权里只能使用有限的想法享受最少的设计，但这样做的结果引发了可怕的攻击。有很多系统——比如 Unix 的行式打印机子系统 lpr/lpd——并不需要以根用户运行，但是在大部分系统中却以根用户来运行它。这也是过去安全失败的一个来源（举例来说，让打印机假脱机操作打印密码文件）。

一些应用程序需要一定级别的特权。例如，邮件投递代理（MTA）必须有权处理用户的邮箱。但是一位谨慎的设计者会把这个特权限制在应用程序的一小部分，而大多数此类代理程序却被写成整个程序都需要以根用户运行。sendmail 是一个经典的例子，它严重的安全漏洞已经有很长的历史了；但是很多其他 MTA 也有问题。一般的影响是，只威胁一个人的邮件的漏洞也许会向外部攻击者提供根用户特权。

有时候药品几乎和疾病一样糟糕。一些程序员为了避免根用户膨胀（root bloat）以及安装非根用户软件并且同时能够安全工作的困难，采取了所有用户都可以访问重要共享数据结构和资源的方法。很多系统把邮件存储在一个全局可写的目录下，每个用户有一个文件，这样伪造邮件是一件非常简单的事情。Unix 文件 utmp——用户登录列表——在各种安全校验中频繁地被使用，但是竟然也是全局可写的！这本来应该以一种服务的形式建立，而不应该是文件，但是一旦初始设计决策确定以后，提供修补问题的解决办法是很困难的。

4.4.5 补救措施

有几类攻击可以使用自动工具来解决。例如堆栈重写攻击，很大程度上是由于 C 语言中缺乏正确的边界校验而导致的（C 语言是编写操作系统的最常用语言）。网络上有各种在 C 程序中检查潜在问题的工具；甚至还有一个叫作 StackGuard 的编译补丁，它在堆栈里靠近返回地址的地方显示一个淡黄色标识（canary）。这可能是程序启动时随机选择的 32 位数值，当函数运行结束时提供检查。如果堆栈在此时被重复写入，那么淡黄色的高亮显示极有可能

会改变 [207]。

但是，通常需要在设计、编码和测试时投入更多的努力。设计者不应该使用太多权限过强的工具，比如在 Unix 中使用 setuid，在 NT 中使用管理员特权，而是应该创建一些具有有限权力的组，并且清楚这个组对系统其余部分的威胁。程序应该只具备那些必要的权利：最少特权定律 [662]。

设计软件时应该保证其默认配置，还有实现某件事情的最简单方法一般是安全的。但是很多系统发布时带有危险的默认配置。

最后，有一个应该关注的相反的观点，正如微软某些高级人员所认为的：访问控制并不重要。计算机正在变为单用途或者单用户的设备，比如传递单个服务的 Web 服务器，不需要采取很多访问控制的方式，因为那样的话操作系统访问控制就无事可做了；把用户彼此区分开来的工作最好留给应用程序代码来实现。至于你桌子上的 PC 机，如果上面所有的软件都来自于一家公司，那么操作系统就没有必要提供隔离了 [588]。并不是所有的人都同意这个观点：NSA 的观点是另外一个极端，他们极度不信任应用层安全并且着重强调应该采用可信任操作系统机制 [510]。但是不管采用这种方式还是那种方式，值得注意的地方仍在于多大程度上有效利用了现代操作系统中装载的访问控制机制。

4.4.6 环境蠕变

本书已经重复地指出了，很多安全失败都是由于环境变化从而破坏了安全模型而导致的。在受限制的环境中可以胜任的机制在一个更加宽泛的环境中会经常失败。

访问控制机制也不例外。例如，Unix 系统最初设计为“单用户 Multics”（因此得名），然后成为一家实验室里有技术的、可信任的人共享一台机器时使用的操作系统。在这个环境里，安全机制的最初功能就是允许错误、防止用户键入错误或者防止因为删除文件或重复写入其他用户的文件而导致的程序崩溃。在这个前提下，原始的安全机制是可以胜任的。

但是 Unix 安全变成了经典的“成功灾难”。Unix 被反复地扩展，而没有正确地考虑到保护机制也需要扩展。Berkeley 扩展（rsh、rhosts 等等）的基础是从一台机器扩展到同一局域网（LAN）内并且处在同一种管理制度下的一系列机器。像 rhosts 这样的机制建立在一个数组（用户名，主机名）的基础之上，而不仅仅是一个用户名，可以看出信任的概念被开始传输开来。

在 20 世纪 70 年代从 Arpanet 发展而来的因特网机制（telnet、ftp、DNS、SMTP）是面向原始的安全的广域网内的主机。这些主机是匿名的，网络中没有安全协议，也没有传送任何认证。远程认证，Berkeley 模型想要使它变得更谨慎一些，但是不被因特网机制支持。Sun 公司的贡献（NFS、NIS、RPC 等等）是基于一个通用的工作站模型，环境中有多个 LAN 并实行分布式管理，但是通常只是在一个单一的机构内（关于 DNS 和 NFS 主题的指南超出了本书的范围，但是在第 18 章“网络攻击与防御”的 18.2 节中介绍了更加详细的背景资料）。

把所有这些不同的计算模型混淆在一起会导致用户晕头转向。它们的原始假设有一部分仍然适用，但是没有一个模型可以应对现在的情况。现在因特网上有上亿台 PC 机和工作站，上百万个 LAN，上千个互连的 WAN，还有管理设施，它们不仅互相独立甚至还可能有冲突（包括彼此竞争的国家级和次级组）。很多工作站根本就没有管理设施。

过去用户是值得信赖的，偶尔也有例外，现在，大量用户都是不可信的，而有些用户既

不可信又敌对我们。过去代码仅仅是存在漏洞，而现在却存在众多的恶意代码。过去对通信网络的攻击只局限于国家情报机构，而现在“脚本小子”（“脚本小子”指从网络上下载攻击工具并使用它们，但是没有任何实际想法的相对来说没有技术的人）都可以做这件事。

Unix 和因特网安全给我们提供了初始设计合理但是被变化的环境击败的例子。

Win2K 及其 NT 系列产品以前版本中的保护机制比 Unix 更加广泛了，但是却用了更短的时间就被攻破了。实际上，我们能说的就是问题仍然存在。

4.5 小结

访问控制机制在一个系统中的多层次上运行，从应用程序到操作系统及硬件。更高级的机制可能会表现更强，但是也倾向于更容易遭受攻击，原因是多种多样的，从内在复杂性到实施者的技术水平。大部分攻击者利用那些有机可乘的漏洞；非常庞大的、广泛使用的或者两者兼备（像操作系统一样）的软件特别容易产生安全漏洞并被人们公开。操作系统也容易受到环境变化的影响，导致破坏了设计时的假设，从而受到攻击。

在计算机操作系统中，访问控制的主要功能是限制特定的组、用户和程序因为错误或者蓄意造成的损害程度。最重要的例子就是 Unix 和 NT，尽管 NT 表现得更强，但它们在很多方面还是相似的。访问控制也是智能卡和其他加密设备的专用硬件设计中的重要部分。人们正在开发新技术来实现面向对象的系统和手机代码。但通常实现起来仍然显得很糟糕。

访问控制中的常见概念，从读、写和执行权限到组和角色，会经常在应用中出现。在一些分布式系统中，它们不会立刻显现出来，因为基本机制可能有很大差别。在公钥基础设施上有一个例子，就是重复实现一个旧的访问控制概念——权能。

研究问题

到 20 世纪 60 年代或者 70 年代早期，大部分访问控制问题都被认识到了，并在试验系统中得到了解决，比如 Multics [687] 和 CAP [804]。从那以后，很多访问控制系统研究就专注于在新环境下重新实现原来的基本方案，比如面向对象的系统和手机代码。

最近的研究思路是怎样把访问控制与允许控制机制联合起来，用于在多媒体操作系统中提供服务质量保证。另外一个课题就是怎样在复杂的大系统中使用角色等技术来有效地实现和管理访问控制。

参考资料

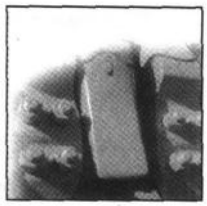
要想得到有关访问控制问题的更加详细的介绍，最好的教材莫过于 Dieter Gollmann 的《Computer Security》[344]。美国海军实验室的一份技术报告对最近 30 年来在操作系统中发现的缺陷给出了有用的参考 [493]。与这个题目（通常就是关于计算机安全）有关的最早报告是由 Willis Ware 写的 [791]。早期最有影响力的论文是 Jerry Saltzer 和 Mike Schroeder 写的 [662]；Butler Lampson 写的有关限制问题的有影响力的论文是 [488]。

Fred Grampp 和 Bob Morris 的论文是对 Unix 安全的经典描述 [350]。这个学科内容最综合的教材是 Simson Garfinkel 和 Gene Spafford 著的《Practical Unix and Internet Security》[331]；Bill Cheswick 和 Steve Bellovin 的《Firewalls and Internet Security》是关于网络的著作 [94]，列举了很多 Unix 系统中的网络攻击实例。

Windows NT4 的保护机制在 Gollmann 的参考书中有简要的描述，更全面的描述见 Karanjit Siyan 的参考书《Windows NT Server 4》[711]。至于 Win2K，我使用了微软的在线文档；毫无疑问即将出现很多教材。在 [79] 中有微处理器体系结构的历史，建筑师 Li Gong 写了一本关于 Java 安全的参考书 [346]。

所有这些课题都发展得很迅速；最受关注的攻击从一年到下一年有了重大的改变（至少是细节上的）。为了紧跟潮流，你不能只阅读教材，还要关注 CERT 最新的公告和诸如 bugtraq 的邮件列表。

第5章 密 码 学



ZHQM ZMGM ZMFM

——凯撒大帝

XYAWO GAOOA GPEMO HPQCW IPNLG RPIXL TXLOA NNYCS YXBOY
MNBIN YOBTY QYNAI

——约翰 F. 肯尼迪

5.1 引言

密码学是安全工程学与数学交叉的地方，它给我们提供了许多建立在现代安全协议之下的工具。对于保护的分布式系统，它可能是一种关键的支撑技术，但要正确地使用它却非常困难。就如在本书第2章“协议”中所看到的，密码学经常保护了错误的目标或者以错误的方式提供了保护，在下面详细地考虑它的现实应用时，还会看到更多这方面的例子。

不幸的是，过去的20年中，计算机安全和密码学研究被隔离开来了。搞安全研究的人通常都不懂那些有用的加密工具；而搞加密的人，也往往不能把它与实际问题联系起来。造成这种情况有很多原因，比如不同的专业背景（计算机科学和数学）和不一样的研究经费（政府总是试图大力发展计算机安全的研究同时又压制密码学的研究）。这让我想起了一个医生朋友给我讲过的故事，她说她年轻时曾经在一个国家工作过几年，因为经济的原因，这个国家缩短了医学专业学习年限，而集中于尽可能快地培养专科医师。一天，一个双肾切除等着移植的病人需要重新做透析，没想到一个外科医生当场就把这位病人从手术室送回家了，因为病人的病历上没有尿样检查，这个医生却没有注意到肾脏被切除的病人是不可能产生尿液的。

就像一个医生既需要懂生理学也需要懂外科学一样，一个计算机安全工程师也应该既熟悉计算机安全知识也熟悉密码学（甚至更多）。本章的内容主要是针对那些没有接受过密码学方面培训的读者的；而密码研究者将会发现这些都是他们早已知道的。由于只有区区几十页的篇幅，而要比较完整地讲述现代密码学至少需要数千页的篇幅，所以，在这里本书不准备花过多的篇幅讲它的数学基础（这方面的书已经有很多了，更深入的研究见章末列出的参考书），而主要解释那些最容易导致混淆的基本知识和结构，如果你像看小说一样经常使用密码学，那么我强烈推荐你多看一些这方面的书籍。

计算机安全研究者常常需要一些密码学术语的非数学定义。基本的术语如*密码学*是指设计密码的科学和技术；*密码破译学*是指破译密码的科学和技术；而*密码研究*通常被简写为crypto，是同时研究密码破译和加密的科学。加密过程的输入信息通常称为明文，加密过程的输出称为密文，这样一来事情有时候就变得比较复杂了，现在已经知道有许多的基本加密方法——包括基本的构造块，比如分组密码、序列密码和哈希函数等。分组密码有可能解密和加密使用同一个密钥，这种情况叫共享密钥，也叫做密钥或对称密钥，当然也有可能是加

密和解密用不同的密钥，这种情况就叫做公钥加密或非对称加密，数字签名方案就是一种特殊的非对称加密方法。

在本章的余下部分，将首先给出历史上的一些简单例子讲述基本概念，然后通过介绍密码学家常常使用的随机预言模型给出一个有条理的清楚描述，最后给出一些实际工作中使用的更重要的加密算法，并讲述它们是怎样用于数据保护的。

5.2 历史背景

根据 Suetonius 的记载，Julius Caesar 通过用 D 代表 A，E 代表 B 等等依次类推的方法来加密他的急信［742］。到 Augustus Caesar 继承皇位时，他改变了皇室遗传下来的加密系统，改用 C 代表 A，D 代表 B，依次类推，在现代密码学术语里，就可以说他把密钥从 D 改到了 C。

Arabs 家族推广了这种思想，形成了单字母替代的加密方法，这种方法是对字母表进行改序得到的密文，如图 5-1 所示，明文用小写字母，密文用大写字母。

```
abcdefghijklmnopqrstuvwxyz
SECURITYABDFGHJKLMNOPQVWXZ
```

图 5-1 单字母替代加密法

OYAN RWSGKFR AN AH RHTFANY MSOYRM OYSH SMSEAC NCMAKO；但是这种密码的解密非常直接，你在小学的时候就可以做到。它的解密原理就是有些字母以及字母的组合比其他的字母（字母组合）要常用的多，在英语中最常用的字母顺序就是 e，t，a，i，o，n，s，h，r，d，l，u。因此可以根据频率分析法进行破译，人工智能研究者对用编程解决单字母替代很有兴趣；他们把字母和连字组合（字母对）出现的频率单独考虑。最后成功地找出了 600 个字母左右的典型密文，当采用更智能的策略时，比如猜测可能的单词这种方法时，可以把密文减少到 150 个字母左右。密码专家通常需要更少字母的密文。

有两种基本的方法可以使密文更加安全（不容易破译）：序列密码和分组密码，对前一种方法而言，你所使用的加密规则依赖于该明文符号在明文符号流中的位置，而在后一种方法中，你可以在一个分组中同时加密好几个明文符号。现在来看看早期的几个例子。

5.2.1 早期序列密码：vigenère 表

一种早期的序列密码通常都是指法兰西人 Blaise de Vigenère 发明的 Vigenère 替代表（它实际上是一种多字母替代加密法），Vigenère 是查理斯（Charles）四世国王时期的一个外交官。这种加密方法是这样进行的：约定 A = 0，B = 1，……，Z = 25，然后把密钥重复地加进明文中，相加后的结果用 26 取模，也就是说如果相加后的结果比 25 大，则减去 26 的若干倍，使得相减后的结果落在［0，…，25］的范围内，也即所得结果仍在［A，…，Z］范围内。写成数学表示的形式就是：

$$C = P + K \bmod 26$$

举一个例子，假设把 P（15）加到 U（20）上得到 35，减去 26 后就是 9，9 对应 J，所以 P 在密钥 U（当然也可以说 U 在密钥 P）下的加密结果就是 J。如果用这种符号表示法，则 Julius Caesar 系统使用的是固定的密钥，即 $K = D$，（注意该系统是模 23，因为 Caesar 系统的字母表常常用 U 代替 V，J 代替 I，且没有字母 W），而 Augustus Caesar 系统使用的密钥是 $K = C$，但 Vigenère 系统使用的是重复密钥，也就是游动密钥。有很多途径可以使它的加法计

算更快，包括使用已经打印好的表格以及野外使用的密码轮等等。但不管用什么方法实现这种技术，用重复密码实现加密的原理都可以用图 5-2 说明。

明文：	tobeornottobethatisthequestion
密钥：	runrunrunrunrunrunrunrunrunrun
密文：	KIOVIEEIGKIOVNURNVJNUVKHVMGZIA

图 5-2　一个 Vigenère 多字母替代加密的例子

很多人都研究过怎么破译多字母替代加密法，包括声名狼藉的登徒子 Casanova 和计算学的先驱 Charles Babbage。但第一个解决方案是由一个叫 Friedrich Kasiski 的普鲁士步兵团军官于1863年发表的[441]，他注意到当给定一个足够长的密文时，将会以密码长度的倍数重复出现一些图案(块密文)。

举个例子，在图 5-2 中，可以看到“KIOV”在 9 个字母之后、“NU”在 6 个字母之后都重复出现了。由于 3 可以被 6 和 9 同时整除，因此可以猜想密钥长度为 3。从而可以推出密文中的第 1，第 4，第 7 等字母是用同一个密钥字母加密的，接下来就可以用频率分析技术猜出这个密钥字母最可能的值。用同样的方法可以对密钥的第 2、第 3 个字母重复这样的过程。

5.2.2　一次一密法

使这种形式的序列密码能够抵抗大多数攻击的一种途径就是使密钥序列的长度和明文一样长而且不重复。这种方法是 Gilbert Vernam 在第一次世界大战期间提出的 [428]，也就是说给定任意一个密文和一个同样长度的明文，都存在一个密钥可以把密文解密成明文，这样无论敌人计算多少次，他们也无法得到正确的消息，因为所有可能的明文看起来都是一样的，这种加密系统就是著名的一次一密法。二战期间，Leo Marks 在特殊任务执行（SOE）部门写过一本密码学方面的书，就是讲间谍怎样把一次密钥写在丝绸上，然后藏在衣服里的，无论什么时候只要一次密钥被使用过，丝绸就会被撕毁或者烧掉 [523]。

下面举一个例子来说明这一切。假设二战时你从一个德国间谍那里截获了一条情报，并且已经知道了它是以“Heil Hitler”开头的，而情报密文的前十个字母是“DGTYI BWPJA”。不难得知该一次一密的前十个字母是“wclnb tdefj”，如图 5-3 所示。

明文：	heilhitler
密钥：	wclnbtdefj
密文：	DGTYIBWPJA

图 5-3　间谍的消息

而一旦间谍烧毁了那片写有密钥的丝绸后，他就可以狡辩说自己实际上是反纳粹地下组织的成员，情报实际是说“绞死希特勒（Hang Hitler)”，这是很有可能的，这样一来，原始密钥也可以很容易地变为“wggsbtdefj”如图 5-4 所示。

密文：	DGTYIBWPJA
密钥：	wggsbtdefj
明文：	hanghitler

图 5-4　间谍狡辩的消息

现在可以看到，从密码学的角度来说，我们得不到任何有价值的东西，一次一密虽然可以做到理想的保密安全性，但它的代价是不能完全保护数据的完整性。比如你也可以把密文改成“DCYTI BWPJA”，这样一来这个间谍就得吃点苦头，如图 5-5 所示。

密文：	DCYTIBWPJA
密钥：	wclnbtdefj
明文：	hanghitler

图 5-5　对图 5-3 消息进行处理以捕获间谍

二战期间，Claude Shannon 证明了一段密文完全保密的充分条件是：对于任意一段给定的明文，都存在和明文一样多的可能密钥，而且每个密钥都是相似的。因此一次一密也就是惟一

能够提供完全保密性的系统 [694, 695]。

一次一密现在仍然被使用在高层外交和智能通信方面，但是由于它使用的密钥和通信量一样多，因此在多数情况下代价太高。对序列密码而言，更普遍的方法是用合适的伪随机数字生成器把短密钥扩展成长密钥序列，然后把数据一次一位的与密钥序列进行异或运算（模二加）来实现加密。使密钥序列在标准的序列随机测试中出现随机性往往还不够，同时还必须使密文具有这样的性质：敌人即使有一串密钥，他也不能根据现有的密钥推测任何其他的密钥。关于这一点本书在下面的一节中将正式讲解。

现在序列密码仍然普遍使用在硬件应用中，因为在这些场合必须尽可能少地使用门电路以节省能量。本书将在随后的章节中讲一些这方面的实际例子，包括用在 GSM 移动电话通信中的 A5 算法（见第 17 章“电信系统的安全”），用在按次付费电视中的多路移位寄存器系统（见第 20 章“版权和隐私保护”）。即使如此，加密过程用软件就可以实现的分组密码方式在很多情况下仍然比序列密码方式要好，所以本书接下来看看分组密码的加密方式。

5.2.3 早期的分组密码：Playfair 方法

早期最著名的分组密码是 Playfair 系统，它是电报的先驱 Charles Wheatstone 先生 1854 年发明的，除此之外，他还发明了一种类似风琴的六角形乐器和惠斯通电桥。这种加密法没有取名叫 Wheatstone 法是因为 Wheatstone 把这种方法演示给了政治家 Playfair 男爵，后者在一次宴会上用餐巾纸把它写给了 Albert 王子和 Palmerston 勋爵（后来的首相）。

这种方法使用了一个 5×5 的方格，里边放置了字母，放置的顺序随密码不同而不同，而且该系统没有使用字母 J（见图 5-6）。

P	A	L	M	E
R	S	T	O	N
B	C	D	F	G
H	I	K	Q	U
V	W	X	Y	Z

图 5-6 Playfair 加密表

加密是这样进行的：首先把明文中出现 J 的地方用 I 代替，接着分成两个一组的字母对，如果有两个同样的字母连在一起，则在中间加“x”把它们分开，在完成最后的字母对时如果需要（即最后只剩下一个字母时）则添加“z”。比如 Playfair 在他的餐巾纸上写的是“Lord Granville's letter”，则分组以后就变成了“lo rd gr an vi lx le sl et te rz”。接着使用下面的规则对其一次两个字母的加密：

- 如果两个字母在同一行或者同一列，则分别由它们后面紧挨着的字母代替，比如，“am”加密为“LE”。
- 如果它们不在同一行或同一列，则它们肯定构成以它们为顶点的矩形，这时就用这个矩形的另外两个定点分别代替它们，比如“lo”加密成“MT”。

根据这两条原则，前面的文本就可以加密成如图 5-7 所示。

```
明文:  lo rd gr an vi lx le sl et te rz
密文:  MT TB BN ES WH TL MP TA LN NL NV
```

图 5-7 一个 Playfair 加密的例子

这种密码的改进形式曾经被一战时期的英国军队和二战时期的美国和德国军队作为野战密码使用过，它们是 Vigenère 加密法的本质提高，这是因为在这些方法中，攻击者所收集到的是统计特性比单字母统计特性更好的连字组合（字母对），它们的分布更加分散，这样攻击者要解密就需要更多的密文。

强调一下，分组密码的输出在直观上具有随机性是远远不够的。比如 Playfair 密文看起来是随机的，但它有这样的性质，即，如果你改变明文串中的一个字母，通常在密文中也只有一个字母会发生改变，因此，用图 5-7 中的密钥，“rd”加密成“TB”，“rf”就加密成了“OB”，同样道理“rg”就加密成了“NB”。可以得出这样一个结论，给定足够多的密文或者合适的单词，密钥表（或者等价的密钥表）就可以被重构出来［326］。而我们期望的结果是：加密系统的输入即使只发生很小的改变，它也会完全影响到输出，即，平均来说，改变输入的一位，输出的一半位都要发生变化，本书在下一节将围绕这种思想进行讨论。

选择长于两个字符的密钥组可以使分组密码的安全性得到显著提高。比如，广泛使用在银行中的数据加密标准（Data Encryption Standard，DES）用的就是一个 64 位的密钥组，它等同于八个 ASCII 字符，而高级数据加密标准（Advanced Encryption Standard，AES）密钥组的长度是 DES 的两倍，在许多场合下，它取代了 DES，本书接下来应该讨论 DES 和 AES 的内部细节，但由于篇幅有限，这里只提一下：8 字节或者 16 字节的密钥组本身是远远不够的。比如，一个银行账户经常在同一个地方进行交易活动，那么有可能在每次进行交易活动的时候，都得到同样的密文，因为它们是用同样的密钥加密的。这就允许一个可以在线窃听的攻击者监视用户的交易活动；同时他还可以通过剪切复制部分密文的办法进行看起来合法但实际非法的交易。只有当密钥分组的长度和消息的长度一样长时，密文才可以包含不止一个密钥分组，后面将会看到几种如何把它们绑定在一起的方法。

5.2.4　单向函数

第三种经典的加密方法就是使用单向函数。单向函数加密可以保护数据的完整性和真实性，而本书以前介绍的几种简单加密方法都不能提供这种保护机制，也就是说，单向函数加密可以通过以某种方式操纵密文使得明文出现可以预测的改变。

在 19 世纪中叶电报发明以后，银行马上成了它的主要用户，并且把它发展成为电子转账系统。当然不是钱通过电线传输过去了，而是支付指令的传输，比如下面的转账指令：

> 伦敦 Lombard 银行：请从我们的账号 1234567890 转给住在 Chesterton 路 456 号的 John Smith £1000，他在剑桥的 HSBC 银行有一个账号：301234 4567890123。并且通知他这是“Doreen Smith 送给他的结婚礼物”。这是由美国 Santa Barbara 的 First Cowboy 银行转来的，转账费用由我们支付。

由于电报消息通过人工操作从一个部门转移到另外一个部门，所以存在操作员操纵支付消息的危险。

因此，银行、电报公司和船舶公司开发了密码本，它不但可以保护客户的交易，而且可以缩短电报的长度——当时国际电报的花费是非常重要的因素。密码本实质上是分组密码，它把单词或短语映射成固定长度的字母或数字组。因此，“请从我们的账户（Please pay from our account with you no.）”就可以映射成“AFVCT”。除此之外，还存在另外一种竞争的技术：转轮机，它的机械加密设备可以产生非常长的伪随机数字序列，再与明文组合就生成了密文。转轮机是由几个人各自独立发明的，他们中的多数都梦想把它卖给银行发财，但银行通常是不会对这类东西感兴趣的，即使如此，转轮机在二战时期仍被军队作为一种高级加密方法采用。

银行意识到不论是机械的序列密码还是密码本都不能保护消息的真实性。举个例子，如果表示1000的密码字是紫红色的而1 000 000是深红色的，那么图谋不轨的电报职员则可以通过与已知交易的编码通信量相比较把1000和1 000 000鉴别出来，并可以用其中的一个代替另外一个，这是相当危险的。

在这种情况下就出现了对密码本关键性的革新，就是通过在代码组里加入一个称作测试码的数字来实现单向编码（现代的加密研究者可能会把它们描述成哈希值或者消息认证码，详细的术语在本书后面定义）。

下面是一个得到测试码的简单例子。假设银行的密码本上有一个数字表格对应着支付的数额，如图5-8所示。为了表示一桩$376 514的交易，银行把53（因为不足百万），54（300 000），29（70 000），和71（6 000）加起来（通常账户上小数额的数字都是忽略的），这样就得到一个测试码207。

	0	1	2	3	4	5	6	7	8	9
x 1000	14	22	40	87	69	93	71	35	06	58
x 10,000	73	38	15	46	91	82	00	29	64	57
x 100,000	95	70	09	54	82	63	21	47	36	18
x 1,000,000	53	77	66	29	40	12	31	05	87	94

图5-8 一个简单的测试码系统

大多数实际系统都比这个复杂，它们的表格包括表示货币种类代码、日期甚至收款人账户。而在更完善的系统中，代码组的长度通常是四个而不是两个，同时为了增加攻击者重建表格的难度，测试码是压缩的，比如，一个7549的测试码可能变成23，这是通过把千位百位数字加起来，十位个位数字加起来并忽略进位得到的。

用现代密码学的标准来看，测试码还不够强大。只要给定数十个或者数百个经历过测试的消息（数目的多少依赖于系统的设计细节），一个恶意的破译者就可以重构足够的表格来篡改交易，这样就可以把一系列精心选择的消息输进银行来实现他的阴谋，如果银行有内应，事情还要容易得多。但银行很侥幸地逃过了惩罚：从19世纪后期直到20世纪80年代，测试码都没有出过问题。多年从事银行安全咨询工作和资深银行审计师餐桌上提供的故事中，我仅仅听到过两起使用测试码的诈骗事件：一个是外部的攻击者，他因为不熟悉银行的操作流程最后失败了；另一个规模较小的诈骗成功了，它牵涉到银行里面的一个职员。在第9章“银行业和簿记系统”中本书将解释取代测试码的更安全的系统，该系统包括了所有与密码认证机制有关的程序保护，如双向控制等。今天，测试码已经成为一个用于认证的单向函数的经典例子。

后面的例子将包括前面几章讨论的实际应用中的函数，如在单向加密口令文件中存储口令，以及在认证协议中处理来自询问的响应等等。

5.2.5 非对称基本加密方法

最后讨论一下现代加密系统中使用的非对称加密方法，说它非对称是因为它使用不同的密钥实现加密和解密。比如，我在我的网页上公布了一个公钥，别人只要用它就可以给我发消息，而只有我用相应的私钥才能够解密这些消息。

在计算机发明以前，也曾有过这样的例子，可能最形象的例子就是邮政服务了。只要写

上我的地址并投进邮筒就可以给我发消息了，这也意味着，只有我才可以阅读它。当然，有很多因素可能使这个过程出错，比如你写错了我的地址（不论是笔误还是故意写错）；警察得到授权打开了我的信；信件被不诚实的邮递员偷走了；一个骗子没有经过我的同意就更改了我的信；或者，一个小偷从我的邮筒里偷走了它，等等。在公钥加密系统中也会发生类似的事情，如输入系统错误的公钥；计算机被黑客攻击了；给我发消息的人可能是被强制的等等。在后面的章节中本书将更详细地讨论这些问题。

另外一种非对称加密应用是数字签名。在这里的含义是：我可以用一个私人签名密钥给信息签名，而别人可以用一个公共签名验证密钥来校验我的信息。同样，在计算机发明以前，也曾经有过模拟签名，使用的是手印或者图章；类似地，不管是对于老式的模拟签名还是新式的数字签名，都有一系列冗长而复杂的因素使签名失效。

5.3　随机预言模型

在深入现代加密方法的设计细节之前，我希望能花几页纸的篇幅来概括不同种类加密方法的定义。（那些对计算机理论知识感到头痛的读者看第一遍的时候可以跳过这一节，本书把它包括进来是因为粗略知道一些随机预言模型，对读者更好地理解最近许多密码学方面的研究论文来说是很有必要的）。

随机预言模型致力于建立这样的思想：如果一种加密方法是“好”的，那么从一个合适的角度考察它时，它不能从一种给定类型的随机函数中分离出来。如果一个加密方法可以通过合适类型的随机函数进行统计和其他测试，而不管使用的计算模型是什么，这种加密方法就称为伪随机加密。很明显，加密实际上是一种算法，要么用门电路组成的硬件实现，要么用软件编程实现。但是，加密以后的输出应该“看起来”是随机的，且不能从合适的随机预言函数中分辨出来，这种随机预言函数是由计算模型允许的测试数目和种类来决定的。

通过这种方法，我们有望从如何正确使用加密方法中把加密设计的问题分离出来。设计加密算法的数学家可以用数学方法证明他们的加密算法是伪随机的。相反，一个设计了加密协议的计算机科学家可能试图证明在实现该协议的加密原语是伪随机的假设下该协议是安全的。而这不见得总是对的，因为在协议正确性的证明中我们就看到了。就像程序一样，定理也可能有缺陷，这可能是过于理想化的错误，也有可能是数学家使用了从计算机科学家那里得到的不同的计算模型所导致的，但不管怎么说，我们还是可以取得一些进展。

图 5-9　随机预言模型

可以把随机预言虚拟成一个坐在黑盒子里的小精灵，这个黑盒子里有一些物理随机性和存储途径（见图 5-9）——在图中它们分别用骰子和卷轴纸表示，这个小精灵接受一个特定类型的输入（查询），然后看看卷轴确认以前是不是曾经回答过这个查询，如果是，则从卷轴纸上找到答案然后回答这个查询，如果不是，则通过掷骰子的方法产生一个随机的答案。可以进一步假设存在一些带宽的限制——也就是说小精灵每秒只能回答限定数量的询问，这种理想化的抽象是很有用的，因为它概括了本书关于序列密码、哈希函数、分组密码、公钥加密算法和数字签名方

案的思想。

最后，注意到加密既要在时间上也要在距离上对数据进行保护，可以使本书关于加密的概念模型得到有益的简化。下面看一个典型的加密系统的例子：在用第三方的备份工具备份数据以前，先对数据进行加密，一旦系统磁盘损坏，我们就能够用备份磁盘对它解密从而得到数据。在这种情况下，仅仅需要一个加密/解密工具，而不是像通信链路那样发送端和接收端都需要一个。这正是本书在这里建模的应用类型：用户把备份磁盘插进加密机器，键入密钥，发出一条指令，随后数据就会以恰当的方式输出。

下面就来看看在不同的加密方法下对这个模型更详细的讨论。

5.3.1 随机函数：哈希函数

本书首先要介绍的随机预言模型就是*随机函数*，随机函数接收长度不等的字母串，随后输出一个固定长度的随机数字串，比如说长度为 n 位。因此，在上面的例子中，也可以说小精灵只有一个关于输入与输出关系的简单列表，当然随着时间的推移，这个列表会稳定地增大（忽略卷轴纸大小的影响并且假设所有的查询都在一个常数时间内被答复）。

随机函数是为了有许多实际应用的*单向函数*或*加密哈希函数*而建立的模型。在 20 世纪 60 年代，随机函数就首先用在计算机系统中作为密码的单向加密，并且——就像第 2 章提到的那样——今天它们仍然用在一系列的认证系统中，它们还用来计算*消息摘要*：给定一条消息 M，使它通过一个伪随机函数得到该消息的一个摘要，记为 $h(M)$。在许多情况下，$h(M)$ 可以代表消息本身，一个例子就是数字签名：如果消息太长的话，签名算法运行时间就会变长，所以通常给一个摘要签名要比给消息本身签名方便得多。

另外一个例子就是时间戳。假设我们想证明在一个确定的日期之前，某一个给定的电子文档是我们所拥有的，可能需要把它提交给在线时间戳服务公司，然而，如果这个电子文档目前来说仍然需要保密——比如说准备申请专利的发明，在这种情况下，该电子文档就只能在以后的某个日期公布——那么就不能把整个文档提交给时间戳服务公司，而只能是它的消息摘要。

哈希函数的输出就是众所周知的*哈希值*或*消息摘要*，对应一个给定哈希值的输入就是它的*原像*，动词*哈希*是指哈希值的计算，有时候*哈希*也是一个名词指代哈希值。

5.3.1.1 随机函数的性质

随机函数的第一个主要性质就是单向性。给定一个输入值 x，可以很容易计算它的哈希值 $h(x)$，但是如果给定一个哈希值 $h(x)$ 而事先不知道 x 的话是很难找到它的原像的（例子中的小精灵也只是对一定的输入给出对应的输出，而不是做相反的事情）。由于输出是随机的，一个想对随机函数求逆的攻击者只能不停地输入直到发生奇迹。伪随机函数也有这样的性质；或者与我们的定义相反，这个性质可以从随机函数中分离出伪随机函数。另外，伪随机函数也是一个单向函数，即使随机函数有足够多可能的输出，攻击者也无法偶然找到期望的目标输出。这意味着选择一个 n 位的输出，攻击者必须尝试 2^n 次才能找到两个有相同值的随机消息。

伪随机函数的第二个性质就是其输出不能给输入提供任何信息。因此，输入值 x 的单向加密就是把 x 与密钥 k 联系在一起计算 $h(x, k)$ 的值。如果哈希函数的随机性不够强，那么在单向加密中使用这样的形式无异于自找麻烦。下面是一个在 GSM 移动电话认证中的

例子。在这个系统中，来自基站的16字节查询信息和手机用户的16字节密钥组合起来形成一个32字节的消息串，然后经过一个哈希函数得到一个11字节的输出[138]。它是这样设计的：电话公司也知道用户的密钥 k，从而当无线链路上的窃听者得到随机查询的 x 值和由 $h(x, k)$ 计算出的相应输出值时可以对计算进行校验，这就要求窃听者不能得到关于 k 的任何信息并且对任何新输入的值 y 都不能计算出 $h(y, k)$ 的值。但是大多数电话公司使用的单向函数单向性不够，这样的后果是：窃听者假装基站，给用户发送大概150 000条合适的查询并得到用户的响应后就可以计算出用户的密钥 k。关于这个失败的例子，本书在第17章的17.3.3节中将会详细讨论。

伪随机函数有很长的输出，因此它的第三个性质就是很难发生碰撞，也就是说当 $h(M_1) = h(M_2)$ 时，它们对应的消息不会相同，即 $M_1 \neq M_2$。除非攻击者能够找到一个快捷的攻击方法（这也意味着加密函数不是真正的伪随机），那么找到碰撞的最好办法就是收集一大堆消息 M_i 和对应的哈希值 $h(M_i)$，对它们分类后寻找匹配的一对。如果哈希函数的输出是 n 位的数字序列，则有 2^n 个可能的哈希值，那么攻击者在找到一个匹配的哈希对之前需要计算的哈希值大概是上述数字的平方根，即 $2^{n/2}$ 个，这个事实在安全工程中是非常重要的，下面将更加密切地关注这一点。

5.3.1.2　生日定理

生日定理，起初称为重捕获统计，20世纪30年代为了对捕获的鱼进行计数而发明的。它假设湖里有 N 条鱼，你捕获了其中的 m 条，做上记号以后又把它们放生了，那么当你再一次抓到你曾经做过记号的鱼时，你需要捕获的鱼的数目大概是 N 的平方根。这个现象的直观解释就是一旦你拥有了$\sqrt{N}$个样本以后，每个样本都有可能与其他的任何一个匹配，所以可能的配对数大概就是$\sqrt{N} \times \sqrt{N}$也就是 N，这正是所需要的捕获数。[⊖]

对安全工程师来说，生日定理有很多应用。比如有一个生物统计系统，这个系统可以对某个人进行身份验证，这种验证的正确率是很高的，因为随机选择的两个个体被错误地认为是同一个的概率只有百万分之一，但是这并不意味着可以把它作为一个可信的鉴别方法用在一个有两万教职工和学生的大学中。这是因为将会有差不多两亿个可能的配对，实际上，你还是可以期望找到第一个冲突——被系统搞混淆的两个人——如果有稍微超过一千人登记注册的话。

在一些实际应用中碰撞搜索攻击不会成为一个问题，比如在查询响应协议中，攻击者必须要对系统提交的查询进行响应，而在这种情况下你又可以避免重复查询（举个例子，查询并不见得是真正随机的而是由加密一个计数器得到的），那么攻击者就不能通过搜索查询进行攻击。比如，在敌我识别（identify-friend-or-foe，IFF）系统中，普通设备都有一个长度为48～80位的响应。

然而，在其他的一些应用中，碰撞是不可接受的。比如在数字签名中，如果能够找到一个碰撞使得 $h(M_1) = h(M_2)$ 但 $M_1 \neq M_2$，那么一个黑手党拥有的书店网站就会给你发送一条消息 M_1 要你签名，消息可能是这样说的："我要预定《橡胶之谜（Rubber Fetish）》这本

⊖ 准确地讲，从 N 条鱼中随机选择 m 条不同的鱼的概率是 $\beta = N(N-1)\cdots(N-m+1)/N^m$，这个结果是由 $N \approx m^2/2log(1/\beta)$ 近似得到的[451]。

书的第7卷并愿意支付$32.95”。当你签名发送出去以后，可能会出现另外一条被你签过名的消息 M_2，消息内容可能是：“我愿意把我的房子抵押为$75 000，并保证现金可以支付给百慕大黑手党财产有限公司”。

正因为如此，用在数字签名方案上的哈希函数一般都让 n 足够大使得它们之间没有碰撞，也就是对攻击者来说要计算 $2^{n/2}$ 次是不切实际的。看看两个最常用的例子，一个是 MD5，它有 128 位输出，因此攻击者大概需要计算 2^{64} 次才能破译；另外一个是 SHA1，它有 160 位的输出，密码分析者大概需要进行 2^{80} 次因数破解才能解密。但至少 MD5 看起来是易受攻击的：早在 1994 年就有人公布了一个设计，它在一台耗资 1000 万美元的机器上只需要运行 24 天就可以发现碰撞。不久 SHA1 也被发现是易受攻击的。因此美国国家标准技术协会（National Institute of Standards and Technology，NIST）近年引进了范围更广的哈希函数——SHA256，它有 256 位输出，还有 SHA512，它有 512 位输出，这样就消除了加密上的快捷攻击——也就是说，攻击者要计算的次数无异于强制搜索——要找到一个碰撞，分别要进行 2^{128} 次和 2^{256} 次尝试，这在一代或二代摩尔定律计算机上计算是无效的。通常，一个谨慎的设计者总是使用尽可能长的哈希函数，在新系统中应该尽量避免使用 MD 系列的哈希函数（MD4 是 MD5 的前身，它由于存在碰撞和容易求逆而被公认为加密性能不好）。

因此，伪随机函数有时也常常被认为没有碰撞或者极难消除的碰撞，这并不意味着它们不存在，就像你从来就不可能找到它们一样——当可能的输入数比可能的输出数大时，它们一定存在。通常意义上没有碰撞是建立在输出足够长的假设上。

5.3.2 随机序列生成器：序列密码

第二种基本加密方法是随机序列生成器，也叫密钥序列生成器或序列密码。它同样也是一个随机函数，但不像哈希函数，它的输入短但输出长（如果有一个输入输出都有十亿位的随机性能非常好的伪随机函数，而且现实生活中绝对不可能处理比这还长的目标，那么扔掉其他而只留下数百位的输出就可以得到一个哈希函数，把除数百位的常数输入外全部填充一个常数时就可以得到序列密码）。从概念上说，把序列密码看作一个输入长度固定、输出非常长的比特流（即密钥序列）的随机预言模型是很普遍的，它用在保护备份数据的保密性方面非常简单：向密钥序列生成器里输入一个密钥就会得到一个随机数字组成的长文件，再把它与明文数据进行异或运算（模二加），就得到了密文，随后就可以把密文送到备份的管理者那里了。如果用前面的预言例子来描述，可以想像小精灵每次得到输入的新密钥时，就会产生一个必需长度的随机数字串，它把这个数字串给我们，同时把副本保留在卷轴纸上以便下次有相同的输入时做参考。当需要恢复数据时，就可以回到生成器，键入相同的密钥，得到同样长度随机数字的长文件，接着把它与密文异或（模二加）从而再次得到所需的明文。除非知道密钥，其他人即使访问密钥序列生成器也不会得到正确的密钥序列。

本书在前面提到过一次一密法和香农（Shannon）关于加密证明的结论，即密文完全保密的充分条件是：对于任意一段给定的明文，都存在和明文一样多的可能密钥，而且每个密钥都是相似的。这种安全性称作无条件的（或统计学的）安全性，因为它既不依赖攻击者可以使用的计算能力也不依赖是否有先进的数学理论可以对密码提供快捷攻击。

一次一密系统与我们的理论模型相当贴近，只可惜它主要用在空间的安全通信上而不是时间上：通信双方的参加者提前共享一个随机产生的密钥序列副本。比如 Vernam 的原始电

报加密机器曾经使用打孔磁带纸，这个系统中通信双方提前产生两个副本，一个给发送方，另一个给接收方。现代的外交通信系统可能使用光存储磁带，隐藏在一个外交公文袋的特制容器里。有很多技术可以用来产生随机数。Marks 描述了牛津那些老妇人的洗牌计数器是如何生成 SOE 间谍写在丝绸上的密钥的。

对密钥序列生成器来说一个重要的问题是：不管加密多个备份磁带还是多个在通信信道中传输的消息，都应该极力避免同样的密钥序列使用多次。在二战期间，俄罗斯的外交通信量超过了他们提前分配给大使馆的一次一密的数量，所以只能重用密钥，这是一个严重的错误。试想一下：如果 $M_1 + K = C_1$，且 $M_2 + K = C_2$，则攻击者把两个密文组合起来就可以得到关于两条消息的等式：$C_1 - C_2 = M_1 - M_2$，所以如果消息 M_i 有足够的冗余度，就可以把 M_1 和 M_2 恢复过来。实际上，文本消息通常都包含有足够的冗余度，使得它极有可能被重现（恢复）。比如在前面关于俄罗斯的例子中就导致了所谓的 Venona 计划，在这个计划中，美国和英国解密了大量俄国战时的通信并且粉碎了一批俄国间谍的活动，正如谚语所言："绝对不要相信第二遍的磁带。"

实际上在任何序列密码系统中都要有上述同样的考虑，普通的加密工程师在实现一个序列密码生成器算法系统时除了密钥外还有一个种子。每加密一次，我们都希望加密器产生不同的密钥序列，所以每次输入到加密器的密钥应该是不同的。如果两个用户长期共享的密钥是 K，他们应该用一个称为种子的消息数字 N（或其他数字）与对方通信，然后把它通过一个哈希函数得到一个工作密钥 $h(K, N)$，这个工作密钥才是输入到加密机中的密钥。

5.3.3　随机置换：分组密码

第三种基本加密方式也是现代商业加密中最重要的加密方式就是分组密码，称之为随机置换，这里的变换函数是可逆的，且输入明文和输出密文都是固定大小的，比如，Playfair 系统的输入输出都是两个字符，DES 系统的输入输出位串长度也都是 64 位。不管可用的基本字母和符号数目是多少，加密都是对固定长度的分组起作用的（如果你想对一个短输入加密，必须对它进行填充，就像在 Playfair 系统中，在后面加"z"一样）。

可以把分组密码想像成下面的过程：跟前面一样，把小精灵、骰子、卷轴纸一起关到一个盒子里，左边是一列列的明文，右边是一列列的密文，当我们请求小精灵对一条消息加密时，它检查左边的明文列表看是否有以前的记录，如果没有，它用骰子产生一个合适长度的随机密文（此时，密文还没有出现在右边的滚动栏里），并在卷轴纸上把明文/密文对记录下来；如果有以前的记录，就从右边的列表中找出相对应的密文给我们。

当要进行解密时，小精灵把加密函数求逆后做与加密过程同样的事情：输入密文，检查是否有以前的记录（这次是在右边的列表中查找），如果找到了，它就把以前配对好的对应消息给出来，如果没有，就产生一条随机消息（该消息在左边列表中没有出现过）给我们并把它记录下来。

分组密码是一个密钥的伪随机置换家族，对每一个密钥，都存在一个与其他置换独立的置换，此时就可以把每个密钥对应于一卷不同的卷轴纸。从直觉上说，给定明文和密钥，加密机器应该输出密文，给定密文和密钥，则应该输出明文，但仅仅给出明文和密文中的一个，那么什么也不能输出。

下面使用本书第 2 章建立的记号来表述分组密码：

$$C = \{M\}_k$$

随机置换模型同样允许我们在分组密码上定义不同的攻击类型。比如，在已知明文攻击（known plaintext attack）中，攻击者仅仅给定一些从与目标密钥对应的预言中随机选择的输入输出；在一个选择明文攻击中，攻击者被允许输入一定次数的明文查询就可以得到对应的密文；在选择密文攻击中，攻击者设法进行一些密文查询；在选择明文/密文攻击中，可以允许他对明文或密文进行查询；最后，在相关密钥攻击中，攻击者使用与目标密钥 K 相关的密钥（比如，$K+1$ 或 $K+2$ 等）进行查询就可以得到回答。

无论在哪种攻击中，攻击者的主观意愿要么是推导出一个他以前没有经历过的查询的答复（伪造攻击），要么就是恢复密钥（不必吃惊，这就是有名的密钥恢复攻击）。

准确地描述攻击是很重要的。当某人找出了一种加密方法的漏洞时，它要么与你平时的应用有关，要么无关。通常是无关的，但有时候这种漏洞会被媒体夸大，所以你需要清楚地对你的老板和客户解释为什么它不会是一个问题。为此，你应该仔细地找到到底发现的是哪种攻击，攻击的参数是什么等等。举个例子，发布的第一个针对 DES 算法的主要攻击需要挑选 2^{47} 个明文才能找出密钥，而第二个改进的主要攻击被证明只需要 2^{43} 个已知明文。虽然这些攻击在科学上非常重要，但它们对实际应用的影响为零，因为实际系统不可能提供如此多的已知明文（更不用说可选的了）给攻击者。这样的攻击常常是指验证性的，即使它们有一些商业上的影响。比如：对 DES 的攻击潜在地破坏了人们对这种加密系统的信心，从而推动人们去探索其他的加密方法。有些情况下，验证性攻击中的一些思想被后来的加密开发所采用。

你应该担心什么样的攻击与你的实际应用密切相关。比如，对一个广播娱乐系统来说，你也许会买一个解码器，观察大量材料后把它与加密后的广播信号相比较；在这里，已知明文攻击就是你应该担心的主要威胁。但在非常多的应用中，选择明文攻击是最有可能发生的。很明显的例子中包括 ATM，在这个系统中，如果你允许客户任意更改他们的 PIN，攻击者就可以通过一个连接在 ATM 和银行线路上的窃听设备窃听可能值的范围和观察等效加密的方式来更改用户的 PIN。一个更传统的例子是外交消息传送系统，外国政府可能会给某个国家的外交官一条消息，让他传送到他的首都，而这条消息是专门为帮助本地的破译者了解外交官密码本上的不全部分而设计的 [428]。一般而言，如果攻击者可以往你的系统中插入任何类型的消息，那么你就应该考虑是否会受到选择明文攻击。

除此之外的其他攻击往往比较专业化，比如，当存在午餐时间攻击者的威胁时，就应该小心选择明文/密文攻击了。午餐时间攻击者是指当合法用户不在时，可能会临时访问加密设备的那些人。当分组密码作为重建哈希函数的构造块（这些以后再讨论）时你就应该关注相关密钥攻击了。

5.3.4 公钥加密和陷门单向置换

公钥加密算法是一种特殊的分组密码，用前面的例子来说就是任何有请求的人都被分配一个特殊密钥，小精灵会对这个密钥加密，但只有密钥的拥有者才能进行解密操作。继续分析下去，用户可能给小精灵一个秘密的名字，这个名字只有他（或她）本人和小精灵知道，他使用小精灵的公共单向函数计算这个秘密名字的哈希值，同时公开这个值，接着指示小精灵对任何引用该值的人进行加密操作。

这意味着交易的当事人，比如 Alice，可以公布一个密钥，而 Bob 如果愿意，即使他们从没有见过面也可以马上加密一条消息发送给她。对他们来说什么是必需的呢？那就是可以访问预言系统。当然这里还有很多细节需要考虑，比如 Alice 的名字怎样才能绑定到密钥上，实际应用中 Bob 是否可以随便发送什么东西给 Alice 等等，这些将在本书后面讨论。

实现公钥加密的一种普通方法是陷门单向置换。这是一个几乎所有人都可以完成的计算，但是只有那些知道陷门（比如一个密钥）的人才可以对它求逆。这个模型与加密哈希函数中的单向函数类似，即使如此本书还是要把它正式提出来：公钥加密系统包括这样一个函数：给定一个随机输入 R，将返回两个密钥 KR（公钥）和 KR^{-1}（私钥）且具有下面的性质：

- 给定 KR，不可能计算出 KR^{-1}（当然也不可能算出 R）。
- 存在一个加密函数 $\{...\}$，使得给定一条消息 M，使用加密密钥 KR，将产生一个密文 $C = \{M\}_{KR}$。
- 存在一个解密函数，使得给定一段密文 C，使用解密密钥 KR^{-1}，将产生原始消息 $M = \{C\}_{KR^{-1}}$。

从实际应用考虑，我们希望预言系统在通信信道的两个终端都可以被复制，这也意味着要么使用防篡改的硬件要么使用数学算法（实际应用中用得更普遍）而不是金属硬件来实现加密函数。这就是为什么我们给出的第二个模型比第一个模型抽象性更少但用处更大的原因。不管怎样，本书将会在后面探讨它的实现细节。

5.3.5 数字签名

本书将要定义的最后一种基本加密方法是数字签名，它的基本思想是一条消息上的签名只能由一个人（个体）创建，但可以被任何人校验。因此可以像在纸张世界里的普通签名一样，在电子文本世界形成电子签名函数。

签名方案可以是确定的也可以是随机的：第一种方式，计算一条消息上的签名往往得到同样的结果；后一种方式，每次计算的时候将会得到不同的结果（后者更像笔迹签名，没有任何两个人的笔迹完全相同，但是银行有办法确定谁是合法客户谁是假冒者）。同样，签名方案要么支持要么不支持消息恢复，如果支持，那么对于给定的签名，任何人都可以在消息产生的地方恢复消息；如果不支持，那么检验人在进行验证之前需要知道或者猜测消息的内容（还有更深入更专业的签名方案，比如盲签名和阈值签名，但本书现在不准备讨论它们）。

严格说来，一个数字签名方案和公钥加密方案一样，有一个密钥对生成函数，即，给定一个随机输入 R 将输出两个密钥 σR（私人签名密钥）和 VR（公共签名验证密钥）且具有下列性质：

- 已知公共签名验证密钥 VR，要计算私人签名密钥 σR 是不可能的。
- 存在一个数字签名函数，使得给定一条消息 M 和一个私人签名密钥 σR 时，都可以产生一个签名 $Sig_{\sigma R}(M)$。
- 存在一个签名验证函数，使得给定签名 $Sig_{\sigma R}(M)$ 和公共签名验证密钥 VR，如果签名计算结果等于 σR，将会输出 $TRUE$；否则，输出 $FALSE$。

如果在随机预言模型中描述，则可以把一个数字签名算法建模成一个把任何输入消息简化为一个固定长度单向哈希值的随机函数，它后面跟着一个特殊类型的分组密码，在这个函数中，小精灵一方面只对当事人进行签名操作，另一方面，对访问的任何人都进行验证

操作。

签名验证可以有两种形式，在最初的方案中，小精灵（或签名验证算法）仅仅依赖签名是否正确而输出 TRUE 或 FALSE；但在消息恢复的方案中，任何人都可以输入签名并且得到对应的消息。从前面的小精灵模型来看，这意味着如果小精灵以前见过这个签名，它将给出记录在卷轴纸上的对应消息；否则，它将给出一个随机值（同时把输入和随机输出记录为一个签名/消息对）。有时这正是我们所期望的：当在低带宽信道上传送短消息时，只传送签名比传送签名加消息要节省空间。一个例子就是在机器印刷邮票或者说邮戳系统中（这项技术在美国和其他一些国家已经投入使用）：邮票包括由邮局仪表产生的带有数字签名的二级条形码，条形码必须包括各种信息，比如邮票的面值、日期、接收者和发送者的邮政编码等，关于这项技术的更详细讨论见第 12 章“安全印刷和印章”的最后部分内容。

然而，在一般情况下，并不需要消息恢复，由于签名的消息是任意长度的，所以可以先把消息通过一个哈希函数然后标记哈希值。哈希函数是单向的，这样的后果是这种签名方案不能恢复消息（如果有更好的签名方案可以实现这一点，那么消息的哈希值也可以从签名中恢复出来）。

5.4 对称加密方法

既然已经定义了基本的加密方法，下面就来揭开它们的面纱，看看它们在实际生活中是怎么实现的。由于密码学的知识大部分是针对数学专业的毕业生的，下面的讲解建立在我与计算机科学专业的学生多年探讨的基础上。我希望非数学专业的学生能够抓住问题的本质。实际上，其实从研究的角度上说，许多密码学上的问题既涉及计算机科学也涉及数学。现今对密码的攻击无非就是猜位、搜寻模式、分类可能的结果等等，这要比自我夸耀实际得多。

这一节我们主要集中在分组密码，然后在下一节转移到怎样通过分组密码实现哈希函数和序列密码以及反向过程（在后面的章节，还会看到一些特殊目的的加密方法）。

5.4.1 SP 网络

香农在 20 世纪 40 年代曾提出一个强有力的加密方法可以通过重复性的替代置换建立。比如，某人可以往一个输入文本块中加入一些密钥内容，然后把输入子集打乱顺序，重复这样的过程数次。香农把一段密文的性质描述为混乱或扩散——混乱就是加入不可知的密钥值混淆攻击者破译明文符号的值，而扩散意味着通过密文传播明文信息，分组密码既需要混乱也需要扩散。

早期的分组密码只是一个由替代和置换电路组合的简单网络，所以称之为 SP 网络。图 5-10 中是一个有 16 个输入的 SP 网络，可以把它想像成 16 位数字的输入，它还包括两层 4 位的可逆替换盒，可以把它当作一个包含数字 0 到 15 的置换查找表。

这种结构的关键是如果我们准备实现一个任意 16 位到 16 位的数字逻辑函数，则需要 2^{20} 位的存储器——每个单独的输出位就需要一个 2^{16} 位的查找表。这样总共需要数十万的门电路，而一个 4 位到 4 位的函数只需要 64 位存储器。人们希望选择合适的参数，这样不知道密钥的攻击者就无法区别一个 16 位到 16 位的随机函数和由这种简单结构重复而产生的函数。该加密系统的密钥可以包括 4 位 S 盒中的某些选项，也可以被加到每轮的输入端产生混乱，同时经过 S 盒输出在结果中产生扩散。

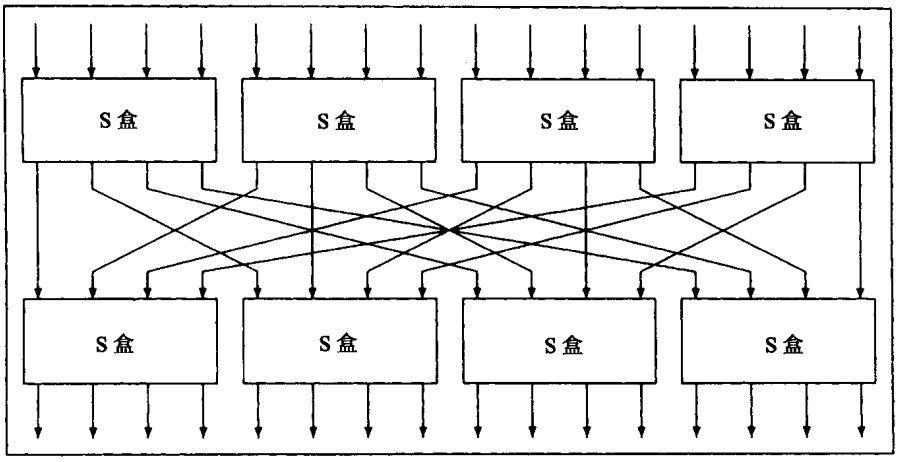


图 5-10　一个简单的 16 位 SP 网络分组密码

要安全地实现这样的结构需要满足三个条件：

1）加密器必须要足够的“宽”。

2）加密器需要足够的轮数。

3）S 盒的选择要合适。

5.4.1.1　分组的大小

首先，一个分组大小为 16 位的分组密码的应用范围会受到很大的限制，因为攻击者只要可以观察到明文分组和密文分组就可以建立一个明文—密文字典。生日定理告诉我们即使输入的明文是随机的，只要攻击者知道比 2^8 稍微多一点的密文分组，他马上就可以找到一个和明文匹配的密文。所以实际的分组密码系统通常使用 64 位、128 位甚至更长的明文和密文分组。如果使用 4 位到 4 位的 S 盒，就需要 16 个（对 64 位的分组大小）或 32 个（对 128 位的分组大小）这样的结构。

5.4.1.2　轮的数目

其次，必须有足够的轮数。在图 5-10 中，两轮是完全不够的，因为攻击者可以把输入位调整成合适的模式从而推理出 S 盒的值。比如，他可以把右边的 12 位保持为常数然后调整左边的 4 位输入来推理出左边顶上的一个 S 盒的值（实际的攻击比这个稍微复杂一点，因为有时候对一个 S 盒的输入的调整不会使任何输出位产生变化。为此还需要改变其他的输入位进行再次调整，但它的实现仍然像学生做练习题一样简单）。

需要的轮数依赖于数据在加密器中扩散的速度，在前面的简单例子中，扩散是非常缓慢的，因为在 S 盒的一轮里，每个输出位仅仅与下一轮的一个输入位相连接。如果不采用线路的简单变换，而是采用当前轮的每个输入位都是先前轮中几个输出的异或（模二加）这样一种线性变换，混乱的效率就提高了。当然如果分组密码既用于加密也用于解密，那么这个线性变换必须是可逆的，在下面几节关于 AES 和 Serpent 系统的介绍中将会看到几个具体的例子。

5.4.1.3　S 盒的选择

S 盒的设计也同样影响到安全所需的轮数，对最坏选择的研究将会使我们更加深入地理解分组密码的理论。假设 S 盒是一个输入位（0，1，2，…，15）到输出位（5，7，0，2，

4，3，1，6，8，10，15，12，9，11，14，13）的变换，那么最重要的输入位应该不加改变地作为最重要的输出位。如果在前面的加密器中每轮都使用了同样的S盒，那么输入分组的最重要位就会经过加密器变成输出分组的最重要位，这样就不能说这种加密方法是伪随机的了。

5.4.1.4 线性解密

对实际分组密码系统的攻击常常要比下面这个假设的例子难得多，但它们的思想是类似的。要做出一个具有下面性质的S盒是完全可能的：输入端的位1等于输出端的位2加上位4，更一般的说，S盒存在一个以确定概率发生的线性近似。线性解密［526］就是通过收集一系列诸如“第一个S盒的输入端位2加位5等于输出端的位1加位8的概率是13/16”这样的关系式来进行解密的，接下来就是寻找把输入位、输出位和出现概率不等于0.5的密钥位用一个代数关系式连接起来的方法。如果能够找到一个在整个密文中持续出现概率 $p = 0.5 + 1/M$ 的线性关系式，那么根据概率学理论，一旦拥有了 M^2 个已知文本，就可以恢复密钥位。如果最优的线性关系式需要的 M^2 比已知文本的所有可能数目（输入输出都是n位时就是 2^n 个）还多时，就可以认为这种加密方法对线性解密攻击是安全的。

5.4.1.5 差分解密

差分解密［102］与线性解密是类似的，但它建立在这样的概率基础上：给定一个S盒上的输入变化，将在输出端产生一个确定的变化。它在一个8位的S盒上的一般表现为：如果迅速地把输入位2、3和7翻转，那么只有输出位0和1以11/16的概率发生翻转。实际上，对任何非线性布尔函数，调整输入位的某些组合将导致输出位组合以不等于0.5的概率发生改变。其分析过程是：考虑输入的所有可能差分模式，寻找那些当输入改变 δ_i 时，输出将会以极高（或极低）的概率生成 δ_0 改变的 δ_i 和 δ_0 值。

作为一种线性解密法，我们接下来就是想办法把这些东西联系起来，使输进加密器的输入差分能够在轮数上以一个可用的概率产生一个已知的输出差分。这样一旦给定足够的可选输入，就能够得到期望的输出从而可以推出密钥。对线性加密法来说，如果攻击该系统所需的文本数目比该密钥不同文本总的数目还要大的话，一般就可以认为这种加密法是安全的（同时也必须注意非常特殊的情况，比如对一个32位的密文分组和128位密钥的加密方法，给定单个配对后差分攻击成功的可能性是 2^{-40}，但是如果给定了一系列密钥下的很多文本，最终就能够解密出当前的密钥）。

这两种方案有很大的不同之处，比如，不搜寻以大概率发生的差分输入，而搜寻不会发生（或者基本不会发生）的差分。即使这种方法对很多系统来说，都存在确定的解密概率，它仍然有一个吸引人的名字：不可能解密法［101］。同样，对特殊的加密方法常常会有各种各样的攻击方法。

分组密码设计需要考虑很多平衡。比如，通过仔细地设计轮函数，可以减少每轮的信息泄漏以及由此所需的轮数。然而设计越复杂，软件运行就越慢，或者硬件上所需的门电路就越多，所以使用简单但数目多一些的轮函数可能效果更好。但简单的轮函数分析起来也更容易。为了给未来更先进的数学工具对安全的攻击留下一定的空间，一个谨慎的密码设计师往往使用比抵挡已知的当前攻击所需更多的轮数。实际上可以证明某一种加密方法可以抵挡所有目前已知的攻击，但没有任何理由说它可以抵挡我们还不知道的攻击（从分组密码的安全性的一般证明可以推出攻击者的计算能力，这就产生一个限定的结果，比如 $P \neq NP$，这样

的结果可能推动计算机科学的变革)。

安全工程师应该牢记的一点就是：分组密码的解密是一种有着很深、很广的理论基础的复杂方法，所以使用一种被专家们钻研透彻了的现有设计比用你自己搞得晕头转向的安全设计要好得多。

5.4.1.6 Serpent 加密

作为一个具体的例子，Serpent 加密算法是一个输入、输出分组都为 128 位的 SP 网络。它通过 32 个轮循环处理，在每个轮函数中，首先，输入 128 位的密钥资料，接着把文本通过 32 个 4 位宽的 S 盒，这样就可以形成从当前轮的每个输出到下一轮的许多 S 盒输入之间的线性变换。当前轮的每个输入位不是简单地来自上一轮的输出位，而是两个输出或七个输出的异或。这就意味着一个输入位的改变会迅速地扩散到整个加密器——这就是所谓的雪崩效应，它使得线性破译和差分破译攻击变得异常困难。而且在最后的轮函数中，明文中被加入了一个比 128 位更多的密钥资料。如果用户提供的密钥达到 256 位，则进行计算的密钥资料就是 128 位的 33 倍了。

图 5-10 是一个实际使用的但输入修正为足够宽以使之有足够多轮数的加密器结构。其中，S 盒是精心挑选的，目的是使线性破译和差分破译难度更大，它们严格地限制了输入输出位的线性相关性以及固定输入位模式后对输出的最大影响。在给定轮数中的 32 个 S 盒是一样的，这意味在一个 32 位的处理器上，用位分割技术的软件实现是效率很高的。

简单的结构使得 Serpent 易于分析，可以证明它能够抵挡当今所有已知的攻击（在文献[40]中给出了一个关于 Serpent 的完整描述，并且可以从[41]中下载它的多语言实现)。

5.4.2 高级加密标准

下面准备讨论高级加密标准（Advanced Encryption Standard，AES)，在 Vincent Rijmen 和 Joan Daemen[⊖] 发明该算法后，这种算法也被称作 Rijndael 算法。这种算法操作的分组大小是 128 位，可以使用 128、192 和 256 位长度的密钥。它是一个 SP 网络，为详细说明，需要固定 S 盒、轮函数之间的线性变换以及密钥加进计算中的方式。

Rijndael 使用的是一个简单的处理单字节输入到单字节输出的 S 盒。从实现角度，它可以当作一个简单的 256 字节的查找表；它实际上是在 GF（2^8）域上通过等式 $S(x) = M(1/x) + b$ 定义的，其中 M 选择的是一个合适的矩阵，b 是一个常数。这种结构对差分和线性给出了严格的界限。

这种线性变换是把待加密的 16 字节值安排在一个方格子中，然后对这些字节进行变序和混和操作（Rijndael 方法是从早期称作方格加密的方法继承而来的，方格加密引进了这种技术)。

这种线性变换的第一步就是变序，规则是：最顶行的四个字节不左移，第二行左移一个位置，第三行左移两个位置，第四行左移三个位置。第二步就是列的混和，规则是：一列中的四个字节用矩阵相乘进行混和，这个过程可以用图5-11说明，该图举例说明了第一列的

⊖ 如果你来自荷兰、比利时或者南非，Rijndael 的发音就是你平时的发音；如果你不是荷兰语系，有时它的发音就像“rain-dahl”。在荷兰语系中“J”不是一个辅音，所以 Rijndael 的发音一点都不像“Region Deal”，同样，Rijmen 也被发成“Raymen”而不是“Ridgemen”。

第三个字节怎样传递值的变化，这种组合的作用就是加密器输入值的改变在两轮以后就能够潜在地影响它的所有输出。

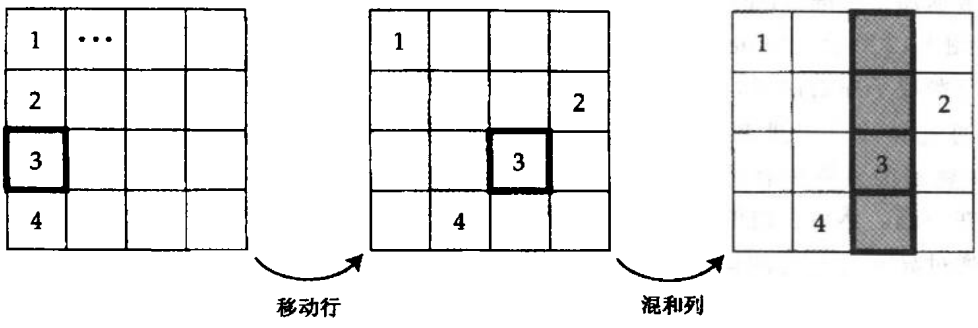


图 5-11 以输入的第三个字节为例说明 Rijndael 线性变换

在线性变换以后，密钥资料就一个字节一个字节地加进来，这就是说在每一轮中都需要一个 16 字节的密钥资料；它们是通过递归从用户提供的密钥资料中得出的。

这种算法在 128 位密钥的系统中需要 10 轮，在 192 位密钥的系统中需要 12 轮，在 256 位密钥的系统中需要 14 轮。这样数目的轮循环可以留有 50% 的安全系数，目前已知的最快捷攻击可以破坏 128 位密钥中的 6 轮，192 位密钥中的 7 轮以及 256 位密钥中的 9 轮。在分组密码界达成了一种共识：即使允许用很先进的攻击方法对 Rijndael 的所有轮函数进行攻击，它们也只能是纯粹意义上的验证性攻击，因为它需要的文本数目是不可能实现的（而 Rijndael 系统对只需可行文本数的攻击的安全系数大概是 100%）。不论是从伪随机性的意义上，还是是否存在快捷攻击方式上，Rijndael 系统都没有任何安全性的证据，但我们仍然可以有足够的自信：对于目前所有实际的攻击来说，Rijndael 系统都是安全的。

在同 AES 的竞争中，Serpent 算法是不成功的（在最后的 AES 会议中，Rijndael 得到了 86 票的支持，Serpent 得到了 59 票，Twofish 得到了 31 票，RC6 是 23 票而 MARS 是 13 票）。即使作为 Serpent 算法的开创者，即使 Serpent 系统的设计比 Rijndael 有更大的安全系数，我仍然建议我的客户使用基于一般意义上的分组密码的 Rijndael 系统，我建议使用 256 位的密钥。这并不是因为即使是 128 位密钥的各种改进形式，它的 10 轮循环也不需要多长时间就能破坏，而是因为在实际制造时，一些密钥位通常会被泄漏，所以长密钥比短密钥要好。这一点本书将在第 14 章和第 15 章详细讨论。“只有在 Rijndael 系统被破坏的情况下才使用 Serpent”这样的说法是毫无意义的：在算法协商协议中存在致命错误的危险性比某人针对 Rijndael 系统进行攻击的危险性要大几个数量级（在本书后面将会看到许多要么使用多协议，要么像 DES 一样多次使用同一个协议而导致非常可观的攻击漏洞的例子）。

Rijndael 系统最终的详细规范将在 2001 年左右作为一个联合信息处理标准（Federal Information Processing Standard）公布。同时，在 Rijndael 的主页上将会用论文的形式描述它的算法[647]；在网上还会有很多它的实现方案，在 [213] 中还有一些关于 Rijndael 早期版本和方格的论文。

5.4.3 Feistel 加密

在实际应用中，大多数的分组密码使用一种更复杂的结构，这种结构是 20 世纪 50 年代

和60年代初期，Harst Feistel的技术人员为IFF开发密码保护的时候发明的。随后Feistel就搬到了IBM并成立了一个研究小组，后来该小组开发了被称为财务交易处理安全方面中流砥柱的数据加密标准（Data Encryption Standard，DES）算法。

如图5-12所示，Feistel加密器具有梯形结构，输入被分割成了左右两组，左边的轮函数f_1经过计算，并通过异或（二进制加法没有进位，在有些Feistel加密器中也使用有进位的加法）与右边的输入合并在一起。然后右边的轮函数f_2也同样经过计算，并通过异或与左边的输入合并在一起（本书使用记号⊕表示异或）。最后（如果轮数是偶数的话），左右边输入相互交换。

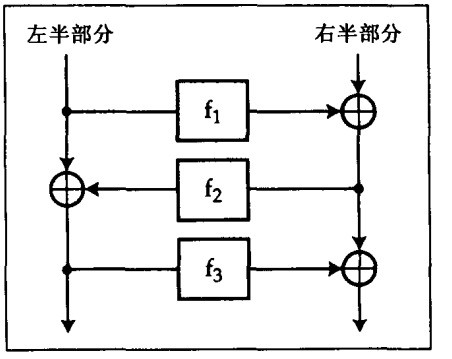


图5-12　Feistel加密器结构

对Feistel加密来说可以看到这样的记号：$\Psi(f, g, h, \ldots)$，这里$f, g, h, \ldots$是连续的轮函数，在这种记号规则下，上例的加密器就是$\Psi(f_1, f_2, f_3)$。能够对Feistel加密器进行解密的基本前提——实际上，也是该设计的关键——就是：

$$\Psi^{-1}(f_1, f_2, \ldots, f_{2k-1}) = \Psi(f_{2k-1}, \ldots, f_2, f_1)$$

换句话说，为了解密，只需要反向使用轮函数即可。因此，轮函数f_i不一定要是可逆的，这种Feistel结构使我们可以把任何一个单向函数变成一个分组密码，也意味着可以更少限制地选择具有好的混乱性和扩散性的轮函数，同时这种结构还可以更容易地满足其他诸如码大小、表大小、软件速度、硬件的门电路数等等设计限制。

5.4.3.1　Luby-Rackoff结论

在Feistel加密中使用的基本理论是1988年由Mike Luby和Charlie Rackoff证明的。他们证明了：如果f_i是随机函数，那么在选择明文攻击下，$\Psi(f_1, f_2, f_3)$是不能从一个随机变换中分离出来的。这个结果不久就引申为：在选择明文/密文攻击下，$\Psi(f_1, f_2, f_3, f_4)$是不可分离的——换句话说，它是一个伪随机变换。

本书在这里省略了很多技术细节，从工程术语的角度，它的作用就在于：给定一个性能非常好的轮函数，Feistel系统有四轮就足够了。所以如果有一个对它非常信任的哈希函数，就可以直截了当地建立一个分组密码系统。

5.4.3.2　DES

DES算法广泛应用在银行、政府和嵌入式系统中。比如，它就是自动柜员机网络的标准。

DES算法实际是一种分组大小为64位、密钥为56位的Feistel加密法。它的轮函数对32位大小的半分组进行操作，包括下面的四个步骤：

1）首先，分组大小从32位扩展到48位。

2）接着，对48位的轮密钥用异或的方式进行混和。

3）然后，混和后的结果通过一行8个S盒，每个S盒有6位输入同时产生4位输出。

4）最后，根据一个固定的模式，变换输出位的顺序。

扩展后的结果、混合密钥以及S盒见图5-13。

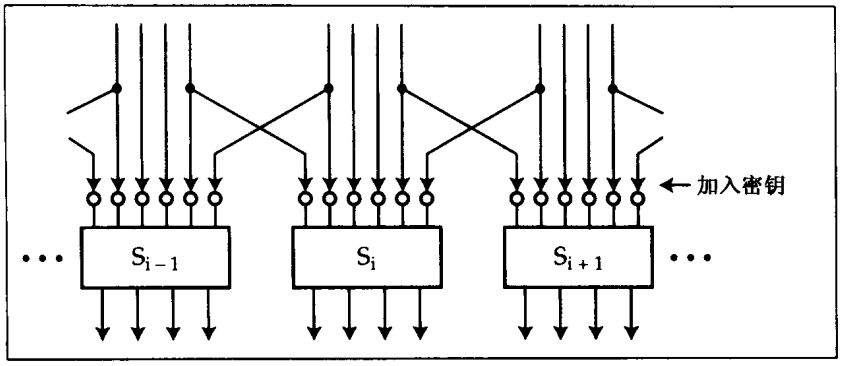


图5-13 DES轮函数

轮密钥来自于用户提供的密钥，它是在12个不同轮中用每个用户的密钥位按照一种不太正规的模式得到的（文献［575］给出了一个完整具体的DES模型；在［681］中可以找到它的代码，在网上的很多地方可以下载到）。

1974年引进DES时曾引起了很多争议，其中最明显的批评就是它的密钥太短。假设某人想找到一个56位的密钥，他采用的是强力攻击——也就是说试验所有可能的密钥——总的耗费时间就是2^{56}次加密的时间，而平均解决时间（总的耗费时间的一半）就是2^{55}次加密时间。Diffie和Hellman指出：一台DES密钥搜寻机器可以由100万个芯片组成，每个芯片一秒钟可以测试100万个密钥，由于100万约等于2^{20}，所以平均搜寻时间大概是2^{15}秒，或者说仅仅需要9个小时就可以找到密钥。他们说这样的一台机器在1977年的花费大概是$20 000 000［249］，而发明了DES的IBM则讥笑说：他们可能会花费美国政府$200 000 000才能制造出这样一台机器（从某种意义上说双方都是正确的）。

在20世纪80年代，曾经有一个经久不衰的谣传说很多情报侦察机构正在建造DES密钥搜寻机器，但直到1997年第一台公共密钥搜寻工具才成功面世，它组织了分布在网络上的14 000台奔腾级计算机，花费了超过4个月的时间才找到DES密钥。1998年，电子新领域基金会（Electronic Frontier Foundation，EFF）制造了一台花费不到$250 000的DES密钥搜寻机；它可以在三天以内破译一个DES查询，该搜寻机器包括1 536个运行在40 MHz的芯片系统，每个芯片包括24个搜寻单元，每个搜寻单元花费16个时钟周期来进行一次解密过程，因此搜寻速率是每个搜寻单元每秒钟可以测试2 500 000次解密，或者说每个芯片每秒可以测试60 000 000个密钥（破译机的设计是公开的，可以从文献［265］中找到）。很明显，现在DES系统的密钥长度要保护数据不被有足够破译能力而且别有用心的攻击者破译的话是肯定不够的，所以银行目前也正在更新他们的支付系统。

对DES的另外一类批评意见是，既然IBM在美国政府的要求下没有公布它的设计原理，那就很可能存在一个陷门使得美国政府能够很方便地访问别人的系统。然而，在差分解密被开发出来并且公布以后，DES的设计原理就在1992年被公布了［205］。事实真相是IBM在1972年就发现了这些技术，NSA发现得更早，IBM在NSA的要求下保留了设计的细节，本书将在第21章讲述这些政治方面的问题。

现在我们对DES进行一个比较彻底的分析，对它最有名的快捷攻击是用2^{42}个已知明文

的线性攻击。超过20轮的DES可以看作是安全的，但实际系统中，它的安全性是受到密钥长度限制的。我还从没有听过一个攻击者可以拥有哪怕是2^{40}个已知明文的例子，所以已知明文攻击在实际中是不可行的。然而，从密钥搜寻上暴露出来的弱点不能忽视，如果摩尔定理继续生效，那么到2015年或2020年，在一台普通PC上用几个月的时间找到一个DES密钥是完全可能的。这也意味着，低安全性等级的系统（比如计程器）在强力攻击下是很容易找到漏洞的（你的反应可能是："给我一个攻击计程器的理由"，是的，我将会给出这个理由，那就是第10章"监控系统"）。

一种阻止密钥搜寻的方法是变白技术，除了56位的密钥（称之为k_0）外，还选择两个64位的"变白"密钥k_1和k_2，在加密之前把k_1与明文进行异或，加密之后把k_2与加密输出进行异或从而得到密文，这种合成密码称作DESX密码，它用在Win2K的文件加密系统中，用公式表示就是：

$$DESX\ (k_0,\ k_1,\ k_2;\ M) = DES\ (k_0;\ M \oplus k_1) \oplus k_2$$

可以证明：在合理的假设上，DESX就有你所期望的性质：它继承了DES抵抗差分解密的优点，同时它抵抗密钥搜寻的能力随着变白数目的增加而增强[457]。

另外一种阻止DES密钥搜寻的方法是用不同的密钥多次使用这个算法，这是由银行网络引进的，现在，银行使用的三重DES（triple-DES）算法已经成为美国政府标准的草案[575]，三重DES是先进行一次加密，接着解密，然后更深一步地进行加密，每次都使用独立的密钥，用公式表示就是：

$$3DES\ (k_0,\ k_1,\ k_2;\ M) = DES\ (k_2,\ DES^{-1}\ (k_1,\ DES\ (k_0;\ M)))$$

进行这样设计的原因就是：设定三个密钥相等，很明显这是一个单DES加密过程，可以很容易得到相同的结果，因此可以对遗留设备提供一个向后兼容的模式（有些系统使用两个密钥的三重DES，它设置$k_2 = k_0$，这是一种在单DES和三重DES之间的折衷方案）。

5.5 操作模式

在实际中，你怎样使用一种加密算法比你选择哪种算法要重要得多。其中一个重要的因素就是操作模式，它具体实现了一个固定大小的分组密码（DES是8位，AES是16位）是怎样被扩展成可以处理任意长度消息的。

有一些操作模式可以在多个密文块上实现分组密码，透彻理解并选择合适的模式，是安全地使用分组密码的重要因素。

5.5.1 电子密码本模式

在电子密码本（Electronic Code Book，ECB）模式中，我们只对每个连续的明文块与分组密码进行加密得到密文，就像前面例子中的Playfair加密一样。对许多简单的操作，比如查询—应答系统和一些密钥管理任务来说这已经足够了，除此之外，它还可以用在提款机系统中加密用户PIN。然而，如果用它加密冗余数据，那么这种模式就会失效，因为攻击者可以推断出明文的有关信息。举一个例子，如果字处理格式有许多空字符串，那么密文中就会有许多由空字符在当前密钥下加密得到的值所组成的密文块。

在一个从80年代末期起开始风靡全球的电子邮件系统中，加密就是使用DES的ECB模

式，其密钥就是用户输入的8字符的口令。如果你观察一下由这个系统产生的密文，就会发现某个特殊的块远比其他块要常见得多——这个块就是与明文中的空操作符相对应的密文块。这就使得针对DES加密系统的一种最简单的攻击成为可能：每次只用字典中的一个口令对空块进行加密，再把加密得到的所有的密文排序就可以得到答案了。这样只要该口令包含在你的字典中，你就可以破译它的任何密文了。

此外，使用ECB模式对多于一个分组长度的消息或者那些需要验证的消息进行加密——比如银行支付信息——是愚蠢的，因为这些消息很容易遭受对分组边界实施的分割—拼接攻击。举一个例子，假设一条银行的消息是："请从总数Y中支付X，他们是以数目Z为参考的"，那么攻击者就可以开始设计一个攻击，使得X中的某些数字能够被Z中的某些数字代替。

5.5.2 分组密码链接

商业应用中针对多个分组消息的加密使用分组密码链接模式，或简称CBC模式。也就是说把前面的密文分组与当前的明文分组进行异或后再进行加密（如图5-14所示）。

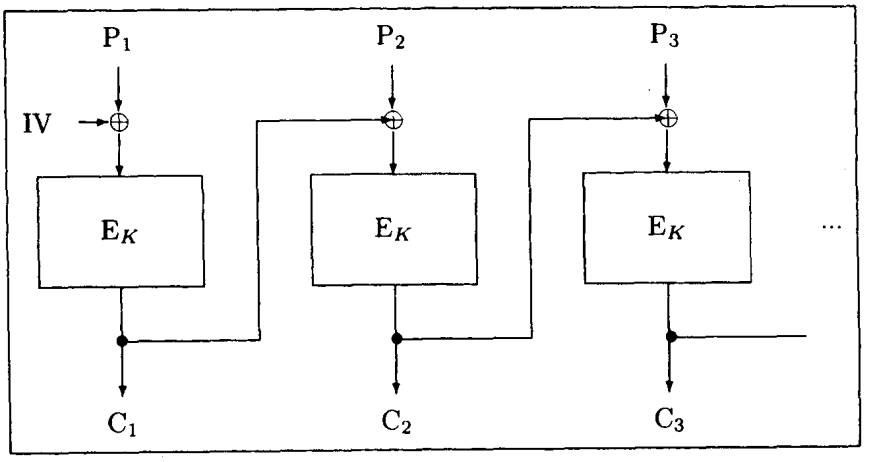


图5-14 分组密码链接（CBC）模式

这种模式可以很有效地隐蔽明文的数据模式：每个当前分组的加密依赖于所有的先前分组。输入IV是一个初始化矢量，它是一个随机数，实现的功能和序列密码中的种子相同，同时保证加密相同的明文分组时，密文不会泄漏铅印的明文消息头的任何信息。

然而，知道一些明文的攻击者可以切割和拼合一条消息（或几条用相同密钥加密的消息的部分），所以它对消息完整性的保护不强。

5.5.3 输出反馈

输出反馈（Output Feedback，OFB）模式就是重复加密一个初始值并用作一个密钥序列，这在前面的序列密码中讨论过。把IV作为初始化矢量或者说种子，则密钥序列的第 i 块将由 $K_1 = \{IV\}_K$ 和 $K_i = \{K_{i-1}\}_K$ 给定：

$K_i = \{\ldots\{\{IV\}_K\}_K \ldots$ 共 i 次$\}$

这是把分组密码转变成序列密码的标准方法，密钥 K 被扩展成块大小为 K_i 的密钥序

列，密钥序列 K_i 与消息 M_i 进行异或运算得到密文 $C_i = M_i \oplus K_i$，这种结构有时候也叫做*加法序列密码*，因为异或运算就是模二加（有些老的手工系统使用模 26 加）。

有时候也使用特殊的密钥序列生成器，比如，第 17 章将要讲到的 A5 算法就比 DES 系统的门电路少得多，因此它被用在电池能量是设计关键参数的移动通信中，当然没有类似的限制时，一般都在 OFB 模式中使用分组密码产生密钥序列。

所有的加法序列密码都有一个重大的缺陷：它们不能保护消息的完整性。本书在 5.2.2 节描述一次一密加密的时候提到过这一点，但是认识到它不仅仅影响系统的“完美安全性”而且会影响到序列密码“实际应用”是很重要的。举一个例子，假设使用序列密码加密资金转移的消息。这些消息具有很特殊的结构。你可能也知道，它的 37 ~ 42 字节包含资金转移的数目，那么你就可以实施下面的攻击计划了：首先，你必须让来自当地银行的数据量经过你的计算机（这一点可以在线路上使用物理连接或者使用第二部分讨论的更加简单的标准路由攻击方法来实现），接着你到银行给你的同伙转账一笔数目不多的钱（比如$500）。那么密文 $C_i = M_i \oplus K_i$ 就会适时到达你的机器。由于你已经知道 M_i 中的 37 ~ 42 字节是什么消息，同时你也知道了 K_i，所以你就可以重新修改消息，给接收银行发一条支付你同伙$500 000而不是$500的指令！这是一个深度攻击的例子，它不仅能攻击具有完全安全性的一次一密系统，对其他比较低级的序列密码系统也适用。

5.5.4　计数器加密模式

输出反馈模式（事实上也是所有分组密码反馈模式）的一个可能缺陷就是它的延迟，因为反馈模式很难并行化。对于 CBC 模式来说，一个完整的密文分组必须计算每个输入分组和输出分组才能得到；对 OFB 模式来说，需要内存把密钥序列存储起来才能进行预先计算。这在超高速应用，比如 155 Mbit/s 骨干网的保密传输中是很不方便的。由于硅片很便宜，我们希望使用流水线来生产加密芯片，从而花费尽可能少的时间来加密新的分组（或者说生成新的密钥序列分组）。

最简单的解决途径常常是通过对计数器加密来产生一个密钥序列：$K_i = \{IV + i\}_K$。和以前的处理方法一样，密钥随后加入明文而得到密文（所以它也同样存在深度攻击的缺陷）。

当在一个高速链路上使用 64 位大小的分组密码，比如 DES 或三重 DES 时，这种模式可以解决周期长度的问题。在 OFB 模式中，一个分组大小为 n 位的分组密码的典型周期是 $2^{n/2}$ 个分组，根据生日定理可以看到，$2^{n/2}$ 个分组以后，密钥序列将会重复（比如一旦有多于 2^{32} 个长度为 64 位的值时，就可以找到相互匹配的对）。在 CBC 模式中也是这样，生日定理使得经过 $2^{n/2}$ 个分组后开始出现重复。但是计数器加密模式可以保证周期长度是 2^n，远远优于 $2^{n/2}$。

5.5.5　密码反馈模式

密码反馈（Cipher Feedback）或者说 CFB 模式是另外一种序列密码操作模式，它被设计成自同步方式，因此即使发生突发错误或者丢失一些位，在一个分组长度之后，系统也能够恢复同步。它是这样实现的：用分组密码加密密文的最后 n 位，然后把输出位中的一个加到下一个明文位中。

解密实现的是相反的过程，这时，密文从右边输入，如图 5-15 所示。因此，即使发生

突发错误或者丢失了一些位，只要接收到足够的密文位，系统马上就可以填满移位寄存器从而实现同步。

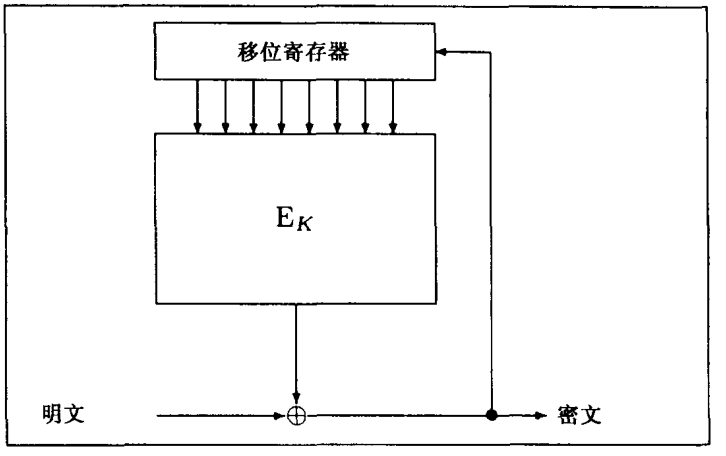


图 5-15 密码反馈模式（CFB）

密码反馈模式应用的场合并不是很多。它是一种特殊的操作模式，在军事上的高频雷达链路中可以见到，那时数字电路造价相对昂贵，现在高频雷达则出现衰落的迹象。现在，由于硅片已经很便宜了，人们往往用链路层协议来实现同步和纠错而不是用密码学来实现。

5.5.6 消息验证码模式

接下来的一种分组密码的正式操作模式并不是用来加密数据，而是保护数据的完整性和真实性，这就是消息验证码模式或简写为 MAC。为用分组密码确定一条消息的 MAC，可以使用 CBC 模式对它加密并且抛弃除最后一个密文分组以外的所有输出密文分组，而这最后的密文分组就是 MAC（为防止分割攻击，中间结果是保密的）。

这种结构使得 MAC 既依赖于所有的明文分组也依赖于密钥。提供固定的消息长度确保了它的安全性。可以证明，在这种情况下，任何针对 MAC 的攻击肯定也会是一个针对基本分组密码的攻击 [87]（如果消息长度是可变的，你必须确保由一个字符串计算出来的 MAC 不能被当作 IV 来计算另外一个字符串的 MAC，只有这样，才能够阻止攻击者通过两个字符串组合得到 MAC 的方法来欺骗系统）。

在实际应用中，往往既需要完整性也需要保护隐私，它是这样实现的：首先，使用一个密钥在消息上计算得到一个 MAC，接着，用另外一个密钥对它进行 CBC 加密。如果对加密和认证使用同样的密钥，则后者的安全性就不能得到保证，剪切—粘贴攻击仍有可能发生。

MAC 还存在其他的可能结构：常见的就是使用一个带密钥的哈希函数，本书在 5.6.2 节中将作详细的讨论，下面我们再次回顾一下哈希函数。

5.6 哈希函数

5.4.2.1 节中讲述了 Luby-Rackoff 理论是怎样用哈希函数构建分组密码的。同样，从分组

密码也可以构建哈希函数（事实上，也可以从序列密码中构建哈希函数和分组密码——因此，可以给出下节的一个结论，即，给定这三个中的一个，都可以构建其他的两个）。

哈希函数的操作模式是这样进行的：在分组密码系统的密钥输入端，每次输入一个消息分组，并用它更新哈希值（假设初始值 $H_0 = 0$）。为了使这个操作不可逆，可以加入前馈控制：第（i－1）次哈希值与第 i 轮的输出进行异或。这就是本书在分组密码中讲述的最后一个操作模式（见图 5-16）。

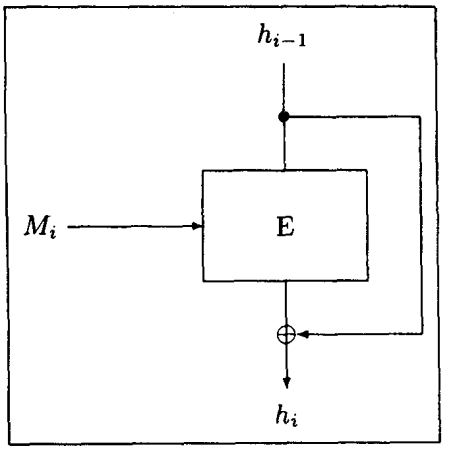


图 5-16　前馈模式（哈希函数）

5.6.1　基础加密的额外要求

生日效应在这里表现为另外一种形式，如果一个哈希函数是用一个 n 位的分组密码构建的，那么可能找到两条消息 $M_1 \neq M_2$，使得 $h(M_1) = h(M_2)$（对消息 M_i 执行大约 $2^{n/2}$ 次哈希操作才能找到一个匹配），所以，一个 64 位的分组密码是不充分的，因为伪造 2^{32} 条数量级的消息在实际中是可以做到的。

这不是哈希函数操作模式比为机密性设计的 CBC 操作模式对基础分组密码要求更严格的惟一方式。这里有一个来自 Treyfer 密码的很好例子，它在电子消费和国内电器设备中随处可见的 8051 微控制器上使用尽可能少的内存（只有 30 字节的 ROM）对数据进行加密设计［819］。

Treyfer 用 ROM 中的 256 个字节清除（代替）S 盒，ROM 中存放的可能是代码，也可能包含为防止商业克隆风险而添加的版权信息。在每个轮函数单元，都要对 8 字节消息和 8 字节密钥进行操作，操作方式是把消息字节与密钥字节相加，相加后的结果通过 S 盒与下一个消息字节相加，再把结果循环移动 1 位（见图 5-17）。循环移位可以处理一些当 S 盒在位平面上有不均匀的随机性（比如，包含了诸如版本信息的 ASCII 文本）时可能出现的问题，最后，算法通过较多的轮数（实际是 32 轮）来补偿简单轮函数结构和 S 盒的非理想性所引起的偏差。

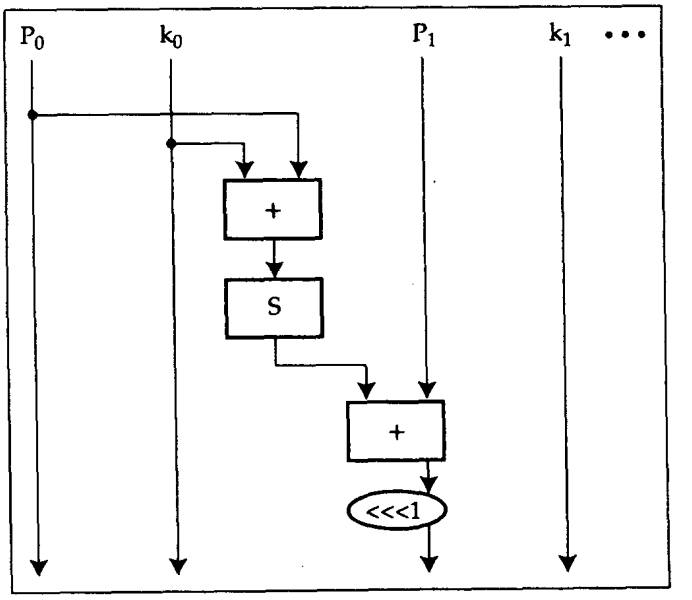


图 5-17　Treyfer 分组密码的基本单元

目前还没有针对 Treyfer 用在保密性和计算 MAC 上的已知攻击。然而，这个算法的确有缺陷，从而阻碍了它在哈希函数上的应用。那就是它容易遭受固定点攻击。给定任何输入，

都存在一个密钥能使输入不发生改变地离开。现在来看看是怎么回事，对每个字节的输入（不论S盒的输出是什么），当它加到右边字节时，它都会向左循环一位，如果对每个输入字节都有这样的输出，那么就很容易选择密钥值，使输入数据在一个轮循环（直到32个轮循环）后不发生改变地离开。发生这种事情的概率与S盒有关[⊖]。这也意味着当Treyfer用在哈希函数上时，很容易就可以找到碰撞（事实上，哈希函数必须是基于分组密码的，因为分组密码能够抵挡选择密钥攻击）。

5.6.2 常用哈希函数及其应用

在一些付费电视系统中，与Treyfer类似的算法也被用在密钥管理协议的哈希函数中，但通常它们都是经过修正的，以防止固定点攻击，这些修正措施包括：添加轮数，以某种方式混和密钥位（密钥调度算法）。

在应用中最常使用的三种哈希函数是相互联系的，而且都是基于密钥大小为512位，分组大小要么128位要么160位的分组密码的改进形式上的。MD4有3轮循环和一个128位的哈希值；近来发现了它的一个冲突[255]。MD5有4轮循环和一个128位的哈希值，而美国安全哈希标准是5轮循环和160位的哈希值。基于这些哈希函数的分组密码都是相似的，但它们的轮函数却是在32位处理器上可用寄存器操作的复杂组合[681]。这样看来只要攻击者不能完成2^{80}次计算，SHA1就是对伪随机函数的一个合理近似；但由于在军事情报机构和大公司实现2^{80}次计算是可能的，所以又引进了256位和512位的SHA版本。

哈希函数有很多用途，其中之一就是计算MAC。一种简洁的用密钥简化消息的哈希计算就是：$MAC_K(M) = h(k, M)$。但在实际中可以接受的简化方法叫做HMAC，除了计算哈希值外，它增加了一个额外步骤：对计算结果重新进行哈希计算。这两个哈希操作使用密钥的不同变体，它们分别是该密钥与两个不同的常数异或后得到的。因此，$HMAC_K(M) = h(k \oplus A, h(k \oplus B, M))$。其中，$A$是尽可能多地重复字节$0\times36$得到的，类似地，$B$也是从字节$0\times5C$得到的，这样做的原因就是使冲突发生更加困难[474]。

哈希函数的另外一个用途就是实现后面将要讲到的委托代理。举一个例子，为确定自己的知识产权，希望对一篇数字文档进行时间戳验证，但同时又不愿意透露文档的内容。在这种情况下，就可以把该文档进行哈希变换以后提交给商业时间戳服务机构。以后当我们公布该文档时，该文档在给定时间建立的哈希时间戳就可以证明我们当时的确拥有了该文档。

最后，在本书将要继续讨论非对称加密的时候，还有两个哈希函数的特殊应用值得一提：密钥更新和自动密钥。

*密钥更新*意味着两个或多个当事人在协商的时间通过单向哈希函数分享密钥：$K_i = h(K_{i-1})$。这样做的一个好处就是如果攻击者威胁到了系统中的一个用户而且盗取了密钥，他也只是得到了当前密钥而不能对后面的数据进行解密，这种性质就是有名的“*后向安全性*”。

*自动密钥*意味着两个或多个共享密钥的当事人在协商时间内发现最后一个密钥发生改变，就用他们已经交换的消息对密钥进行哈希变换：$K_{i+1} = h(K_i, M_{i1}, M_{i2}, \ldots)$。这样

⊖ 令人好奇的是，一个实现置换功能的S盒是很容易遭受攻击的，而一个随机选择的S盒却不具有这样的风险。在许多加密设计中，一定数量的进行置换操作的S盒是必需的或者至少说是希望得到的，而Treyfer恰恰是一个非常有趣的例外。

做的好处是一旦攻击者威胁到了系统而且盗取了密钥，只要管理者交换一条攻击者不知道或者猜不到的消息，攻击者就不能对消息进行解密了，系统可以恢复安全性，这种性质就是有名的“前向安全性”。举个例子，在澳大利亚自动密钥就用在 EFT 的支付终端 [83，85]。使用非对称加密在前向安全性上要求不是特别严格，也就说只要妥协了的终端和攻击者不能控制的没有妥协终端之间能够交换一条消息，即使这条消息非常简单明了，攻击者也无能为力，系统安全性就可以得到恢复，后面将会讨论它的实现。

5.7　非对称加密方法

*非对称加密方法*中用得最普遍的就是公钥加密和数字签名，它们都是基于数论上的。在这里本书只对数论进行简单的介绍，然后在第二部分讨论它的应用，主要介绍实际应用中的几个原理（如果你发现这里边牵涉到太多的数学知识，那么可以跳过这两节，从别的密码学课本上获取这些知识）。

非对称加密技术使得加密的安全性依赖于解决一个特定数学问题的难度，在大多数实际系统中用得最多的是因数分解理论（多用于商业系统）和离散对数理论（多用于军用系统）。

5.7.1　基于因数分解的加密

*质数*就是没有严格意义上的约数的正整数，换句话说就是它只能被 1 和自身整除。按定义规定 1 不是质数，所以质数就是 {2，3，5，7，11...}。算术基本定理说明任何大于 1 的合数都可以分解成一系列质数的乘积，而且其形式是惟一的。找到质数并且把它们相乘得到合数是非常容易的，但是把一个合数分解成一系列质数的乘积则要困难得多。现今最大的由两个大随机质数乘积产生的合数是 512 位（155 个十进制位）长；它进行因数分解需要花费每秒百万条指令的处理器好几千年的时间。近来，一些瑞典学生秘密设法用 512 位数字进行因数分解解决了一个查询加密问题，所以可以说 512 位的合数并不比 56 位的 DES 密钥更“安全”。但是，人们相信如果没有更先进的数学工具，一个 1024 位长度的数字是不能进行因数分解的。

用在公钥加密和数字签名上基于因数分解理论的最普遍的算法是 RSA，是以它的发明者 Ron Rivest、Adi Shamir 和 Len Adleman 的名字命名的 [649]。它使用 Fermat 的（小）定理，即对所有不能整除 a 的质数 p，$a^{p-1} \equiv 1 \, mod \, p$（证明：给出集合 {1，2，…，$p-1$}，它们中的每一个模 p 乘 a，然后每边消去 $(p-1)!$）。Euler 函数 $\phi(n)$ 是一个定义在小于 n 的正整数上的函数，$\phi(n)$ 的值等于序列 0，1，2，...n－1 中与 n 互质的数的个数，所以如果 n 是两个质数 p，q 的乘积，那么 $\phi(n) = (p-1)(q-1)$（证明是类似的）。

加密密钥 N 是一个很难进行因数分解的大数（可取 $N = pq$，其中 p、q 是两个随机选择的大质数），再选择一个公钥 e，它与 $p-1$ 和 $q-1$ 都没有公共因子。而私钥就是 p 和 q，应该保密。设 M 为消息，C 为密文，则加密过程可以定义为：

$$C \equiv M^e \ (mod \ N)$$

解密为相反的过程：

$$M \equiv \sqrt[e]{C} \ (mod \ N)$$

知道私钥（即 N 的两个因子 p、q）的人很容易就可以算出 $\sqrt[e]{C} \ (mod \ N)$，也就可以完成解密。由于 $\phi(N) = (p-1)(q-1)$ 且 e 与 $\phi(N)$ 没有公共因子，密钥拥有者可以找到

数 d 使得 $de \equiv 1 \ (mod \ \phi \ (N))$ ——实际上可以独立地找到 d，对 $(p-1)$ 和 $(q-1)$ 求余，再把它们组合就得到了答案。因此，$\sqrt[e]{C} \ (mod \ N)$ 可以计算为 $C^d \ (mod \ N)$，由 Fermat 定理，解密过程可以表示为：

$$C^d \equiv \{M^e\}^d \equiv M^{ed} \equiv M^{1+k\phi(N)} \equiv M.M^{k\phi(N)} \equiv M \times 1 \equiv M \ (mod \ N)$$

类似的，私钥的拥有者用它来对消息产生数字签名：

$$Sig_d \ (M) \equiv M^d \ (\text{mod} \ N)$$

而且该签名求 e 次幂后对 N 求模就可以得到验证并检查恢复的原始消息（这也就是为什么使用 e 和 N 作为公共签名验证密钥的原因），可以用下面的式子表示：

$$M \equiv (Sig_d \ (M))^e \ (\text{mod} \ N)$$

不论是 RSA 加密还是数字签名，使用它们并不见得就一定是安全的。原因就是它们的加密过程实际上是一个算术处理过程，因此仍然保留着某些算术上的性质。举个例子，如果在明文中存在某种关系，比如 $M_1 M_2 = M_3$，那么同样的关系也有可能出现在密文 $C_1 C_2 = C_3$ 和数字签名 $Sig_1 Sig_2 = Sig_3$ 中。这种性质叫做*乘法的同态性*（数学研究者描述具有这样数学结构的函数为*同态函数*），RSA 的这种同态性也意味着它并不完全符合公钥加密和数字签名中定义的随机预言模型。

有一系列的标准用于阻止这种基于数学结构同态性的攻击，其方法是给算法设定输入的不同部分来固定常数或者随机数。但是它们中的大多数都失败了。比较好一点的解决方法包括用哈希函数以及随机 nonce 和填充（RSA 加密还没有应用时）来处理消息。举一个例子，在*最佳填充非对称加密*（optimal asymmetric encryption padding，OAEP）中，可以把消息 M 和随机数 N 联系起来，使用哈希函数 h 表示就是：

$$C_1 = M \oplus h \ (N)$$

$$C_2 = N \oplus h \ (C_1)$$

从效果上说，这是一个两轮的 Feistel 加密器，它使用 h 作为自己的轮函数，输出结果也就是 C_1 和 C_2 的组合，接着用 *RSA* 进行加密，然后发送出去，接收者用 $C_2 \oplus h \ (C_1)$ 计算出 N，再用 $C_1 \oplus h \ (N)$ 恢复消息 M [88]。

对数字签名而言，事情还要简单一点。总之，在使用私钥：$Sig_d = [h \ (M)]^d \ (mod \ N)$ 之前，即使只对消息进行哈希处理也已经足够了。而在有些应用中，可能希望在签名块中包含更深层次的数据，比如时间戳。

5.7.2 基于离散对数的加密

当 RSA 用在大多数基于 SSL 协议的网页浏览器时，仍然有其他产品（比如 PGP）和政府系统的公钥加密基于离散对数算法。这带来了另外一种风格，其中有些算法使用标准算术，而另外一些则使用一种称作*椭圆曲线*的数学结构。这里只介绍标准形式，因为椭圆情况在本质上与标准算术是一样的，但实现起来更加复杂。

设 p 为质数，g 小于 p，如果对每个 b 从 1 到 $p-1$ 都存在 x，使得 $g^x \equiv b \ (mod \ p)$，则称 g 为模 p 的原根（primitive root）。举一个例子，计算模 7 的原根时，我们发现 $5^2 = 25$，模 7 后得到 4，然后计算 5^3，可以看作 $5^2 \times 5$ 或 4×5，也就是 20，模 7 后为 6，继续这样的过程，如图 5-18 所示。

因此，5 就是模 7 的原根，这也意味着给定任何 y，总是可以得出方程 $y = 5^x \pmod 7$ 的解，x 称作 y 模 7 的离散对数解。对于数字较小的 p，可以用观察法解决，但当 p 为很大的随机质数时，就很难进行计算了，所以可以认为映射 $f: x \rightarrow g^x \pmod p$ 是一个单向函数，但它具有这样的性质：$f(x+y) = f(x)f(y)$ 且 $f(nx) = f(x)^n$。换句话说，它具有单向同态性，因此，它可以用来建立数字签名和公钥加密算法。

$$
\begin{aligned}
5^1 &= 5 \pmod 7 \\
5^2 &= 25 \equiv 4 \pmod 7 \\
5^3 &\equiv 4 \times 5 \equiv 6 \pmod 7 \\
5^4 &\equiv 6 \times 5 \equiv 2 \pmod 7 \\
5^5 &\equiv 2 \times 5 \equiv 3 \pmod 7 \\
5^6 &\equiv 3 \times 5 \equiv 1 \pmod 7
\end{aligned}
$$

图 5-18　一个离散对数计算的例子

5.7.2.1　公钥加密：Diffie-Hellman 协议

为了更好地理解离散对数是怎样用来建立公钥加密算法的，请记住一点：我们期望的加密系统并不需要用户使用一个共享的密钥。请看下面的“经典”方案。

假设 Anthony 想送一封密信给 Brutus，而他们两人之间惟一可用的通信信道是一个不值得信任的信差（比如，一个属于 Caesar 家的奴隶）。这种情况下，Anthony 可以把写好的信放到盒子里并且上锁，然后叫信差把它送给 Brutus，Brutus 收到后又用自己的锁锁好，让信差送回 Anthony。接着，Anthony 就会除去自己的锁再让信差把它送给 Brutus，最后，Brutus 就可以用自己的钥匙打开盒子看信了。

与此类似，任何加密函数都可以实现交换，即具有这样的性质：$\{\{M\}_{KA}\}_{KB} = \{\{M\}_{KB}\}_{KA}$。也就是说 Alice 可以用她自己的密钥 KA 对消息 M 加密得到 $\{M\}_{KA}$ 然后送给 Bob。Bob 接着用自己的密钥 KB 进行加密得到 $\{\{M\}_{KA}\}_{KB}$。由交换属性可知 $\{\{M\}_{KA}\}_{KB}$ 也就是 $\{\{M\}_{KB}\}_{KA}$，所以 Alice 可以使用自己的密钥 *KA* 解密后得到 $\{M\}_{KB}$。她再把 $\{M\}_{KB}$ 送给 Bob，Bob 就可以用自己的密钥 *KB* 解密从而得到消息 *M*。如果这种加密原理用在传统的公钥加密系统中，则密钥 *KA*、*KB* 可能是长期的，如果是为了建立一个带有前向安全性的密钥，则密钥可能会是临时的。

那么怎样实现这种具有交换性的加密呢？如果能够找到数 *p*、*g*，使得以 *g* 为底的模 *p* 的离散对数问题非常难求，那么就可以使用指数函数作为加密函数。举一个例子，Alice 选择一个随机数 x_A，计算 $g^{x_A} \pmod p$ 并把结果和 *p* 一起送给 Bob。类似的，Bob 选择随机数 x_B 并计算 $g^{x_A x_B} \pmod p$，他也把结果和 p 送给 Alice，现在 Alice 就可以去掉她的指数函数了：由 Fermat 定理，她计算 $g^{x_B} = (g^{x_A x_B})^{p-x_A} \pmod p$ 送给 Bob。这时，Bob 也可以移去他的指数函数了，最后就得到了消息 *g*。这种方案的安全性依赖于离散对数问题的解决程度。

实际中，把消息编码成一个原根是一件非常有技巧性的事情，但存在更加简单的方法可以达到同样的效果。第一个发表的公钥加密方案是 1976 年由 Whitfield Diffie 和 Martin Hellman 做出的，他们使用 $g^{x_A x_B} \pmod p$ 作为一个共享密钥加密系统的密钥，x_A 和 x_B 是该系统中双方的私钥。

现在来看看该公钥加密系统是怎样工作的。质数 *p* 和生成器 *g* 对所有用户来说都不是什么很高深的东西。Alice 选择一个秘密随机数 x_A，计算 $y_A = g^{x_A}$，并把它公布在公司电话本她名字的背面，Bob 也是这样，选择随机数 x_B 并公布 $y_B = g^{x_B}$。为了和 Bob 通信，Alice 从电话本上获取 y_B 并形成 $y_B^{x_A}$（也就是 $g^{x_A x_B}$）的形式，对要传给 Bob 的消息加密。在接收端，Bob 找到 Alice 的公钥 y_A 形成 $y_A^{x_B}$（同样也等于 $g^{x_A x_B}$）的形式，这时他就可以解密 Alice 发给他

的消息了。

要提供一个完全的解决方案还需要做更多的工作，比如要仔细考虑怎样选择参数 p 和 g，还有一些细节依赖于你希望系统具有什么样的性质，比如是否需要前向安全性等等。Diffie-Hellman 理论的改进形式包括美国政府的密钥交换算法（key exchange algorithm，KEA）[577]，它用在网络安全产品中，比如 Fortezza 卡，所谓的 Royal Holloway 协议，该协议被英国政府采用并有可能用在第三代移动通信系统中 [50]。

该系统的最大问题是你怎样确认你得到的电话本是一个真实的副本，并且你感兴趣的一些条目还没有过时。本书将会在 5.7.4 节中讨论这个问题。

5.7.2.2 密钥建立

在这样的协议中提供前向安全性机制是相对独立的。与前面的处理一样，设质数 p 和生成器 g 对所有用户都是已知的。Alice 选择一个随机数 R_A，计算 g^{R_A} 并把结果送给 Bob。Bob 做同样的工作，选择随机数 R_B 并把 g^{R_B} 送给 Alice，最后他们都会形成 $g^{R_AR_B}$ 的形式，这就是会话密钥。

Alice 和 Bob 现在就可以用会话密钥 $g^{R_AR_B}$ 加密一个对话了。他们设法创建一个没有任何外界打扰的共享秘密。即使攻击者在这个协议开始之前已经对他们的机器取得了完全访问，因此也知道所有他们存放的私钥，但只要满足一些基本条件（比如，他们的随机数产生器是不可预测的），攻击者就不能窃听到他们的交谈内容。这就是 5.6.2 中提到的前向安全性的较强表现。攻击者不能根据他以前得到的密钥进行后面的窃听工作。假设 Alice 和 Bob 在使用之后同时毁掉了共享密钥，那么他们就具有了后向安全性。随后对他们的机器进行访问的攻击者无法窃取他们以前交谈的内容。

但这个协议有一个小小的问题：即使 Alice 和 Bob 以他们的会话密钥完成整个交谈，他们两个也无法知道相互之间曾共享过这个会话密钥。

假设在本书前面讨论的“上锁”协议中，Caesar 命令他的奴隶把盒子送给他而不是 Brutus，而且他也把自己的锁锁在了盒子上，接着叫信差（奴隶）送回 Anthony，这样，Anthony 就会除去自己的锁再次把盒子送给 Caesar，这样，Caesar 就可以打开盒子看信了。在这种情况下，Caesar 甚至可以成为协议的两个角色，即对 Anthony 假装自己是 Brutus，而对 Brutus 假装自己是 Anthony。对 Anthony 和 Brutus 而言，一个需要解决的问题就是如何在自己的锁上加上签名。

除非能够对临时密钥进行验证，否则上面所说的漏洞将会造成对 Diffie-Hellman 协议的中间人攻击。假设 Charlie 截获了 Alice 传给 Bob 的消息，同时，他还冒充是 Alice 与 Bob 交换了密钥。这样，他与 Alice 共享的密钥是 $g^{R_AR_C}$，他可以用该会话密钥完成他们之间的交谈。同样，他也可以用另外一个会话密钥 $g^{R_BR_C}$ 完成与 Bob 之间的交谈。所以，只要他还继续呆在网络上监听和翻译 Alice 和 Bob 之间的交谈，他们要发现通信已经被破坏就不是一件简单的事了。

在一些安全性的电话产品中，通信双方首先要读取由他们产生的 8 位哈希密钥，并检查双方产生的密钥是不是一致，如果一致，接下来才讨论他们之间的事情。对 Alice 和 Bob 而言，一个更通用的解决方法就是考虑在他们送给对方的消息中签名。

最后要提的一点就是，离散对数及其类似的算法存在于许多其他的数学结构中。比如，椭圆曲线加密就在椭圆曲线上使用离散对数算法，该曲线由方程 $y^2 = x^3 + ax + b$ 给出，在

数学处理上可能要复杂一些，但它们的基本思想都是类似的。

5.7.2.3　数字签名

假设 p 和产生器 g（可能是也可能不是原根）是经过合适方法挑选的公共值，且期望进行签名操作的每个用户都有一个签名私钥 X 和一个公共签名验证密钥 $Y = g^x$。ELGamal 签名方案是这样工作的：随机选择一个密钥 k，计算 $r = g^k \pmod p$。然后建立一个关于 k、r、消息 M 和私钥 X 的线性等式就可以形成签名 s，这样的等式有很多，举其中一个特殊的同时也是 ElGamal 签名方案采用的就是：

$$rX + sk = M \pmod{p-1}$$

所以，s 就可以通过下式计算：$s = (M - rX)/k$，它应该进行模 $\phi(p)$ 计算。当等式两边都用单向同态函数 $f(x) = g^x \pmod p$ 进行处理就得到：

$$g^{rX} g^{sk} \equiv g^M \pmod p$$

或

$$Y^r r^s \equiv g^M \pmod p$$

消息 M 上的 ELGamal 签名包括值 r 和 s，接收者用前面的等式就可以验证它。

要得到一个函数型的数字签名方案要注意许多细节问题。比如，选择不合适的 p 和 g 将使签名算法变得易于攻击。因此可以用哈希函数对消息 M 进行哈希变换，使得可以对任何长度的消息进行签名，那么攻击者就不能使用算法的数学结构来伪造签名过的消息。解决好这些细节并选出其中的一到两个最优化方案，就得到了*数字签名算法*（Digital Signature Algorithm，DSA），这是一个美国标准并被广泛地应用在政府机构中。

DSA（也即 DSS，Digital Signature Standard）假定质数 p 为 1024 位，质数 q 为 160 位，且可以整除 $(p-1)$，参数 $g = h^{(p-1)/q} \pmod p$，h 任选，签名密钥为 x，公用校验密钥 $y = g^x \bmod p$，对消息 M 的签名 $Sig_x(M)$ 就是关于 (r, s) 的函数，这里：

$$r \equiv (g^k \pmod p) \bmod q$$

$$s \equiv (h(M) - xr)/k \bmod q$$

此处使用的哈希函数是 SHA1。

DSA 是没有消息恢复的随机数字签名方案的经典例子。

5.7.3　特殊目的的签名方法

研究者已经发现了大量基于特殊目的的签名和公钥加密方法，本书在这里只描述两种现在使用的方法：阈值签名和盲签名。

*阈值签名*的机制就是：一个签名密钥（或解密密钥）能够从 n 个当事人中分离出来，从而使任何源自于 n 的 k 都可以对消息进行签名（或解密）。对 $k = n$ 的情况，这种结构很容易实现。比如对 RSA 而言，可以把某一个私钥 d 分离出来，这里 $d = d_1 + d_2 + \ldots + d_n$。对 $k < n$，情况要复杂一些（但不是非常复杂）[246]。阈值签名用在有多个独立的服务器进程事务、对结果的投票也独立的系统中。它们也可以实现诸如“七个董事中的任两个都可以对支票签名”这样的商业规则。

*盲签名*用在不知道消息具体内容的情况下对消息进行签名的场合。比如，假设我使用的是 RSA 系统，我就可以选择随机数 R，计算 $R^e M \pmod n$，并把加密后的结果传给签名者，签名者计算 $(R^e M)^d = R.M^d \pmod n$，当把这个结果返给我后，除以 R 我就可以得到签名

M^d，这种方案可能用在数字现金的场合。银行会同意任何正确使用公钥（e，n）的用户拥有一个数字签名，这种签名允许用户拥有任意消息字符串 M，并包含一个惟一的序列号和指定形式的冗余度，但需要用户花费$10。这样的信息段就是所谓的数字硬币，盲签名协议表明客户可以从银行那里得到一个签名“硬币”，而不让银行知道它的序列号，这样的结果对消费者来说就是数字现金是匿名的（这里同样需要考虑很多细节，比如怎样识别使用两次相同硬币的用户，但这些都不难解决）。盲签名和数字现金是由 Chaum 发明的［178］，在本书的第 20 章中，还会看到许多支持数字隐私保密技术的内容［177］。

研究者不断地为专业公钥加密机制找到新的应用。比如在线选举，这是匿名和可计算性的混和，是一个很有前途的应用。

5.7.4 认证

既然可以进行数字签名和公钥加密，那么也需要一些约束用户使用密钥的措施。Diffie 和 Hellman 发明数字签名的时候提出的建议是：经过系统授权的用户应该有一个公钥目录，比如说电话本。根据 Loren Kohnfelder 的理论，更一般的方法是由认证中心（CA）对用户的公用加密和/或签名校验密钥进行校验，给出包括用户名、属性（比如授权）、公钥等等的证书。认证中心可以由本地系统管理员运行，也可以由第三方的服务机构比如 Verisign 公司运行，它的业务就是检验是合法用户后就对用户公钥进行签名。

证书可以用下面的式子表示：

$$C_A = Sig_{K_S}\ (T_S,\ L,\ A,\ K_A,\ V_A)$$

这里使用与 Kerberos 相同的记号，T_S 是证书开始的时间和日期，L 是有效的时间长度，A 是用户名，K_A 是他的公共加密密钥，V_A 是其公共签名校验密钥。在这种情况下，只有管理员的公共签名校验密钥需要通过可信的途径与所有用户进行通信。

分布式系统的认证在第 6 章“分布式系统”介绍。电子商务中的应用在第 19 章“保护电子商务系统”介绍。认证的策略在第 21 章“电子策略”中介绍。本书在这里只指出协议的实现要远比想像中困难。

第一批被采用的公钥协议中的一个协议是由 Dorothy Denning 和 Giovanni Sacco 提出的，他们在 1981 年建议，两个用户，比如 Alice 和 Bob 通过下面的步骤建立共享 DES 密钥 K_{AB}。当 Alice 第一次与 Bob 通信时，她从认证中心得到当前她和 Bob 的公钥证书副本，然后把时间戳 T_A、会话密钥 K_{AB}和用自己私钥加密形成的签名打包成一个密钥包。再用 Bob 公布的公钥进行加密后传送给他，用符号表示就是：

$$A \rightarrow B:\ C_A,\ C_B,\ \{T_A,\ K_{AB},\ Sig_{K_A}\ (T_A,\ T_{AB})\}_{K_B}$$

在 1994 年，Martín Abadi 和 Roger Needham 就指出这个协议有致命的缺陷［2］。比如，Bob 收到消息后，只要 Alice 的时间戳 T_A 保持有效，他就可以伪装成 Alice。来看看他是怎样实现的，假设他想伪装成 Alice 跟 Charlie 通信，他就从认证中心那里得到一个与 Charlie 进行通信的新的证书 C_C，然后从签名协议的消息 3 中去掉外面的加密 $\{\ldots\}_{K_B}$，接着用 Charlie 的公钥（他从 C_C 那里得到的）对密钥包 T_A，K_{AB}，$Sig_{K_A}\ (T_A,\ K_{AB})$ 进行签名从而得到伪造的消息 3：

$$B \rightarrow C: C_A, C_C, \{T_A, K_{AB}, Sig_{K_A}(T_A, K_{AB})\}_{K_C}$$

实际上，一个如此简单的协议——本质上也就是一个单行程序——能够有这样严重的缺陷而在很长一段时间都没有被发现，这是非常值得重视的。而对一段只有几行代码程序的纠错，你可能只需要一到两分钟就能够找到它的 bug，相比之下，对公钥协议的纠错要困难得多。事实上，用共享密钥加密的协议设计起来也是非常困难的，因为它们很容易受到精心设计的有害的中间人攻击，但这有助于推动用形式化方法来证明协议是否可靠。

通常，在认证机制建立过程中当事人的名字不是最重要的。在美国政府和国防项目承包人使用的 STU-III 安全电话中，就有一个建立具有前向安全性和后向安全性的临时密码的协议。为排除中间人攻击，用户有一个加密引导密钥（crypto ignition key），它是一个可以插进电话中的便携式电子装置，不仅可以识别用户的名字还可以识别他们许可的安全等级。一般而言，关于这个主题的书籍总是试图把身份识别当作认证和密钥管理协议的主要目的，而在实际应用中，通常都是对行为进行授权。这往往要比身份识别复杂得多，因为它需要把应用中的一些假设加进协议设计中（事实上，美国国家安全局的安全手册强调随时知道是否有不明身份的人进入系统的重要性，STU-Ⅲ设计就是把这一原则很自然地扩展到电子通信领域的一种方法）。

依赖于公钥证书的一个严重缺陷就是使用户理解所有的密钥并合理地管理它们存在很大的困难，尤其当它们不是用户熟悉的手工控制系统且不能重复实现时 [224]。从系统工程的角度来看，认证也可能存在一系列问题，这些留到本书下一章进行讨论。

5.7.5 非对称加密方法的强度

为提供与对称分组密码相同的保护级别，非对称加密需要的分组大小至少应该是对称加密的两倍。椭圆曲线系统就是这样实现的。一个 128 位椭圆曲线加密方案的受攻击程度与一个分组大小为 64 位、密钥长度也是 64 位的分组密码相当。基于因数分解与离散对数理论的一般加密方案更容易受到攻击，因为存在快捷的攻击算法，比如数域筛选法（number field sieve）攻击，它基于这样一个事实：自然数中存在平滑数（即有许多小因子的自然数）。在写入时间上，使用数域筛选攻击的密钥长度已经达到了 512 位，这是与用密钥搜索方法对 56 位的 DES 密钥进行搜索难度相当的攻击。目前一致认为对 RSA 和标准离散对数加密系统来说，私钥长度至少应该为 1024 位，而对于那些为算法做出重大改进的数学家来说，2048 位长的密钥才能给出一些有用的安全系数。

近来有一些公开的出版物讨论量子计算机。即，使用叠加的量子态，可以使大量计算同步进行。Peter Shor 已经证明如果能够建立足够强大的量子计算机，那么因数分解和离散对数计算将会变得非常容易。但到目前为止，也只能建立非常小的量子计算机，所以很多人对能否有足够先进的技术提高量子计算机运行性能从而威胁到实际应用的系统持怀疑态度。如果它可以的话，那么非对称加密就会被淘汰。而幸运的是，到目前为止，许多用对称加密可以实现的系统同样可以用非对称加密实现，因此使用 Kerberos 理论的改进形式同样可以重新设计许多认证协议。

5.8 小结

许多加密方法失效是因为对它们的使用不正确，所以需要一个清楚明白的模型来告诉我

们加密到底是怎么实现的。随机预言模型就是这样一个有用的模型：它假定每个由加密器返回的新值在统计意义上都与以前所有的输出独立，并且是随机的。

对称密钥应用中的分组密码可以通过仔细的替代和置换操作得到；而对于非对称应用，比如公钥加密和数字签名来说，其理论基础就是数论。在这两种情况下，都有非常多的数学知识引导我们学习。其他的加密方法——序列密码和哈希函数——可以通过合适的操作模式从分组密码中构造。这些加密方法都有不同程度的错误传播、模式伪装和完整性保护属性。

即使常常会出现一些小问题，安全工程的基本性质也并不是难于掌握。在具体实践中，即使是一个系统的某部分失效（或受阻）。要修复系统使它坚固也是一件非常难的事情，而要把加密原理与其他安全措施（比如访问控制和物理安全措施等）很好地连成一个整体也是相当困难的，在后面的章节中，本书将会重新提到这一点。

研究问题

在密码学研究领域，有很多非常活跃的研究思路。其中的大多数是数学研究的一个特殊分支（比如数论、代数几何学、复杂性理论、组合数学、图论、信息论等）。商业事务中的研究往往集中在加密算法、数字签名、组合操作等的设计，因此可以在可用平台上运行良好。有些研究的范围很广，从线性和差分解密到如何对公钥加密协议进行攻击都有。研究更多的是由现今存在的知识主体而不是由应用来推动的，当然也有例外：对版权保护的忧虑就是一个实际的推动力，近来发现高级加密标准的竞争也属于这种情况。

要把握当今密码学研究的走向，最好的办法就是阅读近几年研究会议论文集，比如Crypto、Eurocrypt、Asiacrypt、Fast Software Encryption 等等，它们都由 Springer-Verlag 出版社收集在《Lecture Notes on Computer Science，LNCS（计算机科学论文备忘录）》中出版。

参考资料

由 Diffie 和 Hellman [248]，Rivest、Shamir 和 Adleman [649] 发表的经典论文是关于密码学方面最应该读的文章。最畅销的密码学入门书是 Bruce Schneier 的《应用密码学》[681]，它包括了许多不需要很多数学基础就可以理解的内容，而且还有很多算法的 C 程序源代码。《Handbook of Applied Cryptography（应用密码学手册）》是 Alfred Menazes，Paul Von Oovschot 和 Scott Vanstone [544] 写的，在数学细节描述上，它是最完美的标准参考书。

更多的专业书包括：由 Eli Biham 和 Adi Shamir [102] 写的关于差分解密的标准参考书；最好的解释线性解密的由 Doug Stinson 写的课本 [738]；现代分组密码理论可以在 20 世纪 90 年代“Fast Software Encryption”的会议论文中找到（Springer-Verlag 的 LNCS 系列中出版）；关于操作模式的最初著作是 Carl Meyer 和 Steve Matyas 写的 [548]；Neal Koblitz 有一本介绍公钥加密数学基础的书 [463]；数域筛选攻击在 [497] 中有描述；量子分解在 [698] 中有介绍。

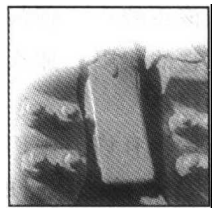
关于随机预言模型和密码学的一般理论方面好书不多。我见过的所有出版物几乎都技术性很强而且很难看下去，被认为最好的也许是由 Oded Goldreich 写的：在 [342] 中可以找到在线碎片方面的知识。如果你需要一些有关 ISBN 方面的东西，去找他的论文《Modern Cryptography，Probabilistic Proofs and Pseudorandomness（现代密码学、概率证明和伪随机性）》[343]，这篇文章常常被数学系研究生引用。在随机性和算法方面讲得不是特别彻底但可读

性非常强的书在［360］中。当前关于密码学理论的研究可以在 FOCS、STOC、Crypto、Eurocrypt、Asiacrypt 的会议上找到。

密码学的历史是十分诱人的，许多老问题现在也层出不穷，所以安全工程师也应该对它们非常熟悉。David Kahn［428］完成了一些标准化工作；在［229、227、228］中有一些来自 Cryptologia 的历史论文汇编，还有一些二战时期密码学历史方面的书［188、429、523、800］；位于 Fort George Meade，Maryland 的国家安全局博物馆也值得一看。

最后要提一下的是，本书中没有任何一章完整地介绍了公钥加密，因为都没有提到“非机密加密”，它是 James Ellis 在 1969 年左右发现的。然而，由于 Ellis 是为 GCHQ——英国政府通信总部，相当于美国的国家安全局——工作的，他的工作是保密的。不久 Clifford Cocks 发明了 RSA 算法，同样处于保密状态。这些故事在［267］中有记载。保密的结果就是他们所作的工作不能被采用——即使由于军队密钥分配开支的推动，英国国防部仍然直到 1992 年才开始为他们的主干网建立电子密钥分配系统。同样需要注意的是，美国国家安全部门并没有预先发明数字签名，他们只是保留着 Whit Diffie 和 Martin Hellman 的研究成果。

第6章　分布式系统



当一台你从未听说过的电脑的崩溃使你的工作陷入瘫痪时，你就知道你使用了分布式系统。

——Leslie Lamport

在前面几章中，可以看到人们怎样使用安全协议向系统验证自己（以及系统之间如何相互认证）；访问控制如何用来管理一个系统中的用户可执行的操作；还有密码如何用来支撑分布式系统中访问控制的一些技巧。但是在搭建一个安全的分布式系统时，有更多的事情需要考虑，而不仅仅是实现访问控制、协议和密码。当系统变得庞大的时候，所出现的问题是呈非线性增长的；常常在复杂程度上会发生质变，而在少许机器和用户构成的网络中微不足道的问题（例如命名）会一下子变得不容忽视。

在过去的35年里，计算机科学的研究者们已经建立了许多分布式系统，并研究了诸如并行、故障恢复、命名等课题。相关理论也随着工业、商业和行政管理的经验积累而不断补充完善。对于设计有效的安全系统来说，这些问题是设计的核心，但却往往被相当拙劣地处理。本书前面已经描述过，对于安全协议的攻击可以看作是并行的失败。如果需要复制数据以实现一个容错系统，那么在对机密性的妥协上所承担的风险就可能增加。最后一个因素是，命名上的困难很有可能成为构建公钥基础设施（public key infrastructures，PKI）的主要障碍。

6.1　并行

如果多个进程在同一时刻运行，它们就被称为是“并行”的，并行方式带来了许多已经被深入研究过的问题。进程可能使用陈旧的数据；可能造成前后不一致的更新；更新的顺序可能重要，也可能不重要；系统可能进入死锁状态；不同系统中的数据可能永远也不能聚集成连续的值；遇到需要得知确切时间的关键时刻，可能会比你想像中要困难。

从总体上来说，设计并行系统是一个艰巨的任务；而不幸的是，课本中的大部分例子都出自操作系统内部构件和线程管理这样一个相对单薄的范围。但是并行控制同样也是一个安全问题；它像访问控制一样用来阻止用户之间有意或无意的干扰。并且，并行问题在一个系统中可以表现为从硬件到商业运作环境的许多级别。在接下来的内容中，将提供尽可能多的具体事例来说明并行方式在安全问题上的效能。当然，绝对的详尽是不可能的。

6.1.1　使用陈旧的数据与呈扩散状态的花费

前面已经描述了两种并行方式的毛病。第一，进攻者可以通过过期的凭据对协议进行反复攻击。第二，存在竞争的情况。前面也提到过Unix下mkdir命令的脆弱性，在这类操作系统下一条分为两个阶段执行的特权指令在进程的中途会以重新命名它作用的对象方式遭到攻击。这些问题已经困扰人们很长一段时间了。IBM的OS/360是最早的多用户系统之一。在

该系统中，任何尝试打开文件的企图都使其可读，并进行权限检查；如果用户被授予有访问它的权限，那么该文件又被读了一遍。用户可以设置事件使得在这个过程中能够更改文件[493]。

有许多实例是关于检查时间到使用时间（time-of-check-to-time-of-use，TOCTTOU）的攻击（关于这类攻击的系统方法可以参见[107]）。不过，阻止它们并不总是很省事，正如在安全状态下扩散变化的花费比较大一样。

举例来说，银行业管理着所有热卡（hot credit card）（不论是被偷窃还是被废弃）的清单；但全世界有数百万这样的信用卡，所以不可能每个商业终端都存有一份完整的无效信用卡清单，而验证该卡与银行之间的所有交易会过分昂贵。好在我们有多级的替代处理方法。终端被允许在一定限制范围（floor limit）内离线处理交易；大型交易需要掌握了所有无效的本地信用卡和已经被废弃境外卡的本地银行进行在线检验；在另外一种限制级别可能需要通过类似 VISA 这样拥有更庞大的国际清单的机构认定；而最大类型的交易则需要卡的发行者的证明。实际上，只有地方交易和大型交易才被立即检查。

信用卡令人感到有趣，因为当人们开始建造公钥证书的底层基础设施、以支持基于安全套接字协议层（SSL）的网络购物和基于 Windows 2000 的公司网络时，恐怕最大的一笔开销将用于撤销那些凭据已经变动的当事人的公钥证书——因为这些人或者改变了居住地址、或者换了工作、或者他们的私钥被窃取，或其他任何原因。信用卡网络是现存最大的管理涉及全球安全问题的系统——这样的系统通常基于多数交易事件是本地化的、低额的或两者兼备的假设。

6.1.2 通过锁定防止不一致的更新

当多人同时对一个文档进行操作时，他们可能使用同一产品——比如 RCS——来确保在某一时刻，对文档的某个给定的部分，只有一个人具有写的权利。这就说明了“锁定”作为一种管理资源——比如文件系统——一致性和减少更新冲突可能性的方式的重要性。另一个技巧是反馈；服务器可以存有那些依赖于它的安全状态的客户列表，并在状态发生变化时通知它们。

在安全的分布式系统中同样也存在问题。以信用卡为例。如果我有一家饭店，一位顾客在入店登记时出示了信用卡，那么我就可以向管理该卡的公司申请预核准，用来记录近期内我要提出一笔款项的事实；我可能会从她可用的信用中登记“达 500 美元”的款项声明。如果该卡是被撤销的，那么第二天，她的银行就会打电话给我并请我与警方联系或是强迫她支付现金（我的银行可能会也可能不会为我担保这笔钱；这完全取决于我与银行协商的合约类型）。这是一个如何在分布式系统中建立突发授权体制（更多细节参见[65]）的公开注册通知模型的例子。

而反馈机制不能提供一种通用的解决方案。凭据发布方可能不想实施反馈服务，客户也会因为被告知她的财务收支情况而从个人立场反对这种方式。以护照作为例子。在许多国家，政府颁发的身份证号码在许多交易场合被要求出示，但政府却不提供任何保护措施，而多数公民就会因为政府存有每次政府颁发的身份证号码被出示时的记录而提出抗议。

总之，那些给发行物（比如信用卡）带来债务的凭据和其他凭据（比如护照）是有区别的。区别之一就是更新顺序的重要性。

6.1.3 更新的顺序

如果两笔大型交易到达政府的银行账户，比如说一笔$500 000的存款和一笔$400 000的取款，它们提交申请的顺序可能不是那么重要。但如果它们针对的是我的银行账户，那么这个顺序就会对结果产生巨大的影响！事实上，决定交易处理顺序的问题没有不出现错误的解决方案。与之紧密联系的是如何处理并行计算的问题，而绝大多数建立高效的分布式系统的技巧在于安排事情的先后，以使进程之间或者是简单的串行关系，或者是彻彻底底的并行关系。

零售业的支票账户系统的常用交易算法是彻夜地分批处理交易，以及在处理所有的取款申请之前先处理所有的存款申请。这种方式所带来不可避免的副作用是被撤回的付款必须被撤销掉。实践表明，失败的付款链会终止，尽管在理论上这并不一定成立。为了限制这种系统性的冒险，以防止未终止的付款取消链破坏全球的银行系统，一些银行间的付款转为采用实时总额处理（RTGS）机制，以此来使交易通过预定的方式到达。不利之处在于处理结果依赖于变幻莫测的网络。信用卡以信用额度实时运行或近似实时运行（每次授权都会减少可使用的信用额度）这两种策略混合运作，而处理过程却像是在支票账户中执行一样。这种方式的缺点在于要通过预核准（大笔交易），才能使用你的卡交易。

现金账户方法近来已成为并行系统研究的课题。其主要思想是后来提交的离线申请向主备份提出尝试性的更新交易。各式各样的技术可以用来防止不稳定性；用于尝试性更新的机制，比如银行的日志，也特别重要[352]。

在其他系统中，交易到达的顺序就远没有那么重要了。护照是一个很好的例子。护照发行者只关心护照的生效日期和失效日期，而不是盖在它们上面的签证顺序。

6.1.4 死锁

死锁是另外一个问题。事情可能因为两个系统相互等待对方采取行动而变得很糟糕。一个关于死锁的众所周知的说明是哲学家就餐问题。一些哲学家围坐在一张桌子旁；每人的左边有一根筷子，而惟一能使他进餐的方法是将两边的筷子都拿起来。如果他们都试图立刻进餐，并拿起了——比如说——右边的筷子，这时死锁现象就发生了（有关这个问题及其解决算法请参见Dijkstra的经典论文[251]）。

当你遇到多层锁时，事情的复杂程度就变得十分可怕，而这些锁分布于已有部分系统失效的（特别是那些意味着锁不可靠的）系统中。许多此类问题都被写入到分布式系统的文献中。参见[64]。但它并不只是一个技术性问题；在商业活动中有很多这种尴尬的情况出现。只要过程是人为可以控制的，那么也许会捕捉到一些伪造的东西，但当它是用软件来实现时，可能就无法有效应付这类问题了。

有时去除这样的伪造是不可能的。在一个大家熟知的商业问题——形式之争——中，一个公司按照自己的方式以一定的顺序发送，另一个公司也按照自己的方式接收，那么合同就是由没有达成一致的贸易过程来控制的。如果贸易的方式更为电子化，那么双方的承诺就会变得更糟糕。

6.1.5 不收敛的状态

在设计用于更新分布式系统状态的协议时，惯用的尝试是ACID——交易应该是微小的、

一致的、独立的和持久的。如果交易是微小的，实质是你可以“完全处理或根本不予理睬”——这样就使得系统失败后的恢复更为简单。如果交易是一致的，那么就会有一些不变的事物能够维持下来，比如账本总是保持收支平衡。这在银行系统是很普遍的现象，并以坚持对一个账户的存款与对另一个账户相等和相反的取款搭配（本书将在第9章“银行业和簿记系统”中深入地讨论）的原则实现。如果交易是独立的，那么它们彼此看起来都是一样的，也就是说，是可以串行化的；而如果它们是持久的，那么一旦经过了处理，就不能再撤回到原来的状态。

这些性质可能太多、不足，或者二者兼而有之。它们当中每一条都会失败或者遭到许多来历不明的攻击，把系统设计成收敛的通常是有效的。这意味着，如果交易额变小，最后会全部变成一致的状态 [565]。收敛通常通过时间戳和版本号等语义技巧实现；交易附加到数据文件的后面而不用重写就足够了。

但是，在现实生活中，肯定有恢复出错和没有完全恢复的交易的方法。安全或者审计管理人员的生活就是跟大量信息的长期战争：明显的赤字（和盈余）在增加，但是有时就是不能解释。例如，不同的国家系统对银行交易中哪些领域是强制性或者可选择性的持有不同看法，因此常常为了付款处理能正常运作而不得不估计数据。有时估计发生了错误；有时会发现利用了直到后来才明白的（如果可能的话）漏洞。最后，事情通过增加一个修正因子会变得一团糟，叫做“分支相异”，而且设立一个维持在低于特定年度阈值的目标。

前面提及的冲突形式给出了一个关于分布式非电子系统不收敛的例子。

在军事系统中，有一个更深的问题涉及到用户请求一些数据但又不能提供给他们。例如，有人可能会问一艘运载军队到伊朗执行秘密任务的军舰的目的地。如果用户不允许知道这些，系统可能通过编造一个伪装故事来隐瞒军舰执行秘密任务的事实（这些问题本教材在第7章“多级安全”里有讨论）。

6.1.6 安全时间

引起安全工程师特别兴趣的最后一类并行问题是准确时间的提供。像 Kerberos 这类认证协议可以通过时钟中的一个错误而被攻击，在网络中仅仅相信一个时间源是不够的。对于网络认证，有一个危险的只依赖安全时间的递归操作，由于主时钟信号必须由其自身认证。如果不能得到正确处理，就会发生某种不良事件，这就是一个“灰姑娘攻击”。如果一个安全关键程序（如防火墙）有一个带有时间锁的许可证，坏人（或者病毒）就能拨快你的钟从而“使软件变成废物”。

有几个可能的途径：

- 可以用无线电时钟装备每一个计算机，但花费是巨大的，并且无线电时钟——即使是 GPS——也能被堵塞，如果对手很强的话。
- 在研究文献中描述过的时钟同步协议，设计成让大量时钟通过“投票”来防止时钟失效和网络延迟。即使这些技术被设计用来抵挡随机（而不是恶意的）攻击，它们通常还通过对消息进行数字化签名得到加强。
- 你可以舍弃绝对时间而使用 Lamport 时间替代，这意味着你所关心的问题仅仅是使事件 A 在事件 B 之前，而不是具体日期 [486]。在安全协议中使用查询—响应而不是日志就是一个例子；另一个例子是有关时间戳服务持续将所有提交给它们文件进行

哈希变换到已经公布的运行总数中，并且能提供特定文档在某日期前存在的证据[364]。

在许多应用中，也许不再使用网络时间协议（NTP）。这需要用时间服务器的时钟选举和认证进行适度的保护。对于许多应用来说这是足够的。

6.2 容错和故障恢复

故障恢复常常是安全工程中最重要的一个方面，也是一个最易忽视的方面。很多年来，大多数关于计算机安全方面的文章都是有关保密性，而其余的大部分是关于认证和完整性；可用性则被忽视。但是一个银行的实际花费恰好相反。可能所有IT方面的第三大花费开支是在可用性和恢复机制上，像热备份处理站点和多层次冗余网络；还有部分将投资在像网络审核等完整性机制上；几乎所有重大的花费都用在加密盒等保密性机制上。通读这本书后，还将发现很多其他应用，从电子战的防盗警报器到保护一个公司不受基于因特网的拒绝服务攻击，这些都是在可用性上很基础的应用。容错和故障恢复占安全工程师工作中很大的一部分。

经典的系统容错通常是建立在像日志和锁定这样的机制上的，当这些机制必须从面临的恶意攻击迅速恢复时需要采取很复杂的措施。它在很多方面涉及到安全问题：失败模型，自然恢复能力，提供的冗余备份地点和抵抗拒绝服务攻击的防御。可以采用下面的定义：失效将导致错误，这是一种不正确的状态；而错误将导致故障，这是一种对系统指定行为的偏离。在一个系统中构建容错和故障恢复将有很多功能组件，像失效检测、错误恢复，必要的话还有故障恢复。故障前平均时间（mean-time-before-failure，MTBF）和修复平均时间（mean-time-to-repair，MTTR）的意思很明显。

6.2.1 故障模型

为了确定哪种恢复类型是我们需要的，必须知道针对我们系统的攻击类型。本节大部分内容将来自针对操作系统威胁的分析中，但是还有一些常见问题需要提及。

6.2.1.1 拜占庭式失败

首先，我们关心的故障类型是普通的或者拜占庭式的。拜占庭式故障模型受启发于n个保护拜占庭的将军，t表示被土耳其人收买在领导机构导致无穷混乱的叛逆者。将军可以通过送快信的人传送口头信息，而且这个送信人是值得信任的。任一个将军可以和另一个将军交换机密和授权信息（可以认为他们对每一条消息都加密并计算一个MAC）。那么多大的叛逆者数目t是可以忍受的？

观察的关键是，如果仅有3个将军，叫做Anthony、Basil和Charalampos，并且Anthony是叛逆者，他告诉Basil“我们进攻”，而告诉Charalampos“我们撤退”。Basil可以告诉Charalampos“Anthony说我们进攻，”但是这不足以让Charalampos确定Anthony是叛逆者。那也很可能是Basil；Anthony可以对他们都说“我们撤退”，但是Basil撒谎说“Anthony说我们进攻”。

这个漂亮的见解归功于Lamport、Shostack和Peace，他们证明这个问题仅当 $n \geqslant 3t+1$ 时有解[487]。当然，如果这些将军能标记他们的消息，那么将没有一个将军敢对两个不同的成员说不同的事情。这说明了特殊数字签名和普通端到端安全机制的力量。通过第三者介绍

当事人或者在他们之间进行交易将节省很多，但是如果第三者变得不可信任那就会增加巨大的花费。

6.2.1.2 容错的相互作用

可以通过一些途径来限制故障率。最常见的是通过故障阻止器和冗余。它们都可以使系统更加具有恢复能力，但是它们产生的效果截然不同。简言之，这两个机制都可以有效地保护数据的完整性，但故障阻止器面临拒绝服务攻击时显得更加脆弱，而冗余方式则使保密性难以达到。如果多个地点有备份数据，那么任何一个受到威胁，安全机密性都将被破坏；如果我有一些数据，而且根据法庭的要求我必须毁坏它，但从所有备份磁盘上清除它将是很困难的。

简洁地说，当复制提供完整性和可用性的时候，也削弱了机密性泄露的抵抗力。以后会再讨论这个问题。事实上，在商业领域中阻止复制和在军事系统中抗干扰抵抗力，分别反映了它们不同的保护优先级。

仍然存在一些不引人注意的陷阱。在一个案例中我作为专家被邀请，我的客户在一个商店使用信用卡时被捕，被指控有一张伪造的卡，而且惊动了警察。他坚称这卡是真的。很久以后，这张卡通过 VISA 检验后发现确实是真的。我们推测是什么原因引发这种结果。在信用卡磁条上有两种冗余数据：一个是简单的校验和通过使用异或操作合并磁轨中所有字节得到，另一个是加密校验和，这将在 19.3.2 节中详细说明。前者是发现错误，后者则是发现伪造。看起来情况是，这个商人的读卡机离线后在某种情况下引起某偶数位出错，在简单校验和中偶然删掉某一个校验位，而使加密校验和检测失败。这是一个失败的警告，给客户的生活造成很大的干扰。

6.2.2 恢复什么

当在一个系统中引入冗余或者其他恢复机制，我们需要非常清楚这样做的目的。一个重要的考虑就是恢复是不是仅限于一个单一的组织。

第一，复制是使服务器更加可信的内在特征。AT&T 已经建立了一个叫 Rampart 的系统，其中一些地理位置不同的服务器能够分别进行计算，通过阈值解密和签名把结果合并起来[639]；这个想法被用来完成密钥管理等任务 [640]。IBM 提供了一个叫做主动安全的不同想法。在这里，密钥将很流畅地通过系统，不管是否有攻击被告知 [379]。这个方法能够使系统从攻击者破坏一个服务器并且威胁到其他的服务器中恢复过来。在很多便宜的商用机器上建立一个安全的“虚拟服务器”已经开始吸引人们设计认证授权服务，因为关于攻击和错误的充分证据表明在其中一台服务器上这是可能实现的服务 [211]。这种方法同样引起了许多国家海军的极大兴趣，其原因在于重要的资源可以通过多台微机而传遍整艘舰船，并从那些大多数类型、不会导致沉船的破坏中恢复过来 [309]。

但是事情常常是非常复杂的。服务器必须保护自己不受恶意用户的攻击。举例来说，一个谨慎的银行将会考虑到很多客户会有机会采取欺骗行为。有时，问题会是另一种方式，我们必须依赖很多设备，每一部分都不可完全相信。比如，在一个没有 ID 卡系统的国家，一个零售商想扩展一个客户的信用可能会询问这个客户三个不同的问题（比方说，燃气账单、电话账单和薪资明细表）来证明客户的姓名和地址。

不信任的导向对协议的设计有影响。服务器将面对很多不可信的客户，而且客户可能会

依靠很多不合格的、无效的或者恶意的服务器，它们都希望在协议中控制信息的流动，目的是抵御拒绝服务攻击。这样，一个面对很多不可信服务器的客户可能希望用一个认证协议，像前面讨论过的 Needham-Schroeder 协议；那样，客户用旧的服务器认证票据将不再是一个漏洞而是一个特征。这个想法可以用在一般的协议设计上 [623]。它给我们提供了协议为什么会失败的另一种解释，是由于设计的原因，还是由于缺少诈骗投资的原因，这是不同的；而且也解释了为什么要为真实世界设计系统，如果所有的主机都不可靠或者特别可疑时，将是很困难的。

在高一点的层次上，重点是安全更新能力。付费电视是一个很好的例子：密钥和其他的订户管理工具通常保存在一个廉价的智能卡里而不是一个昂贵的置顶盒中，因此即使所有的密钥都泄露了，操作者也能够通过发放新卡给订户来恢复安全。在第 20 章“版权和隐私保护”中将会有详细的介绍。

6.2.3 冗余在什么层

以不同的级别从错误、攻击和设备故障中恢复。比如访问控制系统，当系统达到更高层时将会变得更加复杂和不可靠。

一些计算机将冗余备份建立在硬件层，像多 CPU 和镜像的磁盘，目的是减少故障发生的可能性。从 80 年代后期，这些机器广泛应用于交易过程的任务中。一些更加现代的系统用大量的并行服务器来实现这个目标；*廉价磁盘的冗余阵列*（redundant arrays of inexpensive disks，RAID 磁盘）是一个类似的概念。但是没有一种技术能够对入侵者提供防护，更不用说有故障或者恶意的软件了。

下一层上是*进程组冗余*。在这个层次上，我们在不同地点的很多服务器上运行一个系统的多个备份，让它们通过表决来输出运行结果。这能阻止那些通过物理途径访问机器企图摧毁系统的攻击，或者是通过机器破坏，或者是插入未授权的软件并破坏或者修改数据来达到目的。但它不能抵抗授权用户的攻击或者恶意的授权软件的破坏。

下一层是备份。在这个层次上通常按有规律的时间间隔进行一次系统（又称为检查点）的复制。备份副本通常保存在不能覆盖的媒体上，像写保护的磁带上或者 CD 上。也可以保存检查点之间的所有交易的日志记录。一般来说，系统通过日志到来数据的事务处理策略变得可恢复，试着进行交易，并且记录日志，再检查看是否可行。无论什么样的细节，备份和恢复机制不仅可使我们从物理资源破坏中恢复，也使人们相信如果受到一个逻辑层的攻击，比如软件中的一个时间炸弹在特定日期删除客户的数据——也是有希望恢复的。但这些机制并不是完全没有错误。所知道最近的银行碰上的灾难性计算机故障是随着时间的推移，主机软件变得混乱，将会导致银行业务中断，并且几个星期不能恢复正常运行。

备份不同于*后退*，后退系统通常是指当一个主要系统不可行的时候恢复运行能力较差的系统。一个例子就是当电子终端出故障时使用手动“zip-zap”机器来处理信用卡交易。

后退系统是备份冗余机制在应用层的一个例子——机制所能应用的最高层。它要求一个受限的交易必须通过两个职员认可，在所有的交易中保持审计追踪，和一些其他的事情。在第 9 章中将会更加详细地讨论这个问题。

认识到硬件冗余、进程组冗余、备份和后退是不同的机制、能做不同的事情是很重要的。冗余磁盘不能阻止一个恶意的程序员删除你的所有账目文件；而备份不仅不能阻止删除

文件，也不能阻止他缓慢写入越来越多的错误代码。硬件冗余和备份都不足以抵抗对于数据机密性的攻击。另一方面，如果数据处理中心全面瘫痪，世界上最好的加密手段也不起作用。实际恢复计划和机制将会十分复杂，并且将会包含以上提到的所有内容。

6.2.4 拒绝服务攻击

我们想要安全服务能够容错的原因之一是使拒绝服务攻击降低诱惑力、更加困难或者同时具有这两种效果。这些攻击通常是一个大攻击计划的一部分。比如，可能通过堵塞一个主机使它暂时掉线，接着由另外一台在同一局域网（早被破坏的）上的机器接替被阻塞主机的身份一段时间。还有一种可能的攻击是攻破一个安全服务器强迫其他的服务器用缓存的凭据副本。

抵抗拒绝服务攻击的一个非常有力的防护是阻止对手运行一个选择性的攻击。如果重要资源是匿名的——或者至少没有名字服务告知对手将攻击哪里——攻击可能会无效。这些在本书有关防盗报警与电子战的内容中将有进一步的讨论。

如果做不到这一点，并且对手知道攻击哪里，那么有些类型的拒绝服务攻击可以通过冗余和恢复机制来阻止，另外的攻击则无法阻止。比如，TCP/IP 协议对主机就很少有有效的机制来保护它们免遭不同的网络泛洪攻击。一个对手可以发送很多连接请求，从而阻止其他人建立连接。对抗这类攻击倾向于追踪和逮捕犯罪者。

最近，在网上有软件可以帮助对手攻击一系列没有防护的系统并且用这些被攻击系统作为向受害者发动泛洪攻击的遥控系统。在第 18 章“网络攻击与防御”中将讨论这个问题。现在，只能说阻止这类攻击很困难，而且复制不是彻底解决问题的方法。如果只是转到一个备份机器，并将其告诉名字服务器，则名字服务器将很容易把新的 IP 地址告诉攻击软件，就像告诉任何其他人一样。但如果备份机器有充分的能力并能更好地应付负载，那么备份就是一个有用的策略。比如，你可能会故障切换到一个高容量的网络主机服务上。这有点类似于“后退”的相反概念。

最后，当存在一个更加脆弱的后退系统时，一个常用的技术就是通过拒绝服务攻击来破坏该系统的使用。经典的例子是在法国和挪威等国家使用智能卡进行银行付款。智能卡比磁条卡更难伪造，但是每年仍有 1% 的故障，比如静电之类的环境损坏原因。而且，不少外国旅游者仍使用磁条卡。因此智能卡付款系统需要一个后退模式来进行传统操作。许多攻击都是针对这种后退模式的。一条破坏智能卡芯片的诡计是把它连接在电力网上；更常见的方法是采用从外国旅游者那里偷来的信用卡、或者从仍然普遍使用磁卡的国家的罪犯那里进口信用卡。同样，如果一个攻击者可以破坏网络，依靠网络连接基本响应和后退到警铃的安全警报器将会非常脆弱。很少有人注意警铃。

6.3 命名

命名在通常的分布系统中既是一个麻烦之处，也是不重要的部分，但是在安全工程中却变得出奇的困难。一个典型的例子（如千年问题）就是将哪种名字放在公钥证书上的问题。一个证书仅说，“名叫 Ross Anderson 的人允许管理系统 X”是没有用的。在互联网能使用搜索引擎之前，我是我所知道的惟一的 Ross Anderson；但现在我知道有许多 Ross Anderson，我也知道不同的名字对应不同的系统。名字存于上下文中，并且命名当事人在安全系统中变得

重要且困难。

这只是一些分析。所遇到的大多数（虽然不是所有）问题都是因为忽视了常见分布式系统中早已存在的命名教训。

6.3.1 分布式系统的命名观点

在20世纪的最后25年中，分布式系统的研究领域产生了许多命名问题。用来把名字和地址捆绑在一起的基本算法是集合点（rendezvous）算法：用户输出一个名字并四处通告它，而那些寻求引入这个名字的用户将用它搜索。一个明显的例子包括电话簿和文件系统的目录。

但是，分布式系统使用者很快意识到命名将变得十分复杂，这方面值得学习的内容可以参考Needham写的一篇经典的文章[587]。我总结一下其中的观点，并且看看哪一条可以应用到安全系统。

1）*名字的功能是使共享便利。*下面将进行论证：我的银行账号存在的目的是提供共享信息的便利方式，例如我上星期在银行存了款，这星期想通过出纳员将这笔钱提出来。一般来说，当共享数据被交换时名字是需要的。如果我想提出的款项恰好是存款款项，一个款项所有人的存款证明是很有帮助的。相反，在不需要如此使用数据的地方名字不需要被共享或者链接；如果不是通过账户付电话账单那么将我的银行账号和我的电话号码链接起来是没有必要的。

2）*命名信息不可能全在一个地方，所以要解决名字带给分布式系统的所有问题。*这个观点有一个悖论。银行账户和电话号码之间的链接是假定二者都是稳定的。当其中一个依赖于另外一个时，攻击其中一个将会同时影响两个系统。今天的电子银行是通过拨号上网而不是基于网络，银行通过在线呼叫ID识别客户对于攻击来说是脆弱的，它危及了电话交换系统的安全（比如在一个公寓楼内窃听电话分布网，冲击电话公司的计算机，或者贿赂电话公司的职员）。

3）*只需要这么多名字的假设是错误的。*IP地址的缺点，推动了IP版本6（Ipv6）的发展，这已经被详细讨论过。不太清楚为什么要进行昂贵的升级，信用卡行业曾经不得不做的不是Y2K的修补，而是从13位信用卡号码升到16位；发行者原来以为13位就够用；但是系统终端有成千上万的银行（很多银行有很多分支机构），因此一个6位的银行识别号码（BIN号码）是需要的。有些卡发行者有上百万的用户，因此一个9位的账户号码是可行的。而且还要有一个校验数字（其他数字的线性组合，用来检测错误）。

4）*可用的全局名字比你想像的要少。*比如，IPv6的128位地址规划可以使整个宇宙的物体都有一个惟一的名字。但是，对于我们处理实际业务，在终端一个本地的名字必须解析成一个惟一的名字并返回终端的本地名字。在处理的中间过程调用一个惟一的名字可能不会给我们带来任何好处；惟一命名服务将耗时、费钱、甚至可能失败（好像是一定的）。事实上，命名服务通常是一个分布式系统，同我们要保护的系统规模（和安全等级）相同。从这一刻起我们别指望能获得钱财回报。其中一个原因是银行业缺乏建立公钥基础设施的主动性，这个基础设施能给每一位公民提供一个平等的电子ID卡，但银行对于它们的客户早已有了惟一的名字（账户号码）。对银行来说，它们认为增加一个额外的号码没有一点好处，却有增加花费和发生故障的可能。

5）名字意味着承担义务，方案要有足够的灵活性来适应组织的变化。这在 Cloud Cover 设计英国政府的安全邮件密钥管理系统［50］时是一个被忽视的问题。在那里，用户的私钥是根据部门的主钥对他们的名字进行加密生成的。因此，重新组织意味着安全基础设施需要重建。

6）名字将像访问票据和权能一样翻倍。在有关协议和口令的章节中可以看到很多相关的例子。通常，那种认为今天的名字不是明天的口令或权能的想法并不是好主意——记住在 2.4 节中讨论过的乌得勒支诈骗（这是将所有名字转换成公钥的一个争议——Carl Ellison 的言论："密钥在电脑中说话"——但是我们也注意到将密钥和名字链接的困难）。

我已经对当名字被用作口令时会出现什么样的错误给出了很多例子。但是有时名字和口令的角色是模糊的。为了进入大学的停车场，需要在停车场麦克风附近说出我的姓和停车证号码。如果我说"Anderson，123"（或者其他的），哪一个是口令（事实上是"Anderson"，因为每个人都可以走过停车场并注意到场地允许的显示在车上的有效号码）。在这里，应当提一下生物测量学，这将放在第 13 章讨论。

7）如果一个不正确的名字是明显的会使事情简单。在标准的分布式系统中，这使我们对于缓存更加宽容。在付款系统中，当终端掉线时信用卡号码可能被接受，只要卡号有效（比如，最新的数字是一个对卡号前 15 位的正确的校验数字）而且不在热卡黑名单（hot-card list）中。证书在同样的基本概念中提供了高质量的实现。

在哪里检验名字是重要的。信用卡校验数字算法在出售点部署，因此不可避免地要被公开。一个进一步的检验——磁条上的卡号确认值（card verification value，CVV）——通过密钥被确定，但是可以在开户银行、买入银行或者网络交换机（如果人们相信这种第三方的密钥）被检测。这将是昂贵的，而且对于网络中断仍然显得很脆弱。

8）一致性是困难的，并且容易伪造。如果目录被复制，将会发现自己不能读取或者写入，这取决于可用的目录是否太多或者太少。命名一致性在电子商务中以不同方式产生很多问题，最严重的可能是 barcode 系统。虽然在理论上这是很简单的——给每个产品一个惟一的数字码——实际上，那将是可怕的。因为不同的制造者、发行者和零售者在他们的数据库中对于 barcode 会有不同的描述。这样，针对"Kellogg's"搜索产品将依赖于是否插入一个省略符号而产生不同结果，这将在供应链中引发很大的混乱。处理这个问题的建议是非常复杂的［387］。

上面讨论的也有一致性的问题；数据通过一个系统时可能不一致，即使在理论上也是如此。还存在时间性的问题，比如公钥证书的恢复等。

9）别太聪明了。电话号码比计算机地址安全很多。但愿如此，但是它经常被安全系统设计者忽略。银行账户号码比服务于像 SET 这样的协议的 X.509 证书更容易处理，SET 被认为是网上信用卡付账的新标准，但是因为其协议的复杂和花费巨大而失败了。在第二部分中将讨论 X.509 和 SET。

10）一些名字被较早地绑定，其余的则不；如果可以避免，那么绑定过早将是件坏事情。一个谨慎的程序员会避免编制绝对的地址和文件名，因为那将使升级和替换一台机器很困难。可以把这个任务留给一个配置文件或者一个外置设备像 DNS（这是不把地址放入名字中的另一个原因）。这里，有一些出于保护目的的轻微紧张：安全系统常常需要稳定和负责任的名字，因为任何最后一分钟决定采用的第三方服务都可能是一个攻击点。要预先很好地

了解它们允许对交易预授权和进行某些优化处理。

因此，Needham对于分布式命名的10条原则，有9条直接应用于分布式安全系统。（部分）例外是名字是否应过早或过晚绑定。

6.3.2 哪里出了问题

Needham的原则，虽然有用，但是并不充分。它们是在一个对于系统拥有者可以便捷设计和施加命名系统的世界中设计的。当我们把分布式系统抽象到以现代因特网为基础的服务业（或其他的相关行业）时，还有很多东西要讨论。

6.3.2.1 命名和一致性

最明显的不同就是安全协议中的当事人可能通过很多不同的名字形式被知晓——一个银行账户、一个公司注册号码、一个人名加上生日和邮政地址、一个电话号码、一个护照号码、一个健康服务病历号码或者计算机系统中的一个用户ID。

在前面的介绍性定义中提到，一个通常的错误是混淆命名和身份。*身份*是在两个不同的名字（或者是相同名字的实例）与同一个当事人联系起来（在分布式系统中称作*间接名字*或者*符号链接*）时使用的。经典的例子是注册财产的所有权。卖掉房子时使用一个不同于购买时的名字是很正常的：他们或者结婚或者犯罪判刑后改了名字。名字用法的改变也是正常的。比如，Bell-LaPadula系统（下一章将讨论）的DE Bell在1973年的论文中写下他的名字“D. Elliot Bell”；但他经常被认为是David，就是他现在写的名字。土地注册系统也会遇到很多这样的一致性的问题。

一个更加典型的关于身份的说法是“拥有一个银行账户12345678的Jim Smith就是护照号码为98765432并且生日是3/4/56的Robert James Smith”。它可以被看作两个分离系统的符号链接——银行和护照部门。注意到这个身份例子的后一部分包含了更进一步的身份内容，“美国护照部门的文件号码98765432对应于生日注册为3/4/56的Robert James Smith。”一般来说，名字包括很多步的递归循环。

6.3.2.2 文化背景假设

假设每个国家的命名基准经常改变。在说英语的国家中，人们想用多少名字就用多少；一个名字只是为了让别人了解你。但是有些国家限制使用别名，而且另外的国家要求对此进行注册。这会引发很多有趣的故事。至少在这样一种情况中，一个英国公民会通过改变名字来躲避外国税务机关的追查。还存在一个毫不夸张的情形，想从事科研生涯的妇女，因为结婚改变了名字，但是她希望为了职业的需要而保留原来的名字，这意味着在她们的论文上出现的名字将不同于出现在薪水册上的名字。这在我的大学里造成了很多问题，提出的一个解决方法是为薪水册上的名字提供惟一的ID卡，不支持别名。

一般来说，当试图统一两个产生于相互矛盾的假设的局部命名系统时，许多难处理的问题就产生了。随着日常生活日益电子化，系统变得相互连通起来，冲突能够被传播而且有不可预见的影响。比如，一个对我们的大学卡提出质疑的女教授同时也是英国图书馆的理事，她就可以基于母校图书馆的卡名字发给自己许可票据。

人类的命名约定也不统一。俄国人通过名、源于父亲的姓和本姓而被认识。冰岛人没有姓而是通过一个名，如果是男的则跟着一个源于父亲的名字，如果是女的则跟着一个源于母亲的名字。当他们旅游的时候会产生问题。如果一个美国的移民的名字是Maria Trosttadóttir，

并知道 Trosttadóttir 不是姓也不是一个源于父亲的姓。通常比较实际的做法是强迫该移民接受那是一个姓或者源于父亲的姓（比如，如果她的父亲被叫做 Carl，则可以取 Carlsson）。但这会产生不必要的冒犯。

最大的文化差异被认为发生在说英语的国家之间，统一的卡在各自的国土上不被接受（除非是驾照或者健康服务证），被拿破仑（或者是苏联）征服过的国家则可以统一（使用一种卡）。其他的例子更加微妙。德国人不相信一个没有正确的人口注册和 ID 卡的系统的国家会运行正常，但他们很少被问及 ID 号码（比如，开一个银行账户或者是结婚）。他们的卡号不能被用作名字，因为那是一个文件号码，每次新卡发行时它就变了。一个瑞士店主更喜欢用 ID 卡而不是信用卡来注册一个德国客人。但在德国客人结账后，他发现房间东西有些损坏，那他是不走运的，因为他无法容易地找到这个客人。英国护照办公室经常在同一时候发给公民不同的护照，如果他说去古巴或者美国做生意，那么就会有不同类型的护照；所以，如果瑞士店主发现一个英国客人没有付钱就走了，不要指望护照号码可以将他阻止在机场。

在政府和人们名字之间的关系上还有许多其他的假设，它们会随国家不同而有所区别，这种方式会引发微妙的安全失误。

6.3.2.3 名字的语义内容

名字的另一个危险起因于在没有足够的背景研究下从一种类型的名字变成另一种名字。一个银行因为从通过账号保存客户数据转变到用地址和名字保存客户数据而被起诉。银行是想更加准确地将业务瞄准邮寄信件，所以它写了一个程序来连接每个客户的所有账目操作。这对于一个客户的影响是银行对于客户女主人的账目声明发到了他的前任妻子那里。

有时命名很简单，但是有时仅仅是看起来简单。比如，当我得到一个当地游泳池的月票时，出纳员仅仅是拿出一大堆卡里最上面的卡，滑过读卡机系统显示卡还有效，然后交给我。我已经被分配了一个随机的名字——卡上的序列号。许多美国公路收税系统都是这样的。有时一个随机的匿名能够增加商业价值。在香港，对于 Aberdeen 隧道的收税凭证可以通过现金购买或者以重新充值卡的形式打折购买。在提高从英国到北京的汇款效率的过程中，很多人宁愿对较少追踪的汇款付额外的钱，因为他们担心新设立的警察机关的监督。

名字的语义可以变化。我曾经有一个通过随机账户（没有信用检查）使用诚实卡（loyalty card）的硬件商店。在这个商店被一个超市接管后，我有机会将卡改为银行卡，这样超市就成了一个银行（这似乎忽略了贪污钱的规则，因为所有的新银行客户必须被识别并有足够的参考资料认证）。

给客户分配银行账号看起来没有什么问题——但是像上面的例子所示，系统可能会在误导的和危险的名字之间建立联系。

6.3.2.4 名字的惟一性

人类的名字从我们生活在一个小社会时就开始进化。它们不是为了因特网设计的。我们需要面对因特网上许许多多的人（和系统）。正如在本章第一节提到的，我曾经是我所知道的惟一的 Ross Anderson，但是由于有了因特网搜索引擎，我知道了成百上千个同名者。一些人工作在我从事的领域，比如软件工程、电力分配；事实上我拥有 www.ross-anderson.com 和 ross.anderson@iee.org 是幸运的，因为我先得到了它们（比如，rjanderson@iee.org 已经是别人的信箱地址了）。所以即使是将一个非常罕见的名字和一个非常专业的职业合并也可能会造成混淆。

6.3.2.5 名字和地址的稳定

许多名字包含各种类型的地址，然而地址却可能会发生改变。每年大约有四分之一的剑桥电话本地址要变；通过电子信件，这个变化会更大。有一个项目是开发使用加密邮件的人员目录，连同他们的密钥，项目中发现目录条目的主要修改是针对于电子邮件地址的修改[42]（一些人认为那将发生密钥的丢失和被盗；这使得该项目的贡献几乎是零）。

一个潜在的严重问题起源于IPv6。安全团体认为v6 IP地址是稳定的，所以可以建立公钥基础设施来联系不同类型的用户。各种实现机制都希望永久地把真实的名字、地址、甚至文件内容反映到这个128位的字符串上（可参考文[365]的例子）。另一方面，数据通信研究者认为IPv6地址将会有规律地改变。地址的高有效位应该提供变化从而提供更高效率的路由算法，而低有效位用于管理局域网。但这些假设不能同时成立。

分布式系统创建者认为把地址放到名字里是件坏事[565]。但是一般来说，有很多层次的抽象，每一层的地址信息组成上一层名字的一部分。而且，一个名字空间是否有效取决于应用。通常人们在部门和组织的层次上以不同的名字结尾（比如对于我自己，有rja14@cam.ac.uk和ross.anderson@cl.cam.ac.uk。）。所以在名字和地址之间进行清楚的划分常常是不可能的。

授权可以有很多（但不是全部）地址的属性。公钥基础设施的设计者开始认识到如果一个公钥证书含有一列它可能用到的信息，该列信息越多，证书的使用期越短。一个类似的问题困扰使用合成名字的系统。比如，一些网上业务通过电子邮件地址和信用卡号的组合来确认我。这明显是一种不好的实践。与我有几个电子邮件地址完全不同的是，我有几个信用卡。我用哪一个信用卡取决于现在哪一个能够提供最好的现金回馈和可以在最远的航程内使用（因此如果政府通过一个法律使得在网络上使用假名违法，是否意味着我必须坚持把一个ISP和一个信用卡联系起来）？

6.3.2.6 名字使用的限制

这将带给我们进一步的问题。一些名字也许只能在受限的情况下使用。这可以通过法律来规定，比如美国社会安全号码（SSN）和它在欧洲国家的等价号码。有时这也是一个市场问题。当我在网上买东西时我不想暴露我的住址（或者我的电话号码），并且会避开那些需要它们的业务。

值得纪念的假名可能是需要的。在大学里，一个人有时不得不变换电子邮件地址。比如，当一个学生是密码被窃的受害者时。另一个例子是一个名人想有一个私人信箱和一个很“显然的”发往她秘书的信箱。

有时问题会更严重。假名经常用在隐私增强技术上。它们可以用在想不到的场合中。比如，医院用一个病人的号码作为医治数据库的索引，这将允许研究者用假名记录来进行一些限制的研究目的而不需要办理进一步的手续。当健康保障组织合并或者国有健康服务发布一个策略性指示，强迫医院使用统一的名字时会产生很多问题。病人的保密性被完全削弱了（在第20章将会对匿名进一步讨论，在第8章中有医疗数据库的特定应用）。

最后，回到法律和策略上来，一个名字的定义会变得意想不到的棘手。规则要求允许警察收集通信数据——就是说，一个关于谁在什么时间给谁打过电话的记录——这通常比管理电话窃听的规则要松懈得多；在许多国家，警察可以通过询问电话公司获得这些数据。而在美国造成公众大声疾呼的事件是这是否使警察可以收集人们用来存取网页的URL。URL经常

嵌入在作为参数传给搜索引擎的数据中。明显地，警察可以得到某人点击 URL 的列表，比如 http：//www.google.com/search? q = cannabis + cultivation + UK；很多人认为这么大规模的拉网式调查将是一个不可接受的对于隐私的侵犯。另一方面，如果警察被限制监视 IP 地址，他们将很难追踪到使用免费 ISP 服务提供的暂时 IP 地址的罪犯。

6.3.3 名字的类型

名字的复杂包括了组织和技术上的复杂，也包括政治方面。在前面的介绍中提到名字可以指向人和机器，还可以指向组织、角色（“值班军官”）、组和建筑群：当事人角色——Alice 是经理；代表——Alice 代表 Bob；关联——Alice 和 Bob。关联经常表示不准确的访问规则：“Alice 是一个部门经理，Bob 是部门会计组的一员。”

这仅仅是个开始。名字也用在服务（如 NFS 或者公钥基础设施 PKI）和通道（代表线路、端口或者加密密钥）中。同一个名字可能指向不同的角色：“Alice 是一个计算机游戏者”应该有少于“Alice 是系统管理员”的特权。通常在安全中的抽象概念是把他们作为不同的用户对待。这意味着在名字和当事人之间不是简单的映射关系。

最后，名字还有来自底层业务过程而不是系统设计的功能性的延伸。业务主要是想得到付款，而政府想惟一地区分人们。实际上，商业需要一个信用卡而政府需要一个护照号码。建立二者兼顾的系统——有些政府试图鼓励这样做——是一个陷阱。不同的名字有很多语义上的不同。你可以把你的护照给一百万人看，如果你愿意，但是你最好不要这样对待一张信用卡。银行希望给每个存入钱的人开账户；政府希望银行仔细认真地鉴别人们的身份来防止洗钱。要探讨的列表内容很长。

6.4 小结

许多安全分布式系统导致巨额的花费和带来严重的脆弱性，因为它们的设计者忽略了怎样建立（或者怎样不建立）分布式系统的基本教训。这些教训中的大多数现在还是正确的，并且有更多的教训加入。

很多安全漏洞会导致一种或者多种的故障同时发生；系统使用旧的数据、不一致的升级或者错误的顺序或者认为数据一致而实际上却不是。知道故障发生的确切时间比看起来要困难得多。

容错和故障恢复是重要的。提供从故障和随机物理灾难中恢复的能力，对于很多组织来说是保护预算的主要目的。在更加技术性的层次上，保护和恢复机制有着更重要的联系。拜占庭式故障——有缺陷的进程一起发作，而不是随机的故障——是一个需要面对的问题，还涉及加密工具的选择。有不同种类的设备冗余方法，必须采用最佳组合。需要保护的不仅是防止故障和有恶意企图的操作，而且还要阻止故意的拒绝服务攻击企图，这经常是大规模攻击的一部分。

许多问题起因于试图用一个名字做太多的事情，或者假设它位于一个特殊的系统、一种文化或者权限之外。比如，应该通过取消用户的名字来撤销一个用户访问系统的权限而不至于使因其他功能被撤销而被起诉成为可能。最简单的做法是给每一个用户分配一个惟一的标识符但不能用作其他目的，比如银行账户和一个系统登录名字。但是当合并两个因为某种原因使用了不一致名字规划的系统时很多问题产生了。有时这种合并会偶然地发生——比如当

两个系统使用共同的名字组合如“名字加生日”来追踪个人。

研究问题

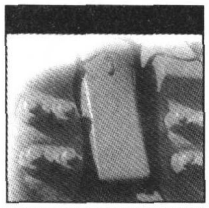
在当前研究中，安全分布式系统趋向于被专家作为一个通信协议和操作系统的边缘问题加以讨论，而不是将它作为独立的学科。因此它是一个相对开放的研究领域，而且在将来的5～10年中会很有希望。

本章涉及了很多技术问题，比如怎么设计安全时间协议和命名的复杂性。但是最重要的问题可能是如何设计系统使其面临恶意攻击更具恢复能力，恰当地进行性能降级，当攻击过后可以很快地恢复安全。这意味着重新采用收敛性应用。在什么情况下可以从崩溃的安全状态完整恢复？需要重新运行恢复（从备份盘重新建立数据库）吗？在恢复机制和特殊保护技术之间有什么关系？保护机制与恢复机制在什么方面可以分开和在什么方面它们应该分开？发生这样的变故后什么东西被丢掉了？

参考资料

有很多关于分布式系统的书。Sape Mullender 的 asthology [565] 很有帮助而且对于研究生很有启发性，Jean Bacon [64] 的推荐给本科生的课本也值得一读。Geraint Price 有一篇调查文献是关于容错和安全的关系 [623]。关于并行的研究文献，比如 SIGMOD 会议，有一些精品文章。但是对于安全工程师最重要的实践主题也许是应急计划。有很多关于这个主题的书；我的书架上的那本书是 Jon Toigo 写的 [749]。

第二部分



在本书的第二部分，介绍了许多安全系统的应用，其中引入了一些独特的保护概念和技术。

这部分包含四个连续的主题。第7章至第9章针对一般的计算机安全问题，讨论应尝试着去做什么事情以及如何在不同的环境——军队、银行和保健所——中完成这些事情。介绍了安全策略模型，它们提出了实际系统尝试实现的保护概念。还将在这几章中首次介绍一些具体的样例分析。其中的一例便是全球范围的自动提款机网络，该例讲解了使用加密技术将熟悉的保护属性从一个银行分行传送到整个分布式系统中遇到的问题。

第10章到第15章将讨论信息安全的硬件工程方面的内容。这包括生物测量学、各种令牌（比如智能卡）的设计、抗干扰和干扰探测、发射安全，以及印章。演示各种技术的新型应用将在这里描述，范围从电子对抗与核武器控制到出租车计价器、卡车限速器和预付费加油计价器等。

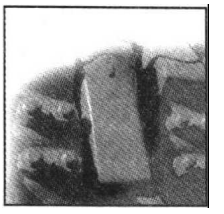
第三个主题针对网络攻击。将从第16章开始这个话题，包含了电子和信息对抗，因为这些活动提供了一些更为极端的例子，并展示了在严重的操作压力下否认、欺骗和利用等技术能够被一个机敏的敌人提高到怎样的一个水平。这章还给出了从警方和情报机构角度出发的监视和入侵的观点，并介绍了一些新概念，比如匿名和通信流量分析。之后我们在第17章通过检查电话系统以及依靠它们的应用技术中的欺诈行为来学习历史上发生的教训。这为第18章有关计算机网络上的攻击以及防火墙和入侵检测等防御技术的讨论做好了铺垫。

第四个主题是电子商务，我将它放在第19章和第20章来处理。安全应用的最高配置是保护网上信用卡交易的方案，比如SSL/TLS；它们也被用来进行诸如医学图像分发等应用中。由此引入了关于公钥基础设施的讨论。另外，我还考虑到关于版权保护的一些技巧，特别是收费电视、DVD(数字化视频光盘)、版权水印。

这样排序的原因之一就是给各章一个符合逻辑的进展。比如说，我先讨论了针对银行磁条卡的欺骗行为，之后继续讲述可以取而代之的智能卡和当今使用智能卡的收费电视系统。说是这样说，可有时候一个纯粹线性的排序是不可能实现的，因为一个特定的技术会经历面向多个应用的多次反复发展。正是由于这样的情况，我试着按照典型例证发生的顺序来讲述。

最后需要说明的是，为了使本书能用作参考书而不是教材，我已经把更多的技术资料放在各个章节的末尾。这样，如果读者因为当前内容延缓了阅读的进度，可以直接跳到下一节继续阅读。

第7章 多级安全



有时，就国家安全而言，保密已经损害了所谓的安全。

——Daniel Patrick Moynihan

我发布命令；
你走漏消息；
他（她）泄露了机密的信息导致了犯罪的发生。
——英国谚语

7.1 引言

在前面军用数据库系统的介绍中提及过，军用数据库系统可以按照不同的分类级别（秘密，机密，绝密……）保存信息，它必须确保一条信息只能被比至少同其分类级别一样高的人读到。这个系统的重要性在于：

- 在美国军方有关计算机学科的巨大投资下，大量的研究工作已经使得军方的防御模型比任何时候都工作得更详细，同时向我们提供了许多二阶甚至三阶的严格执行安全策略的例子。
- 一些为了支持军用多级安全而开发的产品找到了一条新的发展途径，例如作为防火墙和网络服务器的平台。它们可以保证即使防火墙或服务器的软件被入侵，底层的操作系统也不会被破坏。
- 虽然多级的概念最初是为支持军用系统安全而发展起来的，但现在许多商用系统也有了完整的多级策略。例如，电话公司希望它们的收费系统可以反映出调度系统的情况，并且不对调度系统产生影响。
- 由于已存在的利益和因素，多级安全的概念在应用环境中经常会使效率降低，甚至是有害的。

Isaiah Berlin 曾经有一个著名的评论：一只狐狸懂得许多小事，而一只刺猬只懂得一件大事。多级安全的思想就像一只刺猬，它注重的是安全工程。

7.2 什么是安全策略模型

需要采用安全工程的地方，都可以表达为威胁模型—安全策略—安全机制模式。其中关键性的、经常被忽视的那个环节就是安全策略。

安全策略，指的是一份可以简单明确地表达保护机制要达到的目标的文件。它决定于我们对安全威胁的理解，然后再来驱动我们的系统设计。它常见的形式是一份陈述，指明了什么样的用户可以访问什么样的数据。系统的详细说明覆盖了系统的全部功能，而安全策略在

详细说明系统对于保护的要求并且评价这些要求是否被满足方面扮演着相同的角色。的确，一份安全策略可能是系统的详细说明中的一部分，和系统的详细说明一样，它的主要功能也是沟通。

很多组织采用“安全策略”这个词来表示一系列索然乏味的声明。图 7.1 给出了一个简单的例子。这种措辞非常普遍，但对于安全工程师毫无意义。

它的第一个缺陷是它避开了最主要的问题，也就是“谁决定了以及如何决定‘需要了解的’内容?”第二，它混淆了一系列不同级别的声明（组织对这个策略的承认在逻辑上不能是这个策略本身的一部分）。第三，这是一个实施机制，但它更多是含蓄的而不是直率的：“全体职员必须遵守”——但是事实上到底他们要做什么？这种遵守是被系统强迫执行的，还是用户们凭自觉遵守？第四，破坏行为如何被发现以及谁有明确的责任来负责汇报？

Megacorp 公司安全策略
1. 此方针经管理部门批准。
2. 所有职员必须遵守该安全策略。
3. 信息只提供给需要知道的用户。
4. 一切违反策略的行为将被立即报告给安全部门。

图 7-1 一份典型的公司信息安全策略

我们必须对此做些改进。实际上，因为“安全策略”这个术语已经被广泛地用于表示一些经理主义拥护者的陈词滥调，现在可以用三个更精确的术语表述防御要求的规格：

安全策略模型是一个系统或一个通用系统类型所必需的防御属性的简单陈述。它的要点可以在一张纸或更少的空间内写出来。在这个文件中，一个系统的防御目标得到了全体或最高层管理者的批准。这个文件也是将来进行数学分析的基础。

安全目标更加详细地描述了一个特定的实施过程给出的所有防御机制，以及它们如何与控制对象（其中一些典型来自于安全策略模型）联系起来。安全目标是测试和评价一个产品的基础。

保护框架与安全目标类似，但采用了与实现无关的方式表达，以使关于产品及其改进版本的可比较评估有效。它使用了半形式化语言，至少是适当的安全性术语。一个将要按通用准则 [574] 评估的产品必须具有保护框架（通用准则将在第三部分中讨论；它是一个被许多政府采用并相互认可的国防信息系统安全性评估方案）。

在不需要太精确的场合，本书一般用安全策略代替以上三个术语中的任何一个或所有。我决不会用这个术语来表示那种陈词滥调。

有时，我们面临一个全新的任务，必须从零开始设计一个安全策略模型。更普遍的情况是，已经有模型存在；我们只需要选择一个正确的模型，并把它变成一个安全目标。这些步骤的实现都不容易。本节的目标之一就是提供一些安全策略模型，在实际系统下描述它们并且检验一个安全目标能否满足安全策略模型的工程机制（和相关的约束限制）。

最后，“安全策略”的第三种用法是表示一系列为安全防御产品设定的特殊配置，我将在后面的配置管理或在可信赖配置管理中偶然提到它。

7.3 Bell-LaPadula 安全策略模型

一个最著名的安全策略模型是 David Bell 和 Len LaPadula 在 1973 年提出的，以响应美国

空军对分时共享大型机系统安全性的关注。在20世纪70年代初，人们已经意识到很多商业操作系统提供的保护非常脆弱，并且没有改善的趋势。一旦操作系统的一个漏洞被察觉，其他的一些漏洞也会立即被发现（现代的可靠性增长模型可以量化并证明这种悲观的看法是正确的，我将在后面23.2.4节中讨论这些）。人们一直都在担心即使不熟练的用户也能发现系统漏洞，并趁机利用这些漏洞；同时，人们对病毒（恶意代码）威胁的注意与日俱增。五角大楼的全球军事指挥控制系统受到了特洛伊木马的攻击，这件事引起了一阵极大恐慌；它已经攻击到人们心目中的“绝密”系统，攻击成功是非常不容易的。最后，学术界和工业界的研究人员共同提出了一些引人注意的安全防御新概念，这就是我们将在下文讨论的内容。

James Anderson的一项研究使美国政府得出了这样的结论：一个安全的系统应该很好地完成一个或多个方面的任务；执行防御功能的机制必须简单，便于验证并且很少发生变化。这引出了一个引用监控器的概念，它是操作系统中的一个组件，是访问控制决策的仲裁，很小以便分析和测试，但必须保证它的完整。用现在的说法，这样的组件和它们相关的操作程序共同成为可信计算库（TCB）。更加正式的说法是，TCB被定义为组件的集合（包括硬件、软件、人及其他），它的正确运行确保了安全策略的执行，而它的失灵将导致安全策略被破坏。Anderson的报告的目的是使安全策略足够简化，以使TCB能经受仔细的核查。

但什么是最核心的应被优先执行的安全属性呢？

7.3.1　密级和许可

第二次世界大战和随后的冷战，使北大西洋公约组织（NATO）采用了一种通用的安全评价方案，以评价文件的灵敏度。密级是一种分类标记，从不保密到秘密、机密和绝密。分类细节有时候会变化。最初的提议是：如果此信息的损害危及生命，则被分类为“机密”，而如果此信息的损害危及到许多人的生命则被分类为“绝密”。政府雇员经谨慎审核后获得一定的许可（clearance）；例如在美国，“机密”级的用户可以查阅FBI的指纹档案，而拥有“绝密”级许可则可以查阅某人5～15年前的工作背景[244]。

这种访问控制策略是很简单的：如果一个用户获得的许可级别高于一份文件的分类级别时，他就可以阅读这个文件。因此一个拥有“绝密”许可的官员可以阅读一份“机密”级别的文件，而反之则不行。即信息只能向高级别流动，从秘密级别到机密级别再到绝密级别（见图7-2）。信息永远不能向下流动，除非授权人员决定降低文件的保密等级。

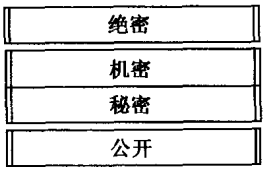


图7-2　多级安全

还有文件（自动）处理准则：“秘密”文件被保存在普通政府办公室的上锁的文件柜中，而更高保密级的文件需要被保存在经过安全认可的、有警卫的并通过影印机进行控制的房屋中（美国国家安全局安全手册[582]给出了“绝密”情报资料的使用程序说明）。

系统迅速变得更加复杂。用于给文件定义密级的损害标准已经从可能发生的军方事件扩展到经济危机甚至扩展到政治困境。英国在“不保密”和“秘密”之间有一个额外的密级——“受限”；美国过去也有这个密级，但在信息自由法案（Freedom of Information Act，FOIA）通过后就被废除了。现在美国有两个特殊密级：“仅供官方使用”（For Official Use only，FOUO）指虽然无保密性密级，但是不能按照“信息自由法案”（FOIA）公开发布的信息；“不保密而敏感”包括FOUO和应FOIA要求才会公开的材料。在英国，“受限”的信息

其实可以免费共享，但是如果记者或其他相关人员泄露了这些信息，可以按照国家安全保密法对他们提出起诉（它的另外一个实际的影响是，一份美国的不保密文件被送到英国自动成为了“受限”文件，然后回到美国时就成为了“秘密”文件。美国军方系统的制造者埋怨英国的策略打乱了美国的密级方案）！

信息，尤其是“机密”以上密级的信息，可以借码字系统受到进一步的保护。例如，包含情报来源和取得方法的信息——如间谍的身份和从外国政府的通信中破译出的电文——通常被定义为“绝密特殊分割情报”密级或者 TS/SCI，这意味着即使对需要知道的人也要在文件中加码字以对其约束。一些码字与特殊的军事行动或情报来源有关，只能被一些指定的用户使用。一个用户必须拥有这个文件所有的码字才可以阅读它。密级标记再加上一组码字，构成了安全等级或（假设这里有至少一个码字）安全分割，这是一系列采用相同的访问控制策略的记录集合。这种划分将在第 8 章更详细地讨论。

还有描述符、警告（caveat）和 IDO 标记。描述符就是像“管理”、“预算”、“指定”之类的词语：它们不调用任何特殊的操作要求，我们可以把一份文件标记为“秘密—管理”而不是简单地标记为“秘密”。警告语比如：“仅限英国人参阅”，或者美国的“不可向国外发表”。还有国际防御组织（IDO）标记，例如 NATO。而由于码字、描述信息、警告和 IDO 的标记之间一般没有明显区别，它们是使系统变得混乱的一个因素（更详细的解释请参见[630]）。

关于访问控制研究的最终一致意见是：只允许信息向上单向传输，这也就是窃听器模型。过去，如果要窃听某人的电话，要在交换机上接一根实体的线，而现在可以由电话交换机上的软件来做到这一切。这就类似于一个额外的参与者把对一个目标的通话变成有三人参加的电话会议。通常的安全要求是被调查的目标不能发觉他自己正在被窃听，因此第三方应该保持沉默——他的存在必须不被调查目标发现。举个例子，现在窃听的过程像一个安静的电话会议，必须小心地保证这个电话会议有益于窃听者，而不是被窃听者。窃听的过程所必须遵循的信息流动策略是：“高级”用户可以看到“低级”数据，而“低级”用户不知道“高级”用户是否得到了任何数据。

7.3.2 信息流控制

上文中提到的用于军事和情报数据密级划分的 Bell-LaPadula（BLP）计算机安全模型是 David Bell 和 Len LaPadula 在 1973 年建立的 [86]。它被认为是多级安全的，采用它的系统被称为多级安全（MLS）系统。它们的基本属性是信息不能向下流动。

严格地说，Bell-LaPadula 模型有两条基本属性：

- 简单安全性属性：用户无法读到安全级别更高的信息。这也被称为不能向上读（no read up，NRU）。
- * -property：用户无法改写安全级别更低的信息。这也被称为不能向下写（no write down，NWD）。

* -property 是 Bell 和 LaPadula 关键的创新之处。它的产生来源于对使用恶意代码攻击系统的顾虑。一个未被授权的用户可能已在系统某个地方留下一个特洛伊木马程序，而一旦“机密”等级的系统管理员执行了它，它就可以将自身复制到系统的“机密”部分，阅读机密信息并试图使该信息无缘无故地无法打开。有一种很可能发生的情况是，敌人的间谍在一

个软件开发实验室工作，他在开发的产品里嵌入了一些代码，这些代码可以寻找机密文件并复制它们。如果这个产品将这些机密文件复制到了它的设计者能看到的地方，则安全策略就被破坏了。如果应用程序可以向下写，那么作为一个 bug，高安全级别的信息就可能被泄露出去。

恶意和错误代码这类攻击来源于外界，因此必须使安全策略的执行独立于用户的行为（以及用户所运行的程序的行为）。必须防止运行“机密”文件的程序输出到“不保密的”文件。更一般地讲，必须防止任何在高端运行的程序给任何低端目标发出信号。这种可以独立于用户行为来执行安全策略的系统通常被描述为具有强制访问控制，它与自主访问控制相反，在采用自主访问控制的系统，例如 Unix 中，用户可以对他们的文件做出自己的访问决策（我将不会太多地采用这个短语，因为它们一般只涉及 BLP 类型的策略而没有包括其他也采用强制性原则的策略）。

Bell-LaPadula 模型直接要求验证系统的设计中提供的各种保护。将简单安全性属性（不能向上读）和 * -property（不能向下写）都给出后，从给定起始状态得出的有关机器状态的各种结果都可以验证，而且简化了形式化分析。

这里还有一些详细的规则，如可信主体，它是指有无密级文件访问权限的当事人。为了简明起见，我们暂时忽略掉可信主体和相互矛盾的安全级出现的可能性，把它们留到下一章讨论。为了进一步使情况简化，从现在起假设系统只有两个安全等级：高级和低级（除非有特殊原因要引入单独的等级）。

多级安全可以通过几种方式来执行。课本机制是通过加强操作系统中某部分来实现一个引用监控器，该部分的职能是监控整个操作系统收到的所有访问请求，检验它们的访问权限并决定是否为这个请求提供服务。实际情况要复杂得多，因此如果要求系统的可信计算库比整个操作系统的内核（加上大量的实用工具）小很多是非常困难的。

另一种随着硬件成本的降低而发展起来的安全途径是备份系统。例如，一个系统可以有一个高安全级的数据库和一个低安全级的数据库，一个泵不断地从低级数据库复制文件到高级数据库。我们将在后面详细讨论这个泵。

7.3.3　Bell-LaPadula 模型的标准批判

对 BLP 模型的介绍引起了一阵兴奋：这是一个很直观易懂的安全策略，并允许人们从理论上来证明。但 John McLean 指出了 BLP 规则并不充分。他引入了 Z 系统，并将其定义为一个 BLP 系统加上一个额外的特性，即一个用户可以向系统管理员提出申请，临时将一个文件由高安全级解密到低安全级。这样，一个低级用户不用破坏 BLP 系统就可以接触到高级文件。

Bell 认为 Z 系统做出了模型不能允许的假设（改变密级标记在此状态下不是正当操作），而 McLean 认为模型没有明确地禁止这一点。最后通过引入宁静属性解决了这个问题。强宁静属性规定了在系统运行当中安全标记不可改变，而弱宁静属性规定安全标记在不违反已定义的安全策略时可以改变。

引入弱属性的动机是在现实的系统中，我们通常希望只遵守最宽松的原则，而且希望能在不保密的安全级别上运行某个程序，即使这个程序的拥有者必须是“绝密”级别的。如果她访问一封“秘密”的电子邮件，则她的级别自动更新为“秘密”；就是说，每当她访问一

个更高级别的数据时，她使用的程序也会升级（这被称为高水标原则）。因为一个安全主体通常是内存管理子系统和文件句柄的抽象而不是过程，这意味着状态随着访问权限的改变而改变，而不是随着数据的真实移动而改变。

这种做法的实际含义是，一个程序累积了它读过的所有文件的安全标记，而这成为了它所写的每个文件的默认安全标记。因此如果一个程序读过了“机密”和“加密”文件，其后它所创建的文件将（至少）是“机密加密”级。这将包括其他文件的临时副本。如果稍后它读到了一份“绝密水仙花”文件，之后它创建的所有文件都将被标示为：“绝密加密水仙花”，而且它将再也不能创建标示为“秘密加密”的临时文件。这种效果对于应用程序来说体现了多级安全的严重复杂性；大多数应用程序因此需要被重写（或者至少要修改）才能在MLS平台上运行。

最后，即使经过改进BLP也没有什么价值，它依然没有处理好主体和客体的创建以及损坏问题（而这是建立一个真正的MLS系统面临的重要问题之一）。

7.3.4 可选模式

多级安全属性可以用许多方式表达。第一个多级安全策略是高水标原则的一个版本，是1967年8月为ADEPT-50写的。它是为IBM S/360大型机开发的一个强制访问控制系统[798]（它应用了级别、分割和组三个概念，组可以代表文件、用户、终端和职位）。因为程序（而不是过程）是目标主体，因此容易受到特洛伊木马的攻击，而且它比所需要的系统实现更为复杂。但它为BLP奠定了基础，也促进了现在的IBM S/390大型机中的硬件安全体系结构的出现[394]。

此后不久，几个工作组建立了格子模型的原始版本，我们将在8.2.1节中更详细地讨论它。这也对Bell-LaPadula的研究工作有很大帮助，同时Honeywell的工程师们当时正在致力于Multics的研究——这引出了名为SCOMP的系统，我们将在第7章的后面讨论它。

不干扰是1982年由Joseph Goguen和Jose Meseguer提出的[339]。在有这个属性的系统内，高级的行为不会被低级察觉。不可扣除性的限制性弱一点，是1986年Sutherland提出的[743]。它可以尝试证明低级（用户）无法百分之百地确定推断出高级（用户）的输入。低级用户可以看到高级用户（文件）的行为，但是无法理解它；一个更严格的定义是任何高级输入的合法字符串都和低级别事件的字符串是兼容的。因此，低级用户可以看到的任何跟踪记录，都有一个类似的不包括高级输入的跟踪记录。但不同的低级事件流可能需要高级别输出的变化或者重新排序高/低级别事件序列。

不可扣除性被提出的目的是寻找一个可以处理网络申请的模型，局域网中既有低级别用户也有高级别用户，其中高级用户对其LAN的通信进行加密（这点还需要很多工作才能保证，将高级通信的内容以空信号打包，使低级用户无法做出通信分析，还要保证信息包大小相同——[659]是这样一个系统的例子）。

不可扣除性是有历史意义的，它是根据Goguen和Meseguer思想的第一个不确定模型。但是它非常脆弱，简直令人绝望。它无法阻止低级用户对高级输入做出有99%可信度的推论。在验证数据库的时候也出现了一大堆的问题，必须考虑到任何能从数据结构中推论出的信息（例如冗余数据的局部视图），还要考虑可执行程序的跟踪文件。这些问题将在8.3节讨论。

经过改良的模型包括广义不干扰和限制性。前一个模型是：如果一个用户在一个合法的系统事件序列中改变了一个高级输入事件，要使结果序列也是合法的，则要改变一个或多个高级输出事件。而对于后一个模型，在高级输出可能发生改变的地方加上了更严格的限制。这是由一些技术原因的需要决定的，以保证两个满足限制性的系统可以组成第三个依然满足限制性的系统（请参见［540］对这些问题进行的描述）。

Harrison-Ruzzo-Ullman 模型解决了与文件的创建和删除有关的问题，而这个问题 BLP 模型并没有涉及。它对访问矩阵进行操作并检验是否存在导致访问权限泄露的指令序列［373］。这个模型比 BLP 更引人注意，但也更加复杂，因此也不易驾驭。

John Woodward 提出了*分割模式工作站*（compartmented mode workstation，CMW）策略，尝试建立一个信息密级不固定的模型（使用浮动标记），这与 BLP 模型中的信息密级固定正好相反［809，351］。但是这个模型最终没有成功，因为标记或者趋向于浮动变化得太远太快（如果操作正确），或者趋向于浮动变化得太慢（但并不能阻止所有恶意文件的流动）。无论如何，CMW 的思想毕竟促进了真实产品的开发——虽然这种产品更多地提供了信息分离而不是信息共享。

强制类型模型，是由 Earl Boebert 和 Richard Kain［122］提出，后来由 Lee Badger 等人改进的。这个模型将每个主体都分配给一个域，而每个客体都分配给一个类型。*域定义表*（domain definition table，DDT）是域和类型之间的访问控制矩阵。这是 Unix 设置中的一个自然模型：各种类型经常被映射到目录结构中。它比诸如 BLP 模型之类的策略更具普遍性，并已经开始解决有关完整性和保密性的问题了。

现在被提出的最受研究者关注的策略模型是由 David Ferraiolo 和 Richard Kuhn 提出的*基于角色的访问控制*（role-based access control，RBAC）［291］。这个模型为强制访问控制提供了一个比 BLP 模型更全面的框架。在这个框架中，访问决策不是由用户名而是由用户在组织中正在运行的函数决定的。事务也许只能被一个指定赋予某项角色的人执行。因此此模型机制包括赋予角色身份（也包括委托）。角色或组，多年以来一直被应用于银行等组织来进行访问控制；RBAC 模型使它变得规范化。它通过允许调用某程序时调整角色身份（即访问权）来处理完整性以及保密性的问题。例如，如果一个进程需要调用一个从网上下载的不可信程序，它将失去对敏感的系统文件进行写操作的角色身份。

7.3.5　Biba 模型

许多教科书会顺便提到 Ken Biba 提出的一个模型［100］，它通常被认为是“Bell-LaPadula 的颠倒”。这个模型只顾及了完整性而完全忽略了保密性。它的关键认识是，完整性和保密性在某种意义上是对偶概念：保密性约束了谁能看到一条信息，而完整性约束了谁可以创建和修改信息。

下面给出一个具体的例子，一台电子医疗设备，例如一台心电图仪（ECG）有两个独立的模式：校准模式和使用模式。校准数据必须被保护起来以防普通用户破坏，因此普通用户只能阅读它而不能对它进行写入。当一个普通用户使仪器复位时，它会丢失当前的用户状态（如，内存中病人的信息），而校准信息将保持不变。

为了给这样一个系统建模，我们建立了一个多级完整性策略，规则是我们只能向上读（也就是用户进程可以阅读到校准数据）和向下写（即校准进程可以向用户进程的缓冲器里

写入数据)；但不能向下读或向上写，否则将导致高级目标被低级数据——潜在的不可信数据——污染。Biba模型经常按照低水标原则——前面讨论过的高水标原则的对偶概念——阐述为：对象的完整性是对该目标创建有贡献的所有对象的最低级别。

这是关于完整性的第一个正式模型。按照Biba模型工作的实际系统有很多。例如，铁路的旅客信息系统从信令系统中得到信息，但决不能影响信令系统（除非通过一个可信的接口，例如一个控制人员)。然而，极少有建立这种系统的人知道Biba模型和它的作用。

一个有趣的例外是LOMAC，它是Linux的扩展，实现了低水标策略[313]。设计它的目的是应付不明原因的来源于网上的恶意代码。系统提供了两种等级——高级和低级完整性——系统文档为高级，而网络为低级。每当程序（例如守护进程）接收到网络信息时，它将自动降级为低级。这样，即使该信息里包含攻击文件并已经成功地派生出了一个根shell，这个根shell也无法像其他的普通根shell一样改写密码文件。正如我们所期望的，许多系统任务（例如登录）都十分警觉并需要可靠的代码。但是请注意，这种方法仅仅能防止病毒对根目录的访问，而不能防止它感染低级文件区并以此为跳板延伸到其他地方。

综上所述，对于完整性的关注可以由强制类型和RBAC模型来解决。但是，在这些模型的通常形式下，当一个对象被调用的时候模型修改用户的特权，而低水标模型只是在对象被读的时候对特权进行修改。后者的策略更加谨慎，因为我们所关心的攻击有时候并非是真正调用某个对象而只是读到它（例如，用因特网上的“数据”致使缓冲器溢出的攻击)。

在第9章讨论银行和簿记系统时，将会介绍一些更复杂的模型。在这些复杂的系统中我们以双重控制机制和审计追踪等形式来保障安全的状态。

7.4 多级安全系统的几个例子

随着20世纪70年代后期的一些研究性产品（例如KSOS[99]，一个以Unix为内核的安全版本）的开发，20世纪80年代初采用多级安全策略的产品零星地出现了。直到1988年，一些公司开始为他们的操作系统开发支持MLS的版本。MLS的概念延伸到了多子安全系统的各种产品。

7.4.1 SCOMP

1983年投放市场的安全通信处理器（Secure Communications Processor，SCOMP）是最重要的产品之一，是Honeywell公司由Multics派生出的产品[311]。这是由于美国国防部认为多密级信息处理需要无多余开销实现。SCOMP的硬件和软件都经过严格的核查，它采用了最小的内核程序和四环保护（Multics是七环）以使系统更简单。它的操作系统——STOP——用这些环来维护多至32个相互独立的分割，并允许信息在它们之间适当地进行单向流动。

SCOMP被应用于例如军用邮件保护系统等专用防火墙，它只允许邮件由低级传向高级，而不能由高级传回低级[234]（通常，一个只能允许信息单向流动的装置被称为数据二极管)。SCOMP的继承者（XTS-300）支持C2G——命令和控制保护装置。它被用于time-phased force deployment data（TPFDD）系统，该系统的功能是设计美国军队的行动和相关的后勤工作计划。总的来说，在TPFDD中，军事计划在形成阶段为最高密级，然后在适当的时候降低保密级别，作为命令发布下去，以便执行。这种高级信息有意降低等级的行为引起了很多问

题，其中一些问题以后再讨论（在 TPFDD 的例子中，在决定释放一个记录之前保护装置都要先查阅它的具体内容）。

SCOMP 的一个最大的贡献就是成为桔皮书的一个模型 [240]，桔皮书也被称为可信计算机系统评价标准（Trusted Computer Systems Evaluation Criteria，TCSEC）。这是第一个关于安全计算机系统的系列标准，于 1985 年提出，至 2000 年 12 月停止使用。虽然之后它被通用准则代替，但桔皮书不仅在美国，而且在盟国之间都有着非常巨大的影响力；例如英国、德国和加拿大的国家标准都以它基础，最后这些国家标准都被包括在通用准则之中 [574]。

桔皮书允许系统在多个级别上被评估，A1 级是最高级别，向下依次为 B3、B2、B1 和 C2、C1。SCOMP 是第一个被评估为 A1 级的系统。它被当前各类研究人员公开广泛讨论。作为第一个 A1 级系统，作为被深度公开的系统，它为下一代军事系统设立了标准。要想符合这种标准现在非常困难：实际上，XTS-300 也只是被评估为 B3 级（本书不提供有关 A1 级评估正确性的形式化证明）。

7.4.2 Blacker

Blacker 是整合 MLS 技术的一系列加密设备。以前，加密设备由几个相互独立的处理器组成，分别用于密文或 Black，终止生成和明文或 Red，终止生成。如果可以协调 Red 和 Black 的处理，就可以防止许多故障的出现。这种设备还可以变得更简单，并提供更大的操作灵活性：它没有被限定在分开的两个逻辑网络中，但可以选择性地提供对加密和完整性的保证，并通过路由器相互影响。但一条更高的要求是“Red”数据不能通过“Black”数据被泄露。

Blacker 于 1989 年投入使用，它给人最深的教训是在一个密级级别的模型下，提供系统管理信息极其困难 [799]。直到 1994 年，它还是惟一被评估为 A1 级的通信安全设备 [97]，所以它对后来的系统也是有影响的。虽然它没有被广泛地使用，但目前仍在使用它的后继产品（Motorola 网络加密系统），它只有 B2 的评价。

7.4.3 MLS Unix、CMW 和 Trusted Windowing

绝大多数可用的 MLS 系统都是 Unix 的改进版本，例如 AT&T 的系统 V/MLS [15]。最初增加安全等级和标记的方法是用组 ID 记录中的某些位，然后使用它指向一个更复杂的数据结构来完成。这个启用的 MLS 属性是为了使系统内核变动最小而提出的。这类产品还包括 SecureWare（和它的衍生产品，如 SCO 以及 HP VirtualVault）和 Addamax。

分割模式工作站（Comparted mode workstation，CMW）允许不同等级的数据同时被操作人员查阅和修改，并能确保该数据的标记能够适当地进行更新。这种需求最初来自情报通信，分析家必须访问“绝密”数据，例如破译的电文或间谍人员的报告，然后准备一份“机密”级别的文件提供给政治领袖和战场上的将领们。这些报告易被捕获，因此不能含有任何危及到情报来源和途径的信息。

CMW 允许分析家在一个窗口阅读“绝密”文件，并在另一个窗口撰写另一个“机密”文件，并且有专门的机制防止意外地将前者复制到后者（就是说，只能从“机密”到“绝密”进行剪切—粘贴操作，反之是不可以的）的操作。CMW 已经被证明在军事、后勤和缉毒方面都很有用 [396]。

在视窗系统中开发强制访问控制时出现的一些工程问题可以参见 [273, 274]，这种视窗系统描述了 Trusted X 的原型，Trusted X 是一个实现 MLS 但信息没有等级标识的系统。它在每个敏感等级都运行一个 X Windows 的实例，只有少量的可信代码允许用户从低层级剪切粘贴到高层级。对于与 Sun 的 CMW 产品有关的特殊问题的讨论，请参见 [281]。

7.4.4 NRL 泵

很快人们就意识到，简单的邮件保护和密码盒会受到很大限制，而除了邮件之外的许多网络服务已经迅速发展起来。传统的 MLS 机制（例如盲目的 write-up 和定期的 read-down）在实时服务中效率很低。

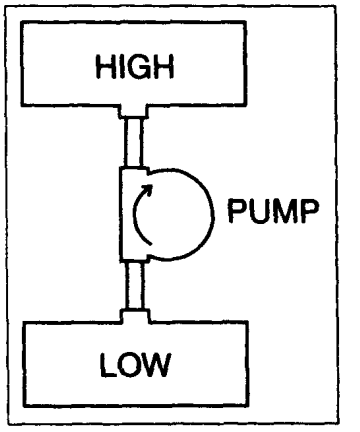


图 7-3 NRL 泵

因此，美国海军研究实验室（Naval Research Laboratory, NRL）提出了泵的概念（见图 7-3），这是一个带有缓冲器的单向信息传送设备（数据二极管），它允许信息单向流动，同时用一些机制（例如定时随机确认信息）来限制可能反向泄露的带宽 [434, 436, 437]。这种方法的吸引之处在于：建立一个 MLS 系统时可以用泵来连接一些相互独立的不同安全级的系统。因为这些系统不能处理多于一个安全级别的数据，它们可以用便宜的商业现货供应（commercial-off-the-shelf, COTS）组件来建造 [438]。随着硬件价格的下跌，使用泵成了一种最佳选择。

澳大利亚政府开发了一个名为星光的产品，它使用了泵技术并结合了键盘切换，是一个好的 MLS 类型的视窗系统（虽然没有任何可视的标记），它使用了一些可信硬件来将鼠标和键盘与高、低级系统连接起来 [17]。系统中没有可信的软件。它与 NRL 泵集成在一起。许多半商业的数据二极管产品也已经被引进了。

7.4.5 后勤系统

军方储备，例如政府的文档，有不同的密级级别。一些有标记的情报设备是“绝密”的，而喷气式飞机的燃料和鞋带就不是；但如果它们的数量和动向会泄露战术意图的话，这种简单的日用品也会称为“机密”。还有一些特殊情况，例如，惯性导航系统在和平时期属于“秘密”等级，而它可能含有属于“机密”等级的激光陀螺仪平台（因而，它的安全等级是变化的）。

由于这个系统需要解决的都是难题，在美国和英国 MLS 后勤工程都花费不菲。英国皇家空军的后勤信息技术系统（LITS）是一个为期十年（1989～1999 年）、耗资五亿美元的工程，而它的目的只是为 RAF 的 80 个基地建立一个仓库管理系统 [571]。它被设计为具有两个级别的操作：飞机燃料和鞋油是“受限”级别；特殊储备例如核弹头，是“机密”级别。最初它们作为两个独立的数据库系统进行操作，并用泵连接以保持 MLS 的属性。这个工程成了名副其实的随着需求的缓慢爬行而费用却乘电梯逐步升高的工程。这些改变之一是随着冷战的结束放松了密级规则。结果发现，几乎所有的“机密”信息都是很呆板的（例如，空投核弹的操作手册，现在被放在战略储备库而不是空军基地）。为了省钱，现在“机密”信息都被刻在 CD 上或者在一个安全地点锁起来。

后勤系统也经常应用特殊的安全功能特性。最经典的例子是军火控制系统会对把炸药和雷管放进一辆卡车或一个弹药库的用户发出警告，安全原则将会因此被破坏 [563]。

7.4.6 紫色的 Penelope

近年来，大多数的政府信息安全机构都无法阻挡运行标准应用程序（例如 MS Office）的用户需求，而这些应用程序不能用于多级安全平台。一个解决办法就是“紫色的 Penelope”。这个软件来自于英国防卫评估与研究机构，可以在一个 Windows NT 的工作站上放置一个 MLS 包装器。它执行 BLP 的高水标版本，在后台显示当前设备的安全等级，并能在阅读更敏感资源时对安全等级做必要的升级。它可以保证作为工作结果的产品得到正确的安全分类。

紫色的 Penelope 允许用户给他们的输出自由分配安全级别，而不是像传统的 BLP 系统一样禁止用户降低安全等级。然而，涉及到降低安全等级的时候，用户必须通过一个可信的路径接口来确认文件被释放，这样可以确保特洛伊木马或者病毒无法不被察觉地释放恶意代码。当然，一个聪明的恶意程序可以把需要保密的材料混到用户要释放的材料里，因此还需要其他的技巧防止这种事情发生。审计追踪可以提供所有降低保密级事件的记录，所以各种错误和攻击（无论来自于用户还是恶意代码）都可在事后被追踪到 [620]。

7.4.7 未来的 MLS 系统

MLS 业界发现了一个机会，即将他们的产品作为防火墙、网络服务器和其他易受攻击系统的平台。MLS 平台在发现和排除安全脆弱性上取得了很可观的成就，因此它能比其他常用操作系统提供更高的安全保障，即使防火墙或网络服务器的软件受到攻击，其底层的操作系统也依然安全。一般做法是用 MLS 平台将可信网络和不可信网络区分开，然后再用简单的代码实现可控的旁路。实际上，一个最主要的防火墙生产商（TIS）直到最近才集中力量开发 MLS 操作系统，而 Secure Computing Corporation、Cyberguard 和 Hewlett-Packard 已经开始提供基于 MLS 的防火墙产品了。MLS 系统一直被用于泵和邮件保护，这意味着 MLS 研究者对于防火墙的问题都有很好的理解（一个典型的设计方案可以参见 [162]）。

但是，在很多商业环境下，BLP 控制器相对于它高额的开发费用，并没有提供足够的保护，而其他面向更大规模市场的产品由于有大量用户的反馈，可以改进得更好。例如我们发现，同一公司生产的两种功能类似的防火墙产品，一种采用 MLS（Standard Mail Guard，标准邮件保护）系统，另一种则瞄准市场（Sidewinder），采用开放源代码。对于用户来讲，前者“永远不能像后者一样尽力”。

也许多级系统真正的前途不是保密性，而是完整性。很多面向专有领域的系统都多少实现了 Biba 模型的某种形式（尽管它们的设计者可能从未听说过“Biba”这个词）。在一个电力应用环境下，关键操作系统，如电力调度，是不能互相影响的；收费系统可以监看它而不能影响它。同样的，收费系统和电力调度系统都把信息输送给查错系统，依此类推，这个链的终端是执行信息系统，它可以观测所有信息（或至少所有信息的摘要），但对操作没有直接影响。

研究人员现在正在构造保密性和完整性兼容的模型，用以观测它们的相互影响并试图解

决如何将它们应用于像智能卡这样的环境 [440]。另一个研究主题是强制访问控制模型如何能提供实时性能担保，以防止受到拒绝服务攻击 [552]。显然从多级保密系统中得到的教训都已经过时了。同样，还有一些故障模型将在下节讨论。

7.5 哪里出了问题

事实上，工程师们从失败的系统中得到的启示比从成功的系统中得到得更多，而 MLS 系统就是一个高效率的老师。人们为了建立一个策略简单而确信度高的系统付出了很大的努力，已经引发了对许多信息流控制的二阶和三阶结果的深入阐述。本节将从阐述理论开始，至商业和工程的角度结束。

7.5.1 组合系统

现在考虑这种情况：一台简单的设备接受了两个“高级”输入 H_1 和 H_2，进行多路传输，用它们和一次一密密钥（也就是一个随机发生器）进行异或运算加密，在 H_3 中输出该一次一密密钥另一个副本，输出密文，此密文是由一个安全性极高的加密系统产生的，该密文被认为是“低级”输出（用 L 表示）。如图 7-4所示。

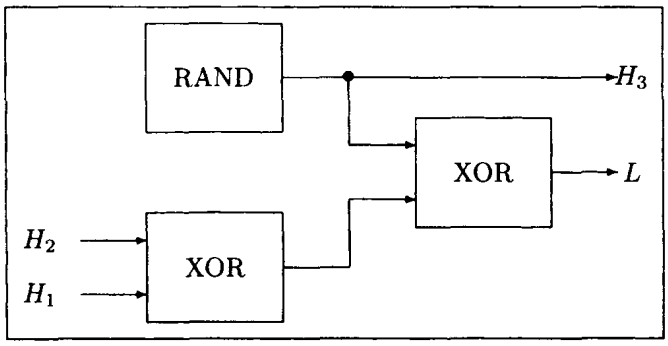


图 7-4 带反馈的安全系统的不安全组合

隔离开来看，这个设备被证明是安全的。但如果允许反馈，输出 H_3 可以反馈到 H_2，其结果是高级输入 H_1 成为低级输出 L。

时间的不一致也会导致两个安全的系统组合为一个不安全系统（参见 McCullough 的例子 [534]）。简单的信息流是不能组合的，它们不满足不干涉性和不可扣除性。一般来说，如何把两个或更多个安全的组件组合成一个安全的系统是一个很难解决的问题，即使是在最理想的组件并提供最“干净”的结果的情况下，也是如此。绝大多数问题来自于系统中各种各样的反馈，如果没有反馈，一些形式化模型是可以进行组合的 [541]。但在实际生活中，反馈是无处不在的，因此安全属性的组合将因为一些细节的接口问题和功能部件的互相影响而变得复杂。

最后，设计两个采用不同安全策略的组件的组合则更困难。这对 BLP 的不同变种的组合来说显得非常不利，而如果其中一个组件采用非 BLP 类型的安全策略就更加麻烦，这种情况我们将在后面两章讨论。

7.5.2 串联问题

串联问题给出了一个实例，说明多级安全系统之间的组合非常困难（见图 7-5）。桔皮书引入了一系列的分级评价标准，它规定了一个系统的级别跨度规则。例如一个评价为 B3 的系统可以允许不保密级别用户处理机密信息，或秘密级别的用户处理绝密信息，而不能允许一个有不保密级别的用户处理“绝密”信息[244]。

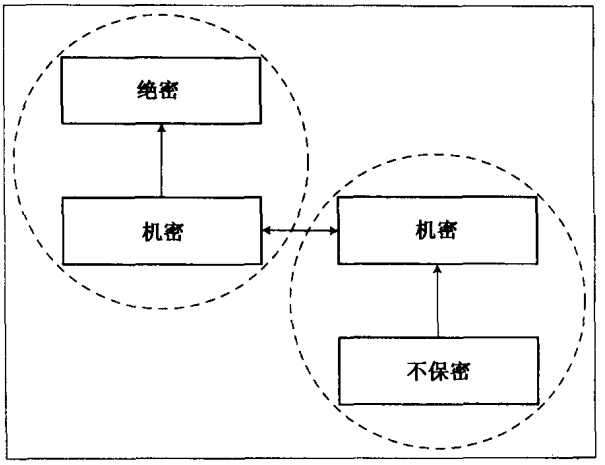


图 7-5　串联问题

如图所示，以破坏安全策略的方式直接连接两个 A1 系统。第一个系统连接不保密和机密，以它的机密级别与第二个系统通信，第二个系统也可以处理“绝密”信息（这个问题我们将在[391] 中详细讨论）。它提出了一个形式化安全模型（以及实际实现）需要考虑的另一种危险。

7.5.3 隐蔽通道

这些被强加到多级安全系统的跨度限制的原因之一来自于一个著名而广泛的问题：隐蔽通道。Butler Lampson 于 1973 年第一次指出了这个问题 [488]：隐蔽通道虽然不是为了通信而设计的机制，但可以错误地用于从高级到低级传输数据。

一个典型的隐蔽通道的情况是：一个高级进程可以通过影响它和一个低级进程共享的资源来对其传送信号。例如在 t_i 时刻，将磁头置于驱动器外侧则将高级文件第 i 位标记为 1，将磁头置于驱动器内侧则将该文件第 i 位标记为 0。

所有资源共享的系统都必须在隐蔽通道容量、资源利用和共享（公平性）之间取得平衡。如果一台机器由高级用户和低级用户共享，而且这些资源没有被分配在固定的程序片里，则高级进程可以通过填满磁盘驱动器或占用大量的 CPU 或总线（前者称为*存储通道*，而后者则被称为*时限通道*，实际上它们经常互相转变）来对低级进程产生影响。还有许多其他情况，如顺序处理 ID、共享文件锁和文件的最后访问时间；在多级安全模式下重复执行这些任务是一项浩大的工程。为了使所占用的带宽最小，人们采用了各种策略。例如，我们可以用调度程序给每个安全级别分配固定的磁盘空间，在每次向下控制发生时都读一次引导扇区；我们也可以给每个级别都分配一个在可用处理时间片中所占的固定比例，不过这个比例不能经常改变。每次改变都可能是一个或多个位被标记，而这种策略会显著地降低可用带宽（在 [435] 讲到一个更复杂的多级设计方案，每个级别都有局部调度程序，外加一个全局调度程序用于维持整个系统的一致性）。

还有一种可以限制隐蔽通道容量的方法是引入噪音。一些机器因此设有随机系统时钟。但有些隐蔽通道容量几乎是永远存在的（[353] 讨论了用以分析隐蔽通道容量和系统性能之间平衡的技术）。

隐蔽通道也会出现在应用层。一个医学方面的例子是，在英国，从泌尿生殖临床医疗诊所（GUM）得到的私人健康信息是高级保密的。没有病人的同意，这些信息不能被其普通的医生看到，也不能出现在普通的病历（低级保密）里。当保险公司未能想起一个妇女曾做过子宫颈涂片检查时，如果她在 GUM 的临床信息被泄露，则有关的医生会被追究责任［551］。保险公司知道该妇女已经在 GUM 做过子宫颈涂片检查，因此就不想付第二次钱（有些人也许会说这是 Polyinstantiation——在本章中讨论——的失败，或者是一种推理攻击——这将在 8.3 节涉及到）。

关于带宽的最坏的一种情况是有关一个特殊应用程序的某种特性。它出现在大型雷达预警系统中，高级（雷达处理器）控制数以百计的天线单元以高速脉冲序列辐射低级（目标），高速脉冲被伪随机信号调制过以防止人为干扰。在这个情景下，雷达编码必须是可信的，因为此时隐蔽通道的带宽达到每秒兆位。

最受关注的是多级完整性系统，例如银行业和公用事业计费系统。在这样的系统中，如果一个程序员把特洛伊代码插入一个高完整性的簿记系统中，则他可以把收费以特定的操作方式转到另一个账户上（例如在一个电话系统中，这个人可以连着拨打三个号码）。虽然平衡控制对这种情况有所帮助（这些将在第 9 章讨论），但编码复审仍是防止这类攻击的惟一方法。

在多级安全分时共享操作系统中，开发者最多可以把隐蔽通道带宽限制在大约 1bps（就是现在的 DoD 目标［241］；用以做出系统分析的技术见［448］）。如果目标是防止大的 TS/SCI 文件——例如卫星摄影图片——从 TS/SCI 用户到“机密”用户的泄露，1bps 的隐蔽通道带宽是可以容许的，并且 1bps 也远小于恶意代码把数据藏在输出流量中并通过安全检查的速率。然而，它不能有效地防止密码系统密钥的泄露。这也是军方为什么更愿意用特殊的硬件而不是软件制造密码机的原因之一。这也解释了为什么跨度限制对封闭安全环境比较宽松——因为在封闭安全环境下，应用程序代码只能被具有适当许可的人员引入（而且可以充分保证系统应用程序不受病毒感染）。在这种情况下，一个 A1 系统可以同时处理绝密和不保密的数据［244］。

7.5.4 病毒威胁

据发现大量的病毒主要存在于市场占有量很大的产品中，如 PC 机和 Mac 机中。但是，当 1983 年 Fred Cohen 用病毒轻易地穿透了多级安全系统时，计算机安全防御界震惊了。一个文件病毒只用了 8 小时就穿透一个以前被认为是多级安全的系统［192］。

有很多像这样用病毒和其他恶意的代码进行攻击的方法。如果访问监控器（或者其他的 TCB 组件）能被破坏的话，病毒就能把整个系统呈现在攻击者的面前，例如发给他一个未授权的许可。这就是为什么密闭的安全环境中会应用稍微宽松的规则的原因。但是即使 TCB 没有被攻击过，病毒仍然能用任何可能的隐蔽通道把信息发出去。

在许多情况中，TCB 能提供一些对病毒攻击的防护，以及对用户或者应用程序软件疏忽泄露的防护——这种泄露甚至比恶意的泄露更为严重。但是，病毒对于军事学的主要影响在于对多级安全增加了被感知的例子。观点是：即使人员能被信任，人们也不能仅仅依赖于技术手段而没有整体的隔离措施来防止病毒传染整个系统，所以人们必须采取任何可能的合理措施来防止它们被发送回去。

7.5.5 Polyinstantiation

另一个非常困扰多级安全研究人员的问题是 Polyinstantiation。假设系统的高级用户创建了一个名为 agents 的文件，而现在该系统的低级用户也想做同样的事情。如果 MLS 操作系统不允许这样做，则就泄露了一个信息——即已经存在一个名为 agents 的高级文件；但如果系统允许，那就会出现两个 agents 文件。

级别	货物	目的地
机密	导弹	伊朗
受限	-	-
不保密	备用发动机	塞浦路斯

图 7-6　美国处理分级数据的方式

级别	货物	目的地
机密	导弹	伊朗
受限	保密的	保密的
不保密	-	-

图 7-7　英国处理分级数据的方式

通常，我们可以用命名惯例来解决这个问题，就是给高级用户和低级用户分配不同的目录。但对于数据库来说，问题还是很难解决 [669]。假设一个高级用户要把已经分级的货物装在一艘船上，而由于系统不允许将此信息泄露给低级用户，那个低级用户将以为船是空的，并试图在船上装其他货物，甚至改变这艘船的目的地。

对于这个问题，在美国最常见的解决方法是：高级用户在分配真正的高级货物的同时，分配给它一个“伪装故事”。因此，基本的数据将会如图 7-6 所示。

没有信息自由法案的英国解决这个问题更简单：系统对试图打开或修改高级文件的低级用户自动回复“机密的”。这种方法如图 7-7 所示。

这使得系统工程更简单。同时防止了由伪装故事产生的错误和隐蔽通道（例如，一个低级用户试图将一些军火运往塞浦路斯）。但缺点是每个人都希望使用可能的最高级别来完成他们的工作（当然，实际上仍然需要给机密任务加上伪装故事，因为这样可以不必泄露它的存在）。

7.5.6 其他一些实际问题

多级安全系统在建造和开发时都面临惊人的花费和困难。下面列出了造成很多花费和混乱的来源：

- MLS 系统通常体积较小，并且由于军方采购机关使用了精细的文件、测试和其他质量控制措施，而使它达到物理健壮性的标准。
- MLS 系统有特殊的管理工具和程序。即使一个受过专门训练的 Unix 网络管理员，也必须经过进一步的培训，才能承担起 MLS 管理员的职务。一项 USAF（美国空军）的调查表明，许多 MLS 系统的功能没有得到完全的发挥 [624]。

- 许多应用程序必须被重写或起码需要经过很大修改才能在 MLS 系统下运行 [655]。例如 CMW 系统，该系统可以在不同的窗口里显示不同安全级别的信息，并禁止用户从高级到低级进行剪切—粘贴操作，但其代码经常在处理彩色地图时出现问题。对文件的访问可能会非常不一样，要访问控制列表的格式也是如此。另一个和商业软件的冲突源于许可服务器，如果高级用户调用一个应用程序，该程序需要向许可服务器申请执行权限，一个 MLS 操作系统会迅速地重新将服务器分级为高级并否决它对低级用户的访问。所以实际上，我们最后经常采用如下方式之一来解决：(a) 使用两个独立的许可服务器，这样就违反了许可条款；或者 (b) 拥有一个跟踪各级许可的 MLS 许可服务器，并且必须是 TCB 的一部分（取决于你选择的平台）；或 (c) 只在某一个级别上许可使用的软件。
- 在遇到一个新的标记时，程序会自动调整级别，因此程序所打开的文件也必须如此。新的文件被默认为可能的最高级别。结果会逐渐导致文件的安全级别过高。
- 通常对“盲目写”的处理很不方便：当一个低级应用程序向一个高级应用程序发送信息时，BLP 禁止发送回任何确认信息。这种效果就像信息消失在“黑洞”中。对这种情况的反应是不同的。一些组织把这个结果当作必然的状况来接受，就像 NSA（国家安全局）的一个首席科学家所说的：“当你这次向上帝祈祷时，你并不期望在下次祈祷之前能得到上帝的确认。”而另一些使用泵的用户则接受剩余隐蔽带宽作为应对方式。
- 信息密级方法并不是完全直接的：
  - 在一项军事操作的预备阶段，一些无害储备（如食品）的位置会暴露战术目标，因此其密级会被迅速升级，这符合了宁静属性不能被简单假设的原则。
  - 密级不一定是单调的，在和平时期被分级为“秘密”的仪器经常含有“机密”级别的组件。
  - 信息有时需要被降级。一个情报分析家可能会把密级为 TS/SCI 的卫星照片粘贴在一份给司令官的“机密”形式评估中，然而，信息可能被病毒隐藏在图片中，并在文件被降级时释放出来。因此，一个降级程序包括了各种不同的过滤器，例如图片的有损压缩，以及能对文本进行清除和重新格式化的文字处理程序，以期望最后剩下的信息是无格式的（关于信息隐藏将在第 20 章有关版权标记部分详细讨论）。
  - 我们会担心攻击者可得到的信息量。例如我们允许将单张的卫星照片解密，但是如果将所有照片同时解密，则会暴露我们的监测能力和我们的情报领先状况。类似地，政府的薪水册本身并不是什么敏感的文件，但记者常常可以通过研究几年间部门职员名单的变化来确定谁是以普通职业为掩盖的情报人员（我们将在 8.3.2 节的“人口普查问题”中深入探讨这个问题）。
  - 另一个相关的问题是一个处理非保密数据的非保密程序的输出有可能是保密的。这也涉及到刚才提到的人口普查问题。
- 还有一些系统组件，例如内存管理，必须对各个级别的数据进行读写。解决此问题的方法是“抽象处理”，并且假定内存管理也是执行 BLP 的可信计算库的一部分。而实际结果是操作系统中的很大一部分（加上实用程序，以及视窗系统软件和一些中

间件，如数据库）常常在可信计算库中终止。“TCB 膨胀”提高了评估的耗费，也降低了确信度。

- 最后，虽然 MLS 系统尽量防止一些不愉快的事情（例如信息泄露）发生，它们也阻止了一些友好的行为（例如使数据降低保密级的高效方式，而这对许多系统都是最重要的）。所以甚至在军事环境中，MLS 系统提供的这种功能也很值得质疑。而相关的教条也给政府系统设计者设下了各种各样的陷阱。一个最近的例子是关于英国法律要监听因特网服务供应商（ISP）（我们将在第 21 章“电子策略”讨论）。对这个法案的反对意见迫使英国政府宣布：对于某个确定的目标，由监听得到的信息是“机密”的。这样如果要使对因特网通信的监听成为可能，则必须令 ISP 重新恢复使用支持 MLS 安全策略的系统，而这种做法是不切实际的，因为它没有考虑时间和预算的允许。因此英国政府被迫宣布因为成本的原因而不再执行这个已放弃的标准。

7.6 MLS 更广泛的含义

非军方的读者一定觉得 MLS 系统是不适合的，而且它们非常复杂。虽然如此，但 Bell-Lapadula 安全模型依然是我们知道的最简单的模型。我们将在后面几章介绍其他一些更复杂的安全模型。

虽然 MLS 程序不尽如人意，但它产生了许多有用的概念和实践知识。一个安全的系统被直接侵入的方式，以及对处于第二、第三级地位的保护机制的担心，对于当前相关研究的发展是很重要的。实际建立 MLS 系统的工作促使人们解决了关于计算机安全的许多方面的问题，例如可信路径（一个用户如何能知道她/他正在和一个原版操作系统交流?）；可信发布（一个用户如何知道她/他在设置一个原版操作系统?）；可信设备管理（我们如何才能确定系统的执行正确?）。如果要把一个简化的保护事例构造成一个更细节的造型就要碰到许多我们在前面讨论的问题。结果得到的教训可以应用在采用不同策略的系统上。一个极好的例子就是最近的 Cipress 项目，一个由 Fraunhofer 学院建立的可以对数字媒体提供严格的复制和使用控制的原型系统 [149]。它的安全策略是高水标策略的一种；一个应用程序结合了一些受保护的媒体流，如果要访问所产生的输出当事人必须拥有控制输入流的所有密钥。这引发了许多前面讨论过的问题，还有：如果一个媒体的所有者废除了一些内容的访问权，则会锁住大量派生出来的工作。

这些内容在关于计算机安全的彩虹系列文档中有所陈述，彩虹系列是 NSA 随着 SCOMP 的开发和桔皮书的出版而制定的，它的思想就来源于桔皮书（这些书都是以它们的封面颜色命名的）。虽然这个系列文档在某些问题上显然是保持沉默的，例如密码机和发射安全，但它显著提高了人们对操作和评估方面问题的重视，而这些问题很容易被忽略（或被当作麻烦事留给最终的购买者去做）。事实上，技术上的保护机制和对操作过程控制的综合是安全工程中最关键的和最容易被忽视的方面之一。MLS 系统上的安全操作通常是安全链中最薄弱的环节。例如，STU-III 安全电话的最大弱点就是在讨论秘密事件时用户经常忘记按下“安全模式”按钮。一个特别的个人例子是关于前中央情报局主任 John Deutch 的。他被推测在家中用不同的电脑来分别处理保密的和不保密的材料，但一些绝密的情报通信文件在他的不保密电脑上被找到，而这台电脑被用来访问高度危险的站点。Deutch 说他不愿意冒着被同事看到的危险而使用 CIA 的保密网络。一个在他家里的外籍仆人访问了他的个人电脑。但是，这

种泄密的危险毕竟比一个入侵者偷偷潜入他家里并复制了他的硬盘要小些。CIA 检察长办公室对这次事件做的调查报告对每个关心安全的人来讲都是个很好的读物 [761]。关于这个问题在第三部分中还有更详细的介绍，还有其他一些案例贯穿全书。

从整体考虑，MLS 模型的影响不全是正面的，其中存在一个战术上的问题和一个战略上的问题。

战术问题是指可信系统组件的存在，加上一大套官僚政治的指南，产生了很强的取代危机思想的趋向。设计者们所做的仅仅是给组件选择一个他们认为适当的安全级别，然后将该安全级别的描述写到整个系统的安全说明书中，而不是有条理地解决系统对于安全性的要求 [624]。

决不能忽视操纵一个系统设计方案的人的动机以及由此带来的成本。Daniel Moynihan [562] 做了一项评论性的研究，是关于美国的外交和军事事务方面的进行强制保密的真实意图和高额成本。按照一个参议员的提问，他发现 Truman 总统从来没有被告知 Venona 电文的内容，因为这被认为是“军队所有”的材料——尽管这是起诉 Alger Hiss 的主要动机。如他在书中写道：“各个部门储藏着信息，而政府变成了一个交易市场。秘密变成了组织的资产，不能被共享，而是用来交换其他组织的资产。”例如 Moynihan 的报告指出，至 1996 年，拥有最原始安全级别分级权的机关减少了 959 个，达到 4 420 个（随着冷战后的预算紧缩），但 1996 财政年度统计安全级别分级行为增长了 62%，达到 5 789 625 件。

尽管需保密的信息急剧增长，情报的质量却在不断下降。政府部门之间的内部分歧和拒绝共享信息，以及缺乏客观的批评使情报的效率大打折扣[⊖]。一个 MLS 系统让分级程序更简单而同时数据共享控制更严格，事实上是削弱了系统的操作效率。

战略上的问题是多级安全在政治和一些工业中已经确立了地位，而且经常被不恰当地使用。一些资深的情报内部人员也证明了这一点 [425]。在很多时候我们需要做一只“狐狸”而不是“刺猬”。在一个简单的、强制的访问控制系统适用的环境中，我们经常需要控制信息的交互流动，而不是信息的向下流动。医用系统就是一个很好的例子，我们将在下一章讨论它。

7.7 小结

多级安全系统在一些军事应用中作为专门的防火墙（邮件保护和泵），也可以在一般的防火墙和网络服务器上作为很好的平台。更广泛的重要性来源于两个事实：从 20 世纪 70 年代中期开始多级安全系统就是计算机安全研究的主要课题，而多级安全系统的假设是许多安全评估方案的基础。让执行者们同时认识到 MLS 系统的能力和局限是非常重要的，这样他们可以在 MLS 系统适用的时候利用大量已有的研究成果，而避免在 MLS 系统不适用时陷入困境。

研究问题

多级保密对于下一代研究人员来说似乎已经逐渐“死亡”了。现在的研究机会在于多级

⊖ 虽然高级官员在谈到这个问题的时候会遵守官方的规则，私下他们也在责备官僚政治带来的惩罚。我最喜欢的一句妙语是一个被激怒的英国将军说的：侏罗纪公园和国防部门的区别到底是什么？一个里面都是恐龙，另一个里面是一部电影。

完整性和多级安全系统与采用其他安全策略的系统之间的交互：比如，一个军队医院怎样把BLP和簿记及后面两章要讨论的病人隐私策略结合起来。

参考资料

一个对MLS系统，尤其是关于数据库方面问题的更好的介绍，是Gollmann的《Computer Security》[344]。Amoroso的《Fundamentals of Computer Security Technology》[15]是关于BLP的不干扰性和不可扣除性安全模型的形式化数学模型的最佳介绍。

大多数公开发表的关于多级安全系统的论文，都可以在以下三个会议的会议录上找到：IEEE安全与隐私研讨会（Symposium on Security & Privacy）（因在Oakland举办，也以此命名）；国家计算机安全会议（National Computer Security Conference）（1995年重命名为National Information Systems Security Conference），其会议集由国家标准和技术研究所出版；还有计算机安全应用会议（Computer Security Applications Conference），其会议集由IEEE出版发行。Fred Cohen在他的书《A Short Course on Computer Viruses（计算机病毒的一个短课程）》中记述了他用病毒破坏MLS系统的实验[192]。这个领域中的一些早期的经典论文被收录在NIST档案文件中[573]。最后，关于公共部门中滥用分级程序以掩盖浪费、欺骗和管理不善这些方面的著作是Leslie Chapman所写的[176]。

第8章 多边安全

隐私只是一个暂时的概念，它产生于人们不再相信上帝能够洞悉一切的时候，结束于政府明白那只是一个需要填满的空白的时候。

——Roger Needham

你没有一点隐私吗？赶快摆脱这种状况吧。

——Scott Mcnealy

8.1 引言

一般来说，我们的目标不是阻止信息在部门的层次间“向下”流出，而是阻止信息在部门间横向穿过。从医疗保健到国家情报的相关应用程序，包括大多数存有顾客、市民或者病人的个人隐私资料的应用程序都是不安全的。它们在信息处理系统中占有重要的比例，但是它们保护措施的设计和实现经常都很差，这就导致了大量损失惨重的事件发生。

在这样的系统中，信息流动控制边界不是像 Bell-LaPadula 模型那样（见图 8-1）是水平的，我们需要边界大部分是垂直的，如图 8-2 所示。

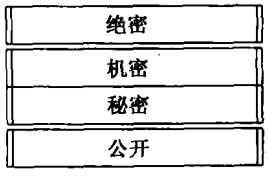


图 8-1 多级安全

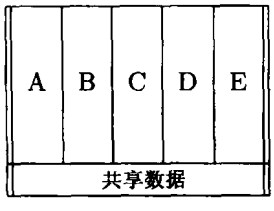


图 8-2 多边安全

这些横向信息流动控制是有组织的，如在一个情报组织中，需要使一个驻外秘密工作机构的名字对监视其他国家工作的对应部门保密。这些信息也可能基于一定特权，像在一个律师事务所，那里不同的当事人事务和不同搭档的当事人必须分开。它们甚至可能是前面两种情况的混合，就像在医药行业，病人的隐私在法律上是病人的权利，但是医院经常会迫使某个特定的部门了解其有限的信息。

横向信息流动控制是一个普遍的问题，下面采用医疗系统作为一个分析清楚和深入研究的例子。这些系统的问题很容易被非专家所理解，并且这些问题有相当大的经济和社会重要性。这里我们不得不说的是：对于那些限制只有特定团队或部门才能访问的特定密级数据的其他行业或政府部门应用来说，这里提供的问题同样会发生，只是稍有改变甚至没有变化。

一个次要的问题是这种安全技术的术语。我们感兴趣的这种类型的信息流动控制采用了许多不同的名字；例如在美国情报部门中，它们被称作分割安全或者分割。我将要采用欧洲的术语，多边安全，因为保健应用程序比情报应用范围大一些，后面的术语也包括匿名等技术的应用——一个经典的不能识别（de-identified）的有关医疗记录数据库研究的例子。这是多边安全的一个重要部分。像阻止公开的信息流动一样，我们也不得不阻止信息通过公布的统计和票据数据泄露出去。

不能识别的数据有着更广阔的应用。另一个例子是人口普查数据的管理。一般来说，其相关的保护技术是推理控制。尽管在术语上略有差别，但是，人口普查数据和医疗研究数据操作者所面临的问题是非常一致的。

8.2 分割、长城和 BMA 模型

在多边安全模型中，实现访问控制和信息流控制至少有三种模型。它们是分割模型，已被情报机构采用；长城模型，它用于描述在专业实践中阻止利益冲突的机制；BMA 模型，被英国医疗协会（British Medical Association）提出来描述医疗伦理道德所允许的信息流动。每一个模型都有超出其起源领域的潜在应用。

8.2.1 分割和网格模型

多年以来，通过使用码字与分级来限制对信息的访问已经成为美国和盟国政府的实际标准。最好的有记载的例子是在第二次世界大战期间的 Ultra 码字，英国和美国采用 Enigma 机对德国的加密的消息进行解码，Enigma 能够破译的事实是如此的重要以至于它值得花费任何代价来进行保护。因此 Ultra 码字的许可只给了一小部分人（除了密码专家和他们的支持人员，还包括盟国的领导人、他们的高级将军和精心挑选的分析家）。没有一个持有 Ultra 许可的人会处于有捕获风险的地方；情报人员永远不会使用这种方式以让希特勒怀疑他的重要的密码已经被破译了。这样，当 Ultra 码字讲述目标的时候，例如意大利人向北非的运送，盟军将会在攻击前大约一个小时左右派一架飞机去定位，接着用无线电波报告它的位置。这个策略被特殊的处理准则执行；例如，丘吉尔在某个急件箱子中得到他的 Ultra 码字摘要，他有箱子的钥匙而他的同伴没有。因为这种特殊的准则可能会使访问码字被认为是一种特权，而不是一个简单的许可（David Kahn [429] 和 Gordon Welchman [800] 描述了 Ultra 安全）。

今天，相同的防范用在一些地方来保护因知情人妥协而会暴露情报资源和方法的信息。例如机构的名字、密码分析学的成果、具有电子窃听装置的能力和间谍卫星的活动。密码的广泛应用导致了大量信息被分割，特别是在绝密以上的密级中。

造成这种情况的一个原因是，密级是由派生工作继承而来的，因此一篇使用源自“机密沙漠风暴”和“绝密 Umbra”的报道在理论上仅能由具有“绝密”授权和“Umbra”和“沙漠风暴”组成员读取。每一个码字的结合都是一个分割，有时候情报机构拥有超过 100 万个有效分割，管理它们是一个重要的问题；其他的机构允许有高密级授权的人们广泛地进行访问。但是当控制机制失效的时候，其损失是非常惨重的；在 Aldrich Ames 案例中，因为在反间谍部门工作，一名能够通过远程服务访问大量分割信息的 CIA 官员，几乎能够泄露美国在俄罗斯间谍机构的整个网络。

从效果上来讲，码字是一种能在计算机之前表示访问控制组的方式，它能够采用 Bell-

LaPadula 的变体进行处理，这种变体就叫做网格模型。密级和码字就形成了一个网格，数学结构中任意两个实体 A 和 B 能够表示为大小关系 A > B 或者 A < B。它们不必是：A 和 B 不能简单地相比（但是在这个例子中，对于网格结构，它们将要有一个上限和一个下限）。如图 8-3 所示，假设我们有一个码字“Crypto”，拥有“绝密”权限的人将被许可读取标识“绝密”和“机密”的文件，但是不能读取“机密 Crypto”除非他有 Crypto 的许可。这种关系如图 8-3 所示。

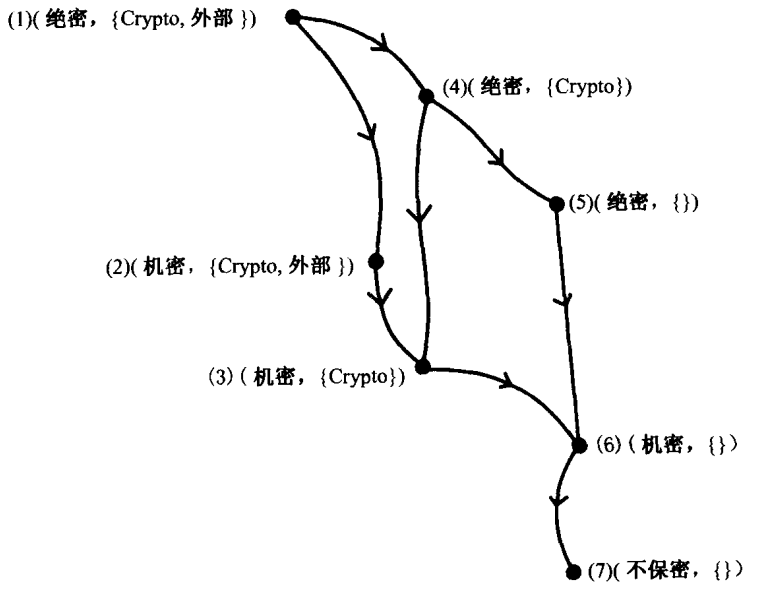


图 8-3 网格安全标识

为了使信息系统支持这个模型，我们需要提取密级、许可和标识的本质，并加入到安全策略中去，接着就可以用它们来派生安全目标、实现和评估。当发生信息流动时，Bell-LaPadula模型好像或多或少被没有变化地穿过。我们仍然像以前一样采用信息从高到低流动，高的分割支配低的分割。如果在网格中两个节点不相容——像图 8-3 中的“绝密”和“机密 Crypto”——那么它们之间没有任何信息可以传递。

事实上，Bell-LaPadula 和网格模型是等价的，它们是同时被开发出来的。

- 美国空军 Roger Schell、Peter Downey 和 Gerald Popek 1972 年提出的早期网格模型[675]。
- 剑桥博士 Jeffrey Fenton 的论文，包括一个有关使用矩阵管理标识的论述[289]。
- 大约这个时期，五角大楼世界军队指挥和控制系统（World Wide Military Command and Control System，WWMCCS）采用原始的网格模型，但是没有 * -property，在第 7 章中有一个这样的例子，面向领域的、关键的处理绝密数据的系统易受到 Trojan 所引起混乱的攻击[674]。这意味着所有的用户对于机器中最高密级的数据库需要许可。
- Case Western 大学的 Kenneth Walter、Walter Ogden、William Rounds、Frank Bradshaw、Stan Ames 和 David Shumway 提出了更先进的网格模型，使用文件和目录属性解决了许多问

题，它是对 Bell 和 LaPadula 的补充 [788，789][⊖]。

- 最后，网格模型被 Dorothy Denning 系统化和推广开来 [233]。

多级安全市场生产的大部分产品能够在分割模型中应用。但是，实际上，这些产品不像它们看起来那么有效。使用多级操作系统很容易使数据处于不同的分割中——只需给它们不同的标识即可（“Secret Tulip”、“Secret Daffodil”、“Secret Crocus”等）。但是这种操作系统将会成为孤立的机制，而不是一个共享的机制；真正的问题是怎样控制信息共享。

一个解决方法是给网格的上限强加一些算法。一个例子来自于沙特阿拉伯政府使用的管理 Haj（一年一度到麦加朝圣的活动）的系统 [385]。大多数分割默认为秘密，不同分割数据的结合是机密的，这样，“Haj-visas”和“Gov-guest”是秘密的，但是它们的结合是机密的。

在许多情报系统中，它们的用户已经在最高许可级别上操作了，数据的所有者不想进行更深的密级划分，因为在这些数据中每一样东西都是可见的。因此从两个分割派生的数据采用网格模型会有效产生第三个分割。数以万计的分割增加很难管理并且它们的应用程序相互纠缠在一起。更为一般的解决方法是采用一种标准分级产品，例如邮件保护装置，能够保证不被信赖的邮件进入到过滤器。但是现在可信计算的核心是由过滤器而不是由保护装置组成。

更糟糕的是，保护装置会失去一些底层操作系统的非常重要的功能。例如，标准邮件保护装置 [715] 是建立在叫做 LOCK 的操作系统之上的，它的基本机制是类型强制，在上下文中它被认为是一种对于进程和文件具有不可改变访问规则的系统。LOCK 的后来版本支持基于角色的访问控制，对于直接管理分割间关系来说，这是一种更合适的机制 [386]。仅仅用它当作支持 BLP 的平台是浪费的。

一般来说，情报系统的使用者所面临的真正的问题是处理不同分割数据库的结合数据和系统清理之后怎样使它降级。多级和网格安全模型在这里几乎没有什么帮助。

8.2.2　长城模型

多边安全的第二个模型是长城模型，是由 Brewer 和 Nash 提出的 [137]。它的名字来自这样一个事实：金融服务公司例如投资银行都用内部准则来避免利益冲突，被称作长城模型。

这个模型的范围并不仅仅是投资银行业。许多行业和服务公司都有需要互相竞争的顾客：软件提供商、广告代理商和会计事务所是这样的例子。一个典型的准则是“最近为某个商业领域的一个公司工作的合伙人不可以接触这个领域的其他任何公司的资料”。因此，在某个固定的时期，一个已经为壳牌公司工作的广告编写人员将不被允许为任何其他石油公司做广告工作。

因此，长城模型特征是自由选择和强制性的访问控制的混合：一个合伙人能够选择任何一家石油公司工作，但是，一旦在这个领域选定就被完全限制了。它也把责任分离的概念引入了访问控制中；一个给定的用户可以操作交易 A 或者交易 B，但是不能同时操作两者。

⊖ Walter 和他的同事理应得到比社会给以他们的更多的认同。他们在相同的研究领域首先获得了重要成果 [788]，但是由于 Bell 和 LaPadula 的成果被美国空军进行了浓重的宣传而忽略了 Walter 等人。Fenton 也被大大忽略了，因为他不是一个美国人。

长城模型吸引安全研究人员注意力的部分原因来自于这样的一个事实：它能够以一种与Bell-LaPadula模型很相似的方式来表达。对于每位顾客c，如果我们记y（c）是c的公司，x（c）是c的竞争对手，那么像BLP一样，它能够用下面两种属性表示：

简单安全属性 当且仅当一个主体s所有能读取的c′，存在 $y(c) \notin x(c')$ 或者 $y(c) = y(c')$ 时，s能访问c。

***-property属性** 当且仅当一个主体s不能读取任何c′并且 $x(c') \neq$①且 $y(c) = y(c')$ 时，主体s能写入c。

长城模型对于访问控制理论是一个创造性的贡献。它激起了有关BLP静态属性的一致性和这种系统形式语义学工作的广泛讨论（例如，Foley [300] 有关不干扰的关系描述）。关于隐蔽通道也有一些新的有趣问题，例如，一家石油公司能够及时发现采用同一家投资银行的竞争对手计划通过邀请专家作为顾问来竞标第三家石油公司，并能及时发现市场上咨询专家的数目急剧减少吗？

然而，事实上，长城模型仍然采用人工方法执行，一个大的咨询软件都为每个职员保存一份“非密级”的个人简历，它包含已经被顾客批准和同意的条款。一个典型的条款如下：

1997年9月～1998年4月： 对于美国零售银行新的分支系统的安全需求的咨询。

这不仅仅是控制。咨询公司的经理应该意识到冲突存在的可能性，如果有疑问，就不应当提供CV（curriculum vitae，个人简历）给顾客；如果发生问题，顾客可能和CV发生潜在的冲突；如果这个措施也失败了，咨询公司就有责任在问题出现的时候报告任何潜在的冲突。

8.2.3 BMA模型

多边安全系统最重要、最有趣和最有启发性的例子可能存在医疗信息系统中。在发达国家，保健领域的花费远大于军事领域的花费；虽然医院没有实现自动化，但是它们发展很快。

保健安全和（特别是）隐私在许多国家已经成为热门问题。在美国，由卫生部根据Health Insurance Portasility and Accountability Act提出的有关隐私立法的讨论在医生、病人、隐私提倡者、研究者和商人之间引起了混乱；最终的立法于2000年底颁布了。澳大利亚正在讨论是否引入智能卡以一种方便的方式记录健康保险数据。在德国（已经有这样的智能卡）正在详细讨论把重要的医疗信息（如当前处方和敏感症状）也放进智能卡中。这里的主要问题是如果当前的有关MedAlert的数据例如敏感症状记录到智能卡中，对在不能使用智能卡的飞机上或者在国外生病的病人来说就存在很大的危险。并不是所有的隐私保护技术都没有危险。

随着基因数据的广泛应用，世界各处的人们正在讨论关于隐私的标准是否应该做根本的修订。例如在冰岛，国家医疗数据库的一个项目不仅要将医疗记录，而且还要将基因和家族谱系的数据合并起来，以便于能够通过一代一代地跟踪遗传病，有关这样的项目正不断地增加。

对于医疗信息的保护也是一个保护其他类型个人信息的模型。例如银行、保险公司和政府机构的个体客户。在整个欧洲（和许多其他国家，包括加拿大和澳大利亚），那里的数据保护法律严格限制这类信息的传播。我将要在第三部分讨论保护数据法律；对于现在而言，

注意到对于一些数据类别（健康、性行为和性选择，政治和商业协会活动和宗教信仰）的数据主体要么同意数据共享，要么有权利否决它，这类模型就已经足够了。这就提出了一个问题，一个人怎样构建安全策略，其中访问控制决策不是由中央管理（如 Bell-LaPadula）或者系统的用户（如分开访问控制）决定的，而是由数据主体决定的。

下面首先看一下访问控制的状况。

8.2.3.1 威胁模型

当前，对于医疗隐私的主要威胁是社会工程学（在第 3 章曾简要提到）。对医疗隐私记录的典型攻击来自于私人侦探，他编造了一个非常可信的故事打电话给医生办公室或者健康保险人。

> 喂，我是在 Hastings 的 Conquest 医院心脏室的 Burnett 大夫。你的病人 Sam Simmonds 处于昏迷中，他的心室跳动无规律，有非常奇怪的症状，你是否能告诉我一些与他的病例相关的信息？

这种类型的攻击经常很成功以至于在美国和英国有人通过它来谋生［260］（此类攻击也不仅仅是针对健康记录的：在 2000 年 6 月，当有人打电话到税收办公室，假装英国政府部长百万富翁 Lord Levy 的身份并发现他在上一年中仅仅交了 5000 英镑的税款时［638］，这位部长非常窘迫。但是有医疗背景的例子是一个很好的讨论它的案例）。

1996 年，在英格兰做了一个实验，一个健康机构（一个政府拥有的销售地区或行政区域健康保险的保险公司）训练所有职员来筛选此类伪装的找借口的电话。他们得到的最重要的建议是他们需要回拨——但不是回拨打电话者提供的号码，而是按照电话薄中打电话者声称为之服务的医院或其他机构的电话号码。每个星期大概能够发现大约 30 个虚假的查询（在当时，全英国大约有 200 个健康机构；所给出的建议在［22］中描述）。

用这种方式训练职员比大多数的技术保护措施更重要。但是在世界上，最好的职员训练也不能保护这样的系统——大量的人访问数目众多的数据，职员总是有粗心和被欺骗的时候；他们看到的记录越多，可能发生的事故就越多。

在一个影响很大的案件中，一名强奸儿童罪犯曾经在马萨诸塞州的 Newton 市的 Newton-Wellesley 医院当过矫形外科技师，他用原来雇员的密码进入 954 个病人的记录中来得到女孩子们的电话号码，他是给她们打淫秽电话［136］而被抓到的，他已经结束了监禁的日子。但还会有很多类似的事故发生。

即使全体职员的行为都是道德的，但是技术理解的缺乏也能导致泄露信息。在二手市场买的旧的计算机或者捐赠给学校的个人计算机经常可以重新获取硬盘上的原有资料；大部分人们没有意识到通常的删除命令并没有真正地删除文件，只是标记占用的空间可被再利用而已。在最近的一个著名例子中，从一台 Morgan Grenfell Asset 管理投资银行在二手市场卖掉的计算机中已经恢复的文件包括前 Beatle Paul McCartney 的金融资料［153］。在健康记录中也有很多相似的问题。尽管全体职员是诚实和谨慎的，仍然会发生设备失窃的事情。大约 11% 的英国家庭医生有计算机被偷的经历，在一个案例中，发生了两个著名的社会女士被采用此类技巧的窃贼敲诈关于中止怀孕的事情［23］。

资源被滥用的可能性依赖于它的价值和能够接触到它的人数。在数据库中不断增加个人信息的同时也扩大了这些危险因素。简单地说，我们能够生活在一个医生接待员能接触到

2 000个病人的资料的环境中：它不时地被滥用，但这是在一个很低的可以容忍的层次。然而，如果5 000个家庭医生的接待员在一个大的美国 HMO 工作或者 32 000 个家庭医生的接待员为英国国家健康服务组织工作，他们就可以接触到数千万的病人记录，这种滥用就有可能产生严重影响了。在一个著名的例子中，美国退伍军人管理局被集体控告泄露了 180 000 名雇员的隐私；他们的系统使他们的同事（和一些病人）可以看到部分记录。隐私问题不仅仅局限于发生在直接威胁病人隐私的组织；还包括一些保险公司和研究机构随手收集的个人健康信息，在 8.3 节将继续讨论它们。

即使对于规模非常小的系统来说，横向信息流动控制也是必要的。这方面有一个来自医院系统的例子，它的设计者相信由于安全的原因，所有的职员应该接触到所有的记录。这个设计决策受到了老年专家和儿童专家的影响，他们的病人经常接受医院许多部门专家的联合治疗；不同部门系统的不兼容阻碍了他们的工作。这个系统在英国的 Hampshire 第一次投入使用，那时卫生部长 Gerry Malone 已经获得了国会议员的位置。这个系统使医院大部分职员都能够看到当地医生在医院病理学实验室做的所有检测。一名护士投诉了她的家庭医生，她在工作的 Basingstoke 医院系统发现了家庭医生为其所做检查的结果；这在当地医生中引发了愤慨，Malone 也因此在 1997 年选举中失去了他的议员位置（两次选举）[32]。

医院可以采取许多简单措施来提高现存系统的安全性。最有效的措施是将原来的病人记录保存在一个孤立的文件中，只提供给一小部分管理层职工可以将此类记录移到主系统中的权限。另一种方法是引入蜜罐陷阱，有关名人的记录中隐藏大量的虚假信息。据报道，一家波士顿的医院用肯尼迪家族的“医疗记录”来达到这个目的；浏览它们的职员能够被识别和限制。Gus Simmons 提出了一个特别灵活的建议，是调查所有咨询有关病人记录的职员并在 30 天之内不允许他们向保险公司做有偿泄露；这缓解了病人对隐私的关注，并使医院的利润最大化 [23]。

然而，简单措施的补丁工作对于安全系统来说并不是一种好的方式。我们需要一种合适的访问控制策略，思考一下优先准则和由实际威胁模型驱动的各种情况。考虑一下哪一种策略对于健康来说是合适的呢？

8.2.3.2 安全策略

1995 年英国医疗协会（British Medical Association，BMA）就面临这个问题。英国政府为国家健康服务引入了 IT 战略，它的安全策略是多级。这个想法就是艾滋病数据库要通过“机密”级别传递；普通的病人记录是“秘密”级别；管理数据，例如麻醉品的处方和治疗账单则是“受限”级别。但人们很快就意识到这种方法无法工作。例如，一个为 AZT 的处方是什么密级？它是一种麻醉品，因此它应该是“受限”级别；它检测一个人 HIV 呈阳性，因此它必须是“机密”级别。因此所有的“机密”级别的 AZT 处方必须从“受限”麻醉品处方文件夹中移去。对于其他大部分处方来说情况是相同的，当它们识别为是一个普通人接受治疗时，应该是“秘密”的。但是接着对任何人来说如何使用处方文件呢？它包括的几乎只是医生为外科手术而写的处方。

第二个问题——现在在美国已经变成了一个大问题——是基于单个电子病人记录（electronic patient record，EPR）想法的策略，这种记录从开始构思到事后分析一直跟随着病人，而不是像传统的系统那样对于同一个病人在不同医院和医生的办公室中有不同的记录，信息在医院和医生的各种处理记录之间流动。对于 EPR 安全策略的设计由于需要遵守现存道德

标准而变得异常复杂［355］。

在我负责的一个项目中，BMA 开发了一个安全策略来填补此类空缺。这个重要的创新是将医疗记录不定义为与一个病人相关的所有临床治疗的全部，而是作为与某个病人相关的事实的最大集合，相同级别的职员都可以访问它。因此，每个病人可能会有不只一个记录，这违反了 EPR 支持者的“纯化论”。但是多个记录究竟怎么样是由法律和事实决定的，依赖于你所在的国家（甚至一个州），你可能不得不使记录保持分离，像人类生育记录、性行为传播疾病记录、监狱医疗服务记录，甚至出生日期的记录（它们同时属于母亲和孩子的保健，不可能仅仅泄露孩子的资料而不泄露母亲的隐私）。随着基因数据的应用，这种情况可能更加复杂。

在许多国家中，包括所有的欧盟成员国，在法律和医疗道德规范上都给予了“病人同意”以特殊的地位。除非病人同意，记录才能被第三者分享，或者法律规定例外的有限范围，如跟踪带有 TB 等传染病患者的日常交往。这种规定随着国家不同而有所不同；在一些国家，HIV 的传播是非常明显的，在另外一些国家则不明显，还有一些国家，HIV 的数据则是秘密收集的。

因此，BMA 安全策略的目标是病人满意原则和阻止太多的人接触到大量可识别的数据库。它并不是去尽力尝试新的东西，而仅仅是对现存最好的方法进行系统整理。它也致力于表达其他的医疗记录管理的安全特征，如安全感或可记录性。例如，它必须能够重新构建过去任何时候的记录内容，以至于在发生玩忽职守的时候，法院能够确定在那个时候临床医生能够看到什么信息（需求分析的细节参见［23］）。

这个策略包括如下九个原则：

1）访问控制：每一个可以识别的医疗记录应该用一个能够读取和添加数据的个人或者群体命名的访问控制列表来标记。系统应该阻止任何没有位于访问控制列表中的人接触到记录。

2）记录公开：临床医生会在访问控制列表上公开一个有关她自己和病人的记录。当提到病人的时候，她可以公开自己、病人和相关临床医生在访问控制列表上的记录。

3）控制：位于访问控制列表中的临床医生必须被标记成对此负责。除非她改变访问控制列表，否则她只能增加除此之外的专业保健医疗内容。

4）同意和通告：当病人的访问控制列表公开的时候，对病人负责的临床医生必须注意到公开记录的病人的姓名或者后来添加的信息，无论其责任有何变化。除非是万不得已或者法律规定的例外，必须得到病人的同意。

5）坚持：没有人能够删除临床医生的信息，除非生效的日期已经过期。

6）属性：临床医生记录的所有信息应当用医疗对象的名字、日期和时间进行标记。一个审计追踪也必须保存所有的删除信息。

7）信息流：从记录 A 派生的信息可能添加到记录 B 当且仅当记录 B 的访问控制列表包含在 A 中。

8）聚合控制：应当用有效的措施防止个人健康信息的聚合增加。特别地，如果任何想加入访问控制列表的人能够接触许多人的个人健康信息，这些病人必须得到专门的通知。

9）可信计算库：处理个人健康信息的计算机系统应该有能够以一种有效的方式使上面的原则生效的子系统。它的有效性应该服从独立专家的评估。

这个策略看起来很普通，但是在技术术语中却有令人吃惊的综合性和基础性。例如，它比Bell-LaPadula模型表述更加严格；它包含第7章的BLP类型的信息流控制机制，而且包含状态（关于访问控制观点的讨论，对于一个技术型的读者来说，它能够在文［24］中找到）。

相似的策略在其他医疗实体中也被开发出来，包括瑞典和德国医疗协会、加拿大健康信息协会和欧盟（这些在文［469］中调查）。然而，BMA模型是最具有细节性的，并且能经受严格的检查；在1996年，它被欧盟医疗组织（UEMO）采用（有关此策略的公共咨询反馈可以参见文［25］）。

8.2.3.3 引导（Pilot）实现

在一个自上而下的安全工程解决方案中，应该首先决定其威胁模型，接着制定策略，最后通过在实际生活中观察它是否有效来测试策略。

BMA兼容的系统在一般的安全工程实践［374］和医疗系统中被实现。有的医疗系统使用这样的访问规则，如“护士能够看到所有她看护的病人90天之内的记录。”（医疗系统最初设计为独立于BMA项目。当我们相互了解时，我们吃惊于我们所采用的方法能够同时被医疗系统开发出来，并再次确信我们已经以一种合适而准确的方式达到了专业设计的期望）。

一个著名的教训来自于构建一个小的可信计算库是非常困难的。医疗记录系统不得不信赖病人管理系统来得知病人身份及其看护护士。在英格兰剑桥的一家医院，采用了基于不同原型的医疗系统，提供给所有职员用以证明其身份的智能卡，他们用它进行登录；结合这两种想法对访问某个特护病人记录进行授权认证可能是一种很好的方式；对于提及的组和认证都被Windows 2000所支持。从更长的时期看，人们现在正研究这样的方法，它采用基于角色访问控制的形式化方式和机制来表达医疗隐私策略（其他学术教程在［231，232，374］中讨论）。

8.2.4 比较分析

在一个给定的应用程序中，网格、长城和BMA模型，应该采用哪一个呢？单独采用网格模型是不够的，它表示孤立的分割而不是在它们之间的管理信息流动。BMA和长城解决了这个问题，但是BMA使访问权限尽可能地分散，而长城模型的访问权限是集中安排的，需要更加清晰的机制来控制逐渐增大的风险以阻止任何一个用户经手太多的数据。

在医疗数据和情报数据的保护需求方面，或者说其他方面，如律师文件、投资银行家，或者广告代理商，他们的保护需求中的差别是非常小的。一些人是非常激进的反对者，需要更加安全的保护机制；但是保护机制的力度不应该与功能混淆在一起。在所有的粗心和不忠诚内部成员的例子中，基本的威胁模型都是一样的。

事实上，基本的策略决定就是是否集中。你能更好地对付大量的小叛徒还是一个大叛徒？医生、律师和其他的专业人士更喜欢前者，而间谍好像更喜欢具有戏剧性的后者。

8.3 推理控制

在医院和其他直接关心病人的组织中，医疗记录系统中的访问控制实现是非常困难的。在次要的应用程序中，像研究数据库、费用控制和临床医生检查，确保病人的隐私更加困难。这是医生保护他们的数据比律师困难的一个方面；律师能够锁住他们的文件，从来不让

外人看到它们，但是医生在各种各样的压力下要和第三者分享数据库。

8.3.1 在医学推理控制中的基本问题

保护这类信息的标准方式是从记录中移去病人的名字和他们的地址，这样就使他们是匿名的，但是这样做并不充分。如果此类数据允许足够细的询问，那么每个人也能被识别出来，而且如果不同临床医生处的信息能够联系起来，这更会是一个特殊的情况。例如，如果我希望查找一位政治家是否出生在1946年的6月2号和是否在1967年5月8号的一场足球赛之后进行锁骨骨折治疗，从此以后他就有了麻醉品或者酒精问题。我可以对这两个日期做调查，接着我很可能从一个国家数据库中得到记录。即使出生年份代替了出生日期，如果记录是很详细的或者不同个人记录能够连接起来，我仍然有可能危及该病人隐私。例如，像这样“告诉我所有36岁有14岁和16岁女儿的妇女的记录，这样的一个母亲恰好有一个女儿患有牛皮癣”也可能在数百万的家庭中缩小研究范围。采用许多条件的复杂查询能够准确地确定研究者想要确定的东西。

由于这个原因，为医生和医院在医疗项目中提供治疗付款的美国保健财政管理局（Healthcare Financing Aclministration HCFA）保有三种记录的集合。分别是完全记录，用来付款；受益人保密记录，仅仅用病人姓名和社会安全号码隐蔽起来。这些记录也被认为是个人资料（它们也有出生日期、邮政编码等），因此仅限可以信赖的研究者使用；最后是公开访问记录，这种记录是从前两类集合中剥离出来的，病人的记录使用诸如“一个居住在Vermont的70～74岁的白人妇女”之类的术语来表示。然而，研究者发现通过用商业数据库的公开访问记录的交叉联系，许多病人身份仍然能够被识别。另外就是个人隐私支持者的抱怨，它来自于最近的General Accounting Office由于安全松散而批评HCFA的一篇报道[333]。

许多已经采用相同技术的国家使用了保健监控系统。新西兰的受益人保密医疗记录的国家数据库，只局限于一小部分特别著名的医疗统计学家接触。对于少于6条记录的查询不予回答[584]。德国有非常严格的隐私法，柏林墙倒塌之后，就迅速强制原东德癌症登记处安装了保护机制[118]。在其他国家，保护已经显得不够充分了，英国国家健康服务组织开始采用严格的制度，接着建立大量的中央数据库来使个人信息能够在政府内部使用，这使医生能够面对面处理相关信息[32]。在瑞典相似的系统由于当地隐私立法者的坚持而被取代了[685]。最具有争议的是冰岛的基因数据库，这个问题将会被简短地讨论一下。

在许多领域，不被识别的个人信息是重要的。隐私增强技术（Privacy Enhancing Technology，PET）在欧洲和加拿大被立法者作为一种广泛的隐私保护机制积极推广（和智能卡、加密和一些其他工具共同使用）。但是，作为前面所举的医疗例子，在细节数据研究者的需求和病人对于隐私的权利（或者其他的数据项目）之间有着严重的矛盾。理解这种技术能做什么和不能做什么是非常重要的。

8.3.2 推理控制的其他应用程序

推理控制问题的第一次正式研究发生在人口普查信息的应用背景中。人口普查收集了大量敏感的有关个人的数据，并根据地理（和政府）单元，像地方、地区和行政区做了简略的统计。这个信息不仅用于一般策略制订，还在很多年中决定选区和政府为公共服务提供资助

的水平。人口普查问题比医疗记录问题简单，因为它的数据仅仅限于标准的形式（年龄、性别、种族、收入、孩子的数目、所受的最高教育程度等等）。

人口普查有两种应用广泛的解决方法，依赖于数据是否在处理过程之前或者处理过程中被识别——或者处理数据的软件是否不可信任或者可以信任。

第一种处理数据的方法来自于60年代前的美国人口普查数据处理。此处理过程是对一千个记录中的一个进行磁带录音——除去了名字，准确地址和其他敏感的数据——数据中也加入干扰来阻止人们得到更多的有关个人的知识（像在一个公司中老板所付的工资）。除了样本记录之外，当地的平均人口水平能被各种各样的属性所选择；极有价值的记录（像非常高的收入）则被排除了。

这种处理的原因可能并不非常直观。但是设想一下，在一个村子中有一个非常富有的家庭，他们的收入与村子中个人的平均收入有着显著的差别，这样可以推断出这样一个假设：村子中其他居民的平均收入和附近村子的平均收入并没有什么差别。因此这就是在平均之前排除极端情况的策略。

在第二种处理方法中，可以识别的数据被保存在数据库中，隐私保护来自于对可能进行查询的控制。在这方面的早期尝试不是很成功，在美国处理人口普查数据时受到各种各样的攻击。问题是能否对包含目标个人的样本建立大量的查询，并重新得到被认为是机密的个人信息。

如果我们的人口普查系统允许范围广泛的统计查询，例如“告诉男主人收入在$50 000 ~ $55 000的家庭数量”，“告诉男主人年龄在40 ~ 45并且收入在$50 000 ~ $55 000的家庭比例”，“告诉男主人收入在$50 000 ~ $55 000并且他们的孩子已经长大和脱离家庭的家庭比例”，等等，那么一个攻击者能够很快锁定有关的个人信息。通过增加额外的外界信息来使普通的控制和其他额外的控制失效，这种查询称之为追踪者，它们通常很容易构建。

一个与推理相关的问题是一个得到大量的不保密文件的对手可能会从中归纳出敏感信息。例如，新西兰一位记者采用调查个人服务列表和寻找过去报道的方法，推断出许多GCSB（新西兰与NSA一样的机构）官员的身份[368]。报道情报官员的职位也可能发生泄密，如果一个敌对者得到该情报官员所在单位的相关内部电话号码，即使没有相应官员的名字一样也能查到相应信息。军队名单也可能被公开，电话号码是“限制”级，但是事实上，一个与军队相关的在情报部门工作的官员可能是“机密”的。结合低级别材料来得出高级的结论被称作聚集攻击。当数据库不断增大并聚集时，个人信息的增加与危险的增大是相关的(但不是相同的)。因此，提供给攻击者的背景资料越多，追踪者和其他的攻击就越容易。虽然采用多级安全策略会使攻击有一定的困难，但是推理或者聚集威胁仍能对系统安全产生影响，并且用于聚集威胁的技术和用于推理攻击数据库的技术是相似的。

8.3.3 推理控制理论

推理控制理论是由Dorothy Denning和其他人在20世纪70年代末和80年代初发展起来的，很大程度上是为了解决人口普查办公室提出的问题[234]。许多现代隐私系统的开发者没有意识到这项工作，在20世纪60年代重复犯了很多此类错误（推理控制不是仅发生在计算机安全中的问题)。下面是大部分重要观点的介绍。

特征公式是一种选择查询集的表达式（在一些数据库查询语言中)。例如“所有在计算

机实验室的教授级别的女性雇员”。最小的包含所有属性（或者它们的非）的逻辑 AND 的查询集被称作基本集合或者单元。查询集对应的统计可能是敏感统计，如果它们满足下面讨论的标准的话（例如集合尺寸太小）。有效的推理控制目标是防止公开敏感统计数据。

如果 D 是可以公开的资料集合，P 是必须保护的敏感资料集合，那么 $D \subseteq P'$，P' 是 P 的补集。如果 $D = P'$，那么说保护是准确的。不准确的保护经常需要在数据库可以回答的查询方面付出一些开销，而这样做对数据库所有者是有用的。

8.3.3.1　查询集尺寸控制

一种十分明显的保护机制是指定一个最小的查询尺寸。像上面提到的那样，当所查询的答案少于六条病人记录时，新西兰国家健康信息系统数据库将会拒绝统计性的查询。但是光靠它自己是不够的，一个显而易见的追踪系统的攻击将设置至少对六个病人记录进行查询，接着把这些记录加进它的目标。设计者选择对少量具有特殊医疗权限的统计员提供受限访问的方式，而不是通过更加严格的数据库查询控制来降低数据库效率。

尽管这样，还需要一个额外的措施，而它常常被忽略。我们必须阻止攻击者查询除一条记录以外的全部数据库记录。众所周知，如果有 N 个记录，查询集尺寸控制阈值是 t，那么意味着在 t 和 N－t 之间是必须允许查询的部分。

8.3.3.2　追踪者攻击

也许对统计数据库的最重要攻击来自于追踪者攻击。有很多这样的例子，在我们的实验室中，仅仅有一个全职教授是女性，因此我们能够通过两个查询发现她的薪水：“教授的平均薪水?”和“平均男性教授薪水?”

这是一个个人追踪者攻击的例子。也有通用追踪者攻击采用查询公式集合能够使任何敏感统计信息泄露。在 19 世纪 70 年代后期关于追踪者攻击有一个令人吃惊的发现，就是提供的最小查询集尺寸 n 小于统计总数 N 的四分之一，关于这种允许的查询类型没有其他的限制，我们可以找到大于 2n 而小于 N－2n 个统计记录集合的查询公式，这些公式会导致通用追踪者攻击。这样，追踪者攻击很容易实现，除非我们在查询集尺寸或者采用其他方式控制允许查询方面设置更加严格的限制。

8.3.3.3　更复杂查询控制

对于简单的查询集尺寸控制有大量的替代方案。例如，美国人口普查采用了“n 响应，k%支配原则)”：它不会泄露由 n 或者小于 n 的数值提供的 k%或大于 k%的信息。像上面提到的那样，其他技术也被用来隐藏具有极端重要价值的数据。当一些医疗数据库对非常见疾病采用同样的处理方式时，人口普查管理局也面临着要处理国家统计层次的高净值的个人信息，而不是地方性的数字的问题。例如，英国医疗系统禁止从地方统计中查询艾滋病的治疗药品销售资料。

8.3.3.4　单元抑制

下一个问题是怎样处理抑制某类统计带来的副作用。例如，一个大学想发布各种课程的混合平均评分，以便于人们能够检查在整个课程中如此评分是公平的。如图 8-4 所示，假设包含学习两门学科的学生数量，一个是主修科目，一个是辅修科目。

下一步假设我们的最小查询集大小是 3（如果假设是 2，那么根据学习地质学和化学的两个学生中的一个评分就能很容易地得出另外一个的评分）；我们不能公布地质学和化学的平均评分。但是如果知道化学的平均评分，那么就能很容易地通过生物与化学平均评分和物

理和化学平均评分重新建立相互关系。因此，我们不得不禁止化学行中至少一个与其他学科相关的平均评分查询；由于相似的原因，我们也禁止一个地质学列的类似查询。但是，如果我们禁止地质学与生物和物理与化学查询，那么我最好也禁止物理与生物查询来阻止依次得出这些数据。剩余列表如图 8-5 所示。

这个过程叫做补码单元抑制。如果在数据库方案中有更多的属性，例如，如果为了表明遵守反种族歧视的法律，身份也被种族和性别掩盖了，那么也许会丢失更多的信息。如果一个数据库方案中包含 m 个字节组，空白的单个单元意味着删除 $2^m - 1$ 个其他单元，和敏感统计资料一起排列在超立方体的一个顶点。很明显，即使精确的保护也能很快地使数据库不能使用（当一个数据库不是同构的时候，情况会更糟糕：有很多支点——那种能够阻止大量查询有答案的单元）。

尽管补码单元抑制有时是可以避免的，比如那些国家层次上被置于地区统计数字之外的高收入（或罕见疾病）信息，但是当我们进行微统计发布，比如进行核查级别表格处理的时候就无法避免。当数据库可以在网上公开查询的时候，可以通过隐式查询控制来达到同样的效果，如果允许对 m 个属性值查询，当且仅当已将 m 个属性设为真或假时，2^m 个隐式查询集合至少有 k 个记录。

主修课	生物	物理	化学	地质学
辅修				
生物	-	16	17	11
物理	7	-	32	18
化学	33	41	-	2
地质学	9	13	6	-

图 8-4 单元抑制之前表格包含的数据

主修课	生物	物理	化学	地质学
辅修				
生物	-	取消	17	取消
物理	7	-	32	18
化学	33	取消	-	取消
地质学	9	13	6	-

图 8-5 单元抑制之后的表格

8.3.3.5 最大顺序控制和网格模型

下一步我们要做的工作是使建立一个追踪者攻击更加困难，限制所能够进行的查询类型。最大命令控制限制了任何能够进行查询的属性的数量。要想使查询行之有效的话，限制将是非常重要的。一项研究发现 1000 条医疗记录，只有 3 个属性的查询是安全的；如果有 4 个属性，就能够发现一个个人记录；如果有 10 个属性，大部分记录都能单独分离出来。一种更加彻底的方法（当它是可行的时候）是拒绝在太多的集合中进行区分人口样本的查询。

下面看一下在分割安全中怎样使用网格模型来定义一个部分顺序和密码组合来控制允许的信息在分割间流动。它们也能在一些数据库中采用稍微不同的方式来使查询控制系统化。

例如，如果我们有三个属性 A，B 和 C（像居住面积、出生年份和医疗条件），我们能够发现任何一个单独的属性查询都是不敏感的，A 与 B 和 B 与 C 的查询也是不敏感的，但是 A 与 C 的结合可能是敏感的。对于全部三个属性的查询也是不允许的。因此，网格模型很自然地分为顶部的禁止查询和底部的允许查询。如图 8-6 所示。

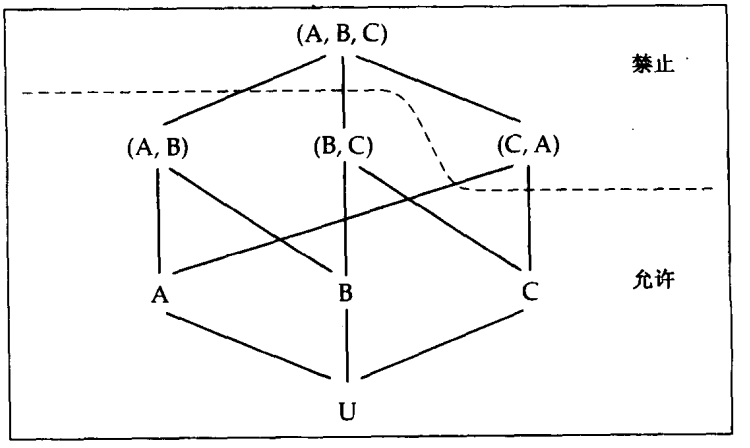


图 8-6　有三个属性的数据库网格模型图表

8.3.3.6　基于审计的控制

像上面提到的那样，一些人通过追踪用户进入数据库的行为尽力绕过静态查询控制所加的限制。一般称之为重叠查询控制，这种控制结合用户已经知道的信息，拒绝用户进行可能会泄露敏感资料的查询。这种控制在理论上好像是完美的，但是在实际中，它有两个无法克服的缺陷。首先，处理过程的复杂性会随着时间增加，经常呈指数；其次，确信你的用户不相互勾结是非常困难的，确信一个用户未用两个不同的名字注册也是非常困难的。即使你的用户现在都是诚实和不同的人，但情况总是有可能发生变化的：它们中的一个很有可能接管另外的一个，或者其中两个将来被一个攻击者所接管。

8.3.3.7　随机化

单元抑制的例子表明：在统计数据库中，如果各种各样的查询控制仅仅是一种保护机制，那么它们经常会产生令人无法接受的惩罚性结果。因此，查询控制经常和各种类型的随机化结合在一起，它被用来降低攻击者的信噪比并尽可能地减弱对合法用户的影响。

这种随机化最简单的技术是扰动，也即增加均值为 0 的干扰和对数据已知的方差。实现这种目标的一个方法是四舍五入或者截断具有确定性规则的数据；另外一种方法是交换一些记录。扰动经常不像人们认为的那样有效，当样本数据库的尺寸很小的时候，它经常倾向于破坏合法用户的准确查询结果；但当样本数据库的尺寸很大的时候（此时，我们可以用任何简单的查询控制方式），它使查询结果只是一个原封不动的整体。还有另外一个担心是相应的均值技术可能被用来消除一些附加的干扰。

一种更好的随机化技术是采用随机样本查询。这是人口普查局采用的另外一种方法。该想法是使所有的查询集具有相同的尺寸，从一个可用的相关的统计中随机选择它们。这样，所有被发布的数据来自于小的样本而不是整个数据库。如果这种随机选择采用伪随机生成数进行输入查询，那么查询结果具有重复性的优点。随机样本查询对于大型医疗数据库来说是

一种自然保护机制，用于研究的相互关系经常只要有几百个这样的样本就足够了。例如，研究给定的某一种疾病和生活方式在某些方面的相互关系，在医生建议病人对他们的生活方式做根本性改变之前，这种生活方式有令人不愉快的副作用，那么这种相互关系必须很强烈才可以提供建议。如果一个教学医院有五百万名病人的记录，其中有五千名病人患有正在研究的疾病，那么随机选择两百名患者的样本也许就足够所有的研究者使用了。

当疾病非常罕见，或者由于其他的原因只有很少的相关统计，这种方式的作用就不是很好。这时一个可能的策略是随机化响应，可以随机地限制我们选择的数据（主体响应）。例如，如果正在研究的三个变量是肥胖、抽烟和艾滋病，我们可以要求每一个受 HIV（艾滋病）感染的主体来记录他们是否抽烟或者他们是否肥胖，但是不能同时要求两者。当然，这也能限制数据的数值。

8.3.4 一般方法的局限性

无论采用何种安全技术，统计的安全性只能在某个具体的环境中和抵抗某个具体的威胁模型时才能被评估。它是否充分依赖比通常使用的应用程序细节更深的内容描述。

一个很有启发性的例子来自分析麻醉品处方倾向的系统。这里的处方是从药房中选择的（除去病人的姓名）。一种更加不能识别的处理方式是除去医生的身份；接着这些信息被卖到麻醉品公司的市场部。系统不得不保护医生和病人的隐私（一个繁忙的家庭医生最不想做的事情就是被竞争对手开出的麻醉品的处方所烦恼）。

这种早期原型系统的一个问题是在四个或者五个有关医生 A 和医生 B 等的实践中，它仅仅替换了医生的名字，如图 8-7 所示。我们知道从处方样本中，一个有区别的麻醉品处方能够识别医生的身份。例如，通过注意“医生 B 一定是 Susan Jones，因为她在一月的第三个星期去滑雪，考虑处方数量的分布程度。医生 C 可能是她的搭档 Mervyn Smith，她已经在那段时间接替她。”这个定位是用每个医生对每个特定麻醉品的处方数量的比例代替处方的绝对数，通过向后或者向前切换一些星期来实现对时间的随机扰动 [530]。

星期	1	2	3	4
医生 A	17	26	19	22
医生 B	25	31	9	29
医生 C	32	30	39	27
医生 D	16	19	18	13

图 8-7 不被识别的药品处方数据样本

一般来说，背景知识很难确定，它很可能随时间不断增加。Latarya Sweeney 已经表明：即使 HCFA 的“公开用途”的文件也经常能通过商业数据库的交互相关性被重新识别 [744]（这种数据库检测工作是评估一个给定的实际统计数据库保护级别的一个重要的部分，就像我们仅仅用密码算法来抵抗有能力和兴趣的对手所作的各种分析一样）。即使没有交互相关性，也有内部的可用的背景信息。医疗研究数据库的用户经常是那些可以对统计数据生成的部分病人记录有访问权限的医生。

主动攻击

主动攻击是非常有效的。它是指用户有能力往数据库中插入或者删除记录。一个用户能够通过增加记录来创建一个包含目标记录的组，加入大量由他创建的不存在的主体。一个（有缺陷的）对策是成批地增加或者删除新的记录。这样的极端做法是分区，当往组中增加记录的时候，任何查询必须要么完全回答，要么一点都不回答。然而，这又是一次同微统计数据图表发布相同的情形。

主动攻击对于数据来说是没有限制的，也能以元数据作为目标。有一个典型例子，由Whit Diffie提出，是关于选择麻醉品攻击。假设一家麻醉品公司通过统计系统了解到各种不同病人组所花费费用总数，为了使它的市场销售更好，它希望找出哪类病人需要何种药品（在Quebec丑闻中，有一个这样的推理攻击）。一个非常有用的技巧是以一种结果方程式很容易解决的方式设置麻醉品的价格。

一个著名的例子发生在冰岛的一个新的医疗研究数据库中，它包含三种链接数据库：一种是国家医疗记录，一种是整个人口的家谱记录和一种按先后顺序得到的基因数据。研究的基本原理是既然冰岛的人口大部分是大约一千多年前少数家族的后代，他们的基因比一般人口的基因会有更少的不同，因此很容易发现基因遗传疾病。

冰岛数据库的隐私问题比一般问题更为尖锐。例如，通过链接家族的医疗记录，在任何公开条件下（家族是冰岛一个普通的单位），通过这样的因素，像他们叔叔、阿姨、叔祖父、婶祖母等等的数量和家族族谱的形状，病人能够被有效地识别。有很多关于这种设计是否能够（甚至在理论上）符合隐私保护立法要求的质疑[33]，欧洲隐私官员严肃表达了对欧洲隐私立法系统可能发生的后果的关切[217]。然而，冰岛政府采用各种手段克服了当地医生的强烈反对使它得以通过。结果是11%的人口反对这个系统，包括大多数医疗工作者。

8.3.5　缺陷保护的代价

剥夺信息识别权利是困难的，这个问题充满了政治因素。但是做一些尝试经常是值得的，即使你所提供的保护是有缺陷的。

一些安全机制如果是危险的，那么它会比没有采用安全措施更加糟糕。非常弱的加密就是这方面一个很好的例子。世界上所有的情报处理机构面临的主要问题是怎么样从大量的国际电话、传真、电子邮件和其他通信方式中过滤出有价值的东西。一个人如果采用有助于将重要通信译成密码的方式来处理重要信息，也许会使他的竞争对手的信息过滤工作更加容易。如果采用的加密方法很容易破解（或者一个系统终端能够被破译），那么结果比用普通文本的通信更加糟糕。

统计安全一般不是这样的。个人信息数据库的主要威胁经常是任务爬行器。一旦某组织可以访问有潜在价值的数据，那么所有能够浏览该数据的方式都将被设计出来。这些方式中的一些可能会引起非常强烈的反对；一个发生在美国的典型的例子是将医疗记录转售给银行来过滤贷款申请。然而，一个有缺陷的不能识别系统也可能会破坏医疗数据对银行贷款部门的价值。如果仅有5%的病人能够识别出来，那么银行就只能告诉这些贷款申请者取消他们的保险，并让保险公司在同意的前提下分发医疗调查表。因此，不能识别的信息也是有帮助的，即使它的作用是预防将来可能的伤害而不是解决现存问题。

除了威胁隐私，任务爬行器也有其隐含安全性。至少有一个欧洲国家，糖尿病登记簿——用来监视糖尿病治疗质量的数据库——在家庭医生和医院糖尿病专家之间被滥用为基本的电子交流方式，因为他们尚无电子邮件。但是糖尿病登记簿从来没有被设计成交流系统的一部分，如果是为了这个目的的话，他们将缺乏安全和其他应该采取的安全机制。即使是最基本的不可识别系统也会对这种滥用加以阻止。

因此在统计安全中，一个人是否使自己最好的东西成为敌人的物品的问题需要一个出色的判断，而不是其他的东西。

8.4 剩余问题

前面两节已经使你相信在一个非常受关注的环境中（例如在医院中），管理医疗记录隐私的问题是相当直接的，然而在从属数据库（例如为了研究、检查和费用控制）环境中，统计安全技术的小心使用将能够解决很多问题。有些相似的技术被用来管理军队组织和高敏感性的商业数据，像即将到来的投资银行的合并和收益的数据细节等情报信息。总之，基本的原则是真正的秘密材料被限制于少数能够识别的个人分割中，更低保密级别的数据版本有更广泛的用途，这不仅包括抑制病人、间谍和目标公司的姓名查询，还能控制任何可能被重新识别的背景和其他信息。

但是，使这样的一个系统在现实中运行良好比它看起来要困难得多。首先，决定信息的敏感级别极端困难，许多原先的期望被证明是错误的。例如，你可能希望许多艾滋病患者对于他们的状况很坦率（在这里艾滋病状况是最敏感的医疗数据），你也期望人们更加信任保健专业人士而不是敏感的个人健康信息，比如信任医生和药剂师而不是商业数据库；但是，许多妇女对可购买的女性卫生产品如此敏感，以至于她们更愿意在超市中使用自动售货装置，而不是到药剂师那里用现金购买，尽管采用这种方式她们必须使用她们的商场卡和信用卡，这样与她们姓名相关的购买就永远记录在买卖数据库中。被人直接看到带有一包卫生护垫是很尴尬的，比在更年期后收到商家赠送的六个月婴儿衣服打折优惠券的潜在尴尬程度更严重。

第二，无论你采用什么样的无懈可击的分割，排除单个故障点是非常困难的。比如 Rick Ames 危及了中央情报局在苏联的布署，他是一名在反间谍部门工作的高级人员，他能接触到很多分割。克格勃海外控制系统由于 Vassily Mitrokhin 也受到了相同的危害，他是一名从1968年以后就对共产主义失望的官员，现在被送到档案室工作，安静地拿着养老金过日子[51]。

在医疗领域，许多真正严峻的问题在于医疗费用支付系统。当一个病人被治疗的时候，付费的要求被发送到保险公司，没有疾病、治疗和费用的具体细节，但是包括病人的姓名、保险号码和其他的细节，像出生日期等。有人建议采用匿名信用卡来支付费用[117]，但是据我所知，没有一家采用过。保险公司想知道哪些病人和哪些医生花费最多。这取决于保险公司是私人保险公司（雇主）还是政府所有的健康机构，例如 HCFA 或者英国的国家健康服务机构。一旦保险公司拥有了大量的病人健康信息，它非常不愿意删除它，以备万一在将来还能用到。

在美国，由保险公司、雇主和其他人保存医疗记录的副本现在被广泛认为是一个严重的问题。像共产主义倡导者 Amitai Etzioni [277] 和自由主义者 Simson Garfinkel [330]，这两个持不同政治观点的作家在这一点上是一致的，但在别的方面相同点就很少。公众的关注促使议会通过了健康保险权利和义务法案（HIPAA），它授权公众服务和健康部门（DHHS）来节制健康数据安全性。现在的讨论是立法怎样执行，如果私人医疗保险部门采用 HCFA 标准，这对大多数病人来说是一件好的事情。但是如果都这么规定了，那么每个人都可能发生无尽的推诿和诉讼。尽管这样，这个法案仅仅是 DHHS 用来规范健康计划、保健票据交换所和保健提供商等机构的，许多医疗信息过程中的组织（例如律师、雇主和大学）则在规范的范围之外。

从其他的国家我们能够得到什么样的教训呢？

像我们上面所说的那样，英国的医疗系统已经成为病人和医生协会冲突的根源。瑞士的医疗系统最初和英国的一样，在主张隐私立法者的坚持下，现在已经具有更难的不可识别性。在德国，富人使用私人保险公司（它遵守严格的数据保护法律），穷人使用医生开办的国有保险公司，除医生外其他人不能接触医疗记录。最根本的解决方式是在日本，费用控制通过规定的费用来实现：通过使医疗的规定价格低于实际费用来劝阻医生采用昂贵的治疗，例如心脏移植手术。这种机制不包含大规模的对个人健康信息的访问，比其他国家一对一的费用控制更有效。卫生保健费用占日本国民生产总值（GNP）的3%，相对于发达国家的7%～8%和美国的15%。日本人比欧洲人寿命更长，而欧洲人比美国人寿命长。1994年2月，俄勒冈州采用了一个日本方式解决方法的修正案并很受欢迎，但是保健业的说客把它当成“配给”并给予激烈的反对。

概括来说，健康记录的隐私问题从根本上来说是一个政治问题。在一个数据库中，医疗记录的数量是否增加取决于健康系统的组织方式和这些记录在费用支付之后是否被破坏——或者至少完全不能识别——这是规划的问题，而不是技术问题。在这样的讨论中，安全工程师的角色是使策略制订者明白他们行为的后果。

其他的隐私问题也牵涉到严重的政治问题。银行顾客的隐私与银行内部采取的策略紧密相连；最好的隐私保护经常来自于各部门经理不愿意其他部门知道他们的客户。接触犯罪和间谍记录依赖于法律执行机构怎样决定相互分享数据，他们内部对是否接触到高度敏感的关于来源和方法信息的选择应该是分散的（冒偶然损失风险），或者集中的（对于总部的背叛可能性很低但代价很大）。

8.5 小结

在这一章中，我们探讨了确保医疗记录隐私安全的问题。这在大量信息安全问题中很具代表性，从通过广义的专业行为来保护国家情报数据库到人口普查数据的保护。

讨论所得出有关医疗记录的问题可能是一个容易的问题、一个比较困难的问题和一个真正困难的问题。

容易的问题是设置系统访问控制以便于某一特定记录的访问只局限于全体职员中的少数人。这种系统大多数能够通过现有的工作实践来自动设计。比较困难的问题是统计安全，怎么样设计医疗记录（或者人口普查反馈）数据库以便于研究者能够进行统计查询而不会危害个人的隐私。最困难的问题是怎样管理两个数据库的连接，在特定的医药案例中，怎样阻止付费信息的传播。对于这个问题惟一现实的解决方案依赖于立法。

研究问题

在不远的将来，大量医学治疗会包括特殊的信息。你的医疗记录可能包含你父母、兄弟和堂兄弟等人的个人健康信息。BMA模型怎么延伸处理与多个人相关的医疗记录呢？

有没有一些隐私采用的访问控制策略方法来保证有关（可能是）数字方式支付费用的统计安全？这样的方法能否保证各种信息都能巧妙放置在一起的隐私安全？

有没有其他的隐私策略的论述？例如，把BMA模型和长城模型结合在一起是否是一种有用的方式？有没有定位别人感兴趣的数据主体的技术或者亚技术方式？

参考资料

有关分割模式安全的参考文献有点分散：在第8章末尾大部分引用的公开论文是NCSC/NISSC和ACSAC的会议记录。像Amoroso［15］和Gollmann［344］的标准教科书包括了网格和长城模型的基本知识。

对于BMA模型，请参看它的标准文件——蓝皮书［23］，其简略版本在［24］，有关安全策略的会议记录在［29］。有关引导系统的论文参见Hastings的［231，232］。有关日本保健更多的信息见［159］。关于美国研究医疗隐私问题的国家研究会参见［581］；在［511］中也有一篇关于使用不能识别数据的HHS研究报告。

对于推理控制，Denning的书［234］是一本经典的参考书，在［238］中有它的内容更新。一本更新的数据库安全的教科书是由Castano等人编的［172］，它的有关统计安全的章节是对Denning书的一个有用的更新，其他的章节也包括一些相关的多级安全和入侵检测问题。

第9章　银行业和簿记系统

计算机没有（还没有）足够的理由成为二级助理。

——CASEY SCHAUFLER

为了避免做蠢事，上帝在徒劳无益地斗争着。

——J.C.FRIEDRICH VON SCHILLER

9.1　引言

银行系统包括三个部分：记录客户账目信息的后端簿记系统、像联网提款机那样的交易处理系统和为银行系统提供交易数据的银行间高额资金转账系统。由于各种原因，这三个部分都很重要。

首先，簿记一直以来都是电脑行业的主要业务，银行业是簿记应用最广的领域。个人应用程序（如 Netscape 和 Powerpoint）现在可以在更多的机器上运行，但账目清算仍然是一定规模业务的核心应用程序。因此，对簿记系统的保护就显出了非常实际的重要性。同时簿记也可以抽象成一种熟知的保护模型，在这种模型中，保密性几乎不值一提，但是记录的完整性（即一旦记录就不许改动）却极为重要。

其次，无论是小型交易系统（如处理从提款机提取 50 美元这种小额交易的系统），还是大型交易系统（如处理数百万金额电汇的系统），都采用了商业加密技术。银行业应用不仅推动了加密算法和相关协议的发展，同时也推动了相关支持技术的发展，如防篡改加密处理器的发明。这类处理器是一种重要且有趣的可信计算库设备，它们与多级安全中所讨论的刚性操作系统有很大的不同。人们首先从出现在（或至少公开记录于）商业密码学领域的错误里得到很多启示。金融业密码应用者已经在那些开放的研究团体之前对有关密码术和访问控制的接口问题进行了研究。

第三，对电子银行业基本技术的了解是巧妙解决电子商务中更高级问题的一个先决条件。事实上，许多 .com 公司在基本的簿记问题上陷入困境，他们一心想着挣钱和建设网站而极易忽略这些问题。

最后，银行业系统作为多级安全的另一个例子，主要关心的是确实性而不是保密性。银行业系统必须防止客户间相互欺骗或者客户欺骗银行，也应该防止银行雇员欺骗银行或客户。同时还要提供强有力的证据，保证没有人在错误地指控他人的欺骗行为后能够逃脱惩罚。

9.1.1　簿记的起源

簿记最早出现在新石器时代的中东，大约公元前 8500 年，也是紧随着农业而出现的

[678]。当人们开始储存和交换他们生产的物品时，需要用一种方法来记录村民在公社仓库里存放物品的情况。最初的时候，每单位的食物（羊、小麦、油等）用一种泥土标记（或称为 bulla）代表，这种标记被放在一个泥土封袋里，并用仓库管理员的图案密封（见图 9-1）。当村民需要取出食物时，封口由管理员拆封，同时需要一个人做公证（这可能是已知的最古老的安全协议）。到了大约公元前 3000 年，这种方式导致了文字的发明 [609]；又经过了 1000 年，出现了相当于许可票据的东西和货物票据等。与此同时，金属锭开始用作中间商品，并常常由试金师封在一个蜡封里。公元前 700 年，吕底亚国的国王 Croesus 开始直接在金属上印上标记，从而发明了硬币 [625]；到了伯里克利时代的雅典，已经有了数量众多的富人，他们在商业中从事银行家的职业 [338]。

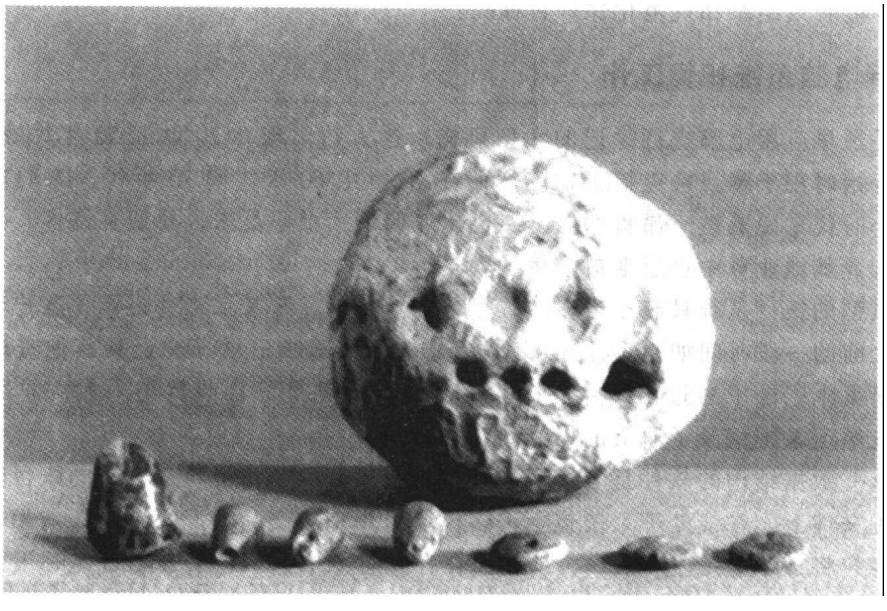


图 9-1　在伊朗的 Susa 发现的大约公元前 3300 年的泥土封袋和标记
（图片刊登经 Denise Schmandt-Besserat 和卢浮宫博物馆的许可）

另一个重要的革新可追溯到十字军东征时代。随着黑暗时代的结束和经商的开始，一些商业的规模越来越大，一个单独的家庭很难独立进行管理。最早知道的现代银行就出现在这一时期；现代银行通过在许多城市设立分行的方式有效地促进了贸易。但是随着经济的发展，需要从家庭外部雇佣管理者，而银行所有者的家庭监督不了这些管理者，这容易带来欺骗等行为，于是就出现了复式簿记的控制机制。复式簿记大约出现在 14 世纪，但关于这方面的书籍直到 1494 年印刷机发明之后才出现 [222]。

9.1.2　复式簿记

像大多数产生深远影响的概念一样，复式簿记的概念也极为简单。每次交易都被分为两个账簿，一个是贷方，另一个是借方。比如，当债务人向公司偿还 100 美元的债务时，这笔金额已被当作借款写入应收账款账簿（公司现在少了 100 美元），同时作为贷款写入现金账户账簿（公司现在多了 100 美元现金）。债务到期后这些账簿都应做到收支平衡，即相互抵消。资产和债务之间应当相等（公司创造的任何利润都是对股东所负有的债务）。账簿应由

不同的职员分别保管，并且每个月末时清算平衡收支（对于银行来说应每天清算平衡收支），那些非常小规模的公司除外。通过合理地设计分户总账系统，可以使每个商店或分店都能独自平衡收支。这种情况下职员如想有欺骗行为，必须两个或更多职员相互勾结才能做到；这种分离责任的原则，也称为双重控制，与查账互为补充。

许多电脑系统被用来完成簿记任务，并作为复式簿记的补充工具。然而，控制经常是不可靠的。复式簿记的特征也许只在用户界面上得到体现，而基本的文件格式却没有完整性控制。即使分户总账被保存在同一个系统中，某个有根访问权限的人——或有物理访问权限和调试工具的人有可能改变其中两个或更多的记录而绕过平衡控制。有很多方法都可以避开平衡控制；职员可能会注意到软件的漏洞并利用它们。

如何才能有效组织和规范化保护目标呢？

9.2　银行电脑系统如何工作

银行是最早应用电脑进行簿记的大型机构。这大约出现在20世纪50年代晚期和60年代初期，开始时用于账目检查等工作。然而在20世纪60年代和70年代，人们发现，即使应用对那个时代来说既慢又昂贵的电脑，也比雇佣一大批职员来说要便宜许多，于是自动化控制在银行业其他业务中也逐渐被采用了。

一个典型的银行系统具有相当多的数据结构。其中，账目主文件记录了客户现有的账目以及在此之前的一段时间如90天内的交易情况；各种总账记录了现金和其他资产的存取情况；各种分类账记录了通过出纳台、提款机、支票分类员等环节但还未汇入总账的交易情况；审计追踪记录则记录了职员的操作内容和时间。

处理这些数据结构的软件是一个通宵批处理程序套件，负责将分类账的账目转到各种总账和账目主文件上。在线处理系统包括许多模块，用来将交易情况送到总账的相关部分。例如，某个客户将100美元存入账户，出纳员在存款总账中记下这笔交易（此时银行对客户账户有增删能力），同时将这一账目归入现金柜中的记录现金额的总账中。查账时的一条重要依据就是所有总账应该相互抵消，如果银行（或它的某个支行）收支失衡，则系统将发出警告，这样人们就可以及时查找原因。

总账系统提供的这种收支平衡性在每天的通宵批处理营业时间内都将进行核查。这意味着，一个程序员如果想往自己的账户上加钱，则必须从别的账户上取走相应数目的钱，而不是篡改账户主文件凭空产生一笔钱。在一个传统商业公司里不同的总账是由不同职员管理的，类似地，应用银行数据处理技术的商店也是由不同的程序员来管理这些总账的。此外，所有的程序代码需接受内部审核员的详细审查并由一个独立的测试部门测试。一旦程序代码通过审查，它将运行在一台专用机上，这种专用机不具备开发环境而只能审查目标代码和数据。

9.2.1　Clark-Wilson 安全策略模型

尽管前面讨论的系统在20世纪60年代就已出现，但是关于它们的安全策略的形式化模型直到1987年才由David Clark和David Wilson（前者是计算机科学家，后者是会计师）提出[187]。在这种模型中，一些数据项受到约束而只能用一组特定的转换程序来处理。

更加严格的来说，有专门的程序用于数据输入，即将非约束数据项（UDI）转换成约束

数据项（CDI）；完整性验证程序（IVP）用于检查 CDI 的正确性（比如簿记的收支平衡）；转换程序（TP）在银行业中被认为是保持平衡的交易方式。在通常的表述中，由它们来保持 CDI 的完整性，同时它们也将足够的信息写入只允许附加的 CDI（审计追踪记录）中来重新构建交易记录。访问控制采用三重控制的方法（主体、TP 和 CDI），这种方法非常结构化从而加强了共享的控制策略。Amoroso [15] 对此的表述如下：

1）系统具有一个用来验证任何 CDI 完整性的 IVP。

2）处理任何 CDI 的 TP 应用程序必须维持 CDI 的完整性。

3）CDI 只能由 TP 改变。

4）主体只能初始化特定 CDI 的 TP。

5）三重控制必须将主体的责任策略合理地分离。

6）UDI 上的特定 TP 能够产生 CDI 输出。

7）TP 的每个应用程序都必须使用于重建它的足够信息被写入特定的只允许添加的 CDI 中。

8）系统必须认证企图初始化 TP 的主体。

9）系统必须只允许特定的主体（如安全官员）能改变与授权相关的列表。

Clark-Wilson 模型也遭受了一些非议。

首先，与 Bell-LaPadula 模型不同，Clark-Wilson 模型涉及了状态的维持。与审计追踪记录不同，当需要掌握那些只得到部分许可的交易情况时（如那些只得到一个许可而实际上需要两个经理许可的交易），状态的维持对于双重控制通常是必需的。如果双重控制通过访问控制机制来实现，就意味着需要将那些部分被许可的交易保存在一个特定的日志文件中。这意味着一些用户状态确实是安全的，从而更难以定义可信计算库。但是如果双重控制通过密码的方式实现，比如让经理对他们许可的交易附上数字签名，那么在处理所有被部分许可的交易时会出现问题因为它们需及时得到第二个许可人员的签名。

第二，模型并不是万能的。它采用了一个概念，即状态迁移必须保持不变，例如平衡性，而不是状态迁移的正确性。交易时一些错误仍然是允许发生的，比如存入时弄错了账号等。

第三，Clark-Wilson 模型回避了最棘手的问题，即如何防备不诚实的职员。第 5 条规则中提到必须将责任策略合理地分开，但并没有解释具体的含义。

9.2.2　责任的分离

关于责任策略的分离有两种基本的类型：双重控制和功能分离。

在双重控制中，两个或多个职员必须一起对一个交易授权。军事上关于双重控制的经典例子是核武器的指挥系统，它要求两个或多个操作人员同时按下各自控制台的某个键，而控制台之间相距很远使得这个操作不可能由一个人来完成（本书将在第 11 章“核武器的指挥与控制”中对此进行更深入的讨论）。民用上的经典例子是银行何时发布承担损失的担保信，以决定别的银行提供的贷款是否有效。如果一个经理可以单独对此做出决定，那么他的同谋者就可以从别的银行获得这笔担保金，并且系统几个月内也不会觉察到。这种情况将在 9.3.2 节中深入讨论。

在功能分离的模式下，两个或多个职员分别处理一项交易的不同环节。典型的例子是公

司的购货。首先经理做出一项购买决定并通知购买部门，购买部门写一份购货订单，库存部门负责记录到货情况，发票被送至财务部，由财务部核对发票、购买订单和库存收据，然后开一张核对表，由财务部经理在核对表上签字。

但购买并没有到此结束，财务部经理要对每月的内部账目清算负责。她的上级总结账目情况，以确保该部门取得预期的利润。内部审计部门可以在任何时候突击检查财务部的账簿。每年一次的外部审计将随机抽样检查某个部门的账簿。最后，如果发现有欺骗行为，公司的律师将尽最大努力挽回损失。

这个模型可以描述为预防—检测—挽回。这三个部分的可信赖度由应用程序决定。由于欺骗行为可能在数月或数年以后才被发现，损失可能很难挽回，由于可能有假的银行担保说明，因此尽可能采取预防措施是比较谨慎的做法，比如采用双重控制等技术。如果预防措施很难得到加强，那么检测必须足够快，挽回措施也必须足够有效，这样可以起到一种防范的效果。一个经典的例子是：由于银行出纳员可以很容易拿到现金，所以每天都必须在停业后清点账目以发现其中的错误。

簿记和管理控制系统不仅是最早的安全系统，同时也促进了许多管理科学的发展和民法的建立。它们是和一个公司的商业行为交织在一起的，并且存在于它的文化背景中。在瑞士银行，两个经理的签名会同时出现在几乎所有的东西上，但是美国则要相对宽松一些。在大多数国家的银行里，职员会受到背景情况调查，被随机地从一个任务指定到另一个任务，以及要求每年至少休假一次。但是这对于很少有欺骗机会的大学院系来说是大可不必的。

设计一个好的簿记系统是很难的，因为它是一个跨学科的综合问题。财务管理员、人事部门、律师、审计人员和系统中的其他人员都从不同的角度考虑这个问题，并且从各自的角度提供部分解决方案，但对彼此间的控制目标却不了解，因此很难真正解决问题。人的因素经常被忽略，因此系统常常处于潜在的危险中，因为相关下属人员或授权的经理能导致双重控制失效。重要的是不仅使得控制能适应文化氛围，而且要鼓励人们来采用它们。比如，在运营良好的银行，职员要掌握管理控制并作为防止勒索和绑架的一种方法。

系统安全研究者过于关注这个问题的一小部分，即创建双重控制系统的部分（一般来说，就是由多于两个的当事人共同控制）。尽管这也并不容易。比如，Clark-Wilson 模型的第 9 条规则提到安全官员能够改变访问权限，因此如何保证安全官员不会为两个经理设立登录账户并通过它们将银行的钱汇往瑞士呢？

一个可能的解决方案就是使用密码技术，并且分离两个或更多当事人之间相关的签名密钥。在一个 NT 网络中，显而易见的管理方式是将用户放在分离的管理域中。在传统的采用主机操作系统 MVS 的银行系统中，可以将系统管理员和审计员的职权分离开；前者可以做任何事，除了查明后者如何监视他的行为 [95]。但是在现实中，双重控制很难自始至终都有效，因为存在许多系统界面提供单点失败；而且，责任分离系统的管理是一种单调乏味的方式。

而实际的结果是，大多数银行的系统管理员能够实现这种类型的欺骗行为。一些欺诈行为的确已经被尝试过，但都失败了，因为后台平衡控制在一两天内就发出了警报，并且洗黑钱控制措施防止了他们逃脱。这将在 9.3.2 节中深入讨论。这里需要指出的是，预防—检测—挽回模型中的串行控制通常比共同控制更加重要。它们最终依赖于系统的某种持久状态，并且与程序员通过将交易原子化从而使得事情简单化的愿望相一致。

还存在宁静（tranquility）的问题。比如，一个会计师在知道明天他将被提升为经理的情况下，会不会在最后时刻对一笔大数目的转账做出两种授权？从技术的角度解决这个问题将涉及长城机制，这个机制支持一个基本原则即认为“X 可以做 Y 事而 Z 不能”（一个经理可以确认一项支付，仅当他的名字不作为交易的创建者出现时）。这样，我们可以得到许多涉及个人、团队和客体标识的例外规则；一旦规则增加（银行的实际情况就是如此），就需要一套系统的方法来检验规则并确保不会出现任何可钻的空子。

从方法学的角度说，银行安全策略就像医疗安全策略一样，可能会最终采用基于角色的访问控制；诸如 Windows 2000 这样的平台可能会在这个领域发挥作用。它能对责任分离进行有效管理，无论是像双重控制这样的并行控制还是像功能分离这样的串行控制。

关于双重控制的最后一点，它对于涉及两个以上公司的交易不是很有效，原因来自解决争端的困难，比如，一个公司的两个经理说钱已经被送出，而另一个公司却说没有。

9.2.3　哪里出了问题

偷窃有很多种形式，从纯粹的机会主义到聪明的内部欺骗行为；如果不考虑偷窃数目的大小，大多数公司发生的偷窃都来自公司内部。对此有许多调查。据最近的一次由 Ernst 和 Young 做出的调查报道，在 1999～2000 年最恶劣的欺骗行为中有 82% 是由内部雇员做出的；几乎一半的欺诈者已经在自身岗位上工作达 5 年之久，其中三分之一是经理 [697]。

典型的电脑犯罪案例有以下几种：

- 银行有一种临时账户系统，用于在交易的一方还没有确认的情况（比如一个账户被错误地输入到一个资金转账过程中）。这种系统是对双重控制系统的一种补充，用于处理陷入困境或不能及时得到平衡的交易。由于它存在被攻击的危险，银行制定了一条规则，即如果在三天内暂时账户的情况仍然不明确，那么将对此进行调查。某个女职员可以利用这个系统，从临时账户借出一定的金额，同时将等额的钱汇入她男友的账户。三天之后，她可以借出另一笔钱来偿还第一笔钱。这样，两年之内，她就可以侵吞掉一大笔金额（银行忽略了一条要求，即所有的职员从上次假期起的 15 个月内都应有至少连续 10 天的假期）。最后，她因为再也无法隐瞒激增的伪造交易额而被捕。
- 内伦敦教育局的一个职员想去澳大利亚探亲，为了得到一笔钱，她创立了一所假想的学校，并把职员的薪水全部汇入自己的账户。这一欺骗行为是由于偶然有人注意到汇报给教育局的有关学校的数目记录不同而败露的。
- 英国黑斯廷斯市的一个银行职员注意到分行电脑系统并不审查地址的变化，于是他首先挑选出一个拥有大笔存款并且每年只结算一次的客户账户，然后把这个客户的地址修改成自己的，并为该账户创建一个新的 ATM 卡和 PIN，然后再将地址改回去。这样，他就从这个客户的账户上盗走了 8 600 英镑。当这个客户来投诉时，并没有人相信她：银行声称电脑系统不可能出错，因此存款被盗一定是她自己的失误所致。这一事件直到后来才弄清楚，是因为那位职员良心发现并在晚上将现金封入棕色的信封放入支行的信箱。这之后支行的经理才意识到事态的严重性。

所有金额巨大（涉及数亿现金）的欺骗都是由于内部松懈的控制所致。巴林银行的破产就是一个很好的例子。该银行的经理们没有看清伪善的经济师 Nick Leeson 的真实面目，而

被他表面上的贸易利润带给他们的利益所诱惑。类似的情况也发生在其他大的金融部门，比如关于 Equity Funding 的丑闻：一个保险公司在他们的电脑系统中伪造了数千个投保人，并且将保险出售给再投保人；其他部门的欺诈行为还有很多，比如英国的 Robert Maxwell 侵吞《每日镜报》报社的养老基金（要了解历史上关于电脑犯罪的案例，可参考 Parker 的文章[602]）。欺诈行为的发生或者是由于受害公司的高层管理者疏忽大意（比如 Barings 的案例），或者是罪犯的肆意作为（比如 Equity Funding 和 Maxwell 的案例）。于是，会计师、股票市场和银行调整人员共同建立了许多关于如何设计簿记和内部控制系统的标准。比如在美国，有个叫赞助者组织委员会（Committee of Sponsoring Organizations，COSO）的机构，是由一些美国的会计和审计团体组成的[196]。在本书 22.4.1.2 小节中将继续讨论 COSO 和解释如何设计内部控制系统。

不断变化的技术将对控制产生影响，因此需要经常关注和维护这些控制系统。比如，由于采用了新系统对银行支票进行快速处理，加州的银行将不会存在存款人要求支票有两个签名的问题。即使是支票上印有“需要两个签名”的字样，银行也可以许可只有一个签名[651]。这看起来是一个关于客户安全而不是关于银行的问题，但是银行支票同样要承担风险，当某个环节（即使是商业交易）出问题时，银行也可能被提起诉讼。共同控制在技术上的易受攻击性有增无减。大多数主要的账目包在内部并不采用复式簿记，而是在其表示层产生复式簿记的一个表象；现在的趋势是朝着事件数据库的方向发展，在一个会计期内所有的交易都被汇总并按照要求生成结果报告。可能需要新的控制策略。一个可能的方法是保留所有原始事件的日志（购买订单、发票、付款单等）并且让程序经常反复核对。人工的方法也是非常有效的。账目软件应授权生产线管理人员使他们能够监督各自部门的收入、花费和承担的任务情况。将技术控制和管理控制两者重叠，使它们彼此取长补短。很不幸，通常的结果却是技术控制仅仅作为管理控制的一个复制品，因此才产生有缺陷的模式并被欺诈者所利用。

应该记住的一些教训如下：

- 交易安全级别的高低并不总是显而易见的。
- 在变化的环境中维护处于运行状态的安全系统是比较困难的。
- 如果是依靠客户的投诉来防止欺骗行为，那么就应该信任客户。
- 总会有那种获得信任的职员，在欺诈后能暂时逃脱惩罚。
- 没有绝对可靠的安全策略；现实生活中总是存在这样或那样的漏洞，因此有可能带来潜在的危险。
- 对一个反常事件，很难立即做出判断是由于欺骗还是错误导致的，因此要尽可能降低交易出错率。

还有一些并不能预料的危险情况，而寻找应对措施通常是最难也是最容易被忽略的工作，因为这不仅需要技术上的支持，比如知识渊博的工业专家、审计人员和从事具体设计工作的和有过类似经历的保险人员，还需要培训管理人员、审计人员和其他检测问题并处理它们的专门人员。这些内容将在第 22 章中继续讨论。

银行业已经经历了一个不断发展完善的过程。英语世界的国家关于银行业的一个普遍规律就是，每年都约有 1% 的职员被解雇。挪用行为是最常见的，通常造成的损失达到数千美元。目前还没有一种有效的方法来预测哪个职员将会犯错误，那些先前忠诚的职员可能会因

受到诸如离婚的打击而自甘堕落，从而养成赌博和酗酒的坏习惯。

9.3　大规模支付系统

电子资金转账系统是最先采用电报的系统之一，这大约是在 19 世纪中期。本书的 5.2.4 小节已经介绍了测试密钥系统的开发过程及其应用于手工计算消息认证码的情况。到了 20 世纪 70 年代，由于下列一些原因，银行家们意识到他们可能需要一种更好的系统：

- 系统密码易受攻击的问题日益突出。
- 尽管测试密钥表被谨慎地保管，但仍不排除银行雇员能够记住其中一些相对简单的规则。如果雇员能获得更复杂的规则，那么即使是在严密的监督下，他也有可能在公开测试一条授权消息的时候乘机测试一条非授权的消息。
- 这个方案并不支持双重控制。尽管测试是由一个职员完成而由另一个职员检查，但这反而增加了风险性（目前已有利用人工认证的方法来实现双重控制，这是在研究如何控制核武器的过程中发展起来的，但是这一技术在当时仍然是机密的。本书将在 11.4 节讨论这些问题）。
- 人们关注的主要是花费和效率。试想一下，让银行电报室的电脑打印一份交易文件，由人工来完成测试计算，然后发一份电报给另一家银行，再由这家银行对测试进行检查，最后将其输入电脑，这一连串过程十分繁琐，因此也是不现实的。这样就提出了一个问题：支付能从一家银行的电脑直接转入另一家吗？

显然，这需要一种全新的设计思路。

9.3.1　全世界银行间金融电信协会（SWIFT）

全世界银行间金融电信协会是在 20 世纪 70 年代由一个银行联盟建立的，目的是为其成员银行间进行的支付操作提供一种更安全高效的方法。它可以看成是一种集嵌入式密码技术、认证和认可服务为一体的电子邮件系统。

SWIFT 的设计限制规则很有趣。一些不诚实的雇员仍然可以勾结起来伪造交易情况，从这个意义上出发，银行并不希望信任 SWIFT。由于当时许多国家（如法国）都强制人们使用密码来实现保密性，因此认证机制必须与保密机制分开。当时并没有发明数字签名，因此许可功能不得不以非数字签名的方式实现。各银行不得不加大对银行间交易实行 Clark-Wilson 类型控制的力度（在当时，Clark-Wilson 模型同样也没有被提出，但是它的一些元素如双重控制、平衡、审计等都已经建立起来了）。

SWIFT 设计可以总结如图 9-2 所示。消息认证是由发送银行生成一条消息认证代码（MAC），然后接收银行检查这条代码来共同确保。最初 MAC 的密钥都是以终端对终端的形式管理的：

当一个银行发展海外业务时，负责谈判的高级经理将与谈判对手交换密钥，交换的方式或者是当面在会议上，或者是会后彼此发送给对方的私人地址。一般使用两个密钥来减小危及谈判安全的风险，由双方各自发送一个（基于这种假设：即使某方银行经理的信件被盗，另一方不会同时被盗），直到双方银行都确认安全收到对方的密钥并安装后整个密钥才会生效。

这种方式下，SWIFT 并没有参与信息的认证。只要 SWIFT 选择的认证算法合理，他们的

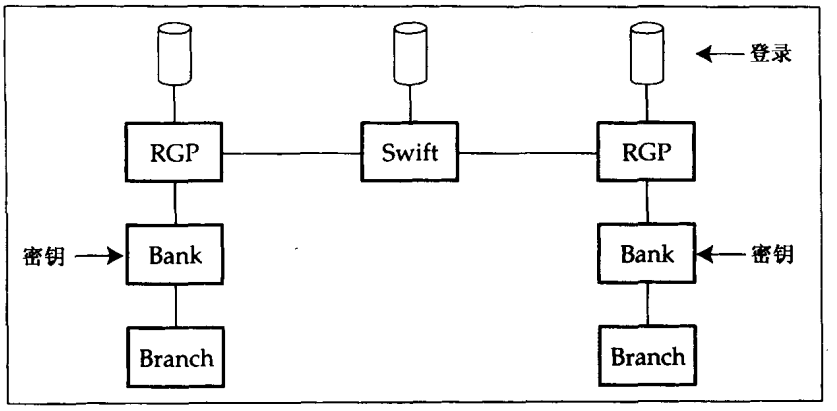


图 9-2　SWIFT 体系结构

职员就不可能伪造一项交易（所用的认证算法应该看成是一种商业机密，但是银行希望在国际上标准化他们的安全机制，因此这种算法可能就是 ISO 8731 中描述的算法 [657]）。通过这种方式，算法得以建立，虽然当时很难对它进行公开分析，但是后来还是做到了（对算法的破解是从 ISO 8731 消息认证算法中发现并公开的 [621]，但是要实际破解一个应用这种算法的系统却是不可能的，因为这需要的信息量非常大）。

尽管 SWIFT 本身与消息认证不沾边，但它的确提供了认可服务。每个国家的银行都把消息发送到一个区域统一处理器（Regional General Processor，RGP）上，RGP 负责记录这些消息并把它们转发给 SWIFT，SWIFT 也同时记录这些消息并通过各自国家的 RGP（这些 RGP 也记录收到的消息）转发给接收银行。通常 RGP 由不同的设施管理公司运营，因此如果银行（或不诚实的银行雇员）企图否认一项已经完成的交易或者声称一项实际上并不存在的交易，不仅要对付 SWIFT 本身，同时还要对付两个独立的本地承包商（为了修改他们的日志记录）。日志可以作为强有力的证据，并且比密码更容易被理解。

保密性取决于银行和 RGP 节点以及这些节点和 SWIFT 工作站之间的线路加密设备。密钥管理方式很直接，密钥被封装在 EEPROM 存储器中，并由专人携带在各设备间运送。在保密性并未合法化的国家，省略这些设备并不会破坏认证和认可机制。

双重控制由专用终端支持（在小的银行）或由主机软件包支持，这种软件包可以与银行主要业务系统集成在一起。操作方法通常是让三个独立的职员共同完成一项 SWIFT 交易：一个负责记录，一个负责检查，一个负责授权（因为检查员可以任意修改信息，这只算是双重控制，而不是三重控制。另外，负责界面维护的程序员同样可以攻击系统）。对交易情况和银行每天的清算情况的检查可以进行相应协调，这表明，如果有人将一条伪造信息输入系统，在 2～3 天内系统就会发出警报。

9.3.2　哪里出了问题

SWIFT 运作了 20 年，从没有关于从外部进行欺骗行为的报道。在 20 世纪 90 年代中期，由于增加了公共密钥机制，SWIFT 得到进一步加强。现在 MAC 密钥已经在采用公共密钥加密的相应银行间通用，MAC 本身也可以采用数字签名进一步获得保护。密钥管理机制已经作为 ISO 标准 11166，从而被应用到其他系统中（比如银行和股票经纪人用来注册和交易股

票份额的 CREST 系统)。关于这种体系结构的安全性已有很多讨论 [47, 657]：与采用公共密钥加密（其中中心认证机构可能错误地认证一个并不属于某银行的密钥）导致的信任集中化不同，(至少是) CREST 采用的公共密钥显得很短 (512 位)。至少有一个同样长度的 RSA 公共密钥已经被一个学生组织秘密破解。

然而，这种系统受到的实际攻击并不涉及支付系统本身。典型的攻击形式是银行程序员企图将一个伪造的消息插入正在处理的队列中。这种攻击通常都不会成功，因为程序员并不了解系统中的其他控制或大型转账的过程控制。比如，银行通常会设定一个比如 1 000 000 美元的透支下限，因此大笔金额的转账需要外汇经销商的预先允许。存在日常后台协调机制；有关洗黑钱的法律条款要求汇报大笔金额的取款；任何已开户并收到大笔电汇金额同时准备大笔转出金额的客户，都必须提供让人信服的理由。因此，将伪造消息插入系统的程序员往往在他取出现金的时候会被逮捕。

其他一些技术攻击手段也会对这些控制制造种种麻烦，比如将特洛伊软件插入银行经理用于启动交易的个人电脑里，从而窃听到从分行到银行主机的信息，破坏银行经理用于登录的认证协议，甚至将一项伪造的交易插入分行 LAN 并出现在相关的打印机上。

事实上，大多数“成功”的大规模银行欺诈行为并不是利用技术手段，而是利用了过程上的弱点，主要有如下一些：

- 担保信的例子很有代表性。一个公司为另一个国家的公司担保贷款是很普通的行为。这可以作为一种 SWIFT 消息或一封信件。但是如果当时没有现金变化，是不能进行平衡控制的。如果伪造的担保被信以为真，所谓的“受益人”就能够从相应的银行借出钱，销赃并迅速消失。只有等受害的银行发现贷款出了漏洞并追查担保时，才会发现担保是伪造的。
- 1986 年，在伦敦和约翰内斯堡之间发生了一个有趣而略微有些奇怪的欺诈行为。当时，南非政府实行两种汇率，而交易究竟采用哪一种汇率是由银行经理决定的。一个负责决定汇率的银行经理与伦敦的一位富商勾结，他们按照兰特（南非货币）对英镑 7:1 的汇率将钱汇到约翰内斯堡，然后又在第二天以 4:1 的汇率汇回来。如此进行了两个星期，他们的这种行为遭到了相关机构的怀疑，警察也介入此事件。结果这位经理仓皇出逃，穿过边境到达斯威士兰，再经奈洛比飞往伦敦。在伦敦，他对报界吹嘘他是如何欺骗罪恶的种族隔离制度的。因为英国没有兑换控制，兑换上的欺骗行为并不算违法，因此他就没有被驱逐。他的同谋者同样得到了数百万的金额而逃脱惩罚，银行甚至不能起诉他们。
- 1979 年，发生了一起资金转账欺骗行为，这也许是迄今为止最著名的一次。Stanley Rifkin 是当时的一个电脑顾问，他从太平洋国家安全银行盗用了一千万美元。他通过从瑞士的一家俄罗斯政府代理机构购买大量钻石避开了洗黑钱控制。他在监听电汇部门的转账时获得了内部使用的授权代码，并通过电话这种简单的方式用这个代码将转账输入系统（双重控制在系统界面处失效的一个经典事例)。他在美国银行休假日的前一天实施了上面的计划，因此留出了用于逃跑的时间。他的失误之处在于没有计划好如何处理得到的这些钻石，如果他把它们藏在欧洲，然后堂而皇之地回到美国协助调查欺骗行为，也许就能很好地逃脱惩罚。然而，他最终还是被捕了。

职业道德要求我们必须时刻警惕引起责任分离控制失常的因素，这些因素往往只是某个

小小的失误。尽管我们能够解决许多系统管理和界面等技术问题，但仍然面临许多控制上的商业系统分析问题，比较常见的就是重要的交易对于随意的检查来说其重要性是不明显的。

9.4 自动柜员机

认为双重控制尽管必要但并不完善的另一个理由是从对“幻影提款”的研究中得出的，即从自动柜员机（Automatic Teller Machine，ATM）进行未授权提款的事例。

ATM，又称为自动提款机，是20世纪影响最为深远的技术发明之一。与它们在社会和经济方面的影响不同，它们对于安全工程学的重要性是不仅提供了一种技术手段，同时也充当一种实例研究。

ATM是第一种大型零散交易处理系统。他们大约出现于1968年；现在世界上大约安装有500 000台这种机器。为ATM开发的技术同样可应用于商店销售点的电子资金转账（electronic funds transfer at the point of sale，EFTPOS或POS）终端。现代分组密码首先大规模应用在ATM网络中，用于生成和验证安全硬件设备的PIN，这些设备位于ATM机和银行计算机中心。这项技术，包括分组密码、防篡改硬件和提供支持的协议，最终也应用在许多别的领域，从邮资支付机到彩票终端。ATM成为“招人喜爱的应用程序”，并促进了现代商业密码学的建立。

9.4.1 ATM的基础

许多ATM机运行时使用的是IBM在20世纪70年代中期为其3614系列提款机所设计系统的各种变体。这里使用了一种称为PIN密钥的机密密钥来对账号进行加密，然后将其转化为十进制并截取之。经过这一操作得到的PIN称为自然PIN；还可以加入偏移量从而得到客户必须输入的PIN。偏移量并不具备真正加密的功能；它仅仅使客户能够选择他们自己的PIN。这个过程的例子如图9-3所示。

在这个系统中双重控制是通过防篡改硬件实现的。密码处理器，通常又称为安全模块，放置在银行的中央电脑室里。它将按照以下的方式执行许多对客户PIN和相关密钥的规定操作：

- 对客户PIN具体数值的操作，以及对密钥或产生和保护密钥所需材料的操作，都在防篡改硬件上完成，这一过程中的具体数值不会泄露给任何一个银行职员。
- 交易卡和PIN通过独立的渠道送到客户手中。交易卡可以应用压模和磁条打印等技术做到个性化；PIN信封在一个独立设备上打印，这种设备包含一台与安全模块相连的打印机。
- 每个ATM的终端主密钥分成两部分打印，并分别由两个不同的官员带到分行并输入ATM机，在ATM机内部这两部分再重新组合成密钥。在银行和网络交换机如VISA之间创建密钥时也常采用这种方法。

账号N	
(在磁条上):	8807012345691715
PIN密钥KP:	FEFEFEFEFEFEFEFE
DES结果 $\{N\}_{KP}$:	A2CE126C69AEC82D
$\{N\}_{KP}$十进制化:	0224126269042823
自然PIN:	0224
偏移量:	6565
用户PIN:	6789

图9-3　IBM生成银行卡PIN的方法

- 当 ATM 机执行 PIN 验证时，PIN 密钥在终端主密钥下被加密后送至 ATM。
- 如果 PIN 验证是通过网络统一进行的，那么 PIN 首先应在终端主密钥下进行加密，然后从 ATM 送到安全模块进行检验。
- 如果银行的 ATM 需要跟别的银行的 ATM 联网，那么在某地加密过的 PIN（比如按 ATM 密钥加密），应该先进行解密，然后按目的地的要求重新加密（比如使用 VISA 系统的密钥）。这种 PIN 翻译功能完全在硬件安全模块中完成，因此银行的程序员不可能知道它的具体数值。

到了 20 世纪 80 年代和 90 年代，硬件安全模块越来越复杂，功能也越来越多。比如 IBM 公司 2000 年的主导产品 IBM 4758，有一个好处就是从网上可以获得它的说明书（命令设置参见 [397]，体系结构和硬件设计参见 [718]）。本书将在第 14 章“物理防篡改”中详细讨论。

但是要像现代 ATM 网络那样，将双重控制的协议从一个银行扩展到世界范围内的成千上万家银行并非易事，原因如下：

- 20 世纪 80 年代中期，许多银行在构建 ATM 网络时并不采用硬件安全模块，而是软件加密技术。因此从理论上说，任何银行程序员都有获得银行客户 PIN 的可能性，解决办法就是颁布使用安全模块的标准。许多国家（如美国）并不重视这些标准，即使是在重视这些标准的地方，一些银行仍然继续采用软件来处理客户的交易。由于软件要使用一些密钥（如用于与 ATM 连接的密钥），知道了这些密钥就可以推测银行账户的 PIN。因此，硬件 TCB 提供的保护并不是万无一失的。
- 让 10 000 家银行都共用成对的密钥是不可行的，因此每家银行都与 VISA 或 Cirrus 等组织提供的交换机建立连接，这些交换机中的安全模块负责转换工作。交换机也能进行清算账目的工作，使银行能够清查每天与系统中其他银行进行的交易，这只需借助简单的电子借贷操作。交换机是非常可靠的，但一旦出现了问题，后果可能非常严重，而问题不仅仅是安全问题，同时还涉及不诚实的职员。出问题的交换机管理员最终将受到起诉，但补救可能要花费数百万。
- 当处理巨额资金交易时，应尽量采取一切办法减少花费。比如，通常会关闭授权响应的认证。结果使得有网络访问权限的人只需简单地主动重复授权响应，就能让 ATM 机接受任何卡。网络管理员声称欺骗行为只要发生，授权控制总是能够将其改回来，这听起来很合理，因为涉及授权响应的攻击非常少。这种捷径从风险和花费的角度看都合情合理，意味着银行面对客户的争论时坚持认为它的 ATM 网络不可能被攻击，因此失误完全在客户一方，然而这种说法歪曲了真相。进一步来说，要求消息认证代码立刻对欺骗做出反应是很困难的，一些银行不完全提供或根本不提供这种功能。因此除非使用更多的加密设备，否则这将影响到银行的业务质量。可以用一句谚语来形象地概括：“最优化就是一个寻找解决方案并且使得花费越少越好的过程”。

理论上还存在许多种攻击 ATM 网络的方法。比如，最常使用的单密钥 DES 加密算法，它甚至用于最高级别的密钥中，然而现在 DES 可以通过彻底的密钥搜索破解。有趣的是，这些系统存在了很长时间并且被从内部和外部攻击过多次，这就积累了很多数据来研究它们是如何失效的。

9.4.2 哪里出了问题

研究与 ATM 有关的欺骗行为是很有趣的，因为 ATM 系统已经成熟，并且有巨大的数据容量和各种各样的操作员。文献［19］是对此进行的一份详尽调研，文献［20］是更深入的研究材料。在此，将总结其中最重要和最吸引人的几点。

20 世纪 70 年代和 80 年代设计 ATM 安全系统的工程师们（本书作者就是其中之一）认为，罪犯们对系统设计都非常精通，同时也很会选择攻击手段。工程师们除了担心许多银行迟迟不购买安全模块和省略有关授权响应的认证代码而引起漏洞外，还受其他一些问题的困扰：加密算法是否有效？防篡改单元是否有效？维护工程师是否能通过使干扰敏感性的电路一次访问失效，而从另一次访问获得密钥？用于生成密钥的随机数产生器是否足够随机？然而他们最担心的还是强化双重控制的可行性。银行经理们往往认为亲自接触键盘等操作是有损尊严的事，因此他们在巡查之后并不是亲自把 ATM 管理密钥的两个部分输入 ATM 机，而是把这两部分密钥都告诉 ATM 工程师，由工程师代而为之。因此有理由相信迟早连 ATM 修理员都会知道银行的 PIN 密钥，并能大量伪造 ATM 卡，或关闭整个系统，从而使公众对电子银行系统丧失信心。

真实的幻影提款主要是由下面三个原因之一造成的：

- 首先是简单的处理错误是造成大量争执的原因。假定美国的客户每年通过 ATM 提款的数额大约是 50 亿，系统平均每处理 100 000 项交易将出现一次错误，这样每年因此引起的争端竟多达50 000起。实际上，错误出现的概率在 1/10 000 和 1/10 0000 之间。其中一种造成错误的原因是，如果网络连接在确认信息到达主机前中断，那么 ATM 机将重新发出一条交易信息，而主机经常会周期性地崩溃，并“忘记”交易情况。有时还会出现一些客户的账目信息记录的却是其他客户的交易情况，而另一些客户在用卡交易时并没有记录（这种卡习惯上称为“董事长”卡，这只是一种幽默的说法）。
- 由信件被盗引起的案例也很多。大约占英国所有支付卡损失的 30%，但是大多数银行的邮件控制情况还是令人担忧。比如，1992 年 2 月，我向银行询问增加交易卡的限额，结果银行却通过邮局寄来两张而不是一张卡和两个 PIN。这些卡到达的几天前，曾发生过一起我所在公寓邮箱被人打开信件被人拆开的事件。最后查出原因是银行并没有专门的按照注册地址发信的系统（我要求银行将卡邮到支行，但是支行却草率地重写下地址就将其邮出）。从那以后，许多银行发现邮件控制是减少发生欺骗行为的一种有效手段。
- 银行职员的欺骗行为是第三大主要因素。9.2.3 节中的 Hasings 的案例也提及到此类欺骗行为。还有很多这样的案例，比如在苏格兰的 Paisley，一个 ATM 修理工在 ATM 机中安装了一台便携式电脑用来记录客户的交易卡和 PIN 数据，然后用伪造的交易卡大量盗用客户的存款。在英国伦敦，一家银行居然愚蠢到在实际系统和测试系统之间采用相同的密钥，维护人员发现竟然可以用测试设备获得客户的 PIN，然后他们就把这种交易卡以每张 50 英镑的价格卖给当地的罪犯。类似的欺骗行为主要发生在英国等国家，因为这些国家的银行否认提款机会出错，而且银行职员知道在收到客户的投诉后通常的做法是隐瞒而不是去调查。

这些错误比起工程师们所担心的情况要简单和直接得多。事实上，惟一一种能预料到却依然发生的欺骗行为在银行网络不通或中央主机离线时仍然让 ATM 机运行期间（在 80 年代很普遍）发生。尽管这样做能够提供 24 小时不中断的服务，但是一旦罪犯（尤其在英国和意大利）学会了打开银行账号和伪造交易卡，他们就可以在深夜网络中断后从许多 ATM 机上同时取走大笔金额 [494]。这类欺骗行为使得 20 世纪 90 年代中期银行一度只允许在线联机式的 ATM 提款方式。

然而，很多欺骗行为都是以不可预料的方式发生的。本书中曾提到过 Utrecht 案例，安装在汽车修理厂销售点终端的窃听器居然被用来获取交易卡和 PIN 数据，然后通过“加密替换”的手段就可以把欺诈者交易卡的账号换成别人的。这样的案例还有很多。

- 罪犯最惯用的方式就是排在 ATM 机队列中，偷偷观察客户的 PIN，然后从废弃的 ATM 票据上获得账号并复制到空白的卡上，就可以使用它们来盗取客户的存款。这种手段首先在 20 世纪 80 年代中期的纽约有过报道，在 20 世纪 90 年代中期的圣弗朗西斯科的 Bay 地区仍然流行。银行也采取过一些对策，比如在条形码中加入额外的数据，或不在票据上打印全部的账号。
- 有些银行系统会出现这样的问题：当一张电话卡被插入 ATM 机时，ATM 机会认为这是先前插入的卡。欺诈者站在队列中，偷看客户的 PIN 并利用这种漏洞盗取钱财。这种错误可能是一种不易发现的涉及读卡机错误句柄的程序错误。类似的错误在调试的时侯也不可能被全部发现。
- 当一个 14 位数的特定序列被输入 ATM 机时，ATM 机将从现金柜中输出 10 张钞票。某个银行将这个特定序列打印在发给支行的操作手册上，三年后该银行遭受了重大的损失。而且该银行必须在所有电脑上都采用软件补丁来终止这项交易，否则损失将不会停止。
- 一个小金融机构对所有客户采用了相同的 PIN，这也是简单程序错误造成的。
- 有的银行考虑过采用数字检查方案使得 PIN 能够被并不具备完全加密能力的离线 ATM 机和销售点设备检查。比如，一个英国银行的客户得到一张信用卡的 PIN 和一张借贷卡的 PIN。前一个 PIN 的 1、4 位之和等于 2、3 位之和，后一个 PIN 的 1、3 位之和等于 2、4 位之和。这意味着一个欺骗者将能在离线设备上使用偷来的交易卡，只需输入一个 PIN 如 4455 就可以了。
- 有的银行在操作上非常不谨慎。1993 年 8 月，我的妻子带着证明人去一家支行，告诉银行她忘记了 PIN。于是出纳员很快就通过与 PC 机相连的打印机打印出一个新的 PIN 信封。这个操作并没有采用双重控制，更糟的是那个支行并不是保存我们账户的支行。没有证据能确认这个人就是我的妻子而不是冒充的，她能提供的惟一身份证明就是我们的银行卡和她的支票簿。在这种松散的过程控制下，任何一个从街上走进来的人都有可能得到其他客户的 PIN，这比密码破解技术简单得多（这类有问题的银行最终不是破产就是被接管）。
- 利用假的终端收集客户的交易卡和 PIN 数据是逐渐被采用的一种欺骗方式。这种方式 1988 年在美国被最先报道。当时，一些欺骗者制造了一种可以接受任何卡和 PIN 的自动售货机，并在里面放上一些香烟，然后把这种自动售货机放在商业街上，他们通过调制解调器获取顾客 PIN 和磁条码的数据。1993 年，两个欺骗者竟想出在康

特涅格的 Buckland Hills 商业街上摆放一台伪造的 ATM 机 [421, 590]。他们先设法获得了一台 ATM 机和它的软件开发工具（所有东西都是贷款买来），然而当他们在纽约使用伪造的 ATM 卡诈骗时终于落入法网，因为纽约的取款机里藏有微型的摄像头。最近一起欺骗事件发生在 1999 年的加拿大，这也是最大的一起。这个案例中采用了技术先进的销售点终端，最终在多伦多及其他地区逮捕了多名东欧犯罪组织成员 [54, 91]。

综上所述，从早期一直到 20 世纪 80 年代中期人们在设计 ATM 安全系统时主要犯的错误是担心罪犯太聪明，而事实上更应该担心的是我们的客户、银行系统设计者、执行者和测试者容易犯下愚蠢的错误。

密码通常只是大型系统的一部分，它之所以受到很多关注是因为它具有数学的趣味性。很少会有人关注那些“乏味”的部分，比如培训、可用性、标准和审计等，罪犯一般不会选择破解密码这种方式入侵系统。同样值得注意的是大型系统拥有众多用户（比如 ATM 网络），因此我们必须准备应对可能出现的各种意外情况和潜在的易受攻击性，这些在测试阶段都是很难发现的。

9.4.3 实际应用

在一些国家（包括美国），银行必须承担新技术带来的风险。有这样的先例，如果一个银行客户声称她没有提款，那么这句话比一个银行专家说这个客户已经提款更加具有分量 [427]，根据这个合法的先例，美国联邦储备系统通过了 E 法令，要求银行对所有有争议的交易进行赔偿，除非银行能够证明确实是客户有欺骗行为 [276]。尽管也出现了一些滥用此法令的情况（客户对此法令的错误理解曾使美国银行一年约花费 15 000 美元），但还是值得的（尤其考虑到破坏者所导致的损失是这项的三倍之多）[813]。

在其他一些国家（比如英国和挪威），银行声称 ATM 系统不会出错，从而逃避责任。他们声称“幻影取款”现象根本不存在，因此来投诉的客户一定是自己弄错了或者在撒谎。然而，越来越多的罪犯因 ATM 欺骗行为而被捕入狱，这一事实不能再被轻易否认了，于是先前的看法逐渐消失了（至少在英国是这样，关于这方面的案例可参考 [19, 20]）。然而仍然有一些其他的不幸事件发生，进而使银行的名声变坏。这些事件中影响最坏的就要算 Munden 案例了。

John Munden 是一个地方巡警，驻扎在剑桥郡的 Bottisham。他的管辖范围包括 Lode 村庄（当时我居住于此地）。1992 年 9 月他假期回家时发现他的银行账户分文不剩，便要求银行作账户结算，于是发现有 6 次来历不明的提款，总金额为 460 英镑（约 700 美元）。他向银行提出投诉，但银行对此作出的反应是起诉他企图通过诡计骗取银行的钱。虽然在案件审讯期间，银行系统的运营和管理岌岌可危，有争议的交易没有得到合理的调查，银行仍然固执地宣称它的 ATM 系统不可能出现漏洞，因为软件是用汇编程序写的。最终由于这位巡警的言词冒犯了银行，他在 1994 年 2 月被判有罪并被警察局解雇。

这桩冤案最终经过上诉而以一种有趣的方式得到平反。就当上诉听证的期限快到时，起诉方提供了一份冗长的来自银行审计师的报告，声称银行系统是安全的。被告方要求以银行专家平等的访问权限进入银行系统，银行方面拒绝了这一要求，因此法庭否定了银行所有的相关计算机证据——包括银行结算。上诉成功了，Munden 也恢复了原职。但这已经是 1996

年7月了——他在监狱中呆了四年，他的家庭也为此承受了巨大的压力。假如这个事件发生在加州，他将有可能获得巨额赔偿金，这是银行家们应该考虑的，因为银行系统正在走向全球化，银行客户也遍布全球。

从上面这个案例中得到的教训就是仅有双重控制是不够的。一个系统要想提供证据，它就必须能够经受住来自诉讼对方的专家的测试。事实上，Munden案例中的银行使用了错误的安全策略。它真正需要的不是双重控制而是认可，即能够为交易的当事人提供先前发生的事件重现的能力。这可以通过安装ATM摄像头来实现。尽管摄像头在美国的一些州已经使用，但在英国还没有使用。

关于认可的问题也出现在其他许多应用中。通常，应当关注的问题不是机制（摄像头、生物测定学和数字签名）而是动机。既然ATM摄像头会削弱关于银行绝对可靠性的说法，为什么英国银行应该安装它（在幻影提款吵得沸沸扬扬的时侯，一家英国银行的确安装了ATM摄像头，但是后来迫于其他银行的压力又不得不拆除）。数字签名只是当真的出现错误时，使得一项交易更难于否认，那为什么人们在网上购物时仍然要使用它？在后续的章节中我们会反复遇到这样的问题。

9.5 小结

银行系统在许多方面都是很有趣的。

簿记应用给我们提供了一个完整的系统实例，它的安全性主要是确实性和可计账性而不是保密性。簿记应用的保护目的是防止和发现内部成员的欺骗行为。Clark-Wilson安全策略提供了一个关于簿记应用运作的模型，可以总结如下：

> 所有交易都应该保持系统的不变性，也就是账簿必须实现收支平衡（总账中支出必须与收入相平衡）。某些交易要求由两个或两个以上职员共同执行。交易记录在提交之前不允许被破坏。

这是基于长期的簿记实践过程建立的，它让研究机构去考虑系统而不是各种Bell-LaPadula模型的变种。

但是手工簿记系统不仅仅只采用双重控制。尽管一些系统的确需要交易横向地由两个或更多职员授权，但是责任分离更多地是以纵向的模式进行，因为系统中不同的人只负责交易的不同部分。设计能有效实现这一目的的簿记系统成为研究中的主要问题，然而这又是被经常忽略的问题，同时也涉及规则引进的问题。另一个常见需求就是许可——当事人应该能够创建、保留和利用其他当事人的相关行为的证据。

远程支付是另一项主要的银行业务，这对于电子商务越来越重要。事实上，采用电汇可以追溯到维多利亚时代中期。由于攻击系统的动机很明显，因盗取大笔金额而被发现的罪犯通常会被起诉，支付系统对于研究可能会出错的环节很有价值。他们遭受损失的历史告诉我们减少基本错误的重要性，从而防止受到程序上的攻击而避免技术控制失效（比如窃取邮件中的ATM卡），以及采取有效的控制对内部的欺骗行为起到威慑作用并及时发现它们。

支付系统对于密码学的发展和应用同样起到非常重要的作用。这里有一个概念上的创新，即密码学可以用于将应用程序的关键部分限制到由防篡改处理器构成的可信计算库上，这个方法自从提出以来就在很多领域得到应用。

研究问题

给簿记应用程序设计交易集合仍然是科学上超前的问题；我们可以利用工具以更系统化和更少出错的方式来完成一项交易。会计师、律师、金融市场调节员和系统工程师都会觉得这似乎是别人的责任，其实这是使多学科研究更有效的一个非常好的机会。

从更加基础的层次，我们甚至没有完全理解访问控制系统，比如 Clark-Wilson 模型和长城模型。在责任分离模型中，哪一个走得更远？如何设计一个双重控制系统？我们能从代码中向中间件抽象出多少关于授权的逻辑概念？我们能把策略和实现手段分开而使企业范围内的策略更容易管理吗？

还有一些有用的区别，如策略、机制和管理的差别，推和拉的区别，以及规范控制和运行时控制的区别。还有一些用于强化专断策略的引擎原型，如 HP 对服务器产品的授权认证[772] 和 AT&T 的 Policymaker [115]。开发这些引擎来解决可能安全协议的完全通用性仍然是一个有待研究的问题。

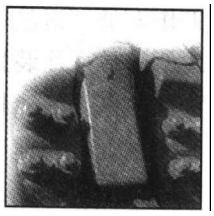
对于密码系统的鲁棒性，安全机制的可用性和担保，这些是很大的研究主题，目前其需求也只是部分描绘出来。鲁棒性和担保只是部分地理解，然而可用性却仍然处于未知状态。在安全性研究这个领域中活跃着更多的数学家而不是应用心理学家。

参考资料

可能还没有一本关于银行计算机系统的综合性专著，尽管有许多可以从国际清算银行（BIS）获得的关于专门的支付系统的研究论文 [72]。当提及开发具有鲁棒性的管理控制和减少损失数量的商业过程时，实体材料显得尤其缺乏（尤其在新的电子商务需要这样的系统的情况下）。1997 年有一次这方面的学术会议 [416]；但是所有尝试这个问题的商业书籍几乎都把注意力放在金融管理和管理控制的软件方面，比如 “tone at the top”，本书将在第 22 章中讨论这个问题。

对于金融交易处理系统的细节问题，引用的文献中，[19，20] 对此作了基本的介绍。[221] 是更加综合性的，[336] 中描述了 20 世纪 80 年代中期的 CIRRUS 网络。最具参考价值的公共域名资源可能是大型的关于当前设备的联机手册，比如 IBM 4758 和 CCA [397]。

第10章 监控系统



如果一个人值班的时间太长，那他很有可能会睡着。

——弗朗西斯·培根

10.1 引言

绝大多数安全系统都涉及对环境的监控。最显著的例子就是防盗报警器。还有些仪表用于测量燃气或电的消耗量。最大规模的此类系统是用来确保核不扩散条约的实施。国际原子能权力机构（IAEA）规定，一个国家的核设施中要安装大量传感器（如地震仪、闭路电视等等），对裂变的物质的运动做即时的、永久的远程记录。还有些车载监测系统，例如导弹遥测仪、出租车计程器、转速表（在欧洲，这类设备用来记录卡车和公共汽车司机的行驶速度和工作时间）。

这些系统有许多共同的有趣特征。比如说，要破坏防盗报警器，有效的办法就是使它停止工作，或者说在更多的情况下，让使用者认为它已经变得不可靠了。这就产生了一种拒绝服务攻击方法，如今这种进攻越来越严重了，而且一般来说很难对付。

正如我们知道的，军队通信系统的设计特点是保证其保密性。簿记系统的特点是保证其记录的确实性。而监控应用系统就是可靠有效性系统的典型例子。如果在我的银行地下室中有个窃贼，我不在乎是谁发现了他，也不在乎他是谁，也就是说系统的保密性和确实性不是主要的考虑因素；最值得关心的是所设计的系统是否可以阻碍盗贼行窃。

银行地下室的报警系统应该能有效地阻止攻击（至少是外人的破坏），这就提出了一个最简单的案例研究。尽管传感器失灵让人烦恼，但是我们更关注攻击通信设施的问题。然而，许多监控系统本身是完全暴露在外的。需给电表通常是消费者的常用仪表，但是，有些消费者却想方设法使其显示的读数不正确。如同计程器一样：出租车司机（或是车主）可能想让计程器显示的公里数或时间比实际要多，而转速表正好相反。卡车司机常常希望超速行驶或者疲劳驾驶很长时间。所以就出现了两种破坏方式。司机要不是使转速表失效，要不就使仪表错误地显示行驶时间和距离数据。这些装置同样因为完全暴露在外而易损坏。在仪表测量和监控系统中（尤其是原子能检测系统），也需要关注这种现象。攻击者不仅仅能从通信的操纵中（比如重新发布旧消息），而且还可以通过伪称其他人也这样干过来谋取利益。

监控系统也很重要，它们与增强软件以及其他数字媒体的版权系统之间有相当多的共同之处，我们将在下一章讨论。这类系统也引入了“拒绝服务攻击”这个能左右电子战方向的广泛问题进行讨论，并使得人们开始着重关注电子商务。

10.2 报警器

报警器的作用远不止用于防盗报警。它们的使用范围从对超市冷藏柜温度的监控（防止

工作人员“偶然地”关掉冷藏柜从而把即将坏掉的食物带回家）到用来阻吓拆弹队的恶作剧式的简易爆炸装置。但是本书认为以防盗报警器和保护计算机机房的报警系统为例来讨论报警器的作用是比较方便的。

报警器的标准和要求总是随着国家和风险类型的不同而变化着。一般采用本地的专业公司完成这项工作。但是作为一个安全工程师，有了解这些内容的必要。在我的职业经历中，报警器经常会对系统的设计产生很大的影响。这一影响已从原先 ATM 的一部分扩展到像宝石批发那样具有大风险的通信系统安全性评估方面，例如保护银行计算机机房持续工作的安全系统。

给有电子工程/计算机科学背景知识的人讲解物理安全的基础知识，比教授其他人更容易些。因此，物理的保护装置和逻辑的保护装置间如何协调将依赖于实现系统的人。你有可能被询问对客户系统安装的意见——系统通常被当地的、可能与你的客户已建立联系的承包商设计并且安装了，而他们对上述所关心的系统安全问题却知之甚少。

10.2.1 威胁模式

设计时所要考虑的一个重要问题就是攻击者拥有的技能水平、设备和决断能力。例如电影《Entrapment》也许是好的娱乐片，但它无法展现偷窃者内心世界的真实想法。目前还没有一个“盗窃的国际标准”，我所了解的最新的分类，是由一个美国军队专家划分的 [74]。

- 德里克是一个 19 岁的瘾君子。他利用一些风险很低的机会去偷东西，例如偷一个录像机来暂时摆脱困境。
- 查理不到 40 岁就因偷盗七次被判入狱。他生命中最后的 25 年里有 17 年是在监狱中度过的。虽然他不很聪明，但狡猾并且经验丰富；在监狱里的那段时间，他获得了许多“知识”。他从小商店和看起来豪华的郊外房子偷起，偷走一切认为可以出售给当地黑市的东西。
- 布鲁诺是一个“有身份的罪犯”，主要偷艺术品。他经营一家小艺术画廊来掩护自己的盗窃行为。他伪造过大学学位证书，18 年前因抢劫而被判入狱。2 年的监狱生活后，他改了名字并且搬到了另一个地方。他偶尔为了解他的过去的情报员做“线人”工作。他想从事计算机犯罪，但是到目前为止他做得最大的一件坏事就是，在 20 世纪 90 年代中期，一个存储器极度短缺的时代，他从一所大学的许多微机中拆除了价值$100 000 的内存芯片。
- 阿布杜拉曼领导着一群受过军事训练的好战分子。他们拥有步兵武器和炸药，由一个声名狼藉的国家给他们提供高级技术支持。在这个国家的军事学院中阿布杜拉曼所在班一共 280 人，他成绩名列第三，但是并未得到提升，原因就是他来自于不同的民族。他认为自己是个好人而非坏人，他的任务就是去盗窃钚元素。

因此德里克是个技术不熟练的盗窃犯；查理有些技能；布鲁诺技术熟练，并且可能有像清洁工人这样非技术的内部人员帮忙；然而阿布杜拉曼就不仅仅是技能高超了，他还拥有丰富的财力资源。甚至可能有技术员或其他被教唆的高技术内部人员的帮助。

社会学家对德里克感兴趣，刑事学家对查理有兴趣，军事学家偏爱阿布杜拉曼，而我们则更关注布鲁诺；他并不是接受采访中最高级别的“平民罪犯”（高级别罪犯应该是类似于为贩毒集团洗黑钱的银行家或者律师们，这些下面会谈到）。但是在没有恐怖主义问题的国

家中，计算机机房实际防御系统要针对像布鲁诺这样的人进行设计（不管是合理还是夸张，实际防御系统的设计总是要满足客户的要求）。

对布鲁诺的一般看法是，当他对当地市政大厅房屋结构经过几天的勘查后，就巧妙地破坏了屋内的警报系统。可能有关这个犯罪首领的报道已经不止一次的在报纸上刊登过。

如何偷一幅画（1）

毕加索的一幅画在一个据说安装了最新警报系统的画廊里被偷了。小偷移开屋顶瓦片，用一条绳子把自己放下去，以便不触动地毯下面的压力垫。他取下画，并按原路爬出去，没有触到地板，而且很可能已经以25万美元的价格把画卖给一个富有的毒贩。

报纸喜欢报道这类题材的新闻，而且它确实时常发生。现实总是既简单又陌生。

10.2.2 为什么不能保护一幅画

设计报警系统的一个常见错误就是被最新的传感技术所迷惑。市场上有众多令人惊奇的材料，比如光导纤维做的线缆，可以把它绕在被保护的物品上，当线缆拉紧或放松不到千分之一毫米时就会产生感应。自然艺术画廊的老板可以买几英尺这种神奇的线缆，把它固定在毕加索画的背面，并和报警器连接起来。

如何偷一幅画（2）

布鲁诺是这样行动的。他首先假装成一名旅游者来参观画展，然后躲藏在清洁柜里面。在清晨的某一刻，他爬出来拿到画直奔安全出口逃去。保安人员按响了报警器，但又能怎样？不到一分钟的时间，布鲁诺已骑上他的摩托。等到12分钟后警察赶到时，他早已逃之夭夭了。

这类偷窃发生的可能性比水手长的椅子穿过屋顶的可能性还要大。这种偷窃方法容易得手的原因是报警器很少能和建筑物的入口控制系统良好地结合起来。许多设计者无法明白无论在哪里他们都不能清楚地确定所有在白天进入特定区域的人。小心谨慎的做法是对“藏在暗处”的坏人采取预防措施，即使只在画廊关门后做一个巡回检查也行。严格的物理安全意味着对人群的严格控制。实际上，最早记录的有关RSA密码系统的应用是在1978年，并不是用于加密通信，而是在工作人员使用的凭据上提供数字化的信号，以便他们能顺利通过爱达荷州瀑布处的钚反应堆进口处的门禁。这些凭据上包含诸如体重和手形［701，705］之类的资料数据。但是仍旧使人感到惊奇的是，无论是用温和的技术手段，例如坐在别人的肩膀上进入，还是通过看门人的帮助，我曾到过的最安全的地方的门禁系统竟然都轻易地被瓦解了。

而且，人们从未深入考虑过警报的响应过程。对大多数系统来说，这一点使破坏变得更简单了。

所以，我们不能孤立地考虑警报装置。一个有效的物理安全系统应包括一系列要素：

制止——发现——警报——延迟——响应

应用不同，系统强调的重点也不同。如果我们的对手是德里克或者查理，主要关心的问

题是如何制止他们行窃。对于阿布杜拉曼这类目标，主要问题就是如何牵制他，拖延足够长的时间，直到舰队赶到。而布鲁诺是最有趣的案例，因为我们没有足够的军事预算来一直监视他，而且相对于阿布杜拉曼，很多建筑物的警卫更担心布鲁诺。基于这些情况，他们要考虑的问题是如何发现盗贼以及如何做出响应。

10.2.3 传感器失灵

防盗警报器使用很多种类的传感器，包括：

- 振动探测器，用来感应警戒线的扰动、脚步声、打碎玻璃声，或者其他一些对建筑物及其周围地区的攻击。
- 门窗上的转换开关。
- 识别体温的红外装置。
- 运用超声波或者微波技术的运动检测器。
- 微波或红外线光束的隐形屏障。
- 地毯下的压力垫，在极端的情况下，可以扩展到在整个地板的每片砖瓦下安装压力转换器。
- 摄像镜头，或许可以再装上运动检测器，用以自动报警或者把实时图像传送到监测中心。
- 设备上的移动传感器，范围从简单的连接线缆到地震仪，再到光导纤维绕成的环。

这些传感器中的大多数都可以通过一种方法或其他方法巧妙地躲避开。警戒线的振动传感器就可通过跃过护栏的办法躲开；对于移动传感器，可以采用非常缓慢的移动方法；而门窗上的转换开关，破墙而入就可以了。设计一个性能良好的传感器组合需要技能和经验（有后者并不能保证就有前者）。

传感器失灵的主要问题是控制虚假报警的次数。超声波运动检测器在空气流动的环境中，例如在中央供暖系统入口处，工作性能就不是很好，而振动检测器在运输过程中也无法使用。像闪电这样恶劣的天气，会影响很多报警系统。飓风天气能使一个小镇警察局每天的电话数量从十几个上升到数千个。在有些地方，甚至正常的天气都可以使保护工作变得困难。保护一个传感器易被入侵者避开（甚至避开警戒线）的地方，对安全工程师来说是一项有趣的挑战（一个有启发的实例是为雪域中的核电站设计入侵检测系统，参看 [74]）。

但不管是在阿拉斯加还是亚利桑那，越靠近被保护目标，就越能准确地控制那里的环境，误报率就越低。相反的，越远离被保护目标，控制误报率就越难。但是要想拖延入侵者，让警卫有足够长的时间赶到，在外围安装可靠的传感器是十分必要的。这就使设计师们处于两难的境地。

如何偷一幅画（3）

风雨交加的夜晚是布鲁诺实施计划的另一个好时机。在不被闭路电视发现的前提下，他会想方设法引起报警器报警。然后快速后退几百码，隐藏在灌木丛中。警卫出来后什么也没发现。过了半个小时，他再次启动报警器。这一次警卫不在意了，于是他就可以混进去了。

虚假报警（无论制造者故意与否）是限制警报器行业发展的祸根。它们对报警的响应能力进行直接的拒绝服务攻击。世界范围内电子战的经验认为超过15%的虚假报警率会降低雷达操作员的工作效率，而且大多数入侵报警响应在开始阶段都运行良好。在攻击无24小时警卫的地方时，故意虚假报警显得尤其有效。因为许多警察部门都有个不成文规定，特定场所的虚假报警达到一定次数后（大概每年2~5次），他们将不再派出警察巡逻车，直到警报公司或者其他相关负责人前去修理好为止。

除拒绝服务这个问题以外，虚假报警也会在其他方面降低系统的性能。例如由天气情况和交通噪音等环境因素引起的虚假报警比率，影响了一般传感器的灵敏度。另外，警报器行业的成功发展极大地增加了报警总数，从而降低了警方对虚假报警现象的容忍度。因此许多人安装远程视频监控系统，通过报警公司的控制台随时监视用户的房屋。警方确定罪犯时也把这种报警系统列于优先考虑地位[417]。

然而，即使是在线的视频连接也不是治百病的灵药。入侵者可以调暗光线，按响火警或者触发附近另一大楼的报警器来避开它。若电话交换机因洪水或飓风产生故障，对盗贼来说，这又是一次绝妙的机会。

除了天气和交通这些环境因素对视频监控系统的制约，布鲁诺的另一个帮凶就是时间。时间一长，许多问题就产生了。长高的草木阻断传感器的光路；警戒线变得松弛；振动传感器运行不稳定；犯罪团体研究出新的入侵技巧。与此同时，岗哨们也因自满而变得松懈。

因此，重点保护地区需要设置若干具有共同轴心的防御地带。最外部的围墙可把醉鬼、野兽和一些普通入侵者阻挡在外；接下来是一片埋有传感器的草坪；再往里是红外线屏障；最后是结构足够结实的建筑物，拖延入侵者进入时间直到骑警赶到（国际原子能组织制定的有关储存多于15g钚元素场所的国际规定是具有一定启发性的[409]）。

然而在很多地方，设置这种防御措施不太可能；它不但造价很高，而且就算有钱支付这笔费用，所在的城市空间也可能不够，例如香港。而且不管喜欢与否，银行的计算机机房一般都会建在办公楼的一层，你不得不尽其所能地做好防护措施。

无论怎样设计，最终选择安装的物理屏障和传感器的组合都不能达到上面防御地带式防御系统的一半。

10.2.4 特征交互

入侵报警器和屏障以一定方式同其他设备相互影响，最明显的是电力。停电会使很多地区处于黑暗和无保护状态，因此重要的报警系统需要安装电池组或其他备用供电设备。另一个较明显的影响是火警器和消防设施。

如何偷一幅画（4）

布鲁诺再一次装成旅游者参观画廊，并且留下了一个定时烟雾弹。烟雾弹在一天早晨引爆，触动了火警，火警直接干扰了防盗报警器，使其忽略了被动式红外传感器的信号（即便接收到了，警戒人员也不会发现他们，因为他正驾驶消防车向现场赶去）。布鲁诺取下毕加索的名画，冲出一条路直奔安全出口。他想方设法在一片混乱中逃跑，即便逃脱不掉，他还有一种更巧妙的办法：就是声称自己是个有公德心的见义勇为者，火警

响起后冒着生命危险去抢救国家宝贵的文化遗产。警察也许不会相信他，但要起诉他恐怕也是件困难的事情。

火警和入侵报警器系统间的相互影响有许多方面。一些防火装置只安装在阻止无意入侵者闯入的障碍物旁边。很多计算机机房备有自动灭火器，考虑到全球变暖问题，一般不使用二氧化碳灭火器，因为它会喷射出大量二氧化碳气体。过量的二氧化碳对未受过训练的人是致命的。它使房间里的能见度降到几英寸，并且产生可怕刺耳的噪声。冲出这样充满气体的房间，比想像的要困难得多。有时，灭火器可以产生类似于报警的功能。比如说，报警器出了故障，一个醉汉闯进计算机机房并点燃香烟，灭火器立即开启，可能在灭烟过程中入侵者会暗自窃喜，笑声相当于报警。这会令你的律师不很愉快。

然而，最严重的特征交互发生在报警器和通信系统之间。

10.2.5　攻击通信系统

一个经验丰富的盗窃犯不仅仅只攻击传感器，也会攻击通信设施。有时，这将意味着攻击传感器和警报控制器之间的电缆设备。

如何偷一幅画（5）

布鲁诺走进一家艺术博物馆，趁工作人员不注意时，剪断一个窗户开关的电线。这样一来，他便可以在晚上从该扇窗户自由出入了。

很有可能工作人员或清洁工人被贿赂、被利用或者被强迫对安全设施进行破坏（尤其当所要提防的对象不是布鲁诺，而是阿布杜拉曼这种人时，就更有可能发生）。因此最好的办法就是经常测试并运行系统（包括传感器）。也就是说要检查备用设备（例如被封条封住的设备），严格地构建管理系统以及防干扰电缆（对重要场所报警器的维修和测试工作不能由一个人来完成，必须是两个人）。

保护报警传感器和控制器之间通信系统的传统方法是物理上的方法：对每一个传感器配备多条电线并埋在混凝土中或者使用密封性很好的高压电缆。目前最先进的方法是对通信系统加密。例如 Argus（最初是为核实验室开发的系统），它运用了 DES 加密技术来保护传感器间的链接。[303]

但是对通信系统更多的破坏，是发生在报警控制器和提供或组织响应的安全公司之间的环节。

如何偷一幅画（6）

布鲁诺给画廊打电话，声称自己是安全公司处理报警的工作人员。他说他正在升级计算机，需要他们报警控制单元中的序列号。一个年轻官员很配合地把序列号给了他，并没有意识到盒子上的序列号就是保护通信设施的密码。布鲁诺花 200 美元买了一个同样的控制器，花了半小时学习如何使用 EEPROM 程序之后，他便有了一个功能作用相同的警报控制器单元，并把它接到要偷窃的画廊的电话线上。这样一来，即便是在情况不正常时，报警器也会显示“一切正常”。

用伪造的警报设备来代替或者用计算机来模拟，这是欺骗。有很多关于“黑匣子”的报告，说的就是欺骗那些过时的或设计不良的警报控制器。其中一个报告就是，盗贼窃取了价值150万美元的从中国进口的玉雕和黄金珠宝，这起偷盗让进口商破产了。这些珠宝玉器存放于新泽西州哈肯萨克市的仓库中，而仓库中设置的警报系统被切断了。通常，那样做会触发安全中心的警报，但是盗贼在外部电缆上附加了一个自制的电子装置，使得信号可以持续显示“一切正常”[371]。

在现代的警报系统里，或者是由地下室的警报控制器发送用密码写的伪随机序列至警报公司，如果被中断就假设发生了最坏的情况，或者是由警报公司发送控制器周期性随机询问信号，这些信号是加密的，并会被反馈回来，就像IFF（敌我识别）那样。

然而，因为警报系统是由不了解安全协议的工程师设计的，所以设计经常存在缺陷。密码算法可能比较简单，或者密钥过于简短（因为没有能力或者受输出规则限制）。布鲁诺可以录下用密码写的伪随机序列，然后以渐渐放慢的速度把它们重放，这样到了星期一的清晨，他就累积下了5分钟的“松散”时间，可以用来发动闪电袭击。

报警失灵的一个更频繁的原因是总体设计失误。一个典型的例子就是，远程维护的拨号上网用的调制解调器端口，使用的是用户从不改变的默认口令。另一个例子就是让密码等于安装设备驱动程序的序列号。除了容易受到社会工程式的攻击外，同样的序列号还经常出现在订单、发票和其他书面文件中，而这些单据是许多人都能够看见的（一般来说，使用现金购买警报控制器是一个好办法。这样就减小了购买一个原本就有缺口的报警器的可能性，但是大公司经常很难做到这一点）。

现在你可能决定不再做艺术画廊生意。但是最厉害的还在后头，是对防盗自动警铃系统最强有力的攻击破坏。它是在3上的变型，其目标不是传感器，而是破坏信号的传导。

如何偷一幅画（7）

布鲁诺剪断画廊的电话线，然后隐藏在几百码远的灌木丛里。他数着进入和离开的穿蓝色制服的人数。如果数量相等，一般可推测出管理人一定在说：“真讨厌，我们要在明早修理它”或者类似这种意思的话。此时，布鲁诺知道他可以开始工作了。

这或多或少是抢劫银行保险库的标准方法。然而，具体的操作形式可以从简单地转接电话公司路边的接线盒，变化成更成熟的攻击模式。就是让不同地区的多个报警器同时报警，这样一来，当地警方就陷入了困境（与扰乱防御系统相比，这是此方法更加有效的原因）。

有一个案例，小偷剪断了新泽西州的三条主要电话电缆，破坏了在哈肯萨克市牧场埋设的三家警察局、数千家住宅和商店的电话及警报设施。他们利用这次机会从美国经销商那里盗取了价值210万美元批发量和800万美元零售量的西恩皮卡德（Lucien Piccard）手表[371]。另一个案例是，俄克拉荷马州代理商谢里夫切断了塔尔萨50 000家的电话线，然后潜入一个麻醉剂仓库进行盗窃[762]。第三例，一个恶棍在伦敦霍尔本电话局引爆了一枚炸弹，中断了哈顿花园珠宝地区许多商店的服务。这种大面积的拒绝服务攻击使警报响应容量达到饱和。这次攻击的覆盖面，相当于一次核武器袭击。

将来，计算机与通信设备所关注的攻击方式可能不包括爆炸性攻击，但是会包括一个基于软件的并对网络设备展开的分布式拒绝服务攻击。与其引起邻近地区的全部警报器报警

(某种程度上可能因为警察的干预而实现不了)，倒不如使纽约几千个报警器随机报警，产生一种类似于飓风或者能源中断的效果，为盗贼们创造下手的好时机。

有关警报系统还有严重关系到保险公司利益的一方面是，电话公司的一些工作人员很有可能被贿赂进行了虚假报警。因此保险公司喜欢警报通信系统由匿名的信息包组成，这样电话公司的大多数工作人员就与任意特定的报警无关了，使有目的的拒绝服务攻击变得更加困难。但是对于实施大多数报警系统信号传输的电话公司来说，喜欢把电话交换机里的传输信号集中起来，这又使有目的的拒绝服务攻击变得容易（有关讨论内容在 [586])。

由于这些原因，伦敦保险市场（世界上大多数主要的再保险生意在那里进行）规定一个地方的警报控制器若加保超过两千万，就必须设置两种独立通信工具。一种选择是采用专用线路和分组无线电通信系统，另一种选择是采用有两条天线的无线电通信系统，这样一来，如果一条线路被干扰破坏，另一条也可以将警报信息传送出去[⊖]。在核工业领域中，国际原子能机构的规章中规定，包含超过 500 克钚或者 2 千克 U-235 的场所的室内必须设有独立的警报控制中心和响应机构 [409]。

最后，尽管物理安全不是这本书的重点，但值得注意的是很多物理安全事件起因于一些可以进入工作场所的愤怒的人员（以前的员工或者客户)。警报系统应该能处理任何时候都可能发生的事件，不管白天还是黑夜。

10.2.6　经验教训

可能有些人不理解为什么一本讲解计算机系统安全的书会花几页的篇幅来描述防盗警报系统。原因如下：

- 处理拒绝服务攻击是很多安全系统设计中最难的部分。况且，当坏人逐渐了解系统的弱点时，它也常常成为最重要的部分。入侵警报系统为我们提供了大量可借鉴的应用知识和经验。
- 尽管通用分布式系统越来越难遵循上面提到的经验，考虑整个系统——从入侵到检测、警报、延迟和响应，仍是广泛适用的。
- 观察报告表明：最外围的防御系统被认为是最可靠的，然而那些不太可靠的防御系统也需要进行部署。
- 在安全工程中，报警失误率和误报警比率间的权衡问题是普遍存在的。
- 在一些难以捉摸的问题上，可以从警报器行业学到不少经验。例如，一些美国飞机场的 X 射线机使用虚假报警插入设备，来确保警报系统和人员可以有效逗留：机场检查人员每换一班就插入一张枪或者炸弹的图像。根据他们的错误率来不断评分。
- 因为不了解这种为查理设计并且希望可以阻挡布鲁诺的威胁模式，所以产生了很多实际的故障。知道什么地方会出错是必要的，不要仅仅依靠犯罪小说家的假想。
- 最后，我们实在不能把一项安全工程设计中的技术部分留给专业子承包商去做，因为关键的职员往往容易成为攻击者。

⊖ 在我还是银行家的时候，有一段时期我常常担心是否会有两个手法较熟练的坏人背地里同时剪断两条电缆。后来我认为对这种威胁模式的担心是没必要的。因为分行的地下室内仅仅储存了 100 000 美元左右的现金。而规模巨大的现金处理中心配有 24/7 名员工，在那里，威胁模式就集中在了那些不诚实的内部人员或劫持人质的犯罪分子等身上。

除了这些系统级的经验外，还有其他一些有关防盗警报行业应用工具的使用经验。前面已经提到过临时爆炸装置；在下一章里，还将讨论防篡改处理器，用来识别以及拆卸攻击装置，并且通过警报响应破坏对方的全部密码材料。

10.3　预付费仪表

下一个案例研究来自预付费计量仪表。在很多系统中，用户可以在一个地方购买代币卡，在其他地方使用卡中储存的币值。这种代币卡可能是密码卡，可能是加磁条的磁卡，或者像智能卡那样可再充值的充值卡。

还有一些例子，包括邮政部门的邮资已付机，图书馆内影印机使用的储值卡，滑雪胜地的缆车使用卡，以及在大学宿舍使用的洗衣卡。很多运输卡都是相似的，特别是如果验证运输卡的自动收票机的终端是安装在公共汽车或火车上，它们通常是离线的。

对这些系统的主要保护目的是防止储值卡被复制或者被伪造。复制一张地铁票不是太难，复制密码也很容易。如果使所有的代币卡独一无二而且在两部终端上都记录其使用情况是不实际的。然而当代币卡的接受装置没有通信信道，不能向代币卡发行机构反馈信息时，事情就变得更加复杂了；这样的话，所有的复制品和伪造品都会在容易受到物理袭击的终端上脱机进行检测。如果仅仅使用一个通用的主密钥对全部代币卡加密，那么恶意攻击者就可以从一个被盗的终端中发现这个密钥，然后随意地复制或伪造，作为代币卡的卖主，与我们竞争市场。

在服务器终端上也存在着攻击。我们当地一家超市的员工自助餐厅内，自动售货卡系统就被巧妙地攻击过。正常状况应该是，自动售货卡充值时，自动售货机首先读出原有的金额，然后输入新增金额，最后写进新增的金额。攻击者先插入一张有余额的自动售货卡，显示 49 英镑，紧接着插入一张空白自动售货卡。然后取出第一张卡，再往机器里塞进 1 英镑硬币，随后机器输进 50 英镑到空白卡中。这样一来，犯罪者就有了两张卡，总价值 99 英镑。防止这种攻击的方法是在机器内设置两个伸出的控制杆，卡紧卡片。但是，这样一来，攻击者只要切下第一张卡的一角就可以达到目的，这种预防措施不易成功（见图 10-1）[479]。

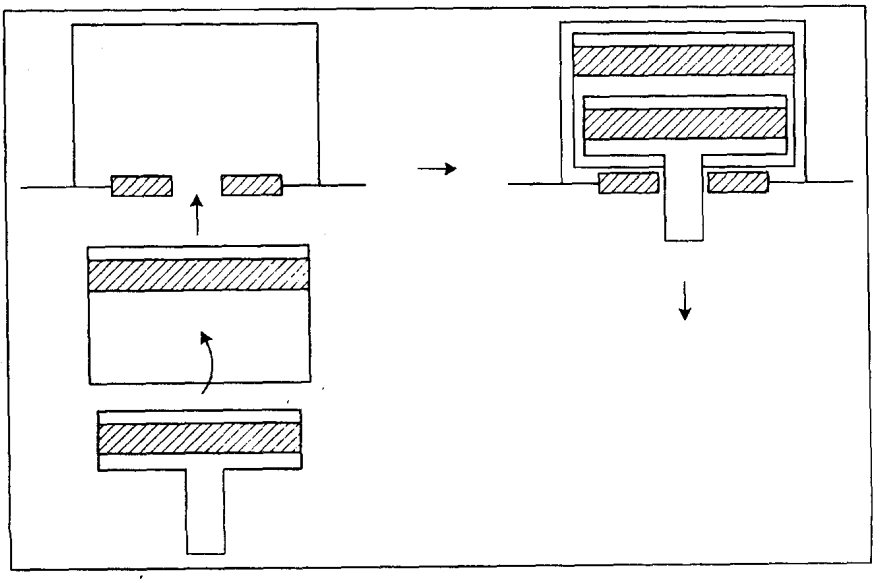


图 10-1　两张预付卡的重叠

这种攻击是有趣的，它表明攻击没有加密的卡没有任何差别。尽管理论上这种攻击可以通过在两终端同时记录的方式制止，但是设计将会很复杂。

但是我们不必因为存在这种欺诈风险而改用其他预付费方式，否则我们就要冒着涉及更加聪明的反预付费验证的风险，从而因忽视系统级问题而成为泰坦尼克效应的牺牲品。在大多数售票系统中，小欺诈是容易的。不愿买地铁票却想搭便车的人可以跳过地铁站的障碍物；可以在其电表旁设置一个用金属丝缚住的旁路开关；在滑雪升降机经过时，在停车时可以使用一台扫描仪和打印机伪造的代币票。我们的目的是防止欺诈变得系统化，至少使小欺诈行为实施起来不太方便，更重要的是应该设计出更严格的技术，防止任何人大范围地伪造代币卡来发展黑市从而影响客户生意。

下面详细讨论的例子是预付费电子仪表。选择这个例子是因为我咨询过一个项目，就是使南非超过250万个家庭实现电气化（纳尔逊·曼德拉在竞选时的重要竞选誓言，这部分内容的具体描述在［39］）。大部分经验教训可以直接应用于其他代币票系统。

10.3.1　需给电表

在许多欧洲国家，不能使用信用卡的户主（可能因为他们享受福利或有诉讼在身，或是其他什么原因）用预付费仪表购买电和燃气。在过去，这些预付表是投币式的，但是硬币的收集工作费用很高，卖主就设计出基于代币卡的仪表代替原来的硬币式仪表。顾客要去商店购买代币卡，可能是智能卡、可任意使用的磁卡或者只是个密码卡。如果自动售货机没有特殊要求，使用密码卡是最方便的：超级市场结账柜台上可以销售代币卡，甚至可以通过电话购买。美国的读者过去可能习惯于通过给电话中心打电话使用自己的信用卡购买密码卡，给仪表充值：密码充值仪表。这个系统与预付需给电表完全相同。

代币卡应该被认为是包含一个或者更多指令的比特串，使用对仪表独一无二的密钥加密，仪表可以对该卡解密并且对仪表自身产生作用。大多数代币卡的功能是读取信息，“仪表12345，分配电量50 kWh!”，但是一些代币卡也有维护功能（见图10-2）。这个思想就是仪表可以分配购买的代币量并能中断代币值供应。

这些仪表的制造业已经发展成为大规模业务。英国大约有一百万台电表使用两种方式的预付费，大约60万个燃气表使用智能卡。其他一些国家也安装了预付费电表，包括

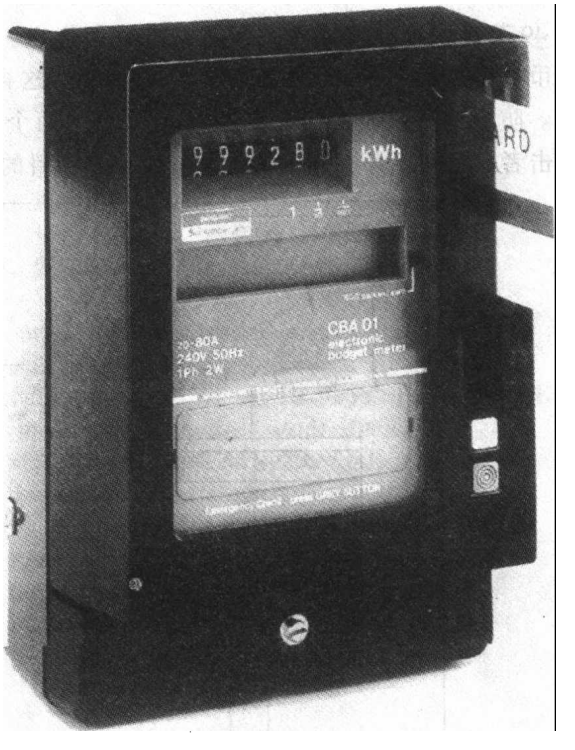


图10-2　预付费电表

巴西、印度、纳米比亚和象牙海岸。第三世界国家发展潜力大，因为那些用户没有地址，更不用说信贷等级。就拿南非来说：发展预付费电表是政府实现其选举的誓言——保证数百万家庭用电的惟一出路。在发达国家，仪表化的主要动力来自于降低管理的费用。电气行业表明，在市区，记账系统可以吞没零售商人 20% 的收入，然而预付费系统仅花费不到 10% 的费用。

10.3.2　系统如何工作

安全系统需要一个预付费仪表系统好像是很明显的事情。代币卡不能被容易地伪造，即使是真的代币卡也不能在不正确的地方使用，或者在正确的地方重复使用两次。代币卡应该是防篡改的（但这样代价比较昂贵）或者是独一无二的（可以很容易地用连续的数字和密码系统实现），但是要想建立一个健全的系统，还需要很多方面的经验。

预付费仪表需要一个密钥从自动售货点验证指令。典型的系统有一个发售密钥 K_V，作为主密钥，在需要的时候通过发售密钥对预付费仪表 ID 加密得出设备密钥。

$$K_{ID} = \{\mathrm{ID}\}\ K_V$$

这跟在第 2 章中讲的停车场进出控制设备的密钥多样化技术是一样的。多样化发售密钥 K_V 得到一组预付费仪表密钥 K_{ID}，这就提供了一个非常简单的解决方案，在这个解决方案里所有的代币卡都是在本地得到的。实际情况经常没有这么简单。在南非，许多人使用长期车票从城镇或家乡到他们的工作地点，因为他们或者工作时间不在家里，或者想要在上班时间购买长期车票。所以他们可以在一个长期车票发售站点注册，该站点有相应的安全协议可以连接到拥有预付费仪表的发售站点从而得到预付费仪表密钥。为了结算，销售数据则从相反的方向传递。这种技术与为 ATM 网络发展的技术非常相似。

用统计意义上的平衡来识别所谓的非技术损失，即包括那些通过仪器篡改或者未经授权的直接接入到主电缆进行的盗窃。这个机制通过一个支流线路的仪表来比较从仪表中读取到的数据，这个仪表可能负责供给 30 个房子的电力，并把代币卡发售给这些房子的户主。这比看上去要复杂一些。用户们保存了代币卡票据，而读电表的人可能会错误地记录读取仪器的日期，其他的错误也有可能发生。正如在第 9 章中讨论的那样，自动售货统计也应用于传统的平衡系统。

自动售货机本身维持着一个信贷平衡。它们依赖于一些防篡改的安全程序，保护卖主不受其他售货机和外国售货机的解密系统的侵害，或是影响到本身的平衡。随着每笔售出业务的完成，这个信贷平衡被打破，直到现金被划拨到当地运营的公司才重新恢复。这家公司接下来又要将钱支付给分销网络中的更高一级公司，以此类推。在这里我们提供了一个被销售点的数值计数器而不是仅仅被金库服务器上的分类账目数据部分加强了的记账系统的例子。理论上有价值的柜台的失衡能够被更高层次的统计意义上的平衡的支票填补。这种分布式安全系统经常能见到，比如说，以万事达卡为应用基础的 Mondex 电子钱包应用模型。

所以，什么地方会出错？

10.3.3　什么地方会出错

拒绝服务对预付费仪表来说是一个严重的问题。因为从电表到发售站点没有反馈的渠道，所以电量的使用多少只是存储在自动售货机上。自动售货机的代理商通常是小店主

或者其他的企业主，他们有少量的资本，被允许用信用卡销售电力。在一些情况下，代理商会将自动售货机弄坏，然后声称它们被盗窃了。这对于小的代理商来说是容易实现的。但是对于有一定规模的组织，比如当地政府，它们被允许通过很多的渠道发售大量的电力时，这种做法是肯定会被揭露的。需要增加复杂度来对付不值得信赖（人为不可信）的当事人。

就像要有防盗警报器一样，环境安全是非常重要的。除了大范围的温度变化以外（就像南非大陆上变化多端的国家一样），许多地区有非常严重的雷雨，电表的微处理器可能会被长达3公里的闪电影响。

当电表被闪电击坏的时候，用户们会发出抱怨，这样他们可以取回信用卡中他们所说的没有用掉但是（由于闪电）被记录的部分。下一步他们会使用主线连到电表当中，仿效电表被闪电击穿（而导致代币值丢失）的情形。于是就会出现某一段线路（在代币卡作用点之下的部分）被破坏则会丢掉数量巨大的电力的糟糕情形。于是，被拒绝服务攻击的系统将很难被识破并且非常流行（如果银行对于离线的电子钱包智能卡没有完全的配平策略，它们可能会变成一个非常严重的问题。当一个顾客说他的卡失效时，银行只能选择如下两种做法：一种是恢复顾客声明丢失掉的金额；一种是告诉顾客她/他的钱丢失了）。

还有更糟糕的情况。预付费程序中发生的代价最大的安全事故是索韦托（南非）的孩子们发现的，当线路输出电压从220伏降低到180伏时，有一种电表会输出信用卡的最大值。于是很快，孩子们把铁链扔到11千伏的高压线支路上并将他们邻居的所有信用卡都充满。这个错误是由于电表中ROM的一个小错误产生的，这个错误没有被检查出来是因为对于接地电压输出测试没有明确的定义。实际上，发达国家制定的环境标准对于发展中国家来说是不够的，需要重写。这件事情的结果是所有100 000个电表被拿回来重新载入ROM，对此负责的公司差一点就倒闭了。

还有其他大量的错误。有一种型号的电表不会出售某种特定电量值，它的检测方式是多少电量对应多少代币值的方式。因此当线路被设置为某一个电流方式的时候，它计费很少，所以人们总是让该线路在这个负荷下工作。另外一个问题允许退款，但是退回的代币卡仍然可以用（依次将退回的代币卡放入“黑名单”中是非常麻烦的，因为所有的代币卡不是顺序地储存和使用的）。另外一个电表记录了最后输入的代币卡的序列号，如果轮流使用两个相同的代币卡进行退款充值操作，则这两个代币会被无限的充值。

对于提款机，真正的安全漏洞来自错误和疏忽，可能非常严重，也可能在事故发生后通过非常幸运的渠道得知。但是有时候这些发现是以几百万的损失为代价的。

这里还有其他一些教训：

- 如果你控制了市场渠道，预付费是非常便宜的，如果你想通过第三方出售预付代币（比如银行和超市）的时候，可能很快它会变得非常贵、麻烦而且带有风险。这通常是因为有很多组织参与的中介引起的安全工程的问题。
- 如果一个生意需要改变，而它同时要影响安全基础设施，那将是非常昂贵的。比如将出售代币卡的方式从本地店铺改为计算机支持的方式，将会是一个非常花钱的事情。
- 使用循环技术，因为这样看起来比在一张白纸上设计代币卡出错的可能性小一些。很多我们需要的预付费电器是从提款机借鉴来的。

- 雇佣很多专家，一个专家很难知道很多领域的问题，即使是最好的专家也会出纰漏。
- 不管做什么，很多很多的小错误还是会发生，所以你绝对需要延长测试的时间。这是所有的问题和错误应该被发现的地方。

电表是一个很好的学习使用代币票据的实例。运输代币票据、影剧院的代币票据，甚至运动会的代币票据都可能是规模很大的代币应用，但是我不知道有多少针对它们失败的研究。在许多的案例当中，终端设备（比如电表和十字转门）是很脆弱的，所以必须防止大规模的欺骗行为。这意味着我们要更多地注意中间的服务器，比如自动售货机，并且让它们更加安全以防止别人的操纵和篡改。还可以使用那些足够安全以至于人们无力开发攻击设备的系统。

下面将要看到的是一类长期经受严重攻击的终端设备，这些设备必须做得比电表更加能够防范攻击。这些具有威胁性的模型包括传感器的使用、拒绝服务、财务欺骗、程序上的失误、计算错误和操作人员自身腐败。有借鉴意义的研究领域是交通监控系统。

10.4 计程器、转速表以及卡车速度限制器

一些计量系统用来监视和控制机动车辆。最熟悉的可能是你的车上用到的里程表。在买一辆二手车的时候你可能会担心是否这个车的仪表已经被拨回过，也就是说，显示出的里程数是否已经被减少了。如果里程表变成数字式的，这种拨回就变成了一种计算机欺骗行为；这在［170］中已经有报道。

另外一个很普遍的例子可能是计程器。如果不会得到惩罚的话，任何出租车司机都会有操纵计程器使之显示更多的行驶路程（或者更长的等待时间）的动机。有很多种其他类型的“黑盒子”用来记录机动车辆的运动，从飞机到渔船，以及装甲的银行运钞车，而且他们的操作者都可能有不同程度的动机来改变这些黑盒子。例如，从1990年开始，一般价值6 000 000的发动机都会配备有黑盒子，用来记录紧急数据。这对律师来说可能是一件好事；同时这也有很多的保密因素，例如，黑盒子的存在于1999年才成为公开信息［768］（在第21章中将谈到这些问题）。

在这里将要用作案例研究的是转速表。一个司机在驾驶车辆的时候因睡着引起的事故通常是喝醉引起的事故量的几倍（例如，在英国，这个比例是20%对3%）。涉及卡车的此类事故会因为卡车司机的疲惫引起致命的伤害。所以大多数国家都会规定卡车司机的工作时间。在美国，这些法律将通过检测站来实施，而欧洲的国家会使用称为转速表的仪器，它会24小时将车辆的速度记录在一个圆形的蜡纸表上（见图10-3）。

这个图表是加载在转速表上的，而转速表是车上的里程器/里程表的一部分。该表在仪器内部的一个转台上缓慢地转动着；记录有三种：速度（外部的跟踪），司机在工作还是休息（中间的跟踪），以及行驶路程（内部的跟踪——每10千米一个记号）。当然也有一些例外，不过在这里就很无关了。在欧洲，你所驾驶的车辆上没有安装转速表，或者你没有将你的起始时间和地点记录在表上，都是违法的。你必须将近几天的表记录都带着，以此来确认你遵守了相关驾驶时间的规则（一般是8.5小时一天，还有对每天中休息时间和每星期休息日的规定）。一些转速表有特殊的指针，用来记录一些环境变量：例如急救车使用的闪光灯，冷藏车的温度，以及装甲车门的开关（这是转速表在北美洲的一种普遍应用）。

欧洲的法律同时将卡车在高速路上的速度限制在100千米/小时内，在其他的路上更低。

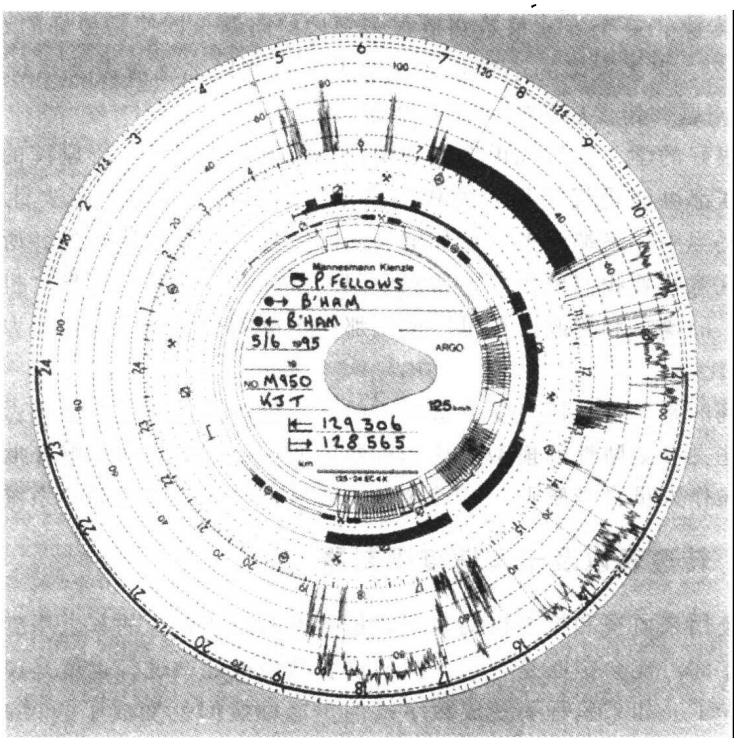


图 10-3　转速表

这并不通过汽车超速监视区和转速表来实行，而是直接通过一个也由转速表传动的速度限制器来实行。转速表还通常用来调查其他的违法行为，例如未经许可的有毒废物的排放，以及由车队管理者用来检测燃料被盗的情况。很明显，一个卡车司机有足够的动机来破坏转速表。[⊖]

EU 正在从基于纸板的系统转换为智能卡系统，这使得此类问题高度专业化。当执行安全工程任务时，首先需要知道哪里出了错。我必须提到的大部分内容都针对于计程器和其他监控器。卡车司机一般希望他的车辆显示出走了比较少的路程，而出租车司机则相反。这对于实际的篡改技术会有一点影响。

10.4.1　哪里出了问题

根据 1998 年对 1 060 位被判决的司机和操作者的调查［31］，违法行为被分为以下几种。

10.4.1.1　一般的转速表非法操作是如何进行的

大约有 70% 的被判决的违法者不是因为篡改，而是利用程序使功能削弱。例如，一个在敦提（英国苏格兰东部海港城市）和南安普敦（英国英格兰南部海港城市）都有房屋的公司可能每天都需要 4 个司机来开一辆机动车，在每个路途方向上大约行驶 500 英里的路程，

⊖ 安全工程中的一个一般原则是一个人不应该聚集目标。因此，NATO 规则禁止在密级信息的容器中传送钱或其他有价值的东西，如果你不想某人设法盗取你的军队的工资表，以及偷走你的间谍卫星照片的话。强制卡车司机损坏转速表来突破速度限制是一个严重的设计错误，但是现在有些很顽固的人已经不容易改变了。

花10小时的行驶时间——这样如果每天只用一个司机是不合法的。一般的欺骗方式是用两个司机，他们在一个中间点，比如彭罗斯（城市名）会合，互相交换卡车，并且将新的纸表插入到转速器中。然后从南安普敦出发的司机驾驶从敦提驶出的车返回家里。如果被拦住并要求出示表的时候，他给出当天从彭罗斯到南安普敦的表，前一天从南安普敦到彭罗斯的表，和大前天从彭罗斯到南安普敦的表，如此类推。这样，司机就给出一个错误的假象：他每个晚上都是在彭罗斯度过的，这样的话也就合法了。这种在工作日内半程中交换车辆的（很普遍）现象，被称为多重幻影。这在欧洲大陆上很难被查出，那里的司机可以星期一在法国的仓库开车，星期二在比利时，而星期三在荷兰。

更简单一点的作弊方法包括错误地设定时钟；假装搭便车的人是正在休息的驾驶员；或者将起始点记录为一个有很普通名称的村庄——比如英国的米尔顿或者西班牙的La Hoya。如果被拦住，驾驶员可以宣称他是从附近的米尔顿或者La Hoya开始行驶的（表10.3中的图就显示了这样的一些违规行为。例如，列出起始点是“B'HAM”，这可以是伯明翰或者白金汉郡，并将时钟从14：30拨回到14：00），这可以从那些重叠的痕迹上看出）。

这些伎俩通常还有司机和计量表格操作者之间的勾结。如果操作者被要求做出一些图表和支持文件，例如付款记录、测量站点以及票据，他的办公室可能很容易被全部焚毁（有许多卡车公司在离卡车有一段安全距离的廉价的小木屋外操作，是很值得注意的）。

10.4.1.2 供给上的篡改

下面所提供的作弊种类，大概占到总数的20%，它们与转速表设备的供给有关，包括电源和动力的提供，电缆和封印。

老式的转速表使用一个旋转式的线轴——与直到19世纪80年代早期还在使用的计程器一样——这种很难作弊。例如，如果你卡住卡车的里程表，那很可能是因为你剪断了电线。而电动的转速表比较容易作弊。这些表从一个变速箱中的传感器得到输入，这样在转轴转动的时候会送出电脉冲。一个常见的作弊方法是将传感器的螺丝旋松大概十分之一英寸。这就引起脉冲的停止，好像车辆是停止的。为了防范这种欺诈，传感器用线和铅印固定下来。但动机不纯者仍然可以贿赂装配工，使之将线以逆时针方向缠绕而不是顺时针，这就使得传感器在旋开螺丝的时候是变松而不是断开。封印是发放到车间而不是独立的装配工这一事实常常会因为前面的欺诈行为引起繁杂的诉讼。

大多数的作弊方法更简单。司机剪断电线或者将转速表的保险丝用一根已经断开的保险丝代替（有制造商曾经试图将卡车的反锁制动系统安到同一根保险丝上来避免这种作弊。但许多司机选择能够更快到家而不是更安全地驾驶车辆）。可以从图10-3中看出在早上11点左右有一个电源切断的证明，有很多地方显示的外部行驶的速度可以从0突然变到超过100千米/小时。这就表明电源被暂时切断了，除非在远距离的描绘上也有这样的不连续。这种作弊方式是可以被察觉的。

10.4.1.3 对于设备的篡改

第三类作弊就是在转速表上做手脚。这大概占到违法行为的6%，不过现在的设备发展导致了这种行为比例的下降，因为操作数字的通信比起过去那样对旋转线轴的违规操作要容易许多。这类典型的违法就是故意实施错误的校准，通常是通过和装配工的合谋进行的，不过有的时候可能是因为司机将设备上的封印破坏了。

10.4.1.4 高技术的作弊

当前用于对设备进行篡改的一些装置可以在图 10-4 中看到。照片右边的塑料圆筒被标记为“电压调节器——日本制造”，而实际上它并不是一个电压调节器（这实际上显示意大利制造）。这个设备被连接在转速表上，然后司机可以通过远端的钥匙链来控制。对它按压第一次可以引起所显示的速度降低 10%，按压第二次可以降低 20%，第三次使得显示降低为 0，而第四次可以使设备恢复正常的操作。

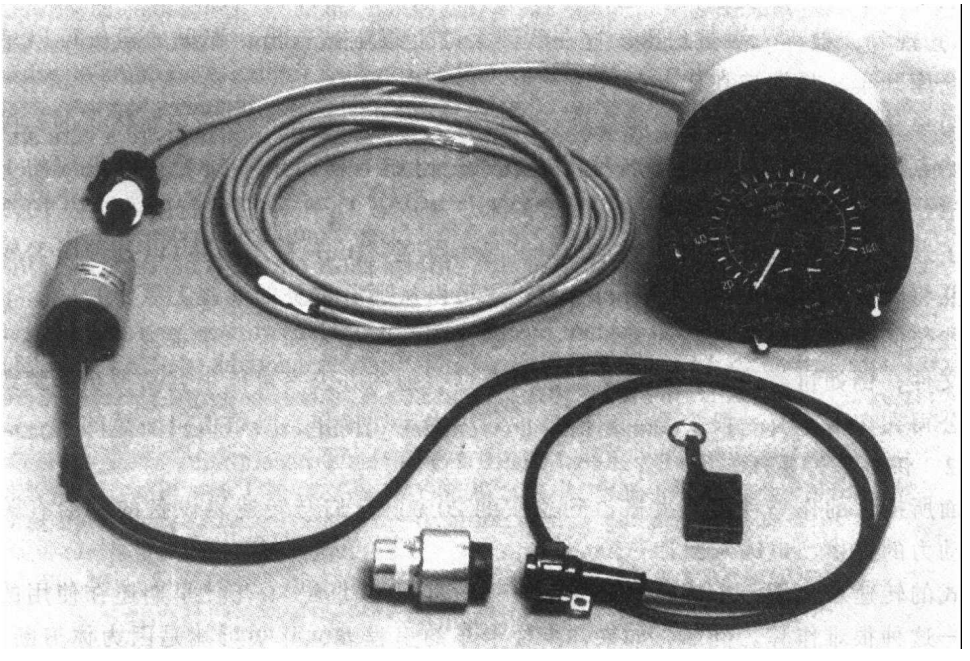


图 10-4　司机通过无线电钥匙链控制带断续器的计程器（此图片获英国 Hampshire Constabulary 许可）

使用这种设备只占到所有违规判罚中的 1%，不过这种设备的使用很广泛。这种设备极难被检查出来，因为它可以藏在卡车上放置线缆的不同地方。警察可能拦住一辆安装了该设备的超速卡车，却无法找到该设备，也就是说无法判罚：被密封的并且明显被正确校准的转速器和警察的雷达或者照相机中得来的证据相矛盾。在军备竞赛中的下一步就是让电子战技术警察来检测和抑制这些“断续器”——这样，毫无疑问，那些不法之徒就会开始使用密码系统来保证钥匙链的通信安全。

10.4.2　对策

制定违规操作转速表的对策，不同国家也有所不同。在英国，卡车会在路边被机动车巡视员拦住并随机检查；特别受怀疑的卡车会在整个国内受到监视；在荷兰，措施集中在检查员拜访一个卡车公司并检查公司的递送文件、司机的时间表、燃料记录等等方面。在意大利，从高速公路收费站得来的数据会用来起诉那些平均速度并不超过速度限制的司机（这就是为什么你经常可以看到卡车停在意大利收费站前）。不过这些措施只可能部分有效，而且司机可以在不同的控制对策之间变动。例如，一个在法国和荷兰之间行驶的卡车司机可以将他的记录保存在法国的一个检查站那里，这样荷兰的巡视员就不能够得到这些信息。

10.4.2.1　智能系统（Tachosmart）

正因为如此，欧盟开始设计一种统一的电子转速器系统，称为智能系统，它将用智能卡代替现用的纸制的表。每个司机都会有“驾驶卡”，这个卡实际上就是卡车司机的执照并且包含过去的 28 天之内司机驾驶时间的记录。每辆机动车都会有一个服务期为一年的机动车单位。特殊种类的智能卡是机械师用来校准设备的，而且通过执法人员在路边读出。

对转换为智能卡最实际的阻碍是并不清楚这种系统如何对付当前已经占到总数 70% 的程序上的作弊。实际上，那对在敦提和南安普敦之间制造多重幻影的司机会使他们的生活更安逸。还可能会花费 10 年时间——一辆卡车的使用寿命——才能转换到新的系统；期间，他们可以用一辆使用旧表系统的卡车和一辆使用新卡系统的卡车。每个司机都有一张表和一张卡，每张卡上记录一天中的 5 个小时，而不需要使用原来容易弄混的两张表。

10.4.2.2　系统级问题

对有关此类问题中系统级别问题的反应，各个国家有所不同。德国想建立一个关于此类表管理的基础系统，这个系统可以接受数字转速表的数据、从现存的纸制表中得到的模拟数据的数字化视图、燃料数据、递送数据，甚至工资——并将这些信息协调起来，不仅用以提供对卡车公司的管理信息，而且为警察提供监控信息。这个想法，和 20 世纪 90 年代中期采用密码管理的系统一样，可以信任大公司，让他们使用自身的车队管理系统，而小公司必须使用许可的车辆监控。

英国并不像德国那样拥有大的车辆监控管理局，所以英国提议包括整合的转速器系统，在卡车上配备 GPS 定位传感器或者使用一个现有的自动计数的地图阅读系统（这是在伦敦首先开始实施的，为了使 IRA 炸弹攻击更困难，如今这个措施已经扩展到全国范围用来监测汽车的逃税者）。

然而，关于隐私问题和国家经济利益方面的不同意见阻碍了欧盟范围的标准化。这取决于各个独立的国家是否需要卡车公司下载并分析关于他们所有卡车的信息。

即使每个国家都可以有效应对此类问题，也不会是万能的，因为存在利益交易。目前，德国警方在实施对于驾驶者的时间控制上比他们的意大利同行更为热心。所以一个意大利司机平常不需要带表，但在穿越阿尔卑斯山的时候就需要将表放入转速表了。同时，德国的卡车司机会用另外一种方式将表取出。这样的网络效应是指在一个指定的国家中所有的司机都处于同一个法律实施的水平上。如果行驶数据可以定期从意大利司机的记录卡上上载并保存在罗马一家卡车公司的电脑里，他们就会处于意大利的法律实施水平上了（或者更低的水平，如果意大利警方并不关心在德国发生的事故）。很容易看到这会使得在具体实施上的效率下降。

10.4.2.3　其他问题

从模拟设备到数字设备并不一定是一种进步。除了电子设备和机械信号的低水平的防篡改能力，还有系统级问题，就是安全状态不能够以统一的方式处理。除此之外，在转速器数字化方面还有很多有趣的问题。

首先，损失数据的具体化，在转速表上冗余的数据会使实施更困难。目前，有经验的机动车检查员在转速表不正常的时候会有感觉，而一旦模拟痕迹被一个二进制的信号所代替，这信号说明驾驶者是否遵守了规则，检查员们就很难依靠过去的经验了（尤其是如果保留了支持纸卡的卡车公司的总部在其他管辖区）。在攻击状态下新的数字化系统不可能比它的上

代模拟系统更能防御此类攻击。

第二，会产生一些新型的拒绝服务攻击（就像针对机械传感器、保险丝等的传统攻击）。一个卡车司机可以用电源对智能卡充电，很容易使其毁损掉；并且根据规章，可以允许司机无卡驾驶15天，等待更换另一张卡。因为在一年中总有1%的智能卡会发生静电损坏，所以故意破坏智能卡的司机是很难被起诉的。相似的智能卡破坏攻击也出现在法国和其他地方的银行智能卡系统中，迫使系统退回到不太健全的后退操作模式。

第三，一些系统（注意那些设置此类计量工具的车间和校准卡）的智能卡功能非常强大。它们可以用来掩饰错误行为和证据，并使速度记录器处于原始状态。这样一来，很可能形成智能卡黑市，某些人使用的智能卡里储存的价值可能总使用不完。解决这些问题的方法是，附加相关制约技术。转速表系统正在被重新设计，采用公钥加密，而不再使用智能卡和车载设备中的普通密码了。

一个特别困难的问题是密钥管理问题。这是一个涉及车载设备系统安全的普遍问题，不仅仅是转速表和类似装置（如计速器），甚至是过去经常用其中来保护小汽车无线电免受偷窃的小汽车门锁和PIN码等普通装置。如果汽车库需要比现有要求更高的安全装置，且其中三分之一的装置有过盗窃记录，那么你将考虑建立什么类型的安全系统？

10.4.2.4　复活小鸭

最近的EU直接指出，为了使图10-4中的断续器使用无效，从变速箱传感器到车载设备，所有的数字转速表必须对脉冲序列进行加密。正如这些装置之间都含有微控制器，且数据传输率相当的低，这些在理论上都不构成问题。但究竟如何分配密钥？如果为车库建立呼叫热线，那可能会造成对热线的滥用。有不良企图的卡车司机同汽车修理工们合谋攻击这种系统已经有好长一段历史了，车库工人滥用帮助热线获取未锁数据盗走小汽车，或获取PIN码盗取汽车上的无线电设备。

复活小鸭（resurrecting duckling）安全策略模型给出了一个解决办法。存在这样一个事实，小鸭子从蛋里出来会把它看见的第一个移动的会发出声音的物体当作妈妈；这叫做印记。相同的，“新生”的车载设备，在移除其包装之后，将认为给它发送密钥的第一个变速箱传感器为它的所有者。传感器在接通电源时起作用。一但拿到密钥，车载设备不再是新生的，在剩余的“生命期间”内只对该变速箱传感器保持有效状态。如果传感器失效，必须更换时，该车载设备就要经历失效并且重新复活作为新生儿的过程，这样才可以附加到新传感器上。车载设备的每次复活记录无法消除，这使滥用车载设备变得困难了。

密钥管理的复活小鸭模型起初用来为医生的个人数字助理或临床监视设备处理数字体温计或其他医疗设备的完全印记。该模型也可以对用户电器进行远程控制。这样对于盗取了家用电器的盗贼来说，由于他无法获取远程控制权仍然无法使用该电器[731]。

另一个该模型的实际运用例子是在武器的安全管理方面。许多执行任务的警察被他们自己所佩戴的枪杀死，这就引发了对武器机械装置安全性的很多考虑。一种方法是设计一种手枪，只有在距警察佩戴的警徽一步左右范围内才可以开枪。而这种设计的关键是警徽和枪的关系管理问题。可能解决的方法是在枪上印上警徽图案，但要滞后一分钟左右。对于警察来说，从军械库取出枪后，在枪上加警徽印记不是一件难事，但对于强盗来说却是个大问题（可以假设警察不能制服歹徒或者不能在一分钟之内追到他，则这个歹徒很有可能跑掉）。这样的机械装置不需要口令就可以减轻现场被掳走军事武器产生的不利影响[106]。

10.5 小结

很多安全系统以不同的方式监控环境中的问题。涉及民用防盗警报器，需给电表、计程器转速表，甚至一些有争议的核安全系统。

保护这些系统最重要的是防止攻击，包括拒绝服务攻击，例如通信中断，用噪音淹没传感器，或者做其他事情，直接或间接地减少用户对其安置的系统的信任度。对于各种数据处理来说，拒绝服务攻击的范围可能被扩大。密钥管理问题可能就是这个问题的一方面，尤其是在应用广泛的低成本分布系统中，中心密钥管理设施显得不合理或者可信域管理者不存在。系统可能不得不处理许多互相存疑的部分，并且由最廉价的微机控制器完成。而且分布式子流的很多组成部分很有可能是掌握在敌人手中的。

通过对这些特定环境问题——防盗报警装置、需给电表、车载转速表的举例说明，应该可以从中获得启发：对因特网的拒绝服务攻击已成为我们面临的主要问题，例如 SYN 泛洪攻击和 DDoS 攻击。

研究问题

目前没有一个通用的工具可以控制嵌式需给系统中的密钥。虽然为自动柜员机网络设计的实现机制（或产品）可能（或）是合适的，但绝大部分设计工作要进行返工；因此常常有安全隐患（将在第 14 章讨论如何应付用于此目的的特殊处理器）。

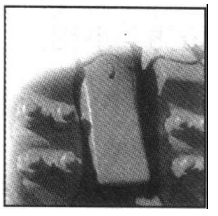
虽然有了一些工业标准（如 CANBUS，用于车载系统间的通信），但在密码系统和其他机制中并没有顶级标准，例如匿名和平衡，用于建立大范围的监控和记账系统。这些标准的建立需要众多工程师的努力。

参考资料

关于报警系统的最好的、最全面的参考教材是 [74]；系统问题在 [586] 中有简明的讨论；通过像美国工业安全组织 [14] 这样的贸易团体或者当地保险业为一些特定国家提供设备；很多国家有非盈利组织，例如美国的保险业者劳工组织 [756]，以及产品的确认、安装方案或两者都有。有关最新传感器技术的研究报告发表在 IEEE Carnahan 文献上 [399]。

预付电子计量器在 [39] 中有所描述，类似的应用——邮政计量器——在 [753] 中。有关转速表的内容，包括智能系统项目，可参看 [31]。最后，用来遵守核武器控制条约的监控系统在 [702] 中有所讨论。

第 11 章　核武器的指挥与控制



在德国和土耳其看到了特别令人遗憾的场景。在飞机跑道上停着一架装有核武器的德国（或土耳其）快速反应式预警飞机，驾驶舱中坐着一位外国飞行员。飞机已经在早期警报时做好了起飞准备，机载核武器也已随时待发。而美国军队惟一的监控措施只有一个站在停机坪上配有一把卡宾枪的 18 岁哨兵。当问到这个德国机场的哨兵当机上的飞行员突然决定紧急起飞（或是出于个人决定，或是因为一条来自德国用以对付美国的命令）时他将如何保持对核武器的控制，那个哨兵回答说他将向飞行员开枪；而 Agnew 指示他向炸弹开枪。

——JEROME WIESNER，
总统科学顾问，在古巴危机事件之后就核武器指挥与控制问题向肯尼迪总统汇报时谈到

11.1　引言

由于担心未经授权使用核武器，或者核技术在不适合的国家或地区蔓延所造成的独一无二的灾难性破害，已使得美国（同其他核国家）花费了大笔资金，不但用于保护核弹头，还用于支持核基础设施、工业和原材料。

相当多的核安全技术已经被公布于众。事实上，对于究竟哪些技术需要保密是有严格限制的，即使该项技术被认为是迫切需要的。许多国家都有能力生产核武器，但都已经决定不这样做（日本、澳大利亚、瑞士……），并同意在民用方式下继续对核材料进行控制。此外国际上还有防止核扩散合约，比如由国际原子能机构（IAEA）强制规定的对核材料进行物理保护的公约［409］。

每年民用核反应堆可生产十一吨钚。所以必须找到保护这类核原料的方法，而这些方法又必须能增加国际上的信赖——不仅仅是在政府之间，还来自越来越对此产生怀疑的公众。

大量的安全防护技术都是从核项目中应运而生的。美国能源部武器实验室——Sandia、Lawrence Livermore 和 Los Alamos 以几乎不受限的预算，为了两代人尽可能安全地保障核武器与材料而运转。我们已经看到他们开发的一些更为通用的副产品，从多于 12 位数字的口令不能在战场上使用到高端的防盗警报器系统。将光纤环绕在被保护的仪器周围以及使用干涉效应来检测长度小于一微米的变化的技巧则是另一项相关技术。它被设计成环绕在军火库核弹头的周围，当核弹头之中的任一个被移动时无失误地报警。

在后面的几章中，我们仍将看到更多源于核工业的技术。例如，动用了美国能源部基金开发的虹膜识别技术是最准确的、采用生物测量学来识别个人身份的系统，旨在控制钚储备库的入口；并且许多关于抗干扰和干扰检测技术的专门知识都是从防止滥用盗窃来的武器和控制仪器中产生的。

在本章中，我描述了这些技术得以开发的背景，以及一些可能在其他地方获得应用或产生威胁的技巧。因为我不是内部人员，所以我只能把公开的资料聚集到这一章里，因此也就

可能会忽略某些重点（一位有着核工业相关技术经验的校对者十分明确地告诉我说这些材料的确“准确但不完整”）。不过，即便是从这些可获得的材料中，还是可以学到很多的知识。

11.2　肯尼迪备忘录

古巴导弹危机过后，美国政府开始意识到，一场世界战争可能会因一场事故而引起。成百上千的美国核武器都被存放在盟国，比如希腊和土耳其这些不是特别稳定并且偶尔互相发生摩擦的国家。这些武器都只是被那些象征着美国的监护力量保护着；没有实际的原因可以解释为什么不能在危机到来时及时获取武器。美国官方还考虑到可能未经授权批准的核武器使用；举例来说，一个地方官员在倍感压力下是否会想“要是华盛顿了解我们的情况是多么的糟糕时允许我们使用炸弹就好了。”这些忧虑都是在总统科学顾问 Jerome Wiesner 所做的三项应急情况研究中被证实的（本章开头所引用的段落可以在［734］找到）。

肯尼迪总统作出的反应形成了国家安全活动备忘录第 160 号。该备忘录命令美国分配在其他国家的 7 000 件核武器应该得到积极的控制，否则应被销毁［705］。

能源部已经开始研究保护核武器的安全设备。基本原则是核弹环境的一个或者多个独立的部分在武器被装备前必须可以察觉到。例如，导弹弹头和一些自由下落的炸弹必须承受零重力，而武器的外壳必须承受成千上万个重力加速度。有一个例外：原子爆破弹。这类核弹被设计成由地面部队带至目标并采用定时导线引爆。对于一个独特的环境传感器来说，必须阻止它们偶然或者恶意引爆的机会。

在当时的技术发展状况下的解决方案是机密触发码，它将激活一个深深埋入武器核心位置的钚陷阱的螺线管安全锁。主要的安全工程问题在于维护。当锁被暴露时——比如说，为了替代动力装置——密码可能会被获知。显然，每件武器都含有相同的密码是不能被接受的。必须采用分组密码——发射码只可被一小部分的弹头所共享。

在肯尼迪备忘录之后，人们建议所有核炸弹都应该使用密码锁保护起来，而且还应该有一个“统一解锁”指令，只有总统或他的合法接班人可以发出这条指令。问题是要找到一种方法可以安全地将这条密码翻译成大量的个人引爆码，每个引爆码都适用于一小批武器。到了 20 世纪 60 年代和 70 年代，问题变得更为糟糕，此时的原则由严厉打击变为“可以衡量的对策”。不同于装备全部核武器或者全部不装备，现任总统必须能够有选择地装备成批的武器（比如“所有在德国的军火”）。

11.3　无条件安全认证码

核安全需求带来了一次性认证码理论的发展。这个概念与发明用来保护电报汇款的测试密钥是类似的，加密传输被应用到消息上以产生短认证密码，也就是所谓的认证码或者标记。由于这些密钥只被使用了一次，认证码就能被做成无条件的安全。它们为认证所做的事情就同一次一密乱码本为保密性所做的事情一样。

重新阅读第 5 章“密码学”，可以知道当由一次一密乱码本提供的绝对安全与攻击方可获得的计算机资源相独立时，计算机安全系统可能被某个已知的计算所破坏，而且取决于其不可行的程度。

不过在认证码与一次一密乱码本之间是有区别的。由于认证码的长度有限，它总是有可能被对手猜出来；而猜测成功的可能性也是不同的，这取决于对手是否正试图猜测一个新的

合法消息（冒充）或者修改一条已经存在的合法消息以获得另外一个合法消息（替代）。

有个例子应该可以清楚地说明这种情况。我们假设一位指挥官批准了部下的一个认证方案，该方案将指令编码成 000～999 之间的一个三位数。指令可能有两个值："攻击俄罗斯"和"攻击朝鲜"。其中一个将被编码为偶数，另外一个被编码为奇数；攻击哪个国家对应哪个值将成为密钥的一部分。消息的真实性将由它被 337 所除得的余数来保证，该余数与密钥第二部分的保密数字是相等的。

现在假设密钥是这样的：

- "攻击俄罗斯"编码为偶数，"攻击朝鲜"编码为奇数。
- 真实的消息被 337 所除得的余数为 12。

因此，"攻击俄罗斯"是 686（或 12），"攻击朝鲜"是 349。

一个控制了指挥官和部下之间的通信信道，并且知道这个方案但不知道密钥的敌人成功冒充指挥官的可能性只有 1/337。然而，一旦他获知一条合法消息（比如说，"攻击俄罗斯"的密钥 12），那么他就能轻而易举地将它加上 337 后变成另一个合法消息密钥。然后（已知他懂得密钥的含义），他就能将导弹发射到另一个国家。如此一来，这种情况下的成功替代攻击的可能性就是 1 了。

由于使用可计算的安全认证，无条件变化可能或不能提供消息保密性：它可能像分组密码那样工作，或者像纯文本消息中的 MAC。类似地，它可能使用或不用仲裁者。有的可能要多个仲裁者，以使它们不必逐个被确信。如果最初的仲裁者错误地支持了受骗的一方，那么遭到他排斥的一方就应该谴责他。

有些方案可能会把无条件变化与计算机安全合并在一起。比如，通过使用传统的加密系统对消息和验证方加密，可为无保密能力的无条件码简单添加计算机安全机密。

从某种意义上来说，认证是双重编码。对于后者，给定一条不正确的消息，我们想有效地找到最贴切的正确消息；对于前者，我们想找到一条正确的消息，使得其他人不可能构建它，除非已经知道了它或者被授权。正由于纠错编码的设计者希望在给定的错误恢复限度内得到长度最短的编码，所以认证码的设计者想把密钥所需的长度尽量缩短以达到给定的可能出现欺骗情况下的界限。

认证技术在民用和军用中稍微有所不同 [703]。更为重要的是，两者的威胁模型是不同的。总的来说，士兵更关心敌人而不是关心叛逆者，而且对于认可并不是那么担心（除了在被迫与别国谈判时，该国可能会否认一项声明密钥已经被叛逆者泄露的消息）。在商业活动中，大多数欺骗都是由内部人员造成的，所以共享控制系统就成为设计认证机制面临的主要问题。

11.4　共享控制系统

从 20 世纪 70 年代后期开始，考虑到苏联针对美国的国家指挥机构（也就是总统及其法定继任）的"斩首行动"可能会使军火库虽然完好无损但是一点用处也没有，核指挥与控制变得更加复杂。还还有人考虑到，超过了一定限度的准备，由于电磁脉冲和其他针对通信的攻击可能带来的影响，关于官方和战地指挥者之间的通信可以被维持的假设是不明智的。此类情况的解决方案是密码数学的另一个分支，被称为秘密共享，它的发展正受到此类应用的激励。具体想法是在面临压力时将备用的控制系统激活，借此政府当权者可以和战地指挥官

联合起来共同允许武器被装备。否则，维持大量武器的详尽的中枢控制将成为无法解决的问题。

有一种简单而直接的方法可以共享控制权：就是两个人各持认证密钥的一半。缺点是假定他们之中的一人接受贿赂，而原始的安全参数仍适用时，我们需要将密钥的长度翻倍。一种更好的方法是给他们每人一个数，把两个数加起来组成密钥。这就是管理银行自动柜员机的密钥的原理。

然而，这可能在指挥应用中仍然显得不够充分，因为没人能够确定操作设备的工作人员会不会不经过讨论或询问就同意发动灾难性世界战争。

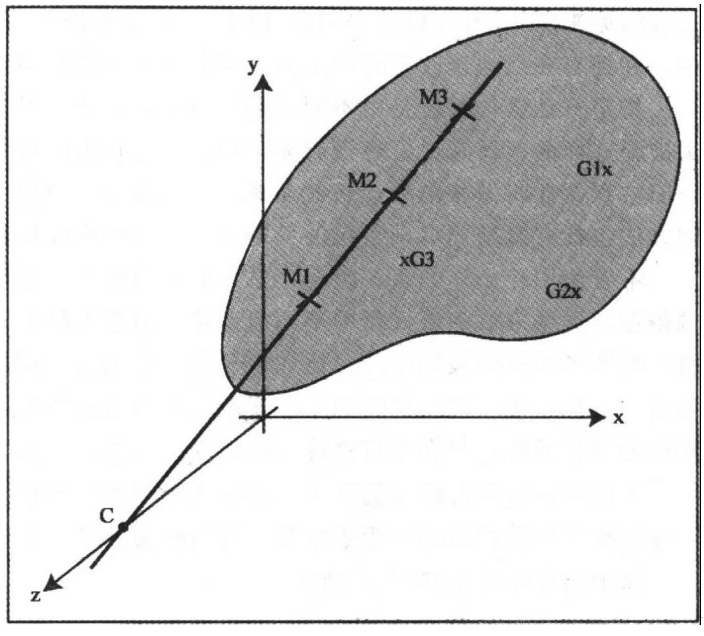


图 11-1　使用几何学进行共享控制

一种更常见的方法是由 Blakley 和 Shamir 在 1979 年分别发明出来的［111，692］。他们的基本思想用图 11-1 来解释。假设英国想制定一条规则，如果首相被刺杀，那么武器是由两个内阁大臣或三个将军装备，还是由一位内阁大臣和两个将军装备。让 z 轴上的 C 点成为必须提供给武器的解锁密码。我们现在过 C 点任意画一条直线，并令每位内阁大臣为直线上的任意一点。现在它们之中任何两个都可以共同计算出直线的坐标位置，并找到交于 z 轴的 C 点。类似地，我们将直线放置在一个任意的平面内并令每位将军为平面上的任意一点。现在任意三位将军，或者两位将军加上一位内阁大臣，都可以重新构建这个平面以及发射代码 C。

将这个简单的结构一般化为 n 位的几何模型，或者一般化为非直线和平面的代数结构，这种技术可以使武器、指挥官和选择权复杂地联系在一起，而只受可获得的带宽限制（关于秘密共享的介绍请见［738］，更多的细节说明在［704］）。秘密共享也促进了阈值签名方案的发展，我曾在第 5 章中讲述过这个内容，该方案也能用于制定法律规则，比如“公司的任何两个副董事长都可以签署支票。”

关于认证码，在民用系统和军队对共享秘密的观点有一点不同。在典型的军事应用中，n 分之二控制被采用；n 必须足够大以使至少两个密钥持有者能做好准备并能履行职责，即使战斗最终失败。许多细节需要注意。例如，指挥官的死不应该使得他的接班人可以使用密钥的两个部分。所以根据规定，它们必须在彼此相隔几码远的控制台同时使用。

然而在许多的民用系统中，许多内部人员会合谋起来破坏系统。典型的例子是收费电视系统，盗版者可能会买来好几打用户订购卡，并反向研究这些卡以获得其中的秘密。显然，收费电视的操作者想得到的是一个稳固强健的系统，以应付多个被泄露的用户订购卡。

11.5　防篡改与指定行动链接

在现代武器中，螺线管安全锁已经被指定行动链接（prescribed action link，PAL）所替代且已用来保护大部分美国核设施（有关指定行动链接的开放源代码信息汇总于［92］）。指定行动链接技术大约从1961年开始发展，但发展速度较为缓慢。甚至过了20年后，美国在欧洲的核弹头大约有一半仍旧在使用四位数的密码锁。随着引入更多复杂的触发选项，密码的长度由原来的4～6位增至最后的12位。装置开始使用多个密码，有单独的启用和授权命令，并拥有在战地修改密码的能力（可能是为了恢复错误的警报）。

指定行动链接系统是由不同的编码交换系统和可操作的步骤提供补充的；对于类似原子爆破弹这样的武器来说，还没有复杂到使指定行动链接难以访问装置核心部分，所以该武器依旧被存储在叫做指定行动保护系统（PAPS）的干扰检测容器中。其他用来防止意外爆炸的方法包括有意削弱爆破系统的关键部分，以使它们暴露于某些反常环境的时候失效。

不管使用什么样的系统组合，总有惩罚措施不让那些想从偷来的武器中获得核物质的窃贼得逞。这些方法随着武器的不同而变化，但都提供可以毁坏含有钚的陷阱和氢化物的瓦斯瓶；破坏各部件的聚能炸药，比如中子发射器和氚加速器；以及能够导致钚的分散而不是聚集的不对称爆破。破坏密码总是被置于优先考虑的位置。人们认为准备培养恐怖分子偷取大量炸弹的反动政府将会做好牺牲一些炸弹（和技术人员）以获得一件可用武器的准备。

要对核设施进行授权的维护，必须使干扰保护措施无效，而这需要一个独立的解锁码。持有各种不同的解锁码的用于维修和发射的设备已经类似武器那样被保护起来。

保护的目标在［734］总结为：

> 现在人们相信，即使有人拥有这样的武器，拿着一套图纸，并享受着某个国家实验室的技术成果，但就是因为不知道密码而不能成功引爆炸弹。

实现这样雄心勃勃的目标需要非常坚实的努力。下面是一些属于该层次的需要注意的例子：

- 在测试显示一毫米芯片碎片可以从携带有制空导弹发射命令的控制设备的保护性爆破实施中存留下来之后，软件被重写使所有的密钥材料都以两个分离的部分存储起来，这两部分被存储在芯片表面相距超过一毫米的两个地方。
- “足球”，这个在总统身后运转的指挥装置，据说厚到不必担心聚能炸药会使保护方法失效（这有可能是一个都市神话）。聚能炸药能够产生速度为8000米/秒的等离子喷射流，这在理论上可以用来使干扰检测电路失效。所以就需要留出一些余地让警报电路有足够的时间将密码存储器清零。

这些注意事项必须延伸到具体实施和操作的细节中去。测试武器的进程不只包含独立的核实和确认，敌方也通过他们的实验室和特工尝试着攻击系统安全。尽管这样，所有被采用的实际措施都是为了防止可能敌人的访问。设备（弹药和控制）被武装部队严密地防护着；时常会出现无人注意的质疑检查；而工作人员必须做好准备随时参加相关检查。

在下面一章中将会讨论更多防篡改的细节，因为它正广泛地涉足到各类应用，从收费电视到银行卡。不过，抗干扰、秘密共享和一次性认证不是得益于核工业领域投入的全部技术，还有更多用于其他领域的精妙的方法经验。

11.6　条约验证

各式各样的验证系统都是用来监视是否履行了核不扩散条约。例如，国际原子能机构（IAEA）和美国核管理委员会（NRC）监视着经过认证许可的民用电力反应堆和其他核设施。

一个例子来自用来监控全面禁止核试验条约（Comprehensive Test Ban Treary [701]）的抗干扰地震传感设备 [701]。目的是在每个不扩散条约签署国的被测点安放足够灵敏的传感设备，以使任何违反条约的举动（比如进行足够强度的核试验）在大多数情况下都能被检测到。这里的干扰检测是比较直接的：地震传感设备被安放在一个钢管内，并被插入到一个周围用水泥浇筑的钻孔中。整个装置十分坚固，可以依靠地震传感设备本身较高的准确率来检测干扰的存在。这种物理保护依靠随机的质疑检查来得到加强。

因为必须作出普遍存在欺骗行为的假设，认证过程会变得更为复杂。由于缺少被双方同时信赖的第三方，并且传输的地震数据量达到每天 10^8 位，数字签名方案（RSA）取代了一次性认证标记。但这只是答案的一部分。比如说某方不承认一条签署的消息，宣称负责生成密钥的官员叛变因此伪造了签名。所以一旦地震数据包被双方同时密封起来，密钥必须经过数据包本身来产生。如果一方建造了设备，另一方就会怀疑它是否含有隐藏的功能。有几个协议被提议用于切断—选择变化，因此，一方可以制造几台设备，而另一方可以拆开其中的一台作为分析样本。一些类似的问题又在电子商务中重新崭露头角（许多电子商务系统建造者都将更多的注意力放在 [701] 所讲述的经验教训上）。

11.7　哪里出了问题

尽管大笔的资金都投入到保护机制的高新技术研发中，核控制与安全系统似乎仍被相同类型的设计缺陷、实现错误、不细心的操作以及其他一些因素所困扰。

最近，位于 Sellafield，处理吨以上量级钚的英国主要废弃物再处理厂被一系列的丑闻和诽谤所困扰。有关废弃物的文件被伪造，辐射泄漏被掩盖起来，工人们的通道被更改，这样他们就可以开车进入禁区；而且还有怠工的报告。核警备力量只能破获 17/158 的窃贼和3/20 的犯罪破坏案件 [495]。现在看来好像核设备将被关闭与消费者失去信心有着密切关系。前苏联的局势就显得更糟糕了。最近的一份核安全调查描述了苏联解体后的十年内废弃的安全机制是如何变化的，裂变材料为何偶尔会出现在黑市上，以及告密者为何会被指控 [401]。

还有一些关于通信和其他被攻击系统的可靠性的问题。怎样确保总统与全球诸多站点的通信？我想在第 16 章“电子战与信息战”中讨论这些问题。

还有一些有关的高科技安全措施失效的情况。其中一例是导致密码数学的新分支——与后面关于版权标记和隐写术的讨论有关的潜在通道研究——发展的可能攻击。

记 [707] 中讲述了开创潜在通道研究的故事。在卡特执政期前，美国与苏联提出一项秘密契约，双方在该密约下能够相互合作来验证洲际弹道导弹的数量。同时，为了保护美国民兵导弹不受可能来自苏联的先发攻击，建议将 100 枚导弹用巨型卡车装载在有 1000 架发射台的基地中移动，设计这些卡车使得观察者无法判断他们是否在搬运导弹。苏联必须毁掉所有这 1000 架发射台才能成功完成先发攻击；这在提到的装备控制中是不可行的。

这样就产生了一个有趣的问题，怎样才能使苏联确信在发射基地至多只有 100 枚导弹，而且还不让他们发现哪些发射台真的装备了导弹。所提出的解决方案是在发射台安装上能够

检测到导弹是否存在的俄方传感器件，用一位信息码标记，并通过美国监视设备发送至莫斯科。传感器能够被包装起来并在安放前被美国人随机地移动位置，这样就使俄国人无法将“有”或“无”信号与特定的发射台关联起来。关键问题是只有这个一位信息码可以被发送；如果俄国人可以在发送消息中偷偷加入更多的信息，他们就可以快速地将每个发射台定位——因为每定位一个基地的发射台只需要十位地址信息（有许多其他的安全要求用于防止任一方的欺骗行为，或错误地控诉对方的欺骗行为；更多的细节请见［706］）。

要理解潜在通道如何工作，可以考虑第5章中谈到的数字签名算法。系统全局数值为质数 p，一个160位的质数 q 除以 $p-1$，和级为 q 的亚群 F_p^* 的生成元素 g。消息M的签名是 r、s，其中 $r=(g^k \pmod{p}) \pmod{q}$，$k$ 为随机会话密钥。从 k 到 r 的映射是比较随机的，所以希望在签名中隐藏10位信息来秘密发送给同谋者的签署人首先需要在如何隐藏码位上达成一致（比如第72～81位），然后试着寻找 k 的数值，直到结果 r 的数值与达成一致所需的数值相等。

这可能引起安全协议的灾难性失败，因为在美俄达成的一致协议中监视消息会首先经过俄国的设备并以俄国的方案认证，然后再经过美国的设备以美国的方案认证。如果俄国人使用像数字签名算法的签名方案，他们就会将配有导弹的发射台的位置识别出来，并拥有先发攻击民兵式导弹力量的能力。

最后，由于在大众新闻中普遍知晓的“导弹竞赛”不再使用。1980年大选后两国关系降温并停滞下来。最终，有关中程弹道导弹条约的统计方法被使用。俄国人会说，“我们愿意看到之后的20架发射台”，而他们可以明显地用他们的卫星看到。冷战过后，随着载有双方观察者的有人驾驶侦察机胜过了卫星，侦查变得更为密切。

尽管如此，潜在通道的发现仍十分重要。但由于该项技术将患有HIV病毒，或者是否被判重刑的情况记录到下一代数字身份证中而遭受谴责。这种做法是不可接受的，但证件发行者如果不这样做也无法被充分相信，补偿方法是使用完全确定的签名方案（如RSA）替代像数字签名算法（DSA）一样使用随机会话密钥的方案。

11.8　保密还是公开

最后，核工业还提供了一个很好的秘密发展历史案例。20世纪30年代，许多国家的物理学者都在自由地享受着炸弹研发的科技进展；但当后来“原子间谍”（Fuchs、Rosenbergs和其他人）将投向广岛和长崎的炸弹设计方案透露给苏联时，事情就转向另一个极端。美国采纳了原子知识要进行*初始分级*的政策。这意味着如果你偏向美国一方并有一个关于核武器的想法，你必须将其保密，不管你是否有秘密文件或者就在核工业领域工作。在宪法的压力下这一点十分清晰。从那以后事情就变得大为轻松了，因为相关保护事项被考虑得很周全。

“我们在新墨西哥州有一个数据库，记载了钚在很高温度和压力下的物理和化学属性”，一位前美国核安全领导告诉我。“我应该以什么层次进行分级？谁打算窃取它，这样做是否有好处？俄国人，他们为自己获取数据。以色列人能弄明白。卡扎菲？他到底拿它做什么？”

随着这类事情的发生，数量惊人的技术不再被保密并被公布开来，至少是其技术概况。从20世纪80年代初科学会议关于认证密码和潜在通道研发结果的早期公布来看，从公开的设计回顾中获得的收益要远胜过对手全面了解正在使用中的系统所获得的可能优点。

不过许多技术操作细节是保密的；那些可能造成破坏的信息，比如一个设施的50幢建

筑中含有警报响应能力的信息，将被标记为不保密受控核信息（UCNI），这就给安全策略模型又加上了一层复杂度。也有相当多的事实真相是需要保密的，比如谁有权杀死谁（同一方）和在什么情况下。

尽管如此，此类信息总体上是公开的（我们确实这么认为）；而且在对最近的分级评论之前，指挥与控制技术被公开地提供给其他国家，包括像前苏联这样的敌对国家。需要再次说明的是，减小突发战争的可能性所带来的好处被认为胜过保密带来的好处。这是最早在19 世纪出现的 Kerckhoff 主义在当代的翻版，系统的安全应该依靠它的密钥，而不是靠保持它的设计隐秘性 [454]。

11.9　小结

核武器的指挥与控制，以及相关辅助行为——从通过对核设施的物理安全保护国家指挥系统的完整性到监视国际装备控制条约——已经为安全技术的发展做出了无法比拟的贡献。

相关财产必须受到保护而与花费无关这样一个相当合理的决定驱动了许多数学学科和在其他领域得到应用的科学技术的发展。本章给出的有针对性的例子是认证码、共享控制方案和潜在通道。我们也开始讨论抗干扰设备，后面我们还有更多相关的内容。

研究问题

找到该领域所用技术的有趣应用，比如认证密码。

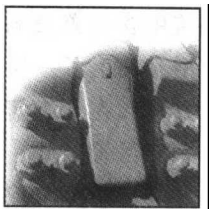
参考资料

Simmons 是认证码、共享控制方案和潜在通道的开拓先锋，他的书 [703] 一直是本章中讨论的大部分技术材料的最佳参考。更多认证和秘密共享的简明介绍在 [738]。

有关核武器的最好的公共信息公开资源之一是美国科学家联盟 [286]。最新公开的核装备技术基本原理细节在他们的网站上可以找到 [286]。公开问题在 [812] 讨论，指定行动链接的可获得公开材料被列在 Steve Bellovin [92] 文章中。

核装置的控制失败问题在文献中被广泛讨论。俄国核装置的问题在 [401] 讨论；美国核安全由核常规委员会监督 [593]；英国的核装置缺点在健康与安全执行委员会发表的季刊报告上有记载 [375]。

第 12 章　安全印刷和印章



印章发挥的效用相当于携带它的那个人。

——Karen Sparck Jones

12.1　引言

许多计算机系统在某种程度上依赖于安全印刷、包装和印章技术来保障其安全防护的重要方面。

- 许多软件产品采用诸如全息标签的技术来防止伪造。当全息标签从包装上去掉时，就可以认为它已被撕掉。这些技术可以提高大规模伪造的代价；另一方面，对于个人而言，认真地实施这些技术有助于可信分发。也就是说，可以向用户保证产品在离开工厂之后没有被篡改。
- 我们讨论如何监视系统，就像计程器，通常使用的是印章技术，使得用户很难用输入来进行篡改。但是，无论密码系统多么的先进，战胜了印章就等于击败了系统。
- 许多安全令牌，例如智能卡，很难真正做到防篡改。对手很容易拆开设备并探查出其中的内容。这样一个系统的真正目标可能是证明篡改，而不是防止篡改。如果有人拆开智能卡并取走密码，那个人不应该能够把它重新封装好，且可以通过仔细的检查。在此，安全印刷是关键技术。如果某银行智能卡确实被篡改了，那么银行就可能会告诉它的客户，它随时愿意奉陪各种争端，除非客户可以制作出完好无损的卡来（尽管如此，银行也不可能逃脱责任，因为客户的辩护律师会要求银行公平对待那些丢失卡和被偷走卡的忠实客户）。

与在计算机系统中直接应用印刷和印章技术截然不同的是，利用现代彩色扫描仪和打印机来制作能够通过各种检查的伪造品已经非常容易，从而实际上已经打开了另外一个防伪领域。钞票印刷机正在促进数字保护技术的发展［109］。这包括不可见的版权标记，它可以使伪造物被检测出，甚至在图像处理软件中就发出警报［357］。数字世界与“搞笑墨水”世界正飞快地关联到一起。

12.2　历史

印章有着悠久且非常有趣的历史。在有关银行系统的章节中，本书曾解释过记账系统的起源为泥制的标牌或者泥土标记。在 Mesopotamia 地区，它们被新石器时代的仓库管理员当作产品的收据。五千多年前，封印系统就已被用来解决争端。具体是通过仓库管理员在泥制的封印上面烤制上自己的标记。

在古希腊、古罗马以及古代中国，印章广泛用于文件的认证。但在欧洲，直到几百年前，它才被用于信件方面。即使在签名已取代印章成为主要的认证机制之后，印章技术仍被

当作重要的备用机制，直至 19 世纪。信件并不是放在信封里，而是折叠几次后，用热蜡和环形图章封好。

在中国、日本和韩国，印章仍被当作重要文件的首选认证机制。在其他地方，印章的重要性体现在公司印章和公证人印章之中。这些印章依附于重要的文件以及一些国家的领导人用在立法文件拷贝上的国家印章。

然而，到了上世纪中叶，印章在西方用于文件已不如用于认证、包装那么重要了。散装货物向包装货物的转变，以及商标的日益重要，不仅产生了更多质量控制的潜在需求，而且导致了居心叵测的人篡改产品的可能性。美国曾深受产品篡改之害，尤其是在软饮料和医药产品方面。在 1993 年达到高峰，共有 235 件案例被报道 [445]。这促使了制造商努力提高其产品的防伪性能。

由于软件容易被复制，加上从 20 世纪 80 年代起，用户对技术性的拷贝保护机制的抵制，使得软件公司越来越依靠包装来抵制假冒。这仅是防止高价值和名牌商品被伪造的大市场中的一部分。这些商品涉及香料、香烟、航空备用件和药品。

简而言之，在印章和其他各种安全包装的行业里已投入了大量资金。但是，大多数的印章防护还是很容易被攻破。

典型的印章包括具有安全印刷技术的基片，它被粘贴或系在密封物体上，所以我们必须首先查看安全印刷。如果整个印章能被轻易地伪造，那么任何数量的粘贴或拴系都没有用处。

12.3　安全印刷

自从 19 世纪早期拿破仑将纸币引入欧洲，以及其他有价文档（例如见票即付债券和护照）的出现，有关安全印刷机和造伪币者的争执就开始了。它们呈现出捕食者和猎物之间共同进化的特征。如果说照相术（1839）帮助了攻击者，那么彩色印刷和钢蚀刻技术（1850）就是抵御者。近年来，彩色复印机和便宜的扫描仪已经遇到了全息技术和其他光学可变设备的抵制。有时候，同一个人会涉及两方，例如一个政府的情报机关设法伪造另一个政府护照的情形（在某种情况下，伪造的对象甚至是货币，例如二战时的双方）。

有时，钞票的设计者会屈服于泰坦尼克效应，即过于相信先进的技术和某些特殊的技巧。一个很好的例子就是 20 世纪 90 年代对英国钞票的伪造。这些钞票具有窗口线——穿过纸张宽约 1 毫米、每隔 8 毫米一条的金属条带。当你用反射光观看时，这些钞票显示出横穿其上的虚金属条带；当你把钞票举起来，并用投射光观看时，金属条带就变成暗的和实心的。复印这些钞票被认为非常困难。但一个犯罪团体发明了一种很漂亮的破解方法。他们通过便宜的热压印过程在纸的表面上放置一个金属条，然后用白墨打印一个实心的条带，以让期望得到的金属样式可见。在审问他们时发现，他们在几年的时间里已经伪造了价值数千万英镑的钞票 [299]（这里可能有一些机构自满的问题，因为欧洲的银行认为造假者只喜欢伪造有三种颜色的美国纸币）。

12.3.1　威胁模型

通常，我们必须在一个威胁模型的环境中评估一项保护技术。一般地说，威胁可能来自于一个受正当资助的组织（比如一个设法伪造别国钞票的政府），它的规模从一个中等规模的组织（例如一个月伪造几百万美元的犯罪集团，或者一个散发伪造葡萄酒标签的组织）到

使用家用或办公室设备的非专业人员。

在印钞业里，20世纪最后的几年里增长最大的领域是非专业伪造。关于如何进行高质量钞票伪造的知识在印刷行业中大量传播。人们可能认为这只会增长专业伪造的水平。但高质量彩色扫描仪和彩色打印机的广泛传播，使许多在以前需要用脏、湿的油墨时期从未想过伪造的人受到了很大的诱惑。过去，非专业人员被视为一些小麻烦，但自从1997或1998年，他们已占据了在美国发现的各种伪造中的绝大部分（他们因国家而异；大多数的英国伪造者利用传统的石版印刷，而在西班牙和美国，采用的是喷墨印刷机[393]）。非专业的伪造者很难对付，因为他们人数众多，且大多规模很小。所以他们的产品需要很长的一段时间才会引起官方的注意，并且他们中很少有犯罪记录。他们制造的纸币通常质量不好，不足以在银行出纳员那里通过。但是在昏暗和嘈杂的夜总会，这些纸币却能流通。

业界区分出三种不同的检查级别，伪造的钞票或文档可能通过也可能通不过这些级别[765]。

> 初级或第一层次的检查由未受训练、经验不足的人执行，例如普通大众当中的一员，或一个商场里的新出纳员。通常，初级检查员没有验钞动力，甚至是消极的验钞动力。如果他得到一张摸起来稍微有点不对劲的钞票，他可能不经仔细看就让它通过检查，否则的话，他需要在成为一个帮凶或者赶快去报告之间做出选择。
>
> 第二级或第二层次的检查由有能力、有动力的人员实地执行。例如在钞票方面具有丰富经验的银行鉴别人员、在产品标签方面训练有素的制造厂商检查员。这类人可能拥有一些特殊的设备，诸如紫外线灯、含化学试剂的笔，甚至是一台扫描仪或PC机。然而，这些设备在成本和大小方面有很多的限制，并且也还是会被技艺高超的伪造者弄明白其技术细节。
>
> 终级或第三层次的检查在制造商或发行纸币的银行的实验室里进行。设计安全印刷的专家（甚至可能是一些底层的工业处理方法）将出现在现场，并提供很多设备和技术支持。

安全印刷技术当前状态的一个可行性总结是：一个伪造品通过初级检查是比较容易的，而要通过终级检查，通常就是不可能的（如果检查过程被很好地设计的话）。这样，二级检查是防伪的主战场（除了诸如钞票印刷等应用需要将注意力放在初级检查上）；现场的检查员能发现出何种伪造品的主要限制与用到的设备的大小和成本有关。

12.3.2　安全印刷技术

传统的安全文件利用了一些印刷方法，包括：

- *凹版印刷*，一种雕刻样式用强力将油墨压在纸上的方法，它在纸上留下一个高清晰度的墨印。这通常用于钞票和护照上的涡形装饰。
- *凸版印刷*，将油墨滚在突起的铅字上，随后压印在纸上，并留下压印。钞票上的号码通常以这种方式印制。这些号码大小不同，并且使用不同的油墨印制，从而防止可以买到的打号设备被利用。
- 独特的压印方法，称为同步印刷。它在纸的正面和背面同时传送油墨。这意味着正反面的印刷可以准确地定位。图案可以部分印在正面，部分印在背面。这样，当钞

票迎着光举起来时（穿透对准），这些图案可以很好地匹配。对于廉价的彩色打印设备来说，要想复制这种钞票是很困难的。同步印刷还有特殊的通道，可以使油墨的颜色随着线变化（即彩虹效果）。

- 用于签署文件的橡皮图章，或者在文件上盖印相片。
- 浮雕压纹和分层层压，用于盖印相片，以及银行卡。它能提高伪造的成本。浮雕压纹可以是物理的，或者用激光雕刻技术将一张相片烧制到 ID 卡中。
- 水印，是将防护特征放置于纸中的一个实例。它是插入纸中的一些半透明区域，在制造时通过改变其厚度得到。还有许多其他的特殊性质正在使用，例如荧光线。一个极端的例子就是澳大利亚的 10 元纸币。它在塑料上印刷，并有一个透视窗。

更加现代的技术包括：

- 光学可变油墨，如 20 加元上的贴片。它的颜色根据视角的不同，由绿变到金黄。
- 具有磁性和声光特性的油墨。
- 只能利用特殊的设备才可见的印刷特征，比如美元上的微缩印刷特征，需要用放大镜才能看到；以及利用紫外线、红外线或磁性油墨进行印刷的技术（其中最后一种用于美元上的黑色印刷）。
- 金属线和金箔，从简单的虹彩特征到拷贝于箔片上的色彩。这些色彩具有光学可变的效果，例如激光全息和精细动态全息。这些特征可在英国的 20 英镑和 50 英镑的钞票中找到。全息是典型的利用光学特征制造出来的产品，看起来像胶片后面的实体物。而精细动态全息是用计算机制造的，能在观看角度稍微变化时显示出许多令人惊讶的效果。
- Screen traps（屏幕陷阱），例如细节太微弱而不能被正确扫描；以及 alias band（伪信号波段）结构，它包含正确尺寸的细节，以组成干扰效果，利用的是普通扫描仪和复印机的点分离技术。
- 数字版权标识，从直接利用微型印刷的傅立叶变换来隐藏信息的图像，到扩展频谱信号，它能被彩色复印机、扫描仪或打印机识别出来，从而使机器停止工作。
- 独特的 stock（原料），例如在制造过程中具有磁性纤维随机伸展特征的纸张，这样，每一张纸上都有特殊的模式，它可以用于数字签名，或利用某种条形码标记或印刷在文档上面。

对于新的 100 美元纸币的设计，见 [566]；对于假币的研究，以及对何种特征提供何种证据的分析，见 [766]。通常，只检查一种安全特征很难判断钞票的真假。许多老技术和一些新技术，能被仿造出来并通过初级检查。凹版和凸版印刷的触觉效果会磨损，因此将假币弄皱和弄脏是造假者的一种标准行为。并且，有经验的假币制造者能利用微弱灰度印刷仿造出水印（尽管水印对于业余人员有着极好的效果）。对于利用电化学技术进行机械拷贝的人来说，全息技术和精细动态全息技术也是容易受到攻击的；或者罪犯会从零开始制作他们自己的母版。

1988 年，当画面为莎士比亚的全息图被引入英国的支票保证卡时，我作为一家银行的代表参观了这家工厂。工厂的人员骄傲地告诉我，出于这个行业对备用供应源的需要，他们已经为一家大型安全印刷公司提供了少量套数的金属图版。而他们的竞争对手根本不可能制造出可以被接受的金箔（莎士比亚金箔是第一个商业应用的衍射全息图，当视角改变时，会

出现全色或画面移动的效果)。这样，连可以接触到真正的印刷板的大型安全印刷公司也不能伪造的设备，一定能提供全面的保护吗？但是当我七年后在新加坡参观时，我在“跳蚤”市场买了一个相似的（但相对大一点的）莎士比亚金箔。其制作者说如果他想的话，他可以仿造出英国的银行卡。这显然是在夸口。但那时，警界的专家估计，在亚洲有超过一百个以上的伪造者拥有生产可获通过的全息图的技术［591］。

技术在不断地进步，而帮助罪犯的技术进步可能来自难以预料的方向，因此技术控制的效果甚微。例如，离子光束工作站——可用于生产精细动态全息母版的机器——在20世纪90年代中期，价值高达数百万美元。但由于它被证明在冶金实验室的工作中极为有用，因而销售量猛增，价格也因此急速下降。现在有很多部门出租该类机器，一小时仅数百美元。所以只依赖于一种保护技术是不明智的。即使一项防御技术被彻底击败（例如，假设制作金箔的机械拷贝已经变得很容易），你至少还可以求助于另外一种完全不同的技术（如光学可变油墨)。

但是设计安全文档要比这难得多。在保护性能、美学和健壮性方面有着复杂的权衡。并且，人们正在意识到，多年以来，设计者将他们的目标集中于防止伪造者通过第二级和第三级审查（技术焦点)，而不是集中在更普通的初级审查（商业焦点)。很多时间花费在编写有关训练人们正确检查文档的难度方面，而没有将足够的注意力放在研究诸如钞票的典型用户如何在潜意识里决定其是否可接受方面。这个缺陷正在受到很大的关注。

最近可吸取的教训有［765］:

- 安全特征应该传递与产品有关的信息。所以最好用虹彩油墨来在钞票上印刷面额，而不是一些模糊的特征。
- 它们应该明显地属于其所在的地方，这样它们才可以嵌入用户关于目标的认知模型中。
- 它们的效果应该是明显的、清晰的和易懂的。
- 它们没有现存的、能够提供模仿基础的竞争者。
- 它们应该是标准化的。

这项工作值得更广泛的关注，因为纸币界是对安全可用性已投入大量考虑的行业中的几个分支之一。(在本书后面的第23章中我们将看到，现行安全产品评估模式的一个主要失败之处就是忽视了可用性)。当应用于文档，例如护照，而不是纸币时，也会有与它们将被使用的国家的政治环境、社会道德有关的话题出现［546］。

可用性在二级审查时也有问题，但是此处的这个话题更为微妙，集中考查审查者为分辨真假而必须遵守的步骤。

对于纸币，这个理论就是你设计了一种纸币，它可能具有20种没有告诉公众的特征。许多特征告诉了二级审查员，如银行职员。不久以后，这些特征就被造假者所知。随着时间的推移，越来越多的特征被暴露出来。最终，当所有的特征都被曝光之后，这种纸币就退出流通，并被其他纸币取代。当重点从手工验证向自动验证转换时，情况会变得更困难。当小偷盗走了自动贩卖机之后，就会拆开机器，并读出软件，获取现行审核方法的完全细节。一旦花上数周或数月来做此事，他会发现对此进行伪造欺诈更加容易。所以当中央银行告诉制造商二级数字水印（或其他任何保密特征）的秘密多项式，并被送到现场应用后，小偷就可以偷走一台机器并在数天后取得新资料。相对于手工系统，机器系统的失败要来得更突然、

更彻底。并且发现安全秘密的周期也比过去要快得多。

至于产品包装，典型的商业模式是伪造的样品被发现并被送到实验室，在这里科学家找出伪造品与真品之间的差异之处——例如伪造品的全息图不是非常准确。然后一些工具被制造出来，并提供给现场的检查人员，以便寻找和追踪伪造品的来源。如果这些工具体积庞大且价格昂贵，那么就很少能被实际应用。如果有不同的公司生产不同的用于鉴别伪造品的设备，那么就很难说服海关官员去使用它们中的任何一种。比如印刷在塑料产品收缩包装上的惟一显微紫外条形码的计划通常不成功，因为执行验证所需的显微镜、笔记本电脑和在线连接的花费很高。至于纸币，可以利用多种特征来实现更强壮的系统，但这又大大提高了读取设备的开销和体积。现在大量的研究精力放在努力发展惟一的标识特征上。例如特殊的包含蛋白质甚至 DNA 分子的化学涂层，它对隐藏的序列号进行编码，可使一种类型的验证设备能检查许多不同的产品。

对于金融设施，尤其是支票，改造是一个比从头复制或伪造更大的问题。在许多欺诈行为中，罪犯们从交易中获取真支票，采用诸如预付押金、用现金预约，然后删除这个号码的方法。受害者按时送出一张支票，但支票被罪犯用容易得到的家用溶剂更改成更大的数目。典型的应对措施是利用在溶剂中会消色、消失的油墨进行背景印刷。但这种保护措施并不彻底，因为有消除激光打印机调色剂的手段（还有更简单的工具，如打印机校正色带）。一个大胆的罪犯甚至给受害者提供经过特殊选择的笔，而该笔用的是容易擦除的油墨 [5]。

安全方面的文献讲了很多有关借贷卡欺诈的事件（例如技术专家对 ATM 使用的加密系统很感兴趣），少量有关信用卡欺诈的事件（在网上有很多关于信用卡欺诈的讨论），以及更少的有关支票欺诈的事件。然而，支票欺诈案很多时候在金额上要比信用卡欺诈案大得多，而借贷卡欺诈案相对于那两种欺诈案来说就几乎无足轻重了。尽管支票欺诈极为重要，但研究者普遍认为它太令人厌烦了。

银行的实际问题是每天要处理大量的支票。这样仔细检查就变得不太可能，除非涉及金额非常大。业余的支票欺骗者骗得的金钱总额以受害者组织的标准来衡量数额不大（上千或上万英镑）。在远东，人们用私人印章或签名图章在支票上签名，而不是手写签名。这使低成本的自动检查成为可能 [395]。然而，对于手写签名，具有可接受错误率的自动验证仍是不现实的。本书将在 13.2 节对这个问题进行讨论。在某些国家，比如德国，通过银行转账而不是支票来进行大多数的商业支付，从而使支票欺诈案受到大大的抑制（即使对于小规模的顾客基金）。做出这样的一个改变，需要克服巨大的文化惯性。但是在线支付的低成本（几分钱或几十分钱）可能会劝服大多数国家的商业最终做出那样的转变。

改动对于典型的银行信用卡部门来说也是一个很大的问题。更改卡上的磁条比重新生成全息图要简单。实际上，在 20 世纪 80 年代早期，系统利用在线终端验证一张卡的磁条信息，然后利用 zip-zap 机器收集实际的交易数据。结果是授权依据磁条的卡号进行，而交易依据压纹上的卡号进行登记。因此，罪犯会用偷来的卡，并利用具有高信用限额的持卡人账户细节对卡进行重编码——例如，从高级饭店外面的垃圾箱中的废交易单中获取——并利用它们进行授权交易，这些交易随后给被盗的账户开账单。由于授权号与记录的账户号码不匹配，银行会取消这次交易。这样银行开始与其客户就债务进行争论，并且改变系统，最终汇票能从磁条中以电子方式获取。

当然，改动也不只是银行业的问题。大多数假旅行文件也是被改造的，而不是从头开始

伪造。改造的方式有：改变姓名、置换相片或增减页面。

最后，一个很有前途的技术是利用光学可读的数字签名，而不是传统的序列号。这可以将印刷物质粘合到下面的基片上，或与附上物质的信息进行绑定。当本书在5.3.5节介绍数字签名时，曾提到美国和其他一些国家正在引入一个新的邮资计费系统，它打印出具有二维条形码的邮票，称为邮戳。邮戳包括邮资、寄信人姓名、收信人的邮政码、邮资计费的序列号以及日期。尽管如此，一张邮票在理论上还是可以从信封上揭下来并粘贴到另外一个信封上——或只进行光学复制——这种安排足以阻止美国邮政局最关注的欺诈行为，它涉及到废品邮寄人贿赂邮政系统的职员，以将大宗的邮件引入到系统中来［753］。所介绍邮戳的样品复制在图12-1中。

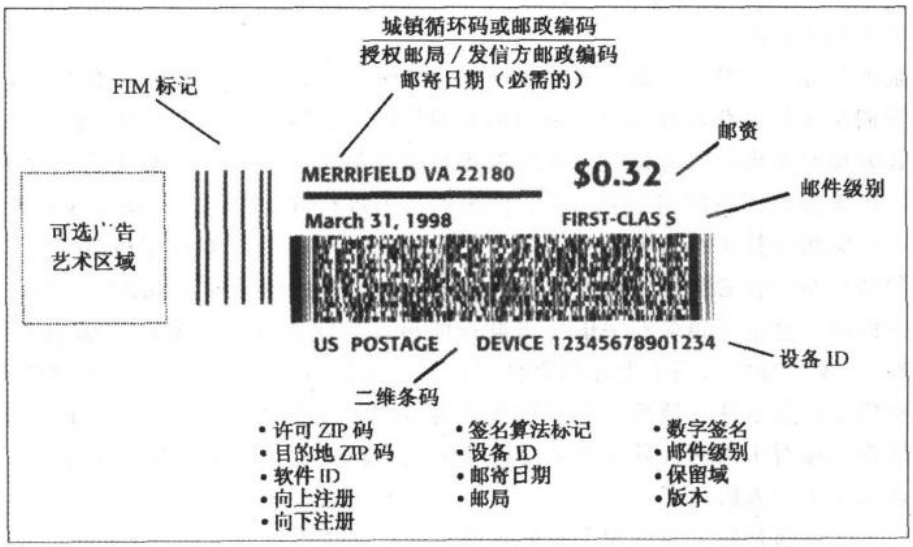


图 12-1　美国邮资计费的新形式（照片经 Symbol Technologies 公司许可）

12.4　包装和印章

这给我们带来了额外的有关包装和印章的问题。

不是所有的印章都采用在被封印的物体上粘贴具有安全印刷的基片的工作原理。本书曾提及用金属线和铅印来防止篡改卡车传感器。还有许多其他的产品采用同样的基本原理，但使用不同的材料。比如塑料带，容易扎紧，但若不割断，却很难解开。本书也曾提及过特殊的化学镀层、微型的条形码，以及其他的一些技巧，用来追踪产品或成批的产品。然而，绝大多数使用的印章是先在一基片上应用某种安全印刷，然后粘贴在要保护的材料上。

12.4.1　基片特性

一些系统在基片材料上附加随机的可变性。回忆在纸上添加磁性纤维的方法；同样，也有水印磁性技术。它将随机的高矫顽磁性信号嵌入到卡条中。随后，这个卡条就可以用标准的低矫顽磁性信号设备进行读写，而独特的随机样式不会被干扰。水印磁性技术应用于瑞典的银行卡、韩国的电话卡，以及我所在大学中某些建筑的门禁通行卡。

同样的想法也被应用在武器控制中。许多材料有着独特的表面，或者用少量炸药填充使材料磨损而达到同样的效果。这使得对诸如重型枪炮等重要装备的鉴别非常容易。对每个枪管或炮管进行标识足以防止双方相互欺骗。枪管、炮管的表面样式可以利用激光颗粒技术进行测量，并记录在日志中，或附着于设备上，以作为机器可读的数字签名 [703]。

类似的技术在邮政系统中得到发展。排列成行的网格被印刷到信封上，然后用微型显微镜来观察纸纤维。纸纤维图案被提取出来并记录在邮局的邮戳标识中，这个标识以数字化方式标记。这使普通纸张的惟一可识别变成可能。这与刚才提到的载有纤维的纸相似，只是更加便宜了。

12.4.2　粘贴问题

然而，许多印章技术需要将安全印刷物质粘贴在目标物上。这引起了如何才能将漂亮的虹彩色的印刷品依附于粗糙的物理对象上，并很难被移除掉的问题。通常的解决方法是利用粘性强于印章基片的胶水。这样，当印章被强行移开时，它会被撕裂，或至少变形。

然而，在许多产品中，实施的方法太简单。许多印章仅需手工工具，再加上少量的耐心，就可以很容易地直接移去。你也可以利用锋利的小刀，对下次收到的用自粘信封包装的几封信做一下试验。许多这样的信封被认为是需要撕开的，而不是剥开的。为达到此目的，信封口处可能刻有一些垂直的槽。但是这个希望能证明损坏的技术通常假定人们在打开信封时，是随意将信封口往后拉，从而使之离开信封体。然而，轻轻提起信封口，并用小刀前后切割，就常有可能切开胶，却不损坏信封口。因而不用留下可疑的标记就可以打开信封（某些胶需要先用吹风机加热，从而变软；或降温变脆）。结果可能是使一个信封在仔细检查时看起来稍微有点皱，但这点皱可以被熨平。这个攻击可以通过初级审查，而在终级审查时可能会失效，但却有可能很好地通过第二级审查：因为在邮寄过程中无论如何都会起皱。

市场上的许多印章可以采用同样简单的技术来对付。例如，有一种彩色粘贴胶布，在被撕下来时会留下“危险”或“不可用”的警告。彩色层夹在两层胶之间。并且底层的胶粘性较强。这样，在封印被篡改时就会留下颜色。但这种胶带仅在从上面揭开时会这样，若从旁边切割，则可以完好无损地移开它并重新使用 [479]。

图 12-2　我们当地游泳池的腕带印章

12.5　系统脆弱性

现在我们从针对诸如特殊的印刷技巧、粘贴和市场等的威胁转向系统级威胁。事实上，确实有很多这样的威胁。

一个可能有用的例子如图 12-2 所示。在我们当地的游泳池，繁忙的时候通过给游泳的人分发腕带来控制拥挤。每二十分钟左右用一种不同的颜色进行分发。有时，所有具有特定颜色腕带的人被要求离开。这个带子用蜡纸制造。在带子

的末端，有一面印有特定的图案和序列号，并粘贴在另一面上。如果你不小心扯掉它，纸就会被横切，结果使其被损坏掉（这与用于某些机场的行李封印十分相似）。

最简单的攻击是打电话给供应者；一个盒子装有100个腕带，价值8美元。如果你不想花钱，你可以一次使用一个腕带，然后轮流从不同的方向轻轻地拉扯它，使它放松。最后的结果如图12-2所示。尽管完整无缺，印刷品还是皱了起来。但这种损害不足以被游泳池边的管理员察觉出来，并且也可能是由无心的应用引起的。其关键点是仔细地两次固定印章造成的损害，与一个新手一次固定的效果相比，差异很小（一个更为厉害的攻击是根本不从印章上移去后面的胶带，而是利用其他的方法——安全别针，或你自己的粘胶——来进行固定）。

尽管如此，这种腕带印章仍然可以很好地用于其目的。人们很少有进行欺骗的动机：紧张训练的人们一次只会游两个小时，而且会在人不多的时候来游泳池。他们也可以买月票，因而可以随时去获取一个具有正在使用颜色的腕带。但是它也例证了许多可能出错的事情。事实上，顾客是敌人；正是顾客应用印章；重复使用的印章，其效果与偶然的失误不可区别；未使用的印章在市场上可以买到；假冒的印章可以低价制造；而有效的检查是不现实的（尽管如此，相对于许多用于高价商业应用的印章产品，这个游泳池的印章仍然难以击败）。

12.5.1　威胁模型的特性

我们已经见过顾客就是敌人的系统，比如银行业。在军事系统，敌人可能是一个不忠诚的士兵，或者试图破坏设备的敌方特种部队。在原子能监视系统中，敌人可能是试图将裂变物质从得到许可的民用反应堆中转移出去的国家。

但是最困难的一些印章任务产生于商业中。它们的困难起因于这样一个事实：对手将使用印章。一个典型的应用是一家公司将一些产品的制造转包给其他公司，但又害怕承包方会生产出多于合同规定的产品。从价值上来看，过度生产是全球范围假冒品的主要来源。犯罪者可以使用授权的生产流程和原材料，而黑市则提供分销渠道。甚至是发现那样的欺诈行为——除非能在法庭上证明这种行为——也很困难。

针对诸如化妆品等高价商品的典型解决方案，可能会涉及到从许多不同的公司购买包装材料，它们的特征对于执行最后安装的工厂来说是保密的。其中一些材料可能会以不同的方式嵌入序列号（例如在玻璃瓶上进行激光刻蚀，在玻璃纸上用仅在UV光下可见的油墨进行印刷）。也可能是在线的服务，因此制造商的当地代理商可以验证在商场里任意购买的商品的序列号。或在包装上印有数字签名，它将所有不同的序列号连接起来，以便离线核对。

孤立使用印章有很多局限，有时候品牌的拥有者就是罪犯。正如一个葡萄园主将额外的1000箱葡萄酒标识为最好的，但实际上这些葡萄酒是由购进的混合葡萄酿造成的。所以南非葡萄酒的瓶上，都带有具有惟一序列号、由政府控制的印章。此处，印章并不能证明冒牌，但却能使不诚实的葡萄园主很难逃避诸如检查和审计等其他控制措施。所以印章机制通常在设计时就要考虑审计、测试和检查过程等概念。

检查比人们想像的要更有技巧。由于认为是真的而在黑市里购买非法商品的批发商可能会去欺骗检查人员，却不认为自己有任何犯罪目的。黑市是一个问题，检查人员期望只对批发商仓库中的授权产品进行检查。而从黑市买来的产品被迅速地卖了出去。批发商也可能完全在暗处；或者可能是其职员在叫卖伪造品。在最近一个引起高度关注的案件中，一主要航

线的职员在远东购买香水、手表等物品，然后在飞机上卖给旅客，并从中获益。航空公司仓库中的货物全是真的（在飞机着陆后的免税店内也是这种情况）。所以通常有必要派代理人出去采购样品，并且印章机制也必须支持这一点。

12.5.2　员工的细心

印章是否正确地粘贴在物体上也要依赖于基层职员的忠诚度。本书在 10.4.1.2 小节中提及在卡车的速度限制系统中，变速箱传感器利用一段金属线安全地放置在适当的位置。在校准汽车间，又利用铅封钳卷曲一个铅环于金属线上。对付的方法是贿赂加油站技工，使他以错误的方式缠绕金属线。这样，当传感器被旋松时，金属线将放松，而不是变紧并破坏封印。所以完全没有必要去业余雕刻班学习做一套印章，并伪造一对用青铜做的铅封钳（除非你不想在贿赂上花钱，或者想借此运营一个加油站）。

应用印章的人可能是粗心和腐败的。在过去的几年里，一些机场首先利用靠近登记队列的机器对乘客的箱包进行 X-射线检查，然后在箱包上应用胶带印章。在大约一半的情况下，我的行李都是如此被检查的，但胶带固定得很差：它并没有横穿公文包和盖子之间的接合件，或者是在一端脱落，或者留有一些大到足以装下一个炸弹的隔间，但它们的接合件仅有一个盖上了印章。粗心和贿赂相互作用。如果有足够多的应用印章的员工是粗心的，那么贿赂其中的一个人，就不能自行证明是不诚实造成的错。

12.5.3　随机失败的效果

当印章因为完全无知的原因而被破坏时，效果是相似的。例如，当卡车引擎是用水蒸气清洗的，速度限制器印章会经常破裂。因此如果交通警察所能发现的证据仅是一个受到破坏的印章而已，那么该司机就不会以篡改罪被起诉（当然，卡车司机都知道这一点）。

也有其他的结果。例如，在打开一个印章完好的信封后，罪犯可以重新把它封好。并在信封上贴上具有“海关启封”或“在运送中破损——由邮局重新封印”字样的标签。更有甚者，他会用胶带把信封粘上，并在上面潦草地写上“送达到错误的地址，请重发”。

必须仔细考虑那些失败和攻击的结果。如果目标是防止产品被大规模伪造，偶然的破坏可能没有多大关系。但是如果要支持起诉，自发的印章失败就可能是一个很大的问题。在极端一点的例子里，太信任印章的健壮性可能会导致误判，并彻底破坏印章产品的可用作证据的价值（随后是商业上的价值）。

12.5.4　材料控制

另一个很普遍的弱点是印章材料的供应不受控制。公司的印章是一个很好的例子。在英国，典型的印章是由两个压纹金属图版组成，这些金属图版被插入到特殊的钳子中。有几家金属图版供应商。一个律师告诉我，他曾经定制了上百个金属图版，却没有受到任何的审核。尽管去定制一个“Microsoft Corporation”的印章会有一点点的危险，但去定制一个名气稍微小一点的目标却应该是很容易的——只需写一封看似来自于一个律师公司的信。

或者考虑由一些速递公司使用的塑料信封，它被设计为在打开时会拉伸或撕裂。这是一项很有前途的技术，但是只要公司的常客有零星的信封供应（这些信封也可在仓库中获得），那么就不能阻止一个攻击者对速递之前或之后的包裹进行篡改。

曾经有一段时间，有一个“城市神话”事件，即警察和安全局无法打开一个信封而不留下任何痕迹，如果信封口已经用粘的胶带增强过，并且还曾用拇指摩擦胶带，使之变亮那就更困难了（我近来收到一些来自银行的文书，正是用这种方式封印的）。这并不完全可信——即使没有任何警察的实验室已经发明对付 sellotape（透明胶带的商标名称）胶的神奇溶剂，19 世纪的沙皇警察已经使用叉状的棍来卷起已封印好的信封中的信，这样，他们就可以将信取出，然后偷看，最后再放回去 [428]。

即使透明胶带可以保证在信封上留下可见的标记，人们也必须假定警察的信封蒸烘部门没有储藏同样的信封，并且收信人也不会太认真，从而发现不了伪造的信封。考虑到很容易就能扫描带有一个公司标志的信封，并用桌面印刷设备进行复制，人们的这个假设就未免太自信了。总之，高精度桌面彩色扫描仪的使用，使得相当多的组织停止使用预先印制的信纸来写信。这就使得伪造者的工作更加容易了。

12.5.5　不保护正确的事物

本书曾提及在 20 世纪 80 年代晚期，信用卡是如何易受攻击的：认证终端读磁条，付款汇票捕获设备使用压纹；骗子们改变磁条，而不改变压纹，就可以击败系统。

也有涉及到局部改动的攻击。例如，由于信用卡上的全息图仅覆盖后 4 位数字，攻击者总是改动其他的 12 位数字。当银行用于生成信用卡号的算法被知道后，所需要的工作仅仅涉及拉平、翻印和在卡的其他地方重做压纹。这些只需要廉价的设备就可以办到。

那样的攻击现在很少见，因为罪犯意识到很少有商店职员会确认凭条上的账户是否与卡上的浮雕数字相同。所以，凭条上的账户数字不需要同卡上的数字有任何的相似。事实上，全息图的存在只是表明，“这是一张一次有效的卡。”

最后，食物和药品生产商经常利用压缩包装和硬质泡沫塑料衬垫包装，来防止他们的产品被更改。如果设计得好的话，要想仿造得很好，以便能通过仔细的审查还是有相当难度的。然而，当选择防护措施时，必须要对威胁模型非常清楚——是假冒、更改、复制、仿造、转换、稀释、替换或者其他方式 [615]？如果威胁模型是一个拿着充满毒药的注射器的精神病患者，那么简单的硬质泡沫塑料衬垫包装和压缩包装就不够用了。真正需要的是篡改感应膜，它对微小的穿透都会进行可见的和不可逆转的反应（那样的感应膜确实存在，但是对于消费者的产品来说还是太贵了，本书将在抵制篡改一章对一种感应膜进行讨论）。

12.5.6　检查的成本和性质

在利用其他的东西来置换银行卡上的全息图——假定用兔子代替鸭子——的犯罪行业里，有很多这样的故事，而店主的反应只是说：“哦，瞧，他们改变了全息图。”这不是对全息图的批评，但这个问题很深，涉及到应用心理学和公众教育。银行家担心当引入新的支票时，在每个人对新的支票都很熟悉前的数周，是伪造者的幸运时期（这是计划引入新欧元纸币时最令人担心的一个问题）。

一个相关的问题是护照、驾驶执照、印有抬头的信笺、公司印章的多样性，以及包装的变化。没有真正的样本供比较，那么检查或多或少就只能限制在初级层次，所以伪造就很容易。即便银行职员有印有外国纸币的书，移民局官员有外国护照的图片，但通常只有少量的有关安全特征的信息；在任何情况下，真正物理样品的缺少意味着产品的触觉特征不能受到

检查。

正如以往提到的，许多技术的限制性因素是现场二级检查的代价太高。如果检查一瓶伪造的香水需要价值 500 美元的设备（如具有扫描仪的笔记本电脑、紫外灯，以及特殊的显微镜），这对于只通过少数高级商店销售的特殊香水是可行的。但是对于中等价值的商品，这却是不可行的。也不可能将这些设备分发给全世界的海关值班人员和市场检察员。

应对的想法仍然是印章，它可以由经过最少训练的公众或员工进行检查。对伪造看得很重的公司，例如大的软件公司，正开始采纳许多由纸币印刷机所倡导的技术。但是高价值商品的包装比纸币更难保护。熟悉很重要：人们对于经常处理的事情有一种“感觉”，如当地钱币，但很少会注意到具有错误包装的某些东西，这些包装是人们很少看到的。例如轿车部件或药瓶。当首次看到某样东西时，人们是很容易受骗的——例如计算机操作系统最新版本的包装。

12.6　评估方法论

本节提供一个系统的方法来评估针对给定应用提供的印章产品。而不是仅仅问：“你能以除了明显的方法以外的其他方法来去除印章吗?”我们需要跟踪它，从设计到实地检查、到制造、应用、使用、检查、破坏，直至退出服务。在此，有一些需要提问的问题：

- 真的有人知道他正在努力做什么来击败系统吗？而到底什么算得上击败系统呢——篡改、伪造、更改、破坏可作证据的事物，或对商业信用进行“PR”攻击？
- 设计这个系统的团队的声誉如何——他们有破解对手产品的经历吗？
- 系统能在实地用多久，技术的进步会使攻击变得非常容易的可能性有多大？
- 印章材料的可获得性有多广——其他的人中有谁能购买、伪造或偷取设备？
- 应用印章的人会粗心或受贿吗？
- 使用印章的方法能保护产品的正确部分（或足够多的部分）吗？
- 什么是质量问题？灰尘、噪音、清扫和产生缺陷的效果如何？产品能经受住天气变化、飞溅的燃料颗粒，或掉入一杯啤酒中吗？如果这样的事情发生，系统能像期望的那样进行显著地反应吗？随机印章失败的概率有多大，有些什么效果？
- 假如一个印章是伪造的，能期望谁发现这个问题呢？如果是公众，他们中有多少人能经常看到真的印章呢？销售商做过实验吗？这些实验能以应用心理学的标准通过检查，并建立适当的错误接受率和错误拒绝率吗？如果你是现场的检查员，设备和训练的成本有多高？
- 有任何证据上的问题吗？如果你准备在法院上结束争端，除了你自己的（或销售商的）专家外，其他方还有人可依靠吗？如果答案是否定的，那么这是好事还是坏事呢？陪审团为什么要相信你——系统的发明者，而不相信被告席上甜美、娇小的大小姐呢？在公平审判的原则下，法官会饶恕她吗——因为反驳你的技术声明是释放她的证据的不可能的负担（正如发生在 Judd vs. Citibank 的事件，这个案件对于提款机的幻影提款制定了法律）[427]。
- 一旦新印章产品被使用了，那些旧印章怎么办？你是否担心有人会从垃圾堆中复制几个旧印章？

当考虑到应用和检查印章的人员是否忠诚和能有效地执行他们的工作时，分析动机、机

遇、技巧、审计和责任非常重要。在印章被对手（如在合同制作的案例中）和愿意收受贿赂的人（如想得到卡车公司生意的汽车修理技师）应用的地方要特别小心，最后，要仔细考虑印章失败和检查错误率的可能后果，不仅要从客户公司和对手的角度出发，还要从背景清白的系统使用人员和合法证据的角度出发进行考虑。

当然，一般而言，整个生命周期保证过程也应该应用到计算机系统上。本书将在第三部分对这个问题进行更多的论述。

12.7　小结

大多数商用的印章产品相对容易被击败，特别是当印章检查由未接受训练的人员随意执行时。印章必须在产品的整个周期中得到评估，从制造到材料控制、应用、验证，到最后的毁坏。在关键的应用中，进行有敌意的针对性测试是非常明智的。印章往往取决于安全印刷。关于这个方面，人们可能会做出许多相似的评论。

研究问题

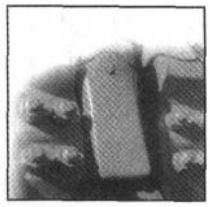
现在，大量资金被投入到这个领域的研究和产品发展上。问题是看起来似乎大多数的投资并没有达到预期效果，或者一些第三级别的产品会继续主宰市场，因为这些产品的成本很低，以及用户不太内行。一个重要的贡献是为印章制定一个更好的评估方法，并且对于安全印刷也是如此。更多有关特定的技术和产品如何被攻破的结果会对削弱供应商的自满非常有用。

参考资料

有关安全印刷的权威性教科书是 Van Renesse 写的 [765]，书中不仅讨论了一些技术手段，如全息图和精细动态全息，还讨论了它们在不同的应用中如何工作。这些应用包括从纸币印刷到护照、包装。这是非常重要的背景阅读。

我不知道是否有关于印章的权威性教科书。许多产品是专利性的，并且依靠罪犯的不知情而取得成功——是我所知道的最不稳定的基础之一。战胜这个忽视的最系统性的努力之一可以参看 Los Alamos 国家实验室的印章易攻击性评估小组发表的一系列文章 [422]。

第 13 章　生物测量学



基列人把守着约旦河各渡口，不容以法莲人过去。以法莲逃亡的人若说："容我过去"，基列的人就问他们说："你是不是以法莲人？"他们若说不是，基列人就问他们："你说'示播列'"；以法莲人因为发不出准确的字音来，便说"西播列"；基列人就把他抓住，杀死在约旦河渡口。当时有 42 000 以法莲人被杀。

——《圣经》"士师记" 12:5-6

13.1　引言

以上所引用的内容可能是第一个被记录的安全协议的军事应用。在这个安全协议中，认证取决于人类的特性——此处是人的口音（在此之前也有少量的正式应用，正如东北非的伊萨人试图通过人的头发来鉴别以扫人，却被雅各布人欺骗；或者通过面部特征来相互识别，本书将在后面的章节介绍这些内容）。

生物测量法是通过量测一个人的解剖学特征或物理特征（如手的几何特征和指纹）、一些根深蒂固的习惯，或者其他的行为特征（如手写签名），或这两者结合的某些东西（如声音）来鉴别人。

在过去的 25 年里，人们开发了大量的生物测量设备；这个迅速增长的市场现在每年有五千万美元成交 [414]。本书前面曾提到在 20 世纪 70 年代末期，人们用手的几何特征来鉴别一个核反应堆的工作人员。但是建立的最好的生物测量技术要比计算机出现早得多——即手写签名、面部特征和指纹的使用。本书将首先对这些内容进行讨论，然后是较特别的高科技技术。

13.2　手写签名

手写签名曾应用于古代中国，而私人刻制的印章被视为具有更高的地位，至今仍在中国、日本和韩国用于重要的交易之中。欧洲是另外的一种情况：印章使用于中世纪时代，但在文艺复兴以后，书写变得很广泛，人们就越来越多地使用手写签名来表示自己对商业和其他文件的同意。随着时间的过去，在西方，签名已被接受为处理事情的标准方法，每一天都有价值数十亿美元的合同通过手写签名来最后确定。而如何用电子签名来代替手写签名，仍是一个热点的政治问题和技术问题。

手写签名有多安全？

伪造的签名被当做真正的签名接受的几率取决于检查签名时的细心程度。许多商场里的银行卡交易在被接受时，工作人员对卡上的签名甚至连看都不看一眼——到这样的程度。以致于许多美国人都不愿意在他们的卡上签名（当在比较一丝不苟的国家如德国和瑞士旅行时，这样做会引起很多问题）。但即使是非常谨慎的签名检查也不会使欺诈的风险降到零。

一个实验表明：总共105个专业的文件检查人员，每个人做144次对偶比较，结果有6.5%的文件误判。同时，让34名未受专业训练、但具有同样教育水平的控制团体做同样的事情，错判率为38.3%；并且，还不能通过给这些非专业人员以金钱激励来提高其执行水平。由专业人员造成的错误是这个行业一直讨论的问题，但被认为反映了检查人员的假想和偏见[81]。当给予参加测试的人员以合理的手写样本，而不只是一个签名时，似乎可以这样认为：验证支票上的签名或信用卡保证人的签名所得的结果错误率会更高。

所以手写签名受限于许多的条约和特定的规则，这些条约和规则因国家不同而存在差异。例如在英国，如果你想从银行贷款购买房子，但你不是这家银行的确定用户，那么这个过程是：先带上护照等文件去律师办公室、在财产转移和贷款合同上签字，然后得到由律师签署的合同。抵押贷款人需要提供政府发行的照片ID号，以使其保险公司高兴；而在几个世纪以前，政府要求记录每一笔不动产的买卖，以便于从财产交易中收集印花税。其他种类的文件（如专家证明）也可能需要以特殊的方式进行确认。许多不寻常的事情可以追溯到19世纪，以及打字机的发明。一些国家要求机器打印的合同在每页都必须大写首字母，而有的国家就没有这个要求。这些不同持续存在了一个世纪。合同中的冲突也会引起严重的问题。有这样一个实例，一件在西班牙进行的交易被认为是无效的，因为交易是以传真方式完成的，最终导致一家英国公司倒闭。

然而，在大多数说英语的地方，大部分的文档并不需要用特殊的措施进行认证。签名的本质在于签名人的意图，所以一个文盲写在文档上的“X”符号与君主的华饰一样有效。事实上，位于E-mail底部的纯文本名字有着同样的法律效力[810]，除了在某些地方，有规则要求交易必须用文字记录。在每一个权限管辖区域内，都可能有成千上万的规则。同时，在法院的案件中，对签名进行争辩还是非常罕见的，因为文章的内容一般就能说明什么人做了什么事情。所以尽管我们的生物测量机制非常弱，但它们在实践当中运行良好——除非被某些程序性的条规所限制。当然，这些条规因国家和应用的不同而异。

将这些混乱的事情整理好，并对电子文档施加一些相对统一的规则，是一项非常国际化活动的主题。在[811]中可找到这些主题的总结，在[68]中有针对国家调查做的相关分析；本书将在第二部分进一步讨论其中的一些问题。到目前为止签名的形式、伪造的难易程度，以及在给定条件下是否具有法律效力，都是非常独立的问题。

尽管如此，有一项应用，对于手写签名的有效自动识别非常有价值，这就是支票清算。

在银行的支票处理中心，人们通常只验证超过一定金额的支票——可能是一千美元或一万美元，也可能是按过去三个月中账号上金钱转移的一定百分比。签名验证由负责人执行，同时将支票图案和顾客的参考签名投影到屏幕上。

验证小额支票是不划算的，除非它能变成自动处理的。所以许多研究人员着手研究对手写签名进行自动比较的系统。结果证明这需要进行非常困难的图像处理工作，因为在一个亲笔签字与另外一个亲笔签名之间有着很大的变动。一个更加容易的选择是使用签名板。它是一个传感器平面，用户可以在上面签名；它不仅记录弧线的形态，还记录它的运动方式（如手的速度、笔在何处抬离纸张等）。签名板运用于某些高价值应用中识别用户，包括安全交易。

与警报系统类似，许多生物测量系统在错误接受率和错误拒绝率之间存在着平衡。通常，银行将错误接受率和错误拒绝率指定为欺诈率和侮辱率。而在生物测量文献中被指定为

类型 1 和类型 2 错误。许多系统可以调试到使一个人的满意度超过其他人。等同错误率是将系统调整到错误接受率和错误拒绝率相等。对于普通的签名辨别系统，等同错误率约为 1%。这对于在诸如银行交易间里进行的运作并不是毁灭性的。如果交易者中的一个人想登录他的 PC 机，而机器拒绝他的签名，他可以重试。如果持续失败，他可以打电话给管理员，并使 PC 机复位。然而，在零售店里，它将是不能容忍的。如果 1% 的交易失败，那么对顾客的冒犯会是不可原谅的。因此银行设定生物测量技术的欺诈率为 1%，而侮辱率为 0.01%，这远远超过签名验证技术现在的情形 [317]。

那么，什么可以用来弥补这个差距呢？英格兰的 Kent 大学做了一个有趣的实验，目的是减少社会福利申请人所进行的欺诈活动，这些申请人在靠近南安普顿的一个邮局里领取救济金。这个系统的独特性在于它能用于屏幕签名，从而支持人工决策，而不是由系统自己作出决策。它的欺诈率和侮辱率大致相等，而不是将系统调整到具有低侮辱率和高欺诈率。当一个签名被拒绝时，这个系统仅是告诉工作人员去做进一步的检查，要求提供签名的人提供驾驶执照或其他相片 ID。在从 343 位顾客那里采集的 8500 个样本中，98.2% 在第一次实验时得到正确验证，在第二次实验时正确验证率升至 99.15%。因此，实验被认为是成功的 [282]。然而，这个比率通过排除替罪羊而获得成功。替罪羊是在生物测量等技术领域中使用的一个术语，它用于指代那些不能对其模板进行很好分级的人。若包括它们，错误率为 6.9% [283]。

通常，生物测量技术机制在有人参加的情况下，倾向于更加健壮，它们用来辅助一个警卫而不是取代他。通过使警卫保持警觉，错误的报警率也可能真正起到帮助作用。

13.3　面部识别

通过面部特征来识别人是所有鉴定机制中最古老的一种，它至少可以追溯到我们远古的灵长目祖先。生物科学家认为这是人类认识行为中的一个重要部分，并且已经得到进化，提供了有效的方式来识别他人的面部特征和表情 [646]。例如：我们极其擅长于发现另一个人是否在注视我们。在理论上，人类通过面部表情鉴别人的能力，似乎要比如今制造的任何一个自动系统都要好。

人类识别面部的能力对于安全工程师来说也是非常重要的，因为现在人们对相片 ID 的依赖极其广泛。驾照、护照以及其他种类的认证卡，不仅直接用于计算机房的门禁，也能用来启动大多数的其他系统。有关密码、智能卡或使用其他诸如虹膜识别技术的生物系统来对用户进行注册的话题，经常是一个实施过程的终点。这个过程从某个人申请工作，开账户或做任何其他事情呈交相片 ID 时开始。

但是即使人们擅长于识别活生生的朋友，他们利用相片 ID 来鉴别陌生人时又有多擅长呢？

简单的答案是他们不擅长。Westminster 大学的心理学家在一家超市连锁店和一家银行的帮助下做了一个非常有吸引力的实验 [450]。他们招收了 44 名学生，并给他们每人发四张信用卡，每个卡上有一张不同的相片，具体如下所示：

- 相片中有一张是“好，好”的，它是真实的，并且是最近拍的。
- 第二张相片是“坏，好”的，它是真实的，但是有点旧；与相片相比，学生现在穿的衣服不同，发型不同，或一些其他的区别，换而言之，这种相片是大多数人用在

相片 ID 上的标准相片。

- 第三张是“好，坏”的，它是从一堆一百张左右不同人的照片中随机挑选的，检查者选了一个与主体最相像的照片，也就是说，它是在罪犯有一堆偷来的卡时，会做出的典型选择。
- 第四张是“坏，坏”的，它是随意挑选的，当然除了性别和人种要相似以外。即是懒惰粗心的罪犯也能做到的典型匹配。

实验在正常营业时间结束以后的一个超市里进行，但仍让有经验的收款员值班，他们知道这个实验的目的。每个学生使用不同的卡通过交款台数次。事后发现，竟然没有一个交款台的工作人员能辨别出“坏，好”和“好，坏”相片之间的差异。事实上，他们中的一些甚至不能说出“好，好”与“坏，坏”相片之间的区别。由于这个实验是在最佳的条件下进行的——有经验丰富的工作人员，充分的时间，在卡被拒绝后，不会有任何尴尬会发生的威胁。真实世界的执行效果会比这更糟（事实上，许多商场都不会将信用卡公司为发现被盗信用卡而提供的奖赏付给收款员。因此，即使基本的检查动力也不会具备）。

银行业对这个实验的反应是很矛盾的。至少有两个在信用卡照片上做过实验的银行，曾经历过在欺诈方面的案件大大降低的情况——比一家苏格兰银行估计的 1% 要少 [67]。全面的结论是：从相片 ID 中获得的效益是它的威慑效果 [293]。

人们很难有效使用面部识别技巧，所以力图使这个过程自动化。这个尝试可追溯到 19 世纪，当 Galton 发明了一系列用于面部检测的弹簧式“机器选择器”时 [328]。但是自动面部识别确实包含有许多相对独立的问题。在身份验证中，接受验证的人直视着有灯光控制条件的摄像机，并且他们的脸与文件上相片的脸进行对比。在司法当中，人们发现了一个相关但更困难的问题，就是需要确定一个嫌疑犯的面部特征是否与低质量记录的安全录像相符。难度最大的是监视，其目标问题可能是对在机场移动的人群进行扫描，然后试图找出一个位于数百个已知嫌疑犯的名单中的人来。

即使从一群人的相片中寻找特定面孔也不是一件轻松的计算机任务 [502]。一个最近的对不同面部特征提取方法的经验性的研究发现，考虑合理的光线、视角和表达方式的变化，没有一种方法是自身充分的，并且错误率高达 20% [10]。几种技术的结合使用会使错误率下降，但达不到 1% 或更少，而对于其他的生物技术来说这种下降是可能的 [556，818]。

简而言之，如果仅根据错误率来考虑的话，这项技术仍不能很好的运作。但是从系统的观点来看，它确实工作得很好。在 1998 年，伦敦市 Newham 区将摄像机放置在大街上，并且开展了一场有关他们的新计算机系统如何能不断对人群的脸进行扫描，从而搜查数百名已知的当地罪犯的行动。这种措施成功地使抢劫、入店行窃和街头犯罪率显著下降。这个系统甚至使行动自由论者感到担忧——尽管它工作起来主要是通过遏止进行 [739]。当然，随着时间的推移和技术的进步，潜在影响和担忧都会增加。

13.4　指纹

指纹是相当重要的。到 1998 年，指纹识别产品占据了所有生物测量技术产品总销售额的 78%。这些产品依据覆盖手指上的摩擦状突脊，并对诸如分叉和突脊末节点的细节进行分类。有的技术也依据突脊皮肤上的孔隙。对前沿的自动指纹鉴别系统的技术描述见 [496]。

使用指纹来鉴别人独立发现的。马克·吐温 1883 年在他的著作——《密西西比河上的生活》中就曾提到指纹。在书中他声称已经从曾经是囚犯看守员的法国老人那里了解到指纹。在此之前很久，17 世纪的中国人已使用指纹作为法律标记，当作印章或签名的替代物；指纹也应用于 18 世纪的日本，当一个文盲男人想与他的妻子离婚时，当时的法律要求他必须使用手印；它们也在 17 世纪的意大利，Malpighi 的作品中提到；在 1691 年，它也被爱尔兰 Londonderry 城的 225 位市民所使用，用在 William 国王对这座城市进行包围之后他们要求赔偿的请愿书上。

指纹的第一次现代的、系统性的应用，似乎是在 19 世纪中期的印度。当时 William Herschel（其祖父是一位天文学家）是 Hooghly 的殖民地官员。他使用指纹来停止对已经死去的囚犯进行监禁，并防止有钱的罪犯花钱雇穷人来替代他们坐牢。Henry Faulds 是日本的一个传教士医生，在 19 世纪 70 年代独立发现了指纹技术，并使其引起达尔文的注意。达尔文随后又推动 Galton 来制定一个用于对指纹样式进行分类的方法。他将指纹分类为圈、螺纹、拱、帐篷等形状，至今仍在使用。

据历史记载，指纹在 1900 年进入警界主流的应用，当时，来自 Bengal 的前任警长 Edward Henry 成为伦敦市警察局专员[⊖]。Henry 的贡献是发展了 Galton 的分类方法，使其变成一个称为 binning 的指标系统。通过依据嫌疑人的 10 个手指中的每一个手指是否有螺纹——环形图案——而分配一位，他们将指纹分为 1 024 位数据文件。单用这种方法，就可能减少需要依据幅度的大小顺序搜索的记录数。

指纹现在基本上被世界上的警察组织用于两种不同的目的。在美国，它们的主要用途是身份识别。FBI 的文件用来核对已逮捕的嫌疑犯，以决定这些嫌疑犯是否还需要被其他的法律执行部门审问。它们也用于挑选求职者；例如：任何人想要得到美国政府在机密级别以上的允许阅览机密文件的授权，就需要有 FBI 的指纹核查。它们也用于司法部门的犯罪现场鉴定工作中。在欧洲，人们携带身份证卡片，因此身份识别工作容易建立起来，司法部门是其主要应用场所。

将在犯罪现场发现的指纹与数据库中的记录进行匹配，匹配度超过一定级别的指纹被当作强有力的证据，用以证明一个嫌疑犯曾经到过犯罪现场，通常足以保证对某个人的指控是正确的。一些国家会从所有的公民和常住的外国人那里收集指纹。

为减少人工指纹匹配的成本，许多自动系统已经发展起来，适合于图像处理步骤的算法在 [522] 中进行了讨论。并且在 [415] 中也有关于 IBM 指纹系统的教程和描述。其中一些系统用于替代以前的人工分类和匹配过程，或力图提高其执行效率 [779]，其他的则使用指纹读取设备来实时进行认证，用于诸如大楼门禁和津贴发放的场合 [258]。它们也用于诸如印度和沙特阿拉伯等国的银行系统中，在那里墨水指纹的使用是很普遍的，因为这些国家中的很大一部分人没有接受过正式教育。

指纹并未因为犯罪组织的缘故而从北美和欧洲的银行系统中取消，尽管你在美国银行取款却又不是其顾客的话，只有很少的美国银行会真的需要指纹。银行方面发现指纹的应用能

⊖ 在西班牙，1892 年它们最先用于 Argentina，在那里，指纹用作谋杀的证据；在古巴，1907 年建立了指纹局，早于美国，其第一次认证出现在 1911 年的 Illinois。Croation 指明 Argentinian 系统由 Juan Vucetich 发展而来，他是来自 Dalmatia 的移民。德国引用了 Breslau 的 Professor Purkinje，他 1828 年写了有关指纹的文章。1898 年意大利在 Calcutta建立了指纹局。成功来自许多人的努力。

使支票欺诈减少一半。一些银行甚至采集新顾客的指纹，并且发现顾客的抵制远较预期得要少，特别是当他们使用扫描仪采集指纹，而不是用墨和纸时［314］。同样，其效果能起到很大的威慑作用；在 FBI 的数据库中匹配一个指纹要比在典型犯罪现场所需要的指纹匹配工作困难得多。因为犯罪现场的指纹匹配工作仅涉及到一百个左右当地比较活跃的窃贼。尽管如此，在美国银行中，指纹已慢慢地被禁止使用，因为它侵犯个人隐私。

这样，指纹识别的效果有多好呢？在法庭应用中，其错误率可以非常得低，它的局限性在于从犯罪现场取到的图像大小和质量。当然，它因国家的不同而不同，并取决于警察的办案程序。英国传统上要求指纹匹配达到 16 点（相应的细节）。一个英国警察专家估计，这仅可能会意外地发生在 40 亿分之一和 100 亿分之一的匹配之间［485］。希腊接受 10 点匹配的细节，土耳其为 8，美国没有设置限制点（相反，它只是对检验人员进行认证）。这意味着在美国，可以用质量很差的指纹作为匹配。但是他们可以接受公开的质疑。在英国，指纹作为证据采用了近一个世纪，也没有发生任何一起获得成功的争议；在美国，确实偶尔会有质疑方成功的争议，并且各方专家之间的争执也是为人所知的。

最近发生的一个案例颠覆了英国人惯有的自信［538］。Shirley McKie——一个苏格兰女警察遭到起诉，根据就是所要求的 16 点指纹匹配。这个指纹匹配是由苏格兰罪犯记录办公室的四名检查人员进行验证的。辩方请来了两个美国检察员，他们提供证据证明其鉴定无效。

McKie 被释放了。因为没有宣布审判团是否一致同意外国专家，或者仅仅考虑到他们的证词是为否定苏格兰专家而做的，苏格兰犯罪记录办公室在超过一年的时间里一直坚称他们的鉴定是有效的。但是到了 2000 年 6 月，案情传到了苏格兰议会，司法部长自己不得不辞职。这个问题似乎是，如果他们承认指纹不是 Shirley 的，他们也可能不得不释放 David Asbury，他在这个案子中被判定为凶手。他的指纹鉴定现在仍被一些专家所怀疑，并且他自己也正在进行申诉［334］。

关于指纹，有四条评论，依次如下：

- 即使 16 点匹配的错误率为警界所声称的十亿分之一（10^{-10}），一旦将许多指纹相互比较，则这个概率理论就会受到攻击。以前工作良好的系统，只是将犯罪现场的指纹与所知道的 57 个当地偷窃者的指纹做手工比较。当每年有成千上万的指纹需要与在线的拥有上百万样本的数据库中的记录进行比较时，这个系统就会崩溃。不可避免的是，迟早需要做足够多的匹配来查找一个 16 点的错误匹配。实际上，由于绝大多数在指纹数据库中存有记录的人是地位低微的罪犯，他们不可能鼓起勇气做出像 McKie 那样的坚决辩护，所以如果已经存在其他的错误判决，我不会感到奇怪。
- 正如图 13-1 和图 13-2 将要说明的，指纹的图像常有很多“噪声”（杂乱信号），例如因灰尘而模糊，所以非常有可能发生错误。检查人员的技能（和偏见）也会影响判断结果，没有经验的法官往往想不到其重要性。噪声在超过一定数量的时候，所引起的错误更加明显。例如，binning 错误率被认为可以导致几个百分比的错误拒绝率［154］。
- 认为任何一个安全机制都是绝对可靠的，会导致产生自满和粗心进而破坏其正确使用。伴随着计算机匹配的引入，却没有人考虑，要将必须的检验点数从 16 提高上去，提高到 20 点。16 点是传统的，系统是绝对可靠的，又肯定没有理由来获得公共

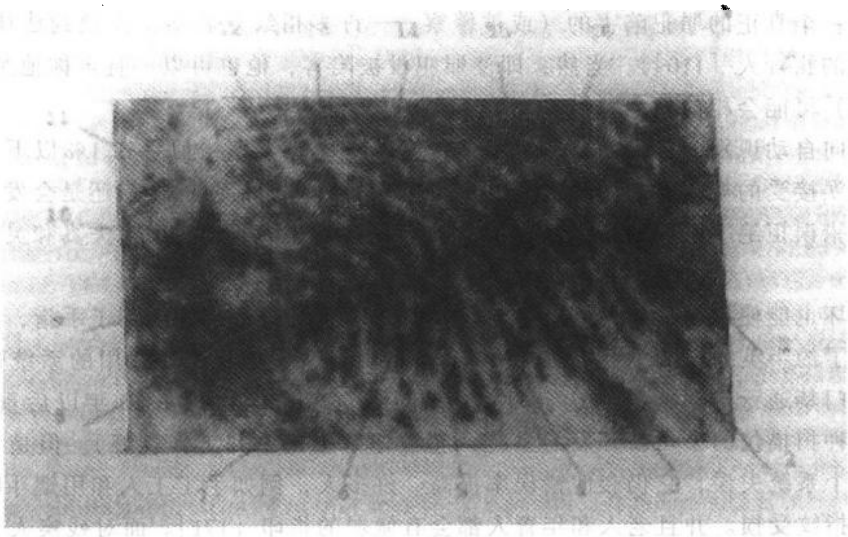


图 13-1　犯罪现场的指纹

图 13-2　带墨的指纹印

基金支持被告，以聘请他们自己的专家。事实上，由于所有可能请到的专家是警察或以前当过警察，所以也没有独立的专家可以聘请。

- 相信绝对可靠性注定了最终失败，这点造成的后果是非常严重的。正如 9.4.3 节所描写的 Munden 案，它帮助盗窃者声明提款机的安全性。一个安全机制是绝对可靠的假设会导致程序、文化假定，甚至法律的出现，它们能确保其最终失败，并可能对涉及的个人带来灾难。

然而，即使当我们确实拥有一个正确的匹配（具有 20 点或 24 点或无论多少点），它的应用也不是完全显然的，指纹有可能利用胶带或用做成的模子进行转移——即便对目标没有任何的认知——使用原来为警察使用而设计的技术。所以，在犯罪现场被发现指纹的嫌疑人

可能是被另一个真正的罪犯陷害的（或被警察——许多指纹欺骗案，涉及到法律执行人员，而不是其他的嫌疑人［110］）。当然，即使罪犯没被陷害，他也可以一直声称他是被陷害的，而（陪审团）可能会相信他。

现在转向自动识别，较好的自动识别系统有着等同错误率，似乎在1%以下。虽然，在理论上，错误接受的概率可以做得任意的小，但在实践中，错误的接受还是会发生，这可能是由于降低错误拒绝率采取的特征处理带来的——例如在特征选择中为失真和灵活性而做的考虑［650］。

指纹破坏也能使识别受到损害。小时候，我在削苹果时不小心削到了手指，这在我的左手中指留下了约半英寸长的疤痕，在1989年当我用这个手指让FBI使用的系统进行扫描以制作建筑物门禁通行卡时，我的这个刀疤使扫描仪失效（但当我在10年以后重新实验时，利用提供这种扫描仪的公司新开发的系统，它被记录下来，并工作良好）。但是，即使刀疤不会导致整个系统失误，它仍会使错误率提高。许多人，例如手工工人和用烟斗吸烟者，经常使他们的指纹受损。并且老人和年青人都会有微弱的指印［171］。面对残疾人、具有生理缺陷的人（比如有额外的手指，以及生来就不具有传统指纹的（少数）人时），自动系统还有很多问题［485］。

也许指纹系统最重要的方面，如同实验室环境中所评测的，不是它的错误率，而是它的威慑效果。这在福利支付系统中曾经特别声明过。即便用于认证福利申请人的指纹阅读机，也有高达5%的错误率［163］，但在实践中它们还是被证明为是一个减少社会福利基金申请人名单的有效方式，所以它们正在被广泛采用［553］。

13.5　虹膜编码

现在我们从传统的鉴定人员的方法转向现代和新颖的方法，在实验室的环境下，用人眼的虹膜来识别人员无疑是自动化系统中错误率最低的。它似乎是控制建筑物入口的最安全的方式，如铍储藏室。

据今所知，每个人的虹膜具有惟一可分辨性。它很容易在视频图像中检测出来，而且不会消失。并且通过角膜（它反过来有自己的清洁机制）与外界环境隔离。虹膜图案包含大量的随机性，并且似乎是指纹自由度的许多倍，它形成于妊娠期的第三和第八个月之间（像指纹图案一样），并且是表型的，因为它呈现出有限的基因影响。甚至对于相同的双胞胎来说也是不一样的（一个人的两只眼也是类似的），并且它们终生稳定。

一种信号处理技术（Gabor滤波器）已被发现并用于从一幅虹膜图像中抽取信息，并形成256字节的虹膜编码，这涉及到在瞳孔与虹膜之间的许多同心圆环上进行的圆形波形转换（见图13-3）并且有着优良的特性，即从同一个虹膜计算出的两编码会典型地在它们的位上匹配90%［218］。这比指纹扫描仪要简单得多。指纹扫描中，定向和细节分类是一项很困难的工作。虹膜编码的速度和准确性已经导致产生许多商业虹膜识别系统［794］。虹膜编码的错误接受率在所有已知的确认系统中是最低的——为空，这可以从美国能源部所做的实验中了解到。等同错误率已显示为低于百万分之一，并且如果某人准备容忍万分之一的错误接受率，那么理论上的错误接受率将少于千万分之一。

虹膜扫描在现实中应用时所面临的主要问题是在获取图像时不能太冒昧。虹膜很小（小于1/2英寸），并需包含几百个虹膜像素点的图像，人将眼睛与摄像机距离保持在几英寸内，

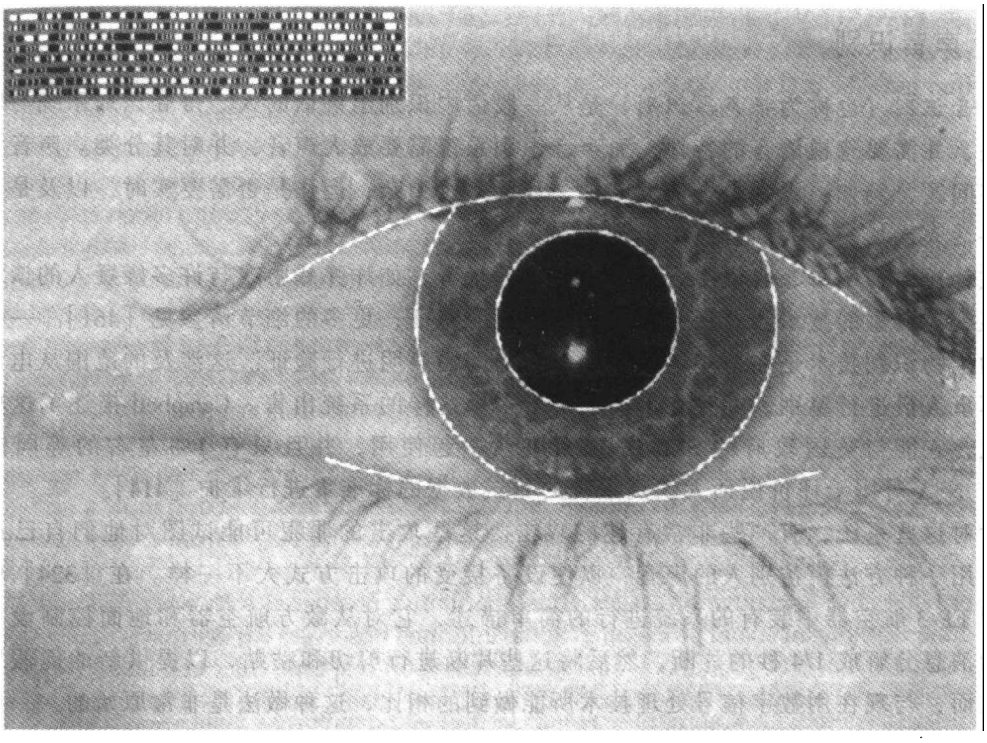


图 13-3　附有虹膜编码的虹膜（John Daugman 许可提供）

而最标准的设备会在两到三英尺的距离内有效工作。在计算机房的门禁中采用这种方式，是可以的，但是在普通的零售领域应用较少能被接受，因为一些人距离摄像机很近时会很不舒服。摄像机不能从几英尺远的距离获取虹膜图像在给定自动面部特征识别，识别范围和距离的前提下，并不是什么技术原因——它仅是代价高一点——而是因为这样容易给人以失去隐私权的感觉（在欧洲，数据保护法将会得到人们的普遍欢迎），还有一些次要的原因，包括眨眼、使眼模糊的睫毛，以及太阳镜。

对虹膜识别系统可能进行的攻击包括——至少在无人看管的工作中——目标物虹膜的一张简单照片，这在有人监督的门禁处可能不会有问题，但是如果多个人开始使用虹膜编码以认证银行卡交易，那么该虹膜编码将被许多组织所知。当虹膜编码可被迅速地比较时（仅单独使用——或一起比较，并统计零比特的数量），它们可能开始承担起名字的性质，而且不会成为密码（正如现行系统中一样）；所以可能会利用虹膜编码将这个人与不同组织的交易联系起来。

对假冒问题的一个可能解决方案是设计能测量虹膜震颤的设备——瞳孔直径的自然振动大约为 0.5 Hz，但是即使这个也不是绝对可靠的。例如，一个人可能会试图在隐形眼镜上印刷目标对象的虹膜样式（尽管现存的空虚的隐形眼镜印刷技术是粗颗粒的，因而是可探测到的）。

尽管困难存在，虹膜编码仍是一个很强大的竞争技术。在正确的环境中，它提供比任何其他方法更强大的确定性，所考虑的个人等同于开始登记在系统上的那个人，他们可以达到具有零错误接受率的自动识别目标。

13.6 声音识别

声音识别（也称为扬声器识别）是从一段话中识别出讲话的人。语音识别系统与转录声音有关，并需要忽视语音的特性；而声音识别系统需要放大声音，并对其分类。声音识别存在许多问题，例如：识别是否依赖文件，环境是否嘈杂，运作是否需要实时，以及是否需要验证讲话人或从一个大的群体中识别出来。

在针对语音的犯罪取证中，通常是对电话录音，如炸弹威胁或与许多嫌疑人的谈话样本进行匹配。典型的技术涉及从光谱中过滤和提取特征；更多的细节请参见［461］。一个更为直接的生物识别技术是在一些电话系统中对等同性声明进行验证。这涉及的范围从电话银行到对军事人员进行鉴别，并且在市场上有十多个这样的系统出售。Campbell 描述了这样的一个系统，它能与美国政府的 STU-III 加密电话一起使用，并且只有 1% 左右的等同错误率［161］；NSA 对用于评价说话人识别系统的测试数据的标准集进行维护［414］。

针对这些系统，有一些非常有趣的攻击。这些攻击的罪犯可能试图对他们自己进行训练，并用一种方法模仿别人的声音，以使设备接受的攻击方式大不一样。在［324］中，有对美国 EP-3 航空器中装有的系统进行的简单描述，它对从敌方航空器和地面控制设备中截取到的消息分解成 1/4 秒的片断，然后将这些片断进行剪切和粘贴，以提供新的但假冒的消息。然而，与现在用数字信号处理技术所能做到的相比，这种做法是非常原始的。一些见多识广的观察者预计数年内，就会有支持实时声音和图像伪造的产品出现。原始声音变体的系统已经存在，并能使受电话性骚扰的女性受害者以男声进行电话回复。更好的系统将能做到这一点：一个呼叫中心总有同一个“人”在问候往它那儿拨叫电话的人。随着这种商业压力对技术发展的驱动，在远距离生物测量技术变得非常困难之前仅是时间的问题。

13.7 其他系统

已经讨论过许多其他的生物测量技术。［553］中有关于这方面的市场调查。基于面部温谱图（脸的表面温度图，从红外图像获得）、耳朵的形状、步态、唇印以及手中血管样式的生物测量技术，似乎不会做成产品在市场上出售。其他的技术可能会在将来提供有趣的生物测量技术。例如在开发数字鼻子用于食品和饮料行业的质量控制方面进行的大量投资会导致数字小狗的出现，它能依据气味识别它的主人。

其他的生物测量技术，比如打字样式，在 20 世纪 80 年代曾用于产品当中，但似乎没有成功（打字样式，也就是所说的敲击键盘的动态。其著名的先驱为在战争时期用无线电报操作员的笔迹来对他们进行鉴定，一如他们使用莫尔斯电码的方式）。

还有其他的一些技术，例如手相几何学，有着有用的瞄准机会的市场。除了从 20 世纪 70 年代开始用于核弹库的门禁外，手相几何学现在还用于机场，由美国移民和归化局对频繁来往的飞行员进行“快速跟踪”。它相当健壮，在实验室条件下，其等同错误率低于 0.1%～0.2%（实际上，手相几何学来源于人体测量学，一个利用骨骼测量技术来鉴别罪犯的系统，它在 1882 年由 Alphonse Bertillon 引入，但过了一代人之后就被指纹所取代）。

另一种生物测量技术值得一提——DNA 配型的使用。它已成为犯罪现场刑事侦破的有用工具，以及在子女养护案例中决定父母。但对于建筑物门禁的应用还太慢。由于 DNA 是遗传型的，而不是表型的，所以其准确性也受限于单卵双胞胎的出现率——约 120 个白人当

中就有一个同样生理特征的双胞胎。也有隐私问题，因为能从 DNA 样本中重新恢复个人的信息。对于刑事 DNA 分析技术的纵览，以及如何建立与欧洲数据保护法相一致的国家 DNA 数据库，见 [680]。

13.8 哪里出了问题

在与安全有关的其他方面，我们发现许多常见的失败例子，原因是由于故障、失误和自满。例如，DNA 配型所面对的主要问题是其最初就有因为不细心的实验室程序而导致的高比率假阳性。这不仅仅会吓跑一些警察部门，这些警察部门将来自不同志愿者的样品送过来，但取回的却是错误的匹配结果。而且也导致了很多有争议的法院案例和宣称的误判。

生物测量技术与许多其他的保护机制相似（报警器、印章、干扰检测围栏……），即环境条件能引起混乱。噪音、灰尘、振动和不可靠的光照条件，都会使它们失效。一些系统，例如声音识别系统，对于酒精摄入者和有压力的人来说是容易失效的。环境假设条件的改变，例如从封闭系统变化到开放系统，从小系统变化到大系统，从有人看守到独立应用，从合作者到反抗的对象，从验证到鉴别——都能破坏一个系统的有效性。

针对不同的生物测量技术系统，还有许多更为具体和有趣的攻击。

- 有一些攻击专门针对用于索引生物测量技术数据的方法。典型的例子是有的罪犯以错误的顺序给缺乏经验的警察看他的指纹，因而其在 Henry 系统下不是以“01101”被索引，而可能变成“01011”。所以他的记录不能被发现，这样，他就可能因为是初犯而处罚较轻 [485]。
- 法庭生物测量技术经常不能像人们认为的那样能说明很多东西。除了指纹或 DNA 样品可能被警察换掉之外，它们还可能是旧的。指纹的时间不能够直接确定，在公众所接触地方的指纹说明不了什么问题。在银行门上的指纹要比留在遭抢劫的保险库上的指纹的说服力小得多。所以在容易遭到抢劫的前提下，清洁步骤对于取证来说是非常关键的。如果一个嫌疑犯的指纹被发现于银行柜台上，但他声称自己三天前确实去过，那么他可能会因每晚分行的柜台都会被擦拭一新的证据而判罪。将这个放到系统术语中，有时效经常是一个关键的话题，一些很难预计的事情能够在“可信计算基”中找到。
- 有时效的另一面是大多数的生物测量系统至少在理论上可以用适当的记录进行攻击。我们曾提到过对声音识别系统的直接攻击，用隐形眼镜上的相片对虹膜扫描仪进行攻击，以及指纹压模。还有更简单的，在那些将指纹用于养老金支付的国家里，有这样一个持续流传的故事，“泡菜坛中的祖母手指”是她留给其家庭的最宝贵的财富。这更加说明了这样一个教训：无人看管的生物测量技术认证设备的运行是不易处理的。
- 一些系统——如著名的手写体系统——对于共谋作案是脆弱的。罪犯可以自愿降低其手写能力。通过使用一些稍有差异的、小孩似的签名样本，他们可以迫使系统接受一个比平常低的阈值。一种可以预料的攻击是：Alice 开一个银行账户，而她的同谋 Betty 从银行将账户的钱取走；然后 Alice 抱怨其钱被偷，并制作出一个周密的辩解。正如警报和共享控制系统一样，商业用户不得不担心合谋的雇员或顾客。而军事威胁模式通常只是面对单个不忠诚的士兵。

- 商业系统的建设也必须担心伪造的声誉——例如，一个经常签名的用户是否能够在签名板上写出两份被认为相同的签名，尽管这两份签名在视觉上明显不同。
- 系统的设计者通常不甚了解统计学方面的知识，并且生日理论尤其得不到很好的理解和应用。例如：在一个具有 10000 项生物测量资料的数据库中，共有大约五千万对。所以即使错误接受率仅是百万分之一，一旦有多于 1000 个人登记时（实际上，是 1609 个人），那么至少有一个错误匹配存在的概率将会上升到超过 1/2。所以鉴定是一项比验证更艰巨的工作 [219]。实际的后果是，当一个为验证而设计的系统在你试图依赖其为证据时会失败。向法官和其他非技术人员解释为何系统的错误率不同于简单样本错误率的一个好方法是："一次机会使其正确，但 N 次机会使其错误。"对于错误率的很好探讨见 [154]。
- 当设计人员认定采用联合生物测量技术的方法可以获得更低的错误率时，统计学的另一个方面起作用了。令人惊奇或可能与人的直觉相反的结果是，联合应用通常会提高错误接受率或错误拒绝率，但同时会使另外一个变得糟糕。一个看待这种现象的方法是，假设你在家中装有两套不同的防盗警报系统，它们同时失效的概率会下降，但错误报警的次数会上升。在某些情形中，例如当一个好的生物测量技术与另外一个非常精确的生物测量技术相结合使用时，效果可能会是完全糟糕的 [219]。
- 大多数的生物测量技术不是对所有的人都很精确，并且有些人的鉴定效果没有其他人的可靠（有时甚至完全不可靠）。老人和手工工人，经常会损坏或磨损其指纹。具有暗色眼球和大瞳孔的人会给出效果较差的虹膜编码。如果那样的系统被广泛使用，那么无指或无眼的残疾人就会有被排除的危险。做记号"X"的文盲更易受到签名伪造的威胁。

  生物测量技术工程师有时将那样的对象轻易地指定为替罪羊，但对于政治领域却是不理智的。一个引发社会倒退的（或看做是）生物测量技术系统可能会遇到原则性的抵制，因为它将残疾人、穷人、老人、少数种族者置于较大的假冒危险中。事实上，一个生物测量技术系统在许多场合下会因法律方面的要求而失败 [626]。对于伤残的（或假装伤残的）的罪犯而言，系统也可能是脆弱的。必须提供操作的后退模式；如果它们是比较不安全的，那么将它们进行强制使用就会受到攻击；若它们是极为不安全的，那么，为何还要使用生物测量技术呢？
- 最后，基督教教义信仰者对生物测量技术感到很不安，他们在圣经的13:16-18 中发现有关反对基督者的描写："他又叫众人无论大小、贫富、自主的、为奴的，都在右手上或是在额上受一个印记。除了那些受印记，有了兽名或有兽名数目的，都不得做买卖。"所以生物测量技术会引起左派和右派人士政治上的一致反对。

所以在生物测量技术准备好应用于大众市场——就如同磁条卡现在使用的方式一样——之前，还有一些较大的问题需要解决。但是尽管成本和错误率方面还有不足，生物测量技术已在许多应用中证明了其自身的价值，最引人注目的地方在于它们的威慑效果很有效。

13.9　小结

一种或另一种的生物测量手段从古代就已经用于鉴别人了，而手写签名、面部特征和指纹就是传统的方法。并且已经建立起使识别工作自动化的系统，这些识别利用传统的和新式

的方法，如手相几何学、声音波纹、虹膜样式。这些系统有不同的长处和弱点。在进行自动识别时，绝大多数的错误率达 1%（尽管虹膜识别要好一点，手相指纹稍微好一点，而面部识别则较差）。在错误接受率（欺诈率）和错误拒绝率（侮辱率）之间存在一个权衡。错误率的统计是令人迷惑的难题。

如果一个生物测量技术使用非常广泛，那么在无人看管的实际运行中，其被伪造的危险就会增加：例如声音合成器、虹膜相片、指纹模型，即使好的、旧式的伪造签名也必须在系统设计时全部考虑到。这并不排除生物测量技术的使用，因为传统的方法（如手写签名）在实践中应用很广，尽管它有很高的错误率。生物测量技术在有人看管的运行过程中，通常会更加强大。因为良好的系统设计、警卫人员和机器识别系统相对的长处和短处，可以相互补充。最后，许多生物测量技术成功地实现了绝大多数或全部的效果，通过威吓罪犯，而不是在鉴定时非常有效。

研究问题

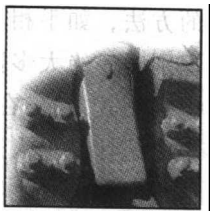
潜在的可获益的研究问题与生物测量技术系统的设计和改进相关。是否有可能建立一个系统——除了虹膜扫描——它能达到银行提出的目标：1%的欺诈率和 0.01%的侮辱率吗？是否有可能建立一个静态签名验证系统，它有足够好的错误率（比如 1%），因为它将用于鉴别支票图像？有全新的、在某些环境下可能有用的生物测量技术吗？

当我写作本章时，在一次与 William Clocksin 和 Alan Blackwell 的谈话中，我想起一项技术，即对轿车进行配置，从而通过司机操纵换挡器和离合器的方式来对其进行鉴别，这种识别可以钩挂在高级轿车报警系统上，报警系统在你的车被盗时，会将其 GPS 位置用电话传到控制中心，中心随后会打电话给你，让你去确认。我们还没有对此申请专利。如果你能使其运转起来，我们所要求的仅是一份致谢——也有些人会想到如何才能阻止保险公司（或政府）得到这些数据访问请求！

参考资料

指纹的历史是相关研究问题很好的阅读材料。标准的参考文献来自 Lambourne [485]，而 Block 收集有许多的美国案例历史 [120]。除了本书中为面部识别和手写识别而引用的参考书之外，在 [433] 还有对 IBM 试验系统的描述，以及在 [181] 对文献的纵览。虹膜编码的标准著作来自 Daugman [218]。对于声音识别，在 [161] 中有一个教程，它主要集中于对说话人进行鉴定。而对于法院方面，见 Klevans 和 Rodman [461]。在《Proceedings of the IEEE》上，有一个关于生物测量系统的专题——卷 85，第 9 号（1997 年 9 月），它提供了对这个技术现行状态的非常有用的简短描述。最后，有关一系列系统的技术细节，有一本由 Anil Jain、Ruud Bolle 和 Sharath Pankanti 编写的书，其中有几个章节的内容是由生物测量系统设计人员撰写的有关该系统的内容。

第 14 章　物理防篡改



如果一个系统按计划运行且正确使用，就会相对容易构建安全的加密系统，而构建一个在错误使用或者一个或多个子部件出问题（或“鼓励”进行错误行为）的情况下仍不会危害安全性的系统却很困难……这是现在仅有的一个封闭世界，离开放世界还有很长一段距离的领域，并且商业加密系统中见到的多次失败也为此提供了一些证据。

——BRIAN GLADMAN

14.1　引言

在前面几章中讨论的技术——涉及屏障、传感器和警报器的物理保护——经常用于保护关键的信息处理资源。

- 银行的主服务器一般会放置在有警卫看守的计算机室中。
- 用于探测非法核试验的地震感应包可能会放在几百英尺深的钻孔里，这个钻孔随后又用混凝土进行了回填。
- 嵌入墙中的自动柜员机实际上是一台放在一吨重的保险箱中的 PC。这个保险箱具有许多特殊的外围设备。不仅包括钞票分发器，也包括探测试图侵犯设备企图的温度传感器，以及用来探测保险箱是否被移动的加速计。一个警报应该会引起设备中的所有密码材料立即被销毁。

但使用大型的防护设备通常是不方便的，这已经导致出现了便携式的防篡改处理器市场。包括从智能卡——它一般执行一套支持如付费电视等应用的有限操作集，到防篡改密码处理器——它被安装在管理提款机网络中 PIN 的服务器上，到精巧的用于军事控制和管理的高端设备。

我应该注明，在防篡改设备和复制设备之间存在着一些相似性。如果一项服务由放置在不同地方的不同服务器实现，它们同时执行交易，并对结果进行投票，那么就有可能需要提供高级别的完整性保护，以防止许多类型的攻击。本书在 11.4 节讨论秘密共享机制时也对密钥材料提供保密性。但防篡改设备至少在理论上能对数据提供保密性。这是关于许多事情能够用数学方法或金属来完成的原则失效的一个体现。

14.2　历史

在密码系统中使用防篡改技术可以追溯到数世纪以前 [428]。海军密码簿曾经受到特别的重视，所以一旦遇到紧急追捕时，密码簿可以从船上扔下去，使之沉入海底。现在，英国政府首相助理用于携带国家文件的公文箱仍是铅衬的，所以会沉入水中。编码，以及最近用于战时密码机的密码，已经用水溶性的墨来印刷；俄罗斯一次性便笺簿印在硝酸盐纤维素上，因此，它们一经点亮就会剧烈燃烧；还有一台美国的战时密码机具有自损的灼热剂材

料，因此在需要的时候它能迅速地销毁。

但是这样的机制依赖于操作员的警惕性，并且密钥材料在突袭时经常被夺取。所以必须做很多尝试来使这个过程自动进行。早期的电子设备，以及一些机械的密码机被制造成打开箱即销毁密码设置。

在密码人员将密钥材料卖给敌方的多起案件发生后，例如美国臭名昭著的 Walker 家族，工程师将更多的注意力放在如何在传输过程中以及终端设备自身上实现密码保护的问题上。目标是“将密码材料的街头价值降到零”。这能够通过防篡改设备（不能从中抽取出密码）或者证明篡改设备（从中抽取密码是显然的）来实现。

文件密码曾经放置在“告密者的容器”中，它是设计用于表现篡改证据的。当电子密码分配出现后，一个通常的解决方案是“fill gun”，它是一个以受控方式发送密码的移动设备。如今，这项功能通常使用小型的安全处理器（例如智能卡）来执行。控制协议的范围从限制一个密码能被分配的次数，到使用公钥加密系统来确保密钥仅装入到授权设备中的机制。密码材料的控制也取得了广泛的用途。在美国和英国，它被集中起来并用于加强正确许可的计算机和通信产品的使用。有效期内的密码材料仅会提供给一个系统（一旦它被正确地授权过）。

一旦初始密码装入，通过使用不同种类的认证和密码一致协议，可以进一步分配密码。在第 2 章“协议”中，本书已经谈论过许多基本的工具，例如密钥多样化，并且在本章的后续部分还将对协议进行更多的阐述。让我们首先看一下对篡改的物理性防御。

14.3　高端物理安全处理器

一个值得研究的例子是 IBM 4758（见图 14-1）。其重要性基于以下两个原因。首先，这是仅有的商业可用的处理器，在本书编写时它已被成功地评估为防篡改的最高级别（FIPS 140-1 级别 4）[576]，这些级别是由美国政府设定的。第二，有很多关于它在公众领域应用方面的文献，包括其设计发展的历史，它的保护机制，以及它支持的事务集[718,795,796]。

这项产品的发展历程简要介绍如下。从计算的最早期开始，计算机因其高价值而进行了物理性保护。然而，多用户操作系统在 20 世纪 60 年代的传播，以及总在它们的防护机制中发现缺陷，意味着很多人可能会接触到正在处理的数据。一些特别敏感的数据——例如长期密码系统的密钥，以及银行客户采用的个人身份识别号（PIN），用于在提款机上识别他们——使人们意识到从商业操作系统获得的保护级别可能还是不完善的。

这导致了独立安全模块的开发，第一个进行成功商业应用的是 IBM 3848 和 VISA 的安全模块。它们都是装入在健壮金属外壳中的微型计算机，具有加密硬件和特殊的密码内存。内存设计为外壳打开时立即归零的静态 RAM（见图 14-2）。在此过程中通过许多帽状开关向密码内存供应电力。这样设备操作者必须重新导入密码。

如何攻击密码处理器（1）

对这样一个设备的直接攻击是让操作人员偷取密钥。在早期的银行安全模块中，主密码放在 PROM 中。PROM 加载在设备的一个特殊插座中，在初始化时被读取或当做在控制台上敲入的数字串。PROM 可以被轻易地窃取，带回家并在业余的设备上读取出来，明码文件密钥更容易被盗取。

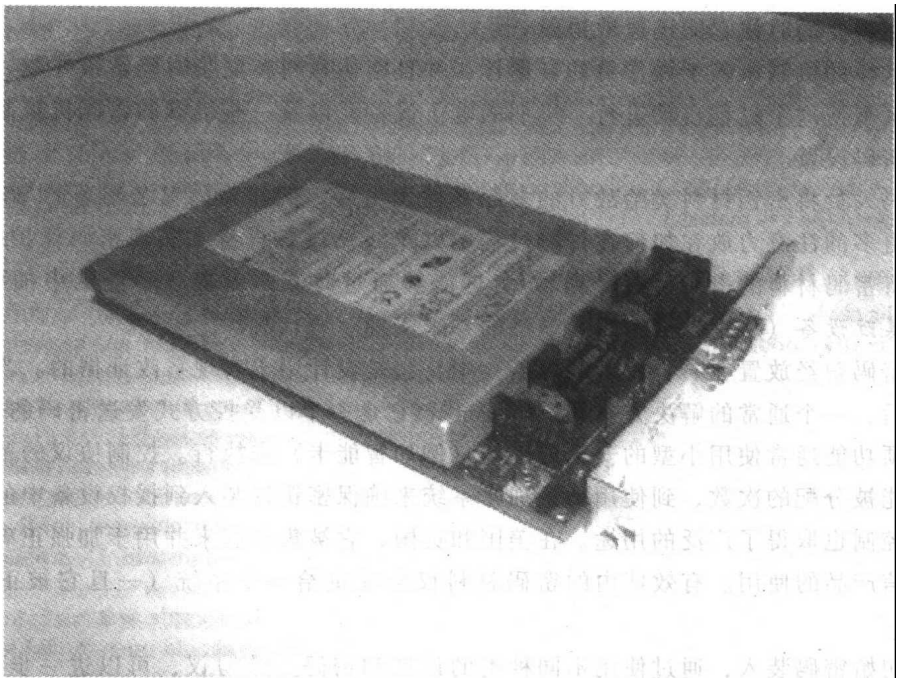


图 14-1　IBM 4758 密码处理器（Steve Weingart 友情提供）

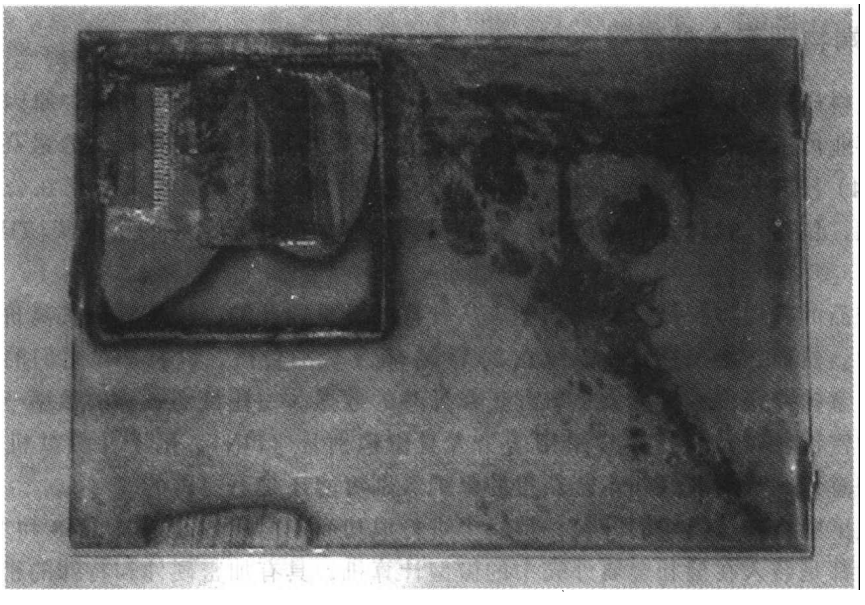


图 14-2　部分开启的 4758，（从左上部向下）显示出电路、铝电磁屏蔽器、干扰检测网和陶制物（Frank Stajano 友情提供）

解决方案是共享控制——使两个或三个 PROM 拥有主密钥，并使设备主密钥是所有组件的异或。这些设备可以保存在不同的保险箱中（虽然是我的后见之明，但异或的使用对此用途的确是一个错误，相反应该使用一个哈希函数，我将很快解释为什么这样做）。

然而，这个过程稍微有点乏味，当它成为例行公事后，也可能会使保密程度降级。理论

上，当对设备进行维护时，它的看管人员应该打开盖罩，销毁有效密码，让维修工程师装入测试码，随后重新装入有效密码。但是具有看管责任的经理经常会直接将 PROM 交给工程师，而不是费心自己来做这些事情。我曾遇到这样一件事情，一个自动柜员机的主密码放在银行分行的通信文件中，这使得任何一个员工都能查看到这些主密码。因此，目标是使重装入密码的次数尽可能的少，比如在维修时或在断电后。所以，安全模块一般由电池来支持主供应电源（至少是对密码内存进行供电支持）。这意味着在实际使用中，当设备初次安装，或在后来的偶然维修后需要看管员载入密码。

人们曾经对密码经常载入还是不经常载入、哪种方式最优的问题进行过争论。如果密码载入的频率非常小，负责人员此前可能从来没有执行过这项任务，并可能因疏忽而泄漏出去，或被职员中技术上敏锐的人所欺骗，从而以一种不安全的方式进行操作（见［19］中有关这些事情的案例）。现代的趋势是倾向于在制造之后、分发之前使机器在安全的设备中产生密码（或录入密码）。那样的密码可能会保留在智能卡中，并用于引导更重要设备的密码。

如何攻击密码处理器（2）

早期的设备对于那些切割包装的攻击者是脆弱的，对于那些能够使盖罩开关不起作用，并在下一次来访时取出密码的维护工程师也是如此。第二代设备对这些问题中相对容易的一些进行了处理，即物理攻击，这主要通过添加更多的诸如光电池和俯仰开关（tilt）等传感器实现。这对于放在进行访问控制的安全区域中的设备是足够的，但最大的问题是防止维修员工的攻击。

许多更好的产品采取的策略是将设备的核心部分（如篡改传感器、保密机、处理器、密钥内存、警报电路）与所有能被服务到的部件（如电池）分隔开来。这样，核心部件就封装在一个坚固、透明的物质，如环氧树脂固体组件中。这个想法是，任何物理性的攻击是“显而易见的”，因为它涉及到诸如切割和打钻的动作，这些动作能被伴随维修技工进入银行计算机室的警卫注意到。

如何攻击密码处理器（3）

然而，如果一个非常厉害的人能够在无人监督的情况下接触到设备，即便是一小段时间（或者是警卫人员没有接受正确的训练），那么仅对设备核心进行封装是不够的。例如，经常有可能用小刀割开封装物，并将逻辑分析器的探针放到核心部件中的总线上。大多数普通的加密算法，如 RSA 和 DES 有着这样的特性：在计算时可以监视任何位平面的攻击者，能够对密码进行恢复［370］。所以在设备运行时能放置探针到设备中的任何一个地方的攻击者，就有可能抽取出秘密的密钥材料。

因此高端产品有干扰检测电池，当它被穿透时会引发其内部密码的破坏。一个早期的例子出现于 IBM 的 μABYSS 系统中，在 20 世纪 80 年代中期，它使用规格 40 的镍铬合金线圈，线圈在嵌入到环氧树脂组件上时很松地围在设备上，然后连接到感应电路［795］。大块移除技术，例如铣削、刻蚀和激光切除会弄断线路，这将擦除密钥。但是环氧树脂线技术对于使用喷砂处理进行缓慢腐蚀是脆弱的；当感应线在封装的表面变得可见时，分流器就能连接到它们的周围。所以 IBM 的下一代主要产品 4753，使用金属屏蔽器，结合印有导电油墨图案的薄膜，并且用有相似化学特性的、持久性更强的材料进行包围。这可以防止那些对薄膜的破坏概率非常高的攻击。

如何攻击密码处理器（4）

攻击者试图使用的下一类方法涉及到内存的利用。它基于这样一个事实：许多计算机内存会保留一些曾经存储在那里的数据的痕迹。有时，所需要的同样的数据在内存中会保留很长一段时间。一个攻击者可能会贿赂汽车修理厂的工人，以得到银行丢弃的安全模块；正如[44]中报道的一样，一旦特定的安全模块在使用同样主密码的情况下运行多年，这些密码的值就会烧制在设备的静态 RAM 中。在通电时，约 90% 的相关位会显现相应密码位的值，这对于恢复密码的值来说足够了。

内存剩磁效应，不仅影响静态和动态 RAM，也影响其他的存储介质。例如，磁盘驱动器磁头，随着时间的过去会改变其排列方式，因此不可能完全重写某段时间以前第一次写入的数据。相关的工程和物理方面的主题在[362]中讨论。NSA 已经出版了有关防止剩磁攻击的条规（《The Forest Green Book》）[243]。

更好的第三代设备有 RAM 保留器，它的工作方式与屏幕保留器的方式很相似；它们都在 RAM 中到处移动数据，以防止数据在内存的某些地方被烧制下来。

如何攻击密码处理器（5）

一个更进一步的问题是计算机内存在低温下会发生冷冻现象。到 20 世纪 80 年代，人们认识到在 -20℃的条件下，当电源拔掉后，静态 RAM 内存中的内容还可以持续一段时间——几秒到几分钟。数据剩磁在更低的温度下可以保持更久。所以一个攻击者可能会冷冻一台设备，去掉电源，切断干扰检测屏障，取出含有密码的 RAM 芯片，并在测试装备中重新充电。RAM 内容也可用离子化辐射进行烧制（对于 20 世纪 80 年代的内存芯片，这需要相当严格的工业用 X 射线机器；但是据我所知，还没有人对现在的非常小的内存芯片设计进行过试验）。

所以，更好的设备应具有温度和辐射警报。这些措施确实很难实施，因为现代的 RAM 芯片在内存剩磁方面表现出很大的差异性，最糟糕的一种内存在室温下就能保留数据长达数秒钟[712]（这表明依赖某些部件的某一特征的危险性，而对于这些部件的制造商而言，这些特性的控制是无足轻重的）。一些军用设备使用保护性的爆炸；用精确计算的铝热剂把内存芯片装载到钢罐中，用以摧毁芯片，而不会导致气体从容器中释放出来。

如何攻击密码处理器（6）

另一套攻击密码硬件的方法，涉及到监测设备发射出来的射频或者其他的电磁信号，甚至向设备注入信号，然后观测其外部视觉效果。这项技术的叫法各异，有称为 Tempest（风暴型）的，也有称为电源分析的，这是一个很大的主题，本书将在下一章中专门论述它。至于 4758，其策略是拥有固体铝屏蔽、以及对总电源供应进行低通过滤，以阻止任何工作在用于内部计算的频率上的信号被泄露出去。

4758 也有一层增强的干扰检测膜，在其中，四个重叠的锯齿形导电图案被涂到聚氨酯板上，它随即又被封装入化学性质相似的物质中。这样，通过切割进入设备的攻击者连探测导电线路都很困难，更不用说与它们相连接了。这些封装分布在金属屏蔽的周围，金属屏蔽中又包含了密码核心（见图 14-2）。这个设计在[718]中进行了详细的论述。

我不知道如何攻击 4758 的硬件。IBM 拒绝向我们出售用于攻击的样品，但在仔细考查了一个样品后，我们确实能提出许多想法，诸如：

如何攻击密码处理器（7）

这里有一些有关如何破解并进入 4758 的推测性的想法。

- 直接的方法是设计一些措施将防护性封装溶蚀掉，探测出网状线，并连通它们周围的分流器。而我想做的第一件事情可能就是试验一下磁动力显微镜。
- 某个人也可以发明这样一种方法：钻一个 8 毫米长，仅 0.1 毫米宽的孔（也就是说，大大小于网状线的直径）。使用现在的机械钻是不可行的，这些钻的方向比率限制在 15 左右，激光烧蚀和离子铣削也是一样。然而，我推测将纳米技术和来自于石油工业的想法进行结合，就能使那样的一个钻最终成为可能。人们可以钻通防护网，并具有相当大的不损坏线路的可能性。
- 在揭开一些设备并明白其硬件的运行原理后，攻击者可能会充填聚能炸药以发送在 11.5 节讨论的等离子流到设备中，从而毁坏干扰检测线路和内存归零线路，当然，要在这些线路有时间反应之前。

这种攻击的成功与否是不确定的，因此很少会被罪犯用于进行攻击。

当我在 2000 年 9 月整理本书的初稿时，曾写下“因此，至今为止针对 4758 系统进行的最可能的攻击，涉及到的仅是逻辑缺陷的利用，而不是物理缺陷。”当我编辑此书的校样稿时，这件事情变成了现实。大多数 4758 的用户使用一个叫做 CCA 的应用程序，它在［388］中进行了描述，并包括许多令其正确使用很困难的特征。出于对其指令集复杂性的怀疑，我把这个系统的手册转交给了一个新的开发人员 Mike Bond，并询问他是否能发现一些易攻击点。到 11 月的中期，他发现了许多的问题，包括一个协议层次的攻击，它可以使一个有能力的对手从设备中提取出所有他感兴趣的密钥。本书将在下面讨论这个攻击。

最后，应该提到的是，安全处理器的设计和制造的主要限制与更普通的警报系统所遇到的问题非常相似。在错误报警率和漏发报警率之间有一个权衡。因而在安全性和健壮性之间有一个权衡。安全处理器经常需要小心处理；如果它们在 -20℃时会自毁，那就不能通过正常的计算机行业渠道进行分发，因为货物经常要面对冬天 -40℃的低温。震动、电源瞬变，以及电磁干扰也是一些设计的问题。军用设备制造商有着特别难的问题。例如，如果将军用作战电台的密码处理器暴露于辐射条件会使其自毁，那么对设备进行充分加固，又会使其太重而无法携带。

14.4　评估

在我们继续讨论较便宜的设备之前，让我们看一下依次列出的、有关物理防篡改设备发展的评估。

描述 4758 设计的前身，4753 的 IBM 的文档［4］，针对攻击者提出了以下的分类模式：

1）第一类攻击者——聪明的局外人——通常很聪明，但对系统的知识了解不够充分，他们只能访问中等复杂的设备。他们经常试图利用系统现存的弱点，而不是力图发明一些新的方法。

2）第二类攻击者——知识渊博的内部人士——有着丰富的专业技术教育和经历。他们对系统中部件的理解程度不同，但有潜在的机会进入绝大多数的部件。他们经常有高度复杂的分析工具和仪器。

3）第三类攻击者——有资金支持的组织——能够组织具有相关技术和技术互补的专家队伍，并由大财团支持。他们能够对系统进行深入的分析，设计复杂的攻击，并利用最先进

的分析工具。他们也可能招募第二类中的对手作为他们攻击团队中的一部分。

在这种模式里，4753 的目标是阻止知识渊博的内部人士，而其后继者 4758 的目标是（并已得到认证）阻止有资金支持的组织。

FIPS 认证体系由美国政府许可的实验室进行操作，并已发布 FIPS 140-1 标准。它发布四级保护体系，第四级为最高级（现在，仅有 4758 被认证为属于这个级别）。第四级和第三级之间有着巨大的差距，第三级仅要求封装。这意味着利用电磁泄露、内存剩磁、打钻、喷砂处理等的攻击者也有可能攻击第三级的设备。我曾经处理过一个经过第三级认证的设备，我用瑞士军刀就能将其封装刮掉！所以 FIPS 140-1 第三级的设备能够（并已经）被 IBM 认为的第一类攻击者破解，而下一个级别——FIPS 140-1 级别 4——却有望对付 IBM 认为的第三类的对手。对于 IBM 的第二类的防卫，在 FIPS 中没有相对应的级别。

由 IBM 工程师编写的有关评估级别的原始文件，曾推荐过六个级别 [796]；FIPS 标准采纳了前三个级别作为其 1～3 级，并将建议的第 6 级别作为其第 4 级。其中的差距，通常被称为“3.5 级”，是许多更好的商业系统所追求的。那样的设备当然试图完全屏蔽第 1 级别的攻击，并使第 2 级别的攻击很困难，第 3 级别的攻击者则将付出很高的成本。

这就是说，我不相信 IBM 的分级是正确的，我知道一个大的有资金支持的组织购买了芯片测试设备，并力图破解智能卡，但失败了；他们总结认为智能卡是完全防篡改的。然而，正如本书将要讨论的一样，许多智能卡已经被第 1 级别的攻击者破解。攻击者的毅力和狡猾，要远比这个组织工资表中人的数量重要得多。

14.5　中级——安全处理器

3.5 级别产品的一个好例子是 Dallas 半导体公司制造的 iButton 和 5002 安全处理器，以及用于保护美国军事通讯为保密级别的 Gapstone 芯片。4758 价值 2000 美元，而这些产品的价值仅在 10～20 美元之间，但是对它们实施攻击却远非那么轻松。

14.5.1　iButton

Dallas 半导体公司制造的 iButton 设计为一小型的、自包含的密码处理器，它有一个 8051 微控制器，具有模块化的指数电路，存储密钥和软件的静态 RAM，一个时钟，以及篡改感应器。它们被嵌入到具有锂电池的钢盒中，电池设计为能维持 RAM 中的密钥 10 年的时间（见图 14-3）。它足够小，从而可以穿在信号环中，或当做密码表带随身携带。早期的一个应用是作为“电子红盒”的访问令牌，电子红盒是为英国政府部长们设计使用的安全笔记本电脑系统。要访问秘密文件，部长们需要将他们的信号环放到笔记本电脑一侧的阅读器里（设计的标准之一曾经是“部长们不必使用口令”）。其他的应用包括 Istanbul 的公众传输系统、阿根廷的停车计费系统，以及美国邮政局的电子邮票，这些都曾在本书前面的章节中提及过 [753]。这个设备现在配备有 Java 解释器，并作为 Java 环进行销售，Java 环是一个用户可以为他们自己的应用而进行编程的篡改感应设备。

一个 iButton 如何会被攻击呢？它与 4758 最明显的差别是它缺少干扰检测屏障。所以当一个人试图从旁边钻孔进入其内部时，就能探测到正在工作的设备，并使篡改感应电路无效。因为 iButton 有帽状开关，可以探测盒子是否被开启，并且处理器颠倒着放在电路板上（在芯片的顶端金属层有一层网）。这不可能是个小的动作，它会涉及到建造自定义的夹具和

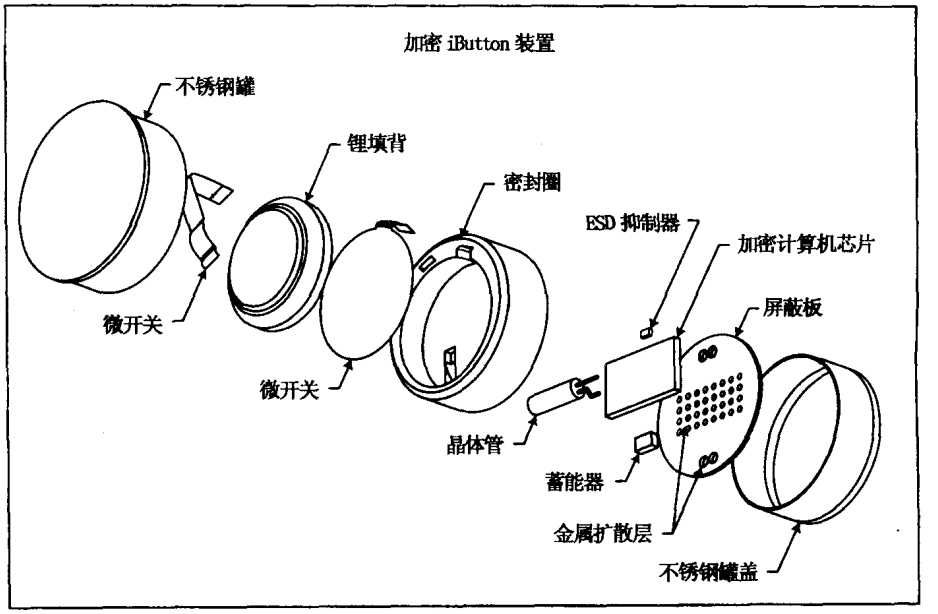


图 14-3　iButton 的内部结构（Dallas 半导体公司友情提供）

工具。简而言之，这是对另一些聪明的研究生具有很大诱惑力的目标，这些研究生想从成为一名硬件黑客的刺激那里获得胜利者的感觉。

14.5.2　Dallas 5002

另外一个来自 Dallas 的中级安全设备是 DS5002 微控制器，它广泛用于销售点终端等设备上，在这些设备上，保存有用于加密顾客 PIN 的密钥。

隐藏在这个设备中的精巧想法是总线加密。芯片已经添加了一些硬件，它们能在数据载入和存储时对内存地址和正在处理的内容进行加密。这意味着设备能以外部内存工作，并不限于只能装入到低成本的篡改感应包中的小量内存。每台设备有独一无二的主密钥，它在通电时随机生成。然后，软件通过串行端口载入，并被加密、写入外部内存，这样设备就可以使用了。电源必须持续供应，否则保存主密钥的内部寄存器就会丢失密钥；这在物理破坏事件被感应到时也会发生（和 iButton 相似，DS5002 也有干扰检测网，建立在芯片顶部的金属层上）。

这个处理器的早期版本（1995）成了 Markus Kuhn 设计的一个精巧的协议层攻击的牺牲品，即*密码指令搜索攻击* [477]。这个想法是一些处理器指令有诸如 I/O 的可视外部效果。特别是一个指令能使内存中的下一字节输出到设备的并口。这个手段是利用测试芯片对处理器和内存之间的总线进行拦截，并在指令流的某些点输入所有可能的 8 位指令，其中的一个指令应该能够解码为并行输出指令，并将下一个“加密内存”字节的明文进行输出。通过变化这个字节，就能建立起一张明文与密文相对照的表格。在使用这项技术得到七位和八位字节序列的加密函数后，攻击者就能够加密并执行一段短程序，从而将内存中的所有内容发送出来。

全部的细节还要更复杂一些。在发现该漏洞之后，Dallas 解决了这个问题，但 DS5002 仍是一个好例子：当试图首次实现一项新的构造精巧的安全系统时，有一些完全没有预料到的

事情会出错。

14.5.3 Capstone/Clipper 芯片

在 1993 年，当美国政府引入 Clipper 芯片作为 DES 的替代品时，整个安全世界为之震动。Clipper 也被称为第三者保管加密标准（EES），是一个防篡改的芯片，它实施 Skipjack 分组密码，使用一个设计为允许美国政府利用 Clipper 能够解密任何加密通信的协议。这个想法，是当用户提供给 Clipper 一串数据和对数据进行加密的密钥时，芯片返回的不仅是密文，还有一个法律强制访问域，或 LEAF，它包含用户提供的密钥，这个密钥用嵌入到设备中的密钥进行加密，并为政府所知。为了防止人们欺骗和伴随错误的 LEAF 发送消息，LEAF 有一个利用“家庭密钥”进行计算的校验和，并被所有可互操作的 Clipper 芯片共享。这项功能被延续到下一代的芯片中，称为 Capstone（压顶石），它集成 ARM 处理器以进行公钥加密和数字签名操作。

几乎在压顶石芯片刚出现时，LEAF 机制中就发现了一个易攻击点 [113]。加密的校验和用于将 LEAF 绑定到仅 16 位长的消息上，这使得往设备中装入随机的消息密钥，直到找出一个具有给定 LEAF 的密钥，成为可能，从而使得具有 LEAF 的消息，在送出去后不会被政府查看到。Clipper 的提议被放弃了，并被其他目标为控制加密“多样化”的策略所替代。尽管如此，Capstone 仍悄悄地进入了政府的服务中，并被广泛用于 Fortezza 卡中。Fortezza 卡是用在 PC 内，将数据加密到保密层次的 PCMCIA 卡。Skipjack 分组密码起初是机密的，但从那以后已经应用到公共领域里 [577]。

此处能引起人们很大兴趣的是它使用的篡改保护机制，因为它可能是单芯片防篡改设备中最复杂的，并在当时被声称为足以抵制“技术先进且受大额金钱资助的对手”[578]。尽管 NSA 声称 Clipper 芯片不是机密的，并可出口。但我还是不能够得到一块芯片以拆开它进行研究，尽管我曾经进行了多次的尝试。

它的后继产品是 QuickLogic（快速逻辑）军用 FPGA，它设计为使其使用者能消除来自于顾客那里的私有算法；它被广告宣传为“绝对不可能颠覆的工程师”。和 Clipper 一样，它利用 vialink read-only memory（vialink 只读内存，VROM）。在内存中，通过往芯片上的金属 1 层和金属 2 层间吹入 antifuse（反引信）来设定位。有足够高电压的编程脉冲用于往将两个金属层隔开的多晶硅中熔制出一条导通电路。更进一步的细节和显微图能在数据手册 [347] 中找到。

基本上有三种方法可以对 antifuse FPGA 进行反向工程。

- 第一个是利用光学或电子显微术来断定被吹入的 antifuse 的存在，首先要去除芯片的顶部金属层。这是极其乏味的；即便位被正确地读出，还是要做很多工作以搞明白这些位是什么意思。
- 一个更聪明的方法是故意不正当使用编程电路。它发送一个脉冲到保险丝，并在电阻下降时立即停止。电阻下降意味着金属已经融化，并建立起了连接；如果脉冲不停止，那么金属可能会汽化，并重新变成断路。这样，必须提供一个用于探测电路是开路还是短路的电路；如果它们不是在编程后充分地禁用，它们就可能用于读出设备中的数据。
- 最快速的方式，当执行的加密算法是已知时会特别容易，将微探测器直接放到门电

路阵列上，并查看信号。适当的分析技术，例如 15.4 节所描述的，应该可以迅速地得到密钥。信号也可以被收集到，通过利用电磁或电子—光学感应器、电压对比显微术和其他一些不断增加的芯片测试技术。即使算法在刚开始时不知道，从观测芯片信号重建算法也比做一个完全的电路重建要快。

这项技术不是没有错误的，如果明智地使用，则一定会有一些潜力。

14.6　智能卡和微控制器

如今最通用的安全处理器是智能卡和相似的自包含安全处理器。大批量购买时，它们的价钱可能是一美元。并正在配置到许多的环境中，诸如电话卡、付费电视用户卡，宾馆的房门锁，甚至（在某些国家）银行卡。

在这样的应用下，对手通常能获得许多样品设备，并将其拿走以随意进行探测。因此，许多针对它们的攻击已经得到了很大的进展。

尽管智能卡现在以“新”的安全解决方案推向市场，但事实上可以追溯到很久以前。早期的专利（可追溯到 20 世纪 60 年代晚期到 20 世纪 70 年代中期）早已过期 [247]。对于智能卡的发展历史，参见 [358]。很多年以来，它们主要用于法国，大多数的起步工作是由政府支持的，在 20 世纪 80 年代末期和 20 世纪 90 年代早期，它们开始在法国以外的国家大规模使用。主要用作 GSM 移动电话的用户身份模块（SIM）以及付费电视台的用户卡。

智能卡是自包含的微处理控制器，装备有微处理器、内存和一个集成到包装在塑料卡中的单芯片上的串行接口。用于银行业和旧式移动电话中的智能卡，使用标准尺寸的银行卡，而新式的、更小的移动电话使用的是尺寸小得多的卡片。当然，智能卡芯片也可以用其他的方式进行包装。例如，大多数英国预付费电子表使用包装在塑料钥匙中的智能卡芯片。Nagravision 付费电视的机顶盒也是一样。由美国政府使用的 STU-III 安全电话中，每个用户有一个加密引导密钥，它也被包装，并且在视觉和触觉上很像物理密钥。

使用智能卡的最广泛的应用是 GSM 移动电话系统，GSM 是由一些美国网络和美国以外的绝大多数国家使用的数字标准。电话手持设备是商品，并通过 SIM 卡来对每个用户进行个人定制。SIM 卡是张智能卡，它不仅包含个人电话本、拨叫历史等信息，也包含将用户认证到网络上的密钥。

使用便宜的智能卡来为比较贵重的用户电子设备提供认证和其他一些安全功能的策略，具有许多的优点。昂贵的设备可以大批量制造，这样每一个部件几乎完全相同；然而智能卡可以提供用户级的控制，能在一个成功的攻击发生时以相对快速和廉价的方式替代。这已经导致许多付费电视的运营商开始采纳智能卡。卫星电视抛物面天线和解码器已经成为耐久性的用户消费品，而每个用户能得到个性化的智能卡，它包含需用于解码他们已经订阅的频道的密钥材料。

芯片卡已用于许多的其他应用中，从宾馆钥匙到公用付费电话。尽管在那样的应用中，常见的卡不包含微处理器，而仅是一些 EEPROM 内存，用以存储一个计数器或认证，以及一些执行简单认证协议的逻辑。

诸如预付费电子计费系统的设备，一般构建在一个微控制器的周围，这个微控制器执行与智能卡相同的功能，但其保护措施较不成熟。这种设备通常设置一个“内存保护”位，以防止 EEPROM 中的内容被一个攻击者轻松地读出。在一些特殊产品上已有许多设计缺陷；

例如，称为 iKey 的计算机认证令牌，有一个主口令，它利用 MD5 进行哈希操作，并存储到处理器外部的 EEPROM 中，使用户可以利用已知口令的哈希操作进行重写，并完全控制这个设备 [459]。

许多低成本的安全设备都是基于某种微控制器（或执行某种认证协议的专用逻辑）。遥控器的数目在增加，其功能是作为一个无线电频率识别器，提供防盗功能，或仅对大量的产品进行“智能标识”。至于更多的系统易攻击性，针对智能卡的攻击也倾向于工作在基于微控制器的设备上，所以从这点上我不分开对待它们。有关针对微控制器攻击的更多细节，参见 [43]。

14.6.1 体系结构

通常的智能卡包括一个 25 平方毫米的硅制电路小片（die），包含一个 8 位的微处理器（例如 8051 或 6805），尽管一些新式的设备开始拥有 32 位的处理器，例如 ARM。它也有串行的 I/O 电路和一个三级的内存体系：ROM 用于保存程序和不可移动的数据；EEPROM 保存用户特有的数据，例如注册用户的名字、账号以及加密密钥、数值计数器等；RAM 寄存器保存计算过程中的临时数据。

用正常计算机的标准进行衡量，其内存是非常有限的。2000 年，在市场上出售的标准的卡，可能有 16K 字节的 ROM，16K 字节的 EEPROM 和 256 字节的 RAM。总线在设备以外是不可访问的；对外提供的连接仅是电源、重置、时钟和一个串口。关于物理的、电子的和低水平的逻辑联系，以及与文件系统类似的访问协议，具体说明在 ISO 7816 中。

14.6.2 安全的演化

当我第一次从智能卡销售商那里听说有关卖点时——在 1986 年，当时我是一家银行的高级职员——我询问如何知道这个设备是安全的。他向我保证，因为用来制造这个卡的机器价值两千万美元，正如制作纸币一样，这样的系统必定是安全的。我不相信这些话，但也一直没时间或工具来证明其声明是错的。后来我从行业执行经理那里听说，没有一个顾客准备为这个严格的安全机制付费，一直到 1995 年左右。所以一直到那时，他们总是依赖于设备的小尺寸、设计的模糊性以及芯片测试工具的相对不易获得性。

改变所有这些的应用是卫星电视。运营商将他们的信号广播到很大的范围——例如整个欧洲——并给用户智能卡，这些卡会计算密钥，用于解码用户的付费频道。因为运营商通常仅购买一到两个国家的电影播放权，他们不能在别处销售用户卡。这就产生了一个付费电视卡的黑市，在其中伪造的卡也可能在销售。另一个主要的因素是 Trek 星，欧洲人从英国卫星广播公司收听广播已有数年，但 Trek 星在 1993 年突然加密，这刺激了许多聪明的年轻计算机科学家和工程专业学生开始寻找其易攻击点。

从那时起，主要的金融欺诈都是以克隆卡的形式出现。首先被报道的是用于给葡萄牙农场主发燃料折扣的智能卡。罪犯与汽油站进行共谋，汽油站将其他的石油销售注册到伪造的卡中去，以获得一部分收益作回报。这个发生在 1995 年 2 月 ~ 5 月的欺诈案，据报道说已经赚到了大约 3000 万美元 [557]。

如何攻击智能卡（1）

最早的攻击目标是卡使用的协议。例如：一些早期的付费电视系统给每个顾客发一张卡

以授权收看所有的频道，然后在空中发送消息，删掉顾客在推广期后没有订阅的频道。这使一种攻击成为可能，它将一设备插入到智能卡和解码器之间，以拦截和丢弃送到卡中的一些信息。这样，订阅者能够删掉他们的订阅信息，而销售商却不能注销掉他们的服务。

同样的攻击发生在德国的电话卡系统中。一个叫 Urmel 的黑客告诫 Deutsche Telekom 公司，通过这样的攻击能使电话卡给予无限制的免费拨叫。他通过在卡和电话之间放置一台笔记本电脑来分析通信信号，从而发现了这个问题。Telekom 的专家不相信他，所以他利用他的知识，在夜总会和旅店销售手工制作的芯片卡，以免被发现 [726]。这样低成本的攻击，对于电话公司来说是特别烦恼的。因为采用智能卡的主要原因就是削减必须在线验证廉价令牌的成本 [78]。本书将在有关版权实施系统的章节中进一步讨论协议失败。也有许多全范围的标准计算机攻击，例如通过送出太长的参数串来进行栈重写。接下来，本书集中讨论针对智能卡的攻击。

如何攻击智能卡（2）

智能卡使用外部供应电源，并在 EEPROM 中储存安全状态信息，如加密密钥和数值计数器。所以一个攻击者可以通过移除程序电压 Vpp，来冻结 EEPROM 中的内容。早期的智能卡接收来自于主机接口上的专用连接的 Vpp。这引起了非常简单的攻击：通过用胶带覆盖 Vpp 接点，持卡人能阻止“删除”信号影响他们的卡。同样的技巧可用于一些付费电话芯片卡；在适当的接点上覆盖有胶布的卡有着“无数的单元”。

解决方法是使用电压倍增器电路，从供应电压 Vcc 内部产生 Vpp。然而，这并不十分安全，因为电路能被攻击者所破坏。所以一个谨慎的编程人员，例如，在一个用户输入错误的 PIN 后会使重试的次数减少一次，再将它读回来并进行验证。它也在每次卡复位时验证内存确实在工作，否则，罪犯就会破坏电压倍增器，然后重复地对卡进行复位，试验每一个可能的 PIN，一个接一个。

如何攻击智能卡（3）

另一个早期的攻击是减慢卡的执行速度，或在一个交易中单步执行卡的操作，通过不断地对它进行复位，并且记录它为 n 次，然后 n + 1 次等。在一张卡里，在工作内存没有归零时，就可能在复位后用适当处理读出 RAM 内容。对于很多的卡来说，有可能利用电子显微镜读出芯片表面的电源（一般大学所能拥有的低成本扫描电子显微镜，不能做超过数千千赫的电压对比显微处理，因而需要降低执行速度）。

现在的许多智能卡处理器，有检测低时钟频率的电路，低时钟频率可以冻结或重置卡。但是，和防盗报警一样，在错误报警率和丢失报警率之间有着权衡，这导致许多由智能卡芯片制造者提供的报警特征不能被 OEM 或应用开发人员使用。例如，利用便宜的读卡器，当卡被加电时，在时钟频率上有一个剧烈的波动，会引起很多的错误警报，这导致一些开发人员不使用这个特征。很明显，低时钟频率探测仪需要仔细地设计。

如何攻击智能卡（4）

一旦付费电视运营商修正好了绝大多数的简单攻击问题，盗版者开始转向使用物理探测器进行攻击（见图 14-4）。大多数智能卡没有对线路的显微镜尺度以外的物理破坏进行保护。例如：芯片表面的一层薄玻璃钝化层，以及通常用某种环氧树脂封装。用于解包芯片的技术是众所周知的，并在标准半导体测试中作了详细的讨论，如 [80]。在大多数情况下，一些军用的发烟硝酸就足以溶解环氧树脂；这样，钝化层就能在需要用到探测的地方被去掉。

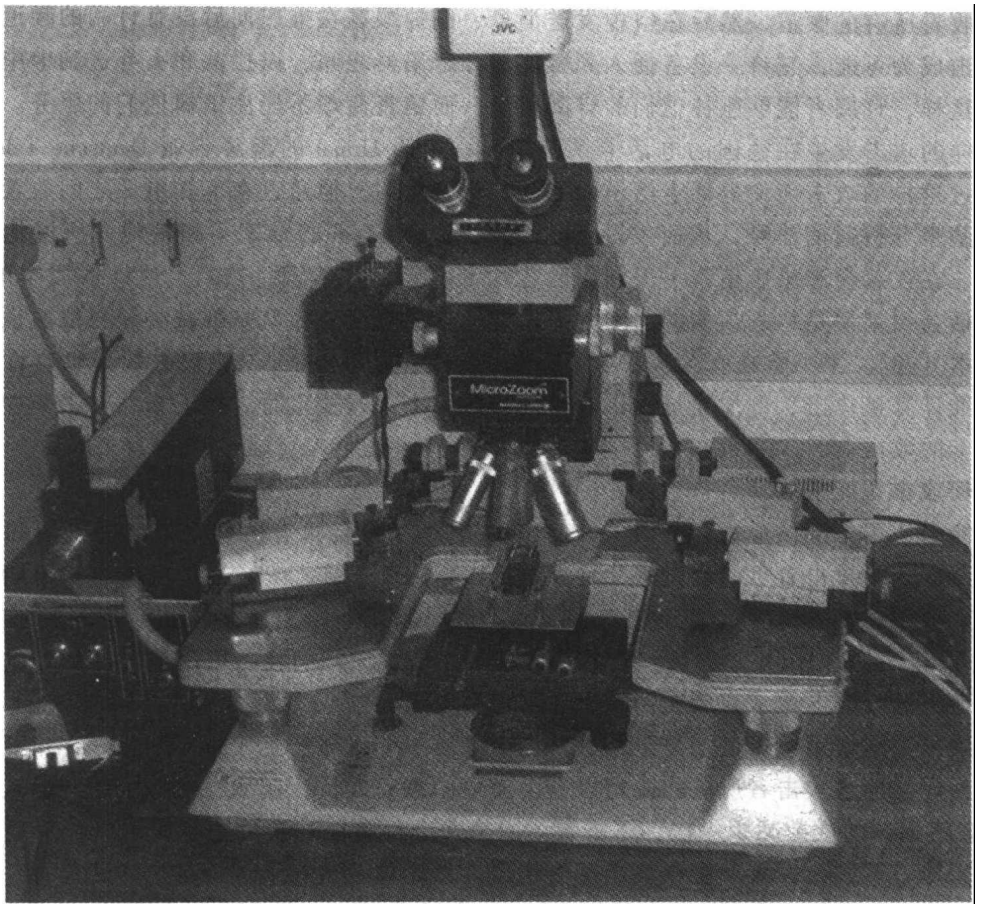


图 14-4　低成本的探测平台

探测站包括显微镜，它具有在芯片表面上实施精密探测的微控制器。它们广泛用于半导体制造工业，用于对生产线上的样品进行手工测试。它们能从二手市场以低于 1 万美元的价格买到，有着特殊的附件，例如激光仪，用于在芯片的钝化层（见图 14-5）中钻洞。

通常，一个探测攻击的目标是处理器的总线。如果总线通信能被记录下来，就能对程序的运行进行跟踪，包括编码和数据。如果攻击者幸运的话，卡的设计者将会在卡复位后于内存中计算一个校验和（这是一种推荐使用的防御行为）。并且这项操作会立即将卡片内存中内容的全部列表呈现给攻击者。所以攻击者会识别出总线，并将总线暴露以供探测。

付费电视卡行业采用的应对这种攻击第一个防护措施，是对每个卡赋予多个密钥和/或算法，并且安排事情，使得仅有正在使用的那些数据会出现在处理器的总线上。当盗版卡出现在市场上，可将一条命令经空中传播，使合法卡的用户从以前未使用的内存区域，激活新的密钥和算法。通过这种方式，盗版卡的用户会蒙受许多的服务损失，一直到探测攻击能够重复和能够分发一些新的盗版卡或现有盗版卡的升级版。

如何攻击智能卡（5）

击败这个防护策略的是 Oliver Kömmerling 的*内存线性攻击法*，借此，分析员采用一种使指令变得不可操作的方法，损坏芯片的指令解码器 [470]。这些指令改变程序地址，而不是增加地址——例如跳或调用。这种攻击的一种方式是在控制线上放置一个接地的显微镜探测

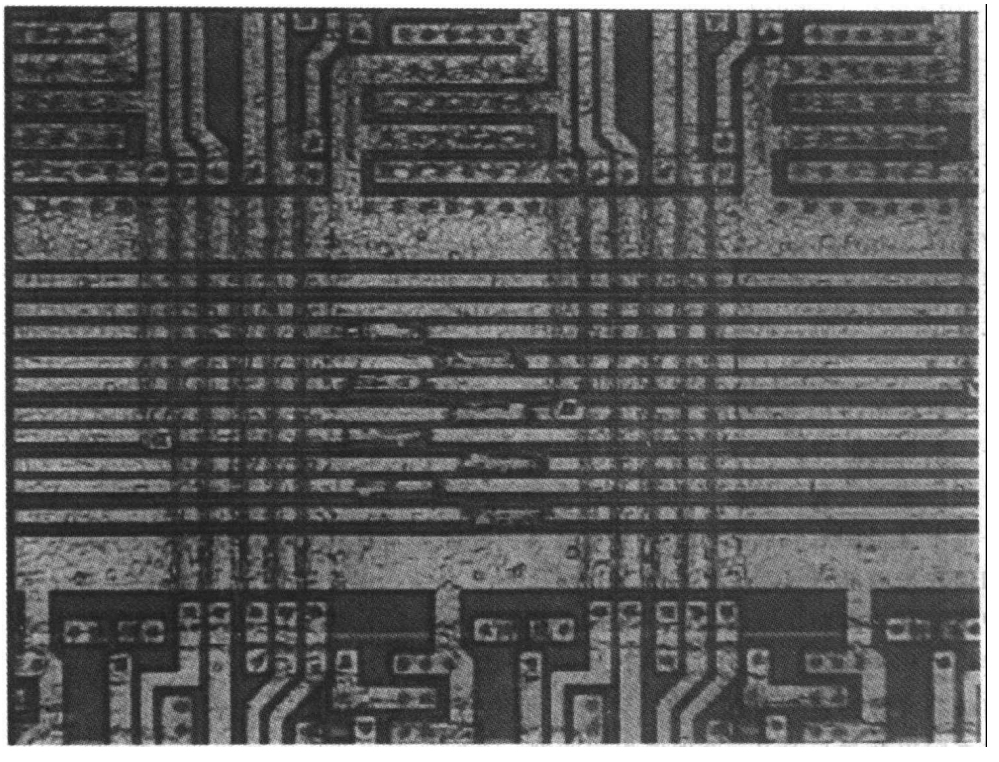


图 14-5　用激光刻蚀技术在钝化层上蚀刻八道沟，
以用于探测的 ST16 智能卡数据总线（Oliver Kömmerling 友情提供）

针，并连接到指令寄存器。这样，在通电时恰好运行的指令会重复执行。内存中的内容也就能从总线中读出来。事实上，一旦设备的一些 ROM 和 EEPROM 被弄懂，攻击者就可以跳过不需要的指令，并使设备只执行他挑选出来的指令。所以利用单个探针，他就能使智能卡执行任意的代码。并且，理论上，他还能使智能卡在串口输出其保密的密钥材料。但是，从总线中探测出内存的内容要更为容易。

实际上，在指令解码器中常有几个位置，接地的探针有阻击控制流被编程改变的效果。所以即便没有完全弄懂处理器，内存线性化也常能通过试验和错误获得成功。一些更现代的处理器有防止内存线性化的自陷电路，例如：硬件访问控制矩阵，它能防止内存中特定区域的内容被读出，除非一些特定序列的命令被呈交。但是那样的电路常常可以通过使用激光束或电子束击穿仔细选择的门电路而被击败。

可以通过卡的试验电路对卡进行攻击。典型的智能卡芯片在 ROM 中有一个自检例程。它在工厂中执行，并允许内存中的所有内容被读取出来以进行验证。在此之后，一个多晶硅保险丝被吹制到芯片上，以阻止攻击者使用同一措施。攻击者需做的所有事情是找到保险丝并修复它——它可能只涉及到利用两个探针做些连接的小工作 [130]。这样，在一些情况下，整个内存中的内容能从串口中读出。一个更为仔细的设计可能会将试验电路放在硅的一部分上，这些硅会在晶片被切割到单独的芯片中时被锯掉。

如何攻击智能卡（6）

付费电视卡行业力图做的另外一件事情是集成硬件加密处理器，以迫使攻击者重建硬件电路，而不是简单地克隆软件，并迫使他们在盗版卡中使用昂贵的处理器。在第一次这样的

实施中，加密处理器是包装在卡中的分离的芯片。这个设计有一个有趣的协议失误：它会一直计算出用以解密正在进行的视频流的密钥，然后将它传送给 CPU。CPU 再决定是否将视频流释放到外部世界。黑客们通过在两块芯片之间接入分支线路的方法破解了这个系统。

更现代的实施方案将加密硬件建造在 CPU 内部。在那里，它包括几千个门电路。对于一个攻击者来说，从芯片的显微图对电路进行手工重建是可行的。但是门电路数如此之多，而亚微过程又非常深，所以一个成功的攻击者可能需要自动的线路图重建：成功刻蚀掉芯片上的层，采用电子显微图，并利用图像处理软件重建芯片的 3-D 图，或者至少鉴定出它的组成单元 [121]。然而，组装所有的设备、写软件，以及整合系统都需要投入很大的精力和开销。

一个更简单也很常见的攻击，是攻击者为伪造请求现有的十多个商业反向工程实验室中的一个，来重建芯片的相关区域。这些实验室为芯片制造商的竞争对手们分析商业集成电路，寻找可能的专利侵犯以获利。他们习惯于工作在具有一定保密程度的环境中，并且攻击者似乎也不是很困难就能潜入到样品中。这个样品的目标是盗版，而不是诉讼。

如何攻击智能卡（7）

智能卡行业发明出来的另外一个防范措施，是给芯片装上保护性表面网格，它放在顶部金属层中以作为地线、电源线和感应线的螺旋图案。这个想法是一旦芯片通电启动，图案中的任何破裂或短路就会被立刻感应到，从而引发自我销毁的机制。

本书曾提及过与 Dallas 处理器相关的网；在实施失误造成的最初的遗憾后，它们被证明是提高攻击成本的一种有效方法。击败它们的合适工具是聚焦离子束工作站（Focused Ion Beam workstations，FIB）。这是一个与电子扫描显微镜相类似的设备，但它利用的是离子束，而不是电子束。通过变换离子束电流，就可以把它当作显微镜或铣削机器使用。通过引入适当的气体，它被离子束照射而引起化学变化，就有可能以数十纳米的精度放置导体或绝缘体。

FIB 在所有种类的应用中是极其有用的设备——从半导体测试到冶金和法庭辩论，一直到纳米技术。这种设备正迅速变得到处可见，并且价格也在迅速下跌。许多大学和工业实验室现在都拥有 FIB。FIB 的时间也以每小时数百美元的租金可以从许多机构租借到。

给定一个 FIB，就可以在它没通电时直接攻击其感应网。人们可以简单地钻孔，到达载有想得到信号的金属线，然后用绝缘体填充上。通过绝缘体的中心又钻另外一个洞，随后用金属填充，并在其顶部镀上一个接点——典型地为数微米宽的铂 L 或 X，它很容易与来自于探测站的针相连接（见图 14-6）。

击败总是通电的感应网要困难得多，但是所需要的工具正由芯片测试工业的实验室开发出来。例如，有一些技术可以磨穿装有合适 FIB 的芯片的背面，并使接点直接与电子设备连接，而一点也不会干扰感应网。

许多其他的防护技术能够迫使攻击者做更多的工作。一些芯片据说是包装在更厚的玻璃中，而不是钝化层中。这个想法是移去这个保护层的明显方式（例如应用氢氟酸）可能会损坏芯片。然而，过去几年中发展起来的光电技术能使攻击者利用激光直接读出电压 [11]。其他的芯片具有诸如碳化硅或氮化硼物质的保护涂层（具有保护涂层的芯片展示于 Fort Meade，Maryland 的 NSA 博物馆中）。这样的涂层能使 FIB 操作员进展缓慢，而不是通过内建的电荷来损坏芯片。然而，在智能卡芯片包装中的保护层，和安全行业中的许多其他技术一

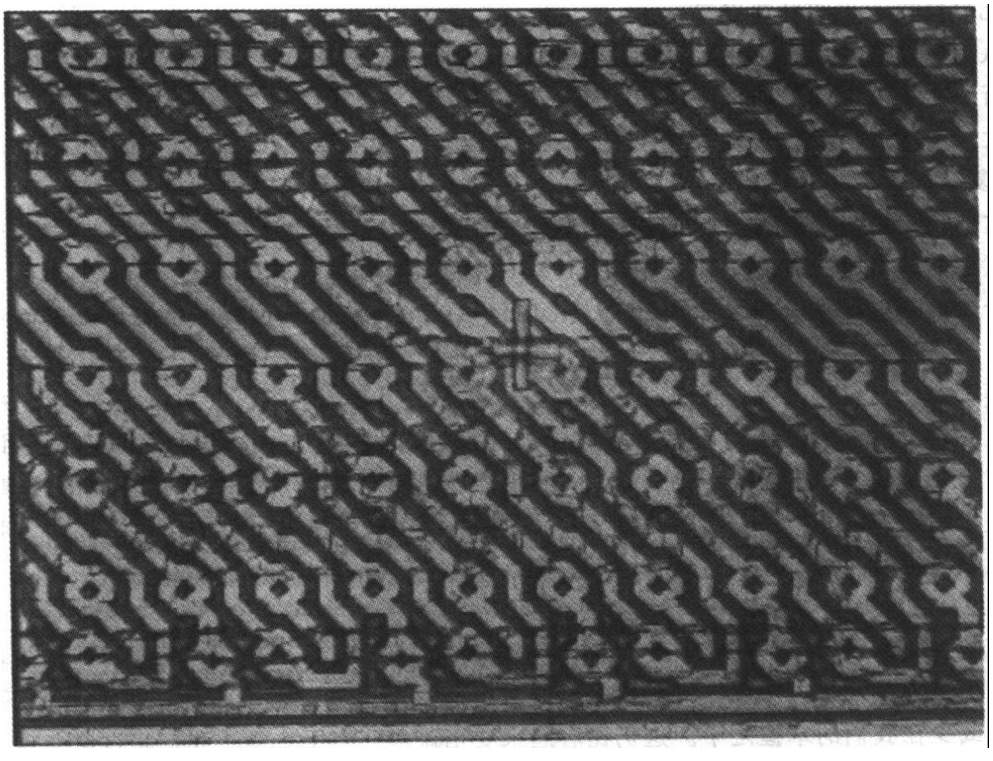


图 14-6　ST16 智能卡的保护网，智能卡带有 FIB 叉点，
用于探测在底下可视的总线（Oliver Kömmerling 友情提供）

样，经常是一种市场的事情，而不是工程的事情。我们的团队最近拆卸了一个芯片，它的销售商声称其拥有一个保护层，但结果却证明，根本就没有什么特殊的保护。

14.6.3　技术现状

在写此书的时候，我知道还没有任何技术或技术的有效联合，能使智能卡抵挡住技术娴熟、性格坚定的攻击者的攻击。一些行业专家甚至认为芯片尺寸包装中的绝对保护仍是不可能的，因为制造一个不可测试的设备是不经济的。

尽管如此，智能卡还是要比磁条卡难以复制得多，并且还有一定的防护余地，可以将智能卡做得更牢固。最新的卡有多达三层的防护网；使用动态逻辑的寄存器会使击毁低时钟频率探测器成为不可能，只能单步执行这个芯片；一些电路不时插入虚假指令，这样，如果你一个接一个地探测总线，就必须做大量的工作来排列得到的探测结果；32 位的处理器，会使探测排列变得更困难；专有的指令集，以及许多其他的技巧都使攻击难以得手；但是行业内部人士说："具有离子束的人总能进来！"

因此，如果你正在设计一个依赖于智能卡的系统，你能采用哪种策略呢？

14.6.3.1　深度防护

付费电视公司使用的方法首先就是深度保护。智能卡可能结合所有以上描述的技术，甚至模糊专利的加密算法。通常，使用自制的加密方案是一件坏事情：Kerckhoffs 原则几乎总能在最后胜出，而一个坏的方案，一旦发布，就会是毁灭性的。付费电视的深度防护提供了有趣的例外，其目标是尽可能减少快速探测攻击的可能性，并迫使攻击者陷入到对整个系统

进行充分反向工程的麻烦中去。

认为即使是业余人员也能将探针放到设备的多个信号线上是一种先见之明。如果设备正在执行一个人所周知的加密协议，并具有很容易弄懂的算法，那么除非存在有效的机制能够引入大量的虚假指令，否则对单个总线进行跟踪就可能使密钥泄漏出来［370］。利用专有（并且复杂的）加密算法，能迫使攻击者去作更完全的分析，并使他延误好几周，甚至几个月。这能使诸如付费电视等行业中的盗版经济产生很大的差异，这些行业需要每年更换一次卡片（当然，由有能力的专家对专利设计进行彻底的评估是非常必要的——并且对于专家来说，分析的不仅是算法抽象的加密能力，也包括这个算法能从可观测信号中恢复的容易程度）。

尽管如此，付费电视公司自身所具有的技术措施仍是不够的。在过去的20世纪的最后几年里，付费电视公司成功地迫使盗版损失从超过总收入的5%减少到可以忽略不计的比例。许多复杂的智能卡起了一定的作用，但是很大一部分的成功来自于反盗版的法律行动，以及技术措施和法律措施的有效结合。本书将在第20章中作进一步的论述，那时，我们要对版权领域进行探讨。

14.6.3.2　防篡改和证明篡改

对防篡改的设备及证明篡改的设备作一仔细的区分，通常是非常有用的。即使前者还不能制作出来为大众市场所用，针对智能卡的攻击是攻击性的，例如探测，因而会留下证据，却或多或少在我们的掌握之中。这仍比看起来要困难——在下一章，我们将讨论非攻击性的进攻。

例如，在银行应用中，智能卡常用于制造和验证电子支票。银行可能有这样一条规定：仅当顾客能提供未受损的卡时才会考虑其提出的争议。这不像它看起来那么简单，因为智能卡总会在无意中损坏。签发给公众的卡每年可能会有1%因材料失效或静电而损坏。如果事情真的发生时，许多国家的消费者法律可能不允许银行不理会这些声明。另外，问题的法律和工程方面还会相互作用。尽管如此。证明篡改的（以及很难探测的）卡仍是风险管理策略的有用部分。

14.6.3.3　行业损失

一个系统朝深度防护的方向发展，还是朝证明篡改的方向发展，将取决于该系统能限制在卡被成功探测后所引起的损失的程度。

在早期的付费电视系统中，系统体系结构迫使所有消费者的卡包含同样的主密钥。一旦这个密钥被人所知，盗版卡就可随意制造，这时卡基必须被替代。现在已配置为使用数字广播的付费电视系统针对不同的卡有不同密钥的加密协议，因此克隆卡也就没什么用。我将在20.2.4.5小节中描述这些协议。

在其他的系统中，例如在2.7.1节中描述的银行卡应用，使用偷盗或伪造的卡所能消费的金额有一定的限制，这是由系统零售商的最低限制、随机的在线授权、热卡黑名单等所设定的。即使一张相对容易伪造的卡也可能是行得通的，因为它仍比替代的磁条卡要难以伪造。

14.7　哪里出了问题

有些系统的失败模式涉及到防篡改的处理器，它或多或少独立于设备是低端还是高端。许多失败的出现是因为设备暴露给了更有能力的攻击者，这些攻击者超出了它的设计者的预

料。早期芯片卡的设计者似乎没有想到罪犯会有机会接触半导体测试设备。更多失败的出现是因为人们保护了错误的事情，或以一种错误的方式保护了正确的事情；由一商业评估实验室发现的许多缺陷表明，这些缺陷大多存在于物理的、逻辑的以及组织的措施之间的接口上[131]。

14.7.1　体系结构错误

技术被错误使用的一个例子是使智能卡作为数字签名的首选设备。一些政府立法提案给予获准使用的智能卡所做的签名以更高的法律效力。这对于制卡行业可能是一个“充分就业法案”，而在技术上却没有多少意义。

在前几节描述的设备中，没有一个设备具有真正可相信的用户接口。一些银行安全模块在其前端设有一把物理锁（或两把），以确保仅具有金属钥匙的人才能执行某特权交易。但是无论你使用 2000 美元的 4758 或 2 美元的智能卡做数字签名，你仍得相信驱动它们的 PC 机。如果它给你显示一行文字“请付给 amazon.com37.99 美元，以得到 Anderson 的《Security Engineering》一书的拷贝”，但真正发送给签名的消息是“请抵押我在 Acacia 大街 13 号的房子，并将收益付给 mafia 不动产公司”，这样，防篡改也不能给你带来多少的安全。

它可能使你的处境更加糟糕，因为你将有一段更困难的时间来取消交易。信息策略专家已经指出，数字签名的建议接入方法，可能会破坏给消费者在网上进行电子商务带来信心的消费者保护法[124]。消费者真正需要的是安全 PC——或至少一个防火墙，以将他们的 PC 与外界最坏的威胁屏蔽开，例如恶意的代码。这是一个分离的工程问题，并与硬件安全没有什么关系。事实上，研究人员正在意识到，掌上电脑可能是进行数字签名应用的更好平台；无论对于探测攻击来说，其脆弱性如何，顾客至少能看到他们正在签什么，并防止设备用于一般的用途[69]。

一个硬件保护技术更为适当的应用例子来自于预付费电子计费系统，这在第 8 章中已经讨论过。这里，防篡改的功能是当销售计费卡的自动销售机被偷之后，对其可能引起的损失进行限制。通过在安全处理器中保留用于加密卡的密钥，就有可能对每一个销售者实施一项信用限制。处理器中也包含一个数值计数器。如果某些人成功地进入了设备中，他们就有可能击败数值计数器，提取出用于单个计费器的加密密钥。但这不会损坏整个计费系统，仅是迫使数千个计费器重新设定密钥。

14.7.2　模糊性和评估错误

许多已经被真实欺诈的方式破解的智能卡系统，似乎早已变得容易攻击，因为它们的操作者不能完全明白其技术和限制。这几乎并不令人吃惊；直到最近，还没有得到有关智能卡如何被攻击的发布信息。这个行业也试图对有关其产品的所有严格的技术信息进行保密。到今天，一个人必须签订一项不泄露协议，以得到智能卡的正确软件开发工具（有 Java 卡、基本卡等等，但这些用解释型语言将硬件和开发者屏蔽开的卡，并不支持用户在设备上运行他们自己的机器码）。

事实上，用于在普通标准体系下评估智能卡的安全目标，集中于维持设计的模糊性。芯片掩码必须是保密的，员工必须是经过审查的，开发者必须签订不泄露协议——许多的保密需求提高了这个行业的成本。模糊性也是出口获准的一般需求，但仍有人怀疑，它掩盖了一

些蓄意插入的易攻击点。例如，我的同事测试的一张卡，在命令其产生私有/公共密钥对，并且输出其公共部分时总是产生同样的数值。

当然，模糊性在大多数的智能卡应用中对顾客没什么用。几乎没有一个真正的、针对实地装配的智能卡攻击，会使用系统内在的信息。它们中的大多数从对一个在零售市场上买到的卡进行探测攻击开始。

更好的防护目标由 VISA 发布，VISA 指定广泛的浸透测试 [777]。然而，因为没有一个现有的产品能通过该测试，这个行业采取的路线是，保护它能做什么，而不是它应该做什么。本书将在 23.3.3 小节讨论基本的经济和政治利益时会讨论这个主题。

14.7.3 协议失败

如同安全工程中其他方面一样，在技术层次上最广泛的失败之一是使用不合适的协议。一个诸如 4758 的设备有一套几百个“动词”的执行命令集，或与加密操作的联合。加密操作可在传送到设备的数据上执行。更多的动词能被应用开发者定义。如何能保证这些动词的一些联合不会使一个用户做出有损安全策略的事情呢？

从 1981 年左右开始，到 1991 年，出现了一个协议攻击，它针对许多银行用以管理 ATM 网络的安全模块。这些有缺陷的设备在本书出现的时候应该全部退出历史舞台了，因为一个安全模块的工作寿命大约是 7 年（但是它们完全破坏了在 20 世纪 90 年代早期发生的梦幻提款诉讼过程中许多银行家做出的声明——“没有任何事情可能会出错”）。

由 VISA 和与 VISA 兼容的销售商——例如 Racal——提供的安全模块有一事务处理，用于产生一个密钥组件，并把其明码值打在附属的安全打印机上。它们也返回其值到调用程序中，调用程序用一个保存在防篡改硬件中的主密钥 KM 加密。

$$\text{VSM} \longrightarrow \text{打印机：} KMT_i$$

$$\text{VSM} \longrightarrow \text{主机：} \{KMT_i\}_{KM}$$

联合众多组件中的两个来产生终端密钥的另一个程序为：

$$\text{主机} \longrightarrow \text{VSM：} \{KMT_1\}_{KM}, \{KMT_2\}_{KM}$$

$$\text{VSM} \longrightarrow \text{主机：} \{KMT_1 \oplus KMT_2\}_{KM}$$

其想法是为了首次产生一个终端密钥，需要两次使用这些事务处理中的第一项，接着是第二项。然后，就有了 $KMT = KMT_1 \oplus KMT_2$。然而，没有任何事情能阻止程序采用旧的加密密钥并在第二次事务处理执行时提供两次，从而产生已知的终端密钥（所有位均为 0 的密钥，因为这个密钥是同自身异或的）：

$$\text{主机} \longrightarrow \text{VSM：} \{KMT_1\}_{KM}, \{KMT_1\}_{KM}$$

$$\text{VSM} \longrightarrow \text{主机：} \{\text{KMT}_1 \oplus \text{KMT}_1\}_{\text{KM}}$$

这个模块也有一个需要使用两次密钥的事务处理。这些密钥在主密钥下加密，并返回用另一个密钥加密的密钥。

$$\text{主机} \longrightarrow \text{VSM：} \{KMT_1\}_{KM}, \{KMT_2\}_{KM}$$

$$\text{VSM} \longrightarrow \text{主机：} \{KMT_1\}_{KMT_2}$$

（其目的在于允许提款机的终端主密钥被替换，或把 PIN 密钥送到用终端主密钥加密的提款机以支持离线 PIN 验证）。

现在攻击就很简单了，这也是破坏性的攻击。拥有一个零密钥，并在 KM 下加密，我们可以转换 PIN 密钥（以及感兴趣的任何东西），从在 KM 下加密变换到在零密钥下加密。这样，向顾客承诺的物理保护会是一个幻影：一个程序员利用两条无特权的指令，就可以抽取出任何一个感兴趣的密钥。

从科学的观点来看，这是非常有趣的。因为由 VSM 实施的安全策略是一种不完善的系统，它介于多级策略（“PIN 是机密的，并且决不能泄露给具有较低密级的进程”）和共享控制策略（“单个银行职员不能够算出客户的 PIN”）之间。从公众策略的角度出发它也是重要的，因为在 9.4.3 节描述的 Munden 案发生时，设备销售商就已经知道了。但是销售商并不承认它，尽管有这样一个事实：它的存在会直接损坏被大力宣扬的误判案的起诉证据。无论法庭案件中的任一方想要依赖销售商保证系统的能力时，就需要记住这一点。

采用的解决方案是去掉具有损害性的指令。这意味着双重控制密钥的管理，现在已涉及到主机上的可信进程，这个进程将有机会访问到密钥材料（这一直是 ATM 所支持的应用，CCA，提供给 4758 的案例）。一个更好的解决方案是利用哈希函数计算终端密钥，例如 KMT = $SHA1$（KMT_1，KMT_2），但这不是向后兼容的。根据我的后见之明，将数学特性与函数进行联合的选择，意味着后来在此基础上发展起来的协议，应该以这些特性可能会被用来搞破坏的方式进行检查。换而言之，联合函数的选择提高了事务集验证的复杂程度。

现在我们讨论一个 Mike Bond 在 4758 上发现的攻击点，它利用中等程度的密码搜集，就能从设备中取出任何一个密钥。易攻击点是双密钥，由 4758 内部使用的三重 DES 加密，能够使其密钥对被切断和接合。给定已知的密钥对 KA 和 KB，以及目标密钥对 KC 和 KD，一个人能将在叠接密钥（KC，KB）和（KA，KD）下加密的结果与对所有的可能性进行强力搜索的结果进行对比，从而一次攻破一个部件的目标密钥。它使攻击的成本从一个标准三重 DES 密钥搜索的 2^{112}减少到单一 DES 的易处理的 2^{56}。这也有可获得的时间—内存权衡；例如，具有 2^{16}试验密钥，如有可能破解 4758，需进行大约 2^{40}次测试加密。对于所有的相关细节，参见 [125]。

让我们回想一下，考虑其中隐含的意思。IBM 花费超过 10 年的时间来发展具备高度策略的产品，它被许多银行使用，以保护价值巨额美元的交易。这个产品被政府认定为是最安全的密码处理器（民用购买可得到），并对其出口进行管制。IBM 进一步保护它，并拒绝出售给我们样品，然而剑桥大学的一个研究生在六周内，通过研究 IBM 网站上的帮助文档就攻破了它。

验证加密处理器事务集的正确性是一项困难且仍未解决的问题。验证一个协议是非常困难的，研究界花费了 20 世纪 90 年代的大部分时间来了解如何做到这一点。一条协议可能只包括 2～5 条消息，而密码处理器可能有数十到数百个动词。许多协议失败了，因为它们的目标不明晰；而密码处理器是作为普遍用途的机器出售，并可能用于实施范围广泛的安全策略。我们还不知道如何将安全策略形式化，更不用说将它们向前追踪到加密原语了。检查没有一些模糊的交易序列能够破坏你的安全策略，是非常困难的；如果策略也不是精确陈述时，它看起来就是不可能的。

14.7.4　功能蠕变

我曾经也给过几个有关功能蠕变的例子，一般环境条件会改变，以及通过破坏安全系统

设计的假定条件而破坏系统。现代加密处理器的灵活性已使其成为一个特殊的问题。

功能蠕变也会与物理的防篡改特性直接相互作用，而这在智能卡应用中显得特别有害。拥有与磁条卡行为非常相似的智能卡，并且在每晚都会清账的银行账户上执行交易的系统，可以很容易地转移到拥有与钱包功能相似的智能卡，并可将钱相互转移的系统。在前一个系统中，不同卡有着不同的密钥，且只与银行共享。因此使单张卡泄密所要做的平均投入，不会超过磁条卡领域中信用卡的复制投入。而在后一个系统中，每张卡有一个密钥，它能使钱转到任何一张其他的卡中，并且集中的账目清算的限制也会松弛得多。一个相对低风险的环境突然就变成了一个相对高风险的环境。

另一个能将低风险环境变成高风险环境的方法是对同一张卡实施多个应用。如果一个以前仅提供健康保险卡或福利申请卡的设备突然变成国家身份证卡，那么它就会吸引更高级别的攻击者。如果大量不同的组织能在卡上运行他们自己的应用程序——这在 Java 卡的目的中已经陈述过——那么在第 2 章描述的选择协议攻击就会成为一种特别危险的威胁，一个罪犯可能设计出专门针对你进行攻击的应用程序。

14.8　什么应该受到保护

许多技术包括蒸汽机、电话，以及计算机——这些设备最初建议的用途，并不是最终显现出的用途（考虑如今市场的规模，如将水从煤矿中抽出来；将文本读给电报员，而不是通过气动导管发送；以及制定出火炮规程表）。

现在流行的使用智能卡的卖点是它们将成为 EU 电子商业规定中构想的高级电子签名设备——也就是说，人们用于签订法律文档的设备，它将是十分的可靠，因而那样一个签名的存在可当作拥有该设备的用户已经签过名的证据。与明显的法律客体很不同（它将证据的承担者从依赖签名的团体转移到设备的拥有者，并且那些设备总能被偷走）。正如以前提到的，存在一些技术问题，如用户不知道智能卡正在签什么。并且如果 PC 软件——它提供要被签名的材料——被保证为无缺陷和无病毒的，那么智能卡会添加什么价值呢？

这个行业已经花费了 20 世纪 90 年代的大部分时间来推销多功能卡的概念。这个多功能卡会替代许多人随身携带的塑料卡和金属钥匙。使其最终发生的应用可能会将银行交易过程放在移动电话上。当移动电话有用于精确地放置智能卡的插槽，那么一个银行就必须从手机网络运营商那里租用卡上的空间。我们将会看到这一事件的发生。

防篡改设备真的能添加什么价值呢？

首先，它们能控制信息处理，通过将它与单个的物理卡联系起来。一个付费电视订购卡能在黑市上买卖，但只要它不是复制的，电话运营商就不会太关注。另一个例子来自于 Dallas 产品，它用于食品工业的质量控制：它被物理封印到食品运输设备上，以提供对温度历史的可靠记录。而另一个是使用密码来在政府网络中实施评估标准：一旦你的系统经过检查并是可信任的，如果你只得到密钥材料，那么就不能轻易地将未注册的、任何规模的系统连接到机密的政府网络。

第二，防篡改设备能保证数据在确定和可验证的时间上被破坏。针对微软的反信任案已经使那些通过扣压 Email 文件传票所带来的损害大大曝光；许多公司喜欢实施这样一个策略：任何一封 Email 在一个固定时间之后就应该被销毁。除非发送者或接收者采用积极的行动来保留它。在我所在的大学里，例如，所有考试的笔试卷和主考者的工作笔记在四个月后

就会被销毁。如果我们保持这些材料过久，就不得不在数据保护法律的规定下，提供给学生访问。但是如果我们很快就销毁了它们，我们就会无法应对这样一个申请。一旦所有的事情电子化，实施这样一个策略将会使我们所保留的所有文件备份复杂化。一个解决方案是用密钥对文件进行加密，并将密钥保存在一个在适当的时候可以通过编程清除密钥的设备中。

第三，这些设备可以减少对人工操作员信任的需要。我记得它们在政府系统中的主要目的是“将密钥材料的街头价值降到零。”对于 STU-III 的加密引导密钥，只有小偷假装为有权利的拥有者，并且仅当他有机会接触到真正的 STU-III 电话，并且密钥或者电话没有发现被偷走的情况下小偷才能获得真正的成功。同样的想法可应用在 ATM 网络中，没有银行想让他们自己顾客的安全依赖于对另一家银行职员的信任度。

第四，防篡改设备能够用于控制数值计数器，正如在 14.7.1 节讨论的预付费电表。它们一般使用诸如 DS5002 或 iButton 等设备来保存用于当地计费系统和授权计数器的自动售货密钥，即使设备被盗，所能买到的电子卡的总价值是有限的。

有一个看起来似乎是一个更广泛应用中的特例，在该例子中，中央服务器进程中的一部分被委托给一个当地的设备。但是我能想到的最引人注目的例子与价值有关。注意，一个有警卫的合作数据处理中心也是一个防篡改的处理器；这种典型的应用经常能被这样一个事实体现，也即它们能集中地实施，如果存在完全可靠的网络的话。例如，如果所有的电子计费系统和销售站都是在线的，那么预付费电子计费系统就能够用直接认证的消息来完成工作。但也应注意，这个委托发生在合作数据处理中心之间，当银行使用“热卡”列表来认证其他银行卡的交易时。此处，防篡改设备可被用于提供特别的保证（尽管通常日志机制在操作方法的持续偿付能力可依赖时是充分的）。

这里给出的是一个不完全的列表，但是这些应用的共同点是提供了独立于周围计算机环境的可信度安全特征。换句话说，当使用防篡改设备力图对可信用户接口的缺少进行补偿时，必须小心。这并不意味着在接口有问题时，完全不会添加什么价值。例如防篡改加密模块（用于 ATM 网络）不能阻止使用 ATM 的小规模盗窃；但是它们能阻止大规模的 PIN 损害，如果使用正确的话。一般来说，防篡改设备经常是一个有用的组件，但只是很少有提供完全的工程解决方案。

最后，值得注意的是防篡改不对合法攻击进行保护。如果你依靠它来将算法保留为专有算法，你的竞争对手就会进行一些专利侵权行动（无论多么的琐碎），其目的只是迫使你公开你的设计。这确实发生过！

14.9　小结

防篡改设备以及系统有很长的历史，并且早于电子计算的发展。计算机在许多方式下能被保护为不受物理破坏，诸如将它们锁在有警卫防卫的房间里，也有一些更廉价和便于携带的选择。

这一章对它们中的一些进行了讨论，从价值上千美元的设备，它们由美国政府注册，以抵制所有现在已知的进攻，到拥有价值上千美元设备的攻击者在几周内就能破解的智能卡。我讨论了许多应用与失误。通常，失败不是硬件屏障或警报的错误，而是未能以正确的方式使用技术的结果。

研究问题

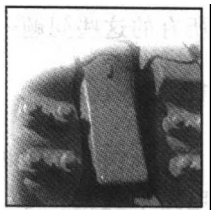
基本上在防篡改处理器设计方面有三种研究。第一个关注的是制造出更快、更好、更便宜的处理器：如何使高端设备提供的保护可以被引入到具有中等价格和规模的产品中，以及如何使中等程度的保护被引入到智能卡中？第二个关注是攻击技术的现状。最新的芯片测试技术如何能用来进行更快、更好、更便宜的进攻？

第三项关注其保护的逻辑方面，即使假定你能在处理器的周围放上特殊的屏障——例如，设想在 Mars 周围轨道上的一个处理器——你如何设计事务集（以及周边的应用），使它能做有用的工作，并且具有高水平的保证，使一些简单的攻击不会被发现？

参考资料

对于加密的早期历史，包括像加权码书籍和水溶油墨等的来源当然是 Kahn 的书 [428]。提到的 IBM 和 Dallas 产品可在网上获得大量的文档 [397]；FIPS 文档也能在线找到 [576]。对于芯片卡技术的介绍，参见 [632]；对于芯片卡的篡改进攻的骇人听闻的细节，参见[43, 44,470]，非进攻性的针对安全处理器的进攻，如电源分析，将在下一章讨论。

第15章 发射安全



在这无边的黑暗中，双方的阵地，营帐接着营帐，传播着轻轻的声响；那站岗的哨兵，几乎各自听得见对方在私下用耳语把口令传授。火光遥对着火光，在那惨淡的照明下，彼此都望见了对方昏沉沉的脸儿。

——莎士比亚《亨利五世》第四幕

15.1 引言

发射安全（emission security，Emsec）指的是防止人们利用泄密发射（Comprormising emanation）来攻击一个系统，其中泄密发射即传导或辐射的电磁信号。发射安全有许多方面的内容。军事组织非常关注风暴（Tempest）防御，它能防止计算机和其他电子设备发射出的杂散射频（RF）被敌方收集到，并用于对正在处理的数据进行重建。智能卡行业已经为电源分析而大伤脑筋。在电源分析中，智能卡执行的计算（例如数字签名）可以通过测量CPU取得的电流进而获取用于重建密钥的方式观测到。这些威胁紧密关联，但同时也有许多的防范措施。

人们通常低估了发射安全的重要性。然而，在20世纪的后25年里，世界上的军事组织用在发射安全上的时间与用在加密术上的时间几乎不相上下。在商业领域，智能卡的应用在20世纪的最后几年里受到了很大的阻碍。因为人们意识到，所有在市场上销售的智能卡对于简单的攻击都是抵挡不住的。通过对终端进行特殊的改装，使其能分析在少量交易中产生的电流，并诱骗消费者使用这些终端，攻击者就能达到其目的。这些攻击无需穿透卡片（至少，当需要用来设计此攻击的研究已经执行），因而可能不会留下任何踪迹。一旦设置成功，这种攻击就比探测攻击耗费的钱少得多，而且使针对不起疑心的智能卡持有人群进行大规模的智能卡克隆攻击也成为可能。

针对其他商业系统的电磁窃听攻击前面已经讨论过，包括对ATM自动柜员机的攻击。人们对于引起混乱的电磁攻击也有许多猜测，例如，凭借着这种攻击，恐怖组织能利用高能微波源来摧毁目标组织的计算机，而不伤害任何人（本书将在有关电子战的章节中详细讨论这些事情）。

主动和被动的发射安全测量，在防止因电磁兼容（EMC）和无线频率干扰（RFI）问题而出现的系统混乱方面是紧密相关的。如果你经常坐飞机，毫无疑问，你会听到机长说："所有的电子设备现在必须关闭，直到我在起飞后10分钟关闭安全带信号，才允许打开。"这个问题正在逐步恶化，因为现在几乎所有的设备都逐渐变成了电子式的，而且时钟频率在迅速升高。那么当越来越多的设备设计为"总是打开"时，当机长说这些的话时，你该怎么办呢——因为关闭开关只能关闭绿色的报告事故灯？

当每天有越来越多的设备与无线网相连接时，以及处理器速度进入千兆赫兹的范围时，

所有的这些问题——RFI/EMC、发射安全和各种不同的电子战威胁——注定变得越来越糟糕。

15.2 历史

电话线路之间的“串音”是19世纪率先使用电话的人们所周知的一个问题，它将双线电路堆叠在一起，放置于电线杆的横杆上。处理这个问题的一种途径是利用“交错法”，借此，线路以一定的间隔相互交叉，以制成双绞线电路。这个问题似乎首先是由军方注意到的，发生在英国军队于1884～1885年远征Nile和Suakin时［569］。

泄密发射出现在战争中可以追溯到1914年。战争的双方铺设战地电话线，用以联系军队和他们的指挥总部，并且长达数英里，与仅有数百英尺远的敌方战壕相平行。这些线是单芯绝缘电缆，并且是地回线，以支持线缆的重量和体积。人们很快发现地面泄露导致了许多的串音，也包括敌方的消息。军队迅速建立侦听岗位，并引入了一些保护措施，其中包括双绞线的应用。到1915年，电子管放大器已经使地面泄露的监听范围大大扩展，电话可以到100码，莫尔斯电码可以到300码。英军发现在无人区的、废弃的电报线的混乱状况提供了非常好的监听渠道，并且给德军提供了大量的通讯信息，以致于将其清除成了一件需要付出生命代价的任务。到1916年，地回线路在前线3000码的范围内被禁用了。当美国加入战争后，这项技术被传播开来。更多的信息能够在［542，569］中找到。

二战期间，无线电工程见证了雷达的出现。雷达是被动的方向性侦察技术，以及低拦截概率的技术。本书将在下一章中对此进行探讨。到20世纪60年代，家用电视机的本地振荡器信号的杂散射频泄露能被“电视探测车”中的方向探测设备进行目标锁定。在英国，电视机的拥有者每年需要支付一定的费用以支持公共广播服务。它的使用已扩展到卫星和有线电视的运营商，他们使用探测车来寻找非法解码器。一些计算机安全领域的人也觉察到，信息能从交叉耦合和杂散射频中泄露出去（具体的例子见［259，791］）。

情报通信领域开始利用射频效应。1960年，在有关加入欧洲经济共同体的谈判过程中，英国首相命令对法国大使进行监视。其国内的情报机构——MI5的科学家注意到，来自于大使馆的加密通讯携带有微弱的二级信号，于是组建设备对这些信号进行恢复。结果证明这些信号是纯文本的，它以某种方式从密码机中泄露出来［814］，这比一个人所能设想到的要更为普遍；关于密码机用无线电频率、以纯文本方式进行广播的例子不只一个，尽管也经常有理由怀疑设备销售商所在的国家已经觉察到了这一点。

在20世纪70年代，发射安全成为高机密性的话题，并从公开的文献中消失。在1985年，这个话题又重新回到公众的注意当中，当时，Wim van Eck，一名荷兰研究员，发表了一篇描写他如何成功地对VDU上的图像从远处进行恢复。泄露发射攻击不仅是可行的，而且能以家制的简单设备进行简单实现的启示，在计算机安全行业引起了震动。

在20世纪90年代的后5年，有关发射安全和相关主题的研究得到发表。1996年，Markus Kuhn和我在［43］中观测到，许多智能卡能通过在它们的电源或时钟线中插入瞬间信号或假信号脉冲，而被破解（发现该项攻击的人不是我们，而是付费电视的黑客）。Paul Kocher也展示过，许多密码系统的常见部署，能通过对花费时间作精确的测量而进行破解［466］。在1998年，Kuhn和我发表了一篇论文，它表明许多来自于PC的泄密发射能通过合适的软件措施，而做得更好，或更坏［478］。在1998和1999年，Kocher表明，加密密钥（用于智能卡）能通过对卡所引起的电流吸引进行精密测量的合适进程而恢复——我将在

15.4.1.2小节详细讨论这个问题［467］。在2000年，David Samyde和Jean-Jacques Quisquater演示过，同样的攻击能通过将小的电磁场感应器放到卡的表面而得以执行［668］。

15.3 技术监视和对策

在对诸如Tempest监视接收器等高技术产品进行介绍之前，我们需要停一下，对窃听器进行一下考虑。利用电磁频谱进行的最简单和应用最广泛的攻击，不是利用某些非故意的设备的某些设计特征，而是利用攻击者引入的自定义设计的设备。

在传输和存储数据时，无论数据通过加密还是访问控制进行保护，在数据的传输和储存过程中，大多数的机密信息或者来源于对话，或者来源于PC上的击键。如果这些信息在这个时候能被对手捕捉到，那么后续的任何保护措施都不可能起到很大作用。

在市场上，可以买到很多不同类型的窃听器：

- 在低端，数十美元就可以买到简单的无线麦克风，你能在拜访窃听目标时将它粘在桌子底下。电池寿命是这些设备的主要限制。它们一般只有几百码的作用范围，且只有几天或几周的寿命。
- 下一类级别较高的设备利用干线的电源进行工作。这些干线指的是电话线缆或其他的外部供应电源。这意味着它们一旦安装好，就能无限期地持续下去。有一些是简单的麦克风，只要窃听者能在一个房间里单独待上几分钟，他就能在电线上安装这种麦克风。另一些则是通过在墙上的天花板中钻通道，来从邻近的建筑物或公寓中插入窃听装置。还有一些窃听装置最近由英国警察用在黑社会监控事件中，它看起来像电子适配器，但实际包含麦克风，无线电传送器，电视录像仪。还有一些窃听装置对数据进行监视——例如，有一种特洛伊木马计算机键盘，它在电缆连接器中包含有用于窃听的硬件。
- 许多现代的窃听器利用现货供应的无线移动技术。它们可以看做是经过稍微改造的蜂窝电话手持设备，它能在呼叫的时候悄悄窃听。
- 在Fort Meade的NSA博物馆中展示着一种奇特设备。在1946年由一群学生作为礼物献给美国驻莫斯科大使。它是一件木制的美国大海豚复制品。大使将其挂在住所办公室的墙上。在1952年，人们发现它里面包含一个共鸣的空洞，当从外面的建筑物用微波对它进行照射时，它就相当于一个麦克风，并且将发生在办公室的谈话重发射出去。直到冷战的末期，美国驻莫斯科大使馆经常被微波照射，所以估计技术的各种变种一直在使用。
- 激光麦克风。其工作原理是往目标对象正在谈话的房间里的反射或部分反射的表面（如窗格玻璃）上发射激光束。声波对反射光线进行调制，它能在一定的距离外被收集起来并进行解码。
- 如今正由政府使用的高端设备。价值超过1万美元，采用低拦截率的无线电技术，诸如频率跳变和定向脉冲发送，它们也能从远处进行开关。这些特性使它们很难被发现。

许多防范措施，能为防范攻击提供一定程度的保护。

- 非线性连接探测器是一个能发现附近隐藏有电子设备的仪器。它的工作原理是因为晶体管、电子管以及其他电子设备中的非线性连接器有纠正入射的无线电频率信号的效果。这种设备发送微弱的无线电信号，并监听这个信号的和声。它可以在几英

尺的范围内探测到未屏蔽的电子设备。然而，如果窃听器就安装在电子设备附近，那么非线性连接探测器也起不了多大的作用。也有非常昂贵的、设计为根本不重发射的窃听器。新闻记者 Duncan Campbell 在 20 世纪 70 年代早期发明了一个非常有趣的探测器变体，用于检测电话窃听——当时安全局使用放大器，顺着电话线重发射谐波。在对他的房屋进行突然搜查之后，这个设备的设计计划被发现了，然后它被宣称“发明”于政府的实验室里，并被归功于一位官方的科学家。

- 现在市场上有很多监视接收器销售。比较好的设备能每隔数十秒钟就扫描一遍从 10 KHz 到 3 GHz 的无线频谱，并寻找那些不能解释为广播、警察、空中交通管制等信号（在 3 GHz 以上，信号被建筑材料削弱了许多，并且设备的天线非常具有方向性，所以普通的频谱搜索不再如非线性连接探测器和物理搜索那样有效。与普遍的想法相反，一些低拦截率的技术并不能提供完全的保护。直接时序扩展频谱能从它的能量谱中发现，跳频能在连续扫描的不同频率处被观测出来。脉冲定向传输确实做得很好。但是监听接收器的有效性正越来越受到窃听器可用性的限制，这些窃听器使用与法定的移动电话和无线电话相同的频率和协议。安全意识较强的组织会一直禁止手机的使用，但在军队外部不会持续太久。例如，直到 1997 英国议会还在禁止手机，但是当政府换届之后，规则就被废除了。
- 挡住视线，诸如在你的实验室周围植树，能有效地防止激光麦克风，但是通常这是不实际的。对于特别敏感的会议，拥有一个屏蔽的内部房间是不太贵的，并且针对这个目的，已有销售商在出售预制的房间，它提供声音和电磁屏蔽。
- 军事组织的一些设施放在完全屏蔽的建筑物中或地下，因此，即使在室内装了窃听器，也不能在外面收到它们的信号 [55]。这非常昂贵，并且在许多场合下是不现实的。其次的选择是确保在建筑物建成后，没有安装诸如麦克风等设备，并经常进行频谱扫描，禁止不可信的访问者（以及诸如保洁工人等承包人）接触最敏感的区域。但这比看起来要困难得多。一个位于莫斯科的新美国大使馆在发现建筑物中安装有大量的麦克风后，就被废弃了。英国反情报局在发现其建筑承包商的一个雇员过去同爱尔兰共和军有过交往之后，不得不将其新总部大楼的大部分拆掉并进行重建。总共花费约 5 千万美元。

此处，传统的压力存在于技术性的防御之间，它们确实非常有效但也非常昂贵，而过程控制，虽然便宜却非常令人生厌。

所有这些说明，技术的发展正在稳步地使窃听变得越来越容易，但对抵御方而言却是越来越困难。当越来越多的设备能够获取情报和短程无线电或红外通信——即“考虑的事情”变成了“闲聊的事情”——对于与已经放在合适位置的设备有关的进攻有较大的余地，而不是与为此目的而必须设置的员工有关的进攻。例如：

- 使用电话带来的风险比许多人能够相信的要更高。越来越多的人因为方便而使用无线电话，并忘记了他们容易被偷听。电话能被改装成可在远程控制下变成可窃听的；一些数字（ISDN）电话内部有一个设备可以做到这一点（据说一些压制性的国家将这个特征作为进口注册的一个条件）。同样，一些 PBX（专用分组交换机）被制造成能进行重编程，从而支持这种监听。
- 典型的笔记本电脑装有麦克风，它在软件控制下可以进行开关操作。并且笔记本电

脑正越来越可能与无线局域网互连。一个攻击者可能利用一个病毒来感染这种电脑，用来监听房间里的谈话，并将它们压缩、加密、并用 E-mail 发回到病毒的发明者。

- NSA 禁止其建筑物内存有 Furby 玩具，因为 Furby 会记住（并可随机重复）当场说的事情。

但是，还有许多其他的方法，使得窃听者可以利用现存的电子设备。

15.4 被动攻击

我们将首先考虑被动攻击，即对手利用任何呈现给他的电磁信号进行的进攻，而不用花费任何精力自行创造。一般来说，有两大类别的被动攻击。能在某种电路（如电源线或电话线）中导通的信号，或者能以无线电频率能量发射的信号。这两类威胁分别被军方指定为 Hijack 和 Tempest。它们不是相互独立的；射频威胁经常有导通的部件。例如，由计算机发射的无线信号能被主电源电路收集并导入相邻的建筑物内。但它在大多数时候仍然是比较合理的分类。

15.4.1 通过电源和信号电缆的泄露

自从 19 世纪以来，工程师已经察觉高频信号到处泄露，并且需要细心地来防止它们引发问题，正如所记载的那样，自从 1914 年，这种泄露已被用于军方目的。信息的导通泄露能通过仔细的设计而大大抑制，通过将电源供应和信号线进行适当的过滤和抑制。这组成了可比较的军用和民用电子设备成本差异的很大一部分。

15.4.1.1 红/黑分离

红色设备（携带例如纯文本的秘密数据）必须用过滤器进行隔离，并与黑色设备相屏蔽（黑色设备能直接向外界传送信息）。具有黑色和红色连接的设备，例如密码机，要取得正确的内容会特别的困难。发射安全标准，例如风暴攻击型保护设备的测试要求的 NACSIM 5100A，以及 NATO 相对应的 AMSG 720B，是机密的，这使得解密更难了 [660]。

所以正确屏蔽的设备倾向于小批量有效，并且为特定防御市场而制造。这使其极为昂贵，并且成本也降不下来。空军基地的操作室有数以千计的电缆从那里发出；若对它们全部进行过滤，并加强足够严格的外围管理以保持红/黑分离，会需要上百万的成本。

15.4.1.2 电源分析

通常，人们不会注意到过滤信号的需要，除非发现有人利用了这个漏洞。最近，一个非常重要的例子来自于智能卡的电源攻击。由于智能卡通常是非常薄的载体上的单硅芯片，所以几乎没有利用阻塞、容器等过滤供应电源的余地。电源也可能在敌人的控制之下。如果你使用银行智能卡在黑手党拥有的商店里购物，那么终端就可能在你的卡中附有额外的电子设备，以对你进行欺骗。

20 世纪 90 年代早期，付费电视黑客和一些政府机构知道，通过测量智能卡从其电源中得到的电流，许多信息能从在智能卡中执行的计算过程中收集到。这种攻击，称为电源分析或干线杂乱信号分析，可能只涉及到插入 10Ω 电阻到地线中，并连接数字存储示波器，以观察由设备引起的电流波动。图 15-1 中有一个关于电源追踪结果的例子。

不同的指令有非常不同的电源消耗分布图，如图 15-1 所示，电源消耗也取决于正在处理的数据。在许多场合下，主要的数据依赖贡献来源于总线驱动晶体管，它非常巨大（见图

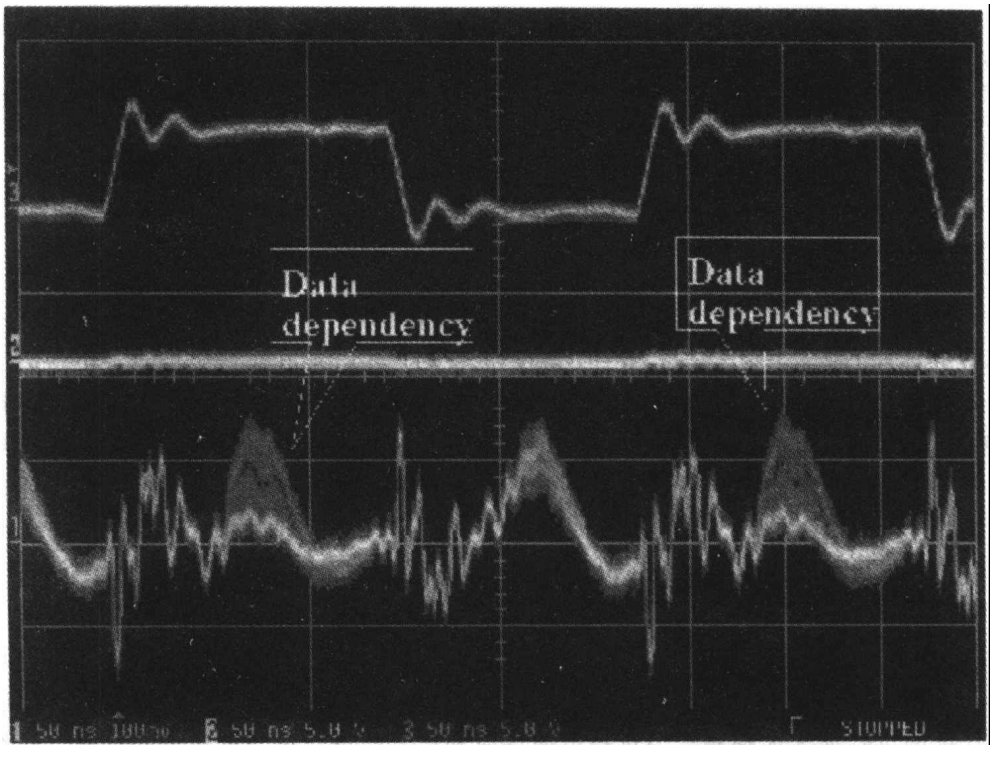


图 15-1　对西门子 SLE44 智能卡的叠加电源消耗进行追踪，表明了数据依赖性（由 Andy Santoso 友情提供）。屏幕上半部分显示的是时钟信号，下半部分显示电源消耗

14-5 的上部）。依靠这些设计，当总线每位的状态改变时，电流能在几百毫微秒的时间里，变化几百个毫安 [547]。因而，总线中每一个数据字节和前一个字节差异的 Hamming 权重（变化数）可以被攻击者得到。在某些设备中，每个数据字节的 Hamming 权重也是可获得的 [549]。EEPROM 读/写甚至能提供更加丰富的信号。

这种泄露的效果使一个懂得密码如何实施的攻击者（比如对智能卡中的软件进行探测并对软件进行反编译之后）能得到有关卡秘密的重要信息，并且在许多场合下，能推算出使用的密钥值。这种攻击是特别重要的，因为它是非入侵性的进攻，并且能利用经过适当改造的终端设备对不起疑心的顾客所携带的智能卡进行攻击。这意味着一旦攻击者能够不怕麻烦地拆开一张卡，弄明白其内容，并设计进攻方式，那么只用少量的成本就可以破解大量的卡。

对智能卡进行电源分析所引起的威胁引起了这个行业的极大关注，在 1998 年，Paul Kocher 对特定信号处理技术进行了发展，从而抽取分组密码，如 DES（数据加密标准）中的密钥位，对电源曲线进行收集，而不用知道智能卡软件中的实施细节。这项技术（*差异电源分析*）工作原理如下 [467]：

攻击者首先收集许多曲线（一般为几百条），通过目标卡执行已经知道的交易——这些交易的加密算法、明文或密文已为人所知。然后猜测密码的内部状态。在 DES 的场合下，密码的每一轮有 8 个查找表，其中 6 位的电流输入与 6 位的密钥材料进行异或，然后用于查找来自于 S 盒的 4 位输出。如果攻击者能接触到密文，她将在最后一轮猜测出 6 个输入到 S 盒中的位。这样，就可把电源曲线分成基于这种猜测的两种集合并予以同步。平均曲线就这样被计算和比较。两平均曲线间的差异被称为*差异跟踪*。

对于每一个进入到目标S盒中的64种可能的6位输入，这个过程都要重复一次。它一般能发现正确的输入值——它将电源曲线分离成两类，每一类都具有不同的S盒输出值——将导致具有显著峰值的不同追踪。然而，对输入值的错误猜测，却通常会导致随机排列的曲线，随之会产生看起来像随机噪声的差异跟踪。利用这种方法，能够找到输入到所讨论的S盒中的6位密钥，随后可以找出用于密码最后一轮的其他字节。在DES的情况下，这种方法能找出56个密钥位中的48个，剩下的采用穷尽搜索就能够轻松地找到。如果密码中有更多的密钥位，那么攻击者可以一次在一轮中解开它。

这项技术造成的结果是：即使能够将智能卡构造为可以抵制探测攻击，它仍可能是易受攻击的，除非智能卡内部建有特定的电源分析抵御系统（事实上，所有1998年在市场上销售的智能卡都被声明为是易受攻击的[476]）。因此，即使不能接触到探测设备的攻击者也能轻易、迅速地发动一个攻击）。

这项技术被广泛推广，并推动智能卡的配置在人们对防御做工作的同时得到提高。在某些情况下，协议级别的防御是可能的；也能设计出这样的协议，它能用每隔几次加密对密钥进行更新，从而防止攻击者得到足够多的数据（一些终端的卖点就是以这种方式进行设计的）。但是大多数现存的协议太确定了，因而不能从根本上进行改变。另一种想法是在加密的方法中引入随机性。例如，在DES的每一轮，一个人可能以随机的顺序查找8个S盒。然而所有能获得的不是差异追踪中的一个大的尖峰信号，而是8个具有八分之一振幅的尖峰信号；因此攻击者必须多收集一些电源曲线。

现在装配的针对电源分析的防御措施是基于硬件的。普通的智能卡有一个硬件设备，它在约每64条左右的机器指令中插入一条假的操作；另一种措施是装备内置时钟，它与外界的时钟只是松耦合的，并且每64次循环改变一次频率。这些措施之中没有一个是十分安全的，因为攻击者可能会使用信号处理技术，来重新排列电源曲线以求得平均值。下一代智能卡可能使用更为健壮的防御措施，例如陶制封装的容器，它能使供应电压正确地去耦，或利用硅设计技术，如双轨编码，它能使电流吸引独立于被处理的数据。然而另一种方法是利用自定义时间逻辑，它不用任何的时钟。当本书编写的时候，这种方法正是一个热点的研究领域。

15.4.2 通过射频信号的泄露

1972年当我第一次在Glasgow学校的计算机中心学习编程时，我们有一台早期的IBM机器，时频为1.5 MHz。在这个机器所在的房间里，当将收音机调到这个频率时，它就会发出大声的哨鸣，并随着处理中的数据而变化。类似的现象被许多人记录下来，其中一些人使用这种噪声作为编译的辅助工具。我所在学校的同事有一个更好的想法：他写了一套具有不同长度的子例程，通过按序调用这些子例程，就可以使计算机演奏一首曲子。我们当时没有想到其中所隐含的安全问题。

现在转向比较现代化的设备，所有的VDU（视频显示装置）会发射出微弱的电视信号——VHF（甚高频）或UHF（超高频）无线电信号，用正在显示的图像的失真版进行调制——除非这些装置经过细致的设计可以防止这种情况的出现。视频信号在设备的许多地方可以获得，特别是在调制过的波束电流中。这个信号包含许多点速率的谐波，其中一些发射信号相对较好，因为线路和其他的部件正在这些波长处共振。利用合适的宽波段接收机，就能使这些发射被收集起来并重新组织成视频信号。合适设备的设计在[259，478]中讨论。与

一般的看法相反，LCD 显示器对于窃听者一般也是很容易攻击的。

其他的研究人员迅速地建立起远程窃听的可能性，窃听的对象从传真机到屏蔽的 RS-232 线，一直到以太网［719，230］。有一些销售“干扰发射机”的公司为此创立。但其正确的实施很困难［60］，因为它们能够干扰电视或其他服务。军用防风暴型攻击设备对于商业部门仍是不可获得的。无论如何，它通常要延迟一代，并且比现货供应的 PC 贵五倍。银行业采用的观点是，“好，我们不对我们的竞争对手做这些攻击，所以他们也不太可能对我们做这些。并且我们不知道从哪儿能得到有效的对策，所以把它放在‘太困难的’文件里”。这个观点在 20 世纪 90 年代末期有点动摇，当 Hans-Georg Wolf 演示了一项风暴型攻击，它能从 8m 外的提款机那里恢复卡和 PIN 数据［239］。然而，对风暴型攻击的防范在商业和非防御部门的行业还是相当少的。[⊖]

同时，随着冷战的结束，军事预算被大笔删减，并且通常只能使用现货供应的设备，而没有别的选择；也没有钱来发展只为政府使用的系统。北约国家的政府机构已经转向 Emsec 保护的零模式。因此，最敏感的设备可保留在离设施最远的房间里，并且为最敏感的系统（如国家情报机关）或者在威胁最高的地方（如海外大使馆）保留屏蔽。尽管如此，北约政府机构用于防护风暴型攻击的经费每年超过十亿美元。

低成本保护技术，称为 Soft Tempest（软风暴），已经出现，并且配置在某些商业产品中（例如 email 加密包 PGP）［478］。软风暴利用软件技术过滤、屏蔽携带有来自于计算机系统的电磁发射的信息，或将这些信息以不可理解的形式向外发布。

Markus Kuhn 和我发现，大多数携带来自 VDU（视频显示装置）的射频能量的信息集中在频谱的顶部，所以将这些成分过滤出来是逻辑上的第一步。我们通过采用适当的低通滤波器对标准字体的傅立叶变换进行卷积，从而去掉其上部的 30%（见图 15-2 和图 15-3）。

结果证明，对于用户看到的屏幕上的内容来说，它有几乎不可察觉的效果。图 15-4 和图 15-5 显示的是图 15-2 和图 15-3 的两个视频信号显示在屏幕上的照片。

发射射频的差异是巨大的，正如图 15-6 和图 15-7 中的照片所示。这表明了潜在的泄密发射，正如风暴型监视接收器所看到的。

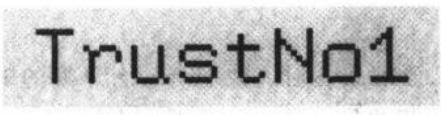


图 15-2　正常文字

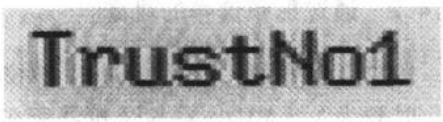


图 15-3　同样的文字，被低通过滤器过滤

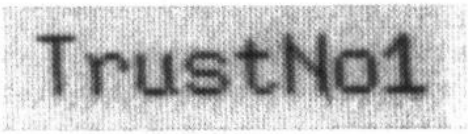


图 15-4　正常文字的屏幕快照

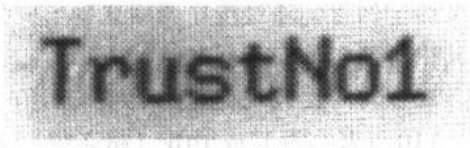


图 15-5　过滤文字的屏幕快照

⊖ 当我从 Wiley 取回原稿进行校对时，第一次听说了一个关于商业风暴型攻击的可信报道。表面上，一个金融机构被一个对手所雇佣的私人侦探所侦查。但是大图像攻击防御仍保留为军用。

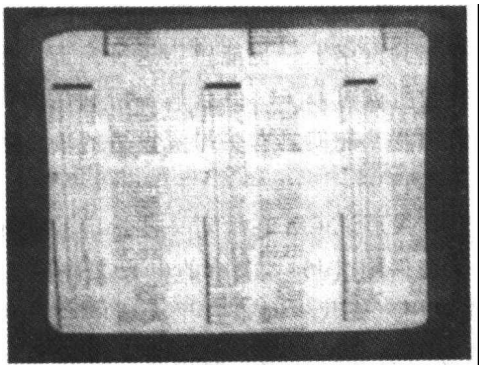


图 15-6　一页正常文字

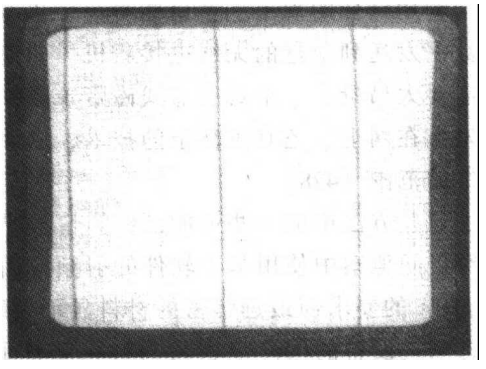


图 15-7　一页过滤的文字

软风暴技术能够提供给 VDU（视频显示装置）的保护级别仅在 10 ~ 20 dB，但是这能转化为区域的差异——它对于一个政府规模的组织来说，可以节省大量的成本 [45]。

也有其他一些用软件技巧就能完全阻塞的攻击。例如，当微控制器循环扫描键盘时遇到某个键被按下时计算机键盘能被窃听到。正在按下的键被调制到来自于键盘的射频发射。通过对键盘扫描的次序进行加密，这种攻击完全能被拦截。

15.5　主动攻击

但仅对键盘扫描样式进行加密并不足以保护它，因为攻击者可以利用主动和被动的技术。针对键盘，这项技术用共振频率的无线电波来对电缆进行照射。由于非线性连接效应，所按下键的码被调制成返回信号，这些信号被电缆重新发射。并能在 50 到 100 码的距离外收集到。为了防止这种现象的发生，必须对从键盘到 PC 的信号进行加密 [478]。

15.5.1　风暴型病毒

对于不同的系统有一些可能的其他主动攻击。1972 年用我们学校的计算机观察到的现象——一个适当的程序会导致一台计算机在收音机中演奏一首曲子，实际上是将其转变成低等级的无线电发射机——这很容易在现代的 PC 上重新实施。图 15-8 和图 15-9 显示出当视频信号是一个 2 MHz 的射频载体，并分别以 300 Hz 和 1200 Hz 的纯音调进行调制后，标准 PC 的屏幕看起来的模样。

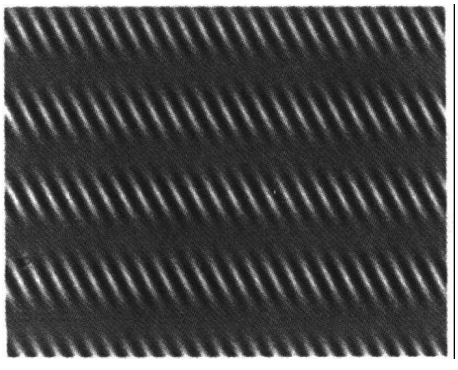


图 15-8　一个 300 Hz 的无线广播信号

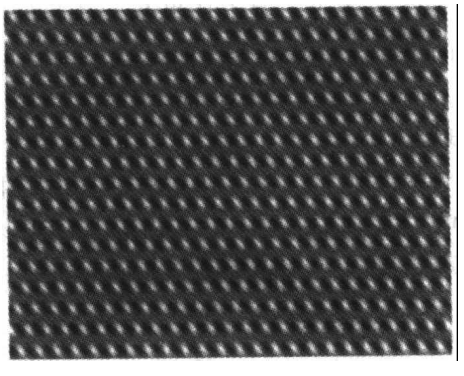


图 15-9　一个 1200 Hz 的无线广播信号

利用这样的现象，就可能写一个 Tempest 病毒，它将传染目标计算机，并将窃取到的机密数据发送到附近的无线电接收机。即使机器没有连到网络上，也能感染这种病毒。接收器不需要太昂贵；一个具有盒式磁带录音机的短波收音机就可以了，并且病毒所利用的代码已经发布在网上。还有更复杂的技术，例如扩展频谱调制，能使具有更昂贵设备的攻击者获得更好的范围 [478]。

这些方法中的一些可能已经为情报界所知，曾有报道说 CIA 在针对一些特定的欧洲国家的经济间谍案中使用基于软件的射频应用（例如，在伴随 [464] 发行的电视文件中）。NSA 应 FOIA 的要求而最近解密的材料显示代码词茶壶被用来指代“对来自于无线电通讯和自动信息系统设备的国际泄密发射（也就是那些因敌意而引起的行动）的调查、研究和控制。”[542, 420] 一个更进一步的例子是攻击那些已经被屏蔽和防发射泄漏认证达到一定频率（比如，1 GHz）的设备，通过利用较高频率（如 10 GHz）的微波穿过通风狭槽对这些设备进行照射，在这个频率这些狭槽变成透明的 [478]。

利用恶意代码攻击的可能性，是为什么风暴测试不仅会涉及被动监听来自于测试设备的发射，而且还要往设备中注入信号，例如长线性反馈偏移编码注册序列的原因。这产生了扩展频谱信号，它在设备的外面可能是可探测的，并因此模仿最坏案例的攻击，在这个攻击中，对手应用软件开发来接管这个设备 [108]。

15.5.2 Nonstop

另一种主动方法，被美国军方称为 Nonstop（不间断型）[55]，是利用偶尔被附近的无线电发射机和其他射频源感应到的射频发射。如果一个正在处理敏感数据的设备靠近移动电话，电话的发射机可能会在其内部感应电流，这些电流通过非线形连接效应而随敏感数据进行调制，并被重新传播。

因为这个原因，通常人们禁止在机密设备的 5 米内使用移动电话。不间断攻击也是轮船和飞机所关注的主要发射安全（Emsec）。此处，一个可以接近这些设备并能进行被动风暴型攻击的攻击者可能会做出比窃听更严重的破坏；但是因为船舶和飞机常常携带有很强的无线电和雷达，所以必须小心，以免信号意外地被调制，并泄露给敌人一些有用的事情。

15.5.3 假信号脉冲

主动的发射安全威胁在智能卡领域也是很重要的，在智能卡中最著名的是假信号脉冲攻击 [43]。在此，如上所提及，对手在卡的电源或时钟供应中插入瞬变信号，并希望引起有用的错误。

例如用于早期银行应用的智能卡有这样一个特征，即不可接受的高时钟频率，会在数个循环后引发智能卡复位，因此瞬变信号很少会导致错误警报。所以，可能用两个非常窄的脉冲来替代单个时钟脉冲，而不会引起报警，从而复位卡。这肯定会导致处理器执行一个 NOP，而不管它被期望为执行什么指令。通过在合适的时间引入假信号脉冲，攻击者就能越过跳转指令，并因而绕过访问控制。

15.5.4 差异故障分析

尽管攻击者并不是非常了解卡的软件，假信号脉冲攻击也仍能成为一种威胁。人们早已

注意到，如果引入随机故障，许多公钥加密算法就会被破解 [126]。例如，当做出 RSA 签名，秘密计算 $S = h\ (m)^d$ 标准地为先执行模 p，然后模 q；再将结果进行联合。然而，如果卡返回有缺陷的签名 S_p，它的模 p 正确，但模 q 不正确，然后，我们将有

$$p = \gcd\ (pq,\ S_p^e - h\ (m))$$

它会立即破解这个系统。这些攻击能轻易地实施，如果卡没有采用针对假信号脉冲的保护措施的话。它们也能扩展到许多对称算法和协议 [103]。

15.5.5 联合攻击

其他的攻击方式将主动和被动方法联合使用，我在第一部分提及过一种技巧，能用来发现被盗智能卡中的 PIN。早期的智能卡系统会要顾客提供 PIN，如果不正确，会减少一次重复的次数。这包括往 EEPROM 中写入一个字节，因此当 EEPROM 电压倍增器电路中的电容器充电时卡所消耗的电流，会明显地上升。注意到这一点，攻击者能简单地复位卡并试验一个候选的 PIN。

15.5.6 商业利用

不是所有发射安全攻击都在秘密的军事侦察或针对防篡改设备的实验室攻击的环境下执行过。本书曾经提及过在英国使用的 TV 探测车，它能发现并抓获不履行 TV 注册的用户，以及偷看付费电视的用户。它也有市场方面的应用。当顾客的汽车驶入会场的停车场地时，美国会场运营商 SFX 娱乐公司通过收集收音机本地震荡器上的杂散射频监视顾客车子里的收音机里正放些什么内容。尽管合法，但也使隐私倡导者感到恐慌。同样的设备被卖给汽车销售商、商业街运营商以及广播电台。

15.5.7 防御

能够用于使智能卡抵御主动发射安全威胁的技术与用于被动场合下的技术类似，尽管不是完全一样。

定时随机性——信号抖动——仍然非常有用，因为没有经验的对手可能不会知道应该在什么时候精确地插入假信号脉冲。然而，一个聪明的对手也可能实时分析来自处理器的电源曲线，并与代码相比较以寻找出关键的目标指令。另外，错误攻击很难用信号抖动进行阻止，因为代码中错误的准确定位通常不是非常关键的。

在有些情况下，防御性编程技术就足够了。例如，在 15.5.5 节描述的 PIN 搜索，在比较现代的仪器中是可以防止的，通过递减计数器、要求提供 PIN、如果 PIN 正确就再次增加计数器。如果你只检查结果，那么针对公钥协议的差异攻击就会变得困难得多。

其他的系统利用特殊的防护硬件，例如一个电路，它将卡的复位与对太高或太低的时钟频率进行探测的电路进行集成。通常的复位包含将时钟频率减半，所以如果攻击者找到了可以使监视功能无效的方法后，他也可能会发现在加电时对卡进行复位是不可能的 [470]。

针对假信号脉冲攻击的现行防御方法不是十分安全的，并且大范围的设备测试是非常明智的。新技术，例如自定时逻辑的使用，通过提供针对主动和被动威胁的高级别保护，可能会使事情得到改善。同时，如果你不得不写一个智能卡的应用，那么基于假信号脉冲的攻击就值得仔细的考虑。

15.6 发射安全攻击有多严重

技术性的监视及其对策是发射安全攻击最重要的一个方面，无论是在政府还是在行业中；它们也可能会继续保持这个样子。窃听器和其他能被轻易买到的监视设备的有效距离非常大，并且日益增长。人们对其对手、雇员，以及其他的一些人进行侦查的动机仍将继续。如果有什么不同的话，向网络世界的变迁将使电子侦察变得更加重要，相应的对策也将占据过多安全预算。

与未设计监测措施（Tempest、Teapot、Hijack、Nonstop 以及各种类型的电源和假信号脉冲攻击）的设备有关的发射安全的那些方面会演变成为另一些技术，这些技术起初在政府部门发展，但是随后在商业产品的设计中开始变得重要起来。

15.6.1 政府

处于敌对国家的大使馆所面临的发射安全威胁尤为真实。如果一个国家被敌国所迫，必须将自己的大使馆放在办公楼的第二层，它的第一和第三层则由当地的秘密警察占据，那么安全问题将是一个极为困难的问题。对所有的电子设备（除了那些故意用来设计骗局的）进行屏蔽，会是解决方案的一部分。在威胁性较小的环境中，硬件 Tempest 屏蔽的使用就很令人怀疑。

尽管利用一些欺骗手段，Tempest 产业在冷战期间得到了维持，人们关于是否有外国机构曾经真正实施过 Tempest 攻击的怀疑一直在增加，尽管存在大量的有关这方面的传闻。据说，例如，在整个北美只有加拿大的情报人员能够使用这些监视技术做出违背美国利益的事，他们偷听美国外交官讨论对别国销售谷物的美方底线；而东德的 Stasi 被发现拥有西德城镇中 Tempest 棚车相应停车地点的地图。但是我还没有发现任何东西，能够明确其可靠的信号源，并曾驱车在英国城镇周围搜索 Tempest 信号。我可以证实，发动那样的一个攻击，在实践中要比在理论上看起来困难得多。政府现在对 Tempest 危险的重视，远比 10 年前要放松得多。

15.6.2 商业

在私有部门，情况恰好相反。错误攻击，以及随后电源攻击的发现，是智能卡行业的一件大事，并使得在那些还没有利用智能卡的国家，银行将要应用的智能卡配备停滞了近两年。要阻止这些攻击被证明是困难的，并且正确地执行防范措施将涉及到下一代的硬件设计。

将来会是怎么样呢？

发射管理的“非安全性”方面，即 RFI/EMC，正在逐渐变得越来越重要。不断增长的时钟速度，加上各种无线设备和网络的引入，以及数字电子向许多以前是模拟或机械的设备中扩散，正使电磁兼容变得更难，并产生更紧迫的问题。不同的工业团体，持有许多不兼容标准，这些标准中有很多正迅速作废——例如，通过不要求 1 GHz 以上的测试，或者通过设定不再合理的防护距离 [455]。

在安全方面，攻击很可能变得轻松。软件无线电广播的出现——在中间频段对信号进行数字化，并在软件中做出所有的解调和后期处理——直到最近，仍是昂贵的、军方感兴趣的

事情 [482]，但是现在蜂窝无线基站等在许多地方得到应用。下一代可能就是消费者的设备，它被设计为具有 GPS 接收机、GSM 电话、无线 LAN 基站等功能，并支持任何其他当地许可的、基于无线的服务——所要做的事仅是对软件进行一个小的改变。一旦人们知道如何对它进行编程，它们可能就会被很容易地用于风暴型攻击中。

最后，发射安全问题并不是完全与电子战无关。因为社会变得越来越依靠那些对于强无线电频率信号（例如由军事雷达产生的高能微波）攻击变得很脆弱的设备，发动那些攻击的诱惑力也会增加。本书将在下一章中对高能无线电频率攻击进行探讨。

15.7 小结

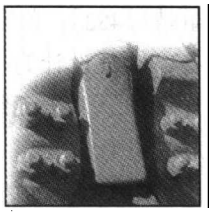
发射安全覆盖了范围很广的威胁，在这些威胁中，系统安全能被泄密发射所破坏，不管采用嵌入的窃听器，还是非国际性无线电频率，或导通的电磁泄露，或以其他方式引起的发射。尽管发射安全开始是一个国家情报部门关心的话题，但是现在对于构建诸如智能卡和提款机等安全产品的公司来说却是一个真实的话题。通过对杂散射频或导通信号进行观测，就能破解这些产品。对这类威胁进行保护不像看起来那样直接。

研究问题

安全行业急切需要一整套为商业应用而定义的发射安全标准。军用标准是机密的，而 RFI/EMC 标准是零散和矛盾的，所以希望出现一种新的、统一的方法是期待已久的。

参考资料

有关发射安全的公开文献资料较少，Eck 写的经典文章 [259] 仍然值得一读；并且直到本书为止，能用一章来论述这个主题的有关计算机安全方面的书仅有 Deborah Russell 和 G.T.Gangemi 编写的一本书 [660]。我们最近在软 Tempest、Teapot 以及相关主题方面做出的工作能在 [478] 中找到。对于电源分析，见 Paul Kocher 的论文 [467]，以及 Tom Messergues、Ezzy Dabish 和 Robert Sloan 的论文 [547]；更多的论文现在正经常出现。最后，Joel McNamara 经营了一个全面的、非官方的 Tempest 网站 [542]。



第 16 章　电子战与信息战

兵者，诡道也……利而诱之，乱而取之。

——孙武《孙子兵法》1.18～20

武力与欺骗无疑是战争中最重要的两大因素。

——Thomas Hobbes

16.1　引言

几十年来，电子战已从计算机安全中独立出来成为一门新的学科，尽管它们有着某些共同的技术（例如密码学）。现在这两门学科已经开始融合进而形成一个新的学科——信息战，在 20 世纪最后的几年中，其在军事中的应用标志着其重要性的建立——即使它的基本概念、理论和学说还未发展起来。

对电子战的了解为什么对于从事安全工作的技术人员那么重要，还存在着其他方面的原因。许多当初从军队发展起来的技术已改作商业应用，有许多指令是相似的。另外，控制电磁波谱的竞争已消耗了如此之多的人才精英与数不尽的美元，我们所能找到的只是在前所未有的广度与深度上的战略战术欺骗。这是一个电子安全领域，使可能引发的敌我双方经历一个相当长时期的攻防协同准备。

拒绝服务攻击曾是人们大大忽视的计算机安全话题，电子战也是我们了解该话题的主要老师，但由于在商业网站上散布的拒绝服务攻击，使其现在已成为中心话题。当我展开讨论时，我会尽可能地指出这些相似点。总的来说，人们常常说计算机安全涉及的是保密性、完整性与可用性；我认为电子战与之相反，或者说是事物的另一面。优先讲下面三个问题：

1）拒绝服务，包括干扰、模仿与有形攻击。

2）欺骗，其目标可能是自动化系统或人。

3）利用，不但包括偷听，还包括在敌人利用其电子系统时获取任何可操作的有价值信息。

16.2　基础

电子战的目标就是要控制电磁波谱，通常认为由以下几个方面构成：

- 电子攻击，像干扰敌人的通信或雷达，用高能微波破坏敌方设备。
- 电子保护，从设计抗干扰系统，到巩固己方设备以防高能微波攻击，再到用抗辐射导弹摧毁敌方干扰发射台。
- 电子支援，指提供必须的情报与威胁识别来允许有效的攻击与保护，不管是有意识还是无意识的电磁能量，都允许指挥员搜索、识别与定位来源。

以上定义摘自 Schleher 的文献 [677]。传统的密码学专题又名通信安全，只是电子保护的一小部分，正如它在更一般的系统中变成信息保护的一小部分一样，电子支援包括信号情报，而信号情报又包含通信情报与电子情报。前者搜集敌方通信，包括军事小组正在联络的通信内容与传输数据；后者关心的是可识别的敌方雷达和其他并没有进行联络的电磁能量。

欺骗是电子攻击的核心，其目标就是通过操纵敌方的理解来降低其情报的精度和目标获取，以达到误导敌军的作用，其有效应用依赖于以下方面是否明确——关于谁（什么）将被欺骗、欺骗的内容以及持续的时间。当欺骗对象是人时，还涉及到对骄傲、贪婪、懒惰以及其他人类恶习的利用。欺骗能够非常好地利用固定的付出代价取得最好的效果，而且也适于商业系统。

当干扰（软杀伤）对敌方的传感器与通信线路并不能构成威胁，反而导致其他的事物被摧毁（硬杀伤）时，有形摧毁是综合应用中重要的组成部分。成功的电子战有赖于并行应用可能的手段。

电子武器系统与其他武器非常相似，系统中有传感器，例如雷达、红外线和声纳；有通信线路，将传感器获得的数据送至指挥员与控制中心；还有输出设备，例如干扰发射器、激光等等。本章首先讨论通信系统问题，因为它们是最具自我调节的系统，然后讲传感器与联合干扰，最后谈电磁脉冲生成器等其他设备。讲完电子战，我们接着讲信息战。

16.3　通信系统

军事通信在 1860 年之前完全靠人力传递，从那以后到 1915 年间的日子里，则靠电报来完成，然后开始了电话传输的时期，直到今天 [569]。如今，典型的指挥与控制结构由各种战术、战略天线台网络构成，它们支持数据、声音、图像、点对点连接以及广播的传输。如果指挥员没有对局势的了解和直接的武力威胁，他就毫无作用可言。但对安全通信的要求已经大大渗透开来，超出了人们当初可能认识到的范围，其威胁也不大相同。

- 一个典型的通信传输是在固定地址之间的通信，例如军事司令部与政府首脑之间。这里存在的主要威胁是密码安全可能被侵入，以致于命令、形势报告等等遭到损害。可能是因为密码分析引起，或者很可能是设备遭受破坏，或者个人进行的颠覆，或者关键材料的失盗。欺骗消息的插入在某些环境中也构成一种威胁。但密码安全通常包括了对通信传输分析（例如通信线路的加密）的保护和对传输信息的机密性与真实性的保护。次要的威胁便是通信线路可能被毁坏，例如光缆或中继站被毁。
- 还有更严格的要求就是与关键人物之间的通信，例如战场特工人员。此时，除了密码安全问题，位置安全也是重要的，特工人员必须逐步缩小被抓住的风险，否则便成了通信监视的对象。如果利用敌方可监听的媒体，像公众电话网或无线电之类来传输消息，那么其大部分的努力将使得己方的干扰分析与无线电测向变得无效。
- 战术通信，例如在司令部与战场上的一个排之间，也有着更加严格的要求（尽管与前面有着细微的差别）。无线电测向也是一个问题，但干扰至少与之一样重要；谨慎的欺骗性消息可能也是一个问题。举个例子：如果有能够捕获敌方空军控制台声音指挥的设备，将其截成碎片然后拼合成欺骗性命令，以期在空战中获得战术优势 [324]。当声音形态技术被发展为商业应用之后，对非保护性通信的戏弄性攻击的风

险将上升。因此，密码安全包括了真实性与保密性和（或）隐藏性。

- 控制与电信学通信，像那些从飞机上发往一颗刚发射的导弹的信号，必须进行反干扰和修改的保护。如果它们是隐蔽的（为了不致使目标飞机的预警），将是令人满意的，但要用必要的能量级来战胜装载有抗干扰系统的飞机是有一定困难的。

通信保护依据不同的环境——内容安全、真实性、反通信分析和无线电测向以及各种形式的反干扰——需要进行一定的综合，它们以某些不明显的方式相互影响。例如在20世纪80年代早期为东欧的某些反对组织设计的无线电波段，通常由美国之音、BBC全球服务等节目占用，而它们通常受到苏联的干扰，除非苏联方面准备关掉他们的干扰发射台，否则他们将很难进行测向。

攻击通常也要求联合多种技术，即使目标不能被分析或定向而只能进行拒绝服务攻击的地方。Owen Lewis进行了简单的总结：根据苏联宣称，对一个军事通信基础设施综合性的成功攻击包括首先摧毁三分之一的物理设施，其次通过像干扰、内部颠覆或欺骗之类的技术把其有效应用的另外三分之一拒绝掉，最后将对手试图用剩下来的三分之一的固定设备传送的所有通信信息屏蔽掉［500］。这种方式甚至还应用到了游击战中：在马来西亚、肯尼亚和塞浦路斯，叛军成功地使电话系统失效，以致于迫使警方创建无线电通信网络［569］。

在20世纪80年代，北大西洋公约组织（NATO）开发了一个类似的系统，称为反指挥、控制与通信军事行动计划（C-C3，发音为C的立方），并在海湾战争中迎来了它的第一个鼎盛时期，用在那场战争中的指挥控制系统，在［643］中有具体的描述（当然，攻击敌方的指挥基地比之历史更为悠久，这就是俗语“擒贼先擒王”的精髓）。

16.3.1 信号侦察技术

在通信可被攻击之前，必须在地图上标明敌方的网络系统。信号侦察中最昂贵最重要的任务便是从无线电信号和电话网络系统和因特网系统中的大量信息中识别并提取出感兴趣的材料。这些技术的应用非常广泛，可也是大量保密的，但某些方面是公开的。

在无线电信号这个例子中，通信情报机构用能够识别大量不同种类信号的信号接收机来维护信号数据库，说明哪些站点或设备用什么频率。在很多案例中，也可以通过信号分析来识别单个的机器。这些线索包括任何非敌意的调频，发射机打开时瞬间的状态，准确的中心频率和最后阶段放大器谐波。这种射频“指纹”识别技术在20世纪90年代中期被降低了密级，用于识别克隆细胞电话，其制造者宣称能达到95%的成功率［341，677］。这是二战中通过传送摩尔斯电码的手而识别无线电话兵技术的直接衍生技术［523］。

无线电测向（RDF）也很关键，过去，这项技术对感兴趣的信号利用两个监视站的定向天线进行三角测量。间谍人员在不得不移动前可能最多只有几分钟的时间将消息发送回去。现代的监视站利用到达的时间差（TDOA）即对比两个站点收到的信号阶段来快速、准确、自动地定位一个可疑的信号。现在，任何超过1秒钟的通话都能成为一个泄漏物。

*流量分析*通过观察来往消息的数量，也可以提供非常有用的信息，不但有即时攻击（在一战中通过急剧增加的无线电消息流量来示意），还有小分队的急速调动以及其他常规事件。然而，在公众网上筛选信息才真正地使流量分析盛行起来，在公众网上其重要性（对国家情报机构和警察行动而言）很难被夸大。

如果你怀疑Alice从事间谍活动（或者毒品交易，或其他什么），你就记录下她打给的每

个人与每个打电话给她的人，这便给你提供了几十个可疑的名单，你再排除像银行与医生之类的电话，因为他们都接听很多电话用于分析（这也就是你的清白者名单），然后对每一个遗留的号码重复这一过程。对这一过程进行好几次循环后，你便掌握了一群成千上万的联系者，将出现不止一次的电话号码筛选出来。如果（说）Bob、Camilla 与 Donald 是 Alice 的熟人，Bob 和 Camilla 与此同时与 Eve 有联系而 Donald 和 Eve 与 Farquhar 相联系，那么所有这些人都被认为是可疑的，现有你画一张人际关系树状图，它提供有关 Alice 关系网的基本的近似，然后通过与其他情报源相对照提炼出结果。

说来容易做来难。一个人可能有几个号码，Bob 可能接到 Alice 从办公室打来的电话，然后从一个电话亭给 Eve 打了电话（实际上，如果你在运营一个 IRA 基层组织，你的信号办事员可能在一个牙医诊所或医疗所或其他地方打零工，那么他将收到许多的电话，他们很可能是应在清白名单上的不相干的人，但那是另外的情况）。而且，你需要找到某种线索将电话号码与人联系起来。即使你有办法访问电话公司的非公开电话数据库，预付费移动电话可能是一个令人头痛的问题，就像克隆电话与出租的专用小交换机那样。我将在有关电信安全的章节中讨论这个问题；现在，我只谈论匿名电话，这已不是什么新鲜事情了。公共电话亭已经存在几个世纪了，但那并不能构成犯罪的通用解释，正如正确使用它们的原则已无法约束绝大部分罪犯那样，在任何情况下都将导致严重的破坏。

信号搜集不受电话公司提供可访问电话内容与通信数据准许的限制，它也需要一个涉及范围很广的专门设备，通过临时的策略安排，对一些模仿国际卫星通信战略的昂贵的固定设备进行距离修正。Nicky Hager 的书［368］中描述了由美国、加拿大、英国、澳大利亚和新西兰操纵的主要固定搜集网络，众所周知的 Echelon，它由大量的搜集站组成，而这些搜集站则监视国际电话、传真、以及使用词典计算机进行的数据通信，这些设备细查那些感兴趣的电话号码、网络地址以及涉及可读机器内容的传输通信；这一切都是由情报分析员输入的搜集字符串来驱动的。在必要的时候，这些固定网络系统由战术搜集设备来补充；Hager 举例描述了澳大利亚和新西兰海军驱逐舰队监听 20 世纪 80 年代斐济（Fiji）军事政变期间国内通信的公文急件。Egmont Koch 和 Jochen Sperber 讨论了美国与德国在德国境内的军事设施［464］；David Fulghum 也描述了机载信号的搜集［324］；卫星也用于信号搜集，同时，也有不被东道国所知晓的隐蔽的搜集设备。

尽管进行了庞大的资本投入，整个操作最困难也最昂贵的部分是信息筛选，而非搜集［490］。然而，与天真的期望相反，密码术带给通信的是更容易遭受攻击，而不是减少攻击（如果密码使用不恰当，也就是人们通常使用的那样）。如果你仅把你认为重要的所有信息加密，你就为你的敌人搜集此信息做上了标志。另一方面，如果每个人都把信息加密，那么隐藏相关信息便会更容易些（因此，信号侦察机构严禁广泛使用密码术，即使每个人都能免费使用该技术）。这就将我们引向了攻击这个谈论主题。

16.3.2　通信攻击

一旦在地图上标出了敌人的网络，你就肯定希望能去攻击它，也就是人们通常所说的“密码破译”，但这是一种十分严重的过于简单化行为。

首先，尽管一些系统能被单纯的密码分析所破坏，但这是十分少见的。大多数成形的攻击都包含了关键材料的窃取，就像在二战中美国国务院的密码本被美国驻罗马大使的仆人偷

走，或像美国对苏联外交通信进行的代号为“Venona”的攻击那样，错误地制造与分发了关键材料［428］。即使基于密码分析的攻击是可能的，它们也经常被错误地分析，就像在二战中英国和美国攻击德国 Enigma 通信的例子那样［429］。这种趋势延续至今。在冷战期间，苏联情报局距今最近的一个事件显露了美国技术上的巨大优势在苏联“使用人工情报来支持通信情报获取（using Humint in Sigint support）”技巧面前变得丝毫不起作用，这个技巧实际上由一些雇佣的出卖关键材料的叛国者来完成，例如 Walker 家族［51］。

其次，窃取内容通常并不是想要的结果，在战术情况下，其目标通常是探测与破坏节点，或者干扰通信。干扰不但包括噪声插入，还有主动欺骗。在二战中，盟军利用德国的扬声器作为假指挥官发送错误指令给德军夜战者，这是在认证技术的发明之前的一场智力战。更近一点，正像我在有关生物测量学的章节中提到的那样，美国空军已经部署了许多基于声波纹的尖端系统。在前面的章节中我已提到情报机构与军事行动分队之间紧密的关系，前者想听到对方的通信，后者是阻止他们的应用［63］。在这些目标间的折衷办法很难找到，如果不干扰你不能读取的通信内容，那也就等于告诉了敌人你得到了哪些情报。

实际上，如果敌方使用密码学（即使是在应该用的情况下），问题就可以简化，这就减少了军事分队与情报机构的困难，你就可以像正常的那样转向无线电测向或者实施通信线路的摧毁。这些方法不但包括了硬杀伤方法，像挖出电缆或者埋掉电话交换机（这两种方式盟军在海湾战争中都用到了），还包括了软杀伤方法，如干扰，将两者结合则是比较经济的。在只需要破坏某个通信线路相对短的时间段的情况下，干扰才是比较有用的权宜之计，但通常比较昂贵：不但是指它占用了设备，干扰发射台本身也成为一个目标（也有很多更有效的例子，例如反卫星传输线路，此时用从隐蔽地点发射的可控制能量的密集光束就能干扰向上传输线路）。

在民用基础设施，尤其是互联网上，使用频率的增加，提出了如下问题：有计划地拒绝服务攻击是否会被用于干扰通信（有塞尔维亚信息战基层组织试图攻击北约网站这类的轶事传闻）。这一威胁仍然被认为是足够真实，以至于许多西方国家开辟了独立的内部网络专供政府与军队使用。

16.3.3　保护技术

从上文便可清晰地看到，通信安全技术不仅包含保护内容的真实性与保密性——这可以利用一个相对简单的方法，诸如加密与认证协议获得成功——还包含防止通信分析、测向、干扰和有形摧毁。如果是应用在通信线路这个层次，加密可以排在这些应用中的第一位，因此所有通信线路看起来在所有时间都有一个伪随机的位流，不管是否有信息传输。但是只对通信线路层加密在通常情况下并不足够，因为敌人俘获一个单网络节点就可能让整个网络处于危险中。

单纯加密不能有效防止对截取、无线电测向、干扰和对通信线路与节点的摧毁。正因为这样，才需要不同的技术。比较明显的解决方案有：

- 专用线路或光学纤维。
- 高度定向的传输通信线路，例如利用红外激光的光学通信线路，或者利用高度定向天线和极高频（即 20 GHz 及其以上）的微波通信线路。
- 低概率截取（Low-probability-of-intercept，LPI）、低概率位置锁定（Low-probability-of-

position-fix，LPPF）与抗干扰无线电技术。

前两种办法是非常容易理解的，而且在可行之处它们通常是最好的办法。光缆网络是难以完全破坏的，除非敌人知道光缆的位置，然后用物理手段切断。即使在大规模火炮的轰炸后，斯大林格勒的电话网络在整个包围期依然在使用（被双方）。

第三种选择本身就是一个有着重大价值的课题，我下面就要讲到（尽管只是简单地）。

有大量的低概率截取（LPI）/低概率位置锁定（LPPF）/抗干扰技术在扩展频谱通信领域发展起来，包括频率跃迁、直接定序扩展频谱（Direct Sequenlce Spread Spectrum，DSSS）和突发性传输。从二战前期起，扩展频谱导致了一个有重大价值的工业发展，这一技术（尤其是 DSSS）应用于大量其他问题的研究，从高分辨率（在 GPS 系统中）到数字图像中的版权标志测距（我将在后面进行探讨）。我们依次看看这三种方案。

16.3.3.1　跳频

跳频是扩展频谱系统中最简单易懂和最易实现的方案，顾名思义，它们就是从一种频率快速地跃迁至另一频率，其频序是由授权的当事人所知的伪随机序列决定。众所周知，跃迁器是 1940 年由女演员 Hedy Lamarr 与电影剧本作者 George Antheil 在晚宴上发明的，她们发明的这项技术是在不被敌人察觉或干扰他们发射的情况下用于控制鱼雷 [484]。跳频雷达是（大概与此同时）由德国独立发展起来的 [686]；作为对英国干扰技术的稳定提高的回应，德国技术员为了让他们的设备能够正常运转而每天改变频率，然后几小时，直至最后每几秒就改变一次 [627]。

跃迁器对不知跳序的敌人的干扰有抵抗能力，这样的敌人可能不得不干扰许多的波段，因此相对于敌方已知情况下进行干扰所必需的能量，这需要更多的能量。输入信号带宽与传送信号带宽的比率称为系统的过程增益。因此，一个 100 位/秒的信号以 10 MHz 的频率传播的话，其过程增益为 $10^7/10^2 = 10^5 = 50$ dB（分贝）。干扰边际定义为干扰能与信号能的最大容限比率，实质上过程增益是对实现过程和其他损耗取模（严格地说，过程增益由最小位能噪密度比划分）。对一个不能预知跳序的敌人而言，最适宜的干扰策略就是部分波段干扰——对足够的波段进行干扰，使得在信号中插入不可接受的错误率。

尽管跃迁器可以提供一个可观的干扰边际，它们对那些仅仅想探测信号是否存在的敌人而言，几乎提供不了什么保护。环视感兴趣频段的信号分析接收机经常能截取这些信号（依据恰当的带宽、环视速率和停顿时间，它还可能几次截获到某一跃迁信号）。

然而，因为频率跃迁器实现起来是那么的简单，它们常常应用于战斗网络系统中，例如人载无线电，只有每秒 50 ~ 500 的慢速跃迁率。为了破坏其通信，敌人就需要一个快速的或功率强大的干扰发射机，这在战场上是十分不便的。快速跃迁器（理论上定义为跃迁频率超过比特率；实际上也就是每秒的跃迁率在 10 000 以上）可以超过大型干扰发射机的工作限度。

16.3.3.2　DSSS

在直接时序扩展频谱（DSSS）中，我们通过一个很高速率的伪随机定序来增加信息的承载时序，通常由某种序列密码来生成。它通过增加带宽扩展了频谱（见图 16-1）。这种技术最先由一个瑞士工程师 Gustav Guanella 在 1938 年的专利申请中进行了描述 [686]，于 20 世纪 50 年代在美国广泛发展起来。其第一次大规模的部署是在 1959 年的柏林。

像跃迁器一样，DSSS 可能提供大的干扰边际（这两种系统具有相同的理论性能），但也能使其信号更难于截获。窍门便是调节某些特征，使得在截取地，如果不知道用于恢复的扩

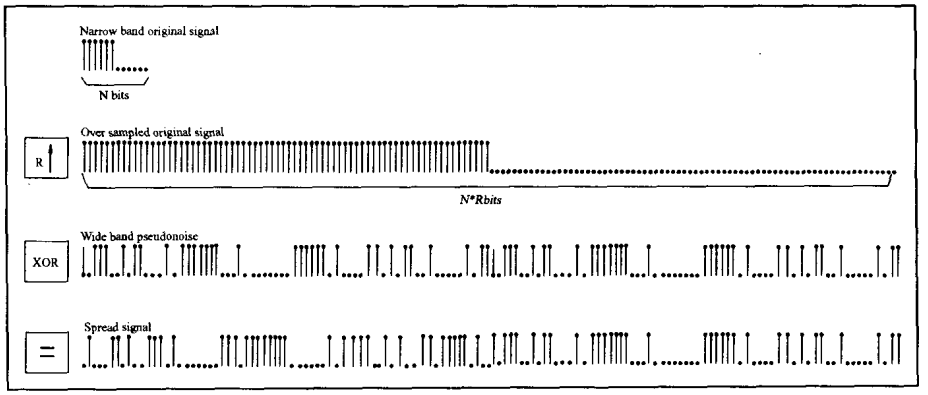


图 16-1 DSSS 的扩展（Roche 与 Dugelay 友情提供）

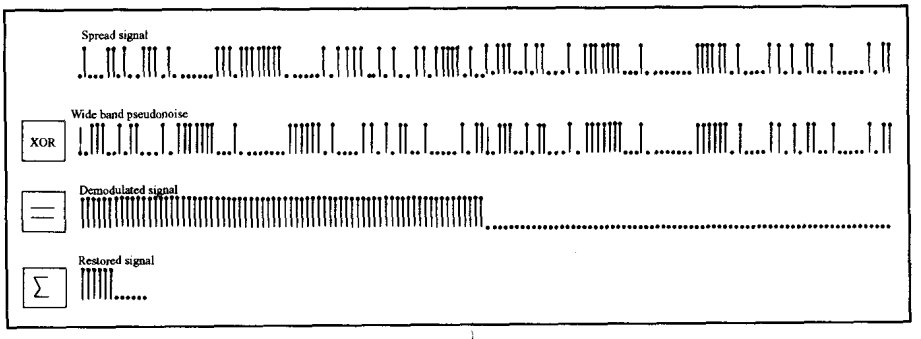


图 16-2 DSSS 不扩展（Roche 与 Dugelay 友情提供）

展时序，那么信号的强度会非常低以至于消失在噪声层中。当然，因为抗干扰信号应该是高能量的，而 LPI/LPPF 信号却要求低能量，所以同时进行这两个工作是很难的；通常的做法是在 LPI 模式下工作，直到被敌人发现为止（例如，在雷达测距范围内），然后将发射机的能量提升到抗干扰模式。

有许多关于 DSSS 的文献；同时这项技术现在已被商业领域所采纳，像各种移动无线电与电话系统中的码分多址（code division multiple access，CDMA）。DSSS 有时被称作“加密射频”，而且引发了许多衍生物。举个例子，当基本调制模式是调频不是调幅时，就叫做线性调频脉冲（chirp）（有关这方面的经典基础数学与技术导论参考书是［616］）。因为各种原因，其工程复杂性比用频率跃迁要高。例如，同步器尤为关键，有权获得参考时间信号（像 GPS 或自动时钟）的用户可以相当轻松地进行；当然，如果你不能控制 GPS，你可能会被同步器任意的攻击；即使你控制了 GPS，其信号也可能被干扰（最近有报道称法国干扰了希腊境内的 GPS，目的是想破坏英国企图卖给希腊政府 250 辆坦克这一事态，因为在这笔交易中，法国是一个主要竞争对手。这就导致英国坦克在荒野的小径中迷路了，当该计谋被揭穿后，希腊人都觉得非常可笑［757］）。其他的策略就是让你的用户轮流提供参考信号。

16.3.3.3 突发性传输

顾名思义，突发性传输就是在敌人无法预料的时段里用非常短的时段来压缩数据并发送出去，又名时间跃迁。通常它们的抗干扰能力并不如想像的那样好（除非以高数据速率扩展

频谱），但它们很难被截取；如果工作循环比较低，环视接收机很容易漏掉它们。它们常常用于同特种部队和情报人员联络时发射的无线电波。

该项技术的衍生物是电光突发性（meteor burst）传输（又名电光散射（meteor scatter））。这依赖于每天数不尽的陨石微粒穿透地球大气层，每一个都留下一个持续约三分之一秒的电离轨迹，并因此在一个“母站”与一个可能达一百英里长几英里宽的地区之间提供一个短暂的传输路径。母站连续地发射，一旦某个“子站”接收到，作为对母站信号的回应它便高速传送数据包。在隐蔽的地面指挥所里用低能量级，用大约 5 分钟的平均反应时间，在 500～1500 英里的距离内，可能获得 50 位/秒（bps）左右的平均数据传输速率；如果用高能量级，且在高纬度地区，平均数据速率可高达每秒数万位。

就像特种部队一样，在阿拉斯加的美国空军用电光散射作为早期预警雷达的候补通信方式，也可用于像监测非洲莱索托降雨量之类的民事应用。在瞄准机会的市场（niche markets）上，因为低比特率与高等待时间可以被接受，但设备的规模尺寸与价格更重要，所以电光散射很难被攻击成功（其相关技术在 [676] 中有所描述）。

16.3.3.4　隐藏与抗干扰的联合

在以下各方面还有相当复杂的权衡：不同的 LPI、LPPF、抗干扰技术和其他方面性能，如它们对衰减和多通道的抵抗以及可同时接纳的用户数量。在特定的干扰技术面前其行为也是不同的，像快变频率干扰（干扰机反复地环视目标频带）与重发器干扰（干扰机按其所能达到的最低的间隔跃迁进行）。某些类型的干扰还进行相互的转化；举例说明，不足以进行全信号拦阻的敌人可能进行部分时间干扰，在 DSSS 中通过发射能覆盖大部分可利用频谱的脉冲，在频率跃迁中则进行部分波段干扰。

也存在着工程方面的权衡。举个例子，在能量这个层面上讲，DSSS 倾向于比频率跃迁高一倍的效率，但在给定设备复杂性的前提下，频率跃迁可以提供更大的干扰边际。从另一方面来讲，利用测向技术对 DSSS 信号进行定位是更加困难的 [287]。

系统残存能力方面的需求可以进一步加强其约束。要防止那些截获了一些无线电信号并因此推断出其现行的关键材料的敌人利用它来干扰整个网络系统。

一个典型的现代军事系统会联合使用紧密射束、DSSS、跃迁和突发性。

- 英国武装部队应用的美洲豹战术无线电在九个 6.4 MHz 波段中的某一个进行跃迁，且有一个容易改变零讯号位置的天线，零信号能够被干扰发射台或敌方的截获站识别出来。
- DSSS、跃迁和时分多址（Time Division Multiple Access，TDMA）都应用于联合战术信息分布式系统（Joint Tactical Information Distribution System，JTIDS），它是由机载报警与控制系统（AWACS）用于与战斗机联系的美国数据链路系统 [677]。TDMA 将传输与接收分离开来，并且让用户知道何时能够得到他们的运行轨迹。DSSS 信号的数据速率达 57.6 KHz，其芯片速率为 10 MHz（因此干扰边际为 36.5 dB），在 255 MHz 波段上能够达到最低跃移为 30 MHz 的跳数。这些跃迁码对所有用户可用，而其扩展码只有个别线路可以接收。其基本原理就是如果设备捕获导致了对扩展码的损害，其系统本身将只允许被 10 MHz 的波段所干扰，而不是整整的 255 MHz。
- 军事之星（MILSTAR）是美国的卫星通信系统，其在地球同步轨道上只有一度角度的电波束（20 GHz 向下，44 GHz 向上）。这一窄波束的效用是用户可以在距敌人三

英里的范围操作而不被其发现 [287]。其干扰防护不同于跃迁；其信道跃迁在 2 GHz 波段上是每秒几千次。

- 设计用于控制 MX 导弹的系统（尽管最后没有部署）在 [337] 文中有描述，并且给出了一个有关极端残存能力工程的例子。为了经得住先发制人的核打击，系统必须能够承受重大级别的通信结点摧毁、干扰与大气噪音。所采纳的设计方案就是一个动态重新配置的网络系统，其频率跃迁器为 450 KHz。
- 法国战术无线电采用了远程控制。战士可在距无线电台 100 米处利用手持机。这就意味着对高能发射站的攻击并不能很大程度上威胁到部队 [216]。

还有一些用于系统层次上的措施，例如干扰相消，其思想就是用一个你正在干扰的波段进行通信，同时对方进行干扰的波形为自己的无线电所知，因此它们相消或者跃迁。这就迫使敌人将其可用能量在一个相对更大的带宽上进行发射，使得干扰变得更加困难，同时使得信号情报工作也变得困难 [644]。

16.3.4 民用与军用的交互

民用与军用通信日益相连。沙漠风暴军事行动（Operation Desert Storm）（针对伊拉克的海湾战争）广泛利用了海湾国家的民用基础设施：在一个相对短的时间内利用卫星、无线电通信线路与租借线路组建了一个巨大的战术通信网络。来自美国不同武装部队的专家声称战争中通信能力的效果起了决定性作用 [398]。显然军队与民兵武装都会不可避免地攻击民用基础设施以防被敌人利用。现在，卫星通信传输尤其易受上传线路干扰的攻击。基于卫星的系统，如 GPS，在演习中就被干扰过；已有关于过分信赖而导致系统脆弱性的探讨 [310]。

全球定位系统（GPS）给出了一个变得越来越相互依赖的例子。是由美国军队海军系统发展起来的，并且具有选择可用性功能，用以将精度限制在 100 码左右，除非用户拥有相关的加密密钥。可这在沙漠风暴中却不得不关掉，因为没有足够的军用 GPS 手持机可以配备，所以不得不用民用设施作为替代。随着时间的推移，GPS 手持机被证明是如此有用，尤其在民航方面。因此联邦航空局（FAA）帮助其寻找出路，克服了选择可用性技术限制，并能提供约为 3 码的精度，与以前声称精度为 8 码的标准军事接受机相比更加出色 [270]。最后，在 2000 年 5 月，克林顿总统宣布终止选择可用性措施（这反而保护了其在战争状态下的可用性）。

民用基础设施提供了一些只为政府机构（尤其在情报领域）使用的防御系统。我在前面曾讨论过预付移动电话，其提供了相对公平的匿名度；安全网络服务器也提供了某些可能；其他的例子如匿名收信人，是一个接收加密电子邮件，然后解密再将包括已解码的信封在内的信件送达目的地的系统。我将在 20.4.3 节中详细探讨这一技术；匿名网络的先驱之一可能是美国海军 [637]。研究阴谋的理论家怀疑这个系统的公开使用会为机密信息提供密封传输的机会。

迄今为止，尽管在网络上的通信安全很大程度上被理解为信息保密性与真实性，看起来好像将来会演变得更像军事通信，在网络中，各种各样的拒绝服务攻击、匿名与欺骗变得日益严重，我将在后面谈到这个话题。从现在起，让我们关注不得不对付目标探测与武器制导电子战的各个方面。同时这些也是干扰学与欺骗学不得不高度发展的领域（实际上，尽管在开放性文献中，有关反雷达电子攻击与防御应用的文章比通信多得多，许多同样的材料却明

显对两者都适用)。

16.4　监视与目标探测

尽管某些传感器系统利用被动测向，但用于发现敌方目标并进行武器导航的主要方法却是声纳、雷达及红外线。其中第一个发展起来的是声纳，在一战时就发明并应用于实战（所用的名字是潜艇探测器（Asdic））[366]。除了在水下战争中，主要的传感器就要数雷达了。尽管雷达是由 Christian Hülsmeyer 于 1904 年发明用作海上反碰撞设备，其重大发展却发生在 20 世纪 30 年代，并在二战中为所有参战国使用 [369, 424]。为其发展起来的电子攻击与防御技术，比起应用其他传感器的系统得到了更好的发展，与雷达相关，“电子攻击”通常意味着干扰（尽管理论上它还包括窃取技术），而“电子防御”则指那些用于维护至少某些方面雷达性能的技术。

16.4.1　雷达类型

一个范围广泛的系统得到了应用，包括搜索雷达、消防控制雷达、地形跟踪雷达，反粒子辐射雷达与气象雷达。它们的信号特征有广阔的变化，举个例子，具有低射频（RF）与低脉冲重复频率（pulse repetition frequency，PRF）的雷达适于搜索，而高频、高脉冲重复频率设备适于跟踪。涉及该技术的一本好参考书是由 Schleher 所写的 [677]。

用作搜索用途的简单雷达设计中含有一个可旋转的天线，其发射一系列脉冲并检测回波。在数字电子学时代之前，这是实现雷达的简单方法，显示管的视野也可能随天线同步进行机械转动。消防控制雷达通常利用圆锥形扫描，电波能够在目标位置四周被圆形跟踪，其回波的振幅能够直接驱动定位伺服系统（私武器控制）。现在的电波通常利用多天线元件电动生成，但其轨迹环绕于残留中心。许多雷达有一个距离闸，用来在距天线特定距离内聚焦目标的电路；如果雷达不得不跟踪所有的物体，比如说 0 ~ 100 英里内，那么其脉冲重复频率就会受无线电波穿过 200 英里所花时间的限制。通常就有了关于角分辨率与跟踪性能的推论。

多普勒雷达通过回波信号中频率的变化来测定目标物体的速度。从杂乱物中识别运动的目标是非常重要的；回波从地面返回。多普勒雷达也有速度闸，将注意力局限于那些相对于天线而言，径向速度在某一约束条件之内的目标。

16.4.2　干扰技术

电子攻击技术有被动与主动之分。

早期广泛应用的防范技术是金属箔条——将箔切割成目标信号的半个波长的薄条，然后散布出去以提供一个假回波。临近二战结束时，盟军战斗机每天撒下 2000 吨金属箔条来降低德国的空中防御。金属箔条可以由直接企图冲破敌军防御体系的战斗机投下（这不理想，因为它们随后会出现在拉伸信号的反射点）或支援战斗机投下，也可利用火箭或炮弹向前发射到一个合适的散布面。而主要的对付金属箔条的反防范措施是多普勒雷达的应用，因为金属箔条非常小，因此它近似静止不动，所以能非常轻易地与运动目标区分开。

其他的技术包括重复发射雷达信号的主动式重发器之类的小型假目标，也有进行简单反射的大型假目标；有时一个飞行器（比如直升机）可以为更有用的运输机充当假目标，这个

原理是十分普通的。在国内使用无线电测向（RDF）的武器可用特制的发射诱惑性射频信号的无人驾驶飞机来诱骗敌人，而红外制导导弹可用照明弹来牵制敌人。

花大资金投入的被动防范手段就是欺骗，因为减少了飞行器的雷达截面（radar cross-section，RCS），因此只能在非常短的距离内为敌人所察觉。也就意味着，举个例子，敌人不得不将其空中防御雷达排布得非常稠密，因而不得不购置许多。秘密行动包含许多技术，而对其专门的探讨则超出了本书的范围。有些人将其视为“极其昂贵的黑胭脂”，但对其而言却不仅仅是这样。因为战斗机的雷达截面是其外形的典型性能的反应，它装有 fly-by-wire 系统，通过低雷达截面持续显示其外形用以鉴别敌人的发射器。

积极防范措施更为多变。早期的干扰发射机是利用目标雷达的频率范围简单地产生许多噪声；这项技术通常称为噪声干扰或阻塞干扰。有的系统采用规则频率模式，如脉冲发射器或快速移动发射体，它在感兴趣的频率范围内往返移动（又叫 squidging oscillators）。但这种信号非常容易被阻塞——一个窍门便是利用警戒频段接收器，该接收器采用与正使用频率相邻的频段，如果接收器显示为干扰信号，则封锁这个信号。也需要注意，干扰并不局限于某一方。如果雷达使用者的对手也使用的话，雷达本身也可以从辅助无线中发射合适的假信号，以伪装真实信号或使其防御超负荷。

在对抗的另一端，存在着硬杀伤性技术，比如抗辐射导弹（anti-radiation missiles，ARM），经常由支援飞机发射，具有定位敌方信号源的功能。针对此类武器的防御包括时断时续地使用假目标发射器与闪光发射器。

在中间则有关于欺骗干扰技术的一大套东西，许多用作自身防护的干扰发射器从某种程度上说就是一种欺骗干扰器；阻塞与抗辐射导弹技术更适合于支援飞机应用。

带一个自身防护干扰器的通常目的是拒绝将距离与方位信息发给攻击机。最基本的手段就是反向增益干扰或反向增益调幅。基于如下观测资料：攻击机天线的指向性通常并不那么完美；除了主瓣之外，它还具有侧向波瓣，通过侧瓣也能发射与接收能量，尽管相比主瓣而言其有效性要低得多。侧瓣特征曲线可通过观察其发射信号绘制出来，而且干扰信号也能随之产生，因此净发射是天线方向的特征曲线的翻转。就攻击机雷达而言，外观就像信号一样来自各个角落；你在雷达屏幕上看到的不再是“尖头脉冲”，而是以自己的天线为中心的一个环。反向增益干扰对老一代的圆锥形扫描火控系统非常有效。

更一般地，这些技术使用延时和/或频率的规则变化来重新发射雷达信号。这可能要么是不相干的，在这种情况下干扰发射器被称作脉冲转发器；要么是相干的——也就是存在恰当的波形——叫作重发器（现在将接收到的波形存储到数字无线电频率存储器（digital radio frequency memory，DRFM），然后利用信号处理集成电路进行伪造是十分普遍的）。

初级防范措施是 burn-through。通过降低脉冲重复频率，使停顿时间增长，因此回波信号增强——只是以降低精度为其代价。更高级的防范措施是距离闸实现（range gate pull-off，RGPO）。在这里，雷达发射出许多比真实脉冲更强的假脉冲，因而引起接收器的注意，然后将它们从相位中移去，于是目标物便不再出现在接收器的距离门槛之内。与之相似，利用多普勒雷达的基本办法就是速度闸实现（velocity gate pull-off，VGPO）。对付老一代的雷达，成功的 RGPO 可使雷达中止锁定，目标便从屏幕上消失了。现代雷达可以很快重新获得锁定，因此 RGPO 必须要么重复进行，要么与其他技术联合使用——通常是用反向增益干扰，并同时中止角位跟踪。

初级的反防范措施是抖动脉冲重复频率。每一个发出的脉冲要么延迟，要么准时，这依赖于由序列密码或随机数生成器生成的滞后定序决定。这也意味着干扰器并不能预测下一个脉冲何时到来，因此不得不跟在它后面。这样的追随干扰只能识别那些再度出现的假目标。而针对反防范措施的对策对雷达而言就是拥有一个前缘跟踪器，它只回应第一个返回脉冲；而与之对应的防范措施可以在一个足够高的能量水平上进行干扰，接收器的自动增益控制电路能够捕获或者进行掩护干扰，干扰脉冲足够长以至于能掩盖最大的抖动周期。

下一个看家手段可能包含了战术成分。金属箔条经常用于迫使雷达进入多普勒模式，该模式使得脉冲重复频率的抖动更难了（因为连续的波形对多普勒而言比脉冲更好），而前缘跟踪器可以与频率灵敏性和智能信号处理联合使用。举个例子，真实目标回波起伏不定，而且有逼真的加速，而简单的脉冲转发器与重发器或多或少发射出稳定的信号。当然，对设计者而言这总是可能的，因为他们是如此的聪明；米格 29 在水平作战时可以通过快速拉升动作获得比一些雷达设计者的预期要快得多的加速度，因此飞行员可用这个策略破坏雷达锁定。当然，现在有高达每秒数百万条指令的机器可用于制造逼真的假回波。

16.4.3　高级雷达与反测量措施

许多先进技术给干扰器提供了优越的条件。

二战中最早由德国发展起来的脉冲压缩是一种直接时序扩展频谱脉冲，用一个匹配的过滤器过滤回波并再次进行压缩，这能产生 10 至 1000 的过程增益。脉冲压缩雷达有抗脉冲转发干扰的能力，但易受重发干扰器的攻击，尤其是那些具有数字无线电频率存储器的重发器。然而，在你不希望目标首先检测到你时，低可能性截取波形的应用是非常重要的。

脉冲多普勒非常像多普勒，发射一系列相位稳定的脉冲，其在许多高端市场上已占有了绝对优势，应用十分广泛。举个例子，在防低空飞行入侵者的空防下视下射（look-down shoot-down）系统中。谈及初级脉冲跟踪雷达，不同的射频与脉冲重复频率有不同的特性：对一个明确的距离/速度而言，我们想要低频/脉冲重复频率，也想降低杂音——但这会导致许多盲点。不得不对付许多威胁的机载雷达利用较高脉冲重复频率，且只搜索那些速度高于某一阈值的运动体，比如说 100 节——但其在后翼追逐（tail chase）中显得微弱。通常的折衷办法是中等脉冲重复频率，但会在空中打击中造成严重的范围不确定。另外，搜索雷达要求长而多变的脉冲，而跟踪雷达却只需要短而和谐的脉冲。一个优点就是脉冲式多普勒可以识别一些非常特殊的信号，例如喷气式引擎里涡轮架提供的调制。主要使用的反脉冲式多普勒欺骗战略是速度闸实现，尽管其衍生技术就是利用欺骗性回波导致多个速度闸值。

单脉冲已变成最流行的技术之一。举个例子，它用于 Exocet 导弹，在福克兰群岛战争中被证明是非常难于干扰的。其思想是采用 4 个相连的天线，因此方位角与海拔数据可以通过每一个回应脉冲利用干涉技术计算出来。单脉冲雷达非常难于干扰且干扰代价十分昂贵，除非利用其设计缺陷；常用的技术包括诸如信息干扰和地形跳跃之类的策略。推荐的首选防御战略就是利用拖曳的假目标。

一个最为现代的计策就是被动相干定位。锁定留意目标的无声警戒系统根本就没有发射器，而是利用商用无线电与电视广播信号来检测与跟踪空中目标 [508]。由于是被动型，其接收机难于被定位与攻击；而且想破坏该系统必然会摧毁主要的民用基础设施，出于各种宣传的原因，这往往是敌人不愿意做的。这一战略对于某些秘密行动技术而言具有中等

效果。

数字无线电频率存储器与其他软件无线电技术支持了许多更为复杂的攻击与防御前景。雷达与干扰的波形也可能适用于战术情况，比以前具有了更大的灵活性。然而对光谱、时间与空间特征的各种组合并不是关注的全部。有效的电子攻击很可能继续要求利用武器和谋略对不同的被动与主动工具进行有效的协调与配合。情报的重要性和精心欺骗设计的重要性都在上升。

16.4.4　其他传感器与多传感器问题

我所讲的许多关于雷达的话题也同样适合于声纳，相当一部分也适用于红外线。使用被动性假目标（照明弹）防范早期的寻热导弹是非常有效的，因为这些导弹利用的是机械探测器，但面对现代的一体化信号处理过程的探测器基本上失效。照明弹就像金属箔条，对于目标而言，它们很快地减速，因此攻击者可以用速度或加速度将之过滤掉。照明弹又像重复干扰器，它们的信号与真实目标相比显得相对稳定而且很强。

主动的红外干扰比雷达干扰要更难，因此应用不是那么广泛。它是通过能造成混乱的某一速率或某一模式发射脉冲以得知敌方传感器的特征。一些红外防御系统开始使用激光使引人武器的传感器失去威力；最近有人承认许多观察到的 UFO 实际上是由各种干扰引起的（包括雷达与红外线）[75]。

一个正在增长的领域是多传感器数据融合，依靠来自雷达、红外传感器、视频照相机，甚至人的输入联合提供比任何单一传感器能够独立提供的更好的目标识别与跟踪。举个例子，轻剑（Rapier）防空导弹利用雷达获得方位角，而其跟踪功能是利用可视条件通过光学原理完成的。数据融合可能比其看起来更难一些。就像在 13.8 节中所讨论的那样，联合两个警报系统通常会导致虚假警报与遗漏警报比率两者之一的增加，并会导致另一个变得更糟。如果当你在雷达或红外线上看到一个尖头信号就命令你的战士紧急起飞，可能存在更多的假警报；但如果你只在当雷达与红外线上同时显示尖头信号时命令起飞，敌人就会比较轻易地干扰你或偷偷逃掉。

当攻击者自身处于一个易被反攻击的平台上时，例如战斗轰炸机，系统问题变得更为复杂。它可能携带有威胁识别、测向及导弹临近警告；同时其接收器也会被自身的干扰器隔离得近似一个聋子。通常的对付办法是将干扰器在非常随机的时间关闭一个非常短的“浏览”时段。

存在多个友好与敌对平台时，事情变得更加复杂。每一方可能都有特别的拥有高能专用设备的支援飞行器，这就从某种程度上成为了能量战——“谁的效率最大，谁就会赢。”地对空导弹（SAM）的肩带上可能装备有多个不同频率的雷达，使干扰变得非常困难。干扰（秘密行动也是）的综合效果是减少了雷达的有效距离。但干扰边际也起作用，还有看谁拥有更多的飞行器；谁的战术使用得当。

考虑多个飞行器的情况，也需要有一套识别敌我的可行办法。

16.5　敌我识别系统（IFF）

二战的技术革新——尤其是喷气式飞机、雷达与导弹的发明——使得肉眼识别目标已不现实，寻找自动敌我识别（identify friend or foe，IFF）手段是十分迫切的需要。在那次战争

中，早期的IFF系统便出现了，它利用的是飞行器序列号或“日期代号”；但这相当于对电子欺骗不设防。自从20世纪60年代开始，美国飞机使用的是XII标志系统，带有密码保护措施，就像2.3节中所讲的那样。当然在这里用到的不是相当难的密码学，而是一种协议，一种操作上的难题。

XII标志系统有四种模式，其安全模式利用32位的查询但回复为4位。这是继承了其前身X标志系统的装置；如果询问与回复不太长，雷达脉冲重复频率（尽管会精确一些）就会降低。XII标志系统以每4毫秒一个的速率发射12～20个查询口令。在原始的执行方式中，实际回复与期望回复之间的代数差别产生的位置偏移显示在屏幕上。结果是如果敌机没做回复或做了随机回复，而己方飞机则会回复使其显示在屏幕的中心附近，那么屏幕就会闪亮。反射攻击就被防止了，而且米格中间人攻击就更难了，因为查询口令使用的是聚焦天线，而接收器为全向（实际上，用作查询口令的天线是一种典型的火控雷达，在更老的系统中则为圆锥形扫描方式）。

我在2.3节中提到，只进行密码保护并不是完美无缺的：敌方可能记录并重放正确的询问，通过利用你的IFF信号进行测向。这在有许多飞行器与发射台的密集作战地区可能真是一个问题，就像冷战期间东德与西德的边界，还有中东的某些地区至今尤存。在这种情况下，回复信号可能因为附近的飞机发出的信号的重叠而降低精度——这就是众所周知的歪曲效应。而另一方面，飞机的脉冲转发器在面对许多查询的情况下可能不能恰当译码——即fruiting效应。总结以上问题，我们需要最小化查询口令与回复的长度，这限制了密码保护措施的有效利用。结果，皇家空军反对美国将XII标志系统作为北约标准的要求而继续使用二战产品X标志系统，每30分钟改变一次密码（有关X标志系统、XII标志系统、皇家空军与美国空军的辩论的详细情况可在[348]文中找到）。这也是另一个将密码用于增加系统设计的效果却总是存在惊人困难的例子。

系统层次的问题甚至更难于控制，其要求是要识别敌方的军队，但IFF系统依赖于与只能确切识别己方军队的目标物体之间的协作。任何中立者，还有那些携带有缺点的或错误的脉冲转发器设备的己方军队根本无法与敌军区别开来。因此，在中立者不包括在内的地区（例如战争时期的海军作战部队邻近的区域），IFF可以用作一个初级装置，它通常更多地用作常规方法的附属方式，例如用于飞行编队之间的联系。从这个角度上看，它还是蛮有效的。

自从海湾战争开始，许多更高级的系统发展了起来，将IFF用于地面部队，因为在那场战争中25%的盟军伤亡人员是由“我方射击”造成的。美军的某一系统综合了激光与射频成分，射手有激光，士兵有脉冲转发器；当士兵被一个合适的询问口令锁定时，他的设备利用跳频无线电发射一个“不要向我射击”的信号[820]。其扩展应用允许飞机利用毫米级的微波无线电发射目标意图。这套系统在2000年以后装备部队。英国发展了一个更便宜的系统，名为鹊（MAGPIE），在这个系统中，己方飞行器携带一个低截获概率的毫米波发射器，射手则携带一个方位接收器[381]（不像美军配备的那样，英国步兵没有任何保护）。其他国家也在发展其他的系统。

16.6 定向能量武器

在20世纪30年代的后期，传说纳粹分子发明了一个能烧掉飞行器点火装置的高能无线

电波系统，这一传闻在英国与美国引起了恐慌。英国科学家研究之后得出结论：这是不可能的［424］。20世纪30年代，在给定相对低能的无线电发射器，简单但健全的飞行电子系统的前提下，英国科学家的结论是正确的。

可是在原子弹出现之后事情就开始发生变化。核装置的爆炸会产生大量的伽马射线光子脉冲，通过康普顿散射依次置换大气分子中的电子。引起的巨大电流增加了电磁脉冲（electromagnetic pulse，EMP），这可被看作是在一个相当短的时间内产生的振幅巨大的无线电波。

在地球大气层内发生的核爆炸，尽管在低频上也有足够的能量并以无线电闪光的形式为上千英里之外所见，但电磁脉冲能量在VHF和UHF波段上占了绝对优势。在数十英里距离内的爆炸，射频能量就可以引起足够大的电流以至于损坏不被硬杀伤所击的绝大部分电子设备。相信地球大气之外的冲击波效应会让这一情况更糟糕（尽管还从没有试验过）。伽马光子在冲击地球大气之前可以飞行数千英里，能够电离形成陆地规模大小的天线。据估计，北欧的绝大部分电子设备可被北海上空250英里高度的百万吨级爆炸所烧毁。因为这个原因，核心军事系统要非常谨慎地进行屏蔽。

20世纪80年代中期，苏联开始了一项关于无核电磁脉冲武器的研究计划，之后西方国家对于电磁脉冲的关注也增加了。就在那时，美国正在欧洲部署“中子弹”——在不破坏建筑物的情况下可以致人于死地的辐射武器。苏联将其描绘为“资本主义炸弹”——在将财产完好无损保留的前提下给人致命一击，并扬言要用“社会主义炸弹”摧毁财产（以电子的形式）同时不伤周围的人一根毫毛。

在二战结束时，关于中空磁控管的发明使得有可能建立一个能量足够高的雷达，它能在几百码的距离内摧毁毫无保护措施的电子线路。从电子管向晶体管与集成电路的转变增加了大多数商业性电子设备的脆弱性。从理论上讲，恐怖组织可以在大卡车上架起雷达，绕某一城市的金融区一周就能洗劫银行。为了在战场上使用，应该提供更紧凑的设备，因此苏联声称建立了由电容器、电磁流体动力学发生器之类器件组成的高能射频设备。

到了20世纪90年代中期，人们普遍担心恐怖分子可能得到前苏联的这些武器致使当局将其卖作商用，以及用作工业电磁屏蔽。有人认为这种努力就像骗局一样消失殆尽，就个人的观点，我也趋于同意这一说法。苏联高能射频炸弹的细节并没有公开，但物理学表明电磁脉冲能量是由空气的绝缘强度和天线的截面所限制的。在核电磁脉冲能量里，对于大气层内的爆炸而言，有效的天线规模可能是几百米；而对于大气层以外的爆炸而言，可能达到几千公里。但对于“通常的”电磁脉冲能量/高能射频，天线好像最多只有几米。北约计划者总结说对核电磁脉冲而言已经得到加强的军事指挥与控制系统不应该受到影响。

至于民用基础设施，我认为恐怖分子用由一吨肥料与原油制成的旧式卡车炸弹就能制造出大量的破坏事件，而不需一个物理学博士来另行设计一个炸弹！无论怎么说，文献［645］是有关电磁脉冲方面的标准参考书。

然而，人们依然关注在美国中部250英里上空的核爆炸产生的电磁脉冲所能造成巨大的经济损失，同时还会直接伤及一些人［53］。这就潜在地为像伊朗和北朝鲜这样的国家提供了威慑武器建造渴望，他们两国都有核野心却只有简单的基础设施。通常来说，对电子通信的巨大攻击对像美国这样高度依赖它们的国家来说，威胁远比对北朝鲜之类的国家大得多，因为它们不怎么依赖电子通信。这一讨论也同样适用于对互联网的攻击，因此我们转而讨论信息战。

16.7　信息战

从 1995 年起，信息战这个词逐渐流行，并通过沙漠风暴实战应用，其流行性得到了催化。在那场战争中，为了降低伊拉克的防御能力，在地面攻击发起之前进行了空中打击；协助盟军的美国国家安全局全体人员的目标之一就是能在战争之初不造成任何人员伤亡，即使伊拉克防空系统那时还完好无损并保持着高度的警戒。这次攻击包含了标准电子战技术的混合，例如干扰器与抗辐射导弹；用巡航导弹攻击其指挥中枢；偷偷潜入伊拉克的盟军特别部队挖出沙漠中的许多通信光缆；还夹杂了据说为使计算机与电话交换机瘫痪而采取的黑客行动（至 1990 年，美国军队就已经有了号召进行病毒制造的竞标活动 [518]）。这些行动使得在空中轰炸的第一天夜晚实现盟军零伤亡的目标得以成功实现。作战计划制定者与智囊机构开始思考如何才能将这个成功扩大开来。

对于其定义却没有达成一致意见。最通常的看法是由 YuLin Whitehead 少校提出的，它由沙漠风暴衍生而来 [790]：

> 在发动常规攻击与行动之前，战略家们……应当使用“信息武器”作为先锋武器使敌军失去判断力。

更富侵略性的观点认为恰当地进行信号行动应包含一切从信号情报到宣传手段在内的行为；而且，现代社会如此地依赖信息，它就应当有能力打垮敌人的战斗力，使之无法应战。

16.7.1　定义

事实上，对于什么是信息战主要有三种观点：

- 这仅仅是当局做了几十年的材料的重新兜售，企图维持当局冷战后获得的预算额。
- 它由广义上的黑客应用所构成——网络系统攻击手段、计算机病毒等等——在国家或国家级组织之间的冲突，为了破坏关键的军事与其他设施，用作实际操作或仅用作宣传目的。举个例子，尽管互联网的设计承受得住热核粒子辐射，但却被莫里斯蠕虫病毒所击倒。
- 它是对电子战学说的延伸，通过控制电磁波谱来控制与战争有关的所有信息。因此，它扩展了传统的如雷达干扰之类的电子战技术，通过增加各式各样的黑客技术，还结合了宣传与新闻管理。

第一种观点是冷嘲热讽的保守派对外宣称的。而第二种是报纸文章里最流行的观点，当然，Whitehead 也持这一观点。它是我在本节中用作引子的话，对于是否真正包含了一些新的东西，无论是技术上的还是学说意义上的，它都没有表达立场。

第三种观点在 Dorothy Denning 的 [235] 一书里有所表达，他对信息战的定义是“作用于目标或利用信息媒体，达到相对于对手获得一定优势的目的。”这个解释涉及面广，不但包含了偷偷闯入，还有所有的电子战以及所有已存在的情报搜索技术（从信号情报到卫星图像再到间谍），还有宣传。她还围绕科索沃战争探讨了网络在宣传和行动主义中扮演的角色 [236]。然而她书中的大部分内容仍集中在计算机安全及其相关话题上。

另一有关信息战的观点，是从一个具有防御设计背景而不是计算机安全背景的作者 Edward Waltz 的书里摘引的 [790]。他是这样定义信息优势的：“搜集、处理与散布一个不中断

的信息流的能力，同时利用或拒绝敌人做同样事情的能力。”这种理论就是说这种优势允许军事行动在没有有效抵抗的情况下进行。这本书比起 Denning 的那本专著，少了许多有关计算机安全问题的技术细节，而是试图重点阐述信息军事行动的军事学说。

16.7.2 学说

当像 Denning 与 Waltz 等作家将宣传行动也囊括入信息战中时，冷嘲热讽的保守派可能评论说什么也没有变。从罗马与蒙古人创造了战无不胜的神话的成就，到二战与冷战期间双方都利用宣传无线电战，再到科索沃战争中对塞尔维亚电视台的轰炸和俄罗斯当局对车臣网络站点的拒绝服务攻击［198］——手段一直在变，但游戏规则并没有变。

但有一个意想不到的转折，可能是因为政府与军方领导缺乏对互联网的了解。当十几岁的毛头小伙让美国政府部门网站大失颜面的时候，熟练的计算机安全专家很可能觉得这与在公共场合的墙壁上涂鸦差不多。毕竟，这做起来容易，去除也容易。但信息战团体也可以像暗中破坏一个国家投入大量资金建立起来的阻止侵入的情报优势一样依葫芦画瓢。

因此在政府首脑及军事首领可以清楚地思考这个问题之前，已有大量的真相需要揭露。举个例子，经常有人阐述说信息战提供了一种不含人员伤亡而取得战争胜利的方法：“只要黑掉伊朗的火力系统然后就等着他们向我们求和。”以下是三个有关的明显备注：

- 长久以来，在不使用武力的前提下用来处理信息系统的拒绝服务攻击大多数时候只能取得短期效果。一台计算机瘫痪了，技术员会发现发生了什么，他们只要从后台进入，重装系统就可运行。几个小时的中止足够让一轮轰炸毫发无损地完成，但好像不太可能让一个国家臣服，在前文中，想通过千年虫造成期望损害的失败就是一个有说服力的警告。
- 从这个程度上来说，发达国家尤其暴露出易受攻击的特点。美国或英国的电力网比起平均发展中国家的计算机化水平要高得多。
- 最后，如果这样的攻击造成伊朗医院内的几十人死亡，伊朗人就不可能将此与普通军事手段造成同样伤亡人员的事件等同对待。实际上，如果信息战把平民作为目标，而不是军人的话，攻击发动者的领导可能会被当作战争犯。皮诺切克案（该案中这位前政府首脑只是因身体原因才免遭引渡）应该是值得深思的。

做了这些讨论后，我将在这节剩下的部分把精力集中于技术问题。

16.7.3 电子战中潜在的教训

从电子战领域得出的最重要的策略性经验，是包含不止一个设备的军事行动比其看起来要更难。当陆军、海军和空军部队不得不协同作战时，事态竟是如此的糟糕——在美国入侵格兰纳达岛时，一个地区指挥官不得不使用他的信用卡到一个付费电话处打电话回总部以发起一场空战，因为不同设施的无线电是不相容的（实际上，这都是软件无线电发展导致的恶果［482］）。当还要包含情报服务时，事态甚至会更糟，因为他们没有在和平时期与战斗人员一起训练过，一旦战争打响，这种组合要花费很长一段时间才能变得有战斗力。局部战争也差不多是这样：在美国目前的制度下，空军可以决定轰炸敌方的电话交换机，但需要从国家安全局和/或中央情报局那里得到攻击的许可［63］。美国陆军的通信策略是要考虑通过传统指挥层次进行通信和扩展利用现存的民用基础设施的需要［672］。

在技术层面，许多概念可以从电子战直接引入到信息保护上来。

- 电子战团体利用监视波段接收器来发现干扰，因此就可以过滤掉（例如，恰好在扫视干扰器通过某一频率的精确时间用空置接收器）。从本质上来讲，利用诱饵地址发现跨度也是这个原理。
- 病毒识别和雷达信号识别也有相似之处。病毒制造者可能使他们的代码多态化，当它传播时形式就发生变化，使其更难为病毒扫描设备的厂家所杀害。与此相似，雷达设计者利用多变的波形来使其很难在数据无线电频率储存器里存储足够的波形进行有效的相干干涉。
- 我们的老朋友——错误接收率与错误拒绝率，可能继续在战术与战略上起主导作用，就像防盗警报或雷达干扰，我们总是值得为造成许多不真实报警的能力（尽管说法有点粗鲁）干点什么：错误报警率一旦超过 15%，系统功效就会大打折扣。至于过滤，通常都会被欺骗。
- 在攻击与防御中涉及到的限制性经济因素，以及新工具可被创造与装备的速度，将会越来越成为软件成本。
- 当涉及干扰时，不让干扰机知道他的攻击是否成功或者取得多大程度的成功就是很有用的。在军事通信中，最好通过降低比特率而不是提高能量的手段来响应干扰；与此相似，当一个不存在的信用卡号出现在你的网站上时，你可能会说，“对不起，错误的卡号，请重试”，但第二次你就应该选用另外一条不同的传输线了（或者攻击者会继续试验）。像“对不起，你所需要的项目暂时没有库存，你将在五天内收到相关信函”之类的说法可能作为新招。
- 尽管深度防御总的来说是一个不错的主意，但你不得不当心不同防御之间的相互作用。电子战中最经典的一个案例就是战略舰将金属箔条撒出以防即将到来的巡航导弹击中舰上的防空大炮。这方面防御的副作用也可以被利用。在网络上最普通的例子是邮件炸弹：一个攻击者伪造了一条攻击性的新闻组信息，看起来倒像是被攻击者发出的，然后受害者就会接到接二连三的辱骂与攻击。
- 最后，某些看法可以从电子战中硬杀伤与软杀伤的不同角色中提取出来。干扰与其他的软杀伤攻击在短期内可能会相对便宜些；它们可用于防范多种威胁；且只具较少的政治后果，但损坏程度的评估还未实现，你可能刚把武器转向其他目标。许多信息战都是软杀伤，这些评论也有望深入探讨。

16.7.4　电子战与信息战的区别

在传统的电子战与潜在地破坏网络的五花八门的攻击之间既存在相似之处，又有区别。

- 粗略地看，有两种战争：公开的战争与游击战。电子战主要在前者情况下盛行，例如在空战、绝大多数海战与沙漠中。在森林与山地里，携带 AK-47 的士兵在同机械化部队作战时能占据优势。而游击战被电子战团体大大忽视了，只是在一定程度上制造并出售雷达以发现狙击手，有时用来隐蔽其迫击炮群。

  在虚拟空间里，“森林与山地”就像是大量不可靠的军队，他们来自于盟军或中立性的平民和组织。分布式拒绝服务（DDoS）攻击，导致许多不设防的机器被破坏，或用于攻击那些通信传输的目标网站，在电子战领域并没有任何的类似物。然而，它

就像对甚至是“开放的”目标如大型商业网站发起攻击的平台，因此在虚拟空间里面，开放的乡村到底处于什么地位还并不明确。

- 对游击战而言，造成不对称优势的另一个可能原因是比较复杂的。大国自己有许多不相容的系统；当用相似的不相容系统与其他大国开战时，并不会带来什么不同，但是与可以制造出简单且相容的系统的小型军队作战时就会稍显劣势。
- 任何一个想攻击美国的人不会像萨达姆·侯赛因一样犯下错误，妄想打坦克战而胜之。游击战应该是常用战术，而且虚拟空间看起来很适用于这种战术。
- 并没有“脚本小子”之类的电子战类似物，人们将攻击性脚本下载下来，并不真正明白它们如何起作用就再发出去。如此能量的武器一般来说是有效的，而且是免费的，在现实空间里这样的事情却很少有类似物。可能最接近的例子要数在像阿富汗这样的国家的无法律地区，所有的人都从事军火交易。

16.8 小结

比起信息战的许多其他领域电子战得到了高度的发展。从技术水平到战术水平再到计划与战略问题，都有许多的经验教训值得总结。当信息战从一个时髦的概念演变到有基础的学说时，我们可以期望这些经验教训对从事者而言非常重要。

研究问题

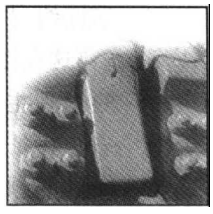
一个非常有意思的研究问题是如何将电子战领域里的技术与经验移植到因特网上去。本章仅仅是概要性的尝试，就是记下可能的类别与区别。

参考资料

有关雷达导论的一本好书是 P.S.Hall 写的 [369]（尽管里面没有技术性的论述）。从雷达到秘密行动再到 EMP 武器的电子战各个技术方面的全面阐述的书，最好的要数 Curtis Schleher 写的 [677]；而总结方面的则要算 Doug Richardson [644]。经典的有关扩展频谱定序的抗干扰性能的介绍请看 Robert Scholtz 的 [686]；而有关扩展频谱的经典数学入门则要参阅 Raymond Pickholtz、Donald Schilling 与 Lawrence Milstein 合著的 [616]；然而这方面的标准参考书一般用 Robert Dixon 的 [254]。[424，425] 是有关英国电子战与科技情报的历史，不仅提供了大量的相关技术是如何发展的，还涉及到战略与战术欺骗的见解，此书的作者是一个真正的内部工作人员，他就是 R.V.Jones。

最后，有关雷达、干扰机与 IFF 系统技术的各个方面的来历有三个值得细读的但可作为补充的不同观点：德国的 David Pritchard [627]、英国的 Jack Gough [348] 与美国的 Robert Buderi 所写的 [142]。

第 17 章　电信系统的安全



我几乎不借助技术攻击。公司要花费上百万美元投资于技术保护，当公司中一个人通过电话呼叫某人时，如果确定他们在公司计算机上做了手脚而降低了计算机的防御能力，或者他将他人正在搜索的信息泄露了出去，这巨额花费就算浪费了。

——KEVIN MITNICK

17.1　引言

因为许多原因使得对电信系统的保护成为一个重要的教学案例。首先，许多分散的系统通常以不太明显的方式依靠底层的固定或移动电话网络系统。其次，在电信方面安全失败的历史具有启发性。早期是一些电话盗用者为了打免费电话对电话公司进行攻击；于是电话系统的易受攻击性开始被那些为了逃避警方电话窃取的盗贼所利用；然后额外收费电话欺骗也被引入，创造了更大规模电信欺骗的趋势；当电信市场更加自由化之后，一些电话公司开始对其他公司客户实施攻击，有的公司甚至相互进行直接的攻击。在每一个阶段，采取的防御手段不但十分昂贵，而且因许多原因显得并不适合。看起来同样的模式正在互联网上重演——只是其历史会更加快速。

17.2　电话盗打

对通信服务的辱骂已逾百年。在邮票发明以前的日子里，邮件是由收信方付费。主动提供的信件变成了一个大问题（尤其是对名人而言），因此收信方被允许查看信件，如果他们不希望付费的话，也有权拒收。不久人们就想出办法，在信封上寄发短消息，而对方看完之后就可以拒收信件，而且想停止这类事件的规章压根儿没有真正起过作用［594］。早期的光学电报利用信号灯或回光仪工作，被提前下比赛赌注的人们所辱骂；这里，解决这一问题的尝试也终归失败了［729］。

电话与之相比也没有任何区别。

17.2.1　对仪表的攻击

早期的计量系统被人们广泛公开辱骂。

- 在 20 世纪 50 年代，有些系统中的接线员不得不靠听硬币掉落在金属平台上的声音以辨别电话亭中的顾客是否付费，因此一些人就用一片金属制成的大小合适的东西击打硬币盒盗打电话。
- 最初，电话接线员没有办法知道电话是从哪儿打来的，她就不得不询问打电话者使用的电话号码。打电话者可以告诉她一个另外的号码，那个电话的主人就必须掏钱了。如果在你自己家的电话这样做是比较危险的，但人们往往在公用电话亭里打。

当接线员开始打回电话证实那些国际电话的号码时，人们就实施社会工程攻击：（“这儿是 IBM；我们希望预定一个打往旧金山的电话，但因为时差，我们希望我们的管理主任今晚在家接听。他的电话号码是 xxx-yyyy”）。因而，在电话亭线路上加了一个特征用以警示接线员。但在英国实施时有一个缺陷：一个用电话亭打电话给接线员的顾客可能缩减剩余时间，留下约 1/4 秒，于是他被断开然后重接（通常是到另一个接线员），这时就没有了有关他是从一个公用电话亭拨打电话的信号了。那么他就可能打电话到任何地方而且按任何市话的标准付费。

- 这个系统也用一个或多个脉冲的方式来表现硬币投入，每个脉冲是由一个简短的开放电流在线路上插入的电阻组成。在许多学院，胆大的学生会放置一个“磁纽扣”模拟学生会的电话亭里的这一过程，因此他们可以免费打电话（这一案例中的账单都送到了学生会，因此磁纽扣并不是十分好玩的）。

有关计量原理的攻击依然在继续。许多国家已将他们的付费电话改为利用集成电路卡用以减少硬币搜集与故意破坏的代价。有些措施非常糟糕（正像我在防篡改章节中表述的那样），坏蛋们制造了大量的假电话卡。其他的攻击有所谓的*盗打电话*：对其他人的电话线进行物理攻击以盗用他们的服务。

在 20 世纪 70 年代，国际电话费还很贵，外国留学生把自己的电话线钳下，接到一个住宅人家的线上打电话回国；一个从未意识到这种盗打方式的电话主人将会收到一张大面值账单。在许多国家，尽管光缆铺设实际由电话公司负责，公司要为房间里的插座提供检修，可电话公司非常坚决地认为户主应该付这些账，而且威胁说如果不付就把他们列入黑名单。现在，长途电话便宜了，盗打电话的金融刺激大大降低了。但至今仍有很多问题，于是挪威电话公司设计了一个系统，在提供电话服务之前需要在墙上安置的插座里的认证设备与电话局的交换软件之间进行查询口令与回复的交流 [426]。

盗打电话给英格兰东北 Cramlington 小镇上的一户人家造成了灾难性的影响。第一个盗打征兆就是他们在住处的电话中听到有人交谈。第二个盗打征兆就是警察局的传唤，听说有人抱怨说他们妨碍了接听电话，抱怨者是三名妇女，她们的电话都是只有一位数字与他们家的不同，所以警方就怀疑是因为他们家的人影响的。当这家人查阅电话账单时，发现了打向额外收费电话的一串串清单；这些都是差不多在那个妨碍电话前突然打出的。后来，这个家庭向电话公司抱怨出了故障，他们的线路得以重新安排，才解决了这一问题。

电话公司否认存在电话分接头的可能，尽管他们的维修人员注意到这家人的电话线在分配盒里被篡改了，然后以此提交了维修报告（电话公司后来声明报告有错）。后来人们发现一个毒品贩卖者就住在附近，看起来合理的情况是他盗接了这家人的电话线，以便用受害者的电话线序列号打额外收费电话。但警察局与地方电话公司都拒绝进入这个嫌疑人的家，声称那是很危险的举动——即使这个嫌疑人现在已被判了六年监狱生活，挪威电话公司拒绝为被告方调查这起盗打电话事件。结局是被害者被判刑，原因是使用 harrassing 电话，实际上人们普遍都认为这是法庭误判。下面还将继续讨论电话公司与警察局之类的社会团体的私下交易，是出于否认盗打电话存在可能性的策略考虑，以掩盖监视行动——或者一些更加邪恶的事情。

就同一话题而言，用无线电话进行盗打是另一种盗打衍生技术。在 20 世纪 90 年代的前几年里，这在法国巴黎变得十分流行，因此法国电信打破电话公司传统，并声称这类事件正

在发生，声称正在利用非法进口的无线电话的受害者们极易受骗 [475]。直到今天我依然不知道任何无线电话（被批准的还是未批准的）与正规空中传输线路之间的区别。新的数字无线电话使用 DECT 标准，考虑到了查询口令 - 回复机制 [769]；但迄今为止其终端站点好像仅仅是将他们的名字发送到基站。

社会工程是另一个广泛传播的计谋。一个窃贼假装是从 AT&T 安全部给你打来电话，问你是否用你的电话卡打了很多到秘鲁的电话。当你否定这一问题，她就会说这个电话显然是恶作剧，但为了能够让对方付费，她会进一步确定你的卡号是否是 123-456-7890-6543？你一定会说不是的（如果你还没有真正警觉的话），你会说是 123-456-7890-5678。因为 123-456-7890 是你的电话号码，5678 是你的密码，这就意味着你已授权那个人可以给你打对方付费电话了。

额外收费电话的出现也导致电话欺诈者设计了各种各样的诡计使得人们给他们打电话：寻呼信息、工作广告、有关亲戚的紧急消息，以 900 打头号码的“低成本”电话卡——你可以自己对之命名。加勒比海的区号 809 以前总是欺骗美国电话用户的一个著名的幌子；最近，那里引入了新的区号，像大鳄鱼岛的区号就是 345，使之更难于发现那些额外收费骗子的电话号码。电话公司的忠告是“不要给陌生的电话号码回电话”[13]。但该怎样实施呢？

17.2.2　信号攻击

盗打电话这个词其实指的就是信号攻击，也指纯粹的长途电话费欺骗。直到 20 世纪 80 年代，电话公司一直使用信号系统，通过在携带有通话流量的相同电路上发送语音脉冲形成内波段。我听说的第一次攻击要追溯到 1952 年，到了 20 世纪 60 年代中后期，美国与英国的盗打电话者想出了许多重新接通电话的方法。典型的盗打方式是用自制的语音发生器，通常被称作蓝盒子。这种盗打先打一个 800 号码，然后发送一个语音传给远端的线路——也就是说，断开通话的对方，让打电话者在线，同时用一根干线接通到电话局。此时这个打电话者就可以打向真正需要的那个电话，而这个电话是免费接通的。众所周知，Steve Jobs 与 Steve Wozniak 在多样化电脑系统之前就建立了蓝盒子 [319]。

盗打电话的源头就是人们爱占便宜的念头。在过去的日子里，许多电话公司都是垄断经营。它们规模庞大，不露面也从不回应顾客。那些在设备盗窃中被电话盗打者分接的家用电话客户发现他们与不必要的电话账单联系到了一起。如果一个年轻人控告你女儿（你根本就不知道）盗打电话，说她没有付那些他打给她的电话费，你立即会发现电话公司在无休止地索取他的名字或支付的款项。电话公司也与国家结盟。在许多国家，有许多信号代码或转换特性可以让警察从警察局里舒适地窃听你的电话，而不用派一个线路员去你家里安置线路插头。如果回到越南和学生抗议的时代，这都是煽动性的材料。盗打电话者都成了反文化主流的英雄，而电话公司与黑社会却紧密结盟。

只要电话信号是用内波段脉冲携带就无法停止蓝盒子之类的攻击，电话公司花了多年的时间与无数的美元用来提升交换机让信号以外波段方式传送，并使用相互独立的通道，这样电话用户就不能轻易进入了。渐渐地，一个地区接一个地区，这个世界就向蓝盒子攻击关上了大门，尽管还有少数地方仍然存在。举个例子，第一例美国空军的军事行动被非战斗人员的信息战攻击所破坏的事件发生在 1994 年，当时两名英国黑客通过阿根廷的旧电话系统经由模拟传输线路进入了罗马空军基地。过去这一手段用于有效地拦截调查员 [722]。但要击

溃一个现代的电话网络系统，必须采取不同的技术手段。

17.2.3 攻击交换机与配置

第二种盗打攻击把目标锁定到了那些用于转换的电脑系统。典型的系统就是电话局里局域网上的 UNIX 机器，也是具备维修安排之类管理功能的计算机。通过潜入这些防范功能少的机器，一个盗打电话者就可以通过局域网进入交换装置——或者进入诸如电话客户数据库之类的附属系统。想要仔细查看有关 PacBell 在这方面的调查，请看［167］；如果想看 Bellcore 的，则请看［462］。

利用这些技术，也可以找到未公开的电话号码，即使没有电话用户需要的知识，电话也可以照打无误，而且各种各样的调皮捣蛋也都成为可能。加利福尼亚一个名叫 Kevin Poulsen 的盗打电话者，在 1985～1988 年间作为根用户进入了许多 PacBell 的交换机以及其他系统；很明显，他像黑客一样进行了许多盗窃行为（他最终被判阴谋掌握了 15 个或更多的伪造的、未被授权的偷来的访问设备）。他还做了一些危害较轻的事情。像盗取未公开的电话号码给著名人士打电话，从洛杉矶广播电台 KIIS-FM 赢得一辆保时捷汽车（每周，KIIS 会送给第 102 个打入电话的听众一辆保时捷汽车，Kevin 与他的同伙拦截打向电台的 25 条电话线的所有电话，并据为己有，打了第 102 个电话，因此得到了这辆保时捷汽车）。Poulsen 还被指控非法窃听与刺探；但这些指控都被驳回。实际上，联邦调查局已经重重地惩罚了他，以至于产生了许多有关当局与电话公司发生不正当冲突的言论，比如“你在需要的时候就窃听盗取我们的东西，那我们也要调查你的非法侵入问题”［294］。

尽管有关 Poulsen 未授权的窃听指控被驳回了，但联邦调查局敏锐地把注意力集中到了外国情报机构通过遥控窃听对美国电话公司计算机进行攻击的可能性上。［167］文中提到的一些攻击是从大洋彼岸发起的，而且这种方式在信息战攻击的背景下用于破坏整个电话系统的可能性已成为几年来国家安全局关注的问题［321，480］。当然，也有处事谨慎的国家设想他们的电话交换设备容易被制造国政府所攻击。

但是尽管高科技攻击时有发生——而且报纸上关于盗打电话的文章倾向于大肆渲染“邪恶的非法进入”这一方面——许多真正的攻击却是相当简单的。许多还涉及了内部人员，他们故意使系统误配置，通过优惠电话号码提供免费电话。因为电话公司的边际成本几乎为零，即多对一个额外的电话提供服务其成本趋近于零，这种事情倒没有什么大不了，但随着现代化高附加值设备的激增，能够进入系统的人就会有动力去投放（或伪造）大量的电话到同伙的黄色服务线路上。由于移动电话的出现，控制降低也使得欺诈更为严重，因为他们与电话公司之间增加了现金支付［200］。内部人员也可能会玩弄诡计，利用那些依赖电话网络安全的设备。在一次使人们联想起 Poulsen 的非法侵入行动中，两名英国电信的职员被解雇，因为他们每个人都从一个打入电话提供处赢得了 10 张 Concorde 的票，而这只有千分之一的概率［754］。

对于局外人而言，Kevin Mitnick 被称作“黑客之王”。当他被捕并判以一系列非法闯入罪名时众多通讯社对此进行了新闻报道，其非法闯入罪名也包括闯入电话系统，并使之成为联邦调查局逃犯追捕处的目标。在出狱之后他证实许多行为都包括了社会工程。他的审判大会的证词，也就是本章开头引用的话，巧妙地概括了这一问题［555］。对于粗心的内部工作人员来说，会使电话公司易受攻击，就像被恶意的内部人员攻击一样——这也非常像医疗系

统及其他许多我已谈到的系统。

17.2.4　不安全的终端系统

在对安装在电话公司房屋内的系统进行直接攻击之后，现代电话系统中的下一个主要漏洞是不安全的终端设备和特征干扰。

有大量关于不法之徒通过欺骗他人拨打一些额外收费电话从而利用人们的应答机的案例。这一问题是由电话公司的交换机引起的，在通话的对方挂断电话后，它还提供了 12 秒的拨号音。因此我可以录下你的应答机 13 秒空白音，后面跟上我想打过去的电话号码的语音，以及想发送的消息；然后我再拨，使电话播放其声音然后挂断。最近，发生了一例用于对付计算机的相似诡计——在你的电话账单上出现长达三个小时打向 Sierra Leone 的色情专线的电话很可能就是你 PC 机上的病毒做的怪。

但真正利用不安全终端系统进行的大欺骗，是把目标对准公司。把目标对准公司的专用小交换机系统（private branch exchange system，PBX）的欺诈在 20 世纪 90 年代中期之前一直是一个大问题，每年造成的商务损失难以衡量［202］。PBX 设备通常用作联传（refiling）呼叫，又名直接内部系统通路（direct inward system access，DISA）的设备提供。最典型的利用方式就是公司销售人员可以打入一个 800 打头的号码，输入 PIN 或密码，然后再打出去，从而可以利用大公司可能获得的相对低价位的长途电话时段优势。正像你期待的那样，这些 PIN 为大家所知，然后可以与那些不法分子做交易［564］。导致的结果众所周知，就是拨号接通欺诈。

在许多情况下，制造商将 PIN 设定为一个默认值，而客户自己从来不改变。在其他情况下，PIN 被那些监视旅馆电话通信以偷窃信用卡号的人所截获；电话卡号和 PBX PIN 容易被他人用于旁门左道。许多 PBX 设计都固定了那些允许远程维修进入的工程密码，聪明人都可以推测生产商为任何 PBX 至少安装了一个后门以便为法律强制部门和情报局提供方便通道（据说也是出口许可的条件）。当然这样的特征被发现并被滥用了。在一个案例中，伦敦警察厅的 PBX 遭到损害，并被那些不法分子用于联传（从一租用线路电话网的一站传送信息至一非租用线路电话网的另一站）电话，从而花掉了警察厅一百万英镑，因此他们控告了电话安装公司，而且这些不法分子也没有抓到［745］。这一事件是尤其伤人的，因为犯罪的动机之一仅是不用搭线就能进入通信获得窃听带来的刺激。

在大多数情况下，拨号接通欺诈是由额外收费设备驱动的，主要的刑事被告是那些与额外收费线路所有者合谋犯罪的欺骗者，次要的刑事被告是有组织的犯罪团伙，他们利用合法公司的电话线路 ID 隐藏自己的电话，例如从美国打到哥伦比亚，或者从英国打到巴基斯坦和中国——通常通过遭受损害的第三国的 PBX 以掩盖其通信（在这些不法分子从美国往外打电话时，看起来有点像在伦敦警察厅发生的案件）。许多公司还不理解看护它们的拨号音的需要，而且即使他们想这样做却不知道该怎么去做。PBX 通常是由那些对安全知之甚少的电信公司的管理人员来运行的，且安全经理通常对电话也不了解。

利用不安全的终端系统有时也会影响到国内的电话客户，由于许多人把计算机与电话连在一起。一个人尽皆知的案例是摩尔多瓦诡计。1997 年，一个色情站点的顾客被告知下载一个“浏览器”程序，这使得他们的电话线被切换并连到了摩尔多瓦的一个电话号码上（由于已关掉了调制解调器的扬声器，因此他们根本没有觉察到）。这个连接会持续一个通宵直

到他们关掉电脑。结果是这种盗打方式使数以千计的电话用户遭受了数不清的经济损失，因为国际长途可是超过了每分钟 2 美元的价格。他们的电话公司试图收取这笔钱，但在遭到强烈抗议后，电话用户拿回了自己的钱，联邦贸易委员会责令并告发了恶作剧肇事者 [284]。从那以后，出现了许多盲目模仿者；最近，AT&T 收到打往 Chad 的电话用户的抱怨，好像是爱尔兰的一家网络公司引起的 [543]。

额外收费的诡计和匿名电话并不是盗打惟一的动机。因为电话开始应用于投票、安全进入公寓建筑、控制对罪犯的有条件假释期以及使金融交易生效之类的任务，更多的盗打动机由此而生，同时也出现了更多创造性的非法分子，尤其是那些使呼叫者的线路 ID 失效的非法进入行动。更极端的例子发生在伦敦，一个骗子突然出现，他要用银行支票购买金条，于是金条销售员打电话到银行核实情况，销售员从电话另一端中得到了确认，他出售了金条。可最后支票被证明是伪造的，是那人的同伙分接了在街上的分配盒子里的银行电话线。

有时，这种攻击还被诚实的民众当作非常有用的方式来使用。最绝妙的例子来自 Udi Manber，情况如下。假设你购买了某东西但发现其中间出现了断裂，而生产商的帮助热线只有一台应答器。为了得到服务，你就不得不去除应答器上的服务。通常可以录下其信息，然后重放，就像顾客的通信一样。非常幸运，应答器的主人会以为它坏了，就会送去维修，这样你就可以打进帮助电话了。

17.2.5　特征干扰

越来越多的电话操纵包含了特征干扰。

- 华盛顿州 Clallam Bay 改造中心的劳教人员，他们只被允许打那些由对方付款的电话，从而使人们发现了由电话公司（Fone America）推荐的自动启动对方付费电话系统的的问题。系统会呼叫被拨号码，然后一个合成音会说："如果你接受一个来自（打电话者的名字）的对方付费电话，请在你的电话上按两次 3 号键。"罪犯们假定对着话筒说了他们的名字，记录下来并插进去。作为一个附加的特征，系统有将这些问候用西班牙语传送的功能。这些人按部就班；当被要求验明自己身份时便说"如果你想用英语听这一消息，请按 33。"这通常能起作用，他们就可以接到社团的 PBX 上，然后跟接线员说接到外线上。华盛顿大学就被这样的诡计袭击了好几次 [298]。
- 1996 年 11 月，英国电信公布了一种称为回铃的电话功能。如果你拨叫了一个正忙的电话号码，你就可以键入一个短码，一旦被叫方的电话闲下来了，你的电话与对方的电话铃声都会响起。不过这一电话要由你来付账。然而，在收费电话上使用时，这是由电话的主人来付，而不是打电话者。拥有私人收费电话的人，如客栈老板及店主，因此损失了许多钱，最后电话公司被迫归还了这些损失 [412]。
- 电话转接是许多电话盗打的根源。已有这样的案例，非法闯入者说服了电话公司的接线员将电话转接给其他本不想上色情线路的人，可怜的受害者便收到了额外收费的账单。
- 会议电话也带来许多麻烦。举个例子，有些国家的足球流氓被罚禁止自由行动，要求他们在球赛期间呆在家里。为了验证这一点，通常要用到检验设备，通过打电话者的 ID 来核实他们的电话号码。应对的计谋就是找一个小家伙用检验设备去开通一个会议电话，然后你带上移动电话到了赛场。如果检验官员问你怎么有人群喧闹声，

你就告诉他这是电视的声音，如果要关掉声音的话，你的配偶会十分不乐意（如果他想给你打过来，你让小家伙转接电话就行了）。

这就带给我们许多由移动电话造成的问题。

17.3　移动电话

从 20 世纪 80 年代初期开始，移动电话作为昂贵的奢侈品的日子一去不复返了，而是成了大科技企业成功传奇故事的典范，世界上年度销售额增长率达 30% ~ 50%。在有些国家，最有名的要数斯堪的纳维亚，许多人都至少拥有一部移动电话，且还有许多新的电子设备附于其上。举个例子，有一种机器，当你用电话拨打在前面显示的数字时，就发给你一听苏打水——饮料钱自然就加到你的电话费里了。这种增长在发展中国家特别快，在那里有线电话常被损坏，人们往往要等上几年时间电话服务部门才去重装。

尽管由于电话费用的原因，大多数人很节俭地使用移动电话（大多数情况下，长时间的通话都是发生在有线电话之间），不过，仍然有罪犯拼命地使用移动电话。因此，在英国，现在有超过一半的警员的窃听电话是针对移动号码进行的。

因此移动电话对安全工程而言非常地重要，一方面它是作为基本基础设施的一部分，另一方面是作为服务传送的渠道。它们也能给我们提供许多有关欺诈技术与应对技术措施的经验教训。

17.3.1　移动电话复制

第一代移动电话用的是模拟信号，没有真正的可靠性。信号收集装置把其序列号清楚地传到空中传输线路（在美国的系统中有两个线路：一个是为设备服务的，另一个是为电话用户准备的）。因此不法分子自己就能建造设备截获邻居家打出的电话（我甚至还亲眼见过一个电话被一个简单的黑客软件重新进行编制）。主要的不法分子之一是电话设备销售员，他们偷来电话设备然后贱卖，顾客通常是那些想向家里打电话的移民或学生。这些电话设备销售员通常在为人熟知的摊位挂出那些复制的移动电话，他们的顾客就会排队花上几美元打电话回家。

有关电话序列号的黑市发展起来了，企业的工程师们造出制栓（tumbler），用于为每一次呼叫提供一个不同标识的移动电话来对付它。制栓设计得使警方都难于跟踪 [406]。对序列号的需求变得如此之大，令人满意的难度也越来越大，即使在像机场这种许多移动电话都开着的地方也能探听一样。与被动探听一样，主动的方式开始使用起来。

现代的移动电话是蜂窝状的，接线员将服务区划分到每一个蜂房，每一个蜂房由一个基站所覆盖。移动电话就是使用那些有着最强信号的基站，当客户漫游时，就有像一个蜂房到另一个蜂房转换时提供的诸如“勿碰”之类的协议（要细看移动电话技术，参阅 [636]）。主动攻击包括了假基站，最典型的就是在诸如快车道桥之类的有许多车辆过往的地方，当电话经过时，他们听到更强的基站信号并试图通过发送序列号进行注册。

有许多手段用于尝试减少大量存在的欺诈。许多接线员拥有入侵检测系统，监视着那些值得怀疑的行为模式，比如每小时从纽约和洛杉矶彼此打出的电话，或者电话数量快速增长的情况。大量前瞻性研究取得了进展。举个例子，一个真正的移动电话一般很有规律地进行漫游和打电话回家，可如果突然停止不打回家了，则通常就是被盗了。

在关于电子战的章节中，我提到了射频“指纹”，是从前的一项机密军事技术，其通过从一个手持机到另一个手持机变化的信号特征识别某一具体的设备，然后将其与向外宣称的序列号联系起来［341］。尽管这一技术也起作用——被英国的 Vodafone 用于排除类似的移动电话近距离蜂房欺骗——但是太昂贵，因为涉及到修改基站（Vodafone 也用了入侵检测系统跟踪用户的电话模式与机动性，这在［769］中有详细的描述；其竞争对手，蜂窝网络（Cellnet），很简单地就拦截了从模拟移动电话打来的国际电话——这就帮助将其高端客户转到了更现代化的数字网络系统上）。另一个提出的解决办法是采纳密码认证协议，但在不更换整个网络的情况下其效果受到了限制，比如说，一个人可用查询口令—回复协议来修改序列号［305］。但是提出的加强模拟移动电话安全的许多原理都被证明是脆弱的［780］。

最终，这一行业认为有必要重新设计整个系统，不只是使之更安全，而是要支持在不需新手持机的情况下从一个国家到另一个国家的漫游能力，还有就是发送与接收短消息的能力之类的新特点（在欧洲尤为重要，因为许多小国家挤在一块儿）。

17.3.2　GSM 系统结构

第二代移动电话用的是数字技术。截止到 2000 年，世界范围内的绝大部分手机用的都是全球移动通信系统（Global System for Mobile Communications，GSM），其设计源于推进国际漫游的出发点，并于 1992 年开始投放市场。美国、日本和以色列都有不同的数字标准（尽管在部分美洲国家存在竞争的 GSM 服务）。

GSM 的设计者企图使该系统免遭复制与其他的攻击；目标就是至少具备它准备替代的有线系统的安全程度。他们都做了什么，以及如何成功的，什么地方又失败了，都是有意义的案例记录。

17.3.3　通信安全机制

GSM 中用到的认证协议在许多地方都有描述，像［141］（该书也描述了一个不兼容的美国系统的机制）。但业界试图保守密码和其他一些保护机制的秘密，因为它们构成了 GSM 安全系统的核心。但这也不起作用；最终有些信息被泄露了，剩下的全被反向工程学发现了。我在这里简单地描述一下，详细的内容可以在［713］这类站点上找到。

每一个网络系统有两个数据库，一个叫本地位置注册（home location register，HLR），包含了自己的移动电话的位置；另一个叫来宾位置注册（visitor location register，VLR），是从其他网络系统漫游到此地的位置。这些数据库使得即将来临的电话可以到达正确的蜂房；请看图 17-1，以获得总的看法。

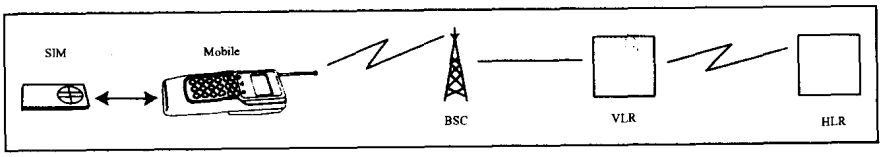


图 17-1　GSM 认证系统构成

手机已属日用必备品。它使用电话用户身份模块（subscriber identity module，SIM）实现个人化，这是当你注册加入一个网络系统服务时得到的智能卡，要将其置入你的手机中。

SIM 可看作包含了三个号码：

1）它有一个个人识别号（personal identification number，PIN），用它可以打开卡锁。从理论上讲，这可以防止其他人使用偷来的移动电话。在实际上，许多网络系统设定的初始 PIN 为 0000，而许多用户对之从不修改。

2）它有一个国际移动电话用户身份识别（IMSI），是惟一与你的移动电话号码相匹配的号码。

3）最后，还有一个电话用户认证密钥 K_i，是一个 128 位的数字，用于认证 IMSI，且为你的本地网络系统所知。

不像银行，因为银行都是使用主密钥生成 PIN，电话公司认为主密钥太危险。与使用主密钥不同，*KM* 出现了，由 $K_i = \{\text{IMSI}\}_{KM}$ 生成认证密钥，该密钥随机产生并保留在与 *HLR* 相连的认证数据库中。

用于认证手机与网络系统之间的协议如下运行。当你一开机，*SIM* 就要求顾客的 *PIN*，一旦正确输入，手机就发出 *IMSI* 并将之发送到最近的基站。这就送到了用户的 *HLR*，随即产生五个三元组问题。每个三元组包含如下信息：

- RAND　一个随机查询口令
- SRES　一个回复
- K_C　一个加密密钥

这些值之间的关系就是：在 SIM 中的认证密钥 K_i 下加密，RAND 提供一个与 K_C 联系在一起的 SRES 输出：

$$\{RAND\}_{k_i} = (\text{SRES} | \text{K}_\text{C})$$

这种加密的标准方法是利用名叫 Comp128 的单向函数，又叫 A3/A8（A3 是指 SRES 的输出，A8 指 *Kc* 的输出）（这是一个哈希函数，要运算 40 轮，具体的描述见 [138]）。其基本的思路类似图 5-9 中所描述的那样：每一轮都包含有混合以及表格检查。有五个表格，每一个分别有 512，256，128，64 和 32 字节条目，这个哈希函数在五轮中的每一块都成功地用到了它们；有八个这样的块。

乍看起来，这是一个非常复杂的哈希函数，好像不可能找出输出哈希值的原图像。然而一旦其设计最终被泄露，其脆弱性就为人所知了。四个字节——i、i + 8、i + 16 与 i + 24，在第二轮的输出只与输入的同位的字节有关。其中的两个输入字节（i 与 i + 8）是密钥字节，因而对任何给定的 SIM 卡都是固定的，而剩下的两个输入字节来自于查询口令的输入。

四字节对四字节的通道叫有限输送管，有可能通过改变来自于查询口令的两个输入字节来推测。既然这些轮次是无法令人相信的，你可以期望两轮之后会发生碰撞，且生日定理确保碰撞发生得很快（因为输送管只有四个字节宽）。一旦所有的细节被算出来，结果是你需要大约 150 000 个合适的选择查询口令才能推断出这一密钥 [781，783]。因此是如果能够访问使用 Comp128 的网络发布的 SIM 卡，认证密钥利用网络上可获得的软件在几个小时内就可以提取出来。差不多所有的网络系统都使用 Comp128。因此用移动电话租借汽车的人，可以利用他的笔记本电脑和一个合适的适配器花上一晚时间复制出一张 SIM 卡；假如一个人要卖给你一部 GSM 移动电话，可能就是对认证密钥进行了“备用拷贝”，以后他就可以把电话账单打到你的户头上。

这一攻击也是有关利用由一小组人评估的秘密密码原语带来危险的例子。用于发现瑕疵的必要的密码分析技术是众所周知的［773］，如果 Comp128 公开接受公众监视，瑕疵就有可能被发现。

至少三元组被发送到了基站，现在它向移动用户提供第一个 RAND。并被传到 SIM，用于计算 SRES。移动电话将之返回给基站；如果是正确的，移动电话与基站就可以使用加密的密钥 K_C 进行通信，整个的认证协议看起来就像图 17-2 中所示的那样。

SIM → HLR	IMSI
HLR → BSC	(RAND, SRES, K_c), ...
BSC → SIM	RAND
SIM → BSC	SRES
BSC → mobile	$\{traffic\}_{K_c}$

图 17-2　GSM 认证协议

在这一协议中也存在易攻击性。在绝大多数国家，在基站与 VLR 之间的通信都是在未加密的微波传输线路上传播的（他们趋向于选择当地有线网络系统，因为当地电话公司通常是其竞争对手，即使并不重要，微波使安装既简单又快捷）。因此攻击者可以发射出一个他精选的 IMSI，然后截取微波通信线路上的三元组传回当地基站。一个德国移动电话接线员，曾悬赏 100 000 德国马克给任何可以将电话账单划到 SIM 卡在律师事务所的移动电话号码上面的人，当我们向他索要 IMSI 时他便声明取消该悬赏［30］。

三元组也可被重置。当你漫游在境外时，不道德的外国网络系统可获得你的五个三元组，然后重复使用，允许你随意拨打电话。这也就意味着网络系统不用提交回你的网络系统以得到更进一步的确认（即使他们这样做了，也保护不了你，因为被访问的网络系统可能在一周或更长的时间内还不会呈报其账单）。因此当你漫游时，你本地的网络系统可能无法切断你的电话（这依赖于电话公司之间合约的条款），因此他们可能仍然有义务向漫游地网络系统支付一笔电话费用。这就意味着即使你认为利用提前付费的 SIM 卡已经限制了你的付费责任，但却仍然以你原地的网络系统尽力从你这儿收取话费而告终。这就是为什么在你成功使用提前付费的 SIM 卡漫游前，通常被要求提供你的信用卡号，以便支付比你预想的还要多的钱才算完事。

我这里还没有关于由外部人员（也就是非电话公司职员的攻击者）利用这些技术进行的欺诈行为的报道。当 GSM 被引入后，那些不法分子简单地从他们一贯的盗打方式转向使用偷盗的信用卡、偷来的身份证购买电话，或者向内部人员行贿［807］。抢劫案也触目惊心，在 Lewisham 的伦敦自治地区，偷盗移动电话的案件占到了街头抢劫案的 30% ~ 35%，其中 35% 的受害者是 18 岁以下的男性［501］。

大约从 1997 年起，预付费移动电话开始投放市场，许多犯罪分子立即对其加以利用。在许多欧洲国家，预付费移动电话可以花不到 100 美元买到，其中还包含了大概可以适度利用三个月的广播时间。预付费移动电话的方式很好，这不只是对躲避警方接线窃听而言，对盯梢、敲诈与其他形形色色的犯罪而言也是。预付费电话也易于被各式各样的简单欺诈所袭击。举个例子，如果在你购买时没有进行身份核对，对你而言就存在了一定的风险，因为可以用一个偷来的信用卡号启用漫游［214］。

除了认证之外，人们期望 GSM 系统能提供两种附加的保护：位置安全和内容保密。

位置安全机制就是指一旦移动电话注册入网，就会发放一个临时移动电话用户识别（temporary mobile subscriber identification，TMSI），其在漫游通过网络系统时充当了位置角色。

对这一机制的攻击用到一个名叫 IMSI 捕捉器（IMSI-catcher）的设备，本来它是供警察人员使用的［308］。这个 IMSI 捕捉器可以在监视可疑分子的警车上进行操作，相当于一个 GSM 基站。由于其比真正的基站要近得多，所以信号很强，移动电话就会尽力向它进行注册。IMSI 捕捉器声称不理解 TMSI，因此手机就向其发送明文 IMSI（如果用户希望从一个网络系统漫游到另一个系统时不想停止电话，且在 VLR 中从失败状态恢复回来，这一特征还是必要的［769］）。此时警方得到授权，截获打往那一移动电话的通信，或者——如果他们忙着呢——就可以进行一个中间人攻击，他们假装是发向移动电话的网络和通向网络的移动电话。

GSM 系统也被人们期望能够提供电话内容保密，一旦认证与注册完成，就对手机与基站之间的通信进行加密。通话被数字化、压缩然后切成信息包，每一小包都通过异或由加密的密钥 K_C 生成的伪随机序号与包序号来加密。在欧洲其通用算法是 A5。

像 Comp128 一样，当初 A5 也是秘密；不过一旦被泄露，攻击就出现了，这点也与 Comp128 相同。这一算法如图 17-3 所示。它有三个长度分别为 19、22 和 23 的线性反馈移位寄存器；其输出进行异或从而形成输出密钥流。生成器中的非线性来自多数占优计时方案，因此比较三个移位寄存器的中间位 c_i，记录中间字节相同的两个或三个移位寄存器。

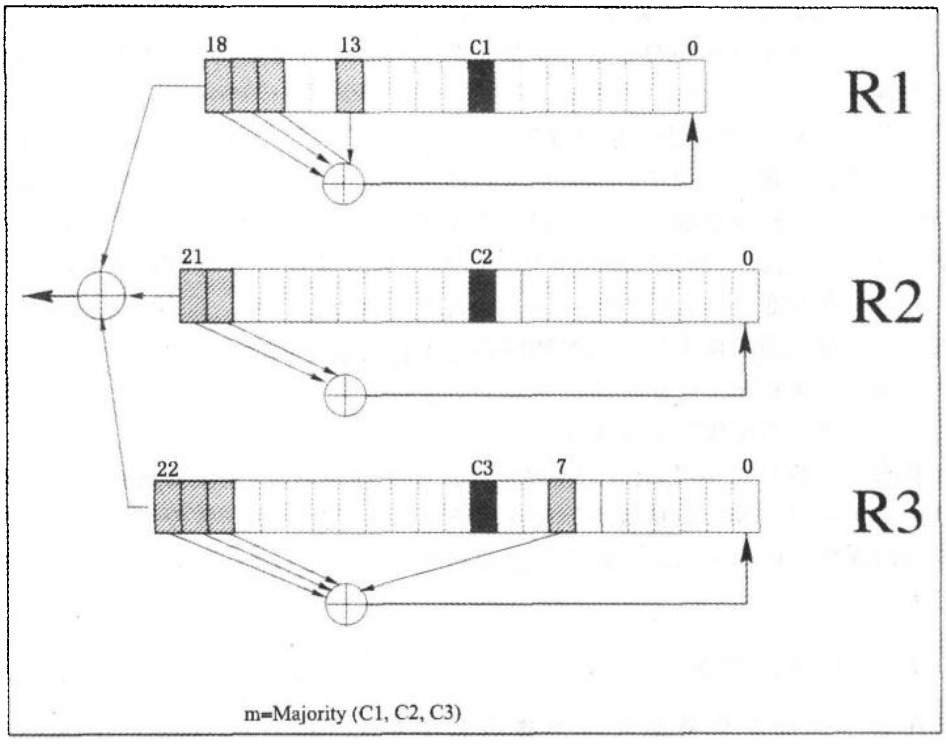


图 17-3　A5（Alex Biryukov 与 Adi Shamir 友情提供）

对这一排列的最明显的攻击是猜测两个相对短的寄存器，然后算出第三个寄存器的值。因为有 41 位需要猜测，你可以想像平均需要计算大约 2^{40} 次。但实际上比这还稍稍复杂一点，因为生成器丢失了状态；而许多状态可能不只一种前兆，因此更多的猜测也是必要的。据说 Alex Biryukov 与 Adi Shamir 通过将大量适合的最优化放在一起发现 A5 用并不太多的尝试

就可破解。其基本思路是计算大量有关算法状态收敛的那些特殊点的文件，然后用观察到的通信来寻找匹配物。一旦给定预计算的文件，这一攻击就可以用几秒钟的通信与几分钟的PC机上的计算，或者几分钟的通信与几秒钟的计算来完成［104］。

反向工程学现行系统也表明A5的密钥是蓄意削弱了的。尽管从理论上讲，A5有一个64位的密钥（移位寄存器里最初的负荷），实际操作中至少把10个显著的密钥位定为零。更有甚者，许多国家出售给电话公司的电话所带的是A5的更进一步的消弱衍生算法，我们称之为A5/2（在澳大利亚当人们意识到所用的是A5/2系统时曾引起了政治骚动）。

算法提供者们对此还振振有辞——安全机制在情报机构的紧急指示下进行蓄意的削弱，是为了易于窃听移动电话。真相总是更加阴险呀！

情报机构从他们自己国家的网络系统与分站（像大使馆）可以窃听的适当的微波通信线路的国家那里获得进入明文的通道。甚至在本国，无论与电话公司达成协议与否，窃听都可能发生；这些从微波通信线路上截取信息的设备十分广泛。但在绝大多数国家，警方会得到法律许可，允许他们直接进入电话公司系统，因为这样他们可以获得更多的信息——像本地注册数据，可以让他们对嫌疑犯过去的行动进行跟踪。在1997年，当通讯社得知电话公司向警方提供定位数据时，瑞士曾发生了一场大动荡［618］。在美国，通信委员会要求移动电话公司能够定位相关人物，“因此911电话可以迅速转接到正确的地方。”这就强加给了每一个移动电话用户，而不是让用户自己决定是否购买移动电话定位设备。美国的隐私维护者们当前正在与通信委员会就此事进行诉讼。

毋庸置疑，在GSM的设计中存在着当局的干涉影响，但从削弱完整性与保密性机制中得到的利益来看，好像仅局限于战术通信情报的情况。考虑在电子战章节中提到的案例，当时新西兰海军派了一艘驱逐舰去监视斐济的军事政变。即使斐济的电话公司被允许使用A5而不是A5/2系统，这也丝毫无法阻挠其任务，因为信号情报官员可以从微波通信线路上截获那些三元组，然后非法闯入其本地注册，或者干些其他的事情。即使所有这些都失败了，密钥还有可能被强硬的军队发现。但要能够快速中止通信才是最方便的——尤其是当你被派遣去做一个国外的维持和平使团成员，而你们的情报人员又从未对该国的当地电话公司中安置的窃听装置（或人员配置）加以重视。

这样看来，原始的GSM安全机制提供的保护比允许使用A5系统的国家的有线网络系统稍好一点，不过在其他地方就稍微不如有线网络系统了。这一通信安全机制的易受攻击性既不会把发达国家的电话用户暴露给许多附加的窃听系统，也不能阻止带给他们莫大不幸的那些欺诈行为。

17.3.4 下一代产品：3gpp

第三代数字移动电话最初称通用移动电信系统（Universal Mobile Telecommunications System，UMTS），但现在称之为第三代协作项目（Third-Generation Partnership Project，3gpp）。其安全性非常像GSM，但解决了许多为人所知的GSM系统的脆弱性。人们期望该系统在2003～2004年能够上市，一些设计细节还处于研制之中，因此本节从某种程度上来讲必然是暂时性的。然而，总体安全策略在［786］文中已有了描述，有关安全结构的最新协议也在［775］中罗列出来了。

密码算法A5与Comp128已被一个名叫Kasumi的分组密码所取代［442］。这是公开的，

且是基于 Mitsuru Matsui 的一个名叫 Misty 的设计，经受了长达好几年的公众监视 [527]。现在所有的密钥都是 128 位。密码用于保护消息内容与信号数据两者的完整性和保密性，而不是仅仅保证内容保密，尽管 3gpp 的第一阶段保护还达不到端到端的水平。在基站之间对传输三元组不受阻碍的传输实践会停止，就像劣等基站的易受攻击性一样。因此 IMSI 捕捉器对第三代移动通信将不起任何作用。作为替换，将会出现一个被法律所允许的窃听的合适的工程界面 [776]。最初，人们只期望它能够提供纯文本；而现在，有关关键材料的提供也将作为一个国家的选择权而获得。

在基本的 3gpp 协议中，基站控制者将认证推回到了来宾位置注册中。本地位置注册现在就称做本地环境（home environment，HE），SIM 就叫做 *UMTS SIM*（USIM）。像以前一样，本地环境选择一个随机查询口令 *RAND*，然后用 USIM 认证密钥 K 对之加密，生成一个回复 RES，一个保密性密钥 *CK*，一个完整性密钥 *IK* 以及一个匿名密钥 *AK*。

$\{RAND\}_K = (RES \mid CK \mid IK \mid AK)$

也有一个序列号 *SEQ* 为 HE 和 USIM 所知。在 RAND 与 SEQ 上计算 MAC，然后序列号被用匿名密钥进行的异或操作所掩盖。查询口令、预期回复、保密性密钥、完整性密钥与掩盖的序列号被合理组织，形成认证向量（authentication vector，*AV*），从 HE 发送到 VLR。然后 VLR 将查询口令、被掩盖的序列号和 MAC 发送给 USIM；USIM 计算出回复和密钥，对序列号进行解码，证实 MAC。如果这些都正确，则将回复返传给 VLR（见图 17-4）。

USIM → HE	IMSI (this can optionally be encrypted)
HE → VLR	RAND, XRES, CK, IK, $SEQ \oplus AK$, MAC
VLR → USIM	RAND, $SEQ \oplus AK$, MAC
USIM → VLR	RES

图 17-4　3gpp 认证协议

3gpp 有许多其他的特征，包括序列号生成、识别与定位隐私机制的细节，还有与 GSM 的反向兼容性，从 HE 到 VLR 传输中的认证向量的公开密钥加密机制，还有不同的可选密码机制之间的协商。在写作本书时许多还没有进行定义。

在 3gpp 的第一阶段，保密性将比在 GSM 中已经获得的得到更高质量的履行：在空中传输线路上进行的偷听将会像以前一样被阻止，而当前对骨干网的攻击，或者通过伪造的基站进行攻击则被排除在外。在 VLR 上，警方窃听仍然是可能的。在第二阶段，3gpp 有希望进行端到端加密，因此从一个手机到另一个手机的电话内容与某些相关的信号可以得到保护。这也导致了政府要求利用密钥契约协议——一个使得在必要的时候密钥可为警方与情报部门获得的协议。令人困扰的问题是：如果移动电话通话发生在下列情况——打电话者利用的是美国生产的手机，可他是英国电话公司的用户，目前漫游在法国，接电话的是手持芬兰制造的手机，正在瑞士漫游的德国电话公司用户，而这一通话经过了在加拿大的长距离服务设备，用到了瑞典的交换机，那么哪个国家的情报机构可以获得密钥的访问权呢（从传统意义上说，绝大多数国家都有办法通过这种或那种办法获得电话内容）。

被英国与法国（至少有这两个）当局所偏爱的解决办法是所谓的 Royal Holloway 协议 [418]，大部分是由 Vodafone 设计的。其将密钥访问权给了电话用户基于的国家（在这个例

子中，就是英国与德国)。这一协议获得了成功，得益于使用了 Diffie-Hellman 密钥交换的衍生技术，用户的私有密钥通过地方电话公司和/或情报机构所知的一个绝密主密钥对他们的名字加密而获得。尽管这一协议被英国民政部门和法国健康部门所采纳，但与电话公司安全哲学部门就主密钥是否是一件坏事仍然存在分歧。除了许多人从私架起来的偷听敲诈中感觉到的不舒服之外，这协议是无趣的、无能的［50］。与地方法令的强迫要求也存在着紧张的关系，在上面的例子中，目标正在漫游的两个国家（法国与瑞士）的警察部门也可能希望找到通道进入通信之中［776］。这一争论仍在继续，一个可能的解决办法就是 tromboning——一项已经建立的窃听技术，通信在其中被安排好了路线：出发前从交换机到监视设备，在接近之前就返回。然而，因特网系统拉管的引入明显延迟了调查对象的警戒。

因而，3gpp 在保密性方面并没有提供一场革命。就像 GSM 一样，其设计目标是安全性应该与有线网络系统兼容［390］，看起来好像也办得到。

账单机制的安全是一个很困难的问题。GSM 的账单机制不适用于 3gpp，因为有以下诸多原因：

- 电话细节录制（call detail record，CDR）在呼叫电话接通之后就出现了，这是一个为人认可的有线网络系统功能，但当环境变为移动电话时，这就成了一个严重的问题。攻击是指一个电话设备销售员利用偷来的移动电话建立一个长时间的会议呼叫，一个也能用偷来的信用卡漫游的预付费移动电话也可以（就像在 17.3.3 中讨论的那样)。其当事人一个接一个地链接又断开这一电话，且他可以通过话费通知设备知道要付多少钱。这一电话在挂断状态持续上几天，一旦它重新接通，一个需付几千美金的 CDR 产生了，警报也响了起来，因此他把电话扔到河里开始用下一个。截止 1996 年，情况变得不可收拾，以至于 Vodafone 引入了一个对任何移动电话而言都只有 6 小时限制时间的办法。
- 然而，6 个小时的通话时间后统统切断所有的 3gpp 电话是不可接受的。许多用户大多数时间需要持续用相关的轻包通信在因特网上联系（比如从他们的笔记本电脑)。
- 电话公司也希望对那些相对高价值的产品和服务传送收费，范围从目前的额外收费服务到基于定位的服务（“给我一张地图，指示我如何开车去最近的麦当劳”）以及多媒体服务[⊖]。顾客不但要向电话公司交费，还要支付给其他服务提供者，另外其收费还与电话的持续时间与数据流量有关，不过人们还要根据获得的服务质量支付费用，比如有效带宽。
- 最后，为了法律执行的目的，欧洲委员会打算要求所有的 3gpp 接线员将移动电话的定位信息至少保留一年。在电话用户账单记录上保留定位资料可能是最廉价的办法。

现存的 GSM 机制显然是不适合的——因为即使加上诸如实时国际清算之类的功能将是极其昂贵的。重新设计看来是必要的，具体的建议是重新设计 CDR，使其包含要求的数据数量、定位与服务质量信息，再建立一个在线的费用控制机制，用来限制每个用户身上发生的费用支付［558］。费用控制机制还没有进行标准化，但可以包括从当地网络系统或网关到

⊖ 假设，给定许多最先用于色情描述的新的通信服务，这将意味着生动的脱衣秀会从一个对下流行为的法律规定比较松散的国家发送并显示到你移动电话的屏幕上。因此许多假装正经的政府会要求得到有关 3gpp 的隐私机制以便核查其内容——就像音乐行业要想方设法阻止用户对用户的拷贝一样。我们会在第 20 章详细谈论这一问题。

本地环境转交的费用支付数据，这能够在可用信用卡耗尽（与预付费 SIM 卡一样）或用骗得的信用卡的情况发生时停止电话。

执行这一思路的建议办法是实现微支付机制［56］。这一思路是电话可以有规律地发送信用支付给每一个网络系统或服务提供商，由于他们需要电话支付。信用支付可以想像成电子硬币，可以进行防伪造密码保护。

在通话开始时，手机通过重复地迭代计算大量的电话赊欠：$t_1 = h(t_0)$，$t_2 = h(t_1)$，以此类推，其中 t_k（某些信用限制 k，典型的是 2^{10}单位）被电话公司签字。然后在通话过程中电话将定期地发布赊欠信息以便支付服务费用。这将由发出 t_k 开始，然后是 t_{k-1}，再是 t_{k-2}，以此类推。如果话费随后被提出质疑——无论是被电话用户还是网络系统接线员——声明存在大量的 j 赊欠的当事人必须提供 t_{k-j}和 t_k，还要附带上证明。由于哈希函数是单向的，除非手机实际上发出了许多赊欠信息，否则这将难于进行。现在，赊欠 t_{kj}可通过应用哈希函数迭代 j 次并证明结果是 t_k 来进行核对（这一协议是多人同时发现的一个例子，由我们在剑桥的小组、Pedersen、Rivest 与 Shamir 三方于 1995 年独立发明［26，605，648］）。

利用信用支付机制的一个优点是：它能从更多的方面保护用户，同时保护电话公司免遭会议电话的欺诈。电话用户至少在理论上能发现像 900 打头的每个电话收费高达 24 美元的那种号码，或伪装的普通号码，或两者兼备。

17.3.5　GSM 安全：成功或失败

GSM 安全到底是成功了还是失败了，这取决于你问了谁。

从密码学的观点来看，它是失败的，一旦将 Comp128 哈希函数和 A5 加密算法公之于众，它们就会被破解。事实上，GSM 经常被引用为 Kerckhoff 原理的一个目标教训——加密安全应该安排在对密码的选择中，而不是机制的模糊性。这一机制迟早会泄露，并且最好先让公众来评估考核，而不是在数以亿计的产品制造出来之后（对于大多数的密码分析者而言，GSM 安全并不是一个灾难，因为它提供了大量的写研究论文机会）。

从电话公司的角度来看，GSM 是成功的。GSM 运营商的股东，例如 Vodafone，他们已经赚取了巨额的利润，但只有一（小）部分的钱是因为 GSM 停止了对查询口令—回复机制的复制而得到的。密码的漏洞对于他们而言是毫不相干的，因为它们从来没有被利用过（至少在对电话收入做出重大伤害的方面没有过）。一项或两项欺诈一直在继续，比如长时间电话会议恶作剧，但是，总而言之，GSM 设计方案对电话公司是有好处的。

从罪犯的观点，GSM 也是很好的，因为不会妨碍他们盗取电话服务；他们一贯的盗打伎俩几乎没有发生变化；而成本转嫁到了信用卡公司以及那些身份证被盗或街头抢劫案的个人受害者。它也不能中止来自匿名电话的呼叫；预付电话行业的兴起使他们更加容易（电话公司也乐意看到这两种变化）。当然，GSM 对于直拨电话欺诈也无能为力。

从大国的情报机构的观点来看，GSM 也是好的。他们能以某种方法不受阻碍地进入当地的或国际的通信系统，且削弱后的 A5 版本也能推动那些针对发展中国家的战术通信情报。GSM 的第二代产品带来了一些有趣的功能，例如由接线员对手机进行遥控［636］。如果你能暗中破坏或假装成接线员，那么似乎就没有什么东西可以阻止你在不知道当事人任何信息的情况下打开其手机，并窃听房间里的谈话。

从警察局和低财力的情报机构的观点出发，事情不是如此的令人兴奋。问题并不是增加的 GSM 网络系统的技术复杂度：毕竟法庭允许的分接头可以留给电话公司（尽管当嫌疑犯是移动用户时发现窃听的电话很混乱）。问题是预付费移动电话的引入，这不仅降低了通信分析算法的信噪比，使得难以进行目标窃听，同时也鼓励了勒索和盯梢犯罪。

从顾客的观点出发，GSM 原本是以完全安全的功能而进行销售的，这准确吗？空中传输线路的加密确实能阻止偶尔的电话窃听。偷听，是具有类似电话的偶然的麻烦事（曾经有许多名人惹上麻烦的高姿态事件，其中英国有查尔斯王子在离婚前被人偷听到他与情妇间的谈话，在美国有涉及到 Newt Gingrich 的事件）。但是世界上几乎所有的搭线窃听电话都是大型的情报机构干的，对于他们来说，任何加密都没有多大的区别。

当我们把目光转向账单时，我们会发现对于用户来说，事情变得更加不肯定。手机的加密认证不能阻止额外费用经营者和电话公司进行的许多欺诈活动。如果这些公司做了什么手脚，那就很难逃脱伪造的费用，因为电话公司可以在法庭上说用户的智能卡和 PIN 只能用在打电话的手机上。同样的事情适用于未使用微支付的 3gpp 上。一个较小的补偿是 GSM 促进了预付费移动电话的广泛使用，它能限制事情的曝光。

所以为 GSM 而设计的安全特征，不会对用户有多大的帮助。它们是从电话公司的角度出发来设计提供“安全”：它们清除了大量的话费欺诈风险，却不会妨碍额外费用商业的运转——不管是真的还是假的。

从另外一个角度看，给用户带来一丝安慰的是 GSM 的脆弱性（长时间的会议电话）可能推动微付费体制的引入。但这也可能会带来副作用，即使额外费用欺诈变得更难。我说“可能”而不是“将”，因为看看电话公司是否会正确地执行这种体制是非常有趣的。这样，就会有商业上的动机来为用户提供一个监控他们支出的用户界面（因此当他们不能发现这些欺诈时，就可以责备他们），而实际上阻碍大多数的用户去真正地监控它（因此电话公司可以从额外费用欺诈收入中分到数亿元的钱）。下面即将对其进行介绍。

17.4　群体欺诈

群体欺诈问题尤其相关，因为在美国增长最快的欺诈是那些不择手段的电话公司向不知情的用户开出了大量的小面值账单。它以各种方式收集电话号码（例如，如果你拨叫一个 800 号码，那么你自己的号码会被传送到远处的终端，不管你是否想阻塞呼叫者的线路 ID）。这样，你就需要支付数美元的费用。你自己的电话公司传递这些费用，因此你会发现没有什么有效的方式可以对这件事情进行争执。有时候，诈骗利用的是一个法律的漏洞：如果你在美国拨打一个 800 电话号码，这家公司可能会说，“我们能马上打过来吗？”如果你同意的话，你被认为已经接受了一笔可能很高的费用。如果你随着通话的进展而回应声音的提示的话，同样的事情也可能会发生。这些行动就是所谓的*填鸭式策略*。

另外一个问题是*猛击*——即未经用户的同意就更改其长途电话服务的提供商。猛击者告诉你当地的电话公司你已经选择了他们的服务；然后你的电话公司将你的长途电话通过他们的服务进行发送；他们希望你不会注意到这个改变，并对账单提出异议；但是你的电话费却会大大增加。一些当地的电话公司，例如贝尔大西洋，允许客户冻结他们已经选择的长途电话运送商 [13]。

如果认为填鸭式策略和猛击方法仅是由小型的、可信任度不高的运营商做的，那就错

了。AT&T 在这方面是最坏的榜样之一，已经被罚款 30 万美元，不仅是因为它所实行的猛击行为，还因为它使用伪造的用户签名使用户看起来似乎已经同意转向他们的服务。当它伪造一对住在得克萨斯州已故的夫妇的签名时，被逮住了 [252]。

另外一个问题是可信度不高的电话公司。在美国，任何一个人都有权利建立一家电话公司。可以先直接建立一电话公司，然后从用户那儿收取现金。而一旦收到来自已经建成的公司的互连话费时，它就消失了。在英国，有一家公司利用普通的电话号码做色情热线的广告，并以此来诱惑警惕性不高的用户；然后这家公司按照用户的居住地址给用户送去巨额的账单，并试图胁迫用户付钱。在目前一个法院审判的类似案件中，他们依据非歧视规则进行辩护：如果英国通信公司能从色情热线中收取大笔的费用，为什么他们不可以呢？

不只是小的运营商沉迷于此类欺诈方式的商业行为。一个甚至影响到大型电话公司的例子是国际电话的短终端。

尽管额外收费号码常被用于或多或少的合法目的，例如软件支持，但许多都是利用了未成年人或自制力不强的人的缺点。所以管理者已迫使许多国家的电话公司对用户提供额外收费号码阻塞服务。但电话公司通过假装额外收费号码是国际号码来回避这个问题。本书曾在 17.2.1 节提到过具有加勒比海号码的诈骗案。现在，许多来自小国家、管理不严格的电话公司已经加入了这种行动，为色情热线提供一系列的电话号码，然后从中提成。

通常，打到小国家电话公司的电话并不会靠近其伪装的目的地。达到这种目的的技巧称做*短终端*，工作流程如下。通常，打往太平洋小岛国图瓦卢的电话要通过澳大利亚城市珀斯的通信卫星，在那儿通过卫星继续转发。然而，对色情热线号码进行标识在通信卫星系统中是非法的，因此它们自动由第二可选的运营商进行转发——新西兰的一家公司（女孩们——或者更准确一点，假装为年轻女孩的、年迈退休的娼妓们——实际上是在英国的曼彻斯特）。从技术上来说，这是用作攻击手段的后退机制的一个有趣事例。从法律上来说，这却很难对付，因为国际协议（内罗毕协议）禁止电话公司有选择性地阻塞国际目的地。这样，如果你想阻止你的孩子拨打图瓦卢的色情热线，你就必须阻塞所有的国际电话，这使你很难与德国的重要客户进行电话联系。

像这样的问题从根本上说是由于管理失败，但却变得越来越普遍（例如，在以上所提到的摩尔多瓦诈骗案中，电话并不是打到摩尔多瓦，而是通往加拿大 [151]）。当技术的发展使得许多新的、复杂的服务变成可能，而管理者又跟不上形势的变化时，这些问题就会变得更糟糕。

即使电话公司自己有时也会与增长的复杂度相冲突。当我写这本书的时候，有两个案件正处于法院审理中。在这些案子中，电话公司正在追查那些注意到以最好的折扣拨打国际额外费用号码时，所花费的钱要比运行额外收费服务少的人。他们宣称这些人已经获得的利润要比平常多两个数量级。电话公司声称这是欺诈；而辩护方认为这是诚实的套利行为。我们必须耐心等待，看看陪审团是怎么想的。

17.5　小结

电话欺诈是令人着迷的案例研究课题。人们欺骗电话公司已长达数十年之久，最近电话公司对这一行为进行了强有力的回击。从一开始，系统就根本起不到保护作用，很容易就能逃避电话费用或将电话改道。能够用来防止这种不轨行为的机制——波段外信号——由于系

统迅速增加的复杂度已经暴露出了更多的脆弱性而被证明是不恰当的。这些漏洞涉及到针对用户的社会工程攻击到 PBXes 之类终端设备的劣质设计和管理再到对各式各样难以预测功能的相互作用的利用。

总的来说，通信中的安全问题是环境变化的结果。环境变化之一就是违反规定，导致许多新电话公司如雨后春笋般涌出。然而，主要的改变是额外收费电话号码的引入，以前电话公司只出售具有可忽略边际成本的服务，突然之间，得到了真正的收益；并且，以前对通话系统进行伪造惟一得到的益处是警察很难监听到电话，突然间可以借此赚到大笔的钱。现在的保护机制不能应对这个发展。

日益增长的复杂度甚至使为了保证 GSM 数字移动系统安全而付出的相关的努力化成泡影。他们的工程师专注于通信安全威胁，而不是计算机安全威胁；他们还十分关心电话公司的利益，但是以顾客的牺牲为代价。下一代的移动服务——3gpp——看起来能够做得稍微好一些；但是我们必须耐心等待，看看它在实践中如何实施。

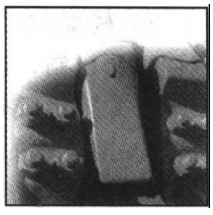
研究问题

对与电话欺诈和搭线窃听相关的问题在电话公司和情报机构实验室之外只做过相对较少的研究。然而，人们开始把兴趣越来越多地转移到协议、其他应用机制以及新型电信设备上了。最近发布的 3gpp 协议方案非常庞大且复杂，因而可能需要正式的方法与安全协议人员花费好长一段时间来进行充分的分析。下一代的增值服务一定会带来新的漏洞。所有这些通信和安全协议与用于分布式系统安全的机制之间的相互影响，无论是对感兴趣研究还是昂贵得令人咋舌的工程错误而言都提供了一个广阔的前景：有关如何使用系统工程技术来控制电信中存在的特征相互影响的定期讨论会已经出现。

参考资料

关于电话欺诈有很多内容分散的文章，但至今为止我还不知道哪篇文章涉及到了所有的内容。目前在美国广泛应用的关于欺诈技术的网站是 Alliance to Outfox Phone Fraud，这是一个行业联盟 [13]。有关电话欺诈基本技术的阐述有许多的参考书，例如涉及 GSM 的有 [636]，在像 [713] 这样的网站上也有很多。在 [400] 中有对 UMTS 的概述，在 [56] 中有“完全肯定（full Monty）”的内容。为了掌握电话欺诈的动态，有用的资源来自《Discount Long Distance Digest》[252]。

第 18 章　网络攻击与防御



那些认为可以通过加密技术解决他们问题的人们，其实并不真正了解他们的问题，而且也不了解加密技术本身。

——Roger Needham & Butler Lampson

18.1　引言

因特网安全问题是一个新兴的且变化很快的领域；那些可以占据报刊杂志头版头条的攻击手段也随时间的推移而发生着显著的变化。无论这些基于网络的攻击是否与你所从事的工作直接相关，它们都很有可能对你产生某些影响。即使你仅仅是利用黑客的故事来说服客户，使他们增加预算来处理那些严重威胁系统安全的事件，这种影响也不能避免。关键问题是，有关这一主题的某些知识对于从事安全工作的工程师们来讲至关重要。

目前存在着一些很流行的观念，认为网络通过加密技术和防火墙技术的保护可以变得安全可靠。其实反驳这些观点的最佳方法就是看一看目前最普通的网络攻击手段（当然，也有许多攻击可以取得和网络攻击一样的效果，但它们却是通过其他传统的方式表现出来的，而非网络。一个关于这方面的例子是来自英国首相办公室的令人难堪的电子邮件泄漏事件，最初怀疑是黑客所为，但这些电子邮件最后被一个名为 Benji the Binman 的私家侦探在首相家中电脑的个人民意调查垃圾箱里找到，而这位侦探因此立刻成名［520］）。

18.1.1　最普通的攻击手段

许多实际的攻击都涉及被攻击系统诸多漏洞或者缺陷的组合。这些漏洞包括我们在前些章节中看到的栈溢出攻击（你可以向程序传递一个过长的参数，而使程序不知不觉中执行了部分不应执行的代码段）和口令猜测，这两种系统漏洞都可以被因特网蠕虫所利用。常用的策略是首先获得目标网络中任何机器的某个账户，然后在这台机器上安装口令嗅探器来获取目标机器的账户，最后使用栈溢出的方法升级为 root 账户。

随着旧软件中各种错误的修正和新发布的软件中不断出现的新错误，存在于系统中的确切漏洞会随着时间的推移而不断变化。然而，攻击手段无非是那么几种模式，一些老的攻击方法仅仅是通过一种新的伪装手段就能重新出现。这里列举了截至 2000 年 6 月为止最流行的 10 大攻击手段［670］。

1）栈溢出攻击，攻击对象为许多 Unix 和 Linux 主机用来获取 DNS 服务的 BIND 程序，它可以立即获取账户访问权限。

2）使用易受攻击的 CGI 程序，攻击对象为 Web 服务器，这些程序常常作为软件供应商提供的例子程序使用，而没有被及时删除。CGI 程序错误常常被攻击者用作接管和丑化 Web 服务器的手段。

3）栈溢出攻击，攻击对象为许多 Unix 和 Linux 主机用来支持本地网络的 RPC（远程过程调用）机制，该攻击可以使入侵者立即获得账户访问权限（在 1999 年到 2000 年初常被用于分布式拒绝服务攻击中）。

4）对微软的 IIS（因特网信息服务器）Web 服务器软件中的漏洞进行攻击，它可以使入侵者立即通过管理员账户访问服务器。

5）对运行在 Unix 和 Linux 上的最常用的邮件程序 sendmail 中的漏洞进行攻击。这些年来，在 sendmail 程序中发现了许多漏洞，最早可以追溯到 1988 年 CERT 发布的指南性文档中。在最近发现的错误中，有一个错误可以使被攻击机器通过电子邮件的方式发送自身的口令文件到攻击者处，然后攻击者再对此进行破译。

6）栈溢出攻击，攻击目标为 Sun 公司的 Solaris 操作系统，它可以使入侵者立即获得 root 访问权限。

7）对于 NFS 的攻击，有关 NFS 的知识稍后会进行简要的描述。攻击也可以针对 Windows NT 和 Macintosh 操作系统中等价于 NFS 的类似功能。NFS 机制被用来在局域网上共享文件。

8）通过猜测用户名和口令进行攻击，尤其在系统 root 和 administrator 口令设置过于简单，或者系统本身提供默认口令，而用户由于嫌麻烦而没有修改口令的场合。

9）对于 IMAP 和 POP 协议的攻击，这两种协议允许远程访问电子邮件系统，但由于错误的配置也会导致入侵者非法访问系统。

10）对于 SNMP 协议中脆弱的认证机制的攻击，该协议被网络管理员用来管理与网络相连接的各种类型的设备。SNMP 使用默认的口令“public”（有一些自作聪明的供应商将其变为“private”）。

从以上这些攻击手段中，我们不难发现，没有一种攻击是可以通过加密技术避免的，甚至防火墙技术也不能避免全部的攻击手段。例如，易受攻击的 Web 服务器可以被放置到商务系统的后端，在它前面配置一个防火墙，但是对于某些恶意攻击该服务器仍然处于一种开放的状态中。而且如果防火墙运行在一个存在缺陷的操作系统之上，攻击者也可以很容易取得对它的控制权。

虽然到本书出版时，也许一些攻击手段已经被修复，但是这些攻击的底层模式还是相当固定的。大多数攻击程序的开发都是利用程序本身存在的错误和漏洞；这其中绝大多数是针对栈溢出进行的。其次是那些对协议本身存在的缺陷（例如 NFS）的攻击以及对于脆弱口令的攻击。

实际上，在攻击者和软件供应商之间正在进行着一场赛跑。攻击者试图去发现漏洞，同时软件供应商尽可能地弥补这些漏洞。那些有能力而且又有着不良动机的攻击者在发现漏洞后并不声张，而大多数已发布的攻击手段中涉及的漏洞早已经众所周知，甚至在网上就可以找到弥补这些漏洞的工具。

18.1.2　技术问题：脚本小子和打包防御

网络带来的一个主要的文化上的变化是，直到最近，对于通信的复杂攻击手段（例如中间人攻击）从根本上促成了国家政府部门的防护措施。当前，一些少年仅仅为了娱乐而进行口令探测甚至更加隐蔽的路由攻击。这里产生的变化就是，某些人编写了一些必要的开发软

件，然后放到像 www.rootshell.com 的站点上，从这里脚本小子能够下载和使用它们。脚本小子这个术语主要指那些爱恶作剧的年轻人，他们利用别人准备好的攻击脚本，还可能指那些不是十分懂技术的人们，他们下载和启动那些自己也不知道是用来做什么的工具软件。随着系统变得越来越复杂，甚至对那些在攻击领域中走在前沿的资深攻击者来说，也不可能掌握操作系统和网络协议中发现的所有漏洞。实际上，破坏攻击正逐渐变得没有那么多技术含量了，与此同时，防御正向着难于管理、更加复杂的方向发展。

如同在第 4 章中所讨论的，因特网协议族最初只是为了一些大学和研究实验室之间可以通过网络进行合作而设计的，设置在这些单位中的主机均可以被充分信任。原先网络的使用者大都是正直可靠并有高超技能的研究人员，但是，这一阶段早已过去。现在使用网络的人绝大多数完全不合格（他们中的许多人长时间通过高速线路连在网上），不过也有一小部分是正直并具有技能的用户，但还存在更少一部分人具有高超技能且心怀恶意，也有一小部分是心怀恶意但仅仅会使用那些现成工具的人。

技术上的欠缺是影响防御质量的一个重要因素。只有很少一部分机构，例如计算机公司、重点大学和军队情报部门，拥有懂得如何进行跟踪和调试防御体系的专业人员。而大多数公司则依靠标准产品和标准服务的组合来进行防御体系的构建。这些产品包括防火墙、病毒扫描软件和入侵检测系统，服务通常是通过这些产品的新配置文件的形式交付使用的。在这些方法中，系统漏洞变得集中起来，此时，一旦这些销路广泛的系统中的某些缺陷被攻击者抓住，那么将有很大范围的被攻击目标受到威胁。

现在让我们来看一些特定的攻击和防御机制。记住，最重要的攻击手段是栈溢出攻击，其次是口令猜测攻击，但是由于在第 4 章和第 2 章至第 3 章中已经分别涉及了这两种攻击手段，我们将讨论第三种攻击方式：网络协议攻击。

18.2　网络协议攻击

在 Unix 和 Windows NT 这些日常使用的操作系统中通常运行着许多网络服务。其中有许多种服务，无论对于攻击者还是系统合法用户都是默认允许使用的，还可以通过类似即插即用那样简单的机制来配置加载运行。我们将从本地局域网和因特网两个范围分析这类问题。在这里，一个通常关注的主题是关于映射方法（映射地址、文件名等等）中存在的诸多漏洞。

本书不适合用来详细解释各种网络协议。所以，这里仅仅提供一个简要描述：网际协议（IP）是一种无状态协议，它将数据从一台机器传送到另一台机器；该协议使用 32 位的 IP 地址，通常写成 4 个取值在 0～255 的十进制数字的形式，例如 172.16.8.93。大多数的因特网服务使用传输控制协议（TCP），该协议在分层结构中位于 IP 层之上，提供虚电路，即将数据流拆开放入 IP 包而到远端重新组装，对那些丢失的数据包则采用重发机制。IP 地址被转换成我们熟悉的因特网主机地址使用的是域名服务器（DNS），这是一个全球范围的分布式服务，其中高层名称服务器指向本地关于特定域的名称服务器。本地网络大多使用以太网技术，网络中的设备具有独一无二的以太网地址，将它映射到 IP 地址则要使用地址解析协议（ARP）。

在协议族中还有许多其他的组件用来管理通信和提供高层服务。它们大多是在网络中仅仅包括可信任主机时开发的，安全问题并没有被考虑进去。所以协议中很少内置认证功能，

试图补救这个缺陷的下一代 IP 技术（IPv6）的广泛应用还需要一段时间。

18.2.1 局域网攻击

现在假定攻击者就是你的雇员；他拥有一台连接到公司的局域网上的机器，他想要获取他人的账户从而实施欺骗。考虑到他拥有到局域网的物理连接，所以，他能够安装包嗅探器软件来获取口令，可以得到 root 用户的口令，进而创建另一个合适的账户。然而，如果公司规定需要使用询问—响应型的口令生成器，或者仅仅在应该使用 root 用户口令的机器上使用该口令，那么这位攻击者就必须使用其他隐蔽的手段了。

一种办法就是试图将机器伪装成目标用户已经登录的状态。ARP 就是一种可能的攻击目标；通过运行适当的代码，攻击者能够在回复 ARP 消息的时候返回错误的应答，从而将自己伪装成被攻击的目标机器。目标机器如果警觉可能会发现这个问题，但攻击者总是能够等到它关机再进行攻击，或者利用其他攻击手段先将目标机器攻击成不可运行状态。一种可能的方法是使用*子网掩码*。

最初，IP 地址使用最前面的 3 位来确定网络地址和主机地址在哪里划分。现在，利用可变的*网络掩码*将 *IP* 地址解释成网络、子网和主机三个部分。当无盘工作站启动时，需要利用广播的方式来请求一个子网掩码，许多机器对返回的任何子网掩码都会不加判断地加以应用。所以，如果发送一个适当的子网掩码给该工作站，就可以使它从子网中消失，不可被访问。

还有一种方法，如果公司使用 Unix 操作系统，用来提供 Sun 公司的网络文件系统（NFS）的服务，这是一种关于 Unix 文件共享的事实上的标准。该文件系统允许许多工作站像使用本地磁盘那样来使用网络磁盘；同时它也存在许多众所周知的缺陷，而这些缺陷可以被登录到同一个局域网上的攻击者们所利用。当一个文件卷首次安装时，客户端向服务器请求一个根文件句柄，即被安装的文件系统的根目录。这个操作并不依赖于时间，或者服务器端产生的数字，而且这个操作是不可逆的。此时并没有用户认证机制，服务器必须完全相信每一个客户端请求。还有，NFS 服务器常常通过另一个与客户请求不同的网络接口来发送应答消息。所以，有可能等到管理员登录到文件服务器以后，伪装成他的身份来改写口令文件。由于这个原因，目前许多站点都使用另外一种替代的文件系统，例如 ANFS。

18.2.2 使用因特网协议和机制的攻击

我们进一步看看因特网协议族的情况，存在的基本问题十分相似，这就是其中大多数的机制并没有真正意义上的认证或者保密性防护手段。尤其在低级 TCP/IP 协议中体现得更加明显。

例如，考虑 Alice 利用三次握手机制来初始化与 Bob 的 TCP 连接的情况，在这一过程中建立了序列号，如图 18-1 所示。

```
A → B:   SYN; 我的号码是 X
B → A:   ACK; 现在变成 X+1
         SYN; 我的号码是 Y
A → B:   ACK; 现在变为 Y+1
         (开始会话)
```

图 18-1　TCP/IP 握手

这一协议正被多得不可思议的方式所利用。既然拒绝服务正逐渐变得重要起来，那就让我们从最简单的拒绝服务攻击开始吧，这就是 SYN 洪流。

18.2.2.1 SYN 洪流

对于 SYN 洪流攻击手段，简单来讲，就是发送大量的 SYN 包，但永远不确认任何的应答消息。这将导致接收方（Bob，见图 18-1）接收过多的 SYN 包记录，从而超过了软件可以承受的负载。这种攻击在 20 世纪 80 年代就从理论上证明了其可能性，但是直到 1996 年，利用该种攻击方法使得纽约的一家 ISP Panix 瘫痪许多天时才逐渐引起公众的注意。

作为一种技术上弥补手段的 SYNcookie，被 Linux 以及其他一些系统所采纳。此时接收方（B）不再保留对方发来的 SYN 包的拷贝，而是简单地发送一个加密过的 X 的版本 Y。通过这种方法，就没有必要再保留那些半开放的会话状态了。

18.2.2.2　smurfing

另一种使主机瘫痪的常用方法就是 smurfing。该方法利用的是因特网控制报文协议（Internet Control Message Protocol，ICMP），该协议使得用户可以向远程主机发送回复报文来告知其处于活动状态。问题出在许多主机共享广播地址的场合。对于广播地址或者机器本地地址使用 ping 程序时，某些因特网协议的实现情况是对二者均进行响应（这是基于可以在局域网范围内探测主机是否处于活动状态的思路而设计的）。所以，这种协议在路由器中对以上两种操作均给予支持。我们把这种同一广播地址的一组主机的响应称为 smurf 放大器。

此种攻击方法是创建一个包，包中的源地址伪造成被攻击机器的地址，然后将其发送到许多 smurf 放大器中。这些机器如果处于活动状态，就会给被攻击的机器发送一个响应包，由于响应包数量巨大超过被攻击机器所能承受负载的极限，从而导致系统瘫痪。smurfing 在典型的情况下被用来控制在线聊天系统（Internet Relay Chat，IRC）服务器，所以，它可以获得整个聊天室的控制权。这种攻击手段可以自动地让网络中许多无辜的机器成为牺牲品。

解决措施的部分内容是需要技术上的变革才可以完成的。1999 年 8 月对于协议标准的变更使得对于广播地址的 ping 操作不再会得到回复 [691]。随着这项措施的实施，网络中 smurf 放大器的数量也随之大大减少了。措施的另一方面是有关社会经济学的：像 www.netscan.org 这类的站点产生了许许多多的 smurf 放大器。那些勤奋的管理员们常常注视着他们的网络，从而可以及时地修复这类攻击；比较懒惰的管理员们也将会发现，他们的带宽被那些攻击者使用得越来越少，而被迫修复这个问题。

18.2.2.3　分布式拒绝服务攻击

于 1999 年 10 月出现的一种攻击手段叫做分布式拒绝服务攻击（distributed denial of service，DDoS），它同样延续了以上两种攻击手段的思路。smurfing 攻击仅仅是利用一种普通的配置错误来达到攻击目的，而分布式拒绝攻击则与之不同，攻击者在一段时间内暗中破坏许多机器，并在它们中安装自定义的攻击软件。在一个预先定义好的时间到来时，或者当一个给定的信号到来时，这些机器都开始利用消息同时轰击目标站点 [253]。这种破坏手段可以通过使用类似 Morris 蠕虫的方法自动得以执行。

到目前为止，DDoS 攻击已经袭击了许多影响力很大的网站，包括 Amazon 和 Yahoo。如果攻击的目标是诸如 DNS 这类的服务，则还会引起更大的混乱，甚至是搞垮整个因特网。这类攻击也许在信息战中是被期望采用的；它也可能是个人的一种破坏性活动。有意思的是，2000 年初被攻击的主机大部分都是来自美国的医学站点。这些主机尤其容易遭受攻击，因为美国食品药物管理局建议对于医学上使用的 Unix 机器，一旦被赋予了某些特定的研究目标，就会有一个众所周知的标准配置。一旦在这些机器上发现了错误或者漏洞，对于那些

安装了攻击软件的机器来讲，它们就自动成为了可被攻击的对象。

在写本书的时候，出现了防止 DDoS 攻击手段的措施，即在底层结构中增加 ICMP traceback 消息机制。它的思路是不论何时只要路由器转发一个 IP 包，它都会按照大约 1:20 000 的比例加入一个 ICMP 包，共同传送到目的地址。这个包尽可能多地包含了前一跳、后一跳的所有细节信息。即使攻击者们采取伪造源 IP 地址的方法来隐藏其行踪，系统管理员也可以利用该包的信息来反向寻找那些必须对引起大规模包流量负责任的主机 [93]。该措施也可以抓住那些滥用开放中继器兜售信息的人，这些中继器不会在电子邮件信息头部加入审计信息。

18.2.2.4　兜售信息和地址伪造

电子邮件服务或者是某些 Web 服务，例如简单邮件传输协议（SMTP）和超文本传输协议（HTTP）等均假设位于它们底层的协议是安全可靠的。大多数服务通常都要用到通过 IP 地址来查询主机名的 DNS 服务。所以那些能够伪造 IP 地址的人就可以滥用设备资源了。一个最普通的例子就是兜售信息的人所采用的邮件欺骗，当然也存在许多其他的方式。例如，如果攻击者将有关你公司网页的不正确信息提供给 DNS 服务器，那么所有访问这个网页的请求将被转向其他站点，无论你采取任何措施都无济于事。由于这种攻击手段通常涉及在本地缓存的 DNS 表格中加入错误的信息，所以称为 DNS 缓存中毒。

18.2.2.5　哄骗攻击

我们可以综合前边讨论过的一些思想和方法，从而形成攻击范围更广（从局域网或域的外部）的攻击手段，称之为哄骗攻击。

假设，如果 Charlie 知道 Alice 和 Bob 是他将要攻击的目标网络中的两台主机，而且希望在 Bob 面前伪装成 Alice 的样子。他可以利用某种拒绝服务的攻击方法来使 Alice 瘫痪，然后初始化一个新连接到 Bob 机器 [559, 90]。这需要猜测 Bob 赋予本次会话的序列数字 Y，请参看前面图 18-1 中的三次握手协议。一个猜测 Y 值的简单有效但花费时间较长的办法就是，Charlie 预先取得和 Alice 的一个真正的连接，然后通过实际的 Y 值变化情况来预测下一次连接时 Y 的取值。现在的栈都使用随机数字生成器和其他技术来避免这种预测的发生，但是随机数字生成器比预期的随机性要差很多，这也产生许多安全缺陷的一个根源所在 [774]。

如果猜测序列数字的工作是切实可行的，那么 Charlie 将能够给 Bob 发送消息，而在 Bob 看来他会相信这些消息确实来自 Alice（尽管 Charlie 是不会将 Bob 的应答读给 Alice 的）。在一些情况下，Charlie 甚至不需要去攻击 Alice，而仅仅通过安排某些事件让 Alice 将 Bob 的应答当成不期望得到的垃圾信息丢弃就可以了。这是一种十分复杂的攻击手段，但是不论怎样，网上有许多可用的脚本能够完成这项工作。

18.2.2.6　路由攻击

路由攻击适用于许多种不同的场合。基本的攻击形式包括 Charlie 告诉 Alice 和 Bob，在他们的站点之间存在一条更加便利的路由，而该路由恰恰通过 Charlie 的主机。路由最初被引入 TCP 是用来帮助信息流绕过不合适的路由器而使路由更加有效的一种机制，但是它也是建立在主机都是安全可靠这一前提之下的。路由选择认为最佳的返回路由就是最好的源路由。当该路由无法承受数据流时，惟一的短期解决方案就是阻塞源路由。然而，它仍然被网络分析系统用来进行诊断。

另一种方法包括重定向消息机制，它也基于主机都是安全可靠这一错误的假设。对此，

更为有效的一种说法就是："你应该把消息发送到另一台网关上"，而你在应用之前也没有做任何检查。用这种方法就可以破坏源路由消息传输。

兜售信息者使我们几乎每一个人都明白，邮件伪造的手法过于琐碎和微不足道。而重选路由的方法要强大得多，因为邮件路由正是以 DNS 为基础。但是随着数量众多的服务提供商的出现及其竞争者的垮台，这种攻击手段的效果变得不是很严重。DNS 缓存中毒仅仅被当作一种可以使用的技巧。

18.3　防御网络攻击

现在，期望系统对于大多数的攻击手段，至少是对那些脚本小子所使用的手段可以有效地防御似乎变得合情合理了。系统管理员们密切地关注着安全公告牌，将所有软件提供商的补丁程序都应用到自己的系统上，这些都将有效地阻止攻击的发生。这部分将广泛地讨论有关配置管理的主题。

18.3.1　配置管理

严格的配置管理对于网络的安全来讲是相当重要的一环。如果你能够保证企业中的所有机器都在运行最新操作系统的拷贝；一旦软件补丁发布就被及时地应用；所有服务和配置文件都不存在严重的漏洞（例如可以被更改的口令文件）；已知的默认口令都随着产品的安装被清除掉；所有必要的东西都依据有组织的原则在合适的地方有备份，那么你将可以处理当今最流行的十大攻击手段中的九条半了（你还必须小心应付应用程序代码中存在的漏洞，例如 CGI 脚本，但是只要无法用管理员特权来运行它们，那么是不会造成什么严重的损失的）。

配置管理的重要程度完全可以和拥有一个完善的防火墙相提并论。实际上，如果要你只能从二者中选择其一的话，你应该忘记防火墙而选择配置管理。然而，这对于许多公司来讲都是很难做出决定的选择，因为它将竭力反对购买和安装一种现成的产品。对于大部分人来讲，做配置管理甚至是将事情搞得更糟。正如前面我们提请注意的地方，美国医院被规定必须使用一种公开的配置方案，它将给那些坏孩子提供一大堆由于错误配置引起的攻击目标。

一些工具可以帮助系统管理员对事情做严格的处理。其中一部分使你做集中式的版本控制工作，以至于补丁可以在前一天晚上被应用，所有事情都将保持同步状态。其他一些像 Satan 这类的工具使用一系列通用的系统漏洞试图闯入你的网络中的机器上［320］。熟悉和精通这些攻击软件是一个不错的主意，可以利用它们来反向攻击攻击者。

这些可以利用的工具的详细情况总是一年一年不断变化，所以在这里并不适合讨论它们的细节问题。系统管理员们在阻止攻击时，同时需要具备技术和责任心两方面的因素，这才是我们目前最关心的。就算是勤奋用心的机构也会发现，要想试图修复系统中所有的因特网连接范围内的安全隐患花费将是十分巨大的，虽然这些隐患在局域网环境还可以忍受，但在广域网范围就很难补救了。另外一个问题也常常发生，由于管理员不注意操作系统的升级和防止服务丢失的补丁程序的安装，使机构中最为重要的应用程序运行在安全性很差的机器上。

这将导致我们将注意力放在对防火墙的使用上。

18.3.2　防火墙

针对因特网安全问题的最为广泛的商业解决方案是防火墙。它是一台位于本地局域网和因特网之间的机器，用于过滤掉那些可能有害于系统安全的数据流。许多机构都对这种类似“盒子中的解决方案”的思路产生了浓厚的兴趣，该技术被广泛接受，以至于它已经被看作是整个机构安全体系中必不可少的一个基本组成部分（许多购买者在购买防火墙时往往选择贵的，而不注重其性能）。

防火墙基本适用于以下三种情况，这取决于它们是否在IP包这一层进行过滤、是否在TCP会话层进行过滤，或者是否在应用程序层进行过滤。

18.3.2.1　包过滤

最简单的防火墙仅仅提供包地址过滤和端口号过滤的功能。这种功能也存在于路由器和Linux操作系统中。通过保证没有标识为从本地局域网发出的数据包从局域网外部发来，该防火墙就可以阻塞前边介绍过的IP哄骗类型的攻击。它还能够通过控制发往某台主机的数据包或者试图连接自己的主机来避免拒绝服务类型的攻击（这两种问题主要针对还在运行Windows 95的用户）。

虽然基本的包过滤功能已经成为Linux操作系统的标准，但是，就那些不断出现的攻击形式来讲，该功能可以被许多种诡计所欺骗而导致防御失败。例如，一个包可能被拆成一些分片，最初通过防火墙检测的分片可以被后续的分片所覆盖，对源地址取而代之的是一个与防火墙所采取的安全策略相冲突的地址。

18.3.2.2　电路级网关

更多的被称为电路级网关的复杂防火墙重新组合和检测每一个TCP线路中所有的数据包。这将比简单的包过滤功能付出更昂贵的代价，它还可以提供附加的功能，这包括通过防火墙之间的信息加密功能而提供的因特网网间虚拟专用网（virtual private network，VPN），屏蔽了那些处于黑名单上的网站和新闻组（据报道，亚洲政府正在为此目的建立国家级别的防火墙体系）。

然而，电路级的保护不能阻止应用程序层的攻击，比如恶意代码。

18.3.2.3　应用中继

第三种类型的防火墙称为应用中继，它作为一种或者多种服务的代理。这些服务包括邮件、远程登录和Web。在应用程序级别上，你可以使用一些强制措施来去除Word文档中存在的宏病毒，去除网页中包含的活动内容，这些都可以提供更加全面广泛的保护来防止多种攻击手段的威胁。

应用中继型防火墙也存在其明显的漏洞，那就是它将成为系统中一个严重的瓶颈。它也会妨碍用户运行一些最新的应用程序。

18.3.2.4　系统入口和出口过滤

目前，几乎所有的防火墙的防御都指向系统外部，试图将不安全因素隔离到系统之外。当然也有少数军用系统会监视系统内部向外部发送的数据流，从而保证机密信息不会泄漏出去。

也就是说，一些商业机构也开始监视系统发向外部的数据流。如果公司中的某些机器正被用来进行拒绝服务攻击而发出请求（就像在[771]中所提出的那样），这些向外的包过滤

至少在理论上可以被用来发现和阻止这种攻击手段。还有，现在偷窃秘密信息的设备（*snitchware*）和技术正有增长的趋势，利用它们可以在没有授权的情况下收集和转发关于在线用户的信息。像“phones home”这类软件，表面上是为了增强版权和便于购物的目的设计的，其实它可以向外界暴露本地硬盘目录中高度敏感的信息。我期望那些谨慎的机构能够增加对这种系统向外的数据流的监控力度。

18.3.2.5　联合

在高度戒备的站点中，是需要使用多重防火墙机制来加以保护的。这可能是一个堵塞数据流（choke）或者包过滤防火墙，从而将外部世界和过滤数据的子网连接起来，这被称为非军事区域（demilitarized zone，DMZ），它包括了许多应用程序服务器或者代理来过滤邮件和其他类型的服务。然后 DMZ 将通过进一步的过滤再被连接到内部网络上，这里要进行网络地址的转换。在机构内部，会有更多的边界控制设备，包括各个分部门的泵，或者其他网络，它们由于重要程度不同所以保护级别也不相同，从而保证机密信息不会随便地流入或者流出，见图 18-2。

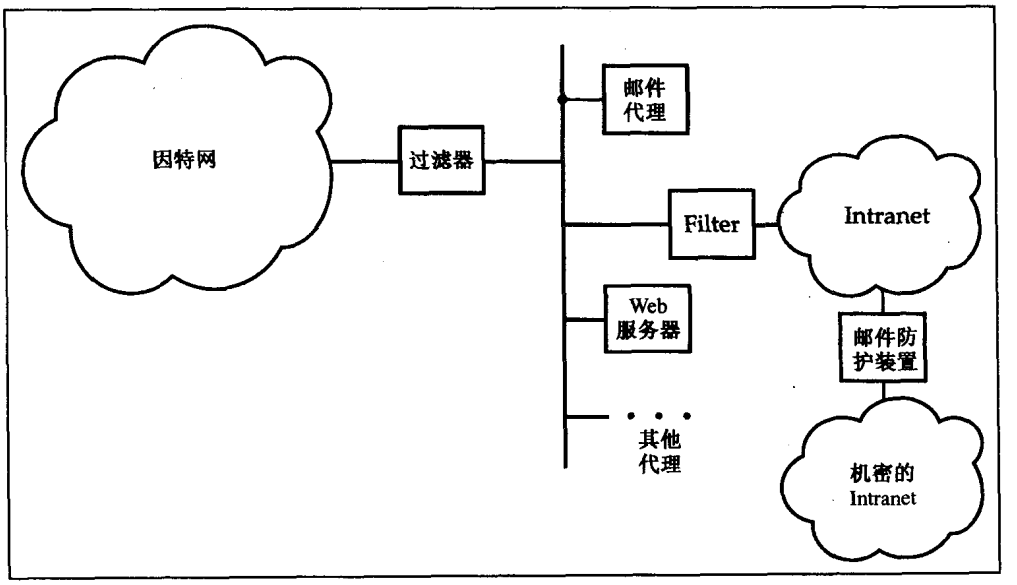


图 18-2　多层防火墙

这类精心安装的设备可能带来更大的操作上的花费，许多日常消息需要被人工检查核对后，才能通过网络。这将在很大程度上阻碍用户通过安装未授权的后门程序来完成他们的工作，例如单机拨号程序。而且，如果系统的主要目标是防止重要信息泄漏的话，这将对病毒的侵入没有多大作用。一旦某些地方出现问题，将会导致极其严重的后果。我稍后将在 18.4.6 节讨论这种类型的问题。

18.3.3　防火墙的作用和局限性

既然防火墙所能够做的只是很少一部分事情，那么就可以把它们做得十分简单，将底层操作系统中的许多复杂的组件移除掉（例如 Unix 系统中的 RPC 和 sendmail 机制）。这种做法去除了许多缺陷和错误的根源。对于许多机构而言，吸引它们的是管理小数量机器的思路，

而不是试图通过适当的系统管理，来对数量巨大的异构型机器进行管理。

从另一个方面来讲，这种简单性所带来的吸引力是诱人的，同时也是危险的。防火墙仅仅在其被正确配置后才能发挥良好的作用，而许多机构并没有学会如何将这件事情做得足够好。它们希望将危险信息挡在系统之外，希望将防火墙插入系统中适当的位置就可以将问题解决。事实并不如此简单。也许管理防火墙所要付出的努力，并不像网络首次运行时需要对每台机器进行配置那样多，但确实也需要做一些事情。在［203］中，有一个如何在 Hanscom 空军基地部署防火墙的学习实例。这项工作包括以下一些内容：调查用户群体来获取网络需要提供何种类型的服务；设计网络安全策略；使用网络监视器来发现正在使用中的不良服务，以及安装以前先在实验室中进行测试。一旦系统被安装和投入运行，将会出现某些问题，例如需要不断进行的维护（这取决于网络成员利用率等因素）和与其他军用基地通信时无法监测的状态，还有调制解调器的共享等等问题。非军用组织往往不注重这些问题的处理。

至少在 20 世纪 90 年代后期，军用网络还为涌入市场的许多产品质量不够理想而感到担心。但商业发展速度很快，有许许多多的供应商加入到这个领域中来，但是可用的专用技术传播却不尽如人意。

大规模的交易中存在安全和性能之间折衷的问题。你安装过简单的过滤型路由器吗？它不需要太多的维护。你试图在 DMZ 中使用应用中继这类完全需要自己配置的防火墙吗？它不仅需要长期不断的重新配置，因为用户可能不断要求大量新型服务可以通过防火墙，同时也会成为系统瓶颈。

发生在英国的一个例子是 NHS 网络，这是一个私人内部网络，其目的是为提供健康服务的用户使用，包括家庭医生、医院和诊所，总共有 11 000 家机构的大约一百万位使用者。起初，这个网络仅仅通过单一的防火墙连接到外部世界。设计者认为这已经足够了，因为在他们看来大部分的网络流量都是在网络内部进行的，在以前的健康服务数据流中确实如此。但是设计者没有预见到的是，随着因特网在 20 世纪 90 年代中期的出现，40% 的网络流量都变为国际化了，不再仅仅局限于内部网络之中。医生和护士发现通过网络咨询位于美国之外的医学参考站点也是一件十分方便的事情。如此一来，试图将所有网络流量从一台防火墙中挤压过去的做法变得不太现实了。而且，既然几乎所有的对保健系统的攻击都来自系统内部，那么中心防火墙机制所起到的作用也不像最初时那样明显了。

另一个关于防火墙（广义的说是指那些边界控制设备）的问题是它们在某种程度上阻碍了用户所需要做的事情，似乎所有操作都必须围绕它们进行。由于大多数防火墙将对网页的请求（典型情况下它将使用 80 端口）不加控制地通过，所以越来越多的应用程序也使用 80 端口，它们利用这种方法就可以通过防火墙的拦截了。在使用 80 端口行不通的场合，可以将全部的服务重新实现为 Web 服务（webmail 就是很好的例证）。这些情况不断地削弱防火墙的效能，这使我们想起了 John Gilmore 的一句名言："因特网中的审查制度其实就意味着破坏，但尽管如此，各项工作还是要围绕它进行。"

最后，有必要回到前面讲过的十大攻击手段上面来，究竟防火墙能够阻止它们中的多少呢？这主要取决于防火墙是如何配置的，如果实际回答这个问题可能要花费不少时间。

18.3.4 加密技术

在防止网络攻击的过程中，许多人都考虑过使用加密技术。的确，有时这将会是一种很

有用的做法。例如，在我工作的实验室的网络中，我们使用一种名为 secure shell（SSH）的产品，它可以提供对于 Unix 和 Windows 主机之间的加密连接［817，1，597］。当我从家中拨号到实验室的网络时，我的通信流量是被加密过的。而且，当我从实验室中的一台主机登录到另一台主机的时候，我使用的口令也不会在局域网中显式传输，它也是被加密过的。

让我们先停下来分析一下这种情况给我们带来的保护。新手和警察局可能会考虑使用窃听的手段来攻击，但是由于拨号上网的调制解调器使用自适应回声消除技术，所以使得窃听它们相当困难。它使攻击者只能在从我家到实验室的连接两端再加入两台调制解调器的办法来进行攻击。所以这种欺骗的可能性就很低了。相比之下，在局域网中被窃听口令所要冒的风险就要高得多了，这在以前实验室的其他部门中出现过类似情况。因此，我们的网络实际上采用另一种低成本的替代办法，那就是使用便携式口令生成器。

另一种方法是在 IP 层做一些加密和/或者认证的工作，这被 IPv6 所支持，也可以在当前的 IP 版本中利用新式的 IPsec 机制来提供支持。在［290］中我们可以得到关于这种协议的评估，它的实现在［782］中被描述。IPsec 具有阻止某些网络攻击的能力，而且将成为设计健壮的分布式系统的一个有用的组件，但是它也不是万能的。许多机器将不得不与所有的客户建立连接，如果我通过破坏堆栈的方法来成为你的 Web 服务器的管理员的话，那么诸如加密或者认证这类做法就帮不了你什么忙了。在网络内部，许多其他的机器将会被攻击，因为某些机器已经被一些不正直的人所操纵。另外，网络中仍旧会出现拒绝服务之类的攻击。而且，部署系统也将是要花费很长一段时间才能够完成的工作。

第三种思路是虚拟专用网（VPN），这种想法是：一个公司的许多分支部门，或者相互存在交易的许多公司之间，通过自己的防火墙对站点之间交互的信息进行加密处理。通过这种方法，因特网可以将它们的局域网络连接起来，但是它们之间传送的信息却没有暴露在外边，不能被监听。VPN 技术也不能避免某些坏分子试图去破坏你的 Web 服务器的堆栈或者在你的局域网中监听口令，但是对于公司来讲，这可以增加发达国家政府提供的广告收益，减少数据在因特网中传递时所遭受中途监听的可能性（必须要说明的是，中途监听大量包数据比某些加密公司所声称的要困难得多，那些没有什么资金的攻击者将会使用其他的攻击手段）。

加密技术也有其弊端。一个很明显的问题就是如果被加密的邮件和网页可以通过防火墙，那么它们将可以携带各种不良信息一同通过防火墙的检查。这将会带来恶意代码的问题。

18.4　特洛伊、病毒和蠕虫

如果这本书写在 5 年以前，恶意代码将单独占据独立的一章。

计算机安全专家们长期以来一直关注来自恶意代码的威胁，或者称为 malware。第一个这种类型的程序叫做特洛伊木马，这个名字来自古希腊军队佯装留给特洛伊人的礼物，一只内部藏有许多古希腊战士的木马，它们随后打开了特洛伊城的大门，使古希腊军队攻陷了城池。用特洛伊木马来形容恶意代码已经沿用了很多年（参见［493］中的讨论）。

还有病毒和蠕虫，它们都是能够自我繁殖的恶意代码，在前面的章节中已经反复提及。关于这三个术语的准确定义目前还存在着争议：一般的使用方法是，特洛伊木马是指一种被不可疑的用户运行且做出恶意事情（例如获取口令）的程序；蠕虫则重在说明其复制性；病

毒是一种可以通过将自己附加到其他程序中的蠕虫。

18.4.1 早期的恶意代码

当足够多的用户共享一个计算平台的时候，恶意代码最容易出现。这可以追溯到20世纪60年代早期。那个时代的机器速度很慢，它们的CPU周期被精细地分配给不同组的用户使用。由于学生总是被排在队列的尾部，所以他们恶作剧地开发了计算机游戏程序，内含特洛伊木马，该木马用来测试程序是否被root用户执行，当发现程序被root运行时，就会生成额外的拥有特权的账户，并使用一个公开的密码。到20世纪70年代的时候，大学中的大型分时共享系统成为了越来越多包含特洛伊木马的恶作剧程序攻击的目标。所有的骗局都这样被开发出来。

1984年，在Thompson写的一篇著名论文中提到，即使系统源代码被仔细地检查，而且被证明没有漏洞，但是一个陷门仍然可能被插入到源代码中来。他的诀窍是将陷门创建到编译器中去。如果该编译程序意识到它正在编译一个登录程序，那么它将插入一个陷门，例如主口令，该口令将适用于任何账户。当然，有些人可能会通过检测编译器源代码的方法来避免这种事情发生，然后再从头开始对它进行编译处理。所以下一步工作要特别留意的是，当编译程序发现自己正被编译时，那么即使不直接出现在源代码中，编译程序也会插入某些漏洞到重新编译过的结果中去。所以，即使你购买的系统是针对操作系统、应用程序和工具经过安全检验的，编译成的二进制代码中也同样会包含特洛伊。所以，我们建议，除了你自己编写的系统外，其他系统都不可相信，脆弱性或者漏洞在你使用的工具链的任何一个环节都有可能被插入到系统中去[746]。

计算机病毒在1984年也终于突然出现了，这要归功于Fred Cohen的论文。他在各种不同的操作系统平台上完成了一系列试验，这些试验表明了如何使代码从一台机器到另一台机器进行自我繁殖，以及如何（如在第7章中所提到的）从多层系统中的一层繁殖到另一层中。这个思想使得计算机界惊惶失措，在短短三年中，第一批真正意义上的病毒在“民间”诞生了。它们几乎都是个人计算机（PC）病毒，当时磁盘操作系统（DOS）是PC上主流的操作系统。当用户们在磁盘上共享程序，或者通过电子公告牌的方式，这些病毒就从一个用户传播到另一个用户。

一个更具报道价值的病毒实例是圣诞卡病毒，它发作于1987年，通过IBM的大型机进行传播。类似最近出现的爱虫病毒，它们都利用电子邮件进行传播，但在当时这种病毒已经相当先进了。第二年又出现了因特网蠕虫，它给更多的人们和普通的大众敲响了警钟。

18.4.2 因特网蠕虫

有关拒绝服务攻击最著名的实例就是1988年11月的因特网蠕虫。这是Robert Morris Jr编写的一个程序，它利用了系统中的诸多漏洞来进行机器间传播。系统中某些漏洞很通用（例如在猜测口令攻击中使用432种常用口令，以及由.rhosts文件所造成的偶然的使用权等）。其他的漏洞则是针对某种特定系统的，例如在4.4.1节中提到的sendmail和fingerd漏洞等。蠕虫一步一步伪装自己，这叫做sh，而且还将自身的数据串进行加密处理（虽然还必须自身携带恺撒密码）。

Morris声称这个代码并不是在因特网上的故意攻击手段，而仅仅用来试验它的代码是否

可以在机器之间进行复制。结果当然是可以。它本身也存在一个漏洞，它本应该识别出那些已经受感染的机器而不再感染它们，但这种应有的特征并没有起作用。这样一来，所造成的结果就是因特网上大量的通信流量被完全阻塞。

假使因特网（或者，更准确地说应该是它的前身 ARPANET）在最初设计时就表现出极好的融灾性来抵御各种攻击形式，这些形式从小到大甚至包括可以威胁到整个网络核心策略的攻击形式，一个学生编写的程序也可以很容易就使因特网完全瘫痪掉。

很少有关于下列情况的报道：不干净代码被清除掉，常规服务在一到两天中就被恢复；受到攻击影响的只有 Berkeley Unix 和其派生的系统，或者说只影响 Microsoft 公司的单一平台；人们之所以保持冷静，不急于恢复网络连接，主要是因为还没有找到问题出在哪里和如何修复它。

18.4.3　病毒和蠕虫如何工作

病毒或者蠕虫在典型情况下都具有两个组成部分：复制机制和负载。蠕虫在运行时，简单地将自身复制到其他地方，也许是闯入其他系统（就像因特网蠕虫所做的那样），或者是作为电子邮件的附件将自身发送到系统的电子邮件列表中的所有地址上（就像最近的一些蠕虫病毒所做的那样）。在 DOS 病毒的年代，病毒最常见的复制方法就是将自己附加到某个可执行文件的尾部，然后将自身融入代码中，以至于实际的执行过程是先跳转到病毒代码处，然后再回来执行原来可执行文件的程序代码。

在 DOS 下，最简单的病毒是那些感染 .com 类型可执行文件的病毒。这种文件类型总是从 0x100 地址处开始执行代码，所以对病毒来讲，只需简单地将自己附加到 .com 文件的尾部，然后利用一条跳转语句来代替 0x100 处的指令，使程序跳转到病毒文件的开始地址处就可以了。因此，一旦 .com 文件运行，病毒就可以随之执行了，它一般是寻找其他没有被感染的 .com 文件，然后去感染它们。当病毒做完了它的工作，那条被替代的指令将被执行，这使得控制又归还给宿主程序。

考虑一个特定的平台，例如 DOS，常常会有额外的技巧可以被病毒编写者所利用。例如，如果目标系统有一个名为 accounts.exe 的文件，则病毒可能是以 accounts.com 的名字被引入系统，而后者将被 DOS 首先执行，这叫做伴生病毒。DOS 病毒还可以攻击引导扇区或者分区表。甚至还有可打印病毒，即病毒所有的操作代码都是可打印的 ASCII 字符，这意味着这些病毒可以在纸上繁殖。许多 DOS 病毒的细节问题在 [512] 中讨论。

病毒的第二个组成部分是负载。它可以被触发器激活，例如日期，激活后做一件或者多件坏事情：

- 对机器的保护状态进行选择或者随机改变（这正是我们对于多级安全系统所担心的）。
- 对用户数据的选取和随机改变（例如，破坏磁盘数据）。
- 封锁网络功能（例如，将复制速度设到最大值）。
- 为了某些险恶的目的而偷窃资源（例如，使用 CPU 进行 DES 关键字查找）。
- 使你的调制解调器拨号到某个收取额外费用的号码，这类通过电话进行攻击的人可以收取你的费用。
- 偷窃甚至公布你的数据，包括加密密钥。

- 创建后门程序，通过它使得创建该后门程序的人可以接管你的系统，或许发起一个分布式的拒绝服务攻击。

到目前为止，最具有破坏作用的是那些在系统中留有后门程序以备后用的负载，还有那些缓慢而且察觉不到其在做破坏工作的负载。对于第二种病毒的例子就是偶尔交换文档中的单词或者文件块，等到这种破坏引起管理员注意的时候，所有备份数据也许已经被破坏了。但是我们等了很久直到2000年9月21日才出现了一则关于负载病毒的报道。瑞士银行UBS提醒其用户注意一种病毒，一旦用户机器感染了该病毒，将会被窃取口令从而访问电子家庭银行系统。

各种各样的病毒编写者也编制一些“善意”的负载，例如为了对公司进行软件升级，为了强迫执行许可证期限，甚至为了周游世界来寻找便宜的机票，这就是所谓的智能代理，但前提是商业网站的所有者允许外来代码在其Web服务器上运行以获取价格信息，这目前还只是一个美好的愿望而已。

18.4.4 竞争

一旦病毒和防病毒软件公司同时出现，那随之而来的就是二者之间的竞争，它们都期望以智取胜。

病毒经常需要使用某些方法来识别自己，从而不会感染同一文件两次，一些早期的防病毒软件对文件进行免疫处理，即通过预先加入足够多的病毒来使病毒认为该文件已经被感染过了。然而，这并不是一种行之有效的办法，而且对于庞大的病毒家族来讲根本行不通。下一代的防病毒软件是*扫描器*，这些程序对早期的每一个可执行文件的执行路径彻底搜寻一遍，以试图找到表征特定类型病毒的字符串。

病毒编写者对此有各种不同的应对措施，例如，将病毒入口点放到主机文件代码中，因此迫使扫描器型的防病毒软件只能通过检查整个文件空间来判断感染情况；也可以针对流行的防病毒软件进行反攻击。最近在病毒的演化过程中出现了*多态病毒*。它们可以在每次复制时改变其自身代码，从而使得很难做出高效的扫描器。典型情况是，这些病毒被加密，同时有一个短小的头部包含解密代码。在每次复制时，病毒利用新的密钥重新对自己进行加密处理，也可以插入一些毫无关系的操作到解密代码中，还可以改变某些无关紧要指令的执行顺序。加密算法往往十分简单，很容易解开，但是也足以使扫描器的速度降到很低的程度。

另外一种防御病毒的主要技术是*校验和检查器*。这是这样一类软件，它保存着系统中所有已被授权的可执行程序的列表，而且还包括对于这些文件的最初版本的校验和。然而，一种流行商业软件也不过是仅仅使用两个不同的多项式来计算循环冗余校验；而这项技术很容易就被病毒编写者击败。在计算校验和过程中使用一种相当好的算法时，相应的策略就是*秘密行动*。这种做法就是病毒监视操作系统对于校验和操作的调用，在校验操作完成后再行动。

18.4.5 近期历史

到了20世纪80年代晚期和90年代初期时，个人电脑病毒已经成为一个很严重的问题，从而产生了防病毒软件编写和防病毒顾问这个全新产业。许多人认为它不会持续太长时间，因为随着DOS向更优秀的例如Windows这样的操作系统转变后，将会解决这一问题。一些防

病毒先锋们甚至卖掉了他们的公司，他们其中的一位在［720］中讲述了他的故事。

但是，解释性语言的传播给欺骗提供了更加肥沃的土壤。这里引起公众惊慌的主要是20世纪90年代后期的恶意Java小应用程序（Java applet），因为人们发现了在浏览器中攻击Java实现的方法，这在相当程度上增加了人们对安全的注意［537］。但是到了21世纪初，主要的病毒感染源存在于Microsoft公司产品，例如Word中的宏语言，而主要的传输机制就是因特网。业界分析声称，是网络保留了防病毒产业［423］。另外一种观点是病毒并没有给我们带来什么威胁，因为用户总是希望共享他们的程序代码和数据，而且在没有可靠的计算平台时，我们能够期望恶意代码之类的东西可以用来开发它们使用的共享机制。还存在一种观点是Microsoft应该负责任，因为它们不顾后果地在诸如Word这类的文字处理应用程序中引入了强大的脚本功能。正如他们所说的那样，你的利益可能会因此发生变化。

无论如何，Word病毒成为了1996年美国病毒的主要感染源，而且随后，在其他一些国家［57］，也出现了类似的情况。到2000年时，宏病毒几乎在所有因移动恶意代码而发生的事故报告中出现。一个典型的宏病毒就是一个宏，它将自己复制到受感染机器硬盘中那些没有被感染的Word处理文档中，然后等到用户共享文档时进行繁殖传播。有一些病毒变种在复制方面表现得更为活跃，例如，它们可以将被感染文档当作邮件发送到被感染机器地址簿中的每个地址处（在［128］中有关于宏病毒的讨论，它指出阻止宏病毒要比阻止DOS病毒困难许多，因为Microsoft编程环境目前的开放程度很低，缺少必要的文档支持，而且也十分复杂）。

过去，恶意代码的问题不值一提。一个有趣的例子是关于David Mazieres和Frans Kaashoek的，他们在MIT开设了一个匿名回复邮件系统。这个设备破译从网上任何地方发来的消息，将它们解压，再进行处理。有些人发给他们一连串25 Mb的消息，而其中仅仅是重复再重复的一行文本，这些消息压缩得很好，密文也被压缩得很好，但是当进行解压缩的时候，它们迅速地填满了spool文件，使得系统一下就崩溃了［531］。还有其他利用解压缩来对别的应用程序进行攻击，例如MPEG解密器。然而，最过分的案例所包括的是恶意代码而不是恶意数据。

18.4.6　防病毒措施

从理论上讲，防御其实很简单：如果从防火墙过滤掉Microsoft可执行文件，那么你就可以将大多数危险因素排除在系统之外了。但在实际生活中，并没有那么简单。一家拥有85 000名员工的加拿大公司正是这么做的，但是许多员工都有基于Web的电子邮件服务的私人账户，所以当爱虫病毒出现时，病毒作为网页进入了公司系统中，而没有通过防火墙的邮件过滤器。这家公司的系统在配置邮件客户端时，使得每位员工在地址簿中都拥有公司所有职员的地址信息。结果就是系统随着85 000个邮件客户端中的每一个都试图向其他85 000个客户端发信而彻底崩溃。

一种病毒的横行传播需要自我维持，它需要传递一种称为传染极限的参量。在这种极限下，其复制速度将超过移动速度［452］。这点不仅取决于病毒本身的传染性，而且有赖于它们所感染的互连机器的数量（和比例）。医学上流行病的模型和病毒传播在某种程度上是相似的，虽然后者是受软件交互的不同拓扑结构所限制（软件共享是高度局部化的操作），所以预计的感染程度总比实际发现的要低。一堂医学课程将大多数有效的措施集中给予讲解和

汇报，而实际中只是有选择地使用某些疫苗［453］。

在与病毒斗争的过程中流传下来一套管理规则。在 DOS 文件病毒时代，这意味着要控制所有加载到组织机器上的软件，而且对所有的事故提供一种集中汇报点。既然病毒主要通过邮件附件或者网页动态内容传播，有必要将它们过滤到防火墙外。另外，也要看用户对于软件默认的设置值是否采取谨慎的态度，例如不允许使用浏览器中的活动功能和 Word 中的宏功能。

事情的本质是需要培训用户，从而告诉他们什么需要做，什么不能做，而且还要随着系统和攻击形式的不断变化而变化。例如，20 世纪 90 年代中期，主要的工作是阻止病毒感染那些用于家庭的个人电脑，它们或者用来工作，或者用来做其他事情（例如小孩用来打游戏），说服他们使用单机扫描软件“扫清”所有到来的电子邮件和携带病毒的磁盘（后来出现了一种更有效的方法，它已被伦敦法律公司采纳，就是对于发现病毒的人奖励一盒巧克力，以促使发现病毒的人将被感染的文件寄送到公司）。既然现在典型的防病毒软件都已经包括像自动扫描和集中汇报等功能，它们还应该提供一些更加细微的功能，例如，告诉人们不要随便打开可疑邮件的附件，如何处理被感染的备份文件等等。但是和治病一样，防患于未然总比到时候再弥补要好得多，软件健康可以和软件控制集成在一起，从而控制非法软件复制和未授权的设备私自使用等。

18.5　入侵检测

典型的防病毒软件产品就是入侵检测系统的一个实例。一般来讲，一种好思路就是假设攻击将会发生，相对于试图去防止所有危险的发生而言，防止某些攻击类型并同时对其他种类的攻击进行必要的检测所付出的代价要小很多。这种用来检测危险事件发生的系统我们通常称之为入侵检测系统。前些章节中的示例是针对某种应用的特殊机制，用来检测移动电话复制、银行出纳员的欺诈行为等。某些股票市场已经安装了专门的系统通过查找可疑活动模式的方法来检测内部人士的交易活动。虽然，它们所进行的工作是那么类似，但是它们的开发者互相并不知情，所以看到系统一次次地被重新创建，这仅仅是重复性劳动，毫无意义。

入侵检测技术在公司和政府网络的安全研究领域中的发展相当迅速。例如，在 20 世纪最后几年中，在美国军方的资助下，几乎从无到有建立了数以百万计的入侵检测系统。这种增长已经通过实现来提示许多系统应该有效地使用日志和审计数据。例如，在 Sun 公司的 Solaris 操作系统中，我们发现在 1996 年，审计数据格式并没有文档支持，阅读它们的工具也没有出现。审计工具被安装似乎只是为了使政府系统的购买者满意的一种做法，而没有实际应用的价值。我们也期望对此有所增强，从而帮助管理员们无论在事前或者事后能够检测到攻击。

18.5.1　入侵检测类型

最简单的入侵检测方法是监听超过规定阈值时发出的告警。三次或者更多次的登录失败、信用卡的支出情况在最近的三个月中超过两次提取可使用钱款或者移动电话呼叫持续时间超过六个小时，这些情况都应该对相应的账户做上标记，从而引起注意。更加复杂的系统通常分成两种类别进行处理。

第一类叫做*误使用检测系统*，它使用的模型类似入侵者的行为方式。例如，在银行系统

中，如果某用户连续三天从提款机中提取超过最大允许提取数额的钱款。还有，在 Unix 的入侵检测系统中，以更加复杂的方式通过使用这种系统的某人寻找被接管的用户账户，因此那些原先只是使用一些简单命令的账户将在日志显示其使用编译器时发出告警。告警也许会被某个特定的行为触发，例如，试图下载密码文件等。一般来讲，大多数的误使用检测系统，像防病毒扫描器一样，查找一个签名，即一种特定攻击方式的已知特征。一种最常见的误使用检测签名就是对蜜罐陷阱很感兴趣，这是指某些为了吸引人注意的迷人的东西。例如，我曾提到过的，一些医院维护假的写有名人姓名的医疗记录，从而引诱那些对医院信誉持怀疑态度的患者就诊。

第二种类型的入侵检测策略是异物检测。这类系统试图使用一种更加困难的做法，该做法在没有明确攻击模型的情况下，寻找不规则或者是异类的行为模式。它希望以此来发现某些以前不曾被识别的攻击手段。这类系统通常使用人工智能技术，其中神经网络技术尤其流行。

误使用检测系统和异物检测系统之间的分界线并不是十分明显。由 Benford 规则提出的分类标准似乎更好一些，该规则描述了随机数字的分布情况。人们也许认为以数字 1 至 9 打头的数字应该均匀分布。但实际上，来自随机自然源的数字，它们的分布并不像数字系统中所表述的那样，而是一种对数分布：大约 30% 的十进制数字以 1 打头（实际上，如果禁止使用初始值 0 的话，所有二进制数字均以 1 打头）。那些搞欺诈的职员，捏造数据来伪造账目，甚至使用随机数字生成器来产生账目，但由于他们不懂 Benford 规则，就常常会被抓住[529]。

18.5.2 入侵检测的普遍局限性

有些入侵是十分明显的。如果你所担心的是那些脚本小子来搞乱公司的网页，那么最明显的一件需要做的事情就是在你的操作间中放置一台机器，每过一秒就取出网页来显示一次，一旦发现网页被改变就马上告警（要注意确保利用外部代理服务器做这些事情，不要忘记不仅仅是你自己的系统正面临威胁，脚本小子可以将主页上的广告代之以色情文学，此时你最希望的就是尽可能快地将此连接抹去）。

然而一般来讲，入侵检测是一项很难解决的问题。Fred Cohen 证明发现病毒（即决定是否某个程序会做出危害系统的事情）和阻止病毒发作同样艰难，这也意味着我们不能期待一个完全的解决方案[192]。

另一种基本的限制来自这样的事实，那就是目前主要存在两种不同类型的安全故障。一种是导致错误的安全故障（在 6.2 节中曾将其定义为不正确状态），而另一种则不导致错误。一个前者的例子是来自银行的窃贼，他对审计追踪进行跟踪。一个关于后者的例子就是被外国情报机构控制的一只无线电麦克风被安放在你的房间中而没有被发现。前者可以利用处理某些你可利用的数据的过程来检测（至少在理论上是这样的，现在先忘掉解决问题的事，只考虑如何发现问题就可以了）。但是后者就不能这样做了。不错的主意是在设计系统时，尽可能使潜在的威胁来自第一类故障，从而避免第二类不好检测的问题，但这种做法往往是不切实际的[182]。

还存在定义的问题。一些入侵检测系统被配置用来阻塞任何可疑的行为，而且，在最为极端的情况中，关闭被感染的系统，使之停止运行。使系统远离拒绝服务攻击的大门，同时

将入侵检测系统变为一种访问控制机制。正如前面已经看到的，访问控制通常是一项很困难的问题，它包括了各种各样有关安全策略的问题。而对于这些策略，用户一般不同意或者理解错误（通常有这种误解，认为可以利用入侵检测机制来做访问控制，而且所有的入侵检测可以使用神经网络来解决，所以在局域网中安放某些神经网络就可以强制实施类似 Bell-La-Padula 的功能。这是一种愚昧的想法）。

我喜欢将入侵检测系统定义为可以监控日志，而且对可疑事件可以引起授权机制注意的系统。这和移动电话操作员的工作十分接近。在金融调查中这将起到很重要的作用，参看[658] 中的讨论，利用一个特别的使用美国内部税收服务的代理，试图跟踪那些隐藏资产和收入的行为。许多都有赖于基于长期经验的怀疑。例如，一个 25 美元的账单将导致发现隐藏在某提名候选人身后的价值 250 000 美元的二手房产。创建一个高效的系统意味着有人、有机器，而且都在他们最擅长的岗位上工作，起到最大的作用，而且也意味着可以使用机器来做初步的筛选过滤。

错误的报警会付出一定的代价。例如，我过去常常在每年 5 月到旧金山去，习惯于在 ATM 提款机中连续五天使用英国记账卡，然后它就会停止工作了。这不但使用户十分沮丧，同时也会让坏人很快学会如何使用它（就像用户使用一样，为了使我整个旅途中都有钱花，必须首先取出够我前五天开销的钱）。和许多安全工程的问题一样，在欺骗代价和攻击代价之间的折衷方案是最为危险的。就像在第 13 章“生物测量学”13.8 节中讨论的一样，不可以期望通过查询许许多多不同的指示器来改善这种折衷方案。一般来讲，必须认为攻击者有足够的耐心可以通过阈值限制，他们或者攻击的速度很慢，或者进行大量的小型攻击。

利用商业入侵检测系统检测特殊的不可跟踪问题是被排除在外的。当保险公司要求使用邮政区号的统计数字来决定收取的额外费用时，许多贫困和人口稀少的地区将承担高额的费用，或者被干脆排除在服务范围之外。这在许多地区是一种不合法的行为。但是问题不仅仅局限于此。例如，华盛顿在航班中引入了向乘客介绍恐怖主义给航班带来的威胁，所以他们必须执行严格的安全检查。据美阿反种族歧视委员会报告，许多发生过的使无辜航空旅客心有余悸的事故也都证实了这些忠告 [516]。

一般来讲，如果所创建的入侵检测系统基于数据挖掘技术，那么面临的最严重的问题是如何区分这些数据。如果使用神经网络技术，那么根本没有办法向法庭解释你的决策来自何种规则，也就很难为自己辩护了。不透明的规则还将违反欧洲数据保护法，该法授予公民知道采用何种算法处理他们的数据的权利。

一般地，大多数不正规的入侵检测系统使用许多不同的技术 [661]。它们试图从应用程序中最大限度地汲取知识，而尽量减缓更新换代的周期。

18.5.3　检测网络攻击的特殊问题

现在转到检测网络入侵的一些特殊问题上，这些问题出于某些原因，要比检测类似移动电话复制复杂许多。对于初学者，可用的产品仍然不会工作得很好，也许它们在实验室中成功的几率达到 60% ~ 80%，同时还具有极高的错误报警率。例如，在写此书时，美国空军使用在局域网中开发的系统还没能检测到任何一个入侵，虽然一旦通过其他方法检测到某种入侵的话，会发现其依然在日志中留下线索。

对于如此拙劣的性能表现，以下提供一些解释，这里并没有按照特别的顺序排列。

- 因特网是一个非常“吵闹”的环境，并不仅仅是指在内容这一级别上，还指在消息包这一级别上。在任何真实存在的站点上，都会接到大量随机的脏信息，而它们中的许多都足够产生一个重大错误报警。根据［89］中 Bellovin 的一项调查，许多错误包来自软件自身的缺陷，其他的是一些过期或者被破坏的 DNS 数据，还有一些本地包，它们离开本地网络，到世界周游了一圈后又回到本地。
- 攻击太少了。如果在每百万会话中存在十个真正的攻击，对这个数字的估计几乎一定偏高，即使系统错误告警率低到 0.1%，那对于真正告警的错误率也将达到 100%。在第 10 章中对于防盗报警我也谈到过类似的问题。这就好似实习医师通过扫描程序发现那些艾滋病病毒超过生物体正常值的情况一样，太少出现了。一般来讲，当信号远远小于噪声时，告警系统疲于奔命，往往连真正的告警也会忽视掉。
- 许多网络攻击是针对某些软件的特定版本的，所以大多数攻击只是关心那些旧版本中才有的漏洞。因此，通常对于检测工具的使用存在失误，认为检测工具必须拥有一个庞大的、经常改变的包括攻击签名的库。
- 在许多情况下，商业组织购买入侵检测系统，只是把它当作一种应该尽到的责任，以示勤奋。这往往是保险公司和咨询机构最满意的结果。
- 被加密的信息，例如用 SSL 加密过的 Web 会话，像利用恶意代码对其内容进行分析或者过滤并不是一件很容易的事情。只是在理论上存在着在防火墙处停止加密，或者通过安装监听设备，其他用户可以共享你的密码。然而，在实际中，这几乎不可能［3］。
- 在防火墙中应用入侵检测的观点也有许多。你可以在包层进行过滤，它速度快，但是对付不了包碎片；你可以重建每个会话，这将增加计算量，所以并不适合于骨干网络中；或者你还可以检测应用程序的数据，这也是一项花销很大的工作，需要经常性地升级来对付新应用程序的出现。

虽然，到目前为止，美国空军利用本地入侵检测系统还没有发现攻击，但是通过网络统计的方法，攻击还是被发现了。柱状图保存了源地址、目的地址和端口的包信息。这是一种检测隐蔽攻击的强大方法，攻击者每天发送一个或者两个包到 100 000 台主机。这类攻击可能不会被本地统计所发现，将在噪声中被忽略掉，不会引起注意。但是当数据收集工作是通过一个大型网络来做的话，可疑的源地址就会很容易暴露出来了。

由于所有的这些原因，显然一个单一产品的解决方案是不太容易获得成功的。将来的入侵检测系统将会在不同的层次上融入许多监控机制，让它们协同工作。这些不同的层次包括网络（骨干网络、局域网和单机）和协议栈（包、会话和应用）。这并不意味着一个明显的划分，例如包过滤将被应用在骨干网，而应用过滤被应用在代理服务器上，大块关键词查询也许会在骨干网中完成（只要 IPsec 不会导致加密后通信量的失踪就可以）。

18.6　小结

防止和检测网络上的攻击，尤其是因特网上的攻击，也许是安全工程中最具有报道价值的方面。这个问题好像不会马上被解决，因为存在各式各样不同的漏洞和缺陷可以被攻击者加以利用。理想情况下，人们会在安全的平台中小心地运行编写的代码，而在实际生活中，情况并不总是这样。但是还是有希望处理这些威胁的：防火墙可以剔除那些最恶毒的攻击，

仔细的配置管理能够阻塞其他大部分攻击形式，而入侵检测可以抓住剩余的攻击企图。

由于攻击技术在很大程度上依赖于主要的软件供应商偶然在开发过程中引入的错误和缺陷，所以它们会经常改变。本章集中解释了基本的基础知识（当然这还相当贫乏）。尽管因特网已经连接了成百上千万的机器，它们运行着不安全的软件，而且也谈不上管理，那些攻击普通软件的脚本程序广泛地分布在其中，但所发生的坏事情几乎与几十年前的完全相同。一个新出现的事件就是分布式拒绝服务攻击，它出现的前提是被攻击系统与大量攻击机器互连。尽管如此，因特网也不是灾难。

也许对于成百上千万不安全的机器的一个合适的类比，就是将它们比作漫步在非洲草原上同样成百上千万的羚羊。狮子为了生存也很难抓到任何一只羚羊，大多数的羚羊利用数量众多作为庇护，可以生存许多年。而情况对于那些十分幼小、十分老迈或者跑在大部队前边贪吃草的羚羊来说就有些复杂了。因特网也是如此，其中也存在相应的猎手，它们仔细地寻找可作为战利品的动物，所以你也需要格外小心（如果你认为新闻界关于“邪恶黑客打倒了因特网”的恐慌与拥有卡拉什尼科夫冲锋枪的饥饿农民有些类似的话，那么应该记住的是，殖民地的农场主利用资金来大量圈地所造成的危害要更加厉害）。

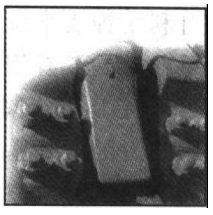
当然，如果你试图贪吃草料，或者不得不针对网络攻击保护那些极其关键的商业系统时，你应该阅读所有黑客的网页，调查所有值得关注的黑客软件，订阅邮件列表，阅读建议和安装补丁。虽然在很大程度上，攻击可以被避免，但类似的防御措施在过长的一段时间之后，也可能不会再起什么作用，形式主义地采取某些措施比起进行适当的操作来讲，对于降低危险程度没有什么作用。

研究问题

在学术领域中，研究正开始向入侵检测技术集中。一个有意思的课题就是通过和生物学类比，使防病毒软件产品更加聪明。IBM 研究识别和培养病毒的自动化技术，参见 [452] 可以带领你观看关于该项试验的整个过程。Stephanie Forrest 和新墨西哥大学的许多同事通过生成大量随机“抗体”来模拟一种免疫系统，然后消灭那些试图“攻击”系统自身的组织 [302]。不知这种与生物学类比的做法是否得当？也不知道这条路可以走多远？

参考资料

关于因特网安全，最著名的一本书是 Steve Bellovin 和 Bill Cheswick 合著的 [94]。另一本不错的书籍是 Simson Garfinkel 和 Eugene Spafford 合著的 [331]，该书对于许多网络攻击和系统管理方面的细节问题是不错的参考。Terry Escamilla 最近编写了关于防火墙的升级，以及对于入侵检测技术调查的书籍 [275]。Fred Cohen 编写了关于病毒繁殖的书籍 [192]，虽然这本书写于宏病毒成为主流问题之前，但还是值得一读。Java 安全的问题在 Gary McGraw 和 Ed Felten 合著的书 [537] 和 Li Gong 的书 [346] 中被讨论。John Howard 的论文对因特网中出现的安全事故进行了调查 [392]。如果你希望可以对新发生的事情有最新了解，那么 CERT 的建议 [199] 和错误报告 [144] 也是非常重要的阅读材料。此外，还有黑客网站 www. phrack.com 和 www.rootshell.com 特别值得一看。



第 19 章　保护电子商务系统

如果你违背市场，市场也将会违背你。

——玛格丽特·撒切尔

19.1　引言

对于电子商务系统的保护将涉及我们在前些章中讨论过的许多主题。失败来自访问控制的错误配置、具体实施上的严重错误、网络服务的窃取，以及对加密技术的不正当使用等等。在本章中，将讨论一些电子商务系统所特有的保护问题，例如，如何处理信用卡的在线支付等。还要讨论一些容易误入歧途的做法。

如果你是一名为 .com 的启动而创建电子商务系统的程序员，应该对本章中的大部分内容非常熟悉。你也许很希望从本章中获得关于访问控制、网络安全，以及银行业的一些知识。最有可能对电子商务系统进行的攻击并不包括因特网协议族或者支付基础设施中存在的漏洞，我们往往对这类漏洞束手无策。

对于启动一个典型的电子商务系统来讲，最大的风险来自内部欺骗。像复式簿记的做法由来已久，所以内部欺骗在传统商务系统中也是最主要的欺骗方式。许多已经启动的电子商务系统并没有特别的内部控制机制。开始时往往只是由极少数的几名彼此间十分了解的人创办，而随着成功地获得利润，很快又会雇用许多新人。这些新人只注重酬劳，对于他们的加盟并没有经过仔细地筛选。例如，据 2000 年 10 月的一项调查显示，37% 的 .com 执行官都有过不光彩的过去，而在传统的公司中由于在人员挑选上面的严格审查，这个数字只有 10% [257]。

19.2　电子商务的电报史

有许多阻碍电子商务发展的问题，其根源来自于一个普遍的观念，那就是认为电子商务是在 20 世纪 90 年代中期才发明的全新事务。这种认识是完全错误的。

各种各样可视化信号的传输技术在早些时候就已经被应用了。已经存在许多这样的系统，包括太阳摄影机（该设备在接收端利用镜子反射太阳光）、旗语（利用胳膊的位置移动来发出代表数字和字母的信号）和标记。在陆地上，可以通过一连串的烽火台发送消息，在海上的船只之间也存在类似的系统。这些系统最初只是由军方采用，但到了拿破仑战争以后，法国政府太阳摄影机网络被开放投入商用。很快，第一类欺诈行为就随之出现了。该骗局直到两年后的 1836 年才被发现，两个银行家向一个操作员行贿，让其利用传送信息时故意出错的方法，暗地里发送股票市场变化信息给他们，使得他们可以在一个安全的地方接收到这些信息。还有一些其他的方法被用来发送赛马结果的信息。虽然出台了许多不同的法律来制止这种行为，但并没有起到什么效果。对于赛马的赌注者而言，惟一的解决办法就是亲

自利用钟表来查看时间，而不只是被动地等待比赛结果，仅仅希望自己是第一个知道比赛结果的人是没有用的。

从18世纪60年代到19世纪40年代，电报技术被技术前驱们发明出来，这些人中最具影响力的就是塞缪尔摩尔斯。他劝说美国国会在1842年拨款资助了一条连接华盛顿和巴尔的摩的试验线路。这件事使人们印象深刻。此时，许多正规的贸易投资开始了，到1850年为止，已经出现了由20家公司分别运营的12 000英里的电报线路。这与20世纪90年代末期因特网的蓬勃发展情况惊人地相似［729］。

银行是电报业务发展之初的第一个庞大的用户群体，而且这些使用者们认为他们需要技术上的保护机制来防止不老实的操作员对事务交易的改变（本书在有关银行系统的章节中讨论过他们开发的测试码系统）。电报同样被用于创建国际市场。纽约的日用品商家第一次能够在几分钟内找出在芝加哥拍卖的产品价格。同样地，到达波士顿的捕鱼船的船长能够得到位于英国的格洛斯特市的鳕鱼的价格。近期的历史表明，电子商务的许多概念和问题都同维多利亚女王时代十分相似［729］。你如何知道在和谁通话？你如何知道对方是否可信？你如何知道货物是否能被送到，以及货款能够到位？在19世纪时，对于这些问题的答案牵扯到中介机构，这主要是指银行，它们通过使用诸如证明、保证书以及可以证明信用度的信函等方式帮助商家，以减轻他们所面临的风险程度。

20世纪60年代，许多国家的银行使他们的簿记机制计算机化，而且引入了国内多银行间可互操作的系统来处理对用户账户的直接支付业务，使得银行能够对企业客户提供诸如工资表的业务。正如在“银行业和簿记系统”中提到的，在20世纪70年代早期，这种业务被扩展到可以进行国际间的支付。电子商务的又一次扩张发生在20世纪70年代末到80年代中期，这是随着电子数据交换（electronic data interchange，EDI）的传播而发展的。从General Motors到Marks和Spencer等公司都纷纷创建系统，从而本公司的计算机和供应商们的计算机互连，使得定购货物可以自动进行。旅游代理处也构建了类似的系统来从航空部门实时订购机票。

1985年，苏格兰银行第一个提供了零售电子银行系统业务，该业务的用户可以使用英国电信系统的快速电视电话咨询服务来进行支付。当Steve Gold和Robert Schifreen攻击了该项服务后（就如同我们在第3章“口令”中所谈到的），新闻界和银行家们才开始陷入恐慌中，他们意识到黑客们能够很容易地获取和更改交易内容。但是一旦这些弊端被解决，人们通过仔细地分析细节性问题就可以得出这样一个结论，那就是实际的风险系数很低。因为，系统只允许在自己的账户和事先通知银行登记过的账户（你的燃气和电力供应商的账户）之间进行交易。

这种模式的攻击虽然可以导致极大的恐慌，但是经过冷静的分析之后就会知道，它们并没有真正成为一个严重的问题，所以到目前为止这种情况仍在继续。

让我们继续回顾这段简要的历史，在20世纪80年代后期和90年代初期，呼叫中心以很快的速度发展壮大起来。尽管人们把注意力都放到Web上面，但是在2000年，这些呼叫中心绝对是最大的从企业到消费者（business-to-consumer，B2C）的电子商务交付渠道。至于因特网，如果一个火星人一直监控我们的电视频道的话，它也将会相信因特网只不过是1995年突然出现的事务，不值一提。我第一次通过在线服务来出售自己编写的软件程序是在1984或1985年，第一次帮助警方调查一起在线信用卡欺骗案件是在1987年。在这起案件

中，罪犯从其在超市工作的女友处获取了近期被盗的信用卡卡号，然后使用这些信用卡从加利福尼亚的公司购买软件，再卖给他的客户。之所以可以这么做，是因为当时的被盗卡列表中包含的仅仅是那些在本国曾经被作为欺诈手段使用过的卡。这种做法也保证了银行不能使无知的客户成为借方。当这件事发生后，罪犯却在警方获取足够证据逮捕他之前外逃。这是由于一场暴风雨清洗了罪犯住宅对面的河堤，从而暴露了警方设下的监视点。

使用信用卡来通过电子手段购买商品突然在 1994 年或 1995 年时成为主流，这时也正是大量用户上网之际。突然，出现了许多诸如因特网系统不安全，信用卡号供应数量应该更多，以及必须引入加密技术的呼声。

19.3　信用卡介绍

在 20 世纪 50 年代发明信用卡后的很多年中，信用卡被大多数银行看作是应该引起高额客户注意的最容易遭受损失的业务。最终，在大多数国家里，商家和持卡人的数量非常大，但交易数量却减少了。在英国，花了 20 年的时间才使银行看到这种业务有利可图，然后突然一下子又变得利润可观。信用卡系统已经成为网络支付系统中极其重要的方式之一了。

通过大量的投资，终于产生了对成千上万的银行、数以百万计的商家和世界范围内数以亿万计的用户构成竞争力的系统，而任何新型支付系统的建立都必须经历一段时间，因为可能出现意外情况，对此我简要说明一下。当你用信用卡在商店购物时，这项交易的事务流将从商家流向银行（商家所要求的银行），银行在扣除所谓的商家折扣后，这一般是 4% ~ 5%，再对商家进行支付。如果信用卡是另一家不同的银行发行的，那么交易事务下一步将流向交换中心，该中心由生产信用卡的注册商（例如 VISA）维护运营，中心扣除佣金后再传递到发行信用卡的银行来获取支付。在银行和信用卡注册商间的日常支付解决了网络现金流通的问题。发行信用卡的机构虽然可从商家的利益中获得一小部分商家折扣，但是相比从扩展信用卡持卡人数量给它们带来的利润来讲，这类银行间互操作的利润要少得多。

19.3.1　欺骗行为

由于信用卡丢失引起的欺骗，其带来的风险一般都是由热卡黑名单（hot card list，挂失信用卡列表）和商家分层限制系统进行管理的。每位商家获得一个本地挂失信用卡列表（原先写在纸上，现在则一般存储在主机中），再加上发卡银行对卡进行的授权限制。商家还可以通过呼叫中心或者在线服务的方式来访问一个国家级的热卡黑名单。再向上还有更高一层的限制机制，商家还可以和信用卡商标注册商联系，商标注册商拥有世界范围内该品牌信用卡的完整列表。再往上还有更高一层限制，交易事务可以被一路返回，直到信用卡发行机构来进行最终验证。

20 世纪 70 年代，在邮件订单和电话订单（mail order and telephone order，MOTO）业务的使用中，商家不需要消费者亲自到现场，当然也不可能检查其信用卡。在这种情况中，用什么方法可以消除那些通过信用卡号购物，然后又导致收条作废的欺骗行为呢？

银行是通过以下办法来降低风险程度的，例如，作为口令的有效期、降低商家分层限制、增加商家折扣以及强调到持卡人住址处交货等，最后一条通过在授权认可阶段来检验核实。但是所带来的主要变化就是将责任都转移到商家身上，使他们的风险程度大大增加。如果你使用一项在线信用卡交易（或者实际上也可以是任何类型的 MOTO 业务），所有的交易

金额与重要的处理费用立即被划到商家名下。不论这一借贷行为是欺骗、争论还是回报，这个过程都是一样的。

当然，即使是让持卡人到现场，也不能保证就此杜绝欺骗行为的发生。多年来，大多数的欺骗行为都是利用偷窃来的信用卡亲自作案的，而且受到过沉重打击的商店都纷纷趋向于销售那些易于保护的商品，例如珠宝和用电设备。银行对此只能采用降低商家分层限制的方法。最近，随着技术保护机制的发展，又出现许多有关信用卡的欺骗事件，这些信用卡甚至从没有到达过真实用户的手中。这类预发行欺骗包括从预批准信用卡的电子邮件中偷取，这些内容往往使用垃圾邮件寄送，甚至使用确实存在和信誉卓著的人名来创建应用，而这些人却并不知道这些应用的存在（*身份偷窃*）。作用于系统上的这些攻击从本质上说，仅仅使用技术手段是很难对付这些攻击的。

19.3.2 伪造

在20世纪80年代早期，电子终端被引入，通过它，销售人员可以刷卡并自动获得一个授权信息。但是这种销售支取方式仍然是从压纹磁条处读取信息，所以欺骗者可以计算出如何重新将账户和有效卡的期满时间编码到被盗信用卡的磁条中，有效卡号往往在高级餐厅的垃圾箱中就可找到。一个重新编码过的卡将被正确授权，但是当商家提交银行汇票索取支付款项时，账户与授权码不符合（一个六位数字，一般是由账户、有效期和金额共同加密产生）。商家无法获得支付款项，必然强烈抗议。

针对这种漏洞，银行在20世纪80年代中期引入*终端支取捕获*的方法。在这种方法中，销售支取通过使用信用卡磁条中的数据自动打印出来。欺骗者的应对措施是伪造出许许多多信用卡，这里面有许多都是由Triad这帮人伪造的。在1989年到1992年间，磁条伪造事件从原先偶尔发生麻烦事件发展成为几乎占到所有欺骗事件一半[6]。VISA对此则采用*信用卡验证值*（card verification value，CVV）机制，通过卡条内容（包括账户、版本号和有效期）计算出来的3位MAC被写在磁条的最后部分。它们最初工作得很好，在1994年的第一个季度中，VISA卡在全世界的欺骗损失降低了15.5%，而同时万事达卡的相应值则升高了67%[165]。结果后来，万事达卡采用了类似的校验和机制。

对此，欺骗者则又使用*快读法*，这种方法是通过使真实用户的信用卡刷过一个特制的、没有经过授权的终端的方法来获取磁条信息的拷贝，然后再将其重新写入一张真正的信用卡中。银行则相应采用入侵检测系统，该系统首先通过对比用户先前的购买历史来试图标识出犯罪交易。

在20世纪90年代后期，信用卡欺骗事件的数量由于另一种简单的犯罪技术的创新而突然猛增。该技术是在快读磁卡信息的欺骗手段成功后，只是获取用户交易的金额而不用它来消费。你在黑手党开的餐馆吃过一顿饭，提供了信用卡，签署了凭单，而且费用没有出现在账单上也没有引起你的注意。也许是一年之后，突然出现了一笔购买珠宝、电子产品甚至是赌博筹码的大额账单。到那时你早已完全忘记了这顿饭，而银行也没有该记录[318]。

19.3.3 自动欺骗检测

因此，在20世纪90年代人们做了大量的工作来加强入侵检测。有许多常用的系统被用

于异常事务的检测，它们使用的技术诸如神经网络，但该技术究竟可以得到什么效果还不明确。当欺骗已经呈减弱趋势一年之后，这些技术却作为最新的欺骗探测系统成功出现 [61]。当欺骗的数量在几年后有所增长时，那些入侵检测系统的制造商会让问题悄然过去的 [714]。

更加具有吸引力的项目是由特定的连锁店采用的，用来查找一些已知模式的滥用手段。例如，纽约地区的电子产品连锁店发现罪犯描述（包括年龄、性别和种族等等）不够有效，而换为采用购买描述，从而在一年中将欺骗行为的数量减少了 82%。这种技术不仅仅怀疑那些高额交易，同时也要告知全体职员当顾客购买商品时不很细心，以及仅使用很少的时间来询问产品质量和性能问题时更要特别留意，这些都是欺骗者爱犯的错误。这些因素也可以被在线监控，但是纽约的连锁店获得成功的一个很重要的因素在网站中很难实行，那就是雇员的奖励。银行对于抓获的每一个信用卡欺骗给予 50 美元奖励，但许多商店仅仅知道遵守而已，所以店员并没有真正努力去发现假卡，或者不愿在顾客面前产生一种尴尬的局面。在纽约，一些商店的员工往往因发现假卡而每周获得 150 美元或者更多的奖金 [525]。

作为处于销售循环之外的网站设计者，对于恶意者惟一应该采取的是一种平和的心态。建议每一个电子商务站点都应该提供一种价格贵得不合理的“白金”选择，当然很少有真正的顾客会去购买 [721]。但此举可以起到两个作用。首先，它允许你从根本上了解购买状况。第二，它适合于由在线经济学家 Shapiro 和 Varian 提出的金发定价模型（Goldilocks pricing），他们指出，在航空系统中提供头等舱票价的真正作用在于推动商业舱机票的销售，它可以使旅行者有足够的理由说服老板（或者他们自己）来购买商业舱机票，因为没有坐头等舱，所以他们还是很节俭的 [696]。另一种思路是对于可疑的交易应采取经过仔细策划的应对方案：如果你仅仅说“这是坏卡，试试别的卡吧”，那么欺骗者将来还会进行欺诈活动。你甚至还可以没收信用卡，使这些骗子不能再用这些卡来行骗，告诉他们这些卡是最近挂失的信用卡，如果再流通将会造成银行业务的混乱（即使银行应对系统设计负一定责任）。一种更好的办法是告诉他们，货物已经脱销，这样，欺骗者就只能到其他地方去了 [721]。

19.3.4　经济学

统计学有选择性的分析结果表明，目前存在许多关于信用卡欺骗行为的错误观点。举例来说，VISA 在同一天中既可以报告信用卡欺骗事件的数量有所增长，又可以报告有所下降 [380]。

然而，从商业出版物中可以发现关于这些数字的一致性模式。信用卡欺骗的实际代价，在最近一次有所增长之前，大约占所有通过 VISA 和 MasterCard 进行的国际性交易额的 0.15% [652]，同时各国的比例也不尽相同，美国 1%、英国 0.2%、法国和西班牙 0.1%。各国流行的商业文化对于这个比例也有很大的影响。例如，美国银行更希望通过大批邮件的形式发行很多预批准卡，从而增加他们的客户基础，同时也产生了不可避免的预发行欺骗，这也是这种方式所付出的代价。在其他国家，银行更倾向于规避风险的运作。

看上去法国比较有意思，乍一看，会出现一种很例外的情形，一种特别的技术确实带来了真正的收益。在 20 世纪 80 年代后期，法国银行引进了芯片卡来处理所有的国内交易，在 1987 年减少了占营业额 0.269% 的损失，1993 年是 0.04%，1995 年是 0.028%。然而，目前跨地域欺骗的数量在增长。法国欺骗者使用国外的信用卡，特别是英国的卡 [315，652]。

而同时用法国芯片卡在那些不使用芯片卡的国家的商家消费［166］。但是，在欧洲，损失削减最多的情况并没有出现在法国，而是出现在西班牙。在西班牙，政策使得商家分层限制减少到0，使所有的交易都在线进行。这使得损失从1988年占营业额0.21%削减到1991年只占0.008%［73］。

这些教训说明，首先，卡欺骗是循环的，随着新型防御措施的出现，欺骗者将学会如何对付它们；第二，最复杂和昂贵的技术上的解决方案在安全领域中并不一定工作得最好。

19.4　在线信用卡欺骗：大肆宣传以及现实情况

我们现在将目光从传统信用卡欺骗转向在线欺骗上来。在20世纪90年代中期存在着极大的焦虑，认为在因特网上使用信用卡将导致极多的欺骗行为的出现，那些可恶的黑客可以截取电子邮件、Web表单和上百万用户所获取的信用卡号。这些焦虑使得各家银行和软件供应商设计出两套协议来保护基于Web的信用卡事务。这就是安全套接层（Secure Socket Layer，SSL）和安全电子事务（Secure Electronic Transactions，SET），我将在下一部分对它们进行解释。

对于这类欺骗行为的大肆宣传做得太过分了。对于电子邮件的中途截取确实是可能的，但这在实际中也是相当困难的事情。这种难度和政府迫使ISP们在它们的网络上安装监听设备来更加容易地进行合法监听的难度不相上下［114］。但是，由于这类设备的花费过高以至于ISP们尽可能地抵制来自政府的压力而不予安装。我将在第21章中对此做进一步的介绍。虽然，利用像DNS缓存中毒的欺骗手法可以将比较流行的网站主页重新定向到你的个人站点之上是可能的，但还是要比试图从老式电话系统获取监听容易很多，没有人会为了从旅馆客人那里获得少量的信用卡号而过度担忧。

信用卡号码在网络中确实可用，但这常常是由于某些商家的被攻击的计算机在用户支付完成后保留了用户的信用卡号码而造成的，这种做法不符合银行颁布的明令禁止的标准做法（当这本书出版时，VISA宣布，从2001年开始，它的所有商家都必须遵守10项最新的安全规定，例如，他们必须安装防火墙，保证应用最新的补丁程序，对于存储和传输的数据必须进行加密，以及经常性地升级防病毒软件［752］）。同样地，基于Web的欺诈交易确实发生了，但主要是由于系统的拙劣实现所导致，而该系统应该在授权期间对持卡人的地址进行核对。摆在.com们面前的真正问题还在被争论着。

很容易就可以拒绝一项交易。基本上，用户所需做的一切就是打电话告诉信用卡公司，“我不批准这项交易”，然后，商家对账单只能自己承受了。这种做法在几乎所有的信用卡事务都在本地进行的年代里是可以工作的，而且大多还是数额十分巨大的交易。但如果客户欺诈性地拒绝一个交易，商家可以将他们告上法庭而迫使他们使用本地信用卡委托机构。除此之外，银行系统也通常具有足够的能力来对本地持卡者的地址进行验证。

但是因特网与老式的邮件订单或者电话订单的体制存在很大差异，在这种情况下，许多交易都是国际性的，数额通常很少，通过信用卡系统核实一个国外地址就存在疑问了。通常，所有呼叫中心的操作员可以做的就是当把地址读入特定国家时，确认该商家是可信的而已。因此，那些逃脱处罚的拒绝交易的情况屡见不鲜。这类事件特别对提供色情服务的站点来讲具有更高的发生几率。毫无疑问，一些争吵就此开始，当在某些冲动的影响下做出的交易出现在家庭信用卡账单上时，持卡人为了维持自己的婚姻只得拒绝交易。其实这里面有许

多都是行家搞出的恶作剧。

在写本书时，据媒体报道，联邦商贸委员会起诉许多成人网站的运营者，这里面包括 playboy.com，起诉他们对理应提供的免费服务向用户收费。骗局就是在网站提供标明“免费”的服务，要求提供信用卡号码，按理应该只是核对用户的年龄是否超过 18 岁而已，但实际上随后会通过各种方法对使用者收费。一些网站甚至对一些从来没有访问过他们的用户收费 [389]（当然，这对于那些利用传统电信欺骗手段进行诈骗的研究者来说并没有什么感到奇怪的，因为这只是对传统欺诈方式的一种新型伪装而已）。如果甚至像 playboy.com 这类大型和“名望很高”的站点也纵容这种做法的话，那么很容易就可以使消费者采用欺诈性的拒绝交易的方法来逃脱处罚，从而获得侥幸成功。

对于在线业务，很重要的一点就是，如果用户对于占总交易额一定比例的交易产生质疑时，那你的利润将逐渐消失，甚至在极端情况下，银行会剥夺你的信用卡获得权。据报道，运动服商家 boo.com 的破产是由于优惠过头了：该站点的业务模式假设不存在不确定的交易以及退款政策。但是他们发出的货有许多尺寸不对，或者颜色不对，或者根本不能吸引客户。最后，由于信用卡惩罚措施导致其破产 [721]。

这段历史告诫我们，技术上的弥补措施不像许多供应商所声称的那样简单，主要的资源都将有其本来的程序。美国快递宣布，它将为其用户提供一次性的信用卡号码，这将对于来自其他人的攻击起到保护作用。为了不至于用光数字，它们将通过其网站来为用户一次发行一个号码（这将导致大量的网络流量）[204]。许多其他的银行家已经得出结论，地址验证是未来的发展方向，而不再是加密技术 [62]。

然而，如果你是一名安全工程师，那么许多客户都希望谈论技术问题，例如加密信用卡号码，所以我们无论如何也要讨论一下可用的加密机制。

19.5　密码保护机制

银行用来对卡业务进行保护的密码机制目前有以下几种：个人身份号码（personal identification number，PIN），被用于自动柜员机（Automatic Teller Machine，ATM）和某些销售点的终端上；信用卡验证值（CVV），在 19.3.2 节中介绍过，该技术使伪造假卡变得十分困难，但对于在线业务来讲没有什么作用，所以又出现了一些新机制。最为广泛使用的是安全套接层（Secure Socket Layer，SSL），它是一种被绑定在 Web 浏览器中的加密系统。

19.5.1　安全套接层

回忆一下公开密钥加密体系，服务器发布一个公共密钥 *KS*，任何 Web 浏览器可以发送一条包含信用卡号码的消息 *M* 到该服务器上，并利用 *KS* 加密：$\{M\}_{KS}$。虽然在实际实现中有些复杂，但是可以利用 SSL 完成。SSL 可以用来支持双向加密和验证，所以 http 请求和响应均可利用 SSL 来防止被偷听和被操纵。

下面是将 SSL 应用在保护询问信用卡号码的网页时的一个简化的描述：

1）客户端发送一个 client hello 消息到服务器，该消息中包含名字 *C*，事务序列号 *C*#，以及随机 nonce N_C。

2）服务器发送一个 server hello 消息，该消息包含名字 *S*，事务序列号 *S*#，随机 nonce N_S，以及包含该服务器公共密钥 *KS* 的证书 *CS*。现在，客户端将证书 *CS* 与公司发布的根证

书进行核对，例如 Verisign，并且存储在浏览器中。

3）客户端发送一个 key exchange 消息，该消息包含 pre-master-secret 密钥 K_0，这条消息通过服务器公共密钥 *KS* 加密传送。还要发送一个 finished 消息，并携带消息认证代码（message authentication code，MAC），这个代码是通过到目前为止发送的所有消息计算出来的，MAC 的密钥使用 master-secret 密钥 K_1。这个密钥是通过对 pre-master-secret 和客户端与服务器发送的 N_C 和 N_S 值进行哈希处理得出的：$K_1 = h\ (K_{CS}, N_C, N_S)$。从这点开始，所有消息都是加密传送的，如果是从客户到服务器方向，我们将它写成 $\{\cdots\}_{KCS}$ 的形式，如果是从服务器到客户方向，我们将它写成 $\{\cdots\}_{KSC}$ 的形式。这些密钥的产生是通过对 K_1 进行哈希计算得出的。

4）服务器也发送一个带有 MAC 的 finished 消息，该 MAC 通过对到目前为止所有的消息进行计算得来，然后就可以开始发送数据了。

$C \rightarrow S$：C，$C\#$，N_C

$S \rightarrow C$：S，$S\#$，N_S，CS

$C \rightarrow S$：$\{K_0\}_{KS}$

$C \rightarrow S$：$\{\text{finished}, MAC\ (K_1, \text{everything_to_date})\}_{KCS}$

$S \rightarrow C$：$\{\text{finished}, MAC\ (K_1, \text{everything_to_date})\}_{KSC}$，$\{\text{data}\}_{KSC}$

SSL 的设计目标包括最小化浏览器的负担，然后还要最小化服务器的负担。因此，公开密钥体制的加密操作是利用客户端完成的，解密操作由服务器完成；标准的加密算法（*ciphersuite*）使用的是 *RSA* 算法，该算法中加密过程比解密过程快许多（这是一种错误的设计决策，因为浏览器一般都比服务器拥有更多的计算周期，*RSA* 算法的使用已经使得加密部件市场活跃起来，大家都在想方设法让 *Web* 服务器的加密速度快起来）。还有，一旦客户端和服务器端建立起一个 *pre-master-secret*，公共密钥的操作就不再需要了，因为后来的 *master-secret* 可以通过对新的随机 *nonce* 值进行哈希处理获得。

一个完整的 *SSL* 协议要比这复杂得多，而且已经经历了多种版本。它支持许多不同的加密算法，例如，出口的浏览器版本被限制为 40 位的密钥，这是美国政府多年来强加到出口许可上面的一种规定。其他加密算法，例如，*Diffie-Hellman* 密钥，用来支持短期密码交互，它可以提供双向保密性。*SSL* 也可以进行双向认证，如果客户端也有证书，同样可以被服务器端进行核对。除此之外，*KCS* 和 *KSC* 密钥均可以分别包含子密钥，用于加密和认证。例如，最常见的是使用流密钥 RC4 进行加密，HMAC 进行认证，由于加密算法不同，所以就需要分别使用不同的密钥。

虽然 SSL 的早期版本有许多漏洞［784］，但最新的版本（被微软叫做 TLS）看上去不错（但是必须小心地实现［116］）。它正被用于除电子商务外的其他领域，一个例子就是医疗隐私，它有可能取代私人网络，而允许保密的患者数据通过因特网传送［175］。SSL 也通过网络应用进行传播，现在它已经被融合到 Win2K 中成为一个选项了，可以利用它在不同域的机器之间建立安全的会话。但是也存在一些问题，应该更多地关注证书的种类和管理，我们将在 19.5.3 中进行讨论。

19.5.2　安全电子事务（SET）

信用卡和因特网早期的使用经验表明，真正危及信用卡号码安全的危险事件并不是来自

对 IP 数据流的监听，这点虽然没有直接的事例可以证明，但从对商家 Web 服务器和其他终端系统的攻击中，我们看到，这些服务器和终端常常保留用户的信用卡号码，这才是最经常被攻击的。所以，在 1995 年到 1996 年间，曾经做了许多努力来开发一种更好的支付协议，它使用数字签名机制而不是信用卡号码。

最终，一个包括微软、Netscape、VISA 和 MasterCard 在内的多家公司的联盟提出了一个名为安全电子事务（Secure Electronic Transaction，SET）的协议。该协议的主要思想是：

- 用户也将拥有公共密钥证书，而不仅仅是商家拥有。所以用户能够签署交易事务，其中包括向他们的银行提交付款委托书。
- 用户签署和加密两个单独的消息。其中一条消息发给商家，这里面包含对货物的描述和价格，但是不包括信用卡号码；而另一条消息发给银行，这里面包含价格和信用卡号码信息，但是不包括货物描述信息。签名信息被加在后面。
- 后台的事务处理，包括从获取银行到商标注册部门再到信用卡发行银行，使用已经存在的系统就可以了。

SET 应该让用户重新获得信心，从而认为在线交易可以安全地进行，而且大大减少了由于欺骗行为所带来的损失（主要是通过拒绝对商家提供信用卡号码）。就业务模型而言，SET 交易就好像持卡人在场一样，这里，银行将担起承担欺骗行为风险的责任，而商家折扣也将被减少。

由于还必须支持数量众多的、不同类型的现存系统，而且来自不同行业角色的需求也不同，SET 甚至比 SSL 还要复杂很多。对此，我还是做一个简单的说明。

首先，用户将她的证书 CC 发送给商家服务器，包含用户自己的公共密码 KC 和一个临时的随机 nonce N_C。服务器用一个已被证明过的公共密码对商家（CS，KS）和商家所属银行（CB，KB）回复，再外加一个事务序列号 $S\#$，然后，用户发送一条消息。该消息包括被商家公共密钥加密过的订单，以及被银行公共密钥加密过的支付指令。再对二者进行哈希操作后，签署上用户的私有签字密码。下一步是授权，可以在线执行或者延期执行，怎么做适当就怎么做。服务器发送支付指令以及一个订单概要到获取银行，该订单概要不包括货物的详细描述，而仅仅是应支付价格而已。银行对此进行审核，并且如果需要的话还要参考信用卡发行银行的意见。如果一切正常的话，它发送给服务器一个授权响应，这和传统方式（包含交易额和授权码）类似，并增加一个签名。

$C \rightarrow S$：C，N_C，CC

$S \rightarrow C$：S，$S\#$，CS，CB

$C \rightarrow S$：$\{Order\}_{KC}$，$\{Payment\}_{KB}$，$sig_{KC}\{h(Order), h(Payment)\}$

$S \rightarrow B$：$\{Summary\}_{KB}$，$\{Payment\}_{KB}$

$B \rightarrow S$：$sig_{KB}\{Auth_response\}$

SET 似乎到了该标准化的程度了，但是市场方面却无法取得成功，下面列举了一些有指导意义的原因。

- 首先，采用 SET 所获得的好处似乎没有预期得好。许多大商家违反了与持卡人之间的协定而将用户信用卡号码保留，这在最初只是作为买卖数据库中的索引来加以使用，但却有望用于其他方面。所以，由此所产生的一条特征就是商家可以从获取银

行处得到信用卡号码。这种做法被认为打消了许多安全性能改善的希望。(实际上，事情并没有那么糟糕，银行可以发行专供 SET 事务使用的信用卡号码，所以偷窃它们将毫无意义。)

- 第二，采用 SET 花费过高。创建一个公共密钥基础设施，并对于所有发行的信用卡持卡人附加公共密钥证书，这种做法的花费将十分巨大。性能也将成为问题。
- 第三，SET 中不包含任何用户信息。如果他们不高兴，在网上利用 MOTO 规则进行交易的用户可以随时撤销一个交易。这不光因为支付问题，而是因为服务问题、产品问题和其他的问题。使用 SET 将转变到好似持卡人就在现场的规则当中，在许多国家其实已经去除了这种保护机制。因此，除非用户情况更糟，以至于极愚蠢地使用 SET，否则不可能对此感兴趣。还有，安装 SET 往往需要下载上兆字节的 SET 钱包，而且还要经历一个费事的认证过程。

最终，SET 因花费过高而收效甚微，就用户而言，这是一场灾难。现在该技术正悄悄走向消亡。也许，最主要的教训就是为电子商务设计系统时，应该从现实的角度来处理事务，而不是从理论的角度来处理事务，而且要考虑你的设计如何才能不光引起客户的兴趣，还可以引起领导的兴趣。

19.5.3 公共密钥基础设施 (PKI)

公共密钥基础设施 (PKI) 仍旧是一个问题。经常在“公共 (密钥基础设施)”和“(公共密钥) 基础设施”两个词语上产生语义冲突。首先，基础设施可以被任何新出现的应用所使用，我把这叫做开放 PKI。其次，是那些不能被使用的，我把它们叫做封闭 PKI。

关于开放 PKI 的例子有：

- 使用 SSL 的商家应该拥有他们的公共密钥的证书。而且，在做出适当的、正当的努力后，一些诸如 Verisign 之类的公司将证明某个公共密钥确实属于某家公司。
- 有许多提议和想法都是基于新型在线服务的，尤其是商家到商家的服务，将基于被认证的数字签名服务。
- 许多政府考虑给他们的公民配备公共密钥证书，也许采用智能卡的形式，作为下一代身份证使用。虽然大多数的商业买卖仅仅关心用户是否付钱，但政府还是提供一系列服务 (例如税收和福利)，这些服务可以被那些能伪装成多个人的欺骗者所蒙蔽。所以，许多政府更关心如何促进 PKI 技术的使用，这将导致法律的制定，我将在第 21 章对此进行讨论。

关于封闭 PKI 的最著名的实例存在于由军方代理处和诸如 SWIFT 等银行业务提供者所运营的网络中。在这里，虽然也使用非对称加密系统，但是不公开任何密钥。既然 Win2K 包含 SSL 作为一种认证机制，可以通过有许多分散站点的公司建立一个安全的、范围广泛的网络。而且，如果站点数量过大，还可以让公司自己通过操纵 PKI 管理密钥。所以封闭 PKI 将会变得越来越普通，即使那些对大众提供的使用非对称加密的服务，也可能不公开公共密钥。一个例子就是 Mondex 电子钱包，它使用 RSA 加密算法，进一步地保护了存在于那些已经很难被篡改的智能卡中的密码。

在写这本书时，PKI 已经成为保护技术中发展最快的技术之一。然而，该技术仍存在许多内在的限制，大多和该项技术所提供的公共服务可以被任何人使用有关。我已经在第 6 章

中讨论了许多潜在问题。命名就非常困难，如果某个证书上说“Ross Anderson 有权管理 foo.com 机器”，那就意味着世界上其他叫 Ross Anderson 的人无法拥有这项证书了。

一种解决命名问题的办法是，对每一项商务活动都使用自己的封闭 PKI，这被认为是在系统的级别上为每位用户提供了一个独一无二的账户号码，而且不与其他用户共享。这就导致了“一个密钥或者多个密钥”的争论。我们是否应该期望利用一个单一的签名密钥来代替每一把金属钥匙、信用卡、swipe 访问卡和其他我们通常带在身边用以表征身份的东西呢？或者这些身份表征中的每一样都由一种不同的签名密钥所代替呢？第二种选择对于商务来讲更方便一些，因为第一种做法将引起大量的管理花费和可靠性问题。这种做法也保护了用户：我不想使用一个甚至可以抵押房产的密钥来打收费电话。而且利用一台其他的显示设备，很容易就可以骗我签名一条消息，从而得到我的密钥（我不知道如何做才能确信甚至是我在自己的计算机上制作的数字签名，我从事安全工作超过 15 年。核对所有存在于显示和签名软件间的关键路径中的软件是一件极其麻烦的工作）。但是，支持 PKI 的机器发展迅猛，甚至提供了一种电话簿的电子替代品，它试图设想我们每个人都将拥有自己独一无二的名字和独一无二的密钥。这其实就是一种开放 PKI 架构。

这会导致一些政治上的问题，例如，我们可以信任哪些证书管理机构（CA）以及为什么可以信任？政府已经做出了各种努力尝试来给证书管理机构发执照，并且强制引入了法律手段可以介入 CA 工作的“后门”规则。政府大都期望世界范围内“单一密钥适用于所有情况”的模型。开放 PKI 对于网络经济的发展来讲也可能是有利的，这将在 19.6 节中讨论：一旦单一 PKI 占据主导地位，由于每个人都必须使用所带来的压力将导致类似垄断的保护（这也就是 VeriSign 股票市值如此之高的原因所在）。

关于具体的实现还有很多细节问题。例如，主要的证书格式（X.509）还没有具有像已发展多年的“热卡”系统所具有的灵活性和全球化可升级性。它只是设想依赖于证书的任何人都可以从证书发行机构下载证书撤销列表。这是一件既繁琐又无聊的事情。更好的用来管理证书撤销的办法也已经被提出，问题是它们能否被具体实现。还有，X.509 设计时就具有验证名字的功能，因为出于许多目的，人们都希望得到证明授权的功能。

下面还列举了其他一些有关证书的局限性：

- 大多数使用者都将其浏览器的安全特征设为不可用状态，即便当软件安装时默认值是可用状态也是如此。回忆一下 SSL 协议的第三步，客户端浏览器将得到的证书与自身存储的根证书进行核对。如果核对失败，浏览器将向客户端询问权限后再继续，但是由于大多数的浏览器在设置时都屏蔽这项功能，此时浏览器仅仅是继续执行而已。这使得很多电子商务站点可以通过使用过期证书或者自己签署的证书来达到省钱的目的。大多数的使用者对于出现的警告框不予理睬（并且也不知道如何对它们进行处理）。
- 主要的证书提供商将公司的名字绑定到 DNS 名字，但是该提供商并不是公司名称和 DNS 名称的授权机构，而且可以因此而拒绝所有应承担的责任。
- 证书市场的竞争被在微软的 Internet Explorer 软件中加入根证书的需求所阻碍（VeriSign 的股票持有者把它当作一个特征而不是漏洞）。
- 即使你获得了一个有效证书，但公司名称和/或者 DNS 名称也可能和你预先设想的有出入，因为网站建立和信用卡获得都是通过其他途径获得的。

- 美国的出口规定意味着大量的站点使用的是脆弱的加密机制。对于 SSL 安全的最近一份调查表明，8 081 个不同的安全 Web 服务器中 32%不属于美国。这都是因为各种不同的原因造成的：密钥位数太短、脆弱的加密算法，以及过期证书也是一个主要的原因 [567]。

在消费者保护法中也存在严重的问题。那就是，假定数字签名确实破坏了签名者的权利，在使用纸张的系统中，欺骗行为的风险就是由某些依靠签名的人引发的。在没有这种假设的情况下，对于持卡人来讲，他购物的地点是否具有包含适当名字的合法证书并没有什么区别。而且，无论如何也没有一种很方便的方法来记录使用者的交易情况，从而显示出其勤于交易。我将在第 21 章深入地讨论这些问题。

简而言之，虽然公共密钥机制在某些应用中是有用武之地的，但是并不像鼓吹它的人所说的那样，对于所有安全问题都有所帮助。这些鼓吹者并没有理解一些真正重要的问题。

19.5.4 电子数据交换（EDI）和 B2B 系统

在 19.2 节中我们给出的早期电子商务的例子，例如电报和电子数据交换（Electronic Data Interchange，EDI），主要都是 B2B 系统。这些系统中提供了许多实际应用的 PKI 的例子。我们在 9.3.1 节中看过一个很详尽的例子，即从 20 世纪 70 年代中期开始使用的 SWIFT 网络，该网络可以在不同国家的银行间传送安全可靠的支付消息，并且在 20 世纪 90 年代时升级使用公共密钥技术。实际上，这种结构也被其他许多系统所采用，包括用来注册英国资产净值和支持银行与经济人间份额交易的 CREST 系统。

一个离我们更近的 B2B 系统例子是 Bolero，这是一个应用于欧洲的系统，用来处理货船装载的货物的电子账单 [262，458，492]。这些电子账单是关于协商船货所有权的有法律保证的文件，它们平均的价值大约为 25 000 美元，但是，还要依赖于石油的价格，一艘油轮中货物的价值可以达到 100 000 000 美元。当货船在大海上时，货物经常会被交易多次，许多交易者都是不可靠的皮包公司。除此之外，对无赖行为制裁的强制性也表明，国家情报机构常常需要介入利用皮包公司来购买石油这类不应该发生的事情。所以，这是一个高价值，同时也是高欺骗性的环境。确实需要一些注意来保证船货账单不能被复制，而且这些账单的所有权历史记录也可以被建立起来。

Bolero 使用了两种主要的保护机制。第一，是防篡改性。IBM 的 4753 或者 4758 型加密处理器卡被用来处理账单，而且还使用一种包含数字签名机制的协议来传送这些账单。第二，是集中注册表。它用来保证独一无二的持卡人的名字可以在任何时候被确定。还有一个注册授权机构（专门审查组织机构的可信度）和证书授权机构（签署被注册机构批准的个人公共密钥）。

另一个例子是保健 EDI 系统，它一般用于发送化验或者检查结果，例如从大医院和实验室发送到家庭医生和本地诊所的 X 光片和细胞化验结果等。这个系统需要的不是不可复制性，而是真实性，再加上保密性。早期系统在提供这类服务时使用一个封闭的私有消息网络。而更多的现代系统则是在消息网关中使用加密和认证机制。每天有许多从医院实验室发到一般性医疗单位的消息被批量处理，并签署了医院的签字密钥，再利用那些一般性医疗单位的公共密钥进行加密，最后通过邮件传送。理想情况下，这些消息应该签署上将对患者进行治疗的以及书写评价的顾问医生的个人签名。实际应用中，做到这点很难，因为实验室和

医生使用各自私有的系统，而这些系统只能通过使用 EDIFACT 网关进行通信。这种网关是专门用来进行格式转换的，对于每个人来说，将它们运行在个人计算机上代价太高了。这种设计引起许多问题。例如，如果一个标有实验室数字签名的记录报告需要作为是否因工作疏忽而被患者提出诉讼的证据时，那么整批记录都必须被保存。而这么做又与隐私规则相冲突，隐私规则规定当患者死亡或者离开时，必须销毁这些医疗记录。

这种问题不光只出现在保健系统中，许多系统都有消息处理功能，从而导致了事务的一部分数据被发送到机构中不同的目标处。保护结构化数据完整性的任务要比看上去复杂许多。

所以，由于某些原因，实现 B2B 系统的安全通信并不那么简单。随着创新型的企业模型的扩散，一定会出现需要付出很高代价的设计性错误。我将在第 3 部分中讨论如何减少这种可能性。

19.5.5　电子钱包和微支付

在 20 世纪 90 年代前半期，许多电子钱包系统发展起来。其思想就是利用某些可以离线工作的机制来代替借款卡，从而避免容易被伪造的问题。典型的实现中包括用户和商家，他们都有芯片卡，每张卡中有一个价值计数器。另外还要加上卡到卡的支付协议，通过该协议，两张卡可以相互证明，一张可能借款，而另一张贷款。在 2.7.1 节中我们已经描述过这个协议的设计。

这种方案的支持者原先抱很大希望。例如，Europay Austria 确信在三年中，它所生产的产品将代替 20% 的现金交易。但是，到了 20 世纪 90 年代末，结果却令人失望。即便是将电子钱包的芯片集成到标准的银行卡中，而且像在比利时的 Proton 和德国的 Geldkarte 那样，面向整个用户群体发行，其使用率也没有什么起色（这项业务的调查见 [763]）。考虑到信用卡的历史，电子钱包起步时的缓慢也许早应被估计到。像信用卡一样，电子钱包也在遭受被经济学家称之为网络客观性的问题，它是指使用卡的数量越多、接受卡的商家数量越多，那么这些卡就越有用处。而不仅仅是那些客观存在的用户，还有那些潜在用户也包括在内。但是，当一小部分人使用电子钱包时，商家们却并没有购买终端的想法。而伴随着这些少量的终端，也反过来只能吸引少量的用户。最多，在这种方案达到众多使用群体之前，也还要有很长一段路要走。在诸如收费电话等应用的推动下，它们也许最终会成功，然后将以真正快速的速度发展。我将在 19.6 节中进一步讨论其潜在的经济规律。

在网络客观性的作用下，一种新型的支付机制在起步时总是缓慢的，但也存在一个例外，那就是*微支付机制*，它是随着第三代移动电话的第二个发展阶段而出现的，这在 17.3.4 节中已有讨论。因为所有的下一代电话的购买者都将使用这个系统来支付电话费用和一些其他的增值业务，增加一种新型业务的额外费用很低，而且在少数几年中用户基础应该也很稳固。这项业务同样也具有风险性，就像 Geldkarte 一样，最终以一个配置广泛而实际用户寥寥无几的系统收场，但是它也同样有可能成功（基于电话的支付方式也会出现一个很严重的问题，很可能在第三代电话中加入运行 Java applet 的能力，但随之而来的是那些恶意代码出现的潜在性，还会引入特征交互以及一般性混淆的问题）。

如果证书授权机构可以通过其他服务中使用的方法来签署用户公共密钥证书的话，那么可以想像，第三代电话将提供对全球范围公开 PKI 更近距离的接触。另一方面，如果顶级公共密钥协议是 Royal Holloway，而且（如同在已建立的政府使用的那样）由第三方保管签字密

钥和私有密钥 [50]，那么它将没有多少用处，因为系统所支持的签名所体现出的证明性价值将被破坏。

19.6 网络经济

上节提到的网络客观性并不仅仅针对试图使用新型支付系统的人们，而是具有更加广泛而重要的影响。该客观性支持的许多通信系统及其附带的应用都要受到它的制约和影响。其影响范围不光是电子商务系统模型，还有许多安全工程中必须对付的其他方面的问题。

一个很重要的发现就是，系统中用户越多，与它们交互的人越多，则系统对每一位用户来讲就越有用（这有时被称作梅特卡夫定律）。有许多有记载的例子都可以说明对于市场早期的开拓者而言，实际的反馈都具有极大的帮助作用，甚至可以使其最终成为一种垄断行业。例如，20 世纪初期，AT&T 在长途电话通信领域的主导地位使其可以打垮一些本地对手，从而几乎成为美国范围内该项业务的垄断提供商，这种情况一直持续到 1984 年才被打破。另外，兼容性也是十分重要的。20 世纪 50 年代，在 CBS 和 RCA 之间进行着一场争论，其焦点就是关于彩色电视机应该使用哪一家的标准。虽然 FCC 认可 CBS，但其标准并没有对数以百万计已经安装的黑白电视机提供向后兼容性，而 RCA 却提供了，所以最终 RCA 胜出。

对于正面反馈的一个最为明显的作用就是，一旦网络的规模超过了某一临界点，它将以迅猛的速度发展。电报、电话、传真机，以及最近出现的因特网都遵循这种模型。另一个作用是，对于那些最先开展的业务而言，将得到最大的回报。对于传统行业，例如汽车行业中，在某种程度上就是这样的。材料供给方的经济状况良好，将帮助大型汽车公司越来越壮大。但是也有局限性：一旦这些公司变得过于庞大了，例如像 General Motors 这样的公司，在与像 Toyota 这样的暴发企业的竞争中反应速度就不够迅速。而对于网络而言，其经济的规模是由需求方决定的，所以在发展空间上几乎没有什么上限。

网络影响不局限于那些被经济学家称做真实网络的传送光电信号的一类网络。它们也应用于虚拟网络，其中最著名的例子就是软件。回忆一下 20 世纪 80 年代中期在 PC 和 Mac 之间的竞争：一旦 PC 的用户数将超过 Mac 的状况趋于明显，软件公司就会将注意力集中到首先开发 PC 类的产品上面，而 Mac 产品的开发只能往后放。这意味着，将有更多为 PC 制作的软件，所以人们也更愿意购买 PC，实际的反馈继续起着作用。

这一规则不光局限于 PC 的体系结构，对于包括应用软件和文件格式在内的各种层次都适用。一旦人们大多使用微软的 Word 软件来编写文档，那么其他人就有更多的理由使用 Word，尽管他们将可能面临 Word 宏病毒所带来的危险。如果不用 Word，那么你的文件格式将很难被其他公司的程序阅读，所以又必须经常进行格式转换。在这种情况中，网络影响就显著加强了。我会在下一部分讨论这个问题。

暴露在网络影响下的一个市场特征就是占据。或者是技术占据，或者是供应商占据。技术占据通常包括补充技术提供商，正如微软在与许多软件供应商联合的情况下，终于打败了苹果公司。这样做的一个副作用就是，那些成功的网络对于补充技术供应商们的吸引力要大于对使用者的吸引力。而这些使用者正是应被起诉的破坏性应用的潜在创造者。一旦，客户在补充技术供应商那里真正投了资，他也就被锁在里面了。Andrew Odlyzko 发现，微软的软件和因特网中大多数用户易用性的欠缺是由于微软和因特网都是通过吸引开发者而取得成功

的。微软强加给用户的支持费用，以及等待个人电脑启动和关机所浪费的时间，都远远超过了他们的营业额［595］。

所以，关于信息技术市场存在三种特别重要的特征。

- 首先，技术通常具有很高的固定成本和很低的边际成本。一块芯片或者一个软件包的首次拷贝将投入很大的花费才可以生产出来。但是，后来的拷贝所需付出的花费就很低了。这点并不是信息市场所特有的，在航空和酒店市场中也是如此。在所有这类市场中，边缘价格趋向于降至制造成本价格，而在信息市场这种成本价格是 0。
- 第二，使用交换技术的用户常常花费巨大，这种技术导致占据。这类市场将保持高收益的状态，尽管它的边缘成本很低。
- 第三，通常就是前面讨论过的网络客观性因素。产品对于用户的价值取决于有多少用户采用它。

所有这三种特征的影响导致出现“胜者为王”的市场结构，从而也出现一些占统治地位的公司。确实，所有的公司都试图通过各种不同的方法，利用这些影响来获得具有竞争力的优势。

例如，一个通常的策略就是区分价格（differentiated pricing）。这意味着对于产品或者服务的标价不是基于其实际价值，而是基于对用户的价值。这与飞机旅行的情况有些相似：你可以花费 200 美元坐经济舱飞越大西洋，花费 2000 美元坐商业舱，或者 5000 美元坐头等舱（如 19.3.3 节中说明的那样，金发定价模型也分为几个等级：设立头等舱的主要功能是迎合需要节俭的坐商业舱的人的需求）。这种业务模型在软件和在线服务领域应用十分广泛。一个基本的程序或者服务可能是免费使用的，但同时其相应的功能更强的版本则需要支付一定的费用才可使用，而那些“黄金”服务更是需要高昂的价格。通常，程序都是相同的，只是有一些功能对于普通用户来说是不可使用的，而对于“黄金”用户来说就可以得到高质量的热线服务了。你将会遇到的许多保护机制都将维护这种差异性作为它们真正的功能。

另一种策略是操纵交换费用。对于 ISP 来说，你的账户所产生的长远效益，从某种意义上讲也可以说是被打折的未来收益，应该等于该用户转向其他竞争者的全部费用。所以，ISP 会想方设法使用户转向他们的过程更加容易，但还是很难让用户们转变。这点适用于存在着统治地位公司的市场中，这类公司董事们希望垄断整个市场，而其他竞争者试图打破这种垄断状态。这些占市场统治地位的公司的董事们试图增加用户转向其他竞争者所需要花费的代价，或者通过间接的方法，例如控制销售渠道，创建补充技术供给商工业；或者通过直接的方法，例如使得系统与其他系统不兼容，而且很难被反向工程破译。其间，许多市场新人试图来做这种反向工作：他们寻求各种方法来重用互补产品和服务的基础部分，从而破译产品原先内置的保护措施。一旦成功，他们就可以使用极低的销售价格补偿用户转变所需花费的费用。而占市场统治地位的公司的董事们相应地通过设计雾件（vaporware，一种已经被宣布但还没有生产的新型软件）来使用户感受到转变的机会成本。

随着技术不断向前发展，即使表面看上去无懈可击的垄断也可能被代替，所以，竞争也并不是那么纯洁。保护机制可以产生广泛的用途，从防篡改设备到私有加密算法。

19.7　具有竞争力的应用和公司间的冲突

经过讨论使我们看到，具有信息安全机制的应用，其目标不是用来保护用户及其数据，

而是用来保护或者攻击垄断行为。

有时，这些被利用的机制，其目的是很明显的。例如，游戏控制台的制造厂商试图封闭其平台，使得他们可以垄断其附属零件的销售市场，还可以强加给游戏软件供应商一些附加的条件。而这些条件中不光包括版税，甚至还有排外协定。对于诸如版权问题的相关法律解决办法还很不够，例如在许多国家的版权法中甚至没有针对某些公司通过反向工程来生产兼容产品的条款。所以只能通过加密的质询—响应协议来鉴别真正的零件和游戏卡带，竞争者则建立专门从事反向工程的实验室来发现密钥。在2.2节中我还提到过对附属物进行控制的其他应用。

另一个例子来自Microsoft Passport。对于这个系统，其公开的用途只有一个：拥有护照的用户对其访问的每个站点不必分别记忆单独的口令，分别记忆口令往往只会伴随着争论和风险。取而代之的是，一些站点使用Microsoft Passport系统来让用户登录，这个系统是一个由微软运行的中央认证服务器。网站的服务器使用Web的重定向功能将携带护照的访问者重定向到这台中央服务器上，认证请求和应答通过用户浏览器的加密cookie功能在它们之间传送。到目前为止，这个系统运行得还不错。

但是，护照所具有的真正功能是很隐蔽，而且很狡猾[727]。首先，把护照附加到所有被访问网站的Web事务中，微软能够收集大量关于在线购物习惯的数据，这可以让使用护照的站点交换信息。这种重定向和cookie机制意味着，实际上，你在所有支持护照的网站上进行的浏览会话变成了单一的会话，而这个会话被微软管理。如果每个站点都可以和其他站点交换数据，那么这些网站所组成的网络的价值相当于这些独立网站价值总和的平方，这存在着一种强大的网络客观因素。所以，这样的网络一定会处于统治地位，而微软希望拥有这样的网络。第二，在商家和护照服务器之间使用的认证协议是Kerberos私有的，这意味着Web服务器必须使用微软的软件，而不能使用Apache或者Netscape的软件。简言之，护照控制Web服务器和购买信息市场，但不是一个安全的产品。它与诸如Hotmail这类的服务（该服务的使用者已经有4千万）捆绑在一起，平均每秒就要对400个用户进行身份验证。护照机制的一些已知的缺陷包括微软保留了所有用户的信用卡详细信息，这可能将成为一个很大的攻击目标；各种可能的中间人攻击；用户可能被偷取到cookie文件的其他人假扮。护照有一个称为“注销”的工具，可以用来清除某个商家特定的cookie，所以，对于共享电脑的用户来说，这样做可以降低一些风险。但对于使用Netscape的用户来说，该功能并不能正常工作[473]。

在维护垄断和破坏垄断、分裂市场和控制市场之间总存在着竞争，这些竞争决定了许多使安全工程师的工作更加困难的外部环境条件。标记语言XML使得文档内容可以被很容易地处理，而且有能力加入具有保护属性的复杂语法结构[46]。然而，它的发展并没有像许多人希望的那样，因为如果网页变得很容易被机器阅读，那么相对应的购物数据也可以被很容易地创建。除此之外，许多在线商家可能要求，对于每个第100项交易就随机打折销售。如果购物者使用工具不停点击网站，直到获得便宜货物为止，对此，上面的技术就无法正常工作了。一般来说，一方面，从Web缓存到匿名通信服务被称作中间媒介，该媒介的设计者希望使用各种不同的方法控制用户事务；另一方面，商家网站又希望打破这种控制，从而直接“拥有”用户。在这两者之间，始终存在着竞争。

我将在22.6节中继续讨论网络经济对于安全所产生的影响。

19.8　还有什么其他容易出现的问题

对于世界上第一家在线银行 First Virtual 的调查表明，最典型的问题是一个烦人的用户凌晨 3 点打来电话，和韩国的某人谈论一笔丢失的应付款项。调查机构一般会发现导致该事件发生的原因在于因特网协议实现中的错误，或者输入邮件地址时有误所造成。对于解决这类问题目前还不是金融机构的核心职能 [129]。

大多数问题出自无法事先预知的漏洞和失误的模式，在其他应用领域也可能出现，而且还会发展下去。许多大型在线商店，包括 Buy.com、Staples.com 和由 Amazon 资助的工艺品零售站点 eZiba.com，都曾经由于价格错误而遭受打击。在 Buy.com，一张商家的优惠券本应该限于只能从任何大于或等于 500 美元的订单中减去 50 美元，而实际上却可以在任何情况下都节省 50 美元，这样一来，那些等于或者小于 50 美元的货物对于使用优惠券的购买者来说就等于免费的了。在 eZiba.com，每位顾客被提供 20 美元的优惠券，但是人们只要登录多次，就可以多次使用这种打折了 [671]。像这类错误并没有什么新鲜的地方，在过去有许多类似的例子，大都由于提供特殊优惠但又存在设计上的缺陷而造成，或者对于流行程度估计不足所引发。在网络中，一个不同之处在于，一个错误很快就会被广泛传播，从而导致严重的损失。

还有一些针对特定系统的其他有意思的攻击。*无线电数据系统*（Radio Data System，RDS）增加了一个数据频道来播放无线电信息，所以接收器能够识别是哪个电台发来的信号，以及正在播放的内容是何种类型。这使得，当你正在驾驶时，收音机可以为你喜爱的网络自动切换到信号最强的发射机上。你还可以让收音机在其他频道出现交通信息时随时中断你正收听的节目（一个 RDS 收音机有两个调谐器，其中一个总是用来扫描频带来寻找高强度或者高优先权的信号）。盗版电台采用伪造交通信息的办法欺骗用户，所以很多汽车收音机都自动选择它们的频段收听 [306]。如果盗版电台播放的是与你刚才收听的电台播放的同一类型的音乐时，这种欺骗很难被发觉。而且真正的电台也没有什么明显的认证机制，也许 FCC 应该作为证书授权机构来发放具有频谱许可的证书来鉴别真伪了。但是，如果一个真正的电台变成盗版后又如何去废除它呢？保护频段分发的完整性是必要的。从更一般的意义上来说，有趣的新型特征的出现往往成为好奇的探索者攻击的对象。

计算机游戏是另一块容易发现攻击的领域。一个例子就是 Quake——一款分布式游戏，它的源代码也是公开的。许多玩家通过修改源代码来改变 Quake 客户程序，从而实施欺骗 [633]。这引发了许多问题，尤其是关于电子商务的应用，这些应用将 Java applet 或者其他代码部署在具有潜在危险性的客户机器上。同时也没有一些明显的措施来使 Java applet 进行自我保护，以抵制虚假的骗局。最近出现了一种谨慎的策略，就是开发站点的非 Java 版本。这不仅意味着你可以和像我这样的人做生意，而我出于安全的原因关闭了浏览器的 Java 支持项；还使你获得反馈机制来阻塞基于恶意 applet 的欺骗行为，而不用关闭整个站点重新开发。

19.9　商家能做些什么

一般来说，我给那些关心电子商务风险性的人们提出的建议是，这些风险性与常规业务和信息技术的风险性没有太大区别。

作为商家，你应该确保开发者理解商务模型，使用结构化的开发方法，全面测试代码，不要自作聪明，查找一些可以被采用的已经被证实的思路和想法，尽一切办法来通过防火墙控制内部系统与外部网络的通信，尤其注意内部控制机制，从而阻碍、防止并检测到内部欺骗行为的发生。还应该知道，无论你做什么，事情也有可能恶化，或者必须被很快地加以改变。要记住：不要太懒。如果你将“我们保证在线购物不可能使你丢失钱款”写于主页上，并用小字印刷体写上“我们有关于所有交易的单独的和权威性的数据”，那么总有一天，消费者权利节目的电视工作者们会光临的。

上面的话都是说给公司 IT 部门主管的。电子商务所表现出的不同之处在于（至少到2000 年中期时是如此），用户信用卡交易会带来更多的风险因素和更大的花费。对此，并没有什么明显的解决办法，但是可以使用一种对破坏起到限制作用的策略，那就是创建可被控制的一套流程，用户在遇到问题时通过它可以提出申诉，而不是直接找到信用卡公司解决问题。例如，你可以打出一条广告，准许兑换打折邮资，如果用户确实希望退款时，让他们填写并打印出表格，并通过普通邮寄的形式寄给你，而不是通过在网上提供表单的形式达到相同目的。

总而言之，要集中看一下商务风险问题，同审计师、保险公司和董事们商量后有规律地升级风险管理文件，要比应用一些最新的安全技术小发明重要得多。商务毕竟还是商务，仅仅由于那些高薪聘用的计算机专家设计出成为老员工们工作学习内容的商务流程，是不能改变商务本质的，所以这些措施并不意味着事情会向好的方向发展。与“网络精神”相联系的风险必须被正确地评价和管理。

最后，有必要注意一些非技术的问题，例如产品责任问题。美国电子商务所享有的一个很大的优势就是美国法律不会强迫执行来自于国外的判决结果。所以，美国的电子商务部门不必担心来自其他遥远国度用户的投诉，即使当地法律允许用户提出诉讼也是一样，因为，判决结果不可能被强制执行。但是在欧洲，存在着国际互惠协定（international reciprocal agreement）：一个英国零售商被希腊当地法院起诉，其判决结果将被英国强制执行。从另一方面来说，美国法院也可以处理并给予惩罚性的损失赔偿；而欧洲国家的这种做法花费巨大，甚至对于小事情也会处理，以至于产品责任问题的案件很少发生。这些考虑与信用卡问题相结合，但并不是包含在其中。它们可能非常复杂，我要说的就是你应该获得法律上的建议：开展业务的最佳地点有赖于你卖什么东西和卖给谁。

19.10　小结

在线企业面临的大部分问题与其他机构面临的问题没有多大区别。而且，来自网络安全的威胁也与传统企业面临的安全问题没有太大区别。对于电子商务来说，真正增加的风险性与传统行业的风险管理机制不能从本地具体的业务向世界范围的抽象业务平滑过渡有关。信用卡交易的复制就是目前最主要的例子。对那些正在高速发展的公司来说，已经雇佣了大量新员工，但是并没有传统的内部控制机制，这也是一个很重要的潜在性风险。

研究问题

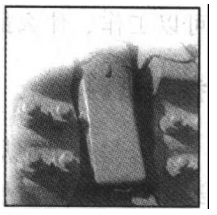
已经被投入具体应用，或者即将被投入应用的对于电子商务保护机制的研究工作大都是在 1994 年至 1996 年完成的。现在正是发展它的第二波浪潮的好机会，我们已经看到了什么

可以工作、什么虽然可以工作但在市场中却只会失败，以及还存在哪些实际的问题。

参考资料

电报的早期历史可以参见 Major General RFH Nalder [569]，而 Tom Standage 讲述了电报在维多利亚时期被快速应用的故事 [729]。关于有组织的信用卡伪造的调查参见 [592]。关于 SSL 的官方标准不容易阅读，可以在 [604] 中找到更好的说明。有关 SET 协议，参见其首席设计者 Li Song 所著的一本书中的描述 [509]。公共密钥认证和基础设施的问题在 [42, 268] 中被分析。最后，我所知道的关于网络经济的最佳著作是由 Carl Shapiro 和 Hal Varian 所写的 [696]。

第 20 章　版权和隐私保护



DeCSS 事件几乎就是一个先兆，它预示着我称作“电脑空间中的审查制度”的战争的到来。依我看来，这种审查制度不是针对色情、新纳粹、炸弹设计、亵渎上帝或者政治矛盾等话题，而是在数字化控制善恶决战的战场上，在过去和未来两大势力间进行的真正死亡竞争，而争夺的焦点就是版权。

——John Perry Barlow

很高兴看到你的个人电脑处在不安全状态。这说明在你购买电脑后，你能够进入电脑系统，安装各种你想安装的软件。是你想安装的，而不是 Sony、Warner 或者 AOL 想安装的。

——John Gilmore

20.1　引言

将支持版权和隐私问题的技术机制放在单独一章中介绍是出于对许多原因的考虑。

从政治的层次上来说，上面 Barlow 的话中所暗指的冲突确实存在。在 William Tyndale（英国剑桥大学出版社的创始人之一）因印刷英文版圣经而被绑在火刑柱上烧死的事件发生之前，对于信息的控制已经快成为政府所关心的核心问题了。这种对于印刷品审查制度的敏感性在 18 世纪继续横行，甚至从更近的一些学说中还会看到关于控制某民族及其竞争对手之间信息空间的战争发生。在最近的几代人中，关于文学、电影和音乐的版权所有者所创造的巨大财富在能够控制的情况下已经形成了另一种强大的影响。

从系统的层次上说，版权和审查制度都属于访问控制问题，所关心的都是针对特定组中的某些人，需要限制其对某些信息的访问权。在版权的情况中，组中包括那些已经支付某些费用的人；而在审查制度的情况中，就需要其他评判标准（例如，年龄是否超过 18 岁或者是否为非新加坡公民，或其他标准）。有时，二者也会有重叠的部分，例如通常对于使用具有“年龄核实”服务功能的信用卡的用户提供在线色情信息是有限制条件的（假定所有持卡人年龄都在 18 岁或者 18 岁以上永远无法起到限制作用，只能引起“变化环境”类型的安全故障）。一般来说，用户真实姓名是很重要的：因为如果身份都不能确定，那么对煽动性言论、版权侵害和诽谤行为所应负的负责也就变得不确定了。

隐私在很大程度上也是访问控制的问题。它用来限制能够知道你的私人情况的人的数量，例如你和谁交换电子邮件，你读什么书，以及你听什么音乐等。理论上，并不存在强制性的理由来决定什么人应该受到限制，而且在大量电子产品出现之前，通常也不需要什么隐私。版权是通过小规模数量的副本价值来保护的。购买一本书或者一张唱片要比做一份单一的拷贝简单和便宜得多，所以那些进行大数量非法拷贝的人将被追查并被起诉。随着影印机和盒式磁带录音机的出现使拷贝方式产生了巨大变化，但拷贝所带来的价格障碍并没有改

变基本的经济规律。所以，书、唱片和影碟可以用现金购买，也可以进行二手交易。但是转到数字世界中的时候，情况就变了。虽然，也存在一些诸如付费电视（pay-TV）之类的系统，这种系统依赖于物理防篡改的设备，大部分版权控制重点都向注册的方式转移。一旦你购买了一套软件产品，就应该作为一个用户进行注册，这种商业模型也扩展到其他可能破坏隐私的媒体中。

在本章中，我将从技术的角度来看待隐私问题。保密性意味着出于对第三方的一种义务而将信息保密，而隐私是指控制信息分发的一种能力。到目前为止我所讨论的隐私应用中，两者似乎有些重叠。例如，医疗中的隐私是通过强加给医生保守秘密的职责来实现的。但是在本章中，我所关心的是用来直接保护自身隐私的机制。这些机制包括从加密电子邮件到在线假麻醉和对于文件系统的匿名网络访问不予回应等等。

从技术的层次上说，在版权和隐私之间存在的压力很大。视频和音轨没有得到物理上防篡改的记号的保护，在理论上就可能被拷贝和共享，但这种方式不可能进行大规模的正规交易，而且对于版权所有者不用支付任何费用。无论对 ISP 施加什么压力来削减像 MP3 这类音频文件的流量，那些无法追踪的通信系统的存在也会使这些努力白费。另一方面，许多已经存在和被提议的电子分布式系统使得加密内容可以被随便使用：为了对其解密，用户必须联系一个服务器来购买一个密钥，这也意味着必须提供你的姓名和地址。这说明存在许多“信息泄漏”，由于提供商只需一个中央许可服务器便可确切地知道谁购买了什么许可权以及什么时候购买的。商家认为这主意不错，但那些希望保守秘密的人就会因此胆战心惊了[260]。

除此之外，出现了许多其他的技术可以解决版权和隐私问题。数据隐藏技术能够被用来在数字视频中不可见地嵌入版权标记。这也可以被用在隐写术（steganography）中，隐写术是将消息隐藏在其他消息中。看一下发生在家庭中的情况，在你发给兄弟的电子邮件中也许包含了从你最喜欢的乐队的最新 CD 中窃取的音轨（它们还将包含组织证明者来监督国际交易讨论会的消息，所以政府的兴趣永远不会离我们远去）。

20.2　版权

对电影、音乐和书刊出版行业来说，版权保护是一件很麻烦的事情（一提到这些，就会联想到全体做假的事情，以至于被计算机界人士称作“好莱坞”）。但是，版权问题并不是伴随因特网而出现的。在很多国家都存在着长期和激烈的争论，争论的焦点在于是否应该对空白的音频或者视频卡带收税，而这些税收又进一步分发给版权所有者。这个问题并没有被限制在电子媒体领域。在英国，一年当中，可以公开借阅的图书馆中外借书刊的所有作者一共可以分得几百万英镑版权费用[629]。回到 19 世纪，还存在着由于照相机的发明而导致书刊出版业出现恐慌。而在 16 世纪，移动印刷电报机被具有很大权力的诸如王子、主教和专业协会等，认为是一个具有很强破坏力的发明。

目前有许多工作都以电子版权管理系统（electronic copyright management system，ECMS）为基础。到目前为止最具重要意义的是收费电视（pay-TV）。我们已经看到了收费电视系统的防篡改机制，而且其中的一些协议以失败告终。我注意到，这类系统更具挑战性，因为攻击者能够购买足够多的访问卡来分解和研究。但是，在我们对高技术系统产生担忧之前，先来看看软件保护问题，因为，在过去的 20 年中，大多数版权问题都出自个人电脑和游戏软

件市场。

20.2.1 软件

对于早期的电脑来说，软件是由硬件提供商或者那些写软件的人免费提供的。IBM 公司在 20 世纪 60 年代甚至建立起关于其用户可以共享该公司编写的程序的方案（这种方案中的大多数都是没有什么用处的，因为这些程序针对性太强，而且也没有多少文档支持，或者很难被别人改写）。所以，保护软件版权并不是什么问题。当时，有拥有计算机的几乎都是规模庞大和很有名望的机构，软件更倾向于需要技术性的维护，所以它们经常需要得到硬件供应商所雇佣的全职系统工程师的帮助。在这种商业模型中，还有其他一些部分。例如，一个银行交易所软件的提供者认为，任何盗用其代码的人都是受欢迎的，因为这些人在没有技术支持的情况下使用该软件将使雇佣他们的银行很快损失数百万。

但是，当 20 世纪 60 年代出现微型计算机时，软件的价值开始变得意义重大起来。硬件提供商们开始大规模地对它们的操作系统和第三方系统企业收取额外的费用。开始，大部分的销售都采用完全的预约机制，从硬件、软件到维护，所以，这时候盗版还不是什么问题。但到了 20 世纪 70 年代中期，一些硬件厂商开始将这种预约机制转向打包机制，即最初为一个面包店编写的软件可以通过不同的参数销售给许多面包店。在此期间出现的版权争端的主要类型是，当一个程序员离开你的公司加盟另一家和你竞争的公司时会发现，他们的源代码中突然加入了许多你公司所特有的东西。所以，接下来的问题就是，他们是否也可以将源代码一并带走并对其重新改造以实现一个新系统。解决这一问题的标准做法是查看“软件胎记”，它是关于某个特定实现如何被完成的一组特征，例如寄存器被 push 和 pop 时采用的顺序等。这将继续导致一个问题，目前存在着各种各样的代码比较工具，这些工具有许多都是在大学中开发的，其目的是用来检测学生在完成编程作业时是否有抄袭行为（这类研究的思路导致产生了许多通用目的抄袭检测工具，该类工具可以对自然语言和源代码进行处理。而且，一般通过最少相同单词的指标来判别一个段落 [376]，该工具被人文科学的学者放到系统中来断定是否培根抄袭了莎士比亚作品，而且还可以从病毒代码的风格中试着断定作者的身份等 [476]）。

随着时间的推移，人们又发明了许多事物来对付软件。所以，一家购买微型计算机来进行股票交易的公司（或者用来签署办公合同）也许会对运行一个统计程序感兴趣，用该程序来准备管理报告。在这期间，安装的机器数越来越多，以至于必须进行软件共享，而不仅仅是偶尔共享了。所以，一些系统企业开始提出版权强制机制。一种通常的做法是检测处理器的序列号，另一种做法是时间炸弹。在 1981 年，当我工作于一家零售股票控制系统时，我们在软件中加入了一条每隔几个月就会显示出来的消息“错误号：WXYZ，请打电话获得技术支持”，其中 WXYZ 是加密过的用户许可证序列号。如果用户的序列号和其所声称的用户信息吻合，那么我们将给他一个密码使系统继续可用几个月（如果他们说的情况不正确，我们将派出销售人员去查证）。这种机制很容易被那些理解其机制的“用户”所破坏，但是在实际中，它可以工作得很好。在大部分的时间里，只有很少量的职员会因为系统弹出错误消息而拨打我们的热线电话。

在 20 世纪 70 年代后期和 80 年代初期，随着微型计算机的出现进而形成了一个大规模的市场，软件盗版真正成为了一个有待解决的问题，软件企业开始生产不需要技术支持就可

以安装和运行的软件产品。开始的反应各种各样。1976 年，也就是微软公司成立一年后，在比尔·盖茨的一封著名的公开信中，他抱怨只有少于 10% 的微型计算机用户付费使用 BASIC 语言软件 [319]。“谁来关心我们这些从事软件行业的人如何得到报酬?” 他问，“这公平吗?”，在他的信中做出了这样的结论：“没有什么东西可以促使我雇佣 10 名程序设计人员来对业余水平的市场提供优质软件。”

这次对于人们公平竞争意识的呼吁是迄今为止最有作用的一次，业界下一步将解决在小型机和微型机之间的明显不同之处。微型计算机没有处理器序列号。人们尝试过三种一般的做法：在机器上增加惟一标识，在机器中内置惟一标识，或者使用任何已经存在的惟一标识。

- 增加硬件惟一标识的标准做法是 dongle，这是一种通常附加在个人计算机并行端口上的设备，被软件用来询问某些信息。最简单的就是使用一个序列号，通常执行一个简单的质询—响应协议。同时，一些高端设备实际上借此进行某些关键性的计算工作。
- 对于软件来说，更加便宜同时也是十分常用的策略是系统被安装到个人电脑的硬盘上时，无法使用单纯拷贝的做法。例如，硬盘的一个扇区可能被标识为已坏，一段关键部分的代码或者数据记录在那里。现在，如果产品使用操作系统提供的拷贝工具从硬盘拷贝所需文件的话，“坏”扇区中隐藏的数据将不能被复制，那么软件拷贝也就不能工作。对此一种不同的做法是需要存在一张主盘，它是通过某些方法定制过的，例如用奇特的方式格式化或者利用激光在上面烧几个孔等等。一般应该在保护拷贝和保护主盘之间存在某些区别。通常人们希望可以为备份需要而做一些拷贝，但不能对拷贝后的东西再次进行拷贝，这叫做拷贝生成控制。
- 在我开发的一个产品中存储着个人电脑的配置信息，如当前安装了哪些卡、有多少内存、打印机的类型等。如果这些信息总在改变，那么该产品会让用户打电话给帮助热线。让人惊讶的是，在每一台个人电脑中有许多独一无二的标识：以太网地址和磁盘控制器的序列号仅仅是最显而易见的而已。倘若你使用某些手段来处理升级的问题，那么你必须可以使用组件的详细信息来将软件绑定到特定的机器上。

对付这些防御手段（或者至少这些软件没有将关键代码隐藏从而避免拷贝）的一种通用的攻击方法是利用调试器来仔细检查代码，并将所有的对于拷贝保护例程的调用都清除掉。许多业余爱好者之所以这样做纯粹是为了好玩，他们比赛看谁可以在软件推出后最先将去除保护后的所谓软件的非保护版本放到网上。甚至那些已经拥有软件拷贝许可权的人们也常常使用这些非保护版本的软件，因为，这些软件更加便于备份，通常也更加可靠。

软件供应商们还使用一些心理技术。

- 许多商业程序的安装例程都在屏幕上加入了注册的用户名和公司，例如，在工具条中。这将阻止用非法用户名注册的人进行盗版分发拷贝，但是也阻碍了某些合法用户在工作中将临时拷贝提供给同事的做法。
- 业界评论员讲述了大量由于没有支付软件费用进而没能得到关键性升级的机构和组织，其系统容易遭到破坏的故事。一个流行的例子是关于驻德国的美国军事基地由于没有对其使用的 VAX VMS 操作系统付费，因此无法得到该系统的安全补丁而遭到攻击破坏。

- 如果早先微软的软件（Multiplan、Word 或者 Chart）考虑到用户可能在调试器下运行，试图跟踪其执行过程，那么软件应该显示一条信息："邪恶之树结苦果，现在将清除程序所在磁盘的数据。"然后，程序找到软盘上的 0 磁道后一点一点地清除数据。

在 20 世纪 80 年代中后期，市场开始分裂开来。游戏市场向硬件保护的方向移动，并最终被游戏控制台产品所控制，这些产品使用一种封闭的体系结构，开发出的软件只能在各自私有的体系框架中销售。然而，商用软件供应商们停止了试图通过先进的技术方法来保护大规模市场产品的做法。还有以下几个原因。

- 除非你准备在防篡改 dongle 硬件上面投入资金，使用它来执行一些关键代码，但这种机制对于真正内行来说也是无济于事的，况且那些没有受到保护的代码将被匿名发布。这些代码将首先处于易受攻击的状态中。
- 随着处理器速度越来越快且代码变得越来越复杂，操作系统接口的层次也越来越高，对于"磁盘坏扇区"的软件保护例程也变得更加难以编写。现在，通过使用虚拟机软件（vmware）在 Linux 系统之上运行一个 Windows NT 系统，应用程序可以完全感觉不到那些机器所特有的特征，例如以太网地址。网络的影响使得软件开销越来越大，保护和盗版侵权变得越来越复杂。
- 保护具有损害性。多重 dongle 阻碍与其他系统的交互。软件保护技术使得产品的健壮性下降，因为，当你的硬盘损坏，当你将备份数据拷贝到一个新硬盘上时就会出现问题了。保护机制也导致了不同供应商的产品不兼容，而这种不兼容是根本没有必要的。甚至，在某些情况中，不同供应商的软件无法共存在同一台机器上。
- 随着软件产品变得越来越复杂，技术支持就变得越来越重要了。而只有在你购买了软件后，才可以获得技术支持。
- 计算机病毒的出现对业界的影响很大。它迫使公司用户购买软件防护产品，反过来意味着偶尔的拷贝也不容易做到。在短短几年中，防病毒程序使得那些防拷贝设计人员的日子越来越不好过，因为对于操作系统的不规范使用将导致系统不停地发出病毒警告。
- 从个人使用者身上是无法赚到什么钱的，因为通常他们仅仅是偶尔使用这些产品，然后就会把它抛到一边而不会去购买。
- 一定程度的盗版将有助于商业的发展。那些获得某工具盗版拷贝的人们如果喜欢使用，将会花钱购买正规版本，或者说服其上司来购买。
- 用户对于微软发布的导致恐慌消息的态度和反应相当冷淡和消极。
- 许多软件供应商对于处理诸如用户（在这种情况下，他可以将软件移到其他新机器上运行）或者某台机器（在这种情况下，他可以将安装了此软件的二手计算机出售给别人）是否获得了软件许可权的问题并没有太大兴趣。因为以上两种情况最为常见，而保护机制将会使得这种或者另一种情况很难处理，并导致问题的出现。使用诸如 dongle 的技术虽然可以很容易地同时解决以上两种情况，但代价又过于昂贵。
- 最后，Borland 通过推出 Turbo Pascal 软件而使整个业界震惊。在那以前，一个典型的编译器的价格在 500 美元左右，而且文档资料贫乏，以至于用户不得不再花费 50 美元来买书学习如何使用该编译器。Borland 的产品价格仅为 49.95 美元，而且在技术上也成为有力的竞争者，并且还附带有一本与第三方产品一样出色的用户手册（所

以，如其他人一样，我一听说此事，就从朋友那里获得了一份拷贝，对它进行试用而且感觉不错，于是就出门将它买了回来）。“薄利多销”被证明是一种可以获取更多收益的商业模型，甚至对于诸如编译器这类特别的产品也是一样。

随后，业界又转向利用法律措施来解决盗版问题。最初，主要是在大部分国家中建立一些反盗版贸易组织。在美国，这一组织叫做软件出版者协会（Software Publishers' Association），该组织针对各大公司广泛使用盗版个人电脑软件的行为进行起诉。随后，各种媒体和小型企业往往收到恐吓信，要求公司对于版权问题制定详细的策略，提供一些被认可的软件审计方案和专门搜捕盗版软件的强制人员等。各公司使用各种骗术来避免被控使用盗版软件。一个典型的骗术是“免疫列表”。例如，我开发的一个交易目录产品包含了许多虚假公司的详细信息，还有发行方服务台电话号码，该服务台的员工将询问公司名称并核查一下该名称是否在付费订户列表中。

最终，业界发现法律手段不但提供了强制性的工具，同时也极大地限制了盗版的发展，像使用时间炸弹这种历史悠久的技术现在在许多法律中已经明令禁止了。例如，在 1993 年，一个软件公司在英国 Scunthorpe 地区的主管收到宣告其有罪的判决书。原因是他触犯了英国计算机滥用法案（Britain's Computer Misuse Act），由于他利用时间炸弹对一个系统进行了未经授权的修改，使得该系统被强制对一项有争议的发货单进行支付［194］。目前，许多法律都认为时间炸弹不可以被接受，除非用户在购买时被明确通知有时间炸弹的存在。

现在重点被转回到技术机制的方向上来。站点许可证协议通过许可证服务器执行，该服务器有点像 dongle，但是在企业网络中的某台个人电脑上实现，其目的是限制可同时运行的某一应用程序的数量。利用分解应用程序代码的方法，这些服务器同样可以无用武之地。但是，随着应用程序代码越来越庞大，这种做法也很难实现了。同时，再加上法律的配合，这种方法一般也就足够了。其他机制，例如经常性地发布升级软件，会使事情变得很繁琐，而且使操作系统变得非常不可靠，以至于每隔几个月它就会完全崩溃掉，而不得不从各种存储媒介中重新安装所有的软件。

因此，软件行业正在逐步形成的模型包括了技术和法律两方面的措施，要理解这两方面的局限性，同时也要接受一定数量的拷贝会发生这一现实（通过这种方法，你可以试图平衡一下总通过完全支付方式购买软件的预算）。亿万富翁比尔·盖茨曾说过这样一句有启迪作用的格言：

> 比如有个国家，每年都有大约三百万台的电脑被售出，但是那里的人们从不购买正版软件。也许某天他们会去购买。只要他们想偷取软件，我们就让他们偷我们的。他们将会对此上瘾，然后我们会在下一个十年中找出如何以某种方式去控制他们［332］。

最新的发展要求我们必须处理在线注册。如果你设计了一个产品，可以让用户通过 Web 站点和你取得联系。例如，下载最新的汇率、病毒特征码或者安全补丁，那么你能够保存一个关于谁使用了该软件的日志。但是，这样做很危险。当微软试图在 Windows 95 中加入注册向导时，曾引起了暴风雨般的反对之声。还有，我的一位同行发现他无法对游艇上的一台机器升级 Windows 98，因为它总是处于离线状态。但是，目前却有向这个方向发展的趋势。

采取不同的方法来处理不同的威胁，这种做法不值一提。大规模的商业伪造可以通过监

控在线注册的产品序列号来发现。但是，这类操作虽被发现，却是调查机构通过反向跟踪产品供应链的方法来停止业务继续进行的，并且通过综合使用我们在第 12 章中介绍的印章和安全包装技术来推迟其商业运作的开始时间。

在对付个人和小型商家时，这样做可能会或多或少地起到一些作用，但是，对于媒体和大型商家来说，主要风险在于少量合法拷贝运行在大量机器上的情况。通常的对策通过软件贸易协会施加的法律压力、外加站点许可权和奖励揭发行为的方法来处理。对于微软这种公司来说，针对商家的业务量要远远多于对个人使用者的业务量，采取以上做法意义就很显著了。也许，这就是业界阻碍在线注册和强制个人用户使用版权的主要原因。考虑到公众的牵引效应，潜在的额外收入会很少。其他的考虑是保护隐私的法律（尤其是在欧洲）很难跟踪到人们变更地址或者购买了二手电脑等信息。

总结一下：21 世纪初期，尤其在面对特定的对手时，没有一种低开销的、可用的保护技术是安全且坚固的。但是，通过对它们的组合使用，大型的软件供应商通常能够得到一个可以接受的结果，尤其是当价格不很离谱以及供应商的产品还算流行的情况下，更是如此。小型的软件公司可能不会太高兴，因为它们的产品仅仅是适应特殊目的而生产的，如果拷贝的风险很低，那么它们常常会收益甚微，而且无法对拷贝的现象进行有效控制。

还存在着其他可以替代的商业模型。一个就是放弃产品限制版本的使用，在线销售密码来对产品的全部功能进行解密使用。Unix 之所以流行，是因为对于大学来说，是可以免费使用的；同时，如果是公司使用的话，就必须付费了。对此，另一种不同的做法是对个人免费发放基本软件版本，但对于公司收取费用，Netscape 就是这样做的。一个更加激进的模型是软件完全免费发放，而通过顾问咨询、技术支持到网站广告等多种收费服务来获取收益，Linux 产业目前正是这样做的。

这些经验使得许多计算机用户相信，最终对于“好莱坞”问题的解决方案存在于商业模型的变化中。但是，在我们迈入保护多媒体内容的世界前，先来大致了解一下历史上的先例。

20.2.2　书刊

在书刊出版业开始之初，Shapiro 和 Varian 呈现给我们一个有用的历史教训［696］。1800 年，在英国只有 80 000 名经常性读者，那时，大多数书刊都是关于严肃的哲学或者神学。在小说出现之后，出现了很庞大的书刊市场需求，而且还出现了许多提供借阅书刊的图书馆。受教育阶层的人被吓坏了，印刷工也被吓坏了，因为图书馆剥夺了他们的图书销售权。

但是，图书馆的做法却吊足了人们对书刊的胃口，以至于到 1850 年时，读者的数量达到了 5 000 000。人们将原先从图书馆借来看的书买回来，书刊市场蓬勃高涨。图书馆原先的做法被证明是对印刷工人们最大的支持，可以帮助建立一个全新的书刊市场。

20.2.3　音频

盗版被用来拷贝音乐和其他音频的历史比拷贝软件还要久远。帕格尼尼对于人们拷贝他的小提琴协奏曲十分担心，以至于他仅仅在预演或者演出前才亲自将曲目分发到乐队演奏处，随后又将乐谱收回（结果是，他的许多作品都被遗失了）。

近些年，有一两起恐慌事件引起了整个业界的关注。当 20 世纪 60 年代盒式磁带录音机

出现时，唱片业在某些国家游说，试图对音频磁带收税，进而将税收收入分配给版权拥有者。技术上的措施也被尝试了。披头士的唱片“Sergeant Pepper”包括一个 20 KHz 的干扰音调，它理论上联合了磁带 21 KHz 的斜频率产生 1 KHz 声音来干扰音乐质量。实际上，这种机制未能正常工作，因为许多唱片机都不识别这种干扰音调。但在实际中，这并不要紧。卡带最大的问题是质量的下降，这在家用设备中最为明显。许多人仅仅使用它们来录制歌曲，以便在汽车里播放。然后，到了 20 世纪 80 年代，Sony 随身听的问世使得卡带的销售量猛增。虽然，还是存在非法拷贝的情况，但同时受到保护的卡带也有很好的销量，整个音乐业都在赚钱。

数字录音带（digital audio tape，DAT）的出现引发了另一个忧虑，因为通过它可以对 CD 的内容进行完全的拷贝。所以，最后又出现了*连续拷贝管理系统*（serial copy management system，SCMS），这是处于磁带头部的单一比特位，用来指示该磁带是否被拷贝过 [410]。这种思路是对来自 CD 的拷贝做上标记，导致它们不能被再次拷贝。通过这种方法，人们可以对其已经拥有的 CD 进行拷贝，并可以在其他地方收听，但是却不可以做拷贝的拷贝。这种机制工作得并不好，因为这个只准拷贝一次的比特位被许多录音机忽略，而且通过简单过滤的方法也可以很容易蒙混过关。再者，这并不是 DAT 未能广泛使用的原因（在 CD-ROM 存储数据的头部也存在一个不准拷贝的比特位。但是几乎普遍被忽略）。

最近，音频拷贝问题重新引起关注，这要归功于用于压缩音频的 MP3 格式的广泛流行。以前，数字音频是通过其大小来被保护的，一张装满没被压缩过的音乐的 CD 是 650Mb。然而，MP3 使人们将数十兆字节的 CD 音轨压缩到仅仅几百千字节大小，从而让通过拨号上网的用户的下载成为可能。MP3 在大学中的使用更加普遍。1998 年时，MIT 中 40% 的网络流量是 MP3 信息。一些学生成为了隐蔽的唱片销售者，在校园中销售音频数据，而不用对版权拥有者交纳版税。

业界最初的反应是寻求技术上的弥补手段。另外一种作为可选方式的音频压缩技术发展起来，它包含了版权保护机制（例如 [483]），但是很难被真正使用开来。美国电影业仍然给计算机产业施加压力，希望可以做出让音乐拷贝更加困难的平台，但是这种平台一直没有出现。

- 首先，个人电脑是一种开放的平台，其本质决定了它可以很容易地在内部拷贝比特流。已经提出了一些建议来关闭这个平台，例如通过将总线加密技术（在 14.5.2 节中讨论）包含到缓存控制芯片中，甚至是主 Intel 处理器线路中。但是，在这个方向上迈出的第一步，即在 PentiumⅢ中加入处理器序列号遭到了大众的抗议。试图保护 DVD 私有权又意味着阻止运行 Linux 的个人电脑使用 DVD，这将导致一场斗争，我会在后面进行讨论。到目前为止，我们看到的都是攻击案例，例如将音乐数据流加密后送到声卡驱动软件的途径。对此，可以修改声卡，从而截获被译码的数据。
- 第二，新型硬件的成功依赖于软件的可用性，反之亦然。为了给音频开发一个新平台，最好对于已有的 CD 和播放器等提供向后兼容。一个窍门是仅仅对 CD 音轨中最为重要的几个比特位加密，再由个人电脑中的声卡驱动程序进行解密操作。通过这种方法，使用不支持该解密操作的播放器的用户也可以听到音乐，而那些具有授权拷贝保护功能设备的用户将得到更高质量的音乐。然而，音乐的质量对于很著名的音乐来说是极其重要的，这不是经济学中的重要性，只有使用改造过的声卡，才能

够解析出全部的声音信号，从而获得最佳的质量。最后，将 Linux 用户排除在下一代音频收听者之外也将导致和 DVD 类似的战争。

- 无论怎样，毕竟已经有许多容易复制的 CD 售出。而对于那些高质量的数字拷贝而言，实际上，它们中的许多是无法完全由业界控制的。

好莱坞下一步要做的就是起诉，其主要的矛头指向那些允许 MP3 共享的网站。商业的 MP3 网站不得不建立了订阅频道，从而希望与音乐界和睦共处。但是，这些频道很快就被诸如 Napster 和 Freenet 等系统替代，它们允许希望交换音轨的使用者们直接互相联系。在本章后面讨论隐私机制的时候，我还会回来讨论这些系统的潜力所在。同时，可以看到，对于音频的保护并不一定要和软件的保护完全区分开：首先人们提出技术上的解决方案，但随着法律的冲击而宣布失败，最后只得综合利用技术和法律两种手段来解决问题，但即使这样做也不能完全杜绝盗版行为。

这并不是说，我期待着一种可以解决所有版权控制问题的手段出台。由于微软与小型专业公司有着不同的需求，而且使用不同的做法，所以人们也期待用各种不同的控制方法来处理目前流行的各类盗版行为，而不是出现一种类似 Bonzo Dog Doo-Dah 乐队的 CD 引起许多人推崇的控制方法。

- 在前一种情况中，听众的密度很高，一个音轨通过个人拷贝就可以广泛流传开来，但是其流行时间可能很短。速度和流行将意味着所有。流行商品的销售确实变得比 CD 的销售更加重要；随着 Netscape/Linux 业务模型的出现，将产品免费出售，而从维护中获取收益将十分有意义（旅游、T 恤衫、球迷俱乐部……）。
- 在后一种情况中，其吸引力是永恒的，但是热心者只是分散的一小部分人，他们之所以拷贝音轨，主要是因为他们觉得花费 17.95 美元来购买 CD 是受到了不应有的剥削。正因为如此，甚至不太可能通过许多的管理来对这些违规者进行起诉。所以，只能希望在价格或者包装机制上做微调，从而吸引收藏者的目光。

我还希望好莱坞可以遵循软件行业的模式，对于拷贝采取一种更加成熟的态度。毕竟，价值 1 000 亿美元的 70% 的市场比价值 500 亿美元的 98% 的市场要好多了。而且，正如适当数量的拷贝可以帮助软件市场一样，它可以帮助音频销售市场：Grateful Dead 音乐组合鼓励私卖磁带录音，因为他们知道，这并不会影响到他们的销售。

20.2.4　视频和付费电视

视频卡带最初的情形和音频卡带十分类似。首先，好莱坞出于害怕，拒绝出于家庭收视的目的而发布电影。这次也同样出现了防盗版的各种技术性措施，例如，Macrovision 系统，该系统通过伪造的同步脉冲来混淆国内生产的录像机（VCR）中的唱片线路，但是又一次证明，这种类型的措施很容易被专业使用者直接的攻击击败。然后，好莱坞开始对视频租赁商店起了疑心，就像书刊出版商在最开始对待图书馆一样。但是，视频租赁极大地增加了录像机的销售数量，也刺激了人们对于拥有自己喜爱电影的欲望。录像机和视频卡带形成了比 rock star 公司的玩具更加畅销的庞大市场。目前，在迪斯尼这样的公司中，销售预录卡带的收入占了总收入的很大一部分。商业模型的改变使得电影发行真正成为了视频销售的广告。

目前，世界上许多十几岁或者更小一些的孩子们都要求其父母给他们收集迪斯尼的卡

带，就像他们的小伙伴们一样，对于视频卡带的盗版必须在包装上做得和正版尽可能相同，这会省去许多业界伪造问题的麻烦。随着在线注册机制之前出现的庞大的软件市场，或者今天的香水和瑞士手表市场的出现，人们使用了一些强制性措施，包括：通过地区代理来购买卡带、寻找伪造窝点、跟踪供应链、对盗版分子进行法律诉讼等。

更加有趣的技术保护机制已经被内嵌到最新几代付费电视系统中了。

付费电视的出现，无论是由电缆或者卫星发送信号，都会提出对条件访问机制的需求，从而允许电视台运营商通过各种不同的方法来限制某一频道的接收。如果，他们仅仅购买了在波兰地区放映电影的权力，那么他们必须阻止德国或者俄罗斯观众也能通过卫星观看到相应节目。成人频道的运营商需要防止英国和爱尔兰地区的信号接收，这些国家对此都有十分严格的审查法律。大多数运营商希望对于诸如拳击比赛这类特殊事件收取额外的费用。

20.2.4.1　典型的系统体系结构

目前，有许多系统发展起来，这在很大程度上取决于机密视频信息的硬件成本（关于置顶盒的历史，参见 [186]）。20 世纪 70 年代的第一代系统性能拙劣，使用一种模拟设备，通过一次次不停转变视频信号的机制来工作，严重干扰了信号的同步，同时由于插入测试信号阻碍了电视的自动增益控制。这些机制很容易被实现，但同时也很容易出现漏洞，打破这些机制甚至不需要使用密码分析学的方法，而只需示波镜和一些耐性就可以了。

第二代系统出现于 20 世纪 80 年代后期，该系统将模拟技术和数字技术结合起来，广播部分是模拟的，而订阅控制为数字的。包括 Videocrypt、Eurocrypt 和 Nagravision 等。这类系统通常具有三个组件：

- 位于电视台的订阅管理服务，电视台发送加密过的视频信号，内部嵌入各种不同的授权控制消息（entitlement control message，ECM），并发行访问卡（诸如智能卡）给订阅者。
- 利用置顶盒将电缆或者卫星信号转变成为电视机可以处理的信号。置顶盒包括描述信息。
- 最后就是订阅者智能卡，它使设备个人化，从而控制置顶盒允许描述哪些解密。这是通过对 ECM 信息的解释以及提供置顶盒中描述电路的密码来实现的。

这种做法意味着，那些复杂而又昂贵的过程，例如对大块视频信息的描述，可以通过一个高效的标准设备来完成。这类设备的产品生命周期往往很长，同时还具有关键的安全功能，这些功能可以在发现攻击后立即进行替换。在卖给用户时，使用低价格的访问卡，这种卡也可以很容易地被替换。如果置顶盒本身必须在系统每次受到攻击后都替换的话，那么在价格上将很难吸引用户购买。

置顶盒从输入的数据流中解码出 ECM 信息，然后将它们传递到卡上。卡处理 ECM 后得到控制消息（例如，“智能卡号码 123356：你的订阅者没有支付，停止工作直到再次通知”）和密码，也就是控制字，再传送给置顶盒。置顶盒使用控制字来描述视频和音频数据流。

20.2.4.2　视频搅乱技术

最普通的视频搅乱技术是剪切和旋转。这种搅乱技术一次剪切视频信息中的一行，此行由控制字节决定一点。然后，交换其左右部分（见图 20-1）。这里涉及到视频信号的模拟到

数字的转换，在缓冲区中存储，以及旋转后从数字到模拟的转换。这个过程可以在一个低价格的超大规模集成电路芯片中完成，这种技术在20世纪80年代中期就已经使用了。

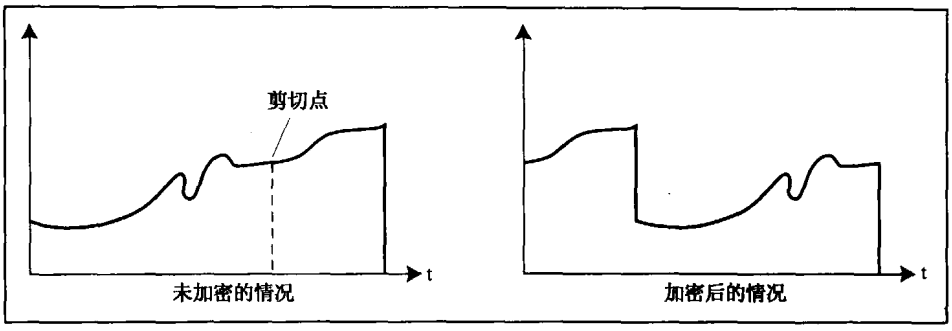


图 20-1　剪切和旋转搅乱

这类系统中存在着固有的不足，即一次只能对一行视频信息加密，但是连续的行往往具有相关性，所以通常使用信号处理技术对图像进行重建也是可以做到的。这项工作在1995年时由Markus Kuhn首先完成，它需要使用位于Erlangen大学的超级计算机进行实时操作才能完成。图20-2显示了被加密过的一帧视频数据，图20-3是该帧被处理后的效果。到写作此书时为止，已经可以利用一些功能强大的个人电脑进行类似操作了，虽然还不能完全实时的进行［733］。如果这类攻击早些时候就出现的话，那么将导致整个系统的完全崩溃，因为不管智能卡对于密钥的管理多么出色，视频信号也可以在没有它们的情况下被解析出来。但是这种搅乱技术持续了足够长的一段时间。付费电视的运营商目前正将他们的客户转向一个完全数字化的系统中，在这种系统中，再利用模拟信号特性进行攻击就无法实现了。

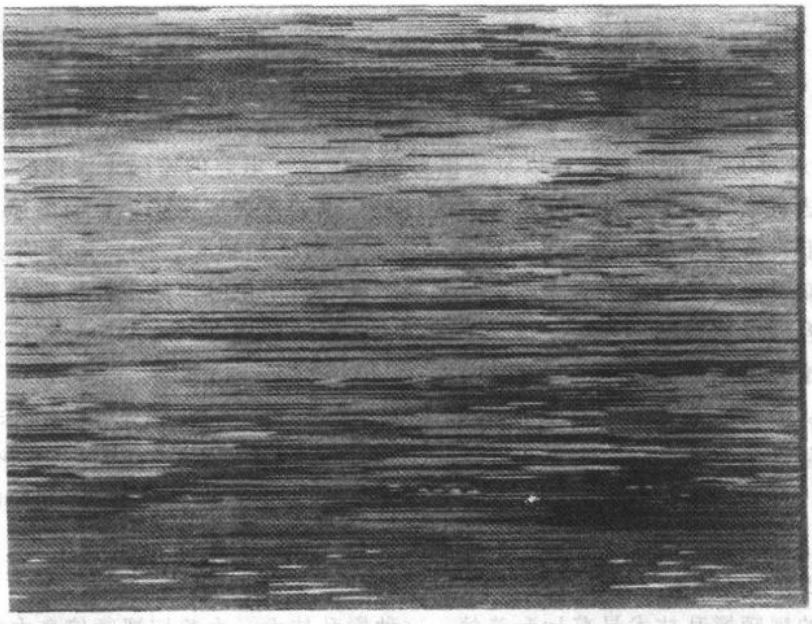


图 20-2　搅乱视频帧

图 20-3　经过处理的视频帧

对于控制字节的生成机制是独立进行的。每隔半秒钟，智能卡需要提供给置顶盒一个新的控制字，然后该控制字被加载到关键字流产生器中，工作方式如下：有两个线性反馈移位寄存器，在 Eurocrypt 系统中的长度分别是 31 和 29，它们可以产生出很长的线性序列。寄存器 1 中的一些位用作从寄存器 2 中选择一位的多路复用器的地址线，这个选出来的比特将变成关键字流序列的下一个比特。每个输出的连续字节成为搅乱器的控制字节（见图 20-4）。

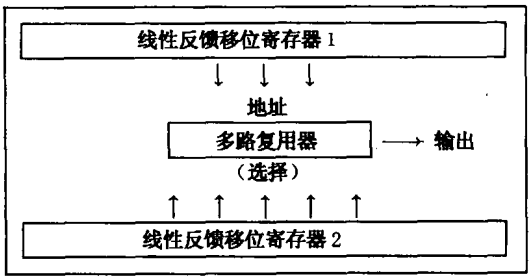


图 20-4　多路复用产生器

设计者们认为，破解这类密码需要猜测密钥，而一个 60 位长的密钥需要猜测 2^{59} 次，这是一种不经济的选择方式，因为一秒钟大约只能猜测两次。但是，在猜测这类密钥时是有捷径的，技巧就是猜测寄存器 1 中的内容，使用这个地址信息来取代从寄存器 2 中发现的比特，而且，如果导致冲突的话，则放弃对于寄存器 1 的当前猜测值（我在 1985 年时曾发现过这种攻击，对它的研究使我对密码学产生了浓厚的兴趣）。目前，从数据间的相互关系中很容易推算出每个控制字的最高位的 4 个比特，但要找出其他位则还要花费一些气力。所以，你可以很容易地从一段关键字流中找到一半的位，然后利用密码分析技术重新找到控制字的所有位。但是，这种方法的计算量和先前那种完全信号处理的攻击方法比较而言并没有多大的改善。对于序列密码，例如搅乱技术，本身可能非常脆弱，但是它仍然可以存在相当长的一段时间。所以，盗版要想成功，则必须想办法去攻击订阅者管理系统。

20.2.4.3　订阅者管理技术

考虑到有相当数量的置顶盒存在对于给定控制字流的视频广播数据无法描述的问题，下一个问题我们来考虑如何让付款用户能够产生控制字。通常，这可以通过白名单和黑名单的方法来解决。但是，最新一代的付费电视系统中的可用带宽很低。通常，每秒钟只能传送10个ECM消息，或者说每天只能发送50万条ECM消息。因此，黑名单的方法成为主要的方式。在一个拥有500万订阅者的系统中，发送一个私人信息到每一个用户将需要花费超过一周的时间。

基础协议是用来让智能卡解释ECM消息。而且，如果当前的节目允许订阅者收看，那么将利用卡中存储的主密钥在一系列ECM消息的基础上计算出MAC，并将MAC提供给置顶盒作为控制字使用：

$$CW = MAC\ (K;\ ECM_1,\ ECM_2,\ ECM_3,\ ECM_4)$$

通过这种方法，如果订阅者停止支付相关费用，那么发送一个ECM消息就可以让他们的卡处于非激活状态，从而使其无法产生控制字。而且，卡也需要对ECM流进行访问，从而计算出控制字。

20.2.4.4　哪里出了问题

对于这种系统的第一类攻击是协议攻击。既然，由智能卡发送到置顶盒中的控制字对于每一个只有在对控制字进行判断后才能描述节目功能的置顶盒都是一样的，那么某人就有可能在智能卡和置顶盒之间放置一台个人电脑，记录下控制字数据流，然后将它们发到因特网：其他人可以先将经过搅乱处理的视频信号记录下来，然后从网上下载控制字文件再对其进行反搅乱处理[532]。使用这种关键字日志攻击方法的服务器确实存在，但是它们对于付费电视系统的影响并不是很大，因为没有多少人准备购买连接个人电脑和置顶盒的特定适配器。其他攻击手段还包括阻塞器，用来防止接收到的ECM消息送至智能卡上。通过这种方法，你自己就可以取消订阅，而不用电视台运营商来取消你的业务。还有主密钥泄漏：某些人购买了二手个人电脑，出于好奇，他想看看硬盘上被删除的文件，对已删除文件进行成功恢复后，得到了一个完整的付费电视系统的预订管理系统，其中包含了系统主密钥。

一旦这些容易采取的方法试过之后，商业化的盗版者们又转而使用一系列攻击对用户智能卡进行反向工程，我在第14章曾描述过。但是，硬件级别的修复将受到新卡发行的限制，运营商们不希望在一年中多次发行新卡，因为这种做法将使每个预订者支付一定的费用，而预订者在这方面的费用通常应该少于每月20美元。所以，必须找到其他有效的防护措施来对付反向工程。

起诉的方法也尝试过，但是并没有起到运营商们希望的效果。在爱尔兰曾发生过一起对于盗版败诉的案件，这使得在一段时期里，整个欧洲几乎就是那些通过电子邮件订货方式来销售盗版卡的盗版者们的天堂。此时，整个产业界都在积蓄力量，试图制定一项适用于全欧洲的法律来替代爱尔兰首都都柏林的法律，但是这项工作花费了许多年，而且损失巨大。例如，到了1995年中期全英国的卫星电视台由于盗版卡的影响，收入损失了5%。

20.2.4.5　如何修复

整个20世纪90年代中期，盗版者和运营商都处于推出对策与反对策的激烈竞争之中。运营商购买盗版卡，对它们进行分析，而且使用各种技巧来导致这些盗版卡无法正常工作。摆在运营商面前的问题是：当系统中所有的秘密部分都处于危险状态的话，那么将如何对盗

版进行反击呢?

似乎不可能再用常规的加密方法来处理这些问题了，但是运营商们还是使用它。一种更加有效的技术是 ECM 消息，该消息包中的内容像代码一样被智能卡执行。通过这种方法，现存的智能卡则不再能够工作，而且由于实现上的差异，也使得真卡和盗版卡很容易被识别出来。在 MAC 算法中加入一种对两个不同平台一定产生不同计算结果的机制，即使是使用随机的时间条件来计算，结果也应该不同。利用这种机制就可以让盗版卡产生控制字无效。

让我们简要地看一下如何废除停止付费订阅者的访问权限问题是很有必要的。每个订阅者的智能卡中包含一个订阅者密钥 k_i，而且一棵关于中间组密钥 KGij 的二叉树将 k_i 连接到当前活动主密钥 KM 上面，如图 20-5 所示。每个操作卡都知道连接自身和主密钥这条线路上的所有组密钥。在这种配置下，如果说密钥 k_2 出现在盗版卡中，而且必须被废除掉，运营商将发送一系列包，使得其他所有的订阅者卡都计算一个新的主密钥 KM′。第一个包是 $\{KM'\}_{KG_{12}}$，它将使一半订阅者立刻重新计算 KM′。然后，再发送被升级过的 KG11 的新版本加密过的 KM′，即 $\{KM'\}_{KG'_{11}}$。然后再把新组密钥 KG'_{11} 用 KG_{22} 加密，以此类推。这样做的效果是，即使拥有 1000 万用户，运营商也只需发送少于 50 个 ECM 消息就可以完成密钥更换工作。当然，这并不是一个完全的解决方案：运营商还是需要考虑如何对付盗版卡中包含几个订阅者密钥的情况，以及如何识别出那些泄漏的密钥，同时又不用对盗版卡进行复杂的反向工程。然而，二叉树确实是一种很有用的工具（使用私人密钥保护组密钥并不是什么很新的技术，在二战期间，Marks 叙述了英国空中特勤团是如何给其间谍发送开放代码，这种代码使用它们的私有密钥进行加密[523]。当这些开放代码利用无线电通过广播的方式发送后，它同时发送给多个间谍诸如“炸毁铁道桥”之类的命令）。其他具有类似需求的应用包括海军特遣部队中用于管理共享的“每日密钥”的工作。

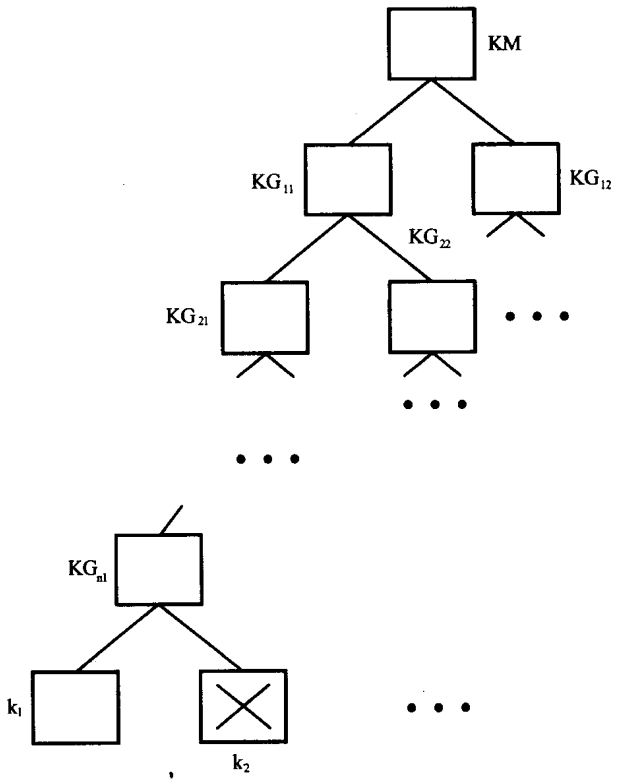


图 20-5　二叉树

心理上的措施同样被使用。例如，一家有线电视台以广播的方式发送通知，说是提供免费的 T 恤衫，但同时又让那些合法用户们看不到 800 的业务电话而只让那些盗版卡的用户看到，这样一来，运营商就可以通过谁打了 800 电话来获得盗版用户的列表了。经济因素也会

带来一些差别。付费电视的盗版之所以成功，主要依赖于它们和传统的软件公司一样都有市场需求，而这类市场是需要时效性的：一个可以在三周内做出99%逼真伪制品的盗版者一定会超越那些用时三个月的盗版者。所以，盗版卡同样有漏洞，要想高效地找出这些漏洞，一定要对盗版经济效益有一个很好的理解。最好是先让盗版者建立起一个坚固的用户基础，然后再对付这些盗版者，让他们失去更多的潜在用户的信赖，这要比立即对付他们有效许多。但是，如果对他们过于放纵，那么这些盗版者将同时获得经济和技术的资源来扩展用户，并进行更高质量的伪造活动。

从付费电视系统中得到的主要技术上的教训是，对于安全恢复应该有事先的计划，而且在产品中应该隐藏一些特征，这些特征在开始并不使用，但可以在后来激活（通常，同样的教训在多年前就应该在另一个行业中学会了，这个行业就是印钞业）。

最后，智能卡应该做得很难被伪造，其中包括在处理器硬件中加入私有加密算法。当攻击者不再利用探测台来读取算法，而必须对芯片进行反向工程时，将大大减少运用技术能力进行攻击的攻击者数量。这些攻击者中的许多人都不得不在该行业中从事顾问或者赞助商的角色。那些还执迷不悟的人，也只能摆摆样子，仔细观望了。大量法律的强制手段可以作为反盗版的最后一个环节。业界可以一路穷追，直至抓获主要的商业性盗版团伙，然后让他们彻底失业，或者将他们送入监狱，或者对他们进行起诉。

例如，在20世纪离我们最近的一起大型付费电视盗版案中，英国盗版者Chris Cary被宣布犯有伪造卫星电视台智能卡的罪行，一家加拿大公司出资105 000美元，让他对该种卡进行反向工程。然后，他将伪造出来的卡通过一家在爱尔兰的公司进行出售，而当时在爱尔兰，这种伪造卡的行为并非不合法行为[568]。卫星电视台的安全顾问雇佣了一个侦探潜入Cary在爱尔兰首都都柏林的销售部门，并暗地里照下了关于这些非法交易是来自英国的各种证据[403]。英国警方并不想对此进行起诉，所以卫星电视台进行了私人起诉，并使Cary最终被宣判有罪。当后来Cary越狱后，卫星电视台的私家侦探无情地四处追捕他，并最终在新西兰将他抓获，Cary是利用一个写有已故人姓名的护照飞到新西兰的[367]。

付费电视所经历的历史使我们又得出了一个商业教训，即一个行业必须同时采取工程和法律两方面的版权保护措施。只采取一种措施是不够的。关于这方面的一个反面例子就是DVD。

20.2.5 DVD

电子消费行业带来了数字化视频光盘（digital video disk，DVD），后来又在1996年时改名为数字通用光盘。通常，好莱坞对此很害怕，并宣称如果DVD技术不具有完善的拷贝保护机制的话，那么第一流的电影将不可以通过DVD方式发行。所以，出现了一种名为内容搅乱保密系统（content scrambling system，CSS）的机制。

还存在着另一种方案，即将世界划分为五个区域，光盘只可以在指定区域列表的播放器中播放。这种机制可以支持传统业务，如首先在美国发行电影，然后在欧洲，最后到其他地区，这样可以在电影票房不佳时减少损失。这种区域代码将是首先被抛弃的东西，目前越来越多的制造商都在其产品中忽略该代码。对于诸如DVD产品的市场全球化正在破坏DVD播放器的销售市场，使得这些播放器只能播放本地制造的光盘。

这样就只剩下CSS方案，因为在DVD刚出现时，CSS还被普遍认为是一种易受攻击的

技术［601］。有这样一个故事：设计者被告知必须在两周内提出一种拷贝保护方案，并且整个方案的实现不可以超过 3 000 个门电路，密钥长度不可以超过 40 位。这样一来，设备就不会与美国出口规定产生冲突。另一个故事是：DVD 协会曾经试图强迫播放器制造商通过压敏二极管（Matsushita）来获得 CSS 专利许可，通过这种二极管的状态可以实现其他拷贝保护机制［119］。无论设计上是谁出的错误，令人好奇的是这类系统居然继续了三年。

对于 CSS 技术的细节性描述目前是起诉的焦点问题，在美国发布了许多禁令，不允许各网站发布 CSS 代码。这样做几乎没有什么效果，因为大多数在美国之外的网站不受此限制（例如［737］）。然而，由于我的出版商在美国，所以我不希望他们雇佣律师来解释我的所作所为，这里我仅仅进行一个适当的简要描述。

CSS 基于序列密码，这种技术和图 20-5 很相似，但是多路复用器被一个完全的加法器所取代。每个后续的密钥序列位的获取是通过后两个从移位寄存器中携带的信息输出进行位加得到的。如果存在五个移位寄存器，并且其互质长度大于 70［656］，那么结合使用移位寄存器的异或操作和组合器的进位加操作将可以产生强大的密码。但是在 CSS 系统中只有两个寄存器，且长度分别为 17 和 25。所以该系统比起上面讨论的系统来说，其攻击将容易 2^{16} 倍。在使用密码加密密钥信息而不是数据信息时，还存在进一步的操作，但是这也仅仅是将复杂度提高到 2^{25} 而已。

下面是对 DVD 协议的描述。每个播放器都具有特定制造商所特有的一个或几个密钥，而每张 DVD 光盘具有一个光盘密钥 kd，并且用当前制造商的 kmi（在 1999 年时共有 409 个）密钥加密：$\{kd\}_{km1}$，$\{kd\}_{km2}$，$\{kd\}_{km3}$，…，$\{kd\}_{km409}$。还有一个光盘密钥 kd 的散列函数，它通过对自身进行加密得到，即：$\{kd\}_{kd}$。实际内容通过从 kd 继承下来的一部分密钥进行保护。当然，密码通过 2^{25} 次试验就可破译，所以任何光盘密钥都可以从单一光盘散列中找到。

所以，CSS 违反了 Kerckhoffs 原则，该原则依赖于保密的算法进行保护。DVD 协会似乎不理解这一点，因为他们试图通过经济压力来维持制造商密钥的保密性。这种思想是如果任何制造商主密钥泄漏，那么该产品将不能适用于未来的光盘，即该播放器不能播放新发行的电影。所以，制造商们都努力实现完美的防篡改机制，至少他们希望如此。但是 CSS 的设计并不支持这些期望。给定系统中的任何密钥，其他所有的密钥也将被同时发现。还有，对于大规模电子产品销售来说，是不允许使用这类进行严格篡改保护的处理器的。

还有一些问题出自个人电脑是一个开放平台这样的事实。DVD 协会所选择的处理办法是，那些生产 DVD 播放器软件的供应商们必须扰乱其源代码，从而使反向工程难以进行。这时恰恰出现了一些关于对系统软件进行扰乱操作技巧的学术论文［58］。这些技巧可以让反向工程从原先的几天加大到需要几周才可以完成的难度，但是，一旦 CSS 技术不再使用时，反向工程还是会出现的。

对于个人电脑的开放性，一个更加严重的问题来自 Linux 系统，这是一个拥有几百万使用者并且源代码公开的操作系统。DVD 协会所采用的基本观念和体系结构在制造适用于 Liunx的 DVD 驱动器时与 Linux 界产生了不一致。所以，随着具有 CD 驱动器的个人电脑被具有 DVD 驱动器的个人电脑所取代，Linux 的使用者或者放弃 CSS，或者放弃 Linux 而使用 Windows 操作系统。在这种情况下，即使每个 DVD 播放器包括付费电视那样的智能卡处理

器，在某些人可以将这些信息读出之前也是需要一定时间的。[⊖]

对于 CSS 机制的一种破译结果就是 DeCSS 程序，它可以使任何 DVD 失去保护。业界对此的反应只能去找律师。美国的存储 DeCSS 程序的网站都通过禁令予以禁止，这些禁令仅仅是使得软件的分发更加广泛，业界的做法显得更加愚蠢 [491]。然而，还存在着令人不快的因素。例如，版权法本来是允许公平使用的，这包括对作品的某一部分进行拷贝，并作为拷贝者自身的学识、在其他地方的引用甚至作为批评嘲讽的对象都是可以的。电影界的律师们对于数字视频拷贝的镇压性做法使得版权拥有者对自己的数字作品完全失去了控制权。这对于大学、公众图书馆和许多其他部门来说都是值得悲哀的，在这里，对于公平使用权力的利用被严重阻碍着。所以在本章开始 Barlow 的一段话中提到的战争就这样开始了。

美国的一家权威机构 Samuelson 认为，不仅仅对于软件业出版商，还包括其他许多行业中的出版商都是一样，出现一些拷贝对于它们来说是有利的 [665]。欧洲的专家更是强烈建议这样做：版权法之所以被容忍，是因为它们并没有大面积地对各种小型违犯者进行强制处理 [610]。即使好莱坞从美国法院得到了它们希望得到的一切也没有多大意义，相同的事情如果出现在欧洲就不会得到相同的结果。在欧洲，版权法仅仅针对那些为了创造兼容设备而进行的反向工程活动，其视频租用条约对于临时拷贝是予以保护的 [666]。我将在第 21 章关于电子策略的讨论中再回到这一问题上来。

另一个要点（在 [491] 中有例子）是对于 DVD 的小规模拷贝是不划算的，因为家用的可录 DVD 光盘相对于那些预录光盘来说花销要大许多。而远东地区的大规模拷贝已经出现。起诉的真正原因在于对 CSS 技术的公开使得任何人都可以创造 DVD 播放器，同时又不用向 DVD 协会支付版税。

总之，DVD 遵循通常的模式：好莱坞出于害怕，拒绝发行其最优秀的电影作品；技术上的措施被用来阻止拷贝，但以失败告终；然后，再采取起诉的方式。一个通情达理的做法是希望影视制作者们最终会看到适当的拷贝并且不会给他们带来什么影响，而且同时他们还是可以利用销售 DVD 赚到很多钱。当然，拷贝现象还会存在，但是它也不完全是琐碎的行为。使用一个 DSL 调制解调器花费许多个小时来发送一个 4Gb DVD 电影给朋友，这时个人电脑硬盘容量就是个问题了。最终，随着所有个人电脑中的 CD 驱动器完全被 DVD 驱动器所取代，我们可以预见到可擦写的 DVD 将成为被广泛使用的存储介质。而且还可以预见到，到那时，无论使用何种新机制来防拷贝都会被挫败的。但是，我还可以预言在未来的 10 年时间里，摆在我书架上的视频卡带和 DVD 的数量还会是一样多的，目前我有大约 50 盘预录卡带和大约两打家用自录卡带，前者主要为家庭使用，而后者大部分都是和我工作相关的旧的电视节目。我保证行业会照此模式发展。

同时，人们还在进行不懈的努力来提高 DVD 的安全性能，这是通过在下一代播放器中装配基于版权标记的机制来实现的。这是一种有趣的机制，值得我们来看一看。

⊖ 这些错误在安全数字音乐倡议（secure digital music initiative，SDMI）中也同样出现，这是一种被提出的 MP3 替代方案。SDMI 使用加密过的音频流，然后在个人电脑操作系统的声卡驱动软件中进行解密操作。同样，这也需要一种水印机制。然而，这同样剥夺了 Linux 使用者的使用权（也许包括了世界上大部分的计算机科学和工程人员），他们只有通过挂载 Windows 操作系统才可以访问最新的音频信息，这势必引起众多的具有相当能力的用户的反对。一种可靠的水印机制，即回波隐藏已经在 [610] 中被 Fabien Petitcolas 所破译。

20.3　信息隐藏

好莱坞对于发现保护版权新机制的兴趣主要集中出现在 20 世纪 90 年代中期。当时由于军方通信的需求以及公众关注政府在控制加密上的进展，导致了信息隐藏技术的快速发展。这项技术在很大程度上涉及到数据隐藏在其他数据中的技术，例如，在一首 MP3 音频文件中隐藏一个秘密消息，或者嵌入在某些执行命令中程序序列号等等。

好莱坞所关心的是版权标记，该标记可以在数字音频、视频和其他类型艺术品中秘密隐藏起来。这类标记或者是水印，即被隐藏的版权信息；或者是指纹，即被隐藏的序列号。

其保密性主要存在于隐写术中，该过程的目的就是要将信息隐藏在某些保护介质中，从而保证其不容易被发现但又确实存在。西门子［700，707］提出的一种通用概念模型如下所述。Alice 和 Bob 都被关在狱中，并在策划越狱行动；他们所有的通信都必须通过监狱看守人 Willie；如果 Willie 发现了任何加密消息，他将把 Alice 和 Bob 关禁闭，同时阻止他们的越狱计划。所以，Alice 和 Bob 必须找到某种隐藏他们的秘密消息的方法，而这种消息的传送从表面上是看不出来的。由于考虑到最新的密码技术领域所取得的进展，我们不妨假设监狱看守人员对于所使用的加密机制了如指掌，所以他们通信的安全性只能依靠 Alice 和 Bob 通过某种方式共享的加密密钥上面。

这与电子战领域有某些相似之处。首先，如果加密被看做是破译可能性极低的通信方式的话，那么版权标记和刚才讨论的困境抵抗通信技术很相似：它使用相同的方法，但是为了抵抗被发现的攻击，它在通信时具有很低的比特速率。我们可以把 Willie 看做是盗版者，它试图破坏音频或者视频信号，从而使版权标记识别器工作失败。第二，诸如直接时序扩展频谱这类技术最初是为了电子战而设计的，但后来发现其在信息隐藏领域也有着广泛的应用。

当然，版权标记并不是必须被隐藏才可以发挥作用。一些电视台将他们的台标显式地嵌入画面图像的角落上，而且许多 ECMS 系统也显式地将控制标记和内容绑定在一起。在许多实例中，这都是一种适当的技术。然而，下面我将集中讨论关于隐藏版权标记的问题。

20.3.1　DVD 标记概念

DVD 协会的当前目标就是找出一个版权标记方案来增强对连续拷贝的管理。视频或者无标记，或者标记为“无法拷贝”，或者标记为“只准拷贝一次”。兼容播放器无法对一个标记为“无法拷贝”的视频进行录制操作。而且，当录制一个标记为“只准拷贝一次”的视频时，将会使该视频的标记变为“无法拷贝”状态。商业化销售的视频应该标记为“无法拷贝”，而电视广播和其他类似媒体的视频应该标记为“只准拷贝一次”。通过这种方法，对于消费者来说，可用的 DVD 播放器对于家用视频可以进行无限次拷贝，对于电视节目的拷贝就要受到一定时间条件的限制，而对于商业化私有信息来说就无法胡乱拷贝了。关于这些提出的机制的概要性描述参见［119］。

基本概念很简单［504］。对于每一个碟片，选择一张票据 X，它可能是一个随机数字，加上拷贝控制信息，再加上对于物理介质来说独一无二的可能信息，例如轨迹引入线的不稳

定度等。使用单向哈希函数 h 计算 h（X），然后计算 h（h（X））嵌入视频中作为隐藏版权标记。这样兼容机将查找水印，如果找到了水印，那么除非提供 h（X），否则播放器是不会播放任何信息的。机器通过对 h（X）进行哈希操作并和标记做比较来检验是否正确。最后，如果仅仅提供了 X 值，那么兼容设备将记录一个被标记的轨迹。在这种情况下，h（X）将被写到新碟片中。通过这种方法，原先媒介中的“只准拷贝一次”标记在新媒介中将变成“无法进行拷贝”的标记了。

拷贝管理使用嵌入标记，这比附加数据强许多，该方法具有保留数字到模拟的转换以及反向转回数字的优势。这将导致许多问题的出现。首先，我们需要一种方法在音频或者视频中嵌入标记，即便会花费许多努力才能实现也必须做到，该标记必须易于识别但很难被攻击者擦除。第二，对于市场需求大的设备必须提供检测机制，使用价格低廉的处理器或者门电路数量有限的定制硅芯片，而且对错误的误报率要很低［554］。例如，如果你的正版 DVD 播放器错误地在你的结婚典礼视频中发现了一个标记，你将不得不购买一个盗版播放器来观看该视频。

20.3.2　常规信息隐藏技术

信息隐藏技术可以追溯到比加密技术更加久远的年代，它源自伪装。也许，最早提出该概念的是 Herodotus，他记录了在希腊和波斯的战争中使用的许多诡计，包括在猎手携带的野兔胃中隐藏信息；将信息刺青到奴隶被刮过的头上，然后再让这些奴隶的头发长过肩膀以隐藏信息；还有就是将信息写在写字板蜡层下面的木制底座上等［377］。弗朗西斯·培根提出了一个在书中嵌入二进制信息的系统，该系统对每个字母的一位在进行两种字体之间转换来实现［607］。直到最近，大多数作者才把隐藏秘密信息看得比加密信息更加重要［805］。军方机构仍然持这种观点，而且使用了各种各样的技术，从间谍在 20 世纪使用的微缩照片技术到第 16 章讨论的破译率极低的无线电技术。

当在其他数据中隐藏数据时，有关的一些术语如下［614］。版权标记（在加密学中被称作嵌入文本（embedded text））被隐藏在隐蔽文本（cover-text）中，从而产生标记文本（marked text），在加密学中被称为加密文本（stego-text）。大多数情况下，在这个过程中需要使用附加的秘密信息，这是标记密钥（marking key）或者加密密钥（stego-key），它们的一部分功能是用来恢复标记或者嵌入文本。这里，词语“文本”可以用“音频”、“视频”等等进行适当替代。

关于嵌入方案提出了许多建议：

- 一种明显的技术是将标记或者秘密信息隐藏在音频或者视频信号中不太重要的比特中。许多公共域加密工具可以做这项工作。但是，这通常并不是一个很好的策略，因为被隐藏的数据很容易通过统计的方法检测出来（那些不太重要的比特与图像的其余部分不再相关），而且，擦除或者替代它过于繁琐。在使用有损压缩技术时，这种信息会被严重破坏。
- 一项很著名的技术是由秘密密钥来决定标记或者秘密信息的隐藏位置。这种方法首先在古代中国被发明。发送方和接收方都有屏蔽纸的拷贝，屏蔽纸上面有许多小孔，用来对随机位置进行屏蔽操作。发送方将屏蔽纸放到一张空白纸上，将信息只透过屏蔽纸写在有孔处，然后将屏蔽纸去掉，在纸上填入伪装信息，这些信息中包含了

秘密嵌入信息的各个字符。这种技巧在 16 世纪时被意大利数学家 Cardan 彻底改造，并且成为了密码学家都知道的 Cardan 格 [428]。

- 一种现代化的信息嵌入隐藏技术实现方法是将版权或者其他信息隐藏在 .gif 格式的图像文件中。一个秘密密钥被扩展成为一个密钥序列，它选择适当数量的像素。嵌入信息就是用像素颜色代码的奇偶性表示的。在实际中，即使是图像中占相当大数目的像素也可以改变自身颜色来得到一个从视觉上无法识别的类似图像 [413]。然而，如果所有像素都进行这种变换，那么被隐藏的数据很容易就可以通过再次变换清除掉。如果伪装图像和嵌入信息的比例控制在 10% 的话，就应该得到一个不错的结果。然后，如果管理员重复这一过程，但是使用不同的密钥，那么将有 10% 的像素被变换，其中只有 10% 的秘密数据被破坏。
- 通常，随着有损压缩而出现的噪声或者变形将导致一些错误的出现，这与采取的嵌入方法无关。要避免这种错误，就必须加入某些错误校正代码。一个产生钞票标记的系统 Patchwork 使用循环码技术。其中关键是选择出像素的两个子集，一个子集通过增加发光度来标记，而另一个子集的像素则采取减少发光度的方法来标记。这内嵌了一个单一比特，钞票采取这种办法来标记上水印，如果不这样做可以参见 [96, 357]。在通常情况下，人们希望嵌入更多信息，而不仅仅是一个比特，可以利用这些内嵌的数据来使引发的错误在很高的级别上可以修复。所以，一种通常的技术就是使用从电子战中借鉴来的直接时序扩展频谱技术 [748]。
- 扩展频谱编码通常在一种变换空间中完成，它尽量使得变化不易被察觉，而且不会由于通常的压缩形式而产生不应发生的结果。这类技术也通常用来与感光过滤技术一起使用，该技术强调编码通常应该在噪声最大或者感觉上最重要的图像或者声音轨道部分进行，这些地方是最不显眼的地方，而且不强调在音乐中安静的段落或者图像中颜色大面积相近的区域中编码信息 [127]。
- 一些方案使用特定媒体的特性。例如，在标记印刷媒体的方案中，将文本线上移或者下移 1/300 英寸的距离 [135]，或者在音乐作品的感知阈值下方增加额外的回音 [96]。到目前为止，这种技术还很不健壮，而基于使用变换空间、扩展频谱和感光过滤技术的嵌入密钥技术还是主流应用技术。

版权标记和隐写术在 20 世纪最后几年中发展得非常迅速。它的历史就是重复密码学的发展历史，但是其推进的时间间隔更加短暂：人们发明标记方案，然后另一些人破译它，最终技术变得更加成熟和健壮。

20.3.3　对版权标记的攻击

贯穿本书，我已经描述过对于加密系统的攻击，偶尔还包括一些密码分析学。但更多的是基于错误的假设、保护错误事物、协议错误和偶然的实现漏洞等。版权标记也没有什么区别。

- 起初，许多人认为最大的版权标记市场将是水印，即嵌入隐藏的版权信息，从而使得作品的拥有者可以在法庭上得到证明。这已经被证明是一种错误的想法。聪明的财产律师们几乎从不会困惑于证明一件展品的所有权问题。而且，他们不会依赖技术措施来解决问题，因为这种做法可能会让陪审团感到迷惑不解，而是依赖于文档，

诸如和乐队或模特签订的合同书等等。版权标记在法律中的用途应该是指纹，也可以把它称作隐藏系列号。

- 首家大型标记系统供应商 Digimarc 创建了一种业务来从网上跟踪知识产权。它已经显示出了其潜力，因为对于多媒体产品来说，一项主要的开支就是跟踪大量的图像版权信息和应属于图像版权人的版税。然而，Digimarc 系统很容易被攻击者破坏掉，这些攻击者可以通过猜测系统主口令或者改变标记软件，从而覆盖掉已经存在的标记来达到攻击的目的。他们还使用“Marc 蜘蛛”，这种程序可以搜寻网络来寻找被标记过的图像，然后报告给版权所有者，但是在 [610] 中也有大量的方法可以阻止它们正常工作。
- 许多标记就是通过简单的附加机制得到的。这导致出现许多可能的攻击漏洞。例如，如果视频中的所有帧都携带相同的标记，那么可以通过算出它们的平均数，然后通过减法将标记算出来。一个更加简单的攻击是向标记系统提供某些已知内容，然后比较其输入与输出，就像在某些加密系统上使用一些选定的文本，进而进行攻击的道理是一样的。如果一个图像 P，携带标记 m，仅仅采用 $P+m$ 的方式，然后，一个使用 m' 作为标记的竞争者通过简单地声称源信息为 $P+m-m'$，就可以让原先的 $P+m$ 图像变为现在的用 m' 标记的图像了。
- 通常，许多设计者都忽略了 Kerckhoffs 定律，即系统的安全应该取决于密钥的选取，而不是所使用的算法。但是当标记作为证据使用时，这一规则比起其他手段来说更具说服力，因为需要在法庭上透露密钥。实际上，由于标记密钥需要被透露，使用多重密钥来进行保护也是十分必要的。例如，某人可能有一个标记并带有一个秘密密钥，这些都可以在系统范围中使用，而且可以用来鉴定哪位用户重新销售了与其许可权冲突的受保护的内容。拥有惟一密钥的第二个标记在该用户被法庭起诉时也可以公开使用。
- 出现过各种开发出一套与标记等价的公共密钥加密机制的尝试，以至于（例如）任何人都可以插入一个标记，而只有一个人可以检测到该标记；或者任何人都可以检测到该标记，但只有一个人可以加入标记。如果标记能够插入到正在制造的音频和视频中，那么前者看上去更加切实可行 [210]。后者适用于好莱坞这类特殊目的。然而，实际做起来要比看上去困难许多，因为存在一种十分普通的攻击手段。考虑到设备将检测到一个标记，攻击者能够不停试验，直到通过对图像进行小范围改动来将标记擦除为止，此时解码器无法再检测到任何标记了 [606，505]。
- 一些纯加密分析技术被开发出来针对特定的嵌入方案。例如，通过增加或者减少一小部分图像亮度的方式来添加标记时，将导致图像的亮度图出现两个峰值，这意味着通过大量图片标记可能被过滤出来 [519]。
- 另一类攻击充分挖掘特定媒体所具有的属性。例如，当典型的 Web 浏览器呈现一系列图像信息时，将一个接一个地无缝显示。所以，一个被标记过的图像通常被分割成一些小图像，它们合起来在网页上显示时就和原图像看上去没有什么区别。但是，在网页中版权标记是不能被检测到的（见图 20-6）[610]。
- 对于版权标记这一方案最普通的攻击手段包括适当地选择性扭曲失真。音频标记可以通过随机地重复或者删除声音采样来引入一些无法听见的抖动的方法擦除掉。使

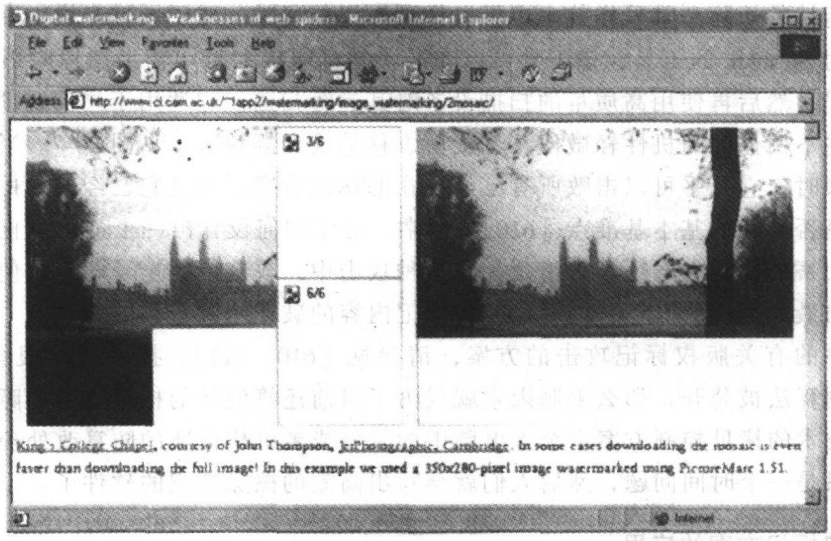
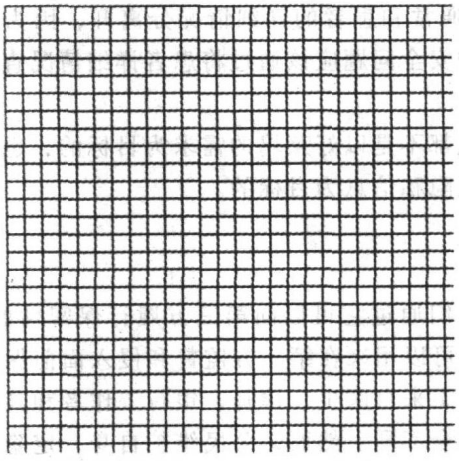


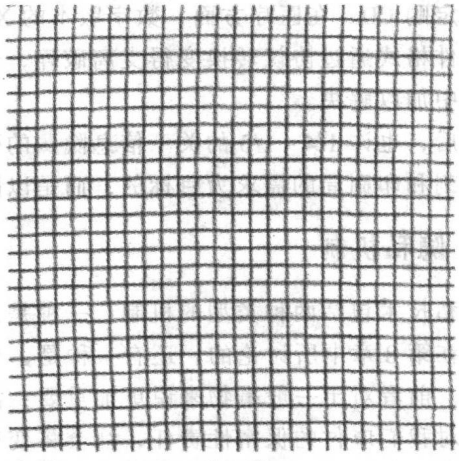
图 20-6　Mosaic 攻击（由 Jet Photographic 网站提供 http：//www.jetphotographic.com）

(a) Lena 的照片

(b) 经 Stirmark 处理后 Lena 的照片



(c) 底层网格



(d) 经过 Stirmark 处理后的网格

图 20-7　Stirmark 效果

用周期性的切除和重采样技术也可以做到标记的移除。对于图像来说，存在一个我们称之为 Stirmark 的工具，该工具可以在高质量打印机上打印输出图像时引入相同类型的错误，然后再使用高质量的扫描设备对其进行扫描。它适用于局部物理失真：图像被一些不明显的随机性轻微伸长、裁剪、移动或者翻转，参见图 20-7。当这种方法开发出来时，它几乎可以击败所有已经存在的标记方案，而且它已经成为用于判断版权标记是否健壮的基本基准点 [610]。通常，对于如何设计出一个可以防止选择性扭曲失真攻击的标记方案还不很清楚。在这种攻击中，攻击者完全了解应该使用可以对标记作出最大程度的破坏，同时又对被标记内容的破坏减到最小的方法。

对于完整的有关版权标记攻击的方案，请参见 [610, 611]。技术发展很缓慢，但是，一旦标记检测算法被公开，那么限制因素就成为了目前还算健壮的标记方案的障碍。如果任何基于标记技术的拷贝控制方案由个人电脑中的软件或者价格低廉的防篡改处理器实现，那么算法泄漏只是一个时间问题，然后人们就会写出高效的擦除标记的软件了。

20.3.4　版权标记方案的应用

标记技术的应用领域要比仅仅在网络上分发 DVD 和图片广泛许多。美国的音频广告通常被标记上一个序列号，使得审核机构可以自动检查电台播放这些广告的频率和他们所说的是否一致。美国销售的彩色复印机也将它们的序列号隐藏在拷贝的位模式之中，这可以作为发现流通领域伪造者的一种方法 [797]。显然，在新版欧洲钞票中都含有数字水印，该种钞票将很快取代目前欧洲的流通纸币；美国雕刻和印刷局也发出呼吁，希望在美国也采取类似的方式。

当大多数版权标记不得不被设计得足够健壮以抵制扭曲失真攻击时，一些用于标记的应用故意设计得尽可能脆弱。一项建议是在图像被应用后以高亮度显示其变化之处，而且可以在保证图像信息完整性时起到作用 [489]。另一个被建议的脆弱水印的使用仍将用于 DVD 拷贝保护方案 [119]。

有一类被提议的应用是与便利性和安全性有关的，而不再只是防止恶意的攻击行为。建议通过无线电的音乐广播应该被标记上 CD 号，以便于喜欢该音乐的人通过按下一个按钮就可自动定购 CD。在医疗方面，数字版本的 X 光通常和患者的详细情况分离开，因为各种私有的文件格式通过协议转换变得支离破碎，这种安全问题应该通过将患者详细情况直接嵌入到图片中加以解决。

最后，也许 1/4 ~ 1/3 的关于信息隐藏的研究都不是以好莱坞的需求为目标的，也不是以美国雕刻和印刷局的需求为目标的，而是以隐藏隐私信息为目标的。

20.4　隐私机制

隐私技术包括两种类型的机制：一种是，人们通过它可以向第三方履行保密性的责任，如我们在第 8 章中所讨论的；还有一种是，个人可以用来在第三方监督和侵入面前保护自己的隐私。前者对于一般事物来说更加重要：没有了来自医生、律师、银行工作者和其他服务提供者对我们应付的保密性责任，社会将变得完全不同。然而，公民维护自身一定隐私的能力也仍然是一种重要的支撑物，隐私机制在很多方面都具有重要性。为了了解这是为什么，我们必须调查一下这些机制可以给我们带来什么。

在前技术社会中，可以利用的保护机制不仅仅包括利用手工或是事先商量好的信号进行的加密，还包括两个人走到远处进行谈话，而其他人根本不知道他们说了什么。如果 Alice 声称，Bob 试图招募她参加起义暴动，而 Bob 总是声称他没有这样做过。事实却是 Alice 希望推翻国王的统治，Bob 极力反对这种不忠的行为。换句话说，一些通信是可以否定的。似是而非的拒绝否定仍然是今天一些通信方式的重要特征，从每天的生活中到最高级别的情报和外交中的通信都是这样。在某些情况下，它可以通过惯例或者习俗来实现：例如，在一些国家，一个参与起诉的人可以写标记为“不要有偏见”的书信给另一个人来提出一种解决方案，从而交换意见，这封信不可以公开作为证据使用。然而，人们在许多情况下没有这种清楚和方便的规则，电子通信的本质意味着“暂时离开一会儿”不是一个可选项。那么然后呢？

另一个问题是匿名。直到工业革命，大多数人生活在小村庄中，在那里，每个人都对其他人十分了解。对于许多人来说，搬到城镇住是一种调剂，那里大家都不彼此认识了解。目前，短语“电子村庄”不仅说明了电子通信手段让人和人之间距离缩短，同时也说明了人们对于任何事情都在线处理的担忧，因为每一笔交易的数据可以收集到，它们聚集到一起可以形成市场轮廓和销售状况，所以，我们好像又回到了类似 17 世纪的某个时期的状态上来了。关于我们的每一件事情都可以被知道。当然，如果你住在诸如德国这样的具有严格数据保护法的国家，那么你可能安全一些（只要你的所有业务都在德国进行）。但是，只要你在美国的一家在线购物站点购物，那么这种保护就不再存在了。有什么方法来谴责这些匿名的在线业务呢？

20.4.1　内容隐藏：PGP

众所周知且广泛使用的隐私工具就是电子邮件加密了。市场上的主导产品——Pretty Good Privacy（PGP）已经做了许多努力来唤起公众的关注，尤其当美国政府开始找寻作者 Phil Zimmermann 的麻烦，威胁他要对其进行起诉，因为他在网上提供可用的加密软件而违反了美国出口控制条款。

PGP 具有许多特征，但在其最基本的版本中，每位使用者都产生一个私有/公共密钥对：为了保护一条消息，你先用私有密钥标记消息的哈希值，将消息和签名用随机选取的会话密钥加密，然后将会话密钥用每位消息的合法接受者的公共密钥进行加密。因此，如果 Alice 希望发送一个加密邮件给 Bob 和 Charlie，她的消息应该是：

$$\{KS\}_{KB}, \{KS\}_{KC}, \{M, sig_{KA}\{h(M)\}\}_{KS}$$

密钥的管理故意留给了使用者，其基本原理是单一的中央认证机构将过于引人瞩目，这样一来，该机构将很容易被攻击或者受到法律的压制。所以，操作模式是每位使用者都需要收集他想要与之通信的其他使用者的公共密钥，然后将其绑定在公共密钥环中，该环在使用者系统中进行维护。公共密钥可以通过许多简单方法加以鉴别，例如将它们打印在使用者名片上面。为了更加简单，PGP 支持密钥指纹机制，它是公共密钥的单向哈希，以十六进制字符串形式表现出来。

另一种帮助使用者管理信任关系的机制是他们可以签署其他人的密钥。它可以简单地用于公共密钥环的完整性保护机制上面，但是如果签名可以被别人使用，那么将会产生更加有趣的事情。公开可见的 PGP 签名组成了可信任 Web：思想就是如果 Alice 希望得到 Bob 的公

共密钥，而她以前从来未与 Bob 通信过，那么她将可能幸运地找到有 Charlie 签名的 Bob 的密钥，以及有 David 签名的 Charlie 的密钥，而 David 是 Alice 已经信任的人。证书链的结果如下：

$$sig_{KC}\ \{KB\},\ sig_{KD}\ \{KC\},\ sig_{KA}\ \{KD\}$$

这可以等同于 Alice 寻找的信任关系，我们称之为 sig_{KA} {KB}，但是 Alice 必须承担一定风险，因为 Charlie 或者 David 可能不怀好意或者他们并不知道 Bob 的公共密钥。

还出现了其他发布 PGP 密钥的方法。其中被广泛使用的是一系列密钥服务器，它们包括了 PGP 密钥的庞大的集合。这里需要小心一些，因为，任何人都可以将密钥附加在任何电子邮件地址上面，地址是 president@whitehouse.gov 的密钥是不受人们控制的，但是你还会去和这类地址通信。有一本关于重要的公共密钥的书已经出版 [42]，其中也包含了一些关于早期 PGP 版本出现漏洞的信息。

当然，加密电子邮件只是解决方案的一部分。在一些国家中，包括俄罗斯、津巴布韦和英国，警方有权要求你对一些密文进行解密，甚至交出密钥。这种权力在一些国家的民事法院也可以通过传票来使用，还可以由许多税收权力机构使用。其他一些情况下，这种强迫将成为一个难题，包括抓获士兵或者间谍的地方；滥用警方权力的地方，例如以假设的犯罪调查为由没收其密钥，但实际上这些警察是被贿赂后来获取商业机密信息的；还有就是在被强盗严刑拷打后，被迫说出诸如银行卡秘密代码或者保险箱位置等信息 [793]。

在这类情况下，对于那些私有密钥生存期很长的系统就会出现严重问题。如果税收人员在对你调查的过程中获得了你的私有密钥，然后将该密钥放到某个服务器上与其他政府机构共享，这些机构就可以解密他们存储的关于你以前的任何消息，也许还可能伪造你的数字签名。

所以，最新版本的 PGP 具有分开的密钥对，分别用于加密和签名。

- 你的公共签名验证密钥是你让人们签名用的一种长期密钥，印在你的名片上，包含在你的签名文件中，等等。
- 你产生一组有时间限制的加密/解密密钥对，然后用长期签名密钥来标记公共加密密钥。
- 当密钥过期后，可以删除私有解密密钥。

美国的防御消息系统使用类似的机制。但是，它还支持使用短期公共加密密钥。每位用户有一个密钥服务器，该服务器一经请求就可以提供一个没被使用过的加密密钥，被用户的签名密钥标记。一旦收到消息并且解密，那么解密密钥就被销毁掉。

然而，对于密码学我们所能做的具有很大的局限性，许多传统的 IT 安全机制仍然可以危及到隐私的安全 [296]。使用密码可以为了流量分析而对你的消息做上标记；鉴别可以明确地识别出使用者，可以移去审查、诋毁和版权侵害的案例一些不稳定因素；在许多法律中，这些加密手段在警方拷问下就没有什么作用了，我们称为*橡胶管密码分析*（像英国这样的国家还是比较文明的，它们出台了一种法律，如果警察要求你说出密码而你拒绝，那么将把你送入监狱。我将在第 21 章讨论这个问题）。

20.4.2　内容否认——隐写术

当威胁模型中包括了武力强迫时，简单地销毁旧密钥的做法就不够了，因为，那些已经

存在的被保护的信息的正确性足以引起我们的怀疑。在这种情况下，使用隐写术只能提供更多的完全似是而非的否定。如果秘密消息被隐藏在不起眼的隐藏对象，诸如 MP3 音频轨道中，那么对方很难怀疑到这上面来，反而幸运地保护了秘密消息。

存储数据是相当困难的。大多数海关机构都有权力要求旅行者们解密他们的笔记本电脑硬盘中能找到的任何资料，以防止存在反政府或者色情资料等等。有许多初步的方法可以用来隐藏文件的存在，例如划分一个单独的分区运行 Linux，而海关人员根本不懂该分区的使用，但是对于一个有一定技术水平的对手来说，这种防御措施就无效了。随着时间的流逝，海关人员也需要合适的工具才能对付这些伎俩。文件可以通过隐写术工具隐藏在大的多媒体文件之中，但是这种做法效率很低。

这导致了对隐写文件系统的设计，该系统具有这样的属性，即用户可以提供一个对象名称，例如文件或者目录名称，同时还要求提供一个口令。如果所提供的口令与系统中该对象对应的口令一致，那么才可以访问。然而，对于没能提供正确对象名和口令的攻击者，并且缺乏猜测的计算权力，将得不到任何信息，甚至不能确定该对象是否存在。这是一种比 Bell-LaPaDula 更加强大的属性，低级将不能证明高级的存在性。使用者可以给海关人员提供一个低级口令，而不告诉他们高级口令的存在，海关人员将永远无法证明用户在说谎。

整个磁盘被加密，文件的碎片在其中分散开，而对于这些分散区域的访问必须提供相应的口令才可进行。同时，由于有时会出现低级用户意外覆盖高级用户数据的情况，所以系统还应该提供某些冗余方案来对数据进行恢复 [49]。一种早期的实现方式见 [536]。当然，一种真正健壮的实现必须考虑我们在第 7 章中讨论的多级安全问题，从隐蔽通道到限制恶意代码对系统带来的损害。对于隐写系统还存在着一些特别难于处理的威胁，例如，如果那些试图推算高级用户是否正在写文件的低级用户可以对系统进行连续快照，那么将会发生什么呢？这个问题目前还没有完全解决，还需要更加完善的实现手段。

20.4.3　联合隐藏——remailer 和译解密码者

然而，对于隐写术的使用也存在一些局限性。正如我在前面内容中提到过的那样，我们的对手常常通过流量分析获取大量信息。即使，Alice 和 Bob 间的通信被加密过，而且密文被隐藏在 MP3 文件中，甚至无论是检查 Alice 或者是 Bob 的笔记本电脑也都没有发现任何可疑的材料（这可能是由于信息被隐藏在隐写文件系统中或者被简单地记忆和删除掉了），但仅仅是 Alice 和 Bob 通信过这一事实就可以泄露秘密。这就是为什么罪犯们更加喜欢使用匿名通信（例如使用预付费移动电话），而不是加密技术的原因所在了。当然，这也有合法性使用问题，例如使用匿名热线服务电话来辱骂受害人、揭发者、警方线人的做法，以及抗议设圈套陷害当前政府的组织。还有一些针对大学教授的匿名学生反馈、讨论会上匿名投票，以及 HIV 测试，你可以通过使用一次性口令来购买测试工具箱获得在线结果。你可以申请一份目前老板发现不了的工作，和那些不使用加密机制的人交换私人邮件，或者和有害的趋势进行斗争。保护隐私也成为了一件很容易的事情，尽管在世界上有许多收集和交易个人信息的业务存在。所以，我们如何才可以在线保证匿名机制呢？

有两种基本的机制，它们都是在 20 世纪 80 年代由 David Chaum 发明的。第一种机制是 mix 或者叫匿名 remailer [177]。这是一种接收被加密消息的设备，解密消息，然后将它们重新发送到在消息内部发现的地址上。在其最简单的形式中，如果 Alice 希望发送一条匿名电

子邮件给 Bob，而途中会通过 Charlie 和 David 的话，她应该如下组织消息：

$$A \rightarrow C: \{D, \{B, \{M\}_{KB}\}_{KD}\}_{KC}$$

Charlie 现在去除最外边一层包装，发现了 David 的地址和一个密文。他将密码发送给 David，David 解密后发现 Bob 的地址和一个密文。他再将密文发送给 Bob，Bob 解密后得到了消息 M。当然，匿名 remailer 可以被法律实施机构或者情报机构作为很具有吸引力的“蜜罐陷阱”来操纵，所以，通常的做法是通过许多后继的 remailer 来发送消息，而且为了破译该消息，需要组织许多信息才可以实现。

对于这种基本的技术有许多明确的表述。为了防止对手通过一个接一个的 remailer 来跟踪消息，通常情况下消息大小是固定的；remailer 经过一个随机的天数后进行批量处理或者转发消息；消息重放可以检测到。一些消息允许回复到不明目标地址处，而其他的消息则不允许这样做；匿名回复也可以通过一个假名服务进行处理 [531]。

匿名连接并不仅仅局限在电子邮件当中，还可以包括任何类型的通信服务：一个实验性的美国海军系统，叫做 Onion Routing（因为消息被嵌套，就像洋葱中的各层一样），该系统可以被当作一种通信原语使用，在它上面，像邮件和网络访问这种服务可以被分层实现 [637]。还出现了一种为 ISDN 数字电话匿名网络设计的技术，它也许可以令人信服地构建在第三代移动通信服务上面 [312, 419]。匿名通信信道的存在确实大大简化了那些更加复杂的、具有匿名需求的服务的设计工作，例如选举和电子货币 [708]；而且，在真实世界当中，它们能够通过非加密方式（例如使用访问令牌的广播方式或者其他低开销的便携设备）来有效地实现 [732]。

当基于 remailer 的匿名通信提供依赖于各种实现的保护机制——例如重放或者其他封闭流量的攻击——还存在一种更加强大的机制，它并不具有如此的依赖性，因此被认为是一种“无条件安全”的匿名等价物。Chaum 还把它称为*译解密码者就餐问题*，之所以这么称呼是由于他受到在 6.1.4 节讨论的分布式系统中的“哲学家就餐问题”的启发。

一些译解密码者围坐在餐桌旁吃饭，服务员通知他们饭菜费用已经被一位匿名的赞助人支付清了，该赞助人可能是他们当中的一位或者是 NSA 中的人。这些译解密码者希望知道他是谁。所以，就餐者中每两个人都共享一个一次一密通信，在通信之后，每一位就餐者都要提供一个输出“我付了钱/我没有付钱”位的功能，而且每个人可以计算出所有这些位的总奇偶性如何。只要没有多于一个的译解密码者说“我付了钱”，那么即使没有人可以确切说出是谁付的钱，也可以得出这样的结论，即偶数代表是 NSA 付的钱，而奇数代表是他们其中一人付的钱 [179]。各种各样的扩展延伸也被提出了，包括“药商就餐问题”，它可以在拍卖可卡因时隐藏投标人的身份，使其不被其他投标人或者卖者所知。除了买者和卖者之外，没有人知道是谁在拍卖中胜出；而且即使是卖者也不能够在执行实际买卖之前发现最高投标人的身份 [732]。

要想适当地进行匿名处理是十分困难的事情。正如我们在上面提到的，匿名 remailer 自身也许就被非法者所控制。一种选择就是从那些主要业务是提供匿名服务的公司购买服务，其中最为先进的就是 Zero Knowledge System。对于这类公司而言，如果被发现不诚实或者对于它们所选择的行业不具有竞争力的话，那么该公司将失去很多东西。另一种选择就是使用由 cypherpunk 或者重点大学中的某研究组操作的 remailer。即便是这样，也还是有可能存在潜在的问题。各种类型的攻击手段，包括选择流量插入，它可以允许强有力的对手（即那些可以

在因特网中很大范围内监控流量的人）跟踪通信者之间的关系［359］。甚至更多的拒绝服务攻击也可能在 remailer 上出现。那些希望服务被关闭的人可以发送大量的垃圾邮件到该系统或者让这些垃圾信息通过该系统；他们可以通过高容量的邮件列表制造邮件循环；他们甚至是那些正被法院传讯的人。对于这类攻击者最好的解释来自于 David Mazières 和 Frans Kaashoek 的运行 MIT 服务器的经验［531］。

另一种可能性是，邮件或者 Web 转发功能被使用者而不是一种中央服务所承担。Crowds 是这样一个系统，在它当中，用户组织到一起并为彼此进行网页转发。通过这种方法，如果他们中的一个人下载了具有反政府的、破坏性的网页，那么秘密警察将要对付许多的怀疑对象［641］。一种类似的方案被某位著名的 CEO 所设计，该 CEO 每天早上总是随机地从其接线总机中接入一位管理者的移动电话呼叫。

出于许多的原因，详细描述技术保护机制是不需要的。现在，有许多在线服务，它们可以让人们匿名浏览网站，例如 Anonymizer［52］。使用者可以创建一个会话，在会话中使用这些服务以及输入他们希望获取网页的 URL 地址；当过滤 http 协议中诸如 cookie 这些部分时，匿名服务将做这些事情，cookie 可以暴露客户的身份。一些服务提供加密会话，在一些情况下收费会很高昂。实际上，任何 Web 缓存都将提供某种级别的匿名机制，因为网页的获取基于使用者的利益。然而，这就意味着缓存将可以包括一些令人十分感兴趣的日志！

实现高质量的匿名机制是十分困难的，这不仅仅由于这里讨论的所有原因，还包括那些在 19.7 节中讨论过的原因：商业网站从来都是创造新型的缓存手段来保证消费者看到它们的广告，从而让他们使用其所提供的服务，有许多手段都能通过这种或者那种方法来破坏匿名机制。关于匿名服务的调查，请参见［524］；对于使用 Web 重定向、Java applet 等等方式破坏匿名服务的讨论，请参见［716，654］。尤其困难的是强行更改一些浏览器的内部状态，例如 Internet Explorer，所以必须十分小心，不要让对方获得对你的个人电脑的控制权。

然而，最常见的匿名服务是网吧和随时可以被丢弃的基于 Web 的电子邮件地址。许多地方都提供每小时固定费用的网络接入服务，而且还提供许多免费的电子邮件账户，它们可以通过浏览器访问，通过打广告来支持，而没有任何用户要求的身份验证机制。这些服务当中的一些提供 SSL 加密访问，从而提供附加的隐私保证。对于隐私十分敏感的情况，结合使用这两种方案是一个十分具有吸引力的建议，因为，倘若你支付现金，那么在你所代表的电子角色和本人之间的所有联系都将不会存在持久的记录。当然，这种服务偶尔也会被滥用。在英国，当一名新纳粹分子在一家伦敦的网吧中下载了炸弹制造信息，并且在黑人和亚洲人居住地区以及一家酒吧制造了爆炸，造成了 3 人死亡和 70 多人受伤后，曾经出现了公共告警。但是，仍然不清楚的是，他从匿名机制当中实际得到了什么［191］。当然，网吧和随时可以被丢弃的电子邮件账户对于合法目的是有用的。

20.4.4　联合拒绝——数字货币

即使你使用一种随时可以被丢弃的电子邮件账户在网上进行购物活动，也通常意味着需要提供一个信用卡号码。正如我在电子商务一章中所提到的，那些商人通常会为你创建一个销售记录，而该记录就是以你的信用卡号码为索引的（尽管这种做法破坏了银行业务运行的标准条件）。你还可能幸运地获得一张标识有虚假姓名的信用卡（这甚至可以是合法的，例如在美国为一家情报机构工作，或者在英国，由于某种原因受到警方保护时）。但是，这仍

然不能阻止将你所进行的交易与为市场商人提供服务的特征数据链接起来的情况。

这导致了如下问题，即是否存在一种货币的电子等价物，也就是说是否有一种匿名的、难以追踪和无链接关系的支付方法。对此曾经做了各种的尝试。一些电子钱包提供商们声称，它们的产品是匿名的，因为钱包本身只具有一个序列号，而且在该钱包和消费者姓名间的链接关系只有发行银行知道（在这些银行当中，有一些由于与标准机构交易信息已经陷入麻烦之中，因为这些权威机构的要求没有被完全满足）。在这里，最能引起大家兴趣的保护概念就是数字货币，这是 Chaum 的另一项发明［178，180］。

在第 6 章中，我解释过数字货币底层的技术思想，即盲签名。消费者依据标准的格式创建一张钞票，再增加适当的随机隐蔽要素后，将其交给银行来获取签名。签名做完之后，隐蔽要素可以被清除掉，只剩下一个数字货币，或者更精确地说，是一个数字出纳员的支票，银行不知道该支票的序列号。还需要额外的特征来保证银行可以察觉是否某个货币被使用了两次［180］；一个现代的电子货币系统在［134］中详细描述。

电子货币已经被尝试过了，但是到目前为止在市场环境中仍然没有取得成功。第一家公司 Digicash 是将其作为产品发布的，最终该公司以破产告终。对于应用的研究一直在继续。目前，已经出现了一些引导性的项目，例如道路收费项目；其他可能出现的应用就是对私人在线电子商务中假名机制的管理［134］；还有就是医药保险方案将采取匿名保健信用卡来保护患者们的隐私［117］。

这些系统背后的基本思想就是消费者与商家之间的关系仅仅能被消费者一方所暴露（例如，通过出示收据）。这些系统需要一个匿名通信系统作为它们基础设施的一部分（否则，商人可以直接读取消费者姓名，例如从电子邮件头部中读取）。这样做可以限制商家们对于电子商务所产生的兴趣，同时也使得代价更加昂贵。还有内在的限制。例如，如果一项在线交易包括商品的配送，那么必须提供一个送货地址。如果产品是无形的，例如软件或者音频，那么版权拥有者也许希望通过某种方法，在你广泛分发拷贝时，对你进行收费。所以，对于电子货币技术的最终使用也许限制在一个封闭的应用中，例如道路收费系统。相关的技术可以被使用，从而保护那些电子大选中的投票者。我将在下一章中重新谈论这个主题。

20.4.5　其他应用和问题

对于元信息的控制，以及匿名和拒绝应用，形成了许多由其他应用产生的相关问题。

20.4.5.1　保持无知的权利

在自动系统中，最难于保证的一件事情就是，无论是采用这里讨论的机制还是采用我们在第 8 章中讨论的机制，如何确保具有不知晓某些事情的权利。一个著名的例子就是，在许多国家中，你有权不了解证明你的亲戚得了遗传病的一份 DNA 测试结果。你的亲戚有权知道这些，而且他可以告诉其他人。理论上，他可能告诉世界上的每一个人。这不仅仅是一个技术上的问题，而且还符合许多国家的数据保护法律［741］。

20.4.5.2　定位安全

在电信系统的安全一章中，我提到过 GSM 中的定位安全机制，这是一个临时的移动用户标识或者叫 TMSI。这种机制使得警方可以相对容易地破案，而且，在一些国家中，电话公司关于移动用户定位的历史日志可以被警方所使用（在瑞士，当人们意识到这件事情的发生时，产生了政治上的不满）。在包括美国在内的许多国家中，现在已经通过了法律法规来

要求这些信息可以用于警方证据的提供，或者在警方命令下必须提供这些数据；还有进一步的需求，例如跟踪那些在紧急情况中进行的移动呼叫。第三代移动服务将提供精确到 250 米范围内的定位信息。而且，在欧洲，要求电话公司保留一年之内的电话定位信息至少看起来是合法的，从而在警方需要时提供这些历史数据。还有，许多业务都计划提供基于定位的服务，其市场定位例如“离你开车将经过的 McDonalds 地区的 Big Mac 距离 50c”。甚至还存在一些为验证机制所特别考虑的建议，例如，如果你处在一个特定的区域当中，例如军事基地，那么移动终端则只允许访问一个系统 [237]。

因此，在公共网络中提供真实定位安全服务的希望极其渺茫。这里并没有技术上的原因，从理论上说，一个人可以使用数字货币来支付网络访问费用，在 [456] 中有更加详细的描述。但是，考虑到商务和政策调整的压力，这不太可能发生。最有可能成为现实的是，用户彼此之间可以获得中等级别的隐私保护。

然而，定位隐私机制可以应用于嵌入式系统中，例如德国的道路收费系统，在这个系统中，数据保护法律禁止在车辆移出收费站时，保留该车辆的详细信息，除非该车辆没有支付道路费 [164]。当然，设计自己的保护措施的制度是始终向个人开放的，因为上面提到的商业人士可能每天都随机地借用不同的移动电话。然而，在缺乏这种极端措施的情况下，定位隐私的应用看起来在未来几年中还存在着许多困难。

20.4.5.3　对等与抗审查系统

如果不可能堵住匿名这个通道，那么你可以在不被抓到的情况下发送那些受版权保护的、亵渎神明的，以及损害名誉的信息。这正是匿名、版权、审查和公民自由这几个问题之间的核心所在。

一个早期的匿名 remailer（anon.penet.fi），随着科学论派采取的法律措施而被关闭。该网站曾经发布了一条后来令它们卷入纷争中的消息。这条消息包括了一份前教会牧师的宣誓书，作为了主要的申诉原因上报。该宣誓书主张一旦成员们被传授教义，那么其他人种则会遭受虚假意识的控制；以及在现实中，耶稣是坏的，而魔鬼才是好的。在历史上，有许多关于宗教的示例，它们公然指责竞争对手是在蛊惑人心并且是邪恶的；而科学论派认为该网站公布的内容是他们的创新，是他们的版权，并成功地将诉讼引入到许多领域中。

因特网业界对此的反应已经包括了许多分布式文件存储中采用的设计方法，其中一些实现方式故意使用匿名机制来使得这种类型的审查机制更加困难。一个早期的方法是永远服务（Eternity Service），它被设计用来通过跨越网络的分布文件碎片来提供长期的文件存储，这些信息是被加密过的，以至于保存它们的人也不知道这些信息包含了哪些碎片，从而无法将这些碎片组装到一起。这时，重新组装的工作只能通过 remailer 机制 [27] 来执行。关于这一点的现代版本是 Publius[⊖]，它也提供了抗审查匿名发布机制 [785]。另一个后继系统是 Freenet，它试图提供通信以及文件存储服务 [189]。

但是，也许最为大量使用分散式文件共享服务的就是 Napster [570]。它使得网络用户可以在线相互共享 MP3 音频文件。在这里，文件并不是被集中维护，因为这样做会招致法律上

⊖ 对于非美国读者：当具有创新精神的 Alexander Hamilton、John Jay 和 James Madison 在编写 Federalist Papers 时使用笔名 Publius，Federalist Papers 包括了自 1787 年到 1788 年间纽约州报纸上刊登的总共 85 篇文章，这些文章使得那些选民确信应该支持美国宪法（United States Constitution）。引用了从美国权利到匿名政治演讲等广泛话题。

上的诉讼。Napster只是简单地提供一个索引服务，以便于那些需要指定曲目的使用者可以发现哪些人拥有该曲目，而且可以进行共享或者交易。除了来自于好莱坞的诉讼之外，Napster已经吸引了大约1000万~2000万的用户以及众多的效仿者（例如gnutella和mojonation）。考虑到MP3由于其文件过大而导致的网络流量，人们对采用隐蔽流量来传送它们很感兴趣，而且还将通过系统来共享其他的数据。现在其他的网络信息也许被秘密地编码到MP3文件当中，或者经过简单的包装使其看上去像是一个MP3格式的文件。

有两种文件使用发展的分支，即抗审查和文件共享，最近都已经同新出现的对等网的概念接合起来了。对等网是在2000年中期突然出现的，并且逐渐涉及到其他的问题，例如移动设备的特殊网络。

早期的计算机网络是多对一类型的（或者，如我们现在常说的客户—服务器类型）：许多终端连接到一台大型机上面。而早期的ARPANET则走向了另一个极端，在这里，每一个被连接的机器都同网络中的其他机器处于对等的状态下。随着ARPANET发展成为因特网，更多的层次和组织加入进来，首先是通过诸如DNS和telnet等服务，后来又出现了大型商业网站的广泛部署。到目前为止，Tim Berners-Lee对于网络所设想的人到人的通信机制已经逐渐变为一种客户—服务器模式的网络，在这种模式当中，人们的个人电脑对于大型网络服务器而言或多或少地更像是一些哑终端。

对等网常常被看作是回到了基本和原始的状态中。网络中的机器处于平等的地位，而且相互之间都可以通信。尽管该项技术的驱动力并不仅仅是版权和审查。网络的快速发展已经将DNS甩在了后面；目前，已经没有足够多的IP地址用来分配了。大多数拨号网络的使用者现在都是由其ISP分配临时的IP地址，所以那些可以让人们相互间对话的应用，如ICQ，已经开始发明自己的命名系统，尽管其功能只是局限在断断续续的连接状态中。

之所以对对等网络感兴趣的另一个原因是诸如蓝牙等特殊移动网络技术的出现。在未来几年当中，你也许将拥有一个包括许多设备的个人网络：管理系统、移动电话、心脏监视器、家用个人电脑以及防盗报警设备，它们当中的一些将可以和火车票分发系统以及办公室中的激光打印机建立临时性连接。对于这类网络的中央集中式控制几乎不可能实现（谢天谢地），目前的因特网基础设施（诸如DNS）也许不能妥善处理这种需求。看起来还需要用户使用多种基础设施，而不仅仅是一种，让这些基础设施适应实际的应用。一些基础设施也许是分等级的（就如ICQ一样），而其他的基础设施则可能为了达到最终的监控能力而采取分散方式（就如抗审查系统一样），还有一些基础设施也许只是临时性的（就如特殊网络一样）。

我们已经开始认识到，许多为像Eternity和Publius这样的系统开发的技术，以及那些为特殊网络安全性，如[731]和[732]这样的网络所开发的技术，可以为我们带来许多东西。这个领域正处于蓬勃发展的时期；不久前出现了一本论文集，其标题是“Peer-to-Peer: Harnessing the disruptive potential of collaborative networking（对等：管理协同网络中的分裂性可能）”，由Andy Oram编辑，O’Reilly出版（我获得这些论文最新拷贝的时间太晚了，所以不能将它们包括在本书的参考书目当中）。

20.4.5.4 颠覆团体处理

一个有趣的工程问题是，将这里讨论的技术集成到系统中，从而给系统提供更高的用来抵制各种监视和强制的能力，这种可能性究竟可以达到一个什么样的程度。这个问题被称作

颠覆团体处理（subversive group computing），而且被认为是有关颠覆团体的一组技术性需求。这里的威胁模型不仅仅涉及深层次的监视和确定的拒绝服务攻击，还包括团体成员经常性的反政府行为。

可以设想一个“隐蔽的超级高速公路”，它允许团体成员通过使用匿名机制彼此通信；传教所用的分布文件存储如果没有这种机制的支持，那么传教活动将被镇压；隐写文件系统帮助团体成员在被抓获的情况下显示出无害性；还有作为一个支撑来限制那些叛变的团体成员可能造成损失的一种细胞机制。这种假定系统也许被认为是某些机制的归纳概括，这些机制是为了使一组服务器能够抵挡和恢复来自于其成员的完整性故障，例如 AT&T 的 Rampart 和 IBM 的 Proactive Security，这些我们在 6.2.2 节中进行过讨论。

对这类技术产生显著兴趣不仅仅来自国家解放组织和反间谍活动机构，还来自于一般性公众政策的观点，因为它们将技术限制到隐私和监督的程度。而且，如果最近的历史可以起到任何指导作用的话，那些来自于由逃避、强迫和版权愿望所驱动的观点至少和那些来自任何特殊的政治解放议程的观点一样多。这些技术将很有可能被应用在更加广泛的犯罪活动中，但是正如 Whitfield Diffie 所说，“如果你为了自由而作战，那么你很可能发现自己是在一家破公司的酒吧中喝酒。”

20.4.5.5　滥用

最后，我们来看看更加一般的滥用类型，例如垃圾邮件、邮件炸弹以及一般性侵犯行为。在现实世界里，这种侵犯是由社会压力以及极端情况下的实物对抗引起的：你在一个不想购买其商品的销售人员面前可以关闭自己的房门。在现金交易中也是如此。如果你从市场摊位上抢过一个香蕉且不交费转身就跑，那么摊主可以在后面追着你要钱。

对于电脑空间中存在的差异，一个关键的情况就是对于安全和控制而言，这些物理方面的因素是不存在的，尤其是使用者在无法攻破的匿名机制的背后躲避起来，那么情况就更加明显了。出于此原因，许多人建议，在网络使用者们已经通过匿名提供身份保护，但随着法庭的命令又可以清除这种匿名的情况下采用一种称为*身份契约*的方案［155］。

一种持反对意见的观点是，考虑到人们很容易就可以得到随时能够丢弃的电子邮件地址，而且可以通过网吧电脑或者预付费移动电话登录，匿名将继续存在，而且偶尔会成为一种滥用因素；到目前为止，我们经历过一些真实世界中的滥用，其他机制很可能至少同跟踪犯罪者这种方法等效。

在研究滥用时，主动提供的商业电子邮件，即垃圾邮件，是我们应学习的主要老师。对于发送垃圾邮件的人来说，隐藏其产品的源地址以防止来自被激怒的网络用户的攻击是通常的做法。可以随时被丢弃的电子邮件地址是他们使用的一种技术；另一种技术是简单邮件伪造。已经采用了许多策略对它进行管理。许多高效系统使用一种称作诱饵地址的系统，该系统在新闻组和邮件列表中被公布出来，从而使得那些发送垃圾邮件的人将这些地址加入到他们的邮件列表中；那些发送垃圾邮件的源地址将因此被加入到黑名单中。此类系统中一个特别显著的变化是那些来自于不明主机的所有消息都会增加几个小时的延迟，是为了看看这些消息在诱饵地址中是否也同时出现［411］。

另一项技术是*速率控制*：一家 ISP 可以限制用户发送的电子邮件消息的数量。一个匿名服务提供者 Zero Knowledge System，已经决定抵制身份契约，而采用速率控制，再加上取消那些使用假名来发送滥用或者非法资料的邮件［821］（有趣的是，英国的 De Montfort 大学在其

防火墙处阻碍 Zero Knowledge 信息的通过，因为它不再执行阻碍色情资讯的策略，一个对于学术自由的争论正在酝酿当中)。这并没有产生严重的妨碍作用，因为匿名不能被反向跟踪到购买者，而且购买者还可以简单地去购买一个新的匿名服务。我们必须看看这在实际过程中是如何成功的。

垃圾邮件与保护机制在许多方面都存在着相互作用。很明显的一个方面就是，那些发送垃圾邮件的人有时使用 remailer 服务来隐蔽自身，所以，对于匿名服务来说，最常规的做法就是实现速率控制（这还可以使得那些通过向单一的伪地址发送大量数据流，从而查看数据流向何处的企图更加困难)。另外一种是反向邮件攻击：通过伪造来自于被攻击者的垃圾邮件（或者其他进攻型消息)，你可以使攻击者被许许多多消息所淹没，而发出这些消息的人正是那些曾接收到其发出的垃圾邮件的人。

最后，ISP 们在追踪实施滥用犯罪者时所遇到的许多真实的困难并不是由那些隐藏在 remailer 背后的人所引起，而是不得不处理真实世界所带来的问题。一个例子是在多个学生共住的学生宿舍中，当滥用流量来自于一根电话线时，决定责任由谁承担是很困难的。或者当某人访问一个免费 ISP 的拨号账户，而其呼叫线路 ID 被阻塞时。ISP 们通常也维护着足够多的日志，因此不久就可以追踪到滥用流量。所以，在实际生活中，与理论恰恰相反，匿名已经相当普遍了 [190]。

滥用，无论是技术上的、社会上的还是犯罪中的，都不会消失，减弱其影响和掌握犯人应负责任的手段将继续成为一个关注的问题。例如，英国 ISP 的观点是“匿名应该通过相关的工具给予明确的支持，而不应该像现在这样，使用与误用都可以”[190]。这些工具的细节设计在一段时间内很可能难以做到。

20.5　小结

21 世纪初期一些最为困难的安全工程问题必须能够处理版权和隐私。在没有防篡改设备的支持下，那些必要的知识产权的执行机制是否还意味着对于谁读了什么书、谁听了什么音乐以及谁运行了什么软件要进行详细监控呢？如果所用工具可以使得人们防止监控，那么这是否又暗示着大规模的盗版活动呢？

这乍一看好像是以前哲学家们关于无坚不摧的大炮碰上岿然不动的柱子这个问题的现代等价物，它们二者在宇宙中同一时间是不可能共存的。但是，通常情况下，事情并不是那么简单。摆在好莱坞面前的问题，以及那些保护某人的隐私空间不被侵犯的问题，都显得更加的微妙，而且解决办法将包含对于一系列工具的巧妙的结合。这些工具具有管理元信息的特性：消息源或其目的地，无论该消息有没有被支付，是否允许拷贝以及是否在一定时间后就可以读取该消息，等等。通过选择一种合适的机制，一些相当微妙的特性组合都可以被工程化。

这并不是说，在版权和隐私之间不会出现冲突，仅仅由于电影和音乐产业的要求就侵犯个人自由的政府没有能够理解这些问题，应该由司法部门对他们的立法进行纠正。还有，大规模盗版的本质不应该使得管理人员和策略制定人员偏离其严肃的真实世界中存在的弊端，例如垃圾邮件和一般性侵犯行为。

这看起来是一种适当的用来结束本书技术性部分的表述。第三部分将讨论策略、保证、经济和管理方面的问题。

研究问题

在版权管理方面存在着许多有趣的研究问题。它们中的一些我已经涉及到了，例如是否可以制造出价格更加低廉的防篡改硬件令牌。其他的问题将引起更加活跃的讨论，例如将版权标记嵌入到数字图像和音频中去的更好的办法。还有商业建模的问题，它们看上去并没有得到太多的关注。例如，我们是否可以重用流行的计算机病毒来找出各种内容隐私的经济价值究竟有多大。

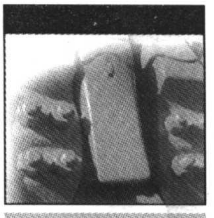
隐私也是一个研究和创新的活跃领域。也许最为困难的问题就是如何防止滥用匿名服务，尤其那些试图将这些匿名服务搞瘫痪而故意滥用它们的人们。

参考资料

软件拷贝保护技术在［356］中有详细的讨论；关于音频和视频保护机制历史的简要描述在［307］中；关于付费电视系统的攻击和防御发展过程参见［532］；更多关于付费电视的信息，以及 DVD 的可用信息可以在许多网站上面找到（有些可能由于法律问题已经取消）；从一个律师的视角来看待这些问题，请参见［361］。

这里是一个关于信息隐藏技术的概貌，包括隐写术和版权标记，在 IEEE 学报上有关于这些问题的专刊［515］；对于标记方案的攻击参见［610，611］。要得到更多细节性信息，有一本最近出版的书［443］。通常，Kahn 是一本很好的历史背景读物［428］。对于匿名 remailer 的介绍最好的一本书就是［531］。最后，在信息隐藏方面继续进行的研究工作可以在［28，59，613］中找到。

第三部分



在本书的最后一部分中，我将涉及三个主题：政治、管理和保证。考虑到目前我们已经具有了一些提供保护的思路和想法，所以这三个主题实际上就是：允许我们做些什么？我们如何来组织？我们怎么知道做到什么程度就足够了呢？

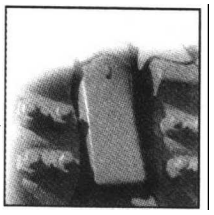
最近，存在许多有关密码系统是否应该使用法律的强制手段加以控制的争论。在美国和欧洲，关于密码系统的相关法律和政策措施的发展成为人们感兴趣的议论焦点，但是我将以很快的速度来回顾一下这个问题。这仅仅是冰山一角而已。

在其他一些地方，安全工程逐渐开始和政治产生冲突。在什么情况下，才可以将数字签名作为法律所承认的手段呢？何种措施可以切实可行地保护人们，使他们远离网络上的不适当资源（而且，究竟由谁来决定哪些资源是不适当的呢）？对于那些怀有敌意的努力所造成的信息战的威胁，商业系统设计者可以从中得到何种暗示呢？以及，个人隐私如何被保护？对于最后一个问题的回答，在美国与欧洲是截然不同的。在美国，是社团的“自我管制”（self-regulation）；而在欧洲，二战的经历使他们更趋向于把隐私作为一种很重要的事物来保护。在欧洲这称为“数据保护”，它预示着在这两块大陆上会存在主要的争论。继任的美国行政部门试图将隐私看作是“一件可以完成的事”或者是可以敷衍了事，仅仅好比是扫除地毯下的灰尘而已。并没有认识到德国人在数据保护上所表现出的不屈不挠，就像许多美国人对枪支控制采取的态度一样。

下一章的内容是关于管理的。在信息安全领域，这一词汇已经变成一个“肮脏”的词语了。那些关于“管理”的索然乏味的文章层出不穷，但是文章的内容却是空洞、泛泛而谈不具有什么意义。但是管理问题本身还是十分重要的。组织上的和经济上的动机常常决定了是否需要创建安全系统。许多系统的最终失败是由于事后才想起将保护措施追加到系统中；或者是由于系统真正的目的与原先设想的目的不一致；又或者是由于设计控制系统的人们并不是那些在系统崩溃后遭受损失的人。经济学为我们提供了许多知识，例如，安全工程通常伴随着不完整的信息，而网络客观性更加难以捉摸。对于剩余风险的管理，以及组织上的实践过程，也是经常导致严重失败的问题。

保证好比是一大罐政治蠕虫。实际上，它仅仅是一个工程问题。对于下列问题，我们如何找到令人信服的答案：我们所创建的是正确的系统吗？我们的创建过程正确吗？这些问题和软件工程（它可以教会我们许多东西）有些类似。但是，当系统暴露在怀有敌意的攻击中时需要一种新的解决办法。还有，大多数声称能够做到保证或保证经过认证的组织，都遇到这样或者那样的问题而最终失败。关于系统安全特征的声明通常是稍加隐藏的权力和控制力的声明，所以如果设备制造商、保险公司的实验室、军方机构以及理论攻击者的声明是相同的，就让人奇怪了。所以，对于安全工程师来说，在项目启动之初就不光制定系统的最终目标，还要制定如何判定其成败的标准，这是一件尤其重要的事情。

第21章 电子策略



经验告诉我们，当政府的目的是要获取利润时，更要最大程度地保护自由……对于自由，最大的危险性来自表面上热情但背后潜藏着阴险的侵犯意图的人，这在字面上容易懂，但你并不容易真正理解它的含义。

——（英国）最高法院法官 Louis Brandeis

在律师和工程师之间的争论就像生气的幽灵一样在两者之间飘过来，飘过去。

——Nick Bohm. Brian Gladman 和 Ian Brown [124]

21.1 引言

信息安全是一种关于权力的问题。它将决定什么人可以被授权或者被拒绝使用某种资源。在过去，它的含义没有经过权衡和检查，而且大家对此一般也是无异议的。银行以自己的而不是用户的喜好方式来创建系统，所以以失败告终；医院为了管理和研究的目的而收集患者数据，但却没有告诉患者本人这一情况；政府强迫电话公司将它们的网络变得更容易使用。但是从20世纪90年代中期开始，许多这类假设都出现了严峻的挑战。造成这一情况的因素包括用户关注程度的增加、IT在人们生活和业务中重要性的增加以及计算能力更加呈现分布状态的事实。使用价格低廉的计算机就可以对一些小型业务进行差额清算和核查利息计算，这使得对银行管理人员来说很难在已经协商好的利率上增加一到两个百分点；一旦家庭医生们开始为控制电子健康记录而和健康保险公司和医院展开竞争，那么后者对于医疗隐私的轻视行为就会遭到爆光；而且一旦普遍存在的个人电脑、电子邮件和加密软件使用监听装置来应对政府监督的话，那么该装置就会成为一个问题了。

政府在电子商务和因特网中的角色通常已经成为了许多争论的根源。在20世纪90年代的大部分时间中，争论的焦点在于由第三方保存的密钥契约上，其观点就是出于法律强制部门和国家情报局工作方便的考虑，加密密钥的拷贝应该由政府保管。坚持这种观点的政府有美国、法国、俄罗斯和1996年后的英国。持相反意见的有诸如德国、爱尔兰和1996年前的英国政府，以及几乎全部的IT产业界，它们认为最好是让因特网在技术和市场的驱动力下自己发展。

稍后我将简要地研究一下某些论点。然而，在21世纪初期，政府策略和信息安全二者之间的关系越来越紧密。在当前这个任何事情都可以随时处理或者在线处理的社会中，政府的服务如何进行组织呢？这些服务包括福利金、法院系统、护照以及税收等等。政府如何避免加深社会中对于穷人、老人和少数民族的排斥呢？这些人对于在线业务也许是最后参与的人群。选举应该在线进行吗？而且如果这么做了，如何防止欺骗行为、贿赂行为以及高压政治现象的发生，从而让这些现象和目前没有采用在线选举时一样少和困难呢？当政府工作自

动化时，我们是否将目前不够高效的冗余人员替换成自动的但同样也是效率低下的人吗？

政府部门，如同商业一样，也会努力维护其在网络空间中的领土。密钥契约只是早期争夺地盘时的一种方式，考虑到情报局是技术上最为精通的公共组织，所以它们这么做也就不足以奇怪了。在许多国家中，也产生了一些令人遗憾的影响，关于谁应该访问这些密码学密钥的争论不仅使政府和当地IT业的关系变坏，而且还使得一些机构控制了制定IT业策略的主导地位。例如，在英国，“国家技术权威机构”（national technical authority）在这方面的影响就引起了信号情报机构GCHQ的一个部门CESG的猜忌。所以正是由于策略使得所有的国家部门密钥都由第三方保管。这将在采用在线选举系统时引起严肃的关注。当然，这些机构想要知道在北爱尔兰谁投了新芬党的票，但是如果它们很容易就可以找出答案，那么北爱尔兰政府的合法性将遭到破坏，而且会比任何战术情报的失败导致更多的伤亡。许多人认为让间谍机构控制国家计算机安全策略无异于用狐狸来照顾鸡窝。但是，还有什么替代办法吗？对于这些机构应该采取何种政治控制，而且如何来控制它们的责任？在哪里以及如何在信息安全领域（一般来说就是IT）创建可以替代的公共部门专业中心呢？

所以，我们首先需要了解的策略问题就是监听、流量分析以及密码技术是如何被加以管制的。

21.2 密码技术策略

在过去的几年中，有关密码技术策略以及相关话题的文章写了很多。在这一节中，我适当地将这些争论放到上下文中加以讨论，抓住它们的要点，理清主要的消息来源。

虽然，对于密码技术的限制已经存在了许多年，而且极大地激怒了普通的使用者们，例如银行业。1993年比尔·克林顿新政府令IT业惊讶的有条件的*第三方保存密码标准*（Escrowed Encryption Standard，EES），即众所周知的Clipper芯片出台时，这些使用者提出抗议。这是一种被提议用来替代DES的标准，具有内置的后门密钥，使得政府机构可以解密任何数据流量（我在14.5.3节中已经从技术的角度进行了解释）。然而，Clipper作为一项技术，其重要性更多来自使密码技术和信息安全技术中加入了政治性因素。

美国方面的观点分为两个极端，其一是政府方面认为密码技术是用来维护消息秘密性的，那么罪犯就可以利用该机制来防止警方通过监听的手段收取犯罪证据；其二是IT业（有些例外情况）持有相反的观点，它们认为密码技术是惟一可以用来保护电子商务的方案，它对于网络未来的发展起到极其重要的作用。公民自由组织支持IT业的观点，并声称密码技术对于保护隐私来说是十分重要的。到1994年，国家安全局（NSA）断定它们将和Microsoft之间进行一场战争，而这场战争它们将会失败，所以就将策略领导权转交由美国联邦调查局（FBI）来控制，而NSA继续在幕后进行指挥。

争论很快就和武器出口控制问题纠缠在了一起，所采取的就是传统上对密码技术的控制方法。美国不允许软件公司在出口产品中包含过于难以破译的密码（通常这是通过密码长度超过40位的标准来判断的）。一位美国软件作者Phil Zimmermann由于编写了PGP系统，而该系统被放到了因特网上，所以被以从事非法勾当遭受陪审团的审讯并被中止工作。在审讯之前，他已经成为了民族英雄，而且利用该产品在市场的主导地位赢得了大量的财富。这种冲突变得国际化了：美国国务院投入了大量的资金和人力来说服其他国家也进行密码技术的控制。

结果是复杂的。一些国家一直采用压制性政权制度，例如德国和日本，它们对于美国的哄诱进行抵制。而其他国家，诸如俄罗斯，借这个机会通过了严格的密码控制法；法国则取消了传统的对于非政府使用密码的限制；英国在20世纪90年代中期在首相约翰·梅杰的影响下采取放任自由的态度，而在2000年时又在首相托尼·布莱尔的影响下采取了严格的法律，即调查权力管制法案（Regulation of Investigatory Powers，RIP）。

在这一过程当中，除俄罗斯和津巴布韦之外的政府所采取的措施都是逐渐向更加细致的方向发展。虽然对于犯罪来说，又出现了新方法，可以通过更加有效和合法的方式来进行抢劫。但是，总体上说来，关于密码策略的观念的斗争还会是一种光明（隐私提倡者和IT公司）逐渐战胜黑暗（警察和间谍）的斗争，这才是符合因特网的最初精神的。

通常，现实会更加复杂一些。所以，让我们将眼光退后几步，在历史的环境中看一看这些争论也是很有用处的。

21.2.1 警方窃听的历史

从最早建立国家时起，统治者就已经尝试控制通信了。在古代，这是通过由送信人对顾客包裹进行检查的方法实现的。从中世纪开始，许多国王或者将邮政服务授权给一个信赖的贵族，或者将该服务作为国家的产业。早期时髦的信件公开和代码破译设备，就是所谓的Black Chambers，在Kahn［428］中被描述。

在最近关于密码策略的争论中，最值得回忆的就是电子通信技术的发明所带来的防御性和返祖性。在欧洲大部分地区，电报业务作为邮局的一部分被创建，并且通常是由政府直接拥有。即便在不是这样做的地方，规章制度也过于严格，以至于产业发展受到严重阻碍，同时也使美国得到了竞争上的优势。国家规章十分丰富，有时，甚至它们自己之间也会产生冲突，这激怒了全欧洲的人。所以，在1865年时，终于成立了国际电报联盟（International Telegraph Union，ITU）［729］。在英国，电报业被一名叫做格拉德斯通的英国政治家于1869年变为国有化（这种做法给政府和商业都带来了创伤，以至于在英国下一个意义重大的国有化直到1945年后才开始）。

电话的发明进一步增加了政府在监督工作方面的兴趣。法律和技术两个方面的抵制，有一段很悠久的历史。美国最高法院在1928年时规定，经过研究和调查，监听行为并没有违反第四修正案（Fourth Amendment）的规定，因为，对于住处并没有造成物理上的破坏，对此法官Brandeis极力反对。在1967年时，法院又转变了想法，规定修正案保护公民而不是住所。第二年，国会规定对于那些有组织的犯罪行为，在掌握一定证据后所进行的联邦监听行为是合法化的（这是Omnibus Crime Control and Safe Streets Act的第三条）。1978年，在对尼克松滥用职权的调查之后，国会通过了Federal Intelligence Surveillance Act（FISA），该法案出于国家安全的考虑对监听手段进行控制。1986年，Electronic Communications Protection Act（ECPA）取消了原第三条的规定。到了90年代初，非常规服务的发展，从移动电话到呼叫转发，都已经开始破坏权威机构的监听能力，这些都是随着技术上的发展而出现的，例如带外信号和调制解调器中的适应性回声消除等。到1994年时，Communications Assistance for Law Enforcement Act（CALEA）要求所有的电信公司都将自身的网络改造成以FBI批准的方式来进行监听的网络。到1999年时，在1 350个法庭命令下，有超过2 450 000个电话交谈记录被合法监听［272，533］。相应的法律是为电信服务提供的18 USC（US Code）2510-2521［759］

(1984年的Cable Act控制了对电缆调制解调器的监听，而且更加具有约束性，所以管理者希望将其废除[439])。

必须注意的是，经过一些严肃的分析，目前至少被授权的监听和没被授权的监听数量一样多[250]。在一些国家中，这个数字可能并不准确，因为如果设备拥有者之一同意监听是无法被控制的，因此来自电话亭的呼叫对市场是免费的。

但是，无论官方数字是两倍或者三倍，很清楚的一点就是民主政权在监听的使用上要比独裁政权少得多。例如，美国1999年中经法律许可的搭线窃听共计63 243起，或者说平均一天中有超过173次的监听。对于前种统计，尽管前东德人口数量仅仅是美国人口数量的零头，但也有大约25 000起电话监听[295]。还存在其他技术监督措施的使用，例如房间窃听器或者人体窃听线路等(在这些国家中，裸体聚会很普遍，这并不奇怪)。

在高度民主的国家中，各种监听方式所产生的影响不足为奇。例如，在美国，只有大约一半的州使用它。而且这么多年来，大部分的监听都集中对纽约、新泽西和佛罗里达州的黑手党进行(虽然最近，宾夕法尼亚和加利福尼亚州也赶了上来)[372]。在欧洲也出现了类似的变化。不管荷兰人的自由主义在其他方面表现如何，监听工作在荷兰仍旧是相当普遍的，监听数量虽然只有1 000个，但其人口只有美国的1/10而已。例如，在这里的杀人案调查中，通常要对受害者地址簿中的每个人都监听一周，从而监视他们对于死亡消息的反应如何。在英国，监听行为是需要部委许可的，并且更加少见，但是警方在严重的案件中可以大量使用窃听器和类似技术。从某种程度上说，这些技术都是可以互换的。

监听的代价是一个严重的问题。在1993年CALEA引入之前，美国警务机构在监听上花费了5170万美元，也许在该问题成为政治性问题之前对其花费的估计更加准确一些[372]。即使没有覆盖ISP，CALEA的实现也需要花费5亿美元。这引发了一些策略问题。这样做到底值得吗？警务机构是否应该削减监听开销，从而在警力人员上投入更多的开支呢？或者他们希望通过逐渐扩展监听的能力来慢慢偿还这笔开支的代价？一旦你开始塑造的基础设施不是用来适应花费和效率的需求，那么就有一些人必须来支付这些费用，而且由于这种基础设施过于复杂，花费还会更多。

21.2.2 流量分析的历史

然而，在已经建立起民主制度的国家中，警方的通信情报并不来自对信息内容的监督，而是从对电话呼叫记录和其他通信数据的分析得到。我在电信安全一章中研究过罪犯是如何利用诸如预付费移动电话和攻击专用分组交换机的方法，从而在众多普通的流量中竭尽全力地隐藏他们自己的信号，以及警方用于跟踪犯罪信息的技术。

这并不是什么新东西。统治者们长期使用他们对邮政业务的控制权来跟踪那些潜在的、具有颠覆性意图的通信者，即使这些信件没有被公开。1840年，邮票的引入对于隐私来说是一个巨大的进步，因为它让匿名发送一封信件变得更加容易。一些国家对于煽动性和诽谤中伤的言论所带来的威胁十分担心，所以它们通过一系列法律来要求必须在信封的背面写上回信地址。另一方面，电报的发展对于监督工作来说又进了一步，诸如发信者、接收者和字数等信息都被记录下来，所以整个流量都可以被编辑，而且可以作为经济活动的一种有效的指示数据[729]。第一次世界大战中，指战员们成功地使用了通信情报来获取敌方的无线电波信号，即使当时无法轻易地对这些信号进行破译[428，569]。后者的难度又重新加强了

通信情报本身。

到了20世纪后期，流量分析技术支持了大量的警方电信侦察方面的应用。例如，在美国，1998年中有1329个监听应用被批准使用（在写本书时，这是最近的可用的比较数据），而同时对于pen register（一种记录特定电话线中所有拨叫号码的设备）共批准了4 886个许可，还要加上另外4 621部电话分机；对于trap-and-trace设备（用来记录呼叫线路中所有到来的呼叫线路ID，即使是呼叫者试图阻碍），共批准了2 437个许可，这也要加上2 770部电话分机。换句话说，对于通信数据的监控与内容监控相比，整整多了11倍。这种模式持续了许多年，而且发展到许多国家。为什么会是这样呢？

实施监听所花费的代价高昂，以至于大部分预算紧张的警力部门只是将它作为最后考虑的手段。相反，那些嫌疑犯呼叫的号码或者是呼叫他们的号码，让我们看到了他们的通信模式是那样得简单。还有，监听通常需要有严格保证的必要条件，大多数国家在警方使用通信数据的问题上面限制极少甚至毫无限制。在美国，ECPA出现之前是不需要任何许可就可以进行数据监听的。即使ECPA出现后，这项许可也很容易得到：在18 USC 3123［759］下面，调查机构的官员仅仅需要向法庭保证，通过监听软件的安装和使用而得到的信息是与一项正在进行的犯罪调查有关就可以了。这可以是任何种类的犯罪，任何一种联邦或者州法律中所规定的重罪或者是轻罪都可以。和监听不同的是，法庭没有权利否决一个形式上正确的应用程序，同时一旦命令被授权，也没有法庭监管。既然经历了CALEA的过程，仍旧需要诸如所发送电子邮件的接收方地址的信息，但是通过索取的方法，可以获得基本长途电话记录信息，这种索取方式并不需要发正式通知给用户。所以，上面关于pen register和trap-and-trace的数字确实低估了对于流量分析的法律实施程度。无论怎样，在18 USC 2703（c)的控制下，电话和电脑服务记录可以提供给除了法律实施机构之外的其他团体。因此，例如，我们发现弗吉尼亚州和马里兰州计划使用移动电话来跟踪数据，从而监控环城线的堵塞情况［710］。当这项法案通过时，国会就可以期望被用作市场目的的电话记录了。

在英国，在毫无控制的情况下，电话公司给警方提供成批的电话费记录情况，这种情况一直持续到2000年通过欧洲法律来强迫政府调整RIP法案为止。从那以后，通信数据只需要警方的高级长官向电话公司发出通知即可，不再需要许可证了。

对于通信数据访问的控制问题逐渐变成一个活生生的问题了。主要问题是通信数据和内容变得越来越混乱，在一个抽象层次上被称为通信内容，在另一个层次上就可能被看作是通信数据。一个很好的例子来自Web URL。实际上，一个URL就是一个网页的地址，但是诸如http：//www.google.com/search？q = marijuana + cultivation + UK包含了输入到搜索引擎中的条目信息以及搜索引擎的名称。无疑有许多警察希望获得提交这个询问的人名列表。同样，将这些数据提供给警方就如同在线演讲产生的急冷效应（chilling effect）一样。这样做在许多司法中都是违反宪法的。

实际上，当英国政府将RIP议案放到国会上讨论时，它就试图保护包括URL在内的一些存在的问题（否认这么设计是最初的目的)。警方可以无限制地访问每个用户进入的URL地址的消息引起了大众对于“Big Browser”的抗议，所以对于通信数据的理解又被重新定义。对于普通的因特网流量来说，目前是指IP地址，但是也包括电子邮件地址和移动电话位置等信息。所有这些信息都可以通过警方机构的长官发出的一个通知而获得。

其他国家使用不同的定义。例如，美国地区法院最近规定，移动设备所属的蜂窝单元已

经足以满足要求了，在移动设备中加入三角测量技术（这种解释正是警方希望看到的）将会侵犯隐私［760］。还有，尽管蜂窝粒度位置信息对于 pen register 许可证的低级标准来说也是不可用的。pen register 许可证也不足以发现 post-cut-through 拨号数字，因为没有一种办法能够事先区分呼叫路由器的拨号数字和访问或者发送信息的拨号数字。实际上，这意味着如果在美国，一个调查目标到一家自助食品商店花几美元购买了一张电话卡的话，那么警察无法知道他给谁打了电话，除非警方获得一个完全的监听许可才可以这样做。给予警方的权利仅仅可以让他们获得嫌疑人联系电话卡制造商的拨号数字，而不是那些在连接以后所发出的拨号数字。

不同国家对于什么是内容、什么是通信数据的不同观点一直在扩大，从而对政治和工程学都产生了巨大的影响。

最后，对于呼叫数据的分析只是众多问题中的一个方面：法律强制实施数据匹配，是指对于大量数据源中数据的处理。最早对于多源数据的正式使用出现在德国，大约在 20 世纪 70 年代，被 Baader Meinhof 恐怖组织用于查找安全藏身处。调查人员寻找那些具有不规则使用高峰的被租用公寓，这些公寓的租金和电费账单都是通过来自不同地区的远程信用卡转账支付的。这确实可以工作：它产生了一个关于数百家公寓的列表，其中有一些就是恐怖分子的藏身之所。用来做这种分析工作的工具目前被许多用于流量分析工作和主要用于警方调查管理工作的产品所集成。它们被使用的程度依赖于当地的规章制度。在英国，曾经出现了一些争论，焦点就是警方访问了由国家卫生机构的药剂师填写的处方数据库系统，而在美国，医生经常由于个人健康信息被调查者通过健康保险公司来索取而收到恐吓。而且，对一些商用和政府的私有数据格式的不清楚也是一种实际中的限制因素。但是，对于警方来说，访问那些诸如电费账单之类的信息是很平常的事情（这些信息可以帮助他们找到种植大麻的人）；而且，从长远观点来看，任何现在被监控和记录的事情将来都有可能被他人索取。在英国和美国中，2001 年初期提出或者已经实行的一些措施将付予警方更多的权力来通过电子方式获取个人数据。

21.2.3 对外国目标的通信情报

我在第 16 章中讨论了关于信号侦察技术方面的问题。现在可以简要地看一看政治和组织方面了。

大多数通信情报，无论是否包括搭线窃听、流量分析或者其他技术，都不是由于法律实施的原因引入的，而是由于国外情报收集的原因。在美国，主要负责此事的机构是国家安全局（National Security Agency，NSA），它的预算（虽然机密）一定是有数十亿美元的，因为该机构中有巨大的设备和成千上万的雇员。NSA 完全使法律实施的 150 ~ 200 条活动的监听线路相形见绌。这种情况和其他国家中的很类似。英国国家通信总局（Government Communications Headquarters，GCHQ）也有数千名雇员，以及公认的 6.5 亿英镑的预算（大约合 10 亿美元）。但多年来，在伦敦警察厅只有一名警官处理伦敦所有的警方监听管理工作（而且他还负责一个电脑犯罪小组）。

不断有信息显示出当代信号情报操作的规模和效率。Kahn 从事密码技术的经历通过描述二战开始前的情况而为当今的情报技术打下了坚实的基础［428］。一位匿名的前美国国家安全局（NSA）分析家，后来被确定是 Perry Fellwock，在 1972 年揭露了 NSA 工作的规模

[288]。“NSA 收集的信息是完全的”，他写道。“包括外国政府正在做什么，计划做什么，以前做过些什么：哪支军队向哪里移动去和谁作战，哪支空军移动到哪里，以及他们的作战能力如何。对于 NSA 来说确实不存在任何限制因素。NSA 的工作包括从越南 B-52 的呼叫到监控苏联空间程序的方方面面。”

Fellwock 的动机是敌对越南，下一位泄密者是英国的战时电码译员 Frederick Winterbotham，他希望写一本关于自己战时成就的论文集，后来由于他去世才免遭起诉。在 1974 年，他揭露了在战争期间盟军成功破译了德国和日本的密码系统 [806]，这使得出现了许多关于二战时期破译情报的书籍 [188，429，800]。从那以后，很少出现那些钟爱研究的新闻工作者揭露的情况了，在这些少量的资料中，有相当一部分是关于官员的腐败和滥用设备来监控不应该监控的目标，例如国内的政治团体。例如，据泄密者 Peg Newsham 揭露，NSA 非法监控参议员 Strom Thurmond 的电话呼叫 [157，158]。James Bamford 在一些公开的资料和与前任雇员的谈话中总结了相当多的关于 NSA 的信息 [70]。但是，一个最为真实的关于美国及其盟国在信号侦察方面采取的组织和方式的资料是由新西兰记者 Nicky Hager 总结的 [368]，这是在新西兰情报部门没能遵从总理的命令而减少和 NSA 的合作之后发生的。

冷战的结束意味着这些机构不得不寻求新的理由来调整它们的预算，一种通常的方式就是通过发展经济情报来抵制与其竞争的国家。这加快了关于信息来源和方法的信息流动。在向欧洲议会提交的报告中包括美国经济间谍的最明显的行径 [278]，议会还关注苏联解体、获得经济焦点的情报，而欧盟成员国是目前主要的关注对象 [160]。

从这些源头中出现了全球范围的信号情报收集系统，就是 Echelon，由 WASP 国家共同运营。WASP 国家包括美国、英国、加拿大、澳大利亚和新西兰。数据、传真和电话呼叫在许多节点上收集，包括位于许多成员国家（或者秘密在水下铺设）的国际通信电缆，商业通信卫星发出和接收的流量发现，专门用于收集敌国潜在信息的通信情报卫星，以及位于各成员国大使馆的监听岗位等 [278]。被收集到的信息通过计算机进行实时的搜索，这些计算机可以叫做字典，因为它依照呼叫者或者接收者的电话号码或者 IP 地址来工作，还可以通过关键字搜寻电子邮件的内容。这些搜索标准被引入到各个成员国的侦察分析系统中，字典收集它们感兴趣的信息然后交给分析系统分析。Echelon 系统的工作似乎有些类似于网络搜索引擎，用于取代查找网页的是实时搜寻世界范围的电话和数据网流量。

下面是一些值得记住的要点。

- 第一，现代军事领域如果没有信号侦察技术将很难运行，而且，在许多情况下，这种因素是致命的。当遇到干扰或者诡计时，那些对于对方雷达和通信系统有更好了解的战士将具有决定性的优势。不具备电子战能力的国家不可能在空战、海战或者陆地上的坦克战中具有竞争力。如果当局军队不准许打游击者使用现代通信工具的话，那么游击战争也不可能起到什么作用。所以，NSA 中大多数人员属于军队，且 NSA 指挥官就是现役将军就不足为怪了。它的大部分工作是关注对于雷达、遥感设备、武器导向、电子对策以及其他敌对或者潜在敌对的国家中此类资源的识别和分析。
- 第二，无线电话、无线局域网和其他基于无线的技术应用的增长，再加上现在什么事情都是在线进行这样一个事实，使得出现了一批机构，伴随而来的是更加丰富的新型信息源 [560]。不管对密码技术策略争论产生的成果如何，时代从没有这样

好过。

- 第三，即使是拥有每年数十亿美元的预算以及成千上万雇员，NSA 也不可能收集到世界上每个地方的电子通信信息。Fellwock 所描述的世界已经不存在了。Sprint 公司的预算比 NSA 的还要多，这有赖于低成本的商业产品而不是高档次的机密产品，所以它在线路监听方面要比 NSA 快许多。即使 NSA 仅仅对英国大学系统感兴趣，并且能够成功地监听到每一所英国大学的网络访问点，它也不能将所有的比特都通过大西洋运到 Fort Meade。这是因为没有足够的传输带宽。监听所有来自日本公司的数据流的任务将比这还要难上一个数量级。因此，情报机构所面临的中心问题和警方遇到的一样：流量选择。虽然原先对于所有穿越大西洋的数据和数据流量进行记录是可能做到的，但是即便是在今天，这件事花费也是巨大的，因为通信带宽大规模增长，同时费用也比数据存储能力的花费降低得更快。关键问题就是流量选择操作是否可以实时进行。
- 第四，虽然其他国家抱怨美国的信号情报收集行为，但它们用道德谈论此事是伪善的。其他国家同样也运行着情报系统，而且在获取经济和非军事间谍信息方面通常更加具有侵略性。在 WASP 国家和其他国家间的真正区别在于，没有人创建这种“系统的系统”。确实，对于通信情报的经济价值方面的工作是受网络影响的，这同其他许多在线行为一样。网络价值发展的速度大于它实际的大小，侦察网络好像和电话网、银行网络或者因特网本身没有多大区别。你对它监听越多，得到这些信息的代价就越低。因此，已经出现了创建“欧洲 Echelon”系统的趋势，该系统包括欧洲大陆国家的警察和情报机构 [269, 280]。

信号侦察对于国家的生存是必需的，但是那些诸如武装暴动等的潜在性危险也不容忽视。一个军队可以是很好的仆人，但也可能成为无法忍耐下去的主人。问题不在于这些资源是否应该存在，而是它们如何被安排到其应处的位置上。在美国，1975 年时 Senator 教堂的听众详尽地阐述了许多滥用权力的例子，例如，非法监控美国公民 [185] 等。国外情报搜集受美国法律 50 USC 1801-1811 的控制 [759]。这并不是完美的。它的需求相比于国内监听来说过于松散，在许多情况下，总统通过简单地授权信息收集而不是获取许可证的方法就可以做到。还有，这也存在着一些已知的漏洞。一个就是通过海外友好服务而通敌协作。当玛格丽特·撒切尔想要暗中监视一名内阁大臣时，她聘请加拿大人来完成这项工作 [322]。而且，如果美国总统真正想要监听一名参议员，毫无疑问，他只需简单地叫英国国家通信总局 (GCHQ) 来做这项工作就可以了，对于 GCHQ，这是一项完全合法的国外情报工作。美国人很幸运：在大多数国家中，情报疏忽的问题甚至都没有被讨论过。

然而，拙劣的控制以及对追查对于自动数据处理系统的侵害或试图侵害的责任者的性质或状态 (accountability) 的差劲表现所造成的后果比偶尔滥用政治权力更加严重。一旦战争打响，对于官僚作风调查的增多被证明是没有什么用处的。在冷战时期，华盛顿也变得很普通了，机构间的敌意甚至比对苏联的敌意还要强。在英国，一个最为不道德的情报战不是针对爱尔兰共和军 (IRA)，而是在警方和 MI5 (军事情报部第五局) 之间展开，后者正是打击 IRA 的领导者。还有很多的理由可以说明，情报的无效及其内部人之间的暗斗，例如 Jones [425]。我们在官僚政治的战场中，将讨论有关密钥契约的全部问题。

21.2.4 密码策略的历史

许多国家在19世纪中期制定法律来禁止在电报消息中使用密码技术，一些法律甚至禁止使用经批准的列表以外的语言。普鲁士要求电报操作者保留所有消息原文的拷贝[729]。有时，理由就是法律的实施，从而防止人们在官方传送信息前就提前得到关于赛马比赛结果或者股票市场的价格信息。但是，其真正的原因是用来关注国家安全。这种模式在20世纪时又重复出现。

在二战中的盟军应用密码技术和信号侦察技术取得巨大成功后，英国和美国政府达成协议继续进行情报合作。虽然很快就有其他WASP国家加入到该项合作当中，但该协议还是取名为UKUSA协定。虽然早在1947年时这项协议就已经创立，但是直到1999年才被正式认可。这期间，实施密码策略的成员国的主要目的是防止加密设备的扩张，而且可以明白其工作机制。它只是向我们这些工作在诸如银行业的人勾画出一个大致的轮廓。最近，有大量前权威人士所写的文章对此进行了更加详细的描述。

21.2.4.1 出口控制

直到20世纪80年代，那些制造密码设备的公司，几乎都是将产品销售给政府部门。基本上，它们不会将产品销售到国外，这一点还是可以信任的。因为这样做将使它们在国内的主要客户忐忑不安。这一点通过出口控制的方法得到加强。所谓出口控制就是指，通过一种尽可能隐蔽的方法，对于任何人关于诸如出口许可等的需要，尽量以开放程度最小化的原则为指导。大多数的事情都是在幕后由官方和一个被信任的出口商代表之间进行协商而确定的[82]。

在这些协商中，权威机构试图让申请者使用的密码技术的健壮性尽可能得低；而且在面对一个更加复杂的用户时，还试图在这些系统中加入一个“后门”程序，在商业中这被称为红线病，通过这个后门可以访问想要得到的信息。任何想要在国内销售高质量的密码软件的人都会遭到来自各方的劝阻。大公司将受到丢失政府合同的威胁，而小公司也没有出路，因为它还要试图得到电信和其他产品的批准许可。在密码问题之外的那些必须由计算机予以控制的问题，将随着技术的发展而发展。到90年代中期时，那些在小孩子卧室中都可以找到的计算机被认为是军需品，制造商们为了数量巨大的出口订单而疲于奔命。这恰恰取悦了官僚者，因为他们得到了工作和权力。当然，权力常常是被滥用的。一次，一个要销售到南斯拉夫学校中的大量英国造家用电脑的出口订单在美国某权威机构的坚持下被取消，其理由是电脑中包含了美国的微处理器。而一家美国公司立刻就得到许可证来出口这些产品。虽然像这样的事情使得出口控制系统声名狼藉，但它还是坚持到了今天。

70年代早期，ATM技术以及其他电子银行应用的发展都需要应用密码保护技术，而且还需要对技术进行标准化和合理的质量保证。部分的解决方案是使用和与导弹技术出口相同路线的密码策略，即仅仅提供最低限量的出口产品，从而防止其他国家的公司有发展该类产品市场的可能性。当密码控制变得十分费力，以至于某些国家的银行，例如巴西和南非，开始通过本地电子公司生产密码设备时，出口许可变得趋于缓和，这种情况将一直持续到该威胁度过之后。

另一部分的解决办法在于对银行密码的标准化控制工作。在20世纪70年代，一个令NSA担忧的问题就是许多国家仍在使用密码机，该机器使用二战时期发展起来的技术就可以

将其破解。使用密码机的国家并不仅仅是那些贫穷的国家，南非一直使用转子机持续到90年代中期，而瑞士则一直使用到90年代初。如何为银行业提供一个完善的加密机制呢，这并不仅仅局限在美国本土，海外银行业也是一样。但同时该技术又不能被外国政府采用，因此，这为侦察收集工作加入了大量的开销和花费。

21.2.4.2 数据加密标准和密码研究

解决方案还有数据加密标准（Data Encryption Standard，DES）。我在5.4.3.2小节讨论时，提到对于56位是否够用的问题出现了大量的争论。我们现在知道了，这种争论是经过深思熟虑的。NSA在当时并不需要用来做DES密码查找工作的机器，这是后来才需要的。但是，出于对DES技术的深刻印象，NSA还是设法阻止其他国家采用DES。转子机继续提供服务，但在许多情况下已经使用微控制器来重新实现，而且通信流量继续被捕获。情报侦察的目标是那些将自己的重要数据经过密码加密过的人，而这仅仅解决了NSA的通信流量选择问题。

第二个动机是破坏关于密码学的学术研究。在20世纪70年代，这是通过直接攻击那些受牵连的人实现的；到了80年代，已经发展为更加精细的策略，即声称那些已经发表的学术研究成果都是陈腐过时的。那些反对资助密码研究的机构会说："我们在30年前就已经做过这些工作了，为什么纳税人还要为此再付一次钱呢?"这种暗示正符合DES的情况（我们仍将看到的一个负面影响就是，密码和计算机安全团体在80年代初期开始相互分离，因为NSA试图压制一个而发展另一个。对于所有的角色来说，今天都将为此付出巨大的代价，这其中也包括NSA。另一个代价是，无论何时NSA出现了错误，正如Clipper的构思一样，都将得到严厉的评判。正所谓"走到哪里，就在哪里出现"）。

到90年代中期时，这条路线似乎走到尽头了。各种密码契约（key escrow）的设计错误表明这些代理机构在密码学方面并没有专门的技术来和开放研究团体相比较。而且，随着试图通过干涉资助资金的方式来影响学术研究的发展方向变得越来越没有效果，它们也变得更加平庸了。

21.2.4.3 Clipper芯片

密码策略是1993年伴随着Clipper芯片的出现而出现的。Clipper出现的直接刺激因素来自AT&T公司向美国国内高级加密电话市场提出的建议性意见，这种电话使用Diffie-Hellman密钥交换和三重DES来保护通信流量。政府对此的响应是，它将使用其巨大的购买能力来促使不同的标准规格取得成功，在这些规格中，具有可以让一些机构在解密时使用的备用密钥。这种做法导致了公众的一致抗议，所以Clipper就没有应用开来。

由于政府可以通过各种伪装的办法来访问密钥，所以也曾经试图增强密码技术的使用情况。密钥契约得到许多新的称呼，例如密钥恢复。认证权威机构保存其客户私有解密密钥的拷贝，这些机构被称作可信任第三方（Trusted Third Party，TTP）。这有点强调NSA关于可信任组件的定义，这种组件是可以破坏安全性的。大部分策略平衡机制都与出口许可有关，因为典型的美国软件公司的产品大部分用于出口，而且由于维持一条专门用于出口产品的产品线花费太大，所以许多公司都利用限制出口的办法来劝阻它们不要提供健壮的密码机制。具有"已批准"密钥契约功能的产品将享有美国出口许可优先权待遇（关于这次斗争的历史还将被全面撰写，第一个可用的草案来自Diffie和Landau [250]；而且由FOIA获取的许多美国原始文献已经出版在[684]中）。

从这个过程里可以获得的一个工程学上的教训就是完全做到密钥契约是相当困难的一件

事情。从制定双方的安全协议转到三方的协议将会增加复杂性和出现严重设计性错误的风险性，而且集中式的契约数据库将成为一个很大的攻击目标 [3]。在需要契约的地方，通常一种好的做法是使用本地机制。在一个军队中，最佳的解决办法是每个军官必须将其通行口令写在一张纸上，并放到一个信封中，印上“机密”字样的印章交给指挥官，该指挥官将其放到办公室中安全的地方。通过这种方法，密钥同用其保护的电子版文件在同一个地方保存，并不存在一个可以被飞机轰炸或者间谍偷取的中心数据库（如果你已经跟随过关于密钥契约的讨论的话，那么你可能会提出反对意见，“但是一名士兵可以存入一个错误密钥然后逃跑，试图通过正确密钥进行敲诈和出售。”我将这个问题摆在一名调查者面前，他看我的样子好像我疯了。我现在相信这种反对意见确实疯了，或者最多也就是小题大做而已。任何人，不管是士兵还是程序员，都可以获取密钥文件来进行敲诈勒索。但在实际应用中，这是一件很少出现的事情，几乎没有人会对此担忧）。

21.2.4.4　欧洲的立法

在欧洲，事情变得有些混乱。下面仅仅是一个简要的描述（在 [472] 中有更进一步的调查）。国际武器控制协议（东西方贸易统筹委员会 COCOM 和 Wassenaar）约束大部分国家政府，要求它们在密码设备上实现出口控制。而且属于欧盟（European Union）的成员国还要受到欧盟关于双重用途货物（dual-use goods）的出口规定的约束。这里的双重用途货物是指那些既可以民用又可以军用的货物。但是，整个欧洲对于加密控制技术反应冷淡，而且各国在实现上也不尽相同。英国法律对于无形产品的出口根本不予以控制，所以，加密软件可以通过电子方式来出口；比利时政府几乎对所有东西都要授予许可证；瑞士则仍旧是密码设备的主要出口国。国内的控制也不相同。法国政府的起点放在禁止大多数的国内密码技术，后来又转移到几乎是完全自由化的状态。而此时，英国却走了另一条不同的路线。

1996 年，在即将卸任的英国梅杰政府通过的最后一系列法案中，有一项就是提议强制实施密钥契约机制。持反对意见的英国工党立即对此进行谴责：“试图控制加密技术的使用从原则上说就是错误的，在实际中难以实行开来，对于信息网络的长期性经济价值来说是起破坏作用的” [197]。可是，一旦当权后，他们的观点立刻改变。新通过的调查权力规定（RIP）法案允许警察要求你提供任何你拥有的密钥。如果你拒绝提供密钥，那么将判刑 2 年；如果你将此事告诉别人，那么将被判刑 5 年。这给人留下一种故意的印象，即“胁迫契约（escrow by intimidation）”，就是通过威逼一些公司使用密钥契约，从而保证他们可以遵从法律实施的要求获得密钥的准则。然而，这种在无法获得密钥的情况下，试图将公司主管送入监狱的做法在业界的游说面前常以失败而告终。关于 RIP 账单的历史，参见 [304]。

另一项贯穿欧洲密码策略开始时期的事情是试图将密钥契约和其他立法提案及标准联系在一起。例如，欧洲电子签名方针（European Electronic Signature Directive）强迫成员国使用那些已经认可的产品来提供高质量的电子签名识别，至少在一个国家中，这意味着用产品来支持契约。而且，正如我们在电信欺骗一章中所提到的，法律实施途径被内置进第三代移动服务的标准当中。

21.2.4.5　红线病和密码 AG 案例

密钥契约经常在不具备用户知识的情况下被应用开来。当瑞典政府获知 Lotus Notes 软件的“出口版本”（该软件被广泛用于公众服务中）故意在密码技术上做得十分脆弱，以便让 NSA 可以访问时，变得非常忐忑不安。而且，至少有一种（美国出口批准的）密码机在 VHF

波段不受阻碍地广播明文。但是，更声名狼藉的是 Bühler 的例子。

Hans Bühler 是一名为瑞士公司 Crypto AG 工作的销售人员，该公司是密码设备领域主要的提供商，专门为那些不具备技术能力构建系统的政府部门提供产品。他 1992 年时在伊朗被捕，权威机构控告他出售的密码机遭到破坏，以至于敌对势力可以获得其明文。Bühler 在狱中度过一段时光后，Crypto AG 花费了 14.4 亿里亚尔（大约合 100 万美元）将他保释出来，然后在他回到瑞士后就将其解雇了。Bühler 后来在瑞士电台和电视上宣称公司被德国的情报部门秘密地控制着，而且公司多年来都涉及情报工作 [143]。对于此事的隐瞒欺骗通常采取的解释是，最终的控制在 NSA（Crypto 的创立者 Boris Hagelin 是 NSA 主要科学家 William Friedman 毕生的朋友），设备例行地犯红线病（red-threaded）[517]。另一种不同的解释是这些辩解是 NSA 编造出来破坏该公司的，因为它是第三世界国家中少数的几个密码设备供应商之一。Bühler 的故事参见 [740]。

一个普通的安全工程师（不涉及情报工作的工程师）应该了解些什么呢？

21.2.5 讨论

当密钥契约的争论在 1994 ~ 1995 年间出现在英国时，由于契约正反两方的游说，使得我选择了一条在当时不太流行的路线。支持契约的人认为，由于密码提供机密性，而机密性又能够帮助罪犯犯罪，所以必须采取某些方法来处理这种情况。反对契约的人认为，由于密码对于隐私是十分必要的，所以不可以采取某些方法来抵制这种机制。我在 [21] 中的论证是，从本质上说，这些争论背后的前提假设都是错误的。大部分的密码应用（在现实世界中，相对于学术界而言）是关于认证的，而不是机密性问题；这些会帮助警察工作而不是阻碍他们工作。就罪犯而言，他们需要不显眼的通信方式，而对电话呼叫的加密是一种引起有关部门对你注意的好方法。就隐私来说，大多数的冲突来自内部人员对授权访问的滥用。最后，对于警方或者调查电子犯罪的审查人员来说，一个更加严重的问题是找到可以被接受的证据，对此，完善的认证就可以帮上忙了。

后来发生的事情在很大程度上证实了最初反对的观点。在 20 世纪 90 年代的大部分时间里，我帮助组织一个以英国剑桥的白领犯罪为主题的一年一次的研讨会，还组织关于密钥契约和相关主题的一些常规性研习会。从这些会议的举办情况来看，对于构成我主要听众的警察和检举人来说，并不能引起他们的兴趣。只要是关于搭线窃听和加密的讨论开始了，他们就离开，大多数警察机关只有被要求时才对该话题产生兴趣。在许多国家中，包括美国和英国，在密码策略方面的领导机构是一个法律实施部门（在美国是 FBI，在英国是国家犯罪情报部 National Criminal Intelligence Service），但是，对于情报界来说这只是一个表面问题——其实 1996 年时，在毫无准备的情况下，该部门就被英国在欧洲专门负责密码策略的机构的代表所承认并容许了。

21.2.5.1 法律实施还是情报

问题连续不断出现的一个根源就是使用法律实施的方式作为收集情报的借口。警察和间谍的打算和目的是相当不同的，这往往连他们自己也容易混淆。前者就是试图防止国内犯罪，而后者试图在国外犯罪。这种理解也许过度单纯化了，但是，这句格言恰恰说出了潜在的紧张不安的状态。例如，警察希望保护证据，而间谍希望可以随意地伪造和复制文件。在讨论欧洲关于密钥契约（“Euroclipper”）的策略过程中，引出了电子签名令，德国政府要求

只有机密性密钥应该被契约化，而不是签名密钥。然而，英国希望签名密钥同样也被契约化。英国的观点遵从了军事学上的教条，即欺骗至少和偷听一样重要，而德国遵从的是警方的教条，即避免使用那些会破坏随后获取的证据的调查技术。

密钥契约如同分级官方文件系统一样，同样可以对官方错误做法提供似乎合理的否认能力。在英国行政事务中使用的密钥管理系统分发签名密钥给最终的使用者，这个过程是通过契约化的机密性密钥加密而进行的［50］。所以，如果一个令人尴尬的电子文档被泄漏到出版单位的话，政府可以声明这份文件是部门安全官员伪造的。这位官员是专门负责防止机密泄漏的，同时，他还是具有获取契约化密钥权限的官员。依靠这一观点，这种系统或者是一个完美的安全工程，它的设计者应该得到嘉奖；又或者这个系统是邪恶和不正当的，它的设计者应该被送进监狱，因为他没有起到应该起到的作用，并且破坏了信息自由的基本原则。

远离签字密钥的话题，情报界看样子将成为密码控制的主要受益者。这不光是由于那些搭线窃听是最为经济的监听方法，就像 Saddam Hussein 一样。如果绝大部分的数据流量都经过加密，那么由诸如 Echelon 这样的系统完成的自动关键字查找功能将受到严重的阻碍。一些人已经注意到每年都有大量的网络基础设施被创建，而且，如果不是一开始就创建密码机制的话，那么以后再创建就会花费很高的代价。因此，在 NSA 坚持这种密码控制路线的每一年中都会出现成百上千个在未来几十年中将处于被监视状态的网络。无论这种做法是否可以给美国和欧洲带来长期的好处，但留给我们最为可怕的一点就是在美国和欧洲间的经济间谍所导致的不稳定冲突因素，我们都将在 20 年中完全处于暴露状态。这个问题并没有得到更多的争论。

这并不是说搭线窃听对于警方没有多大用处。虽然警察机构在没有搭线窃听技术的支持下也可以获得不少信息，而且那些支持搭线窃听的游说者所提供的许多数据都是不真实的［250］，但在一些情况下，搭线窃听仍然是一种经济划算的调查方法。由一名澳大利亚高级情报官员 Walsh 所做的报告提供了关于这些问题的不寻常的平衡调查信息［787］。Walsh 使用搭线窃听的价值、软件漏洞和物理监听三者进行比较，并指出搭线窃听既是最便宜也是在某些情况中惟一的调查技术。但是，他仍然感觉到，政府强行获得密钥信息的做法很可能是无效的。“对于非自身控告原则的祈祷很好地表现出一些可能反应的客气做法的结束”，他冷冷地评论。在他的发现中可以看到，“过早地提出关于密码技术的规则和立法并没有必须的理由和好处。”但是，他确实建议警方和情报机构可以被允许来攻击目标计算机，从而获得对证据信息的访问。[⊖]虽然由于技术上的进步，可能会需要一些对警方的投资，但是也同样会出现一些机会：例如，通过软件来感染嫌疑犯的计算机，从而将该计算机变为一种收听设备。一般而言，警方就如同那些情报部门一样，都是从现代技术中收获成果。

总体而言，网络对于密码契约机制的法律实施的争论所产生的影响是负面的。它侵害了公众的信任和操作的有效性。在情报界也是一样，许多官员对于使用 Clipper 表示非常遗憾。

⊖ Walsh 所做的报告在出版发行时也有一段有趣的历史。最初是在 1997 年作为不保密的文件发布，但在三个星期之后就撤销了这一做法，由于有些人询问为什么在书店无法购买到这本书。然后经重新编辑后出版了此书。但是到了 1998 年，调查人员发现在许多公众和大学的图书馆中出现未经删节的全书非法拷贝，而这些图书馆已经获得了该书的合法拷贝版本，出版公司立刻将此事报告政府，并引起了政府的关注。在 1999 年后期，澳大利亚政府仍旧试图禁止该报告的出版发行［787］。

在这之前，对于密码技术我们并不很清楚：只有极少数的数学家从学术角度进行过研究，虽然它被应用于提款机和付费电视解码器，但是对于通信安全方面的注意还很不够，没有引起公众的注意（当我在1985年第一次编写电子邮件加密软件时，几乎对此不感任何兴趣）。现在这种想法已经改变了。不光许多罪犯使用匿名通信信道，诸如预付费移动设备，而且许多先前购买脆弱的或者具有红线病的密码机来为军队和外交服务的国家现在也开始发展自身的专门技术和产品。然而，正如我们常说的，“策略一旦实行，是不可能回头的”。

21.2.5.2 Carnivore

在2000年夏天，关于搭线窃听、流量分析和密码控制的策略正在形成两个主要的特征。第一个特征是情报和法律实施二者之间的界限变得越来越模糊不清。它们的功能有些重叠，这尤其体现在反间谍活动和恐怖主义的情况中。在诸如美国的一些国家中，一些机构很明确地被赋予两个功能（例如FBI，虽然注意到，在1998年时，12730项罪行中只有45项最终由司法部门（Justice Department）归类为内部安全或者是恐怖事件[751]）。在其他国家中，出现大量的地区性冲突。我曾提到，在英国关于是否由警方或者MI5来处理IRA的问题。自从北爱尔兰和平条约签署以来，这类冲突一直在计算机犯罪领域重复着。冷战的结束，以及许多地域性叛乱的停止，使得许多相关的情报机构拼命地去寻求新的业务路线。

第二个特征是在ISP上增加更多的监督机制。监听数据流量比起过去的监听话音来讲也困难许多。现代的调制解调器使用适应性回声消除技术，该技术使得对本地回路的被动式拦截侦听更加困难。而侦听回路之外的其他部分也将面临一些障碍，例如提供给拨号用户的临时IP地址和包流量机制中逐渐增大的分布特性。俄罗斯和英国都已经引入了相关法律来要求ISP在它们的网络中增加黑盒子，从而达到监督的目的。而在美国，FBI使用一种叫做“食肉动物（Carnivore）”的设备，该设备也具有类似的功能。之所以把这种设备叫做“食肉动物”是因为它可以从数字化搭线窃听中“获得肉食”，Carnivore在[717]中提供了进一步的信息。

Carnivore背后的思想就是：法律的解决方案变得越来越无效，因为技术的变化实在太快了。而且出于诊断的目的，ISP们使用基本工具来监控它们的网络，但这样做也只能获得部分信息而已，或者获取的信息过多了（与法律上规定的获取信息的最小量相冲突）。所以，更佳的办法就是使用技术上的解决方案，这种方案是基于一个多用途平台，该平台的软件可以根据需要进行升级。实际上，Carnivore可以被远程配置，对于这一点，一些ISP并不十分喜欢。除此之外，操作员是完全被相信的，仅仅按下一个按钮就可以收集所有的TCP通信流量，而且这种设备对于那些建立私人业务的责任缺乏审计追踪机制。还存在许多严重的问题，例如如何处理非标准的ISP设备，如何处理位于某些服务上层的服务，例如Web邮件。毋庸置疑，Carnivore和国外同类产品将继续发展，而ISP业务变得越来越复杂，这将使得它们的维护人员更加忙碌了。

至少在美国，对于搭线窃听的更好的法律监督机制意味着，Carnivore没有被取代，但是也只能在法庭授权一个许可证之后才可以被安装，而且，每月部署的次数仅仅通过一只手的手指个数就可以计算出来。在英国和荷兰，看上去好像类似产品将在所有主要的ISP上被安装，用于持续监控网络流量[147]。在俄罗斯，则已经是这么做了。

21.2.5.3 潜在的策略问题

我们花费很大努力来创建一个很少被使用的性能？恐怕许多关于电子策略的争论都包括

Freudians 所称的*替换活动*：当遇到无法解决一个困难问题的挫折时，去解决一个简单的且与原先问题相关的问题就可以了。例如，在英国，至少从伊丽莎白一世女王时代开始，国家的名声就很糟糕，那些确实有罪的罪犯几乎无法被起诉。仅仅当这些人的业务崩溃垮台时，他们才会被捕，就像在 9.2.3 节中所提到的 Barings 和 Maxwell 的例子一样。而且，Leeson 是在新加坡而不是在伦敦被起诉，而 Maxwell 的犯罪事实是在 Leeson 自杀之后才被发现的。在我自己的专业实践中，要花费很长一段时间才可以向警方告发不诚实的银行家：任何人也不会有关于高级银行家被起诉的印象。在美国，每年大约有 1 000 名和副总统同等级别的银行家被起诉，而且超过三分之一的人被判入狱。这种做法大概不符合英国的美德，或者说这是一种美国式的罪恶，但是确实说明了两国的法律实施系统是如何组织的。美国警方官员通常在他们赢得了影响力很大的案件时才可能被升职。所以，美国相关机构，例如 FBI、安全部和地方法院等，都争相将银行家送入监狱，从而得到晋升。相比之下，他们英国的同僚只能依靠上头的恩赐才能晋升，所以对于卓越人士的调查如果没有达到死罪的程度则只能意味着自己职业生涯的结束。因此，英国机构振作精神争相来寻求其他的方式。目前，在众多犯罪中，只有一少部分是通过高明的欺骗手段来获取高额利益的犯罪，而搭线窃听常常是犯罪调查中一种比较经济划算的做法。英国政府中关于警方监视权的公开争论变得越来越少了。

替换活动并不局限在通信情报问题上面。许多基于因特网的儿童性攻击，尤其是对小孩子，这些问题都可以用到替换活动。然而，这种案例的数量很少，而且即使是在一个影响很大的案例中，该案例涉及的罪犯是流行乐曲的偶像人物 Gary Glitter，法官将这种类型的案件称作“可能出现的最为糟糕的类型”，法庭认为 4 个月的判决也就比较适当了。大多数罪犯侥幸成功地逃脱罚款或者是被感化劳教 [12]。所以，这并不是最严重的犯罪类型，而且随着儿童使用计算机，色情文学网络可以追溯到 20 世纪 80 年代，所以这也不是什么新型犯罪了。还有，当你和别人谈论关于儿童保护的问题时，就会清楚在英国每年都有成千上万严重虐待儿童的案例，这种虐待包括，来自家庭成员的虐待、对于具有学习困难的年轻人的虐待、对于仅由地方当局监护的儿童的虐待、未达到法定年龄的卖淫行为等。由于各种不同的政治因素，警方并不总是利用最方便的手段来调查这类案件。至于慈善机构，孤儿院制度的结束使得他们只能通过得到当地政府的许可才可以将容易受到攻击的小孩子收养照顾。而且，儿童组织通过花费慈善经费来抗议网络所带来的罪恶 [168]，这种做法比起游说有名望的中层社会中那些成为 13 岁妓女的顾客，而希望可以使儿童强奸事件数量有所下降的做法要好很多 [528]（对于性犯罪的态度上，以及转移机制上，有一些值得反思的东西 [215]。看来，仅仅由于对上帝的普遍信仰留下一个监督的真空，所有的政府都涌入这里来填充它，所以，恶魔的死亡留下一片空白。关于儿童性虐待的不正常的兴奋被鞭打、被唾弃，这种行为和那些继父继母虐待女孩子的做法是离得很近的。人们转变成“恶魔”，这是他们自己最黑暗的内心恐惧和儿童时代的创伤所造成的）。

所有这些机制的实现，对于安全工程师来说，必须要考虑你的产品或者服务被那些工作效率低下且堕落的人民公仆，或者是无知且伪善的自我评论员利用，从而成为胡乱使用的目标。你不可以忽略你试图创建的系统所处的社会和政治因素。

21.3　版权

在第 20 章中，我讨论过 90 年代关于密码策略的争论很容易引起更大的战争，这场战争

包括匿名服务、审查制度和版权。我在该章中涉及了一些技术方面的问题，也讨论了一些相关的商业和政治方面的观点。然而，我们所处的环境中不仅仅涉及版权问题。一些机制，例如匿名 remailer 和高分布式的文件存储机制，允许人们行使他们的权力来匿名地进行政治演讲，也可以出版那些破坏名誉或者具有煽动性的言论的资料，出版这些资料很有可能被捕或者是判刑。因此，版权实施的游说会得到一些强有力的潜在性支持。

还存在一些地理政治学的因素。在大多数国家中，并没有演说自由权（更不用说匿名政治性演说了），而在美国却是有演说自由权的。甚至在一些欧洲国家中，针对破坏名誉或者具有煽动性的言论的法律可能十分粗鲁野蛮。这里同样存在影响力很大的事例，在许多具有抵制复仇性言论的法律的国家中，例如法国和德国，已经寻找到多种方式来审查美国的在线服务。法院仅仅给 Yahoo 三个月的时间来避免法国用户可以访问到纳粹纪念物的拍卖信息，这在法国是不合法的。2000 年 11 月在柏林全球因特网项目会议上，德国联邦司法部长宣布，她在职期间最大的成就就是停止了在线书籍将“Mein Kampf”的拷贝发往德国。通过翻译人员的解释，她向我们保证她将不会退休，直到盗版者也停止将这些非法拷贝运到美国亚利桑那州为止（考虑到“Mein Kampf”的版权属于德国南部巴伐利亚州政府，他们必须通过繁琐老式的法律方法来获得最终出版权，也许公开指责因特网的罪恶被认为对那些投票者来说更具有吸引力）。

正如 Leslie Lamport 所说，如果你知道，你拥有的分布式系统在一台计算机崩溃时，将不会影响到你完成任何工作，那么当某个国家中一名你从未听说过的法官试图关闭你的业务时，或者至少以口头方式发布负有法律责任的命令，来建议你在自己的国家中应该如何去执行时，你就应该知道自己正生活在全球村中（美国独立于国际法庭判决实施，它为你提供了一些保护机制，参见 19.9 节）。我认为没有人曾经考虑国际法的分布式系统方面的问题，但是这将成为一篇有趣的博士论文的主题。

当然，这开辟了两条路：第三世界国家的领导人和亚洲强人公开指责因特网上的演说自由就是“新帝国主义。”而且大部分欧洲国家在看待色情问题上具有比大多数美国人更加自由宽大的观点。我们可以等到 2020 年，看一看那时的因特网到底是使用美国的关于言论自由的规则，还是使用欧洲的关于色情的规则，又或者是其他什么规则。

由于缺乏对破坏名誉和煽动性言论问题的一致性意见，最有可能被引入的控制方法将会是版权机制。第 20 章描绘了诸如音频卡带和视频卡带这种连续性技术是如何进入市场的，导致了版权所有者的恐慌，但是一旦好莱坞开始学会操纵这些设备时，它又变成有利可图的商业路线。随后的个人电脑软件也遵从同样的模型，只是它的发展更快一些。付费电视有些不同，因为使用了防篡改的订购者令牌，再加上对于令牌伪造者的法律追击，使得盗版侵权始终维持在一个个位数百分比的数量上。好莱坞目前正在试图让 DVD 走付费电视的老路，然而，由于设计和其他方面的错误，看上去很难将二者联系起来。现在看来，DVD 将跟随个人电脑软件或者视频卡带的模式，成为以上二者未来的分发媒体。

这个问题决不是版权和隐私权之间直接的斗争。正如我们在第 20 章中所说，公平使用的教条允许人们拷贝作品的某些部分，做这种拷贝所出于的目的从学习到批评评价都可以。对于公平使用的可能废除对大学和图书馆提出了警告。Pamela Samuelson 提出了一个普遍存在的情绪：“为什么克林顿政府希望将出现的信息超级高速公路转变成为出版人占支配地位的收费公路呢”[667]？

21.3.1 数字千年版权法案

在众多的游说下，1996 年终于在世界知识产权组织，（World Intellectual Property Oragnization，WIPO）的赞助下，在瑞士日内瓦通过了一项条约，通过签字声明有义务协调对待数字版权问题。在美国的实现就是 1998 年的数字千年版权法案（Digital Millennium Copyright Act，DMCA）。该法案禁止任何电子版权管理信息（copyright management information，CMI）被改造来捆绑数字内容，例如所有权和许可权的详细信息；宣布制造、进口货、销售或者提供用于销售的任何避开版权保护技术的产品等都是违法行为。对于为图书馆从事加密研究的人可以得到特殊的豁免权，用来发现和抵制偷窃信息的设备（snitchware）的工作在 18.3.2.4 小节中被讨论。

DMCA 还为 ISP 们提供了一些有限的保护机制，客户或者是其他第三方可以不知不觉地将版权信息放到网站上。其中一个条款是“注意并且记下”：当版权拥有者通知 ISP 存在违反版权规定的行为时，那些违反版权规定的资料必须被清除掉。为了防止这种做法被滥用，还提供了一种称为“注意并且推迟”的条款：如果订阅者提出一个正当的“反对通知”，并且证明了该用户具有合法地使用该资料的权利，那么 ISP 必须立即通报给版权所有者并且在 14 个工作日内对资料予以恢复，除非把事件上升到由法庭来解决的程度。

对于图书馆的免除是一项继续争论的焦点问题，因为数字访问意味着仅仅是具有限制条件的访问，除非你拥有作品的一份拷贝则令当别论。而且，图书馆必须对所有拥有合理期望目的的公平访问进行协商，这样所要面临的最初的许可条件和许可费用将十分高昂（这些问题在［513］中被讨论）。尤其是那些由法院强加给图书馆的问题更加难于处理，这些图书馆包括国会图书馆和其他一些的地方图书馆。传统上，在许多国家中，版权的授予是具有前提条件的，这个条件就是版权拥有者必须将作品的一份或者多份拷贝保存在国家档案中。这样做可以达到多种目的，包括帮助法庭解决版权争端、可以被后代学者所获取以及通过图书馆借阅机制来提供一些不容易找到的书籍。但是，我们如何才能对于那些使用私有平台、具有版权保护机制以及偶尔需要在线访问一个许可证服务器来获取许可的数字作品予以保存呢？然而，DMCA 的第一个大型试验就把注意力放到 DVD CSS 技术的反向工程上面，该技术目前正通过法庭来继续发展。

21.3.2 即将出现的欧洲法令和 UCITA

在欧洲，已经存在一个条件性访问方针，该方针责成欧盟成员国将那些可以通过未经授权和许可就访问诸如付费电视和因特网收费站点服务的设备列为非法设备。随着 DMCA 的出现，对于这类事情的研究工作也不再需要了。一个不同之处在于条件性访问方针保护那些控制服务访问的方法，而不是控制访问某件作品的方法。

对于作品的保护应该通过版权法令（Copyright Directive）的形式出现，在当时如何编写版权法令仍旧处于争论之中。强有力的游说声来自好莱坞和图书馆两方面。这项法令好像在很大范围上同数字千年版权法（DMCA）的作用相同，虽然在细节方面各个成员国可能会有所不同（参见［468］可以得到关于被建议使用的法令和该法令与 DMCA 比较的资料）。由欧洲委员会（European Commission）所考虑的一类观点是权利持有者在版权法所声明的条款没有实现的情况下可以进行抗议，以反对这种欺骗行为。如果真是这样，那么权利持有者所真

正得到的将会更多。还有，一些公司已经将版权控制机制和其他类型的保护机制相结合，这些其他类型的机制包括计算机游戏业中的零件控制等。仅仅将所有权利授权给版权持有者和游戏控制台供应商是不合理的，游戏控制台供应商也可能设计一种保护机制，同时对这类系统的所有侵害都被认定是有罪的，至少也是无效或者是不适当的。

还有，现存的欧洲法律允许使用反向工程来为互操作功能服务，互操作功能保证程序可以和其他程序协同工作，而DMCA却增添了一个限制，即绝对不允许存在很容易就可以找到的可用的商业性替代方案。遵循对DVD CSS技术做反向工程（这对于Linux界是十分必要的）的欧洲式的观点是，DMCA规定过于吝啬、难以接近（就是说，即使好莱坞在美国胜出，对于Linux开发者而言，在欧洲也应该有一个安全的避难所）。

在美国的另一个问题是统一计算机信息事务法案（Uniform Computer Information Transactions Act，UCITA），这是由统一国家法律委员国家理事会（National Council of Commissioners on Uniform State Laws，NCCUSL）所发起的一项典型法律，该法律将会被纳入国家范围的立法机构中。UCITA将对美国统一商业代码（U.S. Uniform Commercial Code）进行升级，从而包括数字贸易，以及管理制造商和消费者之间有关“事务信息”的合同和契约。这种信息几乎包括所有东西，例如，故事、计算机程序、图像、音乐、网页，甚至于在线数据库和交互式游戏也包括在内。UCITA将在很大程度上扩展联邦版权法。许多州对此很不满意，它在许多情况中使用合同法来代替版权法、削弱公平使用、甚至将用于互操作性的反向工程视为非法，而且在使用者阅读条款之前通过收缩包装（shrink-wrap）/一触即发（click-on）的许可证来将用户限制到合同的条款之中。UCITA将对从开发源代码的软件作者到大学范围中的任何人都产生严重的后果[608]。

到如今，不可避免的是将在美国法律和欧洲法律之间产生重要的差异，这种差异将对安全设计产生重大影响。还有，在细微结构的层次上，问题的解决要看众多的州和国家法律机构之间的斗争。这可能最适合好莱坞的胃口，因为它有钱，而且有团体资源，从而可以立刻在许多地方进行游说。但是，这也增加了某些地方成为避难所的可能性，就像90年代初，爱尔兰成为付费电视智能卡的复制活动的避难所一样。

21.4 数据保护

*数据保护*是在欧洲使用的一个术语，它的意思是保护个人信息，防止其被不正当使用。个人信息通常是指可以确认的个人，或者说是*数据主体*所具有的任何数据，例如银行账户详细信息和信用卡购买模型。这个术语与美国的*计算机隐私*大致对应。术语上的差别是由法律和意见上的主要差异引起的。事实上，这可能成为21世纪前10年中关于电子策略的最为棘手的问题之一，随之而来的还有一系列关于人们创建电子商务系统的复杂性因素。

欧洲法律准许数据主体检查他们拥有的个人数据，如果不准确就加以更正，理解它们是如何被处理的，以及在许多情况下防止这些数据未经他们同意就传递到其他组织中。它的意思是说，例如，那些被拒绝贷款的人不光可以看到他们的档案，还可以看到获得此结论的贷款的计算方法。而且，如果一家美国银行不喜欢这么做，情况也是一样。对于国家安全则有豁免权，但这不是指所有的警方数据。大多数商业数据被包括在内，尤其是对于某些数据具有严格控制的场合更是如此，这些数据包括有关私人的健康、宗教信仰、民族、性生活、政治联系等。最后，最近的法律规定个人数据不可以被发送到那些不提供个人数据保护的国家

的组织中。实际上，这就是指美国，美国的关于隐私的法律保护机制是十分脆弱的。

对于设计电子商务应用的工程师来说，这意味着一旦相关的欧洲法律一致被检测并最终发展到提交至欧洲法庭的阶段的话（这大约要到 2004 年或者是 2005 年），那么你所设计的应用处理位于美国本土设备上的欧洲客户数据时，就会是一种非法行为了。一种解决办法是将你位于欧洲国家的服务器放到数据保护实施不很严格的国家，例如英国或者爱尔兰。但是，目前正在出台一些法律来限制各个欧洲政府的自由度，以防止它们不认真贯彻。如果你的业务模型包括收集大量的个人购买习惯、新闻阅读模式等等信息，那么即使在伦敦或者雷克雅未克也会遇到麻烦。

适合于许多业务的另一种解决方案是强制同意。它的意思是，你必须在对客户开展业务之前，坚持让他们同意共享其个人数据。这种方法目前工作得还可以（这正是美国医药保险者所采用的方式并获得成功），但并不保证可以永远成立。网站上这种一触即发并且毫无隐私的协议将被法庭认为是一种不公平的合同条款，在一些国家中，如果客户是少数民族或者信息属于诸如健康之类的隐私的话，这就是无效的合同条款。

欧洲隐私法不使用那些已经成型的来自于 Zeus 的东西，虽然，看看自己的前身还是有些帮助的。

21.4.1 欧洲数据保护的历史

技术恐惧（指技术会对社会及环境造成不良影响的恐惧）并不是 20 世纪后期才出现的产物。早在 1890 年时，Warren 和 Brandeis 就警告过由于“近期的发明和商业模型”所带来的对于隐私的威胁，尤其是摄影技术和爱调查的新闻界和报刊等 [792]。在随后的几年中，在 20 世纪 50 年代大规模的零售商业开始使用计算机以及随后的在 60 年代初银行业也跟着使用计算机之后，人们开始担心，如果一位公民的交易信息可以被收集、整理以及分析的话，那么将会带来什么样的社会影响。在欧洲，大型商业通常可以避免掉这些责难，因为法律条文规定只有政府机构才可以负担得起足够的计算机，从而构成对隐私的严重威胁。人们现在已经认识到对于政府来说，通过使用所有公民的个人数据来作为以后预测的基础材料，从而扩展起控制力度是可能、经济且合理的。而且，考虑到最近许多欧洲国家中的有关 Gestapo 的印象，这成为了一个人权问题。1969 年，伴随着德国 Hesse 政府的上台，这些拼凑起来的数据保护法律也出台了。由于技术变化的速度的影响，那些成功的法律都是属于技术中立型的法律。这些法律一般的主题就是调整者（无论是在国家一级还是州一级），那些使用个人数据的用户必须向调整者报告，而且该调整者还可以指导这些人中止一些对于个人数据的不适当处理。实际的影响通常是，一般性法律逐渐通过过多的特定区域的惯例性的规范表述出来。

随着时间的过去，跨国公司的商务处理逐渐变成一个问题了，很明显，单单是本地或者是本国范围内的行动根本不能和跨国公司的业务相提并论，其作用是不够的。随着 1980 年由 OECD 发布的管理方面的自愿规定 [598]，1981 年 1 月，欧洲理事会大会开始维护数据保护机制，其开始实行是在 1985 年 10 月 [206]。虽然，严格来讲，这次大会是自愿参加的，但是许多国家都由于害怕丢掉访问数据处理市场而到会。该主题被建立在欧洲人权大会中，而且签字的国家必须通过国内立法来实现至少是最低限度的安全措施。数据必须通过合法的手段被获得，而且在数据处理上要体现公平性，各国还要保证在出现问题时可以由法律来加

以弥补。

各国的具体实现的质量在很大范围内都不相同。例如，在英国，Margaret Thatcher 通过尽可能少的实现来对付欧洲法律。虽然也有数据保护团体被建立起来，但是由于缺乏资金和专门技术而无法进行任何工作。还有，许多受到优厚待遇的赞助者都可以获得豁免权，数据保护对他们根本不起作用。虽然不是新闻工作者，但如果你在自己的笔记本电脑上保留有鉴别某人的记录，那么就容易在有要求时提供这些个人的信息给数据主体。在对于隐私有严格政策的国家，例如德国，数据保护团体成为了一种严肃认真的法律实施机构。许多非欧盟成员国家，例如澳大利亚、加拿大、冰岛和瑞士，都在 20 世纪 80 年代和 90 年代初期通过了类似的法律。一些像瑞士的国家采用德国的模式，而其他一些像冰岛的国家，则采用英国的模式。

到 90 年代初期，在各国实现之间很明显地出现了较大差异。随着案例法数量的增多，这种情况进一步恶化。这种各国之间的差异性给商业贸易带来了阻碍。对于许多商业而言，解决办法就是避免将它们处理的数据转移（或者是外部采办）到美国。在 1995 年数据保护法令的压力日渐增长 [279]。由于对于高度敏感数据，诸如健康、宗教信仰、民族和政治联系等的严格控制，使得标准性做法的数量比起原先要求各国需要做的工作进一步减少了。还有就是要防止个人信息被传送到诸如美国这样的“数据避风港”，除非出现相应的控制措施。这个方针对于新型商务模型，诸如应用租赁 [182]，是一件令人头疼的事情。

21.4.2 欧洲和美国间的差异

美国方面的历史主要就是，商业设法让政府把隐私权在很大程度上留给自己，从而形成“自我调整”的模式（如果需要了解更多关于这个主题的美国历史，参见 [572]）。虽然，存在一些拼凑起来的国家和联邦法律，但是它们只是针对特定应用，而且在很大程度上不很完整，联系不够紧密。一般来讲，存在于联邦政府记录和通信中的隐私是受到严格控制的，而健康和商业数据则几乎处于无控制状态。不过一到两个控制还是有的，例如 1970 年的公正信用报告法案（Fair Credit Reporting Act），该法案管理被泄漏的信用信息，而且在很多方面都与欧洲的规则有相似之处。除此之外，还有视频隐私保护法案（Video Privacy Protection Act）或者叫做“Bork Bill”，颁布于华盛顿报出版 Judge Robert Bork 的视频租用历史之后，这正是在他被美国最高法院任命之后所发生的事情。

态度也不尽相同。依照 Westin 所说，大约有 25% 的美国人是“在隐私问题上信奉正统派基督教的人”，他们是赞成法律标准出台的；20% 的人对此则毫不关心，而且还乐意传送一些他们的个人信息以获得小额利益；而大多数人，55% 的美国人的做法则十分实用，他们只是根据法律规定来做出隐私方面的决定。但是，人们已经越来越感觉到，自己已经丧失了对个人信息如何使用的控制权。这还是走在欧洲的后边，在欧洲，隐私被认为是一项基本的人权，它是要求有强有力的法律来支持的 [802]。

很明显，当前阶段中的主要冲突就是欧洲和美国关于数据保护的差异问题。美国的政策制定者不能看到问题的严重性，美国国会通常的观点是：“这仅仅是对于赫尔姆斯——伯顿法案的一种恶意报复，而且我们可以在这个问题上进行一些协商。”他们目前对此的希望就是安全避风港的概念，即美国数据处理器可以简单地进入欧洲客户的合约中，或者补充数据将依照欧洲法律处理的影响。一些公司已经做了这件事情，例如 Citibank 使用这种方案在南

达科他州处理德国持卡者的信息。但是对于那些感觉到自己的权力被侵犯的欧盟公民来说，将产生严重的实际实施问题。而且，在法庭上这些人也会败诉，对此的讨论，参见［802］。

21.4.3　目前的趋势

欧洲关于数据节约的调整工作与商业发展的方向背道而驰。远离我们在本章第一节中所讨论的法律实施监督技术，电子商务使用了从 cookie、clicktrail、钱包、IPR 强制工具，以及告密设备（snitchware）等各种措施来跟踪客户和市场的情况，其中的告密识别可以使软件生产商远程监控客户硬盘的情况。这些信息流是单向的，因为你仅仅维持这些数据，而没有权力来组织他人使用个人信息。企业只是将他们的软件许可给你使用，而不是卖给你。一些作者提出，令人害怕的是，无论使用什么样的调整措施，技术将使得我们在这个世界上无处藏身［323］。

关于这个观点的一个极端的情况由 David Brin 提出［139］。他说，普遍深入的监督技术将不可避免地被权力机构使用，所以问题在于这些技术是否也可以被我们这些人使用。他描绘了未来两种可能的选择方案，第一种就是公民生活在类似东德风格的警察机关的十分令人惧怕的环境当中；而另一种是对于官员的任用是由公众仔细审查决定的。将来我们就会知道，到底是监督模式还是 Web 模式呢？

有一些公开的成功经历。其中，美国的信息自由法案（Freedom of Information Act）可能是最显著的，但也存在其他的成功经历，例如对版税返还的尝试（在冰岛和瑞士的一些行政区中），这种尝试大大削减了那些有钱人由于害怕失去社会地位而人为编造低收入从而避免税收的情况。

在这些考虑背后，主要是由于人们逐渐开始理解隐私的经济价值了。基本的问题是对于数据主体的，个人信息的价值是它的边际成本，而对于它的收集者而言，则又是其平均成本。因此，收集者们将花费比大多数使用者用来拒绝使用它们的更多代价来获取这些数据。另一个经济观点是，如果隐私留给技术去解决，那么收集者的花费将大大减少［323］。一线希望就是人们希望维护的私有数据与市场希望收集的数据通常不是同一种信息。个人秘密更趋向于长期性的（例如 10 年前对于酒精中毒的治疗），而市场数据是短期性的（例如如果我将票价降低 20%，那么对于把机票卖给某人可以增大多少可能性）？

也许，部分的解决方案来自某些诸如在线拍卖的工具。但是，有许多地方 Web 模式总是被认为是无法接受的，例如，团体性的研究和开发实验室，律师的办公室和医生的咨询室。定义分界线毫无疑问要包括许多被迫加入或者离开的东西。

在未来几年中，这个问题的发展将引起安全工程师们极大的关注。该问题不仅仅局限在数据收集上，还包括数据的整理校对。例如，美国的重罪是永远记录在案的，许多欧洲国家则具有对一些违法者进行复原工作的专门的法律制度，在这种制度下，根据犯罪程度的轻重，在经过一段不同的时间后，犯罪记录就会清除掉了。但是，既然 Web 搜索引擎仍然存在，这种法律如何继续坚持下去呢？德国对此的反应是，如果你希望引用一个犯罪案例，你应该从法庭获得正式的去除标识后的记录。但是，如果电子报纸档案是可以被在线查找的，那么这么做还有什么好处呢？除非所有犯罪者的身份无法从电子报告中获知。例如，最近，有许多关于监控前儿童性犯罪者的争论，某些州的法律要求犯罪者的注册可以被公众访问。在英国，在一家周日报刊上公布了一些前犯罪人姓名后，引起了骚乱。关于类似的主题还有

很多，从选民名册的允许使用和那些已经加入国籍的人名列表，到是否把某种类型的公开可用信息的列表都公布出来。结果是，即使数据公开，但在欧洲隐私法的制度下面，使用这些数据仍可能是违法行为。

这在美国尤其困难，这里的法庭坚持将第一修正法（First Amendment）解释为你不能阻止在和平时期对账目进行复制，除了一小部分情况例外，一个典型的例子就是诸如有价证卷交易的调控性职业。也许，市场最终将会结束于这个调控性职业，或者对于重复不真实申明的处罚可以高到引起人们注意的程度。虽然，富人和有名的人士通过许多国家的法庭从诬蔑诽谤中获取实实在在的赔偿金，但二者目前好像还没有在大规模市场中出现。我饶有兴致地期待着第一种情况的出现，一些人使得搜索引擎经营者破产，是为了引起公众的注意，把他定罪为使用过期药物。

美国可能将会制定隐私法，从而足以使中大西洋地区能够防止关于隐私问题的商贸交易战争。Al Gore 许诺“电子权力账单”（Electronic Bill of Rights）来保护人们，防止乱用各种类型的可用计算机处理的个人信息。可以想像的到，就像因特网一样，隐私侵犯将立刻达到一个严重规模，而且在美国，舆论将强迫政治家们不去考虑商业利益，而通过欧洲式的数据保护法。还可以想像到的是，欧洲将开始考虑美国隐私实用主义者的一些观点。虽然，一少部分不幸的人将会有可怕的体验，最坏的是在每个星期普通家庭都会得到许许多多垃圾邮件，多到可以用它们来烤肉。但是，还可能的是，欧洲也许对于美国电子商务的实际情况的反应会变得越来越正统。因此，虽然，两个市场也许会融合到一起，但却存在着一种双方都不愿意看到的真正的风险，而且双方市场都没有小到可以被忽略的程度。

在这种情况下，对于电子商务设计人员来说应该谨慎地保证商务处理过程和商务系统可以处理欧洲式的思路，同时也可以处理美国式的思路。

21.5 证据的问题

我提到过欧洲电子签名法令（European Electronic Signature Directive），它强制成员国必须使用批准的产品来提供对数字签名的高质量识别。而且试图将这项提议和已经批准的密钥契约机制联系起来，还试图强加进签名契约的机制，该机制可以破坏数字证据的价值。

但是，摆在安全工程师面前的证据问题，既不是开始也没有结束。设计一个包括证据产生功能的系统比最初看起来要困难许多。

21.5.1 证据的有效性

当法庭在 20 世纪 60 年代首次面对计算机证据时，出现了对其可靠性的关注，从技术的角度和法律的角度都希望知道是否像谣传所说的那样，计算机证据不足以作为证据出现。不同的立法机构对此的处理态度是不一样的。在一些机构中，计算机证据被认为是有资格作为证据的，但在法庭上，另一方可以对此证据提出质疑；在另一些机构中，这些证据甚至不可以被作为证据出示，除非伴随着一个可以证明计算机工作正常的证书声明（当计算机已经被攻击后，这种做法将引起一些问题）。在美国，大多数的相关法律都可以在联邦证据规定（Federal Rules of Evidence）中找到；而在英国，则是 1984 年的警方和犯罪证据法案（Police and Criminal Evidence Act）和 1995 年的全民证据法案（Civil Evidence Act）。

在许多案例中，证据仅仅可以从对机器的操作中获得，因为，机器运作在商业过程的正

常状态之中，而且如果对证据的获取不需要工程师参与的话，将引发一些问题。例如，在一个我经历的案例中，一位女士被控从邮件中偷窃借贷卡（debit card），是否在她钱包中发现了一封 PIN 邮件被撕下的一角，可以就此断定其偷窃卡的罪行。他们让分店的经理将该卡放进办公室的打印机中，输入 PIN 号，然后卡就被没收了。这个经理作证说，该卡被没收就表明是由于账户被封而不是因为 PIN 号码错误。法庭规定这种证据无效。不过，关于这个问题的法律经常发生变化。

21.5.2 证据的可靠性

即使在那些可以被观察的本地情况中，计算机犯罪取证也提出了复杂且不同寻常的工程问题。即使对于那些十分有经验的系统管理员，取得一次具有及时性且无破坏作用的入侵的可靠性证据也是很困难的事情。随着操作系统变得越来越复杂，日志和其他特征更加不透明，这些证据所起的决定性作用也随之降低。法律实施界对此的响应是推出那些为后来的检查做准备的工具，如硬盘的镜像拷贝工具。然而，这并不是事情的结束，因为数据的复杂性和数量巨大，而且多重互操作也经常可能发生。应用所采用的文件格式常常没有足够的文档支持，它们可能会包含 bug 或者其创建者都不愿意讨论的特征，因为他们会尴尬甚至被控有罪。新型小配件，例如使用封闭式操作系统的掌上型电脑，以及 SIM 卡，对此嫌疑人将不会泄漏密码，这将强迫从业人员必须采取某种我们在第 14 章所讨论的反向工程的技巧。法官和其他律师的技术上的不完全将把事情搞得更糟，一般的结果是争论（和审判）搞乱了事实、推测、假定、推论和意见。

当一个案件要依靠技术来判定时，法律系统的信噪比（signal-to-noise ratio）会相当低。通常，对于不公正判决的保护存在于对手的系统本身。回忆一下在 9.4.3 节中描述的 Munden 案件。在一名男子对其银行账户未经授权被提款提出抗议后，他被错误指控并被判犯有欺诈罪。合理的争辩失败了，上诉成功所采取的方法则是高明的，这种方法是获取一个订单，该订单需要银行对防御专家开放其系统，因为这已用于起诉。当银行拒绝时，被告所告银行的声明按规定不被法庭承认，起诉也随之失败了。因此，如果一个系统被当作证据源是有用的话，那么它必须被设计以经受住怀有敌意的专家的检验。我将在第 23 章中更加详细地讨论这个问题。

怀有敌意的专家问题并不是我们可以期望在短期内就能够解决的。在那些由法庭任命专家的国家中，风险就是他们必须来自开发界，从而对保护系统很关心，他们在检测时不应该带有自己的感情因素。通常，我们可以期望用计算机犯罪取证来解决困难的问题。

21.5.3 电子签名

在这个一般不太令人满意的环境当中，许多人希望事情可以通过 gee-whiz 技术（例如电子签名技术）而变得简单一些。这个术语包括加密的数字签名和可代替的其他技术，例如 tablets 技术，使用者可以在上面潦草地写一些手写签名，从而表明对于文档的同意。在一些情况下，例如全球和国家商业法案（Global and National Commerce Act）中的美国电子签名（U.S. Electronic Signature），该法案的目标是对于任何消费者用来表示赞同意见的“声音、符号或者是程序”给予法律强制力。通过按电话机的号盘（“按 0 表示同意，9 用来取消交易”），点击超级链接来进入一个网站或者点击“继续”来进行软件安装，消费者赞成一定要

使用电子合同［709］。

在许多权限当中，情况已经是这样了。在美国和英国，定义签名的特征是签名者的目的，而在电子邮件消息底部的没有被加密过的名字是具有法律强制力的。它可能很容易被伪造，但是手写签名也已经用了许多个世纪［810，811］。

然而，正如我们在手写签名（13.2节）中所讨论的，目前有许多特殊的需求，一些独特的事务，例如房地产、专利权和版权，这些都可以被在线系统所采用。一些国家，例如澳大利亚，已经简单地通过法律来声称，电子笔迹在原先使用手写笔迹的场合中也适用。其他国家，例如英国，已经通过了法律来给予政府权利以对此问题进行措施调整，从而引入电子笔迹。还有一些国家，例如德国，则是利用法律来实现电子签名，从而适应一些已经被放弃的技术标准。这些法律通常都会受到多种客观性的影响。例如，在英国，希望促进软件和系统的使用来支持密钥契约，而在德国希望支持它的智能卡工业。

在美国的不同州中通过的法律没有被那些隐蔽动机所腐蚀，但还是产生了一些混淆的、对立的情况。有时，数字签名为了常规使用而被采纳，而有时被采纳只是为了有限的目的，例如与政府进行通信。有时，它们与特定技术相关，而有时又不是。在［68，335］中有关于数字签名法律的调查。

目前，正在努力从这些杂乱无章的法律中挑选出一些有用的东西。欧盟发布了一个电子签名法令，该法令在2000年1月生效使用，要求各成员国引入一致的立法机构来识别数字签名，数字签名和手写签名在法律上有同等地位。这个方针宣布了两个不同的标准：一个是电子签名，它代表被贴上或者是逻辑地联系到其他电子数据上的数据，以作为鉴别的一种方法；另一个是高级电子签名，它必须：

- 独一无二地链接到签名者。
- 能够标识签名者。
- 能够创建一种使用方法，通过此方法，签名者可以通过自身单独的控制手段进行维护。
- 被链接到数据上，对于这些数据的任何后期的变化都可以被检测出来。

其基本的思想就是，电子签名包括一个被打在电子邮件底部的名字，或者是一个网页中用来同意某项交易的按钮，而高级的不同用法包括数字签名或者是生物遗传签名。法律制定者和合同撰写者应该能够使用全欧洲一致的术语来识别弱的和强的签名机制。

一个令人为难的问题是上述第三个需求不能由当前消费者可用的电子设备技术来满足。考虑到在个人电脑中存在大量的方法可以破坏签名，所以，将签字密钥放在你的个人电脑中是十分轻率的决定，这可能会使你花费一笔不小的开支。智能卡帮不上忙，一个可以通过浏览器软件中的密钥来编写病毒感染你的电脑和签名信息的坏人，可以很容易地感染智能卡阅读器的设备驱动程序，从而当你下次插入卡时得到被签署的假消息。还有，如果智能卡可以被用在停车计时器以及抵押你的房屋上，那么你对于停车计时器的信任度就与你通常给予配偶和律师的信任度差不多了。在缺乏安全的平台上，一些保护可以通过对不同类型的事务采取不同卡或者其他令牌这一传统方法获得，所以客户能够在加锁或者密钥的方式下保留贵重的卡或者令牌（但是，从我自己来看，我并没有看到使用电子方法处理交易所带来的好处，这些年来，我经常这么做，例如抵押房产）。

Bohm、Brown和Gladman曾经说过，“对于方法来说，应该打雷但不闪电，这样是不会受

到致命的责备的，法律的制定偶尔一次超越了事实发展的实际情况，也是可以原谅的”[124]。然而，对于各个国家来说有足够的空间可以添加解释和规章，来教导人们相信它们自己生产或者是其他供应商生产的智能卡是安全可用的。即使被一台不安全的个人电脑使用，目前的迹象也将表明智能卡一般情况下可以工作得很好（我将在23.3.3.1节中详细地讨论这个问题）。其他国家中的商务将必须接受作为结果的“高级”签名为合法。所以，一旦我们拥有了即使其不安全但法官也必须认为其完全安全的数字签名时，人们将面临何种风险呢？

21.5.4 证据的负担

这是一个对于大多数数字签名法（包括美国各州的相关法律）来说都非常深奥的问题。这里他们提出了一个假设，认为那些适应于某种特定的标准（授权类型的智能卡，被许可的TTP证明过的公共密钥等）的数字签名才是有效的。这公然违抗了传统商业惯例，惯例是签名被伪造的风险应该由依赖签名的人承担，而不应该由制造签名的人承担。

如果一家银行通过支票的付款方式来使你的账户成为借方的话，而你又没有签署过该支票，那么对于借方来说是没有合法权力的，并且必须将钱归还给你。通常，如果某人希望强制执行一个文件，但这违反你的意愿，所以伪造了你的签字，你当然否认曾经签署过该文件，然后对他们来说就是要证明该签字是你所签署的或者是你所授权的。这意味着银行和商人可以为自身决定当需要审核签名时应采取多大的关心程度。如果他们决定仅仅对于那些超过1000美元甚至是超过10 000美元的交易审核签名，这就是他们的关心程度，与客户没有任何关系。我在13.2节中讨论过关于手写签名的错误率问题，在实际中，相关的风险是便于管理的。在第19章中，我从本质上解释了，这和信用卡是相同的。虽然客户通常会在开始时冒50美元的风险，但作为报答，他们可以得到在商家破产或者因其他错误而不能交货时，要求信用卡发行者给予赔偿的权力。

可以理解，银行和商人希望清除掉它们可能被暴露的状况，而数字签名法正是用来解决这一问题的一种方案。正如在第19章中所描述的，VISA和MasterCard已经设计了SET协议来通过数字签名支持信用卡支付，而且一些政府以假设数字签名合法来做诱饵，从而让密钥契约机制被采纳[132]。

很明显，从客户的角度来看这是一件坏事情。这里不很明显的是，对于银行来说，任何使用新的安全性措施来降低客户风险的诱饵也应该在对电子商务的公众信心和银行业本身的广泛关注中被抵制。这不仅仅是数字签名的问题。在英国，已经被证实，那些通过公共终端访问Barclays银行的电子服务的人们将可能被下一位在浏览器中按下返回按钮的用户所攻击，银行试图归咎于客户没有及时清除Web缓存中的内容[747]。如果它们在法庭上对证，我们将很有兴趣知道有多少Barclays部门经理知道究竟什么叫做缓存，银行负责人所拥有的精确数据将引起他们的注意，使他们看到目前这类知识对于正确经营零售型银行业务来说是多么得重要。

可以预见，这种风险的降低将减少银行必须创建安全系统的动机，而且很快就会导致法庭和公众都不会容忍的不公平性的出现。回忆一下，那些降低ATM系统用户风险的一些国家中的银行，它们声称任何抱怨“梦幻提款”的客户搞错了或者是在说谎。然后就是法庭很麻烦地将ATM伪造者都送进监狱，从而破除了它们的虚构说法。银行看起来在学习方面行

动比较迟缓，如果客户的数字签名开始被 B2C 模式的电子商务交易当作一项福音所接受的话，ATM 错误可能会大规模地重复出现。

在部署数字签名之前，许多银行就采用电子银行中的术语和条件，在这种情况下，银行交易记录是权威性的；这些银行已经陷入客户法律和广告真实性规章的麻烦之中。这些问题的讨论参见［124］。

B2B 模式的电子商务则是另一种不同的情况了。正如我们在 19.5.4 节中所讨论的，这些年来，已经出现了一些系统，它们在诸如银行间资金转账、安全注册和载货账单的应用中使用数字签名机制。这些好像就是数字签名技术的主要应用领域，至少在短期内是这样。人们也许会假设，大型商业或者让专家来维护它们用以产生签名的系统的安全，或者支付给其他机构来做这件事情。但是，争论仍然不断，尤其是那些不具备这些资源的小型商务公司。对于伪造数字签名技术的责任问题让法庭很难控制，因为大部分的软件公司对于安全性隐患甚至是一些产品中的小 bug 也完全不负责任。因此，对于电子商务系统设计者而言，一种谨慎的做法就是在订阅人合同中就加入争端解决过程的描述，它可以在欺骗行为发生时，足够公平地处理各种极端的法律问题。

21.6 其他公共部门的问题

一个完全的混合式的其他公共信息安全问题正在出现。它们在国家之间差异很大，我将仅仅给出一些例子。

21.6.1 服务交付

一个典型的政府部门，诸如福利机构或者签证办事处，都要对于一个大型的分布式系统进行操作和维护，因为系统实现了其核心的商务功能。然而，政府通常在构思和实现一个大型 IT 项目时做得很糟糕。许多原因是众所周知的。行政人员工资待遇并不高，所以通常不可能同那些绝顶聪明的 IT 员工相竞争。许多政府部门使用传统方法来做事，自动化程度不高。计划和购买周期比起技术周期来说要长很多。管理文化比起理想的情况来说具有更大的反对风险。将一些专门功能排除在外，不容易建立起具有竞争性的机构，而仅仅是利用市场将它们区分开。我所见过的许多事情都是由于公共部门项目的原因而失败的，因为在其核心部分计算机事务和行政事务间产生文化上的冲突。当然，这个问题开辟了两条路：行政事务趋向于将计算机人员看作有雄心的和有进取心的人，这些人希望打乱历史悠久的行政上的弊端，为了自动化而消除自由决定的权力。

随着网络在行政事务管理能力上所起的作用越来越大，这种文化冲突就会越发严重。政治上的领导阶层希望政府服务可以在线交付，而且服务等级可以达到与私营部门接近的程度。投票者则希望这种程度不要太低。然而，通常自动化将问题变得很糟糕。例如，在英国，国家医疗服务机构（National Health Service）以各种方式降低长时间发展起来的对于医生们的需求，例如为了看专家但很难预约的情况。最近，部长们开始强调，患者可以通过电话预约专家，可以预想到的结果就是在患者数量没有增加的情况下，对于预约专家的需求量将大大增加。在缺乏可运转的价格机制的情况下，这将引起混乱。我们已经看到，一些专家医生通过提前退休的方法来作为增加诊治更多患者的压力的响应。

对于安全工程师来说，一个适当的建议是许多被称作由于“安全”或者“隐私”的原因

而不可能的事物实际上是明令禁止的。缺乏这方面的敏感性的结果就是会出现一些令人不愉快的反面作用，是不可能销售产品的。所以，你应该常常试图在表面原因下挖掘一下，来找出系统预期的客户真正所关注的是什么。

21.6.2 社会排挤和歧视

关于政府服务交付方面的一些独立问题是关于穷人和老人的，他们不太可能上网，因此面临减少一部分人的支持。例如，英国政府希望在图书馆和办公室中放置可以公共访问的因特网终端 [758]。事实上，这种做法是对于公共部门在因特网上的竞争力提供资助。

我们将不得不等着瞧这是否可以流行开来。但是因特网的使用在过去趋向于年轻富裕白人男子，所以下一步将如何发展目前还不清楚。女人和长者们是网络用户群体中增长速度最快的一部分，而将邮件和浏览器集成进卫星电视中的做法也加速了网络的发展。也许对于安全工程师最感兴趣的问题是，公共终端所引发的新型攻击可以发展到什么程度。我们在21.5.4节中看到了利用存在于缓存和类似介质中的信息是如何攻击系统的，这里存在大量缺陷。

另一个与访问等同的安全工程问题是，许多嵌入在保护机制中的假设是被歧视的，其原因可能是由于不合法或者不合乎需要。13.8节描述了许多基于生物遗传学的鉴定系统都失败了，因为上了年纪的人和体力劳动者在指纹阅读器上出错误的可能性太高，而且如果指纹或者虹膜系统被广泛应用，那么有残疾的人士将会被排除在外。盲人在其使用 Web 的过程中已经被严重的损害了，许多网站设计者使用一些技巧性措施来防止他们的网页被扫描。这与 shopping bots 有些类似，从网站所有者的观点来看，这也是一种安全措施。

入侵检测系统是另一个有争议的话题，正如我们在18.5.2节中所讨论的，这是一种自动化系统，用来检测欺诈行为，或者伪造保险单申明行为，或者航空旅客容易碰上恐怖主义者等，这些系统最终往往会与一些种族或社会组织相冲突。另一个问题是用来进行大学教育，由计算机专家设计的系统对于那些受教育较少的人来说很难使用。认为使用者是一件麻烦事的态度必须抵制，安全系统，就像任何其他类型的系统一样，需要被设计用来为实际使用它们的人服务。把“使用者”替换成“客户”或者“公民”将是沿着正确路线方向走出了一小步。

21.6.3 税收保护

一个大多数人普遍关注的焦点就是关于匿名 remailer、数字现金和境外的税收避风港结合起来将使得税收工作不可能完成，这将导致国家政府系统的崩溃。也许，对此最为中肯的表述算是 Neal Stephenson 了 [736]，但是在其他评论中也发现了随声附和的声音。这趋向于忽略许多国家都从营业税和进口税中获取收入的事实。欧洲读者习惯于花费超过5美元来获得一加仑汽油和20美元一杯威士忌，对此情况他们将更加怀疑。

21.6.4 选举

最后，在任何民主国家中，最基本的过程就是对于选举的管理。如果它遭到破坏，整个结构都会坍塌。我衷心希望选举安全长官弗拉基米尔·普京作为俄罗斯总统与国家选举报告系统由 FAPSI 运营的事实没有任何关系。FAPSI 是 1991 年创立的俄罗斯信号情报机构，其前

身是克格勃的第8和第16管理局。它的长官Starovoitov将军，据报导曾是一名克格勃人员，他的机构直接向叶利钦总统汇报工作，而叶利钦选择了普京作为他的继任者[327，430]。

如果英国也进入一个电子选举系统的话，我会相当得关注。如果是CESG（GCHQ的一部分），GCHQ在信息保护方面是我们的“国家技术权威”，CESG的构思和审查机制都和GCHQ有很大关联。我在本章引言中提到，英国的关于所有公共部门密钥契约化的策略将引起许多严重的问题：即使机构不是实际操纵结果，它们也将会对找出谁投了Sinn Féin党的票非常感兴趣。但是哪里还有可以被选择的其他专家核心的意见呢？

在美国，这种情况也许不必担忧，因为对于选举的控制是广泛分布的，该项工作被委派给各个州和成百上千的相关机构共同完成。但是，就此安心也不是十分明智的选择。在50个州中获得委派的全部费用将限制公司的数量，只有少部分公司能够出资来向当前本地系统中提供在线继任者。这笔开销大约是每个州要超过100 000美元，从而来设计和由独立专家测试源代码。对于2000年大选的争论也让州立法机构去采用“现代的”在线系统，对这个问题法律机构一直没有停止过思考。如果一家或者两家公司可以最终控制所有或者是大部分州的选举结果的话，那么它们将受到法律的惩罚。

21.7 小结

政府和公共策略通常关注的问题越来越多地闯入到安全工程师的工作当中。许多国家中，对于密码技术的法律控制就是最显著的例证。虽然可能被误导，这些控制产生了许多有害的和不显著的影响，这些影响中最糟糕的可以破坏法律实施和情报之间的界限。其他将威胁到公民自由的界限，包括流量分析和搭线窃听之间的界限、版权和审查机制之间的界限、实施版权和其他一些事情之间的界限，例如零件控制。增加透明度的工作也许将比采用特定技术解决某些问题更加具有战略意义。

然而还存在许多其他的问题。工程师们必须注意对于个人数据的保护、系统产生的证据的质量、版权法问题、社会排挤和歧视问题等。还有一些机制是我们必须处理好的，例如用于选举中记录选票的系统的完整性。

研究问题

技术问题趋向于包括科学、工程学、实用心理学、法律和经济学之间的复杂的相互影响。在跨学科的研究方面还比较少，本章开始处的格言很好地抓住了这个问题，来自各个领域的人常常是从一个角度谈论问题。对于诸如密钥契约问题的争论正慢慢创造着这样一类人，他们同时具有计算机科学和法律学两方面的经验。电子商务使得计算机科学家和经济学家交谈。加速这一过程的开端几乎都是一件好事情，引入了心理学家、科学历史学家以及其他正面的事物。还不清楚的只是如何在目前的学术和工业研究组织的结构下来做这件事情。

参考资料

对于技术上的争论来说，很容易就从现实中分出一条或者几条岔路。许多被魔法召唤来的恶梦引起关注和花费金钱，就像现代生活中那些类似出现在中世纪地图中的用来掩盖地图制作者的无知的妖怪一样。一个希望所创建的东西可以持久工作的工程师有责任不要失去自制力。由于这个原因，发现那些主要的资料源就成为一件相当重要的事情。这些资料就是由

有经验的权威人士例如 R.V. Jones [425] 和 Gerard Walsh [787] 所写，在制定诸如 Whitfield Diffie 和 Susan Landau [250] 这类政策时都与上面的资料有所关联，这两个政策是政府报告，对于某些策略的形成，例如 NRC 中关于密码策略的学习 [580] 和对于主要资料的编辑（诸如 [684]）的影响很大。

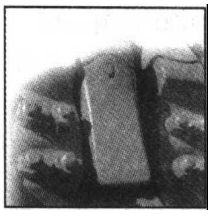
还有一些机构，诸如 EPIC [266]、EFF [264]、FIPR [304]、CDT [173] 和隐私交换（Privacy Exchange）[628] 组织的网站及许多像 politech [619] 和 ukcrypto [755] 这样的邮件列表上面都有许多有用的资料。

关于计算机犯罪取证，我所知道的最好的书籍是由 Tony Sammes 和 Brian Jenkinson 所写的 [664]。Peter Sommer 也有一篇关于犯罪取证和证据问题很好的文章。其中证据问题是由于对一些英国青少年攻击 USAF 罗马空军基地进行控告时出现的 [722]。司法部门的“关于搜查和掌握计算机的指南”也关注了一些问题 [245]。如果希望收集有关计算机犯罪历史的案例，参见 Peter Neumann [590]，Dorothy Denning [235] 和 Donn Parker [602]。

在数据保护问题上，有许多可读的东西，但我不知道哪一个可以作为简明的向导性读物。[802] 中，Alan Westin 提供了一个历史性的概貌，从而描述了欧洲和美国之间产生冲突的过程。Simson Garfinkel 的 [330] 和 Michael Froomkin 的 [323] 调查了美国隐私和监督方面的问题。

目前，有许多关于电子投票的文章。这个问题在很大程度上与通过邮件或者电话投票是相同的，但也不是完全相同。Mike Shamos 在 [693] 中做了关于电子投票需求和一些容易误入歧途的问题的调查。而 Roy Saltman（在 NIST 做了许多年的权威人士）讨论了一些在美国出现错误的问题，以及各种 NIST 建议，参见 [663]。在 [152] 中是关于在美国加利福尼亚州进行因特网投票的可行性报告。最后，Lorrie Cranor 在 [209] 中提供了一个十分有用的关于电子投票的链接列表。

第22章 管理问题



我的经验是，对于开发人员来说，只要具有诚实和表达能力这些安全领域必备的素质，那么他就可以创建一个防护严密的机器。他们也不必要非是安全领域的专家，但是他们必须理解将要创建什么以及它将如何工作。

——Rick Smith

由于技术和系统变得越来越普及和深入，我们今天所面临的一个最为重要的问题是可能会失去那些优良的、富有人性的特性，而这些特性将可以很好地区分成功与失败、公平与不公平、正确和错误……没有一台 IBM 计算机曾经受过这些人文学科的教育。

——Tom Watson

对于管理而言是不存在运算法则的。而在有运算法则的地方，应该叫做经营。

——Roger Needham

22.1 引言

到目前为止，我已经大致描述了各种安全应用、技术和人们关注的各种安全问题。如果你是一名 IT 经理，别人给你钱让你创建一个安全系统，那么你现在就应该去寻找一种系统的方法来选择保护目标和保护机制。这将把我们带入有关系统工程、风险分析和威胁评估等方面的主题。

商学院提供的经验是，管理培训应该在很大程度上通过学习历史案例来进行。如果只将注意力放在基础性学科，例如法律学、经济学和会计学上，那么学习就会变得很生硬。我在本书中正是遵循这个模式进行的。我们仔细分析了一些基本原则，例如协议、访问控制和密码机制，然后又了解了许多不同的应用。现在，我们必须将所有的思路集中起来，讨论一下如何处理一个实际的安全工程问题。组织问题和技术问题也会在这里被讨论。了解将来运营你的控制系统的员工的能力也是很重要的，包括警卫和审计师，还要考虑在他们身上的来自管理和工作组的压力，以及在系统投入运行后如何从他们那里获取反馈信息。

22.2 管理安全项目

安全项目管理工作的核心通常是需求工程，需求工程可以用来计算出什么需要保护以及如何对它们进行保护。在做这件事的时候，理解风险和报酬之间的平衡是至关重要的。安全人士有一种独特的惯性思维方式，就是把注意力过多地放在过去的事物上，而往往忽略了以后的事物。如果一个客户具有 1 000 万美元的营业额，100 万美元的利润和 15 万美元的盗窃损失，那么安全顾问会建议你，当通常股东们真正获得的利益将营业额增长为 2 倍，即达到

2000万美元时，尽管盗窃损失可能会变为原先的3倍即45万美元，但利润也会增长15%。假设边际收益保持不变，而利润达到185万美元，比原先增长了85%。关键问题就是，不要总是认为对于一个漏洞惟一可能做的就是去弥补它，不要相信那些只会说“必须加固安全系统”的顾问。通常情况下，系统的安全性已经足够稳固了。

22.2.1 三家超市的故事

我举一个关于三家超市的小型案例来证明这一点。在运营一家超市的庞大费用之中包括检查人员和安全人员的工资费用，以及由于偷窃而导致的库存报废损失。检查延迟同样是问题恶化的重要原因：仅仅通过减员并不能解决问题，而且使工人更加辛苦的工作将意味着股票进一步缩水。那么也许技术上的措施可以帮上什么忙吗？

一家在南非的超市决定完全实行自动化。所有的产品都贴上RF标签，因为整个购物车将被自动扫描。如果这么做成功的话，那么将达到一石二鸟之效：相同的RF标签将使得偷窃变得越发困难。虽然思路不错，但还是无法和后面我们要提到的barcode技术相竞争。客户们必须使用一种特殊的购物车，这种车既大又难看，而且RF标签也是需要一定费用的。

另一家位于欧洲国家的超市认为，它们的损失大部分来自于一批职业盗窃者，因此考虑建立一个面部识别系统，从而在一位经常关顾的公民进入超市时提醒保卫人员特别注意。但是，通过当前可用的技术，错误识别率还是很高，所以这项技术没有多大作用。最后，超市所选择的方案是民事补偿。当商店扒手被抓住时，在当地的法律机构对其进行也就是相当于一顿午餐的小额罚款后，超市还要将他告到民事法庭以希望得到被浪费时间、损失的收入、律师费用和其他可能想到的任何事情的赔偿。这些费用加起来几乎可以抵得上一辆汽车的价格，然后他们就可以到扒手家中搬走他的全部家具。到现在为止，这种思路一直都还不错。但是，他们的管理将过多的精力放到如何减少损失方面，而不是放到如何增加销售额上面。最后，他们开始失去市场份额，然后看着他们的股票价格下跌。将目光转向一个基于安全的解决方案也许是该超市营业水平下降的征兆，这要比把它说成是一个原因要恰当许多，这种做法确实应该对超市出现目前的状况负一定责任。

在英国一家名为Waitrose的超市看上去情况要好一些。该超市引入了自助式销售扫描。当你进入超市时，需要在一台机器中刷一下你的会员卡，该机器会向顾客分发一个便携的barcode扫描器。当你从货架上拿下商品时，对其进行扫描，然后将商品放入到购物袋中。在检验台，你交还扫描器，然后得到一张购买商品的打印列表，再刷卡，最后就可以去停车场了。这种做法也许看上去比起当年传统商家将货物摆在柜台后面的做法冒险许多。事实上，这其中有许多精巧的控制机制在工作。将服务限制到会员卡的所有者身上，不光可以使管理者们将已经知道的商店扒手排除在外，还可以帮助会员卡广开销路。要想拥有一张会员卡，你必须获得一个可信任状态，这种状态对于你在商店购物时遇到的邻居都是可见的。相反，丢失会员卡（可能是由于因偷取被抓住或者是另一种更可能出现的情况，即拖欠付款）将是一件令人尴尬的事情。而且，那些受到信任的人们将不会再有欺诈的想法出现，因为这么做对他们来说并没有什么好处。当然，保卫人员还是应该在显示屏前注意那些可疑的逗留在超市中几百英镑一瓶的高档酒架之前的顾客，在嫌疑人到达检验台时，系统最容易查出问题所在，这种做法给了员工一种非直接面对顾客的方法来重新检查其购物包中的物品。

22.2.2 平衡风险和报酬

商业的目的在于利润，而利润正是相对于风险性的报酬。安全机制通常对于风险/报酬等式起到十分关键的作用。但是，最终正确处理该平衡关系的还是公司的董事会。在这种风险管理工作中，董事们将制定出各种策略，听取包括律师、保险精算师、安全工程师等人的建议，以及听取他们的市场、运营和财务部门的建议。一个合理的团体风险管理策略要比对于信息系统攻击的运营风险涉及更多的东西。还有那些非IT的运营风险（例如火灾和水灾）和法律风险、汇率风险、政治风险等许多其他风险。公司老板们需要从全局的角度做出明智的选择。他们工作中很困难的一部分就是注意到那些来自不同学科的顾问们的观点往往具有局限性，不够全面。

这些顾问需要相互理解彼此的角色，然后一起工作，而不是试图破坏他人的工作。但是，如果公司老板不问难于回答的问题，而且即使问到一些也不会太深入的话，那么这些顾问们就会彼此之间友善相处，从而维持一个大多数都同意的意见，而这种意见往往是和实际情况背道而驰的。指派给安全工程师完成的最有价值的任务是，作为一个中立的局外人来对集体审议提出自己的意见。而完成该任务则需要极强的外交技巧。事实上，在我亲身经历的顾问工作中，大概有三分之一的情况是这样的：在客户公司中，至少有一个人对于问题是什么和如何修复这个问题是相当清楚的，他们仅仅需要一个可靠的雇员来说服大多数不愿意改变意见的同僚们的观点（这就是为什么那些总能够提出高质量和确定性建议的著名的咨询公司通常都在专家方面具有优势的原因。然而，在那些确实需要专家的场合中，工作只注重“外表的服装”，而不注重内涵，所以一些看上去相当引人入胜的东西也可能出错）。

虽然，政府中的目标和管理结构也许会有少量的差异，但是它们却使用的是完全相同的原则。风险管理通常比较困难，因为人们通常都愿使用符合一系列标准的方法（例如桔皮书Orange Book），而不愿使用逐个案例的需求工程（case-by-case requirements engineering）。James Coyne和Norman Kluksdahl在[208]中阐述了一个关于信息安全的典型案例的学习，这个信息安全机制被疯狂应用于美国的国家航空和宇宙航行局（NASA）。在那里，军方在航天飞机运营中财政预算的增加导致在休斯顿的任务控制中心（Mission Control Center）建立起一个安全小组，他们来填补DoD解散后留下的空缺。这个小组被赋予了极大的特权，成为独立于开发和运营组之外的组织。该小组的不合理要求逐渐变得与预算和运行限制所规定的种种内容毫不相干。而且它与其他机构的关系也越来越呈现对抗性。最终，不得不取消掉这个小组，而实际上该小组什么事情也没有能够完成。

22.2.3 组织问题

虽然本章是关于管理话题的，但是对于如何培训和评定保卫人员并不像如何去创建一个可用系统那样得到更多的重视。然而，你也有必要来理解保卫人员（以及审计师和其他检验人员等），否则你的工作将是不彻底的。许多系统的失败是由于它们的设计者们对于实际操纵系统的人员的能力、动机和纪律性没有进行符合实际情况的假设。这种假设不是一次性分析就可以解决的问题。例如，一个起初的低欺骗率将导致人们洋洋得意而疏忽大意，直到突然事情激发才恍然大悟。还有，一个机构中外表上引起的变化，例如合并和收购，也将破坏控制。

许多令人惊讶的人们品德上的缺陷将影响他们在机构中的行为方式，而且在你的设计中必须对此予以宽容。

22.2.3.1　自满周期和风险自动调节

组织中的自满所造成的影响可以通过出现在美国的电话欺骗得到很好的证明。这是一个有7年周期的事件：在任何一年中都会有一个“Baby Bells”遭到严重的伤害。这导致了它的管理者雇佣专家清理很多事情，而且将任何事情都置于控制之下。此时，那些未被攻击的“Baby Bells”当中的另一个很有可能会成为下一个被攻击的目标。但在接下来的6年中，情况松弛下来，然后回到第一年时的状态。

人们已经做了一些关于如何控制风险曝光时间的相关的工作。Adams研究了强制性安全带法律的影响，而且得出的结论是这些法律实际上根本不能挽救人的生命：这种做法只是将伤亡从拥有汽车的人转向了那些行人和骑脚踏车的人罢了。汽车安全带使得司机感觉更加安全，所以他们开起车来就会更快，从而将他们可以察觉到的风险恢复到原先的水平。Adams把这个叫做风险自动调节（risk thermostat），这个模型也被其他应用所证实[8，9]。自满周期（complacency cycle）被看成是风险自动调节的完全的体现。无论这些现象如何描述，风险管理仍然是一种交互式的行为，它包括对于所有类型的反馈和补偿行为的控制。作为这种控制结果，伴随着公路交通的意外事故，系统可能更加稳固；或者伴随着Baby Bells的周期性作用变得更加振荡。

反馈机制可以在一些风险降低系统的性能方面提供系统的限制。通过第10章中描述的警报系统以及在18.5.3节中讨论的入侵检测系统等防护措施，那些攻击事件或者意外事故，又或者是其他原先希望防止出现的事务的影响范围将减少到“没有足够的攻击”的水平线上。也许，系统总会达到一个平衡点，在该点上所有的哨兵都去睡觉了，或者真正的警报被那些非警报信息所淹没，又或者是预算达到了甚至是超过了一个危险点。如何使用技术来调节这个平衡点目前还不清楚。

风险管理也许会成为世界上一个最大的产业。它不光包括安全工程师，还包括火灾和伤亡服务、保险机构、道路安全业和大部分的法律职业。然而，令人吃惊的是，对于这一主题，我们所了解的东西还很少。工程师、经济学家、保险精算师和律师们都会遇到来自各个方面的问题，使用不同的行业语言并且得出相互矛盾的结论。其实，文化因素也起着很重要的作用。例如，如果你将风险看作是那些知道成败可能性有多大而不知道结果会如何的事务，而将不确定性看作是甚至连成败可能性都不清楚的事务，那么大多数人则会倾向于风险性而不去考虑不确定事务。在成败可能性可以显而易见地直接判断的时候，处理风险通常采用直觉。但是，当知识不清楚或者不确定时，人们则会产生出各种恐惧和偏见。所以，也许最佳的良药就是教育。但是，对于安全工程师来说，还是存在一些特定的事情应该去做，或者一定要避免。

22.2.3.2　和可靠性交互

导致机构中出现作用拙劣的内部控制的一个重要原因就是系统的正确性不是足以信赖的，所以，大量的事务总是出现错误，必须通过手工的方式来纠正。而对于这种混乱情况非常高的宽容性破坏了内部控制，因为对于许多保护机制而言，它导致了高的错误报警率。它还将引诱职员放松警惕，使他们认为：如果自己没有发现错误，那么偷窃行为也没有发生。

一个重复的主题就是关于质量和安全二者之间的相关性。例如，有资料显示对于软件质

量的投资将减少计算机安全问题所影响的范围，无论安全性是否是高质量程序的一个目标，情况都是如此。而且，从安全的观点来说，最高效的质量措施是代码演练（code walk-through）[292]。或者通过阅读或者通过被批评而获取的知识将对许多程序员产生有益的影响。

当你试图获得客户董事会关于某些保护措施的支持时，可靠性将是你最大的一个卖点。错误将对客户业务造成巨大的经济损失。也没有人真正理解软件到底是什么。如果错误被发现，欺骗行为就应该显而易见了。当然，所有这些情况都应该通知最高管理层，而不要因为麻烦而不去做这件事情。

22.2.3.3 解决错误问题

当面对一个难以处理的问题时，人们往往会猛烈地攻击相关且更加容易的问题。我们在21.2.5.3小节关于公共策略的内容中看到过这个问题所带来的影响。在私营部门里，置换活动（displacement activity）也十分普遍。一个例子来自智能卡产业。正如我们在14.7.2节中所讨论的，防止智能卡遭受微探针（microprobe）攻击的困难使得业界将注意力集中到保护与智能卡业务相关的其他东西上面。即使是厂方访问者，也需要在接待会上签署保密协议（nondisclosure agreement，NDA）。只有这样，该访问者才可以使用编程手册。大多数技术资料是完全不可用的，而且设备制造商拥有几乎是核心一级的物理安全性。物理安全性过度的杀伤力也许对天真的用户来说，印象最为深刻。但是，几乎所有的对于智能卡系统的真正攻击都是使用探测攻击的方法，而不是利用任何类型的内部信息。

人们解决该问题的驱动来源于无法处理不确定性的事务。管理者希望使用在checklist控件的小格子里打勾来解决问题的方法。如果一个机构需要处理一种当前的风险，那么必须找出一种方法来维持这个控制过程，以及防止其变为checklist中一个固定不变的条目。但是，仍有不少看法认为应该使用checklist来代替过程，因为它们要求更少的管理方面的关注和努力。我在7.6.6节中说过，军方系统墨守成规的指南具有很强的取代关键性思考的趋势。设计者们不是去考虑仔细检查系统的安全需求，而是仅仅伸手去拿他们的checklist。对此商业系统也没有太多的不同之处。

另一个有关组织的问题是，当曝光的程度具有政治敏感性时，一些伪装就可以使用了。典型事例是这样一个问题，即判断攻击者来自内部还是外部。我们在一个又一个的系统中已经看到，攻击者来自内部人员是主要问题，无论是由于他们中的一些人怀有恶意或者是由于他们中的大多数都粗心大意。但是，过于公然地违背管理者和IT员工的意愿而强制实施控制的做法也是草率的，因为，这将导致系统与他们格格不入，而且通常很难让他们自己来管理这些控制措施。同时，也很难将产品卖给一个需要适当的防止内部攻击的典型公司的董事会了，因为这意味着打击了向他们报告的员工们的诚实性和可靠性。

因此，安全管理者要不断寻求获得大量的财力来防止根本不存在的“邪恶黑客”。所以，她可以将其中的大部分精力花在控制上，从而管理真正的威胁，也就是不诚实或者粗心大意的员工身上。关于这个策略，我将保持十分谨慎的态度，因为，没有清晰理由的保护机制很容易在操作的重压下受到侵蚀，尤其当这些机制被看作是官僚政治所强迫接收的事物时。通常，我将采用一些精妙的和谈判式的技巧，而且将会把控制市场化，从而作为一种减少错误和保护员工的方法。银行管理者喜欢双重控制的保护锁，因为他们知道，这将减少其家人被当作人质的风险性。而且还需要在事务中通过有限的方法来加入两个签名，这样一来，当出现错误时，就可以多一个肩膀来承担了。但是，这些适用于保护措施的言论对于其他地方就

缺乏作用了。

22.2.3.4　不合格或没有经验的安全管理者

即使IT安全管理者十分有能力，情况有时也会很糟糕。她必须使用各种计谋在那些管理同僚们认为是纯收入的活动中增加投入。在实际生活中，情况会变得更差。许多传统的公司中，升职到最高管理层依靠的是资历和人际关系。所以，如果你希望升到CEO，那么你就必须在公司花费20或者30年的时间，而同时又不能冒犯太多的人。一个安全管理者绝对是你最不愿意做的工作，因为，这将意味着你随时都在向别人说不。一点都不令人奇怪的是，在美国政府机构中，计算机安全管理者的平均任期仅仅为7个月［384］。

这件事情在那些重新组织的机构中显得更为复杂。在那里，中央计算机安全部门也许每隔几年就会被创建和撤销，而IT审查功能也在IT部门、内部审查部门和外部审查人员或者顾问之间来回调整。安全功能比起其他商务过程来说，得到的是更少的持续关注和分析思考，更多的则是屈服于由于敬业精神而产生的在格子中打勾的做法。

22.2.3.5　道德冒险

公司通常设计系统使得风险可以由第三方当事人负责。我在第21章中曾经提到，一种对于数字签名机制的攻击就是将那些与被伪造签名相关的风险由信赖方转向所谓的签名人身上。例如，将大部分与在线银行相关的风险从银行转到客户身上。我还在第9章中讨论过，一些国家中的银行声称，它们的自动柜员机不可能出现错误，因此对此产生的任何争议都是客户自身的错误。

除了公共策略方面的问题，以及我将在22.6节中所讨论的宏观经济的影响外，这对于那些内部员工关系产生滑坡的公司来说都是具有影响的。为了使人们应该多加小心地转移动机而产生一种道德风险（moral hazard），这要求公司在适当的风险管理技术上面投入资金。更糟的是，如果一家公司的策略是否认某些特定类型的欺骗手段出现的可能性，那么这种做法将使得自己完全开放给员工，于是一些员工开始欺骗公司，因为他们知道起诉自己对于公司来说将是一件十分尴尬的事情。

当那些做出系统设计决定的人不愿意为他们的行为承担责任的时候，就会产生一种稍微有些不同的道德冒险。这有许多可能的原因。IT员工的营业额可能很高，这其中很大程度上要依靠那些合同工；如果有一位新出现的管理新星，没有人愿意和他争论，那么他只能算是设计组中的使用者而已；或者即将到来的业务处理重建也许会让那些忠实的员工变成暗地里寻求其他职业的求职者。无论如何，当你设计一个安全系统时，一个不错的主意是考虑一下你的同僚们，并且问一下自己，如果在三年后系统出现问题时，他们中有哪些人可以承担这个责任。另一种动机上的错误还会发生在，当机构的某个部门由于某些处理过程所产生的利润而受益，而其他部门则需要对该过程带来的损失而支付相关费用时。通常，市场部门由于销售额上升而得到夸奖，而财务部门则需要背负可观的债务。一个人也许会想，他们应该在这些部门之间保持一个风险和报酬的平衡，但是通常都不会这样做。上面所提到的三家超市的例子，仅仅是众多例子中的一个而已。这些公司将通过一段时间才可以从风险承受者发展成为可以抵制风险的实体，而很少会成为对风险采取反击的实体。Adams在［9］中提到，经历风险和风险抵制都和个体类型有很大关系：前者趋向于利己主义者、公司的企业会计；而后者则倾向于等级制度中的统治者。因为后者常常可以支配官僚机构。并不使我们感到奇怪的是，稳定的、已经被创立的机构都趋向于风险抵制而不仅仅是迎合那些合理的经济学中

的理论。

有哪些工具和概念可以帮助我们拨开官僚政治中的迷雾，从而从基本原理中决定一个系统在保护方面的真正需求呢？

本章的剩余部分将按以下顺序组织。在下一节中，我们将看一看基本的方法论问题，诸如自顶向下的和反复的开发方法。在此之后，我将解释如何将这些东西应用到特定的安全需求工程中去。在看过这些之后，我将回到风险管理的问题上，看一看技术方面的工具。然后，我还要讨论一些关于经济问题的话题，最后讨论那些容易出现错误的问题。

22.3 方法论

大型的软件项目通常要花费比预计更长的时间才可以完成，花费也会比预算要高很多，而且最终产品中的 bug 也会比期望的多（有时这被以胡夫金字塔的创建者基奥普斯命名为 Cheops 的法律）。到 20 世纪 60 年代，人们开始谈论软件危机的话题，虽然危机这个词对于那些到目前为止已经经历了两代时间的事务（就像计算机不安全性）的出现有点不太恰当。但不管怎样，软件工程这个词由 Brian Randell 于 1968 年提出，并定义如下：

> 软件工程是关于良好工程原则的创立和使用，为了获取经济的软件，这些软件十分可靠，并且在真实的机器上面可以高效地运行。

这个概念包含了人们的一种期望，即采用一些被证实的科学基础理论和一系列设计原则，通过造船和飞机所采用的相同的方法来解决问题 [583]。从那时起，虽然从来没有一个完全成功软件工程，但是也取得了很大进步。

软件工程是关于管理复杂性的理论，而这种复杂性有两种。一种是伴随复杂性（incidental complexity），它包括在编程过程中使用不适当的工具，例如汇编语言是所有早期的机器都支持的语言，编写一个现代的、具有图形化用户界面的应用程序时，如果使用这种语言将是一件极其单调乏味且很容易出错的工作，几乎不可能完成。另一种复杂性是处理那些大型和复杂问题的内在复杂性（intrinsic complexity）。例如，银行管理系统将包括上千万行的程序代码，对于任何一个人来说，要想完全理解其含义都是一件十分复杂的事情。

内在复杂性在很大程度上可以通过技术工具的使用得到处理。这其中最为重要的是高级语言，它隐藏了许多处理机器特定的细节的复杂操作，从而使程序员可以在一个适当的抽象层次上开发代码。还有一些形式化的方法可以帮助我们查找出特定的错误模式，从而使程序员的工作被检查。最明显的安全工程的例子是 BAN 逻辑，用来检验密码协议，这一点我在 2.7 节中已经描述过。

内在复杂性常常需要方法论的工具，从而可以将问题分解为可以管理的子问题，而且限制这些子问题之间相互交互的程度。许多在市场上可以买到的工具将帮助你完成这项工作，你所利用的工具将很可能成为客户策略。但是，这里有两种基本的方法：自顶向下和反复。

22.3.1 自顶向下设计

系统开发的典型模型是瀑布模型，这个模型是由 Winston Royce 于 1970 年为美国空军设计的 [653]。其思想是从一个系统需求的简单说明开始，然后再对其详细描述，使需求变成

一个详细的规格说明，实现和测试系统的组件，作为一个系统进行集成和测试，然后生产系统并投入实际操作当中去运行（见图 22-1）。

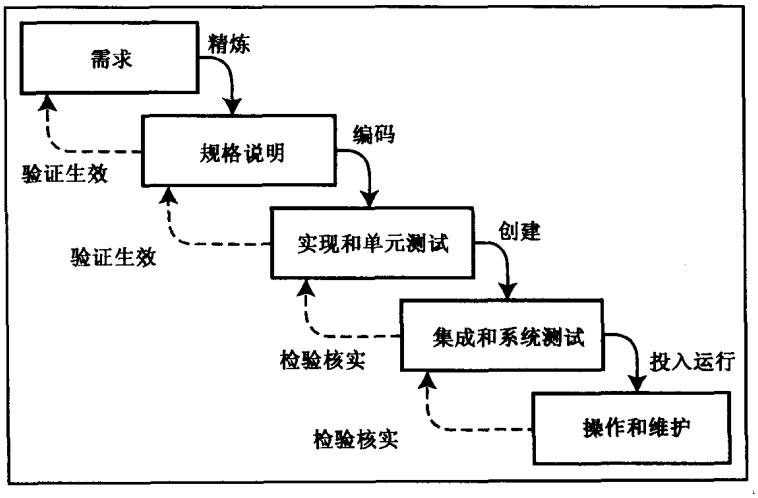


图 22-1　瀑布模型

在这种设计模型中，需求使用用户语言编写，而规格说明则采用技术语言编写；单元测试检查各个单元是否与规格说明相违背，而系统测试检查是否需求都得到了满足。在这个链中最开始的两步，存在着关于是否创建了一个正确系统的反馈（验证生效），而接下来的两步中的反馈则是确定是否正确地创建了系统（检验核实）。实际实现时，也可能不止这四个步骤，一个通常的表述是，随着需求被进一步发展成为更加详细的规格说明而产生的一系列更详细的步骤。但是在这里只是顺便提一下罢了。

关于瀑布模型中最关键的内容是，开发流不可抗拒地从需求的第一条语句到实地系统进行部署。尽管每一阶段都有反馈给前一阶段，却没有从系统测试到需求的系统级反馈。因此可以认为这是瀑布模型的优点，也是它的弱点。

瀑布模型的优点在于，它必须要求对于系统目标、结构和接口的早期澄清；通过提供明确的目标，从而使得项目管理者的工作更加简单；同时通过将逐个的花费加入到每一步骤中也可以增加开支的透明性，这对于任何规格说明中的后期变化也是如此；它还和一系列十分广泛的工具都是兼容的。在可以使用这个模型的地方，通常这都是最好的方法。关键问题是是否任何开发和原型工作中的需求都可以事先详细地被了解清楚。例如，有时我们要编写一个编译器或者（在安全领域中）设计一个防篡改密码处理器来实现一个众所周知的交易装置，并且需要通过特定级别的 FIPS 评估。

但是，一般细节性需求事先是不清楚的，所以一个反复的方法就是必需的了。对此，存在一些可能导致这种情况发生的原因。也许，需求就连客户也不清楚，而且原型对于澄清这些疑问是必需的，而不是引起更多的争论；技术也许在不断变化；环境也在不断变化；或者项目的关键部分包括诸如人机界面之类的特征的设计，从我们的经验中可以知道，该特征将包含几种原型（不论对于一个保护系统的内部细节进行多么好的工程化处理，用户界面的问题也将被考虑，而且如果业务模型允许，引入一个导航会是明智的选择）。

22.3.2　反复设计

许多开发项目仅仅需要一种反复的方法来设计，但是这种反复也许永远不会因设计者满意而停止。你可能为客户创建一个原型系统，客户支付费用，然后说："不，我希望采取这种方法来代替目前的方案。"那么，你就必须创建另一个系统，从而又会出现其他的缺陷。所以，总也不可能将所有事情都处理好。

通常有两种方法来处理这种情况。第一种是 Barry Boehm 的螺旋模型。在这个模型中，开发过程按照事先协商好的反复数量进行。在每一个反复过程当中，都要创建一个原型并且进行测试，管理者可以通过这些来评估每个阶段的风险，从而决定在下一个反复过程中应该做些什么或者如何来减少损失。之所以把这种方法叫做螺旋模型是因为，整个过程通常被描述成图 22-2 所示的样子。

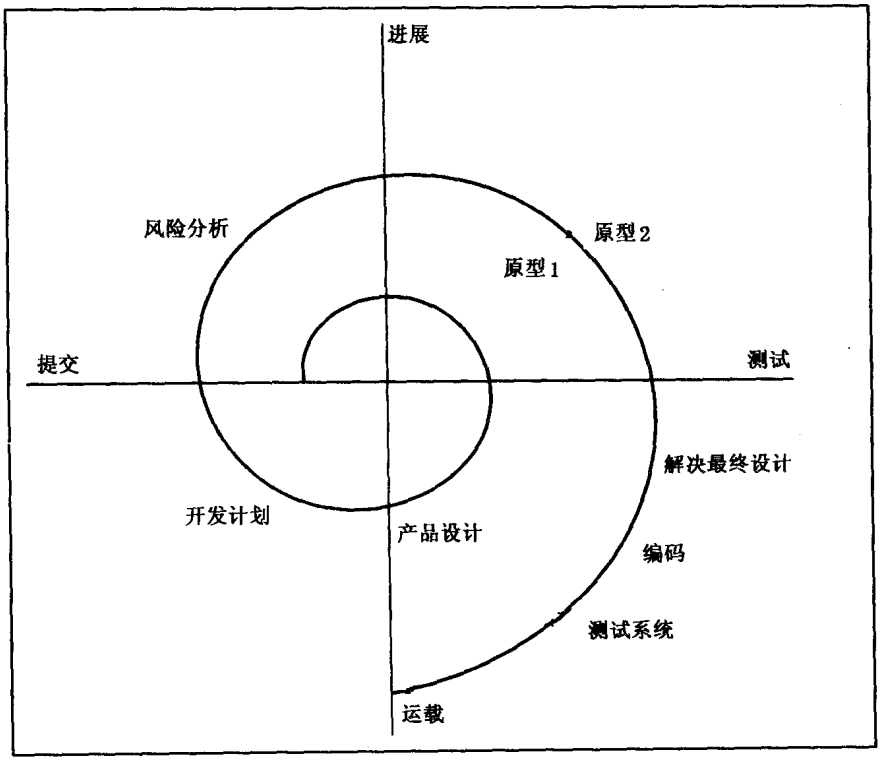


图 22-2　螺旋模型

另一种常用的模型是演化开发。该模型的重要性已经逐渐增加，因为这正是打包软件行业的常用工作方式，而且最近在"极限编程"的呼声下变得十分流行。不幸的是，这种方法往往在理论课程和软件工程教材中被忽视。

随着软件从为正规项目开发的预约软件（bespoke software）向着包（package）的方向发展，软件产品变得更加复杂，以至于软件不可能再从无到有的进行开发（或者是二次开发）。软件包的作者们不断将众多的功能加入包中，使其适应更加广泛的市场需求。确实，微软已经不止一次地来尝试重写 Word 软件，但是每一次都不得不最终放弃（也许，关于演化开发模型最好的书就是由微软管理者 Steve Maguire 所写的 [521]）。在这个观点中，产品并不是一

个项目的结果，而是一个过程的结果，这个过程包含了对早期版本的不断修改。

关于演化开发，最为关键的一点就是像生物物种的每一代都必须通过生育来让物种繁衍下去一样，每一代正在发展的软件产品也必须具有生育能力。完成此功能的一项核心技术是回归测试。每经过一段时间，也许一天一次，所有被分工于同一产品升级的不同功能部分的小组成员都要检查他们的代码，然后进行编译并创建成可执行程序，再通过输入一大批数据进行自动测试。这个步骤将测试程序是否可以正常工作，并且确定那些原先已经清除掉的bug没有再次出现。当然，通常一个被创建的应用根本就不能工作是可能的，而且可能存在一个相当严重的错误被当作主要的修改问题实现。无论如何，我们总是编写可移植代码，这些代码可以被用在beta测试版本中，也可以用于软件开发后期的任何阶段中。

测试技术也许是20世纪90年代软件工程实际应用中最大的改进。在自动回归测试被广泛使用之前，工程师们估算在修复的bug中有15%引入了新bug，或者是重新引入了老bug[7]。但是，出于许多原因，自动测试对于安全工程师来说没有多大用处。安全特性更加分散，而安全工程师数量有限，所以需要在测试工具上投入更多的资金。此外，那些可用的工具相比于可以被真正用于通常的软件工程界的工具而言不够完整，也过于原始。许多我们希望找出并且修复的错误，例如栈溢出攻击，都趋向于出现在新特征当中，而不是旧特征中。特定类型的攻击通常也很容易通过使用特殊的补救措施来进行修复，例如在4.4.5节的栈溢出例子中所提到的淡黄色标识（canary）。许多安全缺陷是由细小的、来自系统抽象层的bug所引起，例如，当规格说明错误与用户接口特征相结合，这类错误就很难通过设计自动测试检测出来。但是回归测试仍然十分重要。它可以发现那些并不能完全理解的受变化影响的功能性。

相同的机制也被应用于安全关键型系统（safety-critical system）中，这种系统和安全系统在许多方面都是相似的。可以从这些系统中吸取一些有用的教训。

22.3.3　来自安全关键型系统的教训

关键型计算机系统可以被定义为那些面对所有可能的故障原因，有一些特定种类的故障原因可以被避免的计算机系统。由于这个特定种类的故障原因不同，系统可以分为安全关键型、业务关键型、保密关键型、环境关键型或者其他等等。关于安全关键型的明显例子包括航班控制和自动刹车系统。关于这个话题有许多的文章，而且很多方法论可以被发展以帮助管理风险性。

总的来说，这些方法论趋向于采用瀑布模型的设计观点。通常的过程是确定危险性和识别风险因素；决定一种策略并相应地进行处理（避免、限制、冗余……）；追踪危险性直到硬件和软件组件，而将它们标识为关键状态；识别同样处于关键状态的操作过程，研究各种应用心理学和操作研究问题；最后，决定一个测试计划，然后开始测试工作。这种测试所产生的成果不光是一个你可以充满自信地实际投入运行的系统，还是一个可以正常运行安全用例（safety case）的系统。

如果某些你应该关心的事情出现问题，那么安全用例将提供证据；通常它包括危险分析，链接到组件可靠性和人为因素问题的文件，以及测试结果（在组件和系统级上均进行测试），这些结果可以显示达到了所要求的错误率。

理想的系统设计要完全避免危险。一个很好的例证来自于发动机的回动电路，参见图

22-3。在左边的最初设计中，一个双极双掷开关将转变当前通过发动机的电流的流向。然而，这有一个潜在的问题：如果仅仅是两个开关中的一个移动了，那么电池将短路，其结果将是引起火灾。解决办法是交换发动机和电池的位置，正如右图中所描绘的那样。在这种情况下，开关问题将仅仅使发动机短路，而不再是电池了。

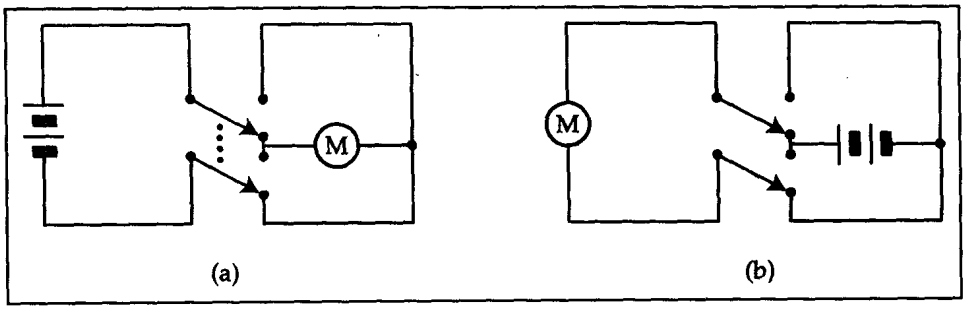


图 22-3　发动机回动电路中的危险消除

在安全工程当中，危险消除也是很有用处的。回忆一下在 9.3.1 节中对于 SWIFT 的早期设计的例子：用来在银行间验证交易的密钥在银行之间直接交换。通过这种方法，SWIFT 人员和系统无法伪造有效的交易，并且被给予更少的信任。通常，最小化可信计算库是在最大程度上对于危险的消除。

一旦所有可能出现的危险都被消除，下一步就是来识别那些可能导致事故的故障原因。通常自顶向下识别出现错误的事物的方法是，做一个故障树分析：这棵树的根代表不希望出现的行为，其后继节点是可能原因。这种做法对于安全工程来说相当直观和明显；图 22-4 中显示了一个故障树（或者叫做威胁树，因为在安全工程中通常称为威胁树）的例子，该例子是描述银行自动柜员机的欺骗行为的。威胁树在美国国防部（U.S. Department of Defense）中是标准的应用。

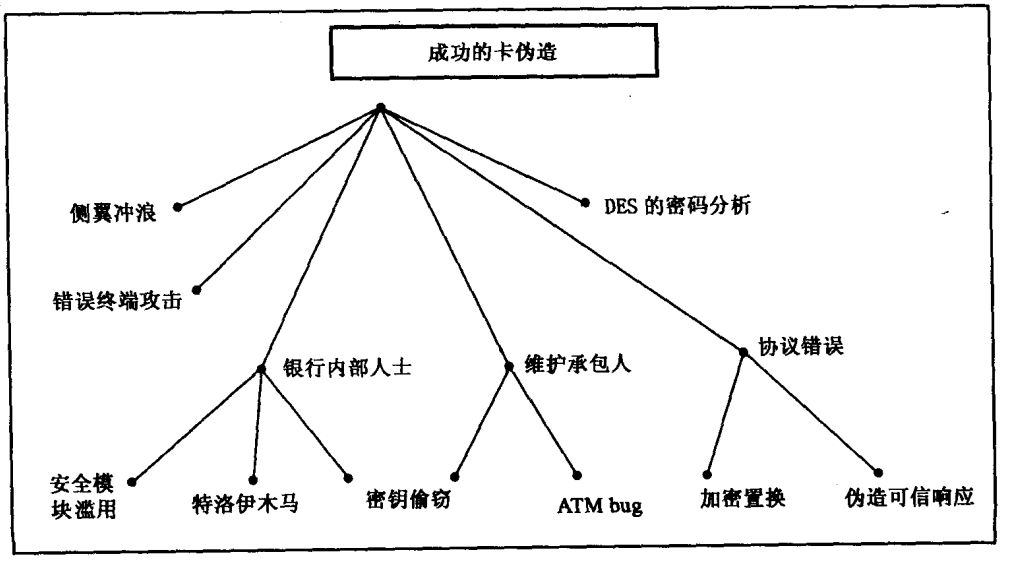


图 22-4　威胁树

下面是关于威胁树如何工作的讨论。你从每一个不希望的结果出发，然后通过记录下每一个可能的原因进行反向工作。你可以增加每一个前提条件，然后再进行递归处理。接着，处理树的叶子节点，你应该看到各种由技术攻击、操作失误、物理破坏等等可能破坏系统安全的原因组成的产物。注意到，这并不等于它是一本关于系统的攻击手册，而且它是高度分门别类进行安排的。但是，它必须存在。如果系统评估者或者是授权人可以找到任何严重的攻击方式，它们就可以导致产品出现故障。

回到安全关键型的世界中来，另一种处理危险分析的方法是故障模式和效果分析（failure modes and effects analysis，FMEA），由 NASA 率先提出。这是一种自底向上的方法，而不是自顶向下的方法。它包括跟踪对于一个特定故障所产生的一系列结果，这些故障是指由每个系统组件引起，而所跟踪的结果一直要包含从发生故障到对任务产生影响的全过程中。这种方法通常用于安全工程中，它可以让我们清晰地看见任何一个保护机制中所出现的故障的整个流程。

对于故障模式的真正彻底分析应该同时包含自底向上和自顶向下两种方法，而且还存在各种不同的方法来管理产生的大量数据。例如，你可以创建一个危险矩阵来对抗安全机制；而且如果安全策略是，每个严重危险都必须通过至少两个相互独立的机制来加以限制的话，那么你就可以检查一下，是否有两个实体在一个相关列中存在了。通过这种方法，你可以通过图表的方式证明，在一个危险出现时，至少有两个故障必须出现从而导致该事故发生。这个方法可以不加修改地应用到安全工程当中，正如我们下面要解释的。

安全关键型系统中提供了许多技术来处理故障和错误率。组件失效率可以通过统计学的方法进行测量；软件中的 bug 可以通过各种不同的技术跟踪，这些技术我会在下一章中讨论；而且还存在许多关于不同活动程度的操作错误的概率。对于电报机的概述就是，错误率来自于工作的复杂性和相似性、很大的压力和许多自认为成功的骄傲情绪。在那些任务本身十分简单的场合，情况通常就是这样的，自满情绪往往会很严重，而错误率也许是在100,000个操作中仅仅有一个。然而，当这项工作被第一次放到一个十分混乱的环境中执行的时候（在这种环境中需要逻辑思考，并且操作员往往在压力之下工作），就会出现相对于成功完成任务的失败可能性。系统的设计者们（例如核反应堆系统），十分注意（至少从 Three Mile Island 以来）当危险信号首次出现时，那就是严重错误发生的时候。相类似的，在安全系统中，诸如让高级管理者创建主加密密钥，这种情况虽然不经常发生，但却是极为重要的工作。在这种情况中，一些超乎寻常的错误往往就会出现。

一个著名的例子就是，当一家银行希望创建一组三个主密钥来将提款机网络链接到 VISA 时，需要一个终端来驱动安全模块 [20]。一位承包人故作热情地借给银行一台笔记本个人电脑，以及相关的用于仿真所要求类型终端的软件。通过这些设备，高级管理者们按期创建了一个所需要的密钥，并将它们发送到 VISA。没有人意识到，大多数的个人电脑终端仿真软件包能够将所有经过其的事务登记到日志当中，而这正好是承包人所做的事情。随着这些密钥被创建，他完全获取了区域密钥，后来使用这些密钥解密了银行的主 PIN 密钥。

当进行安全需求工程时，对于那些将执行每一个关键性任务的员工的技术水平要特别关注，还要对错误的可能性进行评估。这里要特别小心：一名飞机设计者能够依靠具有商业领航员执照的人的良好的可预见技能；而且造船商也知道一名海军海员的力量和弱点。对此，安全工程师通常就没有这么幸运了。许多安全故障使我们想起了由美国加利福尼亚州中部的

约塞米蒂国家公园（Yosemite National Park）的护林人员所做的关于如何防止熊来获得露营者食物的设备的有关评论：这是一项无法解决的工程问题，因为聪明伶俐的熊要比那些不说话的露营者更加有头脑。

这里还有关于易测性的问题。对于冗余系统的一个通常问题是错误伪装（fault masking）：如果输入是由三个处理器投票的多数选择所决定的话，并且当一个处理器出现故障时，系统也将会工作得很好，但是该系统的安全边缘已经被破坏。一些飞机坠毁的事故来自飞机导航系统的故障或者飞行控制系统的性能紊乱；虽然飞行员们注意力很集中，但是他们的显示系统不再可靠，在压力之下，他们的反应也就只能依靠这些显示设备，而不可能去检查其他设备是否正常工作了。更加严重的故障将会是灾难性的。一个安全方面的例子是9.4.2节中提到的ATM问题。在这里，银行对所有客户发行相同的PIN。在这个例子中被应用到PIN中的故障由于对防范措施的操作而被掩盖，保证了即使是银行的审计和安全职员也只能获得他们自己个人账户的PIN信封。很明显，需要采取一些办法来解决如何在故障的立即影响被掩盖的情况下，也可以让故障保持可见状态和可测试状态。

从安全关键型系统中我们得出的最后的教训是，虽然存在安全需求说明书和安全测试标准来作为律师或者调整人员所负责的安全用例中的一部分，但是更好的做法是在安全用例中集成进一般性需求和测试文档。如果安全用例是一些独立的文档集合，那么在获得正式批准后，很容易就可以迫使其退出，因此不能够正确地继续维持下去。在另一个方面，如果它是产品管理的一个集成部分，那么不但可以随着产品升级，而且还很有可能被其他领域的专家所留意，这些专家也许会设计出可能与之具有交互性的特征。

作为一般性原则，安全必须被内置进开发的系统当中，而不仅仅是式样翻新而已，这一点对于保密也同样适用。主要的不同之处在于故障模型上面。远胜于随机故障所带来的影响，我们还要对付那些使系统中某些组件在很短时间内就可以出现故障，并且破坏力极大的怀有敌意的对手。实际上，我们的工作就是进行计算机编程，使其给出在最有可能出现问题时那些巧妙的、怀有恶意的错误的解决办法。这个被称做“对撒旦的计算机编程”，已经同更常见的墨菲计算机区分开来[48]。这让我们看到了安全工程之所以十分困难的一个原因：撒旦的计算机很难被测试[682]。

22.4 安全需求工程

在第7章中，我将安全策略模型定义为对于一个系统或者系统的某一类别所必须具有的保护特性的简明陈述。这是由威胁模型驱动的，该模型在第3章中引入，它罗列了系统应该能够对付的攻击和故障。安全策略模型被进一步地细分成安全目标，安全目标是对于一个特定实现所提供的保护机制以及如何与控制目标相关联的详细描述。安全目标形成了测试和评估一个产品的基础。策略模型和目标合起来可以被称作安全策略，而制定安全策略以及获得系统拥有者支持的过程叫做需求工程。

需求工程是管理安全系统开发中最关键的工作，同时也是最为困难的工作。这就好比是在那些“橡胶撞击路面”的场合。这里正好是最困难的技术问题、最敏锐的官僚政治权力的斗争以及对于避免责任而最应该付出努力这三者的交汇点。这里可用的方法论始终比起适用于其他系统工程中的方法论要滞后一些。

在我看来，最关键的一点是产生安全策略的过程，而安全目标与产生代码的过程并没有

什么本质上的区别。依赖于应用，你可以使用自顶向下、瀑布方法、有限的反复方法，例如螺旋模型，或者持续迭代过程的演化模型等。在每种情况下，我们需要创建方法来管理风险，还要让风险评估驱动策略发展和演变。

一旦系统被部署后，风险管理必须继续存在。一种臭名昭著的做法就是声称一个新发明的事物是如何的有用，而且对于该事物的攻击可以被事先预料到。电话公司花费了20世纪70年代整个10年的时间来找出可以阻止通过盗用电话线路免费打电话的行为。正如它所证明的，真正的问题在于那些滥用系统打电话，而使警方无法跟踪的骗子们。一些人担心这些骗子攻击银行智能卡系统，从而在早期的电子钱包中引入了大量的后端保护机制。但是，取而代之的是，攻击出现在付费电视中的智能卡应用上面。其他人担心在网络事务中所使用的信用卡号码的安全性，其实真正的对于在线商务的威胁并不是黑客，而是偿还和争论。正如他们所说的，“街道找到了自己对于所有事物的使用价值”。其要点就是，不要期望在第一次尝试时就得到完全正确的保护需求。在许多情况中，策略和机制在系统第一次创建时制定，然后随着环境（和产品本身）的发展而推翻重来，但是保护却不一样。必须存在一种机制来监控以及作用于不断变化着的保护需求。

在这节中，不像前面各节，我将首先描述关于发展保护需求的话题，因为它更加具有一般性，并且也容易运用。

22.4.1 管理需求的发展

在大部分时间里，安全需求必须是出于以下四种原因中的一种。第一，我们也许需要修复一个bug。第二，我们可能希望改善系统，因为随着我们对于某种类型攻击的经验的增多，希望调节一些系统控制的机制。第三，我们可能想处理不断发展变化中的环境因素；例如，如果在线订单系统原先只被限制在少数的主要供应商上面，现在需要将其扩展到公司的所有供应商，那么控制机制很可能就需要重新考虑。最后，在机构内部还可能会出现变化，公司间不断合并、管理层收购、商业活动重新变革，无论你怎样称呼它们，都需要在安全需求上作出相应变化。

当然，这些原因中的任何一个都将会引起根本上的转变，以至于你必须考虑重新开发而不仅仅是对系统进行简单发展。如何将两者区分开来是不可避免的事情，但是正如我将要解释的，许多进化性的想法被应用在一次性实现的项目当中。

22.4.1.1 bug修复

大多数的安全增强方案被划分成两类，即bug修复或者产品调整。幸运的是，它们通常都容易实现，因为正确的结构已经摆在那里不需要调整了。

如果你销售的软件完全是保密关键型的，几乎所有能够和外界通信的东西都具有潜在危险，那么总有一天，你会得到关于该软件存在漏洞或者被攻击的报告。在早些时候，产品的卖主对于一个产品新版本的反应可能是先等待几个月，除了提出警告或者是拒绝响应外什么事情都不做。现在，大众的期望值比以前要高了许多。对于占有市场份额很大的产品，你可以期待新闻公开；即使是那些应用于特殊目的的产品，也会有被新闻覆盖的危险。简而言之，你最好是对此有一个计划。这将包括四个组件：监控、修复、分发和再保证。

首先，要保证尽可能快地得到关于系统缺陷的信息，而且最好是在新闻界（或者是某些不法分子）知道此事之前。听取用户意见是十分重要的；要为他们提供有效的方式来报告产

品中存在的 bug。还考虑对提供 bug 者使用激励机制，例如可以是产品的下一次免费升级、奖券甚至是现金。这种做法可以使一些人对于监控这些报告或者是阅读相关邮件列表（例如 bugtraq）更加负责任 [144]。

第二，要能够对所提出的 bug 进行适当的响应。在诸如银行这类有着关键时效（time-critical）处理需求的机构当中，一位产品项目组的成员收到别人通过名片上所写的家中的电话，报告系统在凌晨 3 点出现问题，并需要立即修复，这种事情的出现是很平常的。对于一个小型软件公司来说，这可能有些过分，但是你必须知道，那些打住宅电话的人对于被呼叫方的技术可能有迫切需求；注意对于每一个关键技术必须要保证不仅仅只有一个人知道；而且还要提供技术支持过程。例如，紧急情况 bug 修复必须尽可能地经过完全的测试过程。而且，文件也必须被升级；这对于进化安全性改进来说极为关键，但是往往会被忽视。当由于 bug 修复改变需求时，还需要对文件进行修改（也许还有威胁模型，甚至是顶级风险管理文档）。

第三，可以保证尽快给用户分发补丁或者其他修复方式。这一点必须事先计划好。有赖于产品的不同，其细节也可能会不一样：如果，你只有仅仅几个用户，他们在数据中心的服务器上运行你提供的程序代码，而且时时刻刻都有人在看守，那么工作将变得很容易。但是，如果这项工作涉及成千上万份用户软件拷贝都需要打补丁的话，就需要十分小心了。这可能看上去很简单，只需要让用户每天访问你的网站一次，来检查是否需要升级就可以了。但是，要想安全地做到这一点，你必须处理好许多细节性的问题。服务器可以处理因此而激增的流量吗？你已经提供给用户足够的法律通知来使得他们的软件可以经自己修改吗？那些反对者，例如不满的前公司员工，可以通过控制这种机制，从而破坏整个用户群吗？

最后，要有对付新闻的一个计划。你所需要做的最后一件事情就是，在你疯狂地修复 bug 时，对于众多记者的电话，你必须在电话接线台操作员处予以阻拦，不要让这些电话影响你的工作。在你的 Word 处理器中存储一些针对不同错误程度向新闻界发布的通知模板，以便于你只需要从中选择一个模板，然后填入细节就可以。这个通知要在第一个新闻界人士（也许是第二个）打来电话之前公布出去。

22.4.1.2 控制调整和整体管理

诸如银行等开发自己的计费系统和其他内部控制机制的机构的主要处理过程就是根据经验来调整它们的系统。一家拥有 25 000 名员工的银行也许每天由于小型偷窃或者盗用的原因而解雇一名员工，而且，传统上都是由内部审计部门来提供损失报告，并且建议通过系统变更来减少大多数欺诈阴谋的发生。我在 9.2.3 节中给出了一些例子。

对于安全工程师而言，最重要的是具有内部控制方面的相关知识。关于这方面的书籍比较缺乏：审计在很大程度上是在工作中学到的，而不是通过课程和会计学标准文档就可以知道是怎么一回事。这里有一份由 Janet Colbert 和 Paul Bowen 所写的关于内部审计标准的调查报告 [193]；最具影响力的是来自赞助组织委员会（Committee of Sponsoring Organizations，COSO）的风险管理框架（Risk Management Framework），该委员会是一个美国会计和审计团体的组织 [196]。如果系统被用在美国公共部门或者那些被美国产权投资市场（U.S. equity market）所摘录的公司中，它就是对系统进行判定的准绳。

COSO 模型的目标不光在于内部控制方面，还包括财政报告的可靠性和对于法律法规的依从性。它的基本过程是一个进化循环：在一个给定的环境当中，你评估风险，设计控制，

监控它们的性能，然后重新开始这个循环。COSO 强调整体文化中软件方面多于硬件系统设计的问题，而且可以被看做是一个向导，用来管理和证明处理的过程。通过这个过程，系统逐渐向前发展。然而，其核心包括内部控制过程。通过这些过程，高级管理层可以检验他们的控制策略是否被实现和取得既定目标，以及如果没有达到效果的情况下如何修改它们。

对于安全工程师来讲还需要做出努力的是要多了解一些特定信息系统的审计功能。IS 审计员不应该只对安全具有表面责任，否则将会发生利益冲突：对于那些由审计员本人设计的系统或者是该审计员需要对系统操作负责的场合，是不应该让该审计员参与评估的。更为适合的做法是让该审计员监控事件的处理过程，观察那些不标准或者呈现可疑状态的事物，并且提出改进意见。对于安全工程而言，大部分的技术资料都是一样的；如果到目前为止，你已经阅读并且理解本书所讲的内容，你将有 50% 的希望在认证信息系统审计员（Certified Information Systems Auditor，CISA）考试中取得好成绩。关于这个考试的细节，参见 [408]。信息系统审计和控制协会（Information Systems Audit and Control Association）是管理 CISA 的机构，它提炼了 COSO 的思想，被称作对于信息和相关技术的控制目标（Control OBjectives for Information and related Technology，COBIT），这更加适合 IT 的需要，更加国际化，比起 COSO 更加具有可访问性（COSO 可以在 [407] 上下载）。COBIT 所覆盖的领域不仅是工程需求，而且职员管理、变化控制和项目管理也成为内部审计人员需要参考的内容（在职的安全工程师也需要熟悉这份资料）。

这些通常的标准必然是相当模糊的。它们提供给工程师一个上下文环境和一个顶级 checklist，但是很少提供关于某些特定措施的明确建议。例如，COBIT 5.19 中说："关于恶意软件，诸如计算机病毒或者特洛伊木马，管理应该建立一个具有足够可防范、检测和纠正的控制措施的框架"。更加具体的标准需要被制定出来，从而将这些笼统的原理应用于具体的特定应用当中。例如，当我 80 年代在一家银行安全部门工作时，我就依靠银行国际结算指南进行工作 [71]。在这些标准存在的地方，通常就是安全进化活动最终的支点。

一个不错的想法是向客户的内部审计部门提供高带宽的通道来进行通信。但是，如果完全依赖于反馈就不是一个好的思路了。通常，那些知道如何破坏系统的人正是那些在实际中使用系统的人。问他们就可以了。

22.4.1.3 发展环境和公共悲剧

我描述了许多系统，这些系统在环境变化后就崩溃了，而且在这些系统中，对于保护机制的适当修改的做法很少被使用、甚至是避免或者干脆被遗忘了。当使用零售点销售终端后，和 ATM 技术结合工作得很好的 Card-and-PIN 技术就容易受到错误终端攻击了；那些可以在零售点应用中很好地管理信用卡号码和 PIN 的智能卡在防付费电视盗版方面就显得力不从心了；而且在那些主要威胁来自内部而不是外部的场合中，甚至连验证协议之类的基本机制也不得不重新设计。军用环境在战争期间发展得特别快，因为攻击和防御是共同发展的；R.V.Jones 将二战期间盟军在电子战中取得的胜利大部分归功于这样的事实，即德国使用严格的自顶向下的开发方法论，而这种方法产生了那些被很好地进行工程设计的设备，但是 6 个月的开发时间实在太长了 [424]。

应用中的变化并不是惟一的问题。操作系统的升级可能向底层平台中引入一套全新的 bug。正如商务向电子商务变化这类规模上的变化能够改变收支方程，这是由于这样一个事

实，即许多系统用户可能是处于国外那些不够健全的计算机犯罪法（或者根本就没有这类法律）的环境中。还有，那些已经被专家们熟知的可能攻击类型，由于没有在现实中真正出现而被忽略了，但这些类型的攻击突然开始发作。对此的一个很好的例子就是分布式拒绝服务攻击。

当你拥有系统时，事情仅仅是困难而已。你通过确保机构中的一些人对维护安全等级负有责任的方法来管理风险；这将包括由内部审计官僚机构所规定的每年一次的评论，或者是更改控制的一个方面。维护有组织的内存是十分困难的，这要感谢 IT 和安全员工们很高的营业额，这一点我在 22.2.3.4 小节中已经讨论过。

系统已经足够强壮了，但是在许多真正难于处理的问题出现的场合中，没有人真正完全拥有这个系统。对于建立标准的责任，例如 ATM 如何检查 PIN，是十分松散的。在这种情况中，那些已经制定出大多数标准的公司（IBM）将失去其在业界的领导地位；它的继任者微软对这个市场则不感兴趣。密码设备由许多专门的公司销售。虽然 VISA 惯于认证设备，但是大约在 1990 年时也停止该项工作，而且万事达信用卡从没有涉及过这种业务，所以没有人或者公司可以对此进行管理。每个游戏者（设备生产者或者是银行）都存在这样的动机，就是将安全分界线再向远处推进一些，当最终某些事情出现问题时，情况就不是这样了，这会发生在其他人身上。

这个问题经济学家很熟悉，他们管这个叫做公共悲剧 [507]。如果 100 个农民被允许在一个乡村公共地放羊，而乡村中草的数量有限，那么当另一只羊加入进来时，羊的主人将获得全部的利益，而其他 99 个农民所遭受的损失仅仅是很少数量的草的下降。因此，他们不会出来反对，但是人们都宁愿自己增加另一只羊来尽可能多地弥补已损失资源。这样做的结果就是羊的数量越来越多。在农业中，这个问题是通过公社机制来解决的，例如让教区议会建立一个放牧控制委员会。在 10 世纪的 Saxon 村庄中，牧牛者已经被很好地组织起来进行这项工作了；一个摆在我们面前的挑战就是设计一些技术和有组织控制的混合系统，它可以使我们仅仅在一个更大规模的因特网上得出一个可比较的结果。

22.4.1.4 组织变化

组织问题并不是导致安全故障的惟一原因。其实导致安全故障的原因还包括组织内存的丢失和对于监控变化着的威胁环境的总体机制的缺乏。这些常常是导致安全故障的主要原因。

在 20 世纪 90 年代早期，管理模式用于业务处理重建，这通常意味着用业务计算机系统中的变化来强制改变人们工作方式的变化。在那些设计拙劣的系统与愤怒的员工的冲突中将导致灾难的发生，这就是最好的例证。

也许，最著名的例子就是伦敦救护车服务。该服务拥有一个手动系统，通过此系统，产生的紧急情况呼叫被写到表格上并通过传送带送到三个控制者那里，他们分配车辆并将表格送到无线电发报机。业界相互之间的关系很糟糕，所以要想降低成本是存在压力的；管理者可以通过自动化的方法来解决所有这些问题。许多事情出现问题，而且随着系统的逐步采用，越来越明显地表明不可能处理那些已经建立起来的工作实践带来的问题，例如全体员工都上错了救护车（随着高级职员都使用高级车辆，其他员工也就拥有自己钟爱的车辆）。新系统在 1992 年 10 月 26 日被强制投入运行，但管理者们却不希望知道，通过重新组织空间，控制者们和发报员们必须使用终端而不再是纸张了。

结果是系统彻底崩溃。许多实际的反馈循环被建立起来，并导致系统日益增多地失去车辆的线索。异常消息不断增多，在显示屏上滚动出现，由于速度太快没有看清，所以导致信息丢失；事故不断出现是由于车辆分配者不停地查找车辆；随着系统响应时间的增长，来自于患者的呼叫变得增多起来（平均每位呼叫者的紧急情况呼叫时长要 10 分钟）；随着堵塞的增加，救护车员工们感到灰心丧气，在他们新的数据终端面前按错按钮，从而得不到想要的结果，试图通过语音通道来进行呼叫，这又会增加阻塞；随着越来越多的工作人员恢复到原先他们所理解的工作方法上，他们甚至更多更频繁地出现错误；许多车辆被分派响应同一个急救呼叫，而对于有些呼叫却没有派出一辆车；最终，整个服务彻底失败。据估计，大约有 20 个人由于没有及时得到护理援救而直接导致死亡。到了 26 号下午，这已经成为主要的新闻报道内容了；政府介入进来了，而且到了第二天，系统被转变到半人工操作状态。

这仅仅是许多灾难中的一个例子，但是它对工程师而言却很有价值，因为它被由此产生的公众质询相当好地予以了证明 [723]。从我专业的角度来看，那些类似试图强行改变共同文化的做法，在这里就是代替计算机系统，严重地破坏了民心以至于诚实正直成为了所关注的话题（我的许多顾问工作都与全体重组的压力甚至是国家政治危机这类环境有关）。

在极端的例子中，由原始的全体重组所导致的环境中的一步改变更像一次性项目而不像是进化性的变化。这通常成为回退时的有用的基础，例如对于外部威胁的理解；但是内部威胁环境也许变得从根本上产生差异。这在银行业中尤为明显。15 年前，银行部门由类似伯父式的人管理，而员工都是有名望的中年妇女，她们希望可以将毕生都投入到银行工作当中。现在，管理者被产品销售专家取代，而出纳员则是些年轻人，其工资收入近乎最低水平，大约每过一年就会跳槽到其他公司工作。这简直就不是原先所想的同一种业务。

22.4.2 管理项目需求

对于一个一次性项目来说，这给我们带来了许多更加困难的问题，即如何去做安全需求工程。最普通的例子也许是从无到有地创建一个电子商务应用，不论是新启动的业务还是为了一项已经建立起来的需要创建新的分发渠道的业务。

从无到有地创建事物是一件很容易出现事故的业务，而且在很多这样的例子中，那些大型软件项目都以失败告终。这些问题和是否灾难就是安全问题或者软件完全不能工作这么简单的问题在很大程度上是一样的；所以做安全工作的人也可以从一般性的软件工程的文章中学到许多东西。

对于大型软件项目灾难的最著名的学习资料是由 Bill Curtis、Herb Krasner 和 Neil Iscoe 所写的 [212]。他们发现，对于需求的错误理解是最应该被责备的：对于应用领域知识的缺乏导致对于需求的不确定和冲突，这将反过来导致双方沟通的失败。他们建议，解决办法是找到这样一位“特别的设计人员”，他对问题有着彻底的理解，并且还假设将对系统完全负责。

千年虫提供了另一个有用的学习实例。对此，许多关于软件工程的作者们还不得不进行消化和融汇贯通。如果一个人承认许多大型的商业和政府系统实际上需要广泛的修复工作，而且惯例做法是很大一部分大型开发项目是推迟交付或者是根本没有交付使用的，那么在 1999 年末所发生的普遍混乱的预测就是不可避免的事情了。但是，预测的混乱情况并没有发生。当然，对于小型和中等规模的公司所使用的系统所面临的风险，往往是言过其实了 [37]；然而，一些大型公司的系统，它们所进行的操作对经济起到极其重要的作用，这些公

司包括银行和一些从事公共事业的公司等，这些系统才真正需要修复。但是，尽管情况确实如此，但至今仍没有得到有关大型机构破产的报告。这就支持了 Curtis、Krasner 和 Iscoe 的理论观点。对于 Y2K bug 的修复需求现在已经完全清楚："我希望系统可以继续工作，就像现在一样，虽然到了 2000 年甚至更远的将来。"

作为一名需求工程师，你需要获得全面的关于应用领域的知识，还有关于可能对系统进行攻击的人的信息，以及他们所使用的工具。如果可以找到应用领域的专家，并且他很出色，那么情况就更好了。当你会见他们的时候，试图区分出哪些工作是出于某种目的被做的，而另外一些则是那些"仅仅是如何在现有基础上完成"的事情。要经常性地对事情为何被做的原因进行探测，并且对于事后合理化也要十分敏感。尤其注意那些将要变化的事物。例如，如果处理用户投诉时依赖用户具体情况而给予不同的待遇，而你的工作是将该业务在线运行，那么你就必须询问专家有什么替代的控制方法可以使用，这些方法可被应用于难以看出用户年龄、性别和社会地位的场合中（这应该在 20 世纪 60 年代民权运动发生时就被完成，但是后期再做总比不做要强）。

当解决一个新应用时，调查一下该应用的历史。我贯穿全书都在试图做这件事，而且说明了问题重复的方法。为了找到在 21 世纪，电子银行将变成什么样子的答案，一个不错的想法就是先要了解它在 19 世纪时会是什么样子；人类本性并没有多大改变。利用历史相似性将更加容易地让你用建议来说服客户公司的董事会。

你将很可能发现，一个新项目的安全需求说明书需要反复修改。所以，很可能需要使用螺旋模型而不是瀑布模型。在完成第一遍处理后，你将描述新应用以及它和任何现存的、可以找回丢失的历史信息的应用之间有哪些不同之处，制定出一个可以预防那些你能够察觉出的风险的模型，并且起草一个安全策略（我将在下一节中详细描述关于风险分析和管理的问题）。在经过第二遍处理后，你也许将从客户的中级管理层和内部审计人员中获得意见，而同时可以很快地写出文献（从内部审计指南到像本书这样的书籍），这是为了得到有用的 checklist 表项和你可以重复使用的想法和思路。这项工作的成果将是一个被修正的、更加量化的风险模型，一个安全策略和一个安全目标，来大致描绘策略在现实生活中如何被实现的问题。它还将提出系统如何根据这些标准被评估。在第三遍处理中，文件将循环传阅给更加广泛的一组人，他们包括客户高级管理层、外部审计人员、保险业者也许还有外部评估者。

22.4.3 并行处理

通常，你身边根本就找不到应用领域的专家，因为此时系统只是被首次创建，或者你正在创建的系统将成为一些已经存在的私有系统的竞争者，而这些私有系统的拥有者根本不想把他们曾经出现的损失，即损失历史，所换来的经验与你分享。一个可能要被问到的有趣的问题是，大家如何才可以仅凭想像所有可能出现问题的事情就制定出一份需求说明书来呢？业界中常规的做法是雇佣一家顾问公司来制定这个安全目标；但是，我在 10.3.3 节中所描述的经历建议我们还是应该并行使用几位专家，这样做会比较好。具有加密、访问控制、内部审计等等经历的人可能会从不同的角度看待一个问题。这与软件测试之间具有有趣的可比性。在软件测试领域，并行测试比起串行测试来说具有更高的效率：每位测试者注重测试的不同方面，这样同其他方法比起来将会发现更多不易发觉的错误（我将在下一章中引入一种

更加定量的模型来说明这一点)。

这促使我在1999年时进行了一项试验，来看看是否通过许多不同人提出的草案，可以很快地组合出一份高质量的需求说明书。其思路是大多数可能发生的攻击都可以被他们当中的至少一个人考虑到。因此，在大学考试试题当中，我曾经提过这样一个问题，那就是如果公司计划竞标公开抽奖的许可证时，那么究竟什么才算是适当的安全策略呢。

结论在[36]中被描述。典型的攻击是攻击者们也许在和内部人士合谋，一旦草案制定结果已知的情况下，他们将试图做一个赌注，无论是修改赌注记录还是伪造入场卷；或者做一个不用付赌金的赌注；又或者是操纵自动贩卖机，该机器只会对小额奖金予以给付，一旦顾客获得了大奖，它们就会消失。随后出现的安全策略遵循这些可能性应该在一台服务器上在线注册，对该服务器的保护应优先于对所草拟的方案的保护，它们二者都要防篡改和抽取足够信息来伪造许可证；还应该对真正的供应商进行信用限制；以及应该提供某些方法来识别伪造供应商。

来自于学生们的有价值的和新颖的投稿可以分成许多层次，包括策略目标声明、对于特定攻击的讨论和关于特定保护机制尺度的争论等等。在策略一级，有许多精巧的发现，从而维护公众信心和应付那些来自于私营公司高层管理者的威胁。在技术细节一级，一个学生讨论了来自于冗余机制的威胁，而另一个学生则讨论了关于安全时间机制的攻击，而且发现，在奖券终端中使用无线电时间信号将会遭到人为干扰（这一点在某个实际奖券系统中确实被证明是一个存在的漏洞)。

学生们还提出了许多设计者通常都会忽视的常规性的检查列表（checklist）条目，例如“许可证必须与一种特定的图案相联合。”这看上去似乎是显而易见的，但是那些使用购买日期、许可证序列号以及服务器所提供的随机口令作为MAC计算输入的协议设计却表现为似是而非的肤浅的检查。有经验的设计人员是非常欣赏这种检查列表的价值的。

从这个案例的学习中，我们可以得出的教训就是，需求工程就像软件测试一样，可以接收一定程度的有益的并行机制。如果你的目标系统有一些新奇的东西，那么与其雇佣一位顾问来冥思苦想20天，还不如考虑使用15个具有不同背景的人每人思考一天呢，然后叫一位顾问利用一周的时间将这些想法集中起来，提炼出一份一致的文档。

22.5　风险管理

无论威胁模型和安全策略的自身发展或者是在一次性项目中的发展，它们的核心部分都要包括对于优先权的商务考虑，我们在保护机制上花费了多少，并且用来防御了些什么。这就是风险管理，而且它应该被用于管理那些非IT风险问题的广泛框架之中。

许多公司出售关于风险管理的方法。其中的一些是采用自助个人电脑软件的形式，而其他的则是采用顾问服务包的形式。你使用哪一种则要取决于客户的策略；例如，如果你正在向英国政府出售产品，那么你很有可能必须使用名为CRAMM的系统。这类系统的基本目的就是将各种安全花费区分先后顺序，同时又对此向高级管理层提供一个财政方案。

最普通的技术就是对于每个假定的损失情况来计算年度损失期望（annual loss expectancy，ALE)。它表示一年当中能预计到的众多事故所引起的损失值。对于银行计算机系统的典型ALE分析可能由几百个部分组成，包括图22-5中所罗列的条目。注意可以得到普通损失类型（例如“出纳员拿走现金”）的精确数据，而对于那些不算是普通的高风险损失，诸如

大宗转账欺骗，其影响范围在很大程度上就只能靠猜测了。

损失类型	数量	发生频率	年损失期望
SWIFI 欺骗	$50 000 000	.005	$250 000
ATM 欺骗（大型）	$250 000	.2	$100 000
ATM 欺骗（小型）	$20 000	.5	$10 000
出纳员拿走现金	$3 240	200	$648 000

图 22-5 每年损失期望的例子

在美国政府部门应用中，ALE 被 NIST 标准化为一项技术从而得到使用［602］。但是在现实生活中，产生这样一张表格的过程经常是过于反复的猜测性工作。顾问列出了所能够想到的所有威胁，并加上想像出来的可能导致因素，计算出 ALE，把它们加到一起，然后得到一个荒谬的结论。例如银行的 ALE 大于其所有的非营利性收入。那个顾问然后将总额硬拉到某一数额上，这一数额正是该顾问所考虑到的董事会可以忍受的最大安全预算（或者这一数额是其客户，即主要内部审计人员，告诉她的最可能获得的最大预算值）。为了得到正确的结论，损失概率也被篡改（电子数据表真是一项伟大的发明）。如果这些听起来有些危言耸听，那么我深表遗憾，但是这些都是很有可能发生的事情。关键就是，ALE 是有些价值的，但是不应该做什么都依靠 ALE。

对于管理大型但是不常发生的风险时，保险通常会起到帮助作用。但是，保险业也并不是完全地具有科学性。很多年来，对于银行家合同的年保险值，这里包括计算机犯罪和员工背信弃义，只占保险额数目的 0.5%。这代表了编写策略的 Lloyds of London 公司的纯利润。然后，由于出现具有争议的呼声，这个数值长到了 1%。这类策略对于每个事故都要扣除掉保险人应该对被保险人支付的 50 000 ~ 10 000 000 美元的金额，所以他们从等式当中仅仅移走了很小一部分非常大的风险。如果由一位很有经验的保险评估员来检验计算机系统并建议一些安全增强措施的话，那么将带来真实的效益；但是一个典型的银行也许需要支付不少于 6 位数的金额来做这件事情。

许多公司之所以可以获得有关计算机犯罪的信息，其主要原因是由于勤奋，其实做其他许多事情也都是要归于勤奋的。所对付的风险表面看上去是可操作的，但实际上通常是具有法律、法规和 PR 风险在里面的。通常，它们是通过“绝大多数的做法”来进行处理的，就像我通常对因特网安全所做的比喻一样，就像是美洲草原上的上百万羚羊中的一只一样。这就是为什么计算机安全是一种由时尚驱动的业务的一个原因所在。在 80 年代中期，黑客们是主要关注的问题，所以那些销售回拨调制解调器的公司的业务十分火爆。从 80 年代后期开始，病毒开始流行，所以使那些销售反病毒软件的人富裕起来。最近，随着对电子商务的夸张描绘，防火墙逐渐变成了一个新兴产品。有许多威胁因素和产品被电视上和金融媒体中出现的公司的 CEO 们关注着。在这些杂乱因素的影响下，从事安全工作的专业人员一定要保持健康的怀疑心态，而且要为理解真正的威胁究竟是什么而不懈努力。

最后，知道计算机和通信安全是什么对于特定应用的决策来说是需要的。客户的 CEO 迟早必须做出一个选择，而你所能做的最好的一件事情就是在正面和反面意见中给出一个适当的和诚实的评价。

22.6 经济问题

摆在安全工程师面前的许多问题在经济学当中都是有其根源的。顾问们通常把设计失败的原因解释为“客户并不想得到安全的系统，但是仅仅想得到最大众化的安全水平，我也必须每周在产品中投入 10 000 美元的预算。”意识到这并不仅仅是管理上的愚蠢是很重要的。

我第一次讨论网络影响是在 19.6 节。具有更多使用者的网络对于每个使用者来说将更加具有价值，从而导致一个强有力的正面反馈以及通常是首先行动者受益最大。这就是“我们将在星期二发货，在第三版时让系统运行正常”这句话中的哲学思想的根源。虽然这些常常被愤世嫉俗者强加在微软身上，但是在网络经济应用的场合中，这通常是市场当中合理的经济行为。

网络经济对于安全管理处理有许多其他方面的影响。公司们通常更喜欢一个私有的、模糊的解决方案来使得客户被困在当中，而且也增加了其他竞争公司在试图创造兼容产品的过程中所遇到的问题，而一般不会再去采用标准的、经过良好分析和测试的解决方案。只要可能，这些公司就会使用专利算法（即使这些算法并不怎么出色）来作为一种对制造商强加许可条件的方式。这可以回忆一下在 20.2.5 节中所讨论的，DVD 内容搅乱系统是如何被用来作为一种需要兼容设备制造商同意一系列的版权保护机制的方法（而且该系统由于阻碍了 Linux 操作系统运行在下一代个人电脑中而导致应用失败）。网络拥有者和创建者将呼吁下一代应用的开发人员，即使是将有效的安全管理做得不切合实际，也要在用户上而不是开发者上投入大量的支持花费。安全工程师需要学习网络经济学课本，例如 Shapiro 和 Varian 的[696]，从而理解公司们为了保持垄断或者破坏垄断所采取的与保护机制相结合的各种做法。

还有本地经济问题。安全是关于权力，而设计是为那些为了可见利益而投资完成设计的人服务的。我在第 8 章中曾描述，由保险公司而不是保健供应商所设计的医药支付系统为何在保护患者隐私上最终失败，这是由于与保险公司希望得到客户的最大信息量产生了冲突而造成的。第 9 章中描述了许多国家中的银行是如何通过多年的努力来让客户容忍风险和欺骗所带来的花销的。第 21 章中解释了一些数字签名法律是如何将伪造签名的风险由依赖于该签名的个人转移到那些应用数字签名的人身上的。本章的 22.4.1.3 小节中解释了公共悲剧，在这里，许多游戏者都将他们的风险倒入公共池中，所以每个人都通过捷径获得巨大的利益，但同时在出现问题时只是和别人共享损失，所以损失将很小；结果就是标准会下降得很快。

对于公共悲剧一个特别的例子来自于最近分布式拒绝攻击的泛滥，该技术在 18.2.2.3 小节中有讨论。在这些攻击当中，故意破坏者们攻击许多个人电脑以及安装攻击软件来通过大量消息轰炸特定目标，这些消息则超过了该机器所能够处理的极限。成为这类攻击牺牲品的可能性很低，以至于大多数通常的使用者都可以忽略它，所以对于这类攻击的处理并不会干扰个人电脑的正常保护。然后，正如牧场被过度放牧一样，因特网也将越来越快地变得不安全起来。随着越来越多的人安装高带宽、总保持连机状态的因特网连接，不安全性将变得更加严重。Jean Camp 和 Catherine Wolfram 已经在因特网安全和环境污染之间建立了一个有趣的并行机制，参见 [156]。

来管理这种情况的最佳方法就是应该能够管理大部分都落在某方的风险。这在民事侵权行为的法律当中已经是一种普遍的原理了，但是有许多行业或者应用都通过这样或者那样的

方法来避开这一点。在分布式拒绝服务攻击的例子当中，那些随机受到攻击的人中很少有提出诉讼的，因为大多数家用PC使用者对于安全方面是没有任何防护能力可言的；而且，无论怎样，对于那些不幸的受到攻击的个人提出诉讼的可能性也是不大的。Hal Varian已经建议，那些被攻击用户的因特网服务提供商应该承担这种风险中的绝大部分[771]。这将引发出对于防火墙的需求，利用这些防火墙不但可以管理到来的数据流，还可以对于发出的数据流进行控制和管理。这种想法的背后是基于一种策略的，对于该策略，我已经在6.2.4节中进行过描述，即对于网站的拒绝服务攻击的响应，可以将网站复制成为具有更高性能、更加分布式的服务器。随着这种服务的使用权被租用，必要的经济学动机就可以通过一种或多或少透明的方式来实现（得到更详细的信息，参见[816]）。

在实际生活中，安全设计背后的驱动力通常与保护终端用户隐私以及减少他们被欺骗的风险性的无私期望没有任何的关系。这种动机更像是希望获取垄断权、对于本质上相同的服务却对不同用户收取不同的费用，以及减少风险性。通常，这种做法是相当合理的。但有时，情况并非如此；将出自ATM欺骗的风险转移到用户身上的英国银行安装了许多安全机制，以便于在出现争论时可以使用它们在法庭上争辩，它们已经认真不懈地采用了许多措施；它们最终在ATM安全方面比起美国银行来讲花费了更多，美国银行总是忍受这种可靠性，认为安全是风险管理当中的一个合理的事物[19]。

在理想情况下，对于创建不安全系统的不正当经济动机的清除将使许多问题非政治化。安全工程将成为合理风险管理而不是风险倾倒中的一项内容。但是，也不必因此而感到十分紧张。

22.7 小结

编写安全需求说明书通常是整个工程进行当中最为困难的一件事情。就像开发系统本身一样，它可以包含一个一次性项目，或者是采取有限的反复过程，又或者是连续进化的过程。进化最容易管理，虽然由于规模、环境和商务结构的变化，可能会有些复杂。从无到有地建立一个新系统最难，并且还容易产生错误，但是还是可以从其他地方得到一些技术和教训。

在缺乏任何好东西的情况中，我对项目管理者的建议是专心投入到创建具有某些重要的保护需求的应用程序当中，你必须尽最大努力来精确理解这些需求的含义，将它们加入到说明书中，然后使用那些你平常使用的方法论跟踪其实现、测试和部署的全过程。但是，假设在第一次时你不能够完全正确地做到这些事情。所以要确保可以通过一些规定的方法来获取出现错误和环境变化的反馈信息，从而将这些信息加入到增强和维护系统的过程当中。安全必须作为一个集成部分加入到管理系统生命周期的过程当中。

研究问题

本章中所讨论的问题相比于安全工程领域中的任何其他问题而言都是十分关键的，也是十分困难的。但具有讽刺意味的是，人们对它们的关注程度却很低，因为这些问题都处在几个学科的边缘上，例如软件工程学、应用心理学、经济学和管理学。与这些学科相衔接的每个接口都存在许多问题需要研究，但前提是你必须具有相关的背景知识。当创建一个足够健壮可面对恶意攻击的系统时，你也必须将它们创建成为面对通常的人类行为也十分健壮才

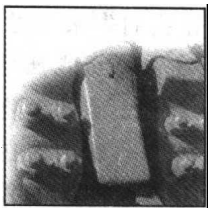
行。如果能够说出这两者之间的区别，那么就很有希望能够做出一些有用的东西。

参考资料

关于管理信息系统的开发过程的著作有很多，也很分散，并且还包括各种学科中的知识。其中有一些十分著名的书籍是每个人都应该阅读的，例如 Fred Brooks 的《Mythical Man-Month》[140] 和 Nancy Leveson 的《Safeware》[498]。关于软件工程的标准教科书，诸如 Roger Pressman [622] 和 Hans van Vliet [767] 所写的那些书，这些书包括了项目管理和需求工程的基本理论。关于软件生命周期的经济学在 Fred Brooks 和 Barry Boehm 的书中被讨论过[123]。关于管理软件进化的微软方案在 Steve McGuire 的书中 [521] 进行了描述。对于其他工程原理，并行方法也是很有用的一种。由 Henry Petroski 所写的一本书中讨论了桥梁建筑的历史、桥梁倒塌的原因和那些土木工程师们应该如何从倒塌中学到东西：一个已经被创立的设计范例被一次又一次地扩展，但往往会由于某些不可预见的原因突然失败 [612]。关于风险管理方法和工具的一项调查，请参见 Richard Baskerville [77] 或者 Donn Parker [602]；IFCI 中还有许多有趣的历史案例可以参见 [402]。计算机系统故障是另一个需要学习的主题；最好的资料源就是 comp.risks 新闻组，由 Peter Neumann 编写并出版的书正是该新闻组的精选内容集 [590]。

在商务学校的文献当中对于组织问题都有翔实的介绍，但是对于外部人士来说就比较迷茫了。对于此类文献的重要说明已经由 John Micklethwait 和 Adrion Wooldridge 提供，他们同时也提出了许多高度相关的话题，诸如最高管理层的矛盾，他们让底下的具体管理者通过解雇员工而将机构变得更加复杂，但同时又在鼓吹信任的美德 [550]。熟悉这些资料对于预见到客户最新的重组所带来的保护结果是十分有用的。最后，对于基本的经济学，我所知道的最好的一本书是由 Carl Shapiro 和 Hal Varian 所写的十分受欢迎的大纲 [696]，以及由 Hal Varian 所写的一本标准教科书 [770]。

第 23 章　系统评估与保证



如果它可以被证明是安全的，那么也有可能不是安全的。

——Lars Knudsen

我任何时候都在想，如果你暴露出一些弱点，那应该是一件好事情。

——美国司法部长 Janet Reno [642]

23.1　引言

我在本书中已经提供了安全工程学方面的许多素材，它们中有一些确实是很难做到的。但是，我还是把最艰难的主题留到了最后。这就是保证的问题，无论系统是否将投入使用都会涉及此问题；还有就是评估的问题，即你如何使别人对此信服。

基本说来，保证应归到关于系统本身是否具有能力的问题，从而激发人们将系统设计得确实具有足够完善的功能。但是，你又如何定义足够一词呢？以及如何定义系统？如何处理那些保护错误事物的人们，就因为他们的需求模型过时或者从根本上讲就是错误的？还有，你如何体谅人为错误呢？许多系统只有在具有经验的专家的操作下才可以正常运行，因为往往他们才具有很高的警惕性；而如果是由普通人员来使用的话，那么可能就不适合原设计目标的实现初衷，因为它对于错误的容忍度过于敏感。

但是，如果实现保证都很困难，那么评估将变得更加困难。它是关于你如何使老板、客户，而且在极端情况下还可能使陪审团所信服，使他们觉得该系统确实可以达到既定目标；该系统也确实可以正常地工作（或者系统可以在过去某些特定时间内正常地工作）。评估之所以既是必需的又是艰难的，其原因通常在于一位当事人管理保护的费用，而另一位当事人则管理故障风险的开销。这将会导致很明显的人与人之间的相互猜忌，而第三方评估方案，诸如通用准则（Common Criteria）已经被市场化为一种使评估更加透明的方法。

23.2　保证

对于保证的可行定义是“我们所估计的系统在某些特定方式下不会出现故障的可能性。”这种估计是基于许多因素的，例如开发系统的过程；开发者或者开发小组的身份；特定的技术评估，例如形式化方法的使用或者故意引入许多 bug 由测试小组进行测试，来看看它们当中有多少可以被系统捕获；还有就是经验，它最终取决于随着系统被测试、使用和维护，系统所具有模型的可靠性是如何增加（或者衰减）的。

23.2.1　不正当的经济动机

对于保证的讨论，一个很好的切入点就是看看各种不同当事人的动机。作为基础知识，

让我们先来考虑一下那些需要保证的事物。

- 功能性是很重要的，但却往往被忽视。那些保护了错误的事物或者是虽然保护了正确的事物，但却采取错误的方法的情况实在是太常见了。例如，可以回忆一下第8章中，在保健环境中使用Bell-LaPadula模型是如何导致出现许多问题的，而出现的问题的数量甚至超过了该模型实际可以解决的问题数量。
- 机制力度常常在新闻中出现，这都要归功于美国对于密码技术所采取的出口控制机制。许多产品，例如DVD，被加入了40位长的密钥，因此具有了固有的漏洞。机制力度独立于功能性，但却能够与后者相互影响、相互作用。例如，在第14章中，我谈及到防止智能卡探测攻击的困难是如何导致业界保护其他的，与此相关的不重要的事物，例如芯片屏蔽的机密。
- 可实现性是保证通常所关注的问题。它涉及对于给定的功能性和机制力度，产品是否已经被正确的实现。正如我们已经看到的，大多数实际生活当中的技术性的安全故障是由于编程bug，例如栈溢出、竞争条件这类错误。找出并且修正这些错误需要在保证这一环节上付出很多的努力。
- 易用性是一个容易被漏掉的因素，很少人会提到这一点，就像人们很少在节日宴会上谈及鬼怪一样。也许系统级别的故障（与纯技术性故障相对应）中的大多数都涉及到一个大型的用户接口组件。对于安全系统设计者来说，紧紧跟随保护的技术方面的发展，同时还不能忽视人类品德上的弱点是很平常的事情。这里有一些值得注意的例外情况。在第9章中所描绘的计费系统被设计用来处理用户错误；还有在第12章中讨论的安全印刷技术常常经过最优化处理来使得未经训练和粗心大意的人们可以更加容易地发现伪造者。但是，易用性除了涉及到用户之外，还与开发者有关。在第4章中谈及的开发者易用性问题是那些提供给商用操作系统的访问控制机制并没有被使用，因为使用管理员权限，运行这些代码实在是太简单了。

这四个因素在很大程度上是相互独立的，而且系统创建者必须选择一种适当的组合来作为他们的目标。例如，一位个人计算机用户也许需要高易用性、中等程度的保证（因为高就意味着昂贵，而且我们可以忍受那些临时出现的病毒），高机制力度（它们不会花费太多）以及简单的功能性（因为功能性更加重要一些）。但是，市场发展并没有体现出这些东西，其实经过短暂的思考就会明白这是为什么了。

商业平台销售商努力去获取丰富的功能性（快速的产品版本更新防止了市场被商业化，而且那些被补充技术供应商们所获取的大量市场占有份额也将遭到破坏），低机制力度（除了密码机制，关于契约的争论将使得供应商们将强大的加密看作是一种重要的市场特征），低实现保证（所以，军方级别的密码机制很容易被特洛伊木马破译），以及低易用性（应用编程人员往往比客户更加关心此事，因为他们可以增强网络应用的客观性）。

在第22章中，我描述了这种状况为什么不会很快地改变。“星期二出货且在版本3之前使系统正常运行”的策略并不像一些关于比尔·盖茨的批评性评论所说的那样，认为这是他身上的一种道德缺陷，而是因为网络经济中所固有的巨大的先下手为强原则决定的。而且，迫使应用开发者们使用操作系统访问控制的机制将更加疏远他们，从而增加了这些开发者为竞争者平台编写代码的风险性。因此，当前商业系统的不安全性从经济学家的角度来看是相当合理的，然而从使用者的角度来看就是不受欢迎的了。

政府机构理想化的思路也受制于经济学的影响。他们的梦想是能够使用商业现货供应（off-the-shelf）软件，更替一小部分组件（例如通过去除商业密码以及在原位置上插入 Fortezza 卡），而且最终还可以产生一些可用于现存防御网络的东西。换句话说，他们需要 Bell-LaPadula 功能性（他们从来不会注意到不能够支持一些供应商的其他用户的需求这类问题）和高实现保证。他们对于易用性则投入较少的关注，因为他们假设员工都是可以被训练且具有严格纪律性的人（可是这种假设是错误的）。还有需要低密码力度，这可以限制那些潜在敌人从市场上其他的高保证系统中获取利益的可能。考虑到不仅仅是高保证的花费，这种想法是不现实的，对此我将简要讨论一下。除了高保证花费外，还有就是到市场销售之前所花费的时间、满足开发团体的各种需求，以及经常性的产品版本更新会阻碍市场化的需要等等。还有，较大的网络通常会吞没较小的网络；所以，不可能期望一百万政府计算机使用者都愿意成为微软 Office 软件的用户。

在用户鼓吹者、平台提供商和政府之间的对话也许会被谴责，因为这就像是聋子们之间进行对话一样。但是，那并不意味着就不需要来谈论关于保证的话题。

23.2.2　项目保证

保证是一个十分类似于代码或者文档开发的过程。正如在代码和说明书中会存在 bug 一样，在测试过程当中也会存在 bug。所以，保证可以被作为一种一次性项目或者成为持续进化的过程。后者的一个例子就是关于已知计算机病毒的大型数据库，它是由反病毒软件提供商们经过多年积累而创建起来的，并用来做他们产品的衰退性测试。当对进化开发过程中的一个步骤使用项目技术进行管理以及作为一个特征在集成到系统级回归测试之前进行测试时，保证也能够成为包含这两项技术的一个组合。这里，你还必须找到将特征测试嵌入到回归测试集中的方法。

既然如此，首先让我们来看一看项目问题，然后再讨论进化问题是有帮助的。

23.2.2.1　安全测试

在实际测试中，安全测试通常涉及到阅读产品文档、回顾程序代码，然后进行许多项测试（这被称为白盒测试，与黑盒测试相对，在黑盒测试中，测试人员只拥有产品本身，而没有设计文档和源代码）。过程如下：

1）首先查询任何明显错误，对于明显错误的定义取决于测试人员的经验。

2）然后查找一般性错误，例如栈重写错误。

3）然后查找一系列的次一般性错误，诸如在本书各章中所描述的那些错误。

这个过程通常被特定评估环境的需求所实现。例如，也许需要显示出每一个被控制目标都被至少一个保护机制所确保的情况；在有些行业中，例如银行检查，还存在或多或少的检查列表（例如，参见 [72]）。

23.2.2.2　形式化方法

在第 2 章中，我给出过一个形式化方法的例子：可以用来检验密码协议的特定属性的 BAN 逻辑。在职工程师们所采取的形式化方法都是在大学中能广泛接触到的，但是在现实世界中却不可以应用在所有地方。这在安全商务领域当中也不是完全正确的。在诸如设计加密协议这类情况中，存在着许多问题，在这里仅凭直觉通常是不够的，而形式化确认是很有帮助的。军方的购买行为将是十分持久的，而且还需要使用形式化方法作为更高级别评估的

一个前提条件，这在桔皮书和通用准则中都有规定。我将在后面具体讨论这些内容。现在，应该具有足够知识来谈一谈关于对相对小型和简单的产品（诸如线形加密设备和像智能卡这类用于原始计算机的操作系统）限制高级进化级别的问题。虽然如此，形式化方法并不是一贯正确的。证据也可能存在错误。而且通常还会发生错误的事情被证明是正确的情况[673]。在本章开始处对于 Knudsen 的话的引用涉及到大量的、先前被证明是安全的密码算法或者协议都出现了问题。这些问题通常是由于某个证明假设不切实际，或者是已经过时而造成的。

23.2.2.3　内奸

就如同定理证明者和测试人员也可能出现错误一样，所以这些错误可同样可能出自那些专门为测试者制定测试列表的人当中（而且，可能是来自安全教科书的作者，那些测试列表的作者就是从这些教科书的指导下制定出来的）。这就是古老的“监守自盗”问题，正如《圣经·新约》中所阐述的：由谁来监督这些看守人员呢？

一个人能够做的事情很多，但其中只有极少一部分是用来声明机构中某产品是完全没有故障的。很明显的一个就是故障注入（fault injection），它通过将许多错误故意随机引入到代码当中来实现。如果有 100 个这样的错误，而且测试人员发现了其中的 70 个，再加上还有 70 个并不是故意引入的错误，一旦 30 个剩余的故意错误被移除，你能够想像在系统当中还存在 30 个你并不清楚的 bug（这假设未知错误的分布形式同已知错误是一致的；现实情况通常比这种假设要糟糕一些 [133]）。

即使在没有故意插入 bug 的情况下，也可以通过看一看由哪些测试人员发现了哪些错误而得到一个粗略的估计。例如，我将本书第 7 章提供给一些人阅读，我将在他们的观点基础上形成的勘误草案提交到了讨论会上。考虑到他们所发现的错误，以及回顾其他章节的情形，我估计在本书中还遗留着大概有 36 处的错误。这个例子的大小还没有大到足以证明单个猜测正确性的程度，在那些具有足够样本数目的情况下，我们可以使用统计学技术来分析，对此我将简要地进行描述。

另一个因素是新型攻击被发现的速度。在大学系统当中，我们通过让研究生攻击某些目标的方法来进行培训；新的漏洞和被攻击目标最终被写在研究论文当中，这些论文可以给他们带来名望，而且最终使他们可以升学。在政府机构和公共测试场所中的机制有一些不同，但是其整体的影响是相同的：大批有能力且有动机的人们寻找新的被攻击目标。学术界通常出版书籍，政府科学家们通常就不会这样做，而团体研究者们有时会这样做。所以，随着新思路的出现，你需要应用某些增加新过程到测试集中的方法，而且要牢记这个过程永远不会结束。

最后，在产品发布后，我们还要获得关于产品中那些已知 bug 实例被发现的速度的反馈信息。这对于可靠性增长模型可以提供有价值的输入信息。

23.2.3　处理保证

在近些年中，很少强调关于产品的保证措施问题，例如测试和更多关于处理措施的问题，例如谁开发的系统等。正如任何具有系统开发经验的人都知道，一些程序员所编写的代码中包含的 bug 可以比其他程序员少一个数量级之多。还有，一些机构的产品比其他机构生产出的产品要具有更高的质量。这都是产业界十分关注的话题。

在高质量和低质量开发小组之间的一些差异通过直接的管理干涉是可以被修正的。也许最著名的就是人们是否对纠正自身 bug 负责。在 20 世纪 80 年代，许多机构将系统开发的瀑布模型解释为有一组人编写说明书，而另一组人编写代码，有一组人则做测试工作（包括一些 bug 修复工作），还有一组人做维护工作（包括对 bug 的修复进行测试）。这些组之间的通信仅仅通过项目文档进行。这被证明是合理的做法，因为对于人们来说可以在某一时间更加高效地集中精力于单一工作之上；打断程序员的工作，而让其修复一个他自己都已经忘记的、6 个月之前所编写的代码当中的 bug，这将花费他一天的时间，如果是让维护程序员来做这件事情可能只需要一个小时。

但是，这种方式的影响就是编码者产生了许多兆字节的含有 bug 的代码，而让那些拙劣的测试人员和维护人员清除这些 bug。随着时间的过去，产品的质量和生产效率都会下降。产业界的分析人员将 IBM 公司在 90 年代早期花费了 1000 亿美元资产进行此类开发而差点倒闭的原因归于此 [169]。作为其对手，微软考虑到从这件事中所得到的至关紧要的教训，随着编写大型程序不断出现问题，公司策略改为“如果你编写了代码，那就由你来修复其中的 bug。”bug 应该被尽可能快地修复；而且即使 bug 也同死亡和缴税一样不可避免，程序员们也决不应该放弃编写干净代码的想法。

机构的其他可控制方面对于输出质量也起到很重大的影响，这包括你的机构在雇佣人员方面是否英明，以及如何培训他们专业技能和工作习惯（参见 Maguire 对于微软策略的扩展性讨论 [521]）。

许多年来，内部审计人员在评估安全代码质量的同时也需要处理所发现的问题。这比你想像的要困难许多，因为绝大部分机构的质量文化是难以明了的。而一些规则（例如“修复你自己的 bug”）看起来又相当普遍，强制加入大量的特殊规定将引起墨守成规的循规蹈矩文化，而不是动态的竞争性文化。因此，最近的工作都面向于小组能力的全面评估；领导这项工作的是来自卡内基-梅隆大学的软件工程学院的能力成熟度模型（Capability Maturity Model，CMM）。

CMM 是基于这种想法的，因为小组需要经验，所以它的发展需要经历一系列的级别。这个模型包括 5 个级别，即初始级、可重复级、已定义级、已管理级和优化级，随着级别的增加会加入一些新事物。因此，例如，项目计划必须被引入并从初始级转向可重复级，而相应的评论也可以从可重复级向已定义级转变。在 [767] 中有更完全的描述和参考书名；对于在安全工作当中采纳 CMM 的尝试也已经做过一些，许多供应商们都已经使用它了 [545, 822]。

一个更加通用的质量保证方法是 ISO 9001 标准。该标准实质就是一家公司必须将设计、开发、测试、文档、审计过程以及管理控制等进行文档化。要想了解到更多的细节性信息，参见 [767]；目前，整个业界的咨询顾问都在帮助公司获得 ISO 9001 认证。在最好的情况下，它可以为增长的过程改进提供一个框架；公司能够监控那些出错的事情，反向跟踪直到源代码一级，然后进行修复和防止其再次发生。在最糟糕的情况下，它只是在循规蹈矩式文化下的一次练习而已，仅仅是将混乱变成了更加墨守成规的混乱而已。

许多作者评论到，组织机构具有一个自然的生命周期，就如同人一样。而 Joseph Schumpeter 则认为，经济低迷导致产生了一种有价值的社会进化现象，即淘汰那些已过时或者不再适应市场需求的公司，这与大火可以让森林重新恢复生机是一个道理。那些成功的公司会

变得越来越自满和循规蹈矩，以至于一些内部人士选择追逐舒适的生活方式，而其他人则选择离开公司（以前经常会说，那些曾经离开 IBM 公司的员工都是真正优秀的人才）。公司过快的增长速度也会带来问题：微软内部人士抱怨许多当前存在的问题都应该归因于 20 世纪 90 年代后期成千上万新员工的流入，他们中有许多人的动机是希望从股票交易当中获取好处，而不是完成编写高质量代码和让程序运行在世界上各个地方的计算机中的使命。

计算机产业界中的公司的诞生、消亡和再生的循环速度比起其他行业要快许多，这都是因为技术进步和多重网络客观性共同作用的结果。电信业由于计算机和通信业的融合以及电话公司必须将 15 年的产品循环周期缩短为 15 个月，从而跟上微软的步伐而产生重大改变。安全业界开始感受到同样的压力。那些开发密码器或者军用 MLS 系统这类增值合同而有规律地工作数十年的开发小组突然就暴露在技术和市场的强大压力之下，而且还被要求创建与先前完全不同的事物。一些小组成功了，如同 MLS 提供商 TIS，它重新将自己打造成为防火墙供应商；而其他小组则失败甚至从业界消失了。因此，人们对于这些 MLS“老人”的价值是持怀疑态度的。一般情况是，开发组通常只是依赖于一位或者两位关键性的领军人物，而且当这些人离开并重新起家的时候，该小组的能力在一夜之间就化为乌有了。

诸如 ISO 9001 和 CMM 这类认证机制如果可以确保通过某些方法来远离那些失去领军人物、失去光彩和能力的开发小组的话，那么将变得更加具有说服力。人们正在试图考虑这种解决办法可能存在于使用在食品指南中的等级系统当中。宣布建立一家新的“旧金山最好的亚洲餐馆”将使那些先前使用该标题的餐馆被淘汰出去。当然，如果需要认证很容易过时的资产，那么将不得不利用公司强大的市场优势投入等努力来获取该认证了。这是切实可行的：餐馆指南系统正常工作，而且相应的学术观点也在走类似路线。

23.2.4　保证增长

基于处理的保证的另一个方面是大多数客户并不对开发小组产生任何兴趣，而只对产品本身有兴趣。但是，今天的大部分软件都是打包式而不是预定式的，其开发过程将是持续进化增强的模式，而不再是一次性项目的模式。那么，关于正在发展的产品的保证级别还有任何有用的建议吗？

如果由于产品进化而引入的新 bug 和旧 bug 的清除可以保持相同速率的话，那么这类产品的质量可以得到平衡。但是，对此并没有什么保证机制（还有第二重的影响，例如衰老，当重复性增强使得代码变得越发复杂，以至于其底层的可靠性和可维护性下降，但是为了叙述简单的缘故我将忽略这一点）。

当控制 bug 引入速度取决于我已经描述过的开发控制类型时，测量 bug 被清除的速度将需要不同的工具——在测试中关于如何增强软件（或者通常说是系统）可靠性的模型。

关于现实生活中的变化我们所知道的要多一些，因为除了软件工程师之外，许多的人都对此有兴趣。

当测试人员希望在系统当中找出单一的一个 bug 时，存在一个合理的模型，即泊松分布（Poisson distribution）。当前概率为 p，bug 经过统计性随机测试后保持未被发现状态的概率变为 $p = e^{-Et}$，在这里，E 依赖于它所能够影响到的可能输入的概率 [506]。在系统可靠性由单独一个 bug 所决定时，例如当我们在系统中查找第一个或者最后一个 bug 时，其可靠性的增长是呈指数级的。

但是广泛的实践调查表明，在大型和复杂的系统当中，第 t 次测试失败的可能性并不是与 e^{-Et} 成比例，而是与 k/t 成比例，其中 k 为一个常量。所以系统可靠性的增长速度很慢。这种现象首先在 IBM 大型机操作系统所产生的相关 bug 历史中被发现和记载［7］，而且在许多其他著作中被予以证实［514］。由于 k/t 的故障概率意味着平均无故障时间（mean time between failure，MTBF）也大约将变为 t/k，可靠性随测试时间呈现线形增长的趋势。这种结论通常被保险关键系统界称作，“如果你希望平均无故障时间为一百万小时，那么你测试系统的时间至少也要一百万小时［150］”。这就成为驳斥那些复杂关键系统开发过程有失误的一个主要观点，因为它们在投入使用之前并没有经过完全的测试，例如弹道导弹防御系统。

有关 k/t 行为的原因描述出现在［105］中，而且是在［133］中许多通用假设的条件下被证明的。后者使用了统计热力学技术，而且它的核心思想是彼此间独立的 bug 数量足够大（达到某种统计学假设的数量要求）的场合中，其中每个 bug 的出现概率为 $p_i = e^{-E_i t}$，且这些 bug 经历很长一段时间后将消失，那么对于整个系统而言，所有单个 bug 的 $e^{-E_i t}$ 统计值总和为 k/t。如果，它们消失的速度比这个速度还要慢的话，软件就永远不可能完全正常工作；而如果它们消失的速度非常快的话，那产品将很快成为无 bug 产品，这通常是我们无法做到的。

这个模型给我们带来了其他许多有趣的结论。在通常是合理的假设条件下，最有可能出现：需要进行一百万小时测试，从而达到 MTBF 为一百万小时的规则是不可避免的，而且测试时间还要依赖于代码和测试范围的初始条件而增加常数倍。这合起来就会成为对墨菲定律另一个版本的一个证明，即那些经受一系列选择过程后的还存在的漏洞数目是最多的。

这个模型与生物物种在竞争性选择下的进化数学模型十分相似。bug 所扮演的角色大致上对应于基因降低适应性所起到的作用。但是，两者实现是显著不同的。墨菲法则，即那些经受一系列选择过程后还存在的漏洞数目是最多的，也许对于工程师来说是一件坏事情，但是对于生物物种来说就是一个好消息了。当软件测试清除掉一小部分可能的 bug 时，与所应用的测试过程是一致的，而生物进化使得一个物种来适应环境变化，这个过程只需在物种初期死亡时花费很低的代价，而同时又尽可能多地保持了多样性。这种多样性帮助物种们在未来的环境突变中继续生存。例如，如果许多兔子被蛇捕食，那么兔子中那些更加机敏而不是速度快的就会被选择出来。兔子在速度上的弱点仍然保留，但是如果狐狸到来的话，在选择性掠夺的影响下，兔子群中的平均奔跑速度将迅速提升。更加严格地说，自然选择的基本规律可以表述为，一个具有高基因变异的物种可以更加快速地适应于变化的环境。但是当 Fisher 在 1930 年证明该观点时［297］，他同样也证明了复杂的软件被移植到一个新环境中时，将呈现出可能出现的最大数量的 bug。

进化模型还指出，在可靠性获取方面存在着来自于可重用软件组件的基本限制，这些可重用组件包括对象或者库；经过良好测试的库只不过意味着整体的故障率由新代码控制。它同样解释了安全关键型系统中的一个发现，即测试结果通常只是一个表现拙劣的性能指示器［506］：由测试人员测量的故障时间仅仅依靠于程序的初始质量、测试范围和测试次数，所以并没有提供关于程序在另一个环境当中所可能表现性能的进一步信息。这里还存在一些无法预料到的结果，而这些结果从对照的角度来看就再明显不过了。例如，每一个 bug 对于总体故障的贡献是独立于包含该 bug 的代码是经常性执行还是偶尔执行的。但凭直觉来看，执行次数少的代码被测试的次数也会少一些。最后，正如我们在 22.4.3 节中所提到的，采用

不同测试人员并行的方法比起串行的方法将更加经济有效。

简而言之，复杂系统只有通过漫长的测试才可以变得更具可靠性。因此，一旦有成千上万的人阅读过此书并且报告出其中存在的 bug，那么本书就具有相当的可靠性了；但是，如果它是第二版，其中包括了许多新内容的话，那么新 bug 一定将存在于这些新内容中了。对于那些市场占有率很高的软件，它的广泛使用从理论上说是一种快速调试过程；但是，在实际当中，由于网络经济而必须不断出现的软件新版本则大大限制了原先所期望出现的快速调试过程。

如果一种 bug 被定义为可以导致安全漏洞，而不仅仅是原先的漏洞的话——只要 bug 的数量达到可以做统计的程度，那么这些结论为什么不能适用于软件的整体情况呢，这个问题好像并没有理由来回答。

23.2.5 进化和安全保证

可靠性的进化发展对于软件工程师来说要比一个生物物种的进化糟糕许多，但是，对于安全工程师来说，这种情况甚至还要更加糟糕。

让我们在讨论细节性机制问题之前先来看一个简单的例子。假设像 Win2K 这样大型和复杂的系统具有 100 亿个 bug，每一个 bug 的 MTBF 为 10 亿个小时。还有，假设 Paddy 为爱尔兰共和军工作，而且他的工作就是入侵英国军队的计算机，从而获取在贝尔法斯特的告密人名单。同时，Brian 在军队中是做保证工作，即阻止 Paddy 这类人的行为。所以，他必须在 Paddy 掌握这些 bug 信息之前了解各种 bug 的情况。

Paddy 在白天还有一份工作，所以一年当中他只能够进行 1 000 小时的测试工作。而另一方面，Brian 拥有 Windows 系统全部的源代码，许多博士，还有商业性进化测试场所的控制权，CERT 所发的内部小册子资料（这是一份用来与其他 UKUSA 成员国进行信息共享的资料），以及通过执行政府方案派出顾问到关键性行业当中，例如电力和电信行业，去找出如何攻击它们的方法（其实是建议他们如何保护他们的系统）。而且，Brian 在一年当中可以进行 1 000 万小时的测试工作。

一年之后，Paddy 发现了一个 bug，而 Brian 则发现了 10 000 个 bug。但是，在 Brian 所发现的 bug 中包含 Paddy 所发现的那个 bug 的可能性只有 1%。即使 Brian 宣布了戒严令，将全英国 50 000 名计算机专业大学毕业生都关到格洛斯特郡的集中营里，让他们一起来搜寻 Windows 源代码，他每年还是只能完成 1 亿小时的测试工作。在十年之后，他将发现 Paddy 的 bug。但是到那时，Paddy 又将发现另外 9 个 bug，而且很有可能 Brian 不能全部知道这 9 个 bug。更糟糕的是，Brian 的 bug 报告将成为一个盲点，Bill 将会把它们都过滤掉。

换句话说，Paddy 在他这边进行测试是有动力的。甚至一个具有适度能力的攻击者就能够破坏那些大型和复杂的系统。只要存在足够多的不同的安全漏洞来，那么就没有什么方法可以阻止这件事情的发生。惟一的一线希望在于，如果你所有的漏洞都是栈溢出，而且你开始使用一个新的编译器来捕捉它们，那么对此建模，就只会出现一个单一的漏洞，这样你就可以逃离概率统计陷阱。

23.3 评估

对于评估（evaluation）的专业定义是“关于系统达到或者没能达到一个事先规定的目标

的证据收集过程。”（评估通常与测试有些重叠，所以有时会把它们混淆）。正如我曾提到的，这里的证据也许只是需要用来使老板相信，你已经完成了工作任务。但是，它常常被用来让那些将来依赖于该系统的人放心，让他们认为开发或者运行该系统的人已经熟练精巧地完成了这项工作。当保护机制的实现方和依赖于该保护机制的一方的观点不一致时，二者间逐渐增加的紧张关系就会成为基本问题。

有时，这种紧张关系是简单而且可见的，例如当你为被保险的标准设备设计防盗报警系统时，同时在保险公司的测试场所中进行检查员的鉴定。有时，这种紧张的状态仍然可见，但是却更加复杂，例如设计政府安全标准，而该标准试图在众多相互矛盾的、制度中的利益间进行调和，或是当雇佣公司的审计员来评论系统并且告诉老板它是适合某种目的时。当评估涉及到多个当事人时，问题将更加困难；例如当一个智能卡制造商希望得到政府机构（该机构试图鼓励使用例如密钥契约等特征，而这并不能引起其他人的兴趣）的评估许可证，从而可以将卡销售给银行，而银行反过来则希望使用该卡获得某种可靠性机制来防止客户的欺骗行为。这些做法看上去都具有相当的蒙蔽性，但是却并没有涉及到任何人的非法欺骗的行为。这些欺骗性行为是由管理者的个人和部门的需要所引发出来的很自然的特性。

例如，管理者们经常购买那些他们自己也知道不是最佳甚至是有缺陷的产品和服务，但是这些产品是来自大公司的供应商们。这就减少了当系统出现问题时他们遭到解雇的可能性。法人律师们并不把这种行为看作欺骗行为，甚至还称赞其为一种努力勤奋的做法。最终的结果也许是，信赖方，客户，无论发生什么情况都是什么也不说，所以当出现问题时就很难纠正银行、供应商、评估者或者是政府机构的问题。

另一个严重并且很普遍的问题是词语“保证”和“评估”通常被解释为只是应用在系统中的相关技术方面，而忽略了其易用性（没有涉及到其更加广泛的适当内部控制和良好的团体管理的问题）。公司主管还希望得到保证，保证的内容是具有可管理的过程、在账务中没有实质性错误、依照法律办事，以及许许多多其他的事情。但是，许多评估方案（尤其是通用准则）故意忽略掉系统当中的责任和组织管理成分。如果存在任何的想法来支持这些评估方案，那么对这些内容的评估就被认为是为了客户 IT 审计人员，或者系统管理者来创建配置文件的事情了。但不管如何，我在后边仍会将注意力放在技术评估上面。

很容易我们就可以将评估划分为两种情况。第一种就是评估由信赖方执行；这包括由 NASA 对关键任务代码所进行的保证评估、独立验证和确认，以及上一代军方评估标准，例如桔皮书。第二种就是评估由非信赖方中的某人执行。目前，这通常意味着通用准则评估过程。

23.3.1　信赖方的评估

在第 10 章中，我讨论了许多关于保险公司对防盗报警系统的关注，以及批准设备在一定大小的风险范围内被使用等方面的考虑。如果系统足够简单，该批准过程可以自动处理；保险业控制具体执行测试过程所在的测试场所。这些过程也许可以包含一个固定的预算（也许是一个人花费两个星期，或者是 15 000 美元的开销）。评估人员从开始就有相当明确的思路，知道一个防盗报警系统应该做什么和不应该做什么，依赖于固定的预算而寻找缺陷，然后编写评估报告。然后，测试场所或者批准、或者拒绝，还可以要求进行某些改进。

在 7.4 节中，我描述过另一个评估模型，这个模型在 1985 年到 2000 年间被创立，在

NSA 国家计算机安全中心(NSA's National Computer Security Center) 被用于评估那些建议美国政府使用的计算机安全产品。这些评估按照桔皮书进行，桔皮书就是可信计算机系统评估标准 [240]。桔皮书和其支持文档制定了许多评估等级：

C1：按用户组自主访问控制。实际上，这相当于没有任何保护机制。

C2：按单用户自主访问控制；客体重用；审计。C2 对应于被仔细配置的商业系统；例如，C2 评估通过 RACF 用于 IBM 大型机操作系统和 Windows NT 系统中（这两个系统都需要在特定版本和配置下才可以达到 C2 标准，例如对于 NT 系统来说，就被限定为无盘工作站)。

B1：强制访问控制。所有对象携带安全标号，安全策略（Bell-LaPadula 或者其变体）独立于用户行为而被强制执行。标号被强制用于所有的输入信息。

B2：结构化保护。类似 B1，但在这里必须具有一个正式的安全策略模型，而且该模型已经被证明和安全公理具有一致性。必须为系统管理和配置管理各种工具。TCB 必须被正确地结构化，而且其接口要被明确定义。隐蔽通道分析必须被执行。必须从用户到 TCB 提供一条被信任路径。严格的测试，包括必须进行渗透性测试。

B3：安全域。类似 B2，但是 TCB 必须最小化；它必须调节所有的访问请求，而且防篡改，能够经受正式的分析和测试。还必须具有实时监控和报警机制，而且，在实现中必须应用结构化技术。

A1：验证设计。类似 B3，但是形式化技术必须被用来证明 TCB 说明书和安全策略模型之间的等价性。

系统的评估等级决定了系统可以处理何种信息的属性。我在 7.5.2 节中给出的例子就是一个被评估为 B3 级的系统通常将所处理的信息分为不保密、秘密和机密三类，或者是秘密、机密和绝密（完整的规则可以在 [244] 中找到)。虽然这些等级在 2001 年底将要被取消，但是在业界它们还是具有决定性影响的。

桔皮书评估的业务模型是根据传统的政府服务工作情况而制定的。政府官员希望所要使用的某些产品被评估；NSA 将分配人员来进行此项工作；他们将做这项工作（考虑到传统行政事务的谨慎和拖延，这项工作可能需要花费 2～3 年的时间)；如果成功，产品将加入到被评估产品的列表中去；最后，纳税人将收到账单。这个过程是由政府来发起和控制的，政府将依赖于评估的结果；而制造商就像是站在门口的请求者。因为整个过程需要花费一定的时间，所以被评估产品通常都是比当前的商业产品滞后一到两代，而且通常在投入上还要高出一个数量级。

在美国，桔皮书并不是惟一的评估方案。我在 14.4 节中曾经提到过一种用于评估密码处理器防篡改的 FIPS 140-1 方案；这种方案像项目承包人一样使用多个测试场所。独立的项目承包人也是为了独立验证和确认（Independent Verification and Validation，IV&V）方案而使用的。该方案由能源部建立，并被使用在核武器系统当中，后来被 NASA 采纳用于载人航天飞行系统中，它保留了核武器评估系统中许多相似的内容（至少在火箭末端是这样的)。在 IV&V 中，有一种简单的评估目标：0 缺陷。这个过程仍然是由信赖方，即政府来推动和控制。IV&V 承包人是创建该系统的公司的一个竞争者，而且它的报酬是与所发现的 bug 数目紧密联系的。

其他政府也有类似的方案。加拿大政府使用加拿大可信任产品评估标准（Canadian Trusted Products Evaluation Criteria，CTPEC)，而一些欧洲国家制定了信息技术安全评估标准

(Information Technology Security Evaluation Criteria, ITSEC)。其思路就是一种共享的评估方案可以帮助欧洲防卫项目承包人利用他们庞大的经济规模和美国供应商们展开竞争；在英国、法国和德国，欧洲人不再需要出具其他单独证明。ITSEC 综合了桔皮书和 IV&V 中的思路，这体现在它也具有许多不同的评估级别；而且对于除最高级别外的其他所有级别，其项目内容都可以承包出去。然而，ITSEC 引入了一种有害的创新机制：政府所要求的评估不是由政府来支付费用，而是由那些对其产品进行评估的供应商支付。

这正是通常行政事务中一石几鸟的思想：节省公共开支，同时又提升了市场的竞争环境。但一般来说，如果石头出了问题对于那些过于精明的捕猎者来说所造成的危害比对任何一种鸟所造成的危害还要大一些。

这种规则上的变化导致出现了一种受到批评的不正当动机。它鼓励制造商寻找那些可以尽快使其产品投入使用的评估承包人，这些承包人可能询问很少的问题、收取费用也相对较低、或者花费的时间很短，又或者这三种情况全部都存在（目前，在 FIPS 140-1 中也存在相同的方式，公司们开始依赖于第三方的评估）。为了公平，规则制定者对于这点的潜在危害性已经意识到了，而且也已经制定出了相应的方案，通过这些方案，承包人必须获得批准成为一个商业许可评估机构（commercial licensed evaluation facility, CLEF）。对于 CLEF 可能撤销其许可的威胁就来自试图通过抵销商业压力的办法来走捷径的做法。

23.3.2　通用准则

这里将调整一下通用准则的发展阶段。桔皮书最初目的是用来发展可在所有主流的操作系统当中标准化的保护措施，而不是一种被用来适应政府市场的价格高昂的附件（正如被桔皮书所评估的产品所变成的那样）。问题以过小的市场来分析，而解决办法又将它们扩张成面向大市场。因为防卫项目承包人憎恨那些必须获得关于他们产品单独评估的做法，这种情况在美国、加拿大和欧洲都是如此。所以，协定使得各国的评估方案变得支离破碎，并取而代之以单一的标准。这项工作在 1994 ~ 1995 年时就完成了，而且欧洲模型逐渐取代了美国和加拿大的替代方案。通过 ITSEC，在通用准则之下评估完全可以进行，但是最高级别的评估还需要使用 CLEF 来完成，而且这些评估应该被所有参与国认可（虽然任何国家在国家安全面临极其危险的境地时，可以拒绝承认某项评估）；并由供应商们支付评估费用。

这里与桔皮书有一些不同之处。最为至关重要的一点就是，通用准则比起桔皮书具有更大的灵活性。它不是希望所有的系统都遵照 Bell-LaPadula 模型，而是一个产品将对照着保护轮廓来进行评估。该保护轮廓至少在理论上能够被客户设计出来。这并不意味着国防部放弃了多级安全以及扩大保护范围，但它可以让许多商业 IT 供应商们使用通用准则方案，而因此消除在 23.2.1 节中所描述的有害的经济动机。通用准则热切希望创造一个广泛的影响，可以导致商业世界在某种程度上自适应于政府的做事方式。

23.3.2.1　通用准则术语

为了更加仔细地讨论通用准则，我需要首先介绍一些行话。处于测试状态下的产品被称为评估目标（target of evaluation, TOE）。检验被执行的精确程度被称为评估保证级别（evaluation assurance level, EAL）；它的范围是 EAL1 ~ EAL7。其中 EAL1 表示进行功能测试就足够了，而 EAL7 则要进行彻底的测试，包括正式的验证设计。为商业系统获得的最高的评估级别通常为 EAL4，虽然有一种智能卡操作系统是具有 EAL6 评估级的（然而，这是在 ITSEC 下

而不是 CC 下获取的)。

一个保护框架就是一组安全需求和它们的基本原理以及一个 EAL。框架应该通过一种与实现独立的方法来表达，从而可以在产品和版本间进行评估的比较。一个安全目标（security target，ST）是为了一个给定的评估目标，对保护框架进行的一种精确表述。除了评估一个产品外，还可以评估一个保护框架（这种思想是来保证它的完全、一致和技术合理性）以及一个安全目标（从而来检查正确地提炼了一个给定的保护框架)。当从无到有地设计某些事务时，思路是首先创建一个保护框架，对其进行评估（如果一个适当的评估还不存在的话)，然后对安全目标也做同样的事情，最后评估实际的产品。所有这些行为的最终结果应该是提供对保护框架的一个注册以及被评估产品的目录。

一个保护框架应该描述环境的假设、目标以及保护需求（根据功能和保证而确定)，还要将它们分解为组件。这里有一种程式化的方法来做这件事情。例如，FCO_NRO 是一个功能组件（因此以 F 打头)，它与通信（CO）相关，而且涉及到身份认可（NRO)。其他类包括 FAU（审计)，FCS（密码支持）和 FDP，它的意思是数据保护（这并不是欧洲法律中所说的数据保护，而是指访问控制，Bell-LaPadula 信息流控制以及相关特性)。组件目录在支持 MLS 系统时是有较大偏差的。

还存在一些其他的目录：

- 威胁，诸如 T.Load_Mal——“数据加载故障：一名怀有恶意的攻击者在组织数据的过程中产生错误，从而威胁到 TOE 的安全功能。”
- 假定，诸如 A.Role_Man——“角色管理：对于 TOE 中角色的管理采用一种安全的方式来执行”（换句话说，让开发者、操作者等等的行为规矩一些)。
- 组织策略，诸如 P.Crypt_Std——“密码标准：密码实体、数据认证和批准的功能必须和 ISO 以及相关产业和组织的标准相一致。”
- 目标，诸如 O.Flt_Ins——“故障插入：TOE 必须可以抵制通过插入错误数据而进行的重复性探测。”
- 保证需求，诸如 ADO_DEL.2——“修正检测：开发者为了将 TOE 或者其中的一部分交付给用户，将执行证明过程。

我曾经提到过，一个保护框架将包含一个基本原理。这通常包括一组表格，用来显示每一种威胁是如何被一个或者多个目标所控制的，也可以反过来说，即为什么每一个目标的必要条件是一些威胁、环境假设以及支持解释的组合。我还将证明为了特定强度的功能所选择的保证级别和需求是正确的。

获得这些信息的最快方法就是阅读一些已经存在的框架，例如智能卡的保护框架[579]。通过这些保护框架，可以提供很长一列的可能出错的事物以及开发人员能够控制的一些事情，所以这是一个十分有用的检查列表。然而，对于卡保护而言，真正重要的方面可以在 O.Phys _Prot 中找到，“物理保护：TOE 必须可以抵制物理性攻击或者能够在此类攻击所需信息的理解上创造各种困难”。一个没有经验的读者也许没有意识到，这个目标是整个事情的核心；而且正如我在第 14 章中所解释的，它是很难被满足的（对于 100 ~ 101 页中的任务，有能力的攻击者仍然可以做到，但是却用术语表达出来，将使外行的读者不理解了)。通常，标准和使用这些标准所产生的文档都是很难阅读的，所以他们破坏了本来希望可以带给非专业工程师的应有价值。

对于安全工程师来说，通用准则仍然是有用处的，这体现在它们提供如此广泛的事务列表来进行检查。它们还能够提供一个管理工具来维持对各种威胁类型的跟踪，从而保证它们都被正确处理（否则，在有大量细节的情况下，很容易遗忘其中的某种威胁）。但是，如果客户坚持评估，尤其是在一个较高的层次上，那么这些列表就容易从一种帮助变为一种负担。在接受所有这些将导致的花费和延迟之前，理解通用准则没有做什么是很重要的。

23.3.2.2　通用准则没有做什么

文档中都声称通用准则不处理管理安全措施，也不处理“技术性—物理性”方面的问题，例如 Emsec，及密码算法、评估方法论以及标准如何被使用的问题。文档中还声明不要假设任何特定的开发方法论（但是后来就依赖于假设使用瀑布模型的方法）。这其实是同意了根据经验来开发策略的方向，而对产品的重评估则不再被考虑。而且，并不存在关于保护框架适应真实世界的证据需求；我已经看到了一些故意忽视关于相关攻击话题的书籍出版的行为。换句话说，标准避免了所有安全工程当中困难和感兴趣的内容，而且能够很容易地变成动臂装卸机的特许权。

最常见的批评（不是指代价和墨守成规方面）就是，标准过于关心设计的技术方面问题。例如，在 ADO_DEL.2（以及其他别的地方），我们发现程序相比于技术保护来说被看作是次要的东西（其思想就是在那些技术性装置不可用的地方求助于程序）。但是，正如在 12.6 节中所解释的，当评估一个真实的系统时，你必须在过程中的每一个阶段都要评定职员的能力和动机。这是一件很基础的事情，而不是可以在后期才被添加进去的。

更加基础的是，商业过程不应该被可用技术的限制而驱动（而且尤其是被那些可用的、高代价的、过时的军方技术所限制）。系统设计应该由商业需求驱动；而且技术机制只应该在那些被证明是应该使用该技术的地方使用，而不是因为这些技术存在就被使用。特别地，技术机制不应该用于那些作用低于其控制花费的场合，或者是程序上的控制十分便宜的场合。还记得为什么塞缪尔·摩尔斯在 19 世纪早期可以打败所有的竞争对手而成功创建了电报。当时人们试图创建调制解调器，所以他们能够从一端到另一端传递文本；摩尔斯意识到这一点，在使该项技术可用后，接着教会人们应用该技术创造的产品花费是很低的。

关于标准所出现问题的理论有许多。通常，实际引起的争执各不相同，而且很少会引起人们的兴趣。

23.3.3　什么容易出现问题

在我所经历过的一些或者是受到影响的案例当中，没有一个案例使用通用准则的方法并且证明是满意的（也许，那是因为仅仅当事情出现问题时，才会叫我去解决问题——但是我的经验还是表明了在评估过程当中缺乏健壮性）。

首先必须提出的一个要点就是做评估工作的 CLEF 对于他们向当地情况机构的登记表示十分感谢，这样他们的员工全部都是经过调查的。但这就为制度上的腐败留下了一条相当广阔的路径。

腐败并不是必须包括金钱交易甚至是直接交易好处。例如，当工党在英国 1997 年大选中胜出时，我很快就接到一个来自贸易和工业部门的官员打来的电话。他想知道我是否认识利兹大学的某位计算机专家，这样该部门就可以资助我的部门一些资金来与他们进行合作研究。我得知这位即将到来的科学部长是利兹地区的选民。这并不意味着这位部长让他的官

员来为其本地大学寻找一些钱，但几乎可以肯定，许多官员都曾经试图找他闲聊类似的事情。

23.3.3.1　腐败、操纵和惯性

这种抢先的阿谀奉承行为在评估测试执行过程中变得越来越普遍。在我的经历当中，最过分的例子是发生在英国国家医疗服务机构（British National Health Service）。在来自医生的压力之下，该部门同意对卫生部门网络上传输的数据进行加密；而 GCHQ 使得对于使用密钥契约的期望变得毫无秘密可言。一些测试版软件被安排使用；其中的一个用户从 Danish 供应商获取的商业加密软件，而该供应商没有密钥契约，该软件花费 3 000 英镑，而其他的试验则使用来自一个英国防卫项目承包人的软件，该承包人具有密钥契约，软件花费 100 000 英镑。使 GCHQ 尴尬的是，Danish 软件可以工作，但英国提供商的软件却根本无法工作，毫无用处。这种情况很快就通过让具有 CLEF 许可的公司来评估测试版软件的方法得以解决。但在它的报告当中所声称的正好相反：契约软件工作良好，而国外的产品存在各种各样的问题。也许 CLEF 被告知了应该在报告当中写些什么和不应该写些什么，还有可能是员工所写的都是 GCHQ 所希望看到的东西。

有时，为取悦消费者所表现出的热心是显而易见的。在冰岛的健康数据库（参见 8.3.4.1 小节）的使用环境中，它的创办人想要抵制来自于师生们的关于隐私问题的批评意见，所以他们聘请一家英国的 CLEF 来为其编写保护框架。在标准的行话中，这也可以被表述为创办人最初的设计和主张；它故意避免注意到设计中的缺陷，这些缺陷甚至已经被记录在案或者在冰岛电视台中公开讨论过 [38]。

有时，这些保护框架可能非常的健全和可靠，但是将它们映射到实际应用中时就不是这样了。例如，欧洲政府和 IT 供应商目前正为了“高级电子签名”的规范而进行着工作。关于“高级电子签名”，我在 21.2.4.4 小节中提到过，它不久将被看作是欧盟所有成员国中手写签名的等价事物。目前的建议是，产生签名的设备应该是一种评估级别在 EAL4 级之上的智能卡（其框架 [579] 是为增强的 EAL4 而设计的，它正如所提到的那样，足可以用来防御来自于除那些具有很高能力的攻击者之外的所有攻击者的威胁）。但是，对于个人电脑而言，并没有提出某种要求用于向你显示那些你认为正在签署的资料。最终的结果将是一种在个人电脑的病毒或者特洛伊木马被发送到智能卡上面的“安全”（在认可的情况中）签名。

当然，内部人士断定甚至还可以通过更加老练的方法来操纵系统。一个很好的例子来自 10.4 节当中曾经描述过的法国如何阻止英国和德国反对基于智能卡技术的电子转速表。他们编写了一个不是十分严格的保护框架，并且将其发送到一家英国的 CLEF 进行评估。CLEF 是一个军队软件公司；无论它们关于 MLS 的知识如何，其对智能卡是一无所知的。但是，这并不会导致它们就会拒绝这项业务。它们也不知道英国政府反对采用保护框架。因此，英国只能在以下两种方案中选择其一，即接受把安全标准作为既成事实这种有缺陷的方式，或者是破坏存在于通用准则当中的信心。

考虑到所有的腐败、贪污、贪婪、无能力以及操纵的情况，从中发现一些好的、老式的墨守成规型的东西就像是呼吸到新鲜空气一样了。一个例子就是由美国政府开发的保健保护框架。尽管在保健系统中使用 MLS 保护的问题有许多，正如我们在第 9 章中所提到的那样，但是它还是框架最终所使用的模式 [34]。它假设没有使用者是怀有敌意的（尽管事实上，

几乎所有的对于健康系统的攻击者都来自系统内部），而且坚持支持多级安全，尽管在第 9 章当中我们说过级别在这个环境当中无法工作。它也没有提供任何规则是有关于如何管理分级和分割的，但是留下了访问控制策略决定来获取“需要知道”的一切东西。

23.3.3.2　潜在问题

通常，通用准则的结构是面向 MLS 系统以及那些支持它的设备的，例如政府防火墙和加密盒。考虑到开发这些设备的机构的使命，这就不足为奇了。这些设备假定使用者都经过培训而且十分服从命令、小型系统可以被正式地审核、一致的 MLS 类型的安全策略以及缺乏高层次攻击，例如法律挑战等。这使得它们对于现实世界中的大多数应用来说确实是没有什么用处的。

至于组织方面，我在 23.2.3 节中曾经提到，当那些可信任的开发小组失去光彩时基于过程的保证系统却并没有丢失其鉴定资格，那么系统将会失败。这被很明显地应用到 CLEF 上面。即使是 CLEF 被独立于情报系统的中立机构所授予许可，随着关键员工们的离开以及技术跟不上发展等原因，许多事情也将恶化。随着客户寻求更加容易的评估，将不可避免地出现等级膨胀的情况。然而，目前，没有一种可以被使用的机制，通过该机制，确实不具备竞争能力（甚至是不老实）的从业者能够挑战 CLEF，并且将其从列表中删除。在缺乏对于行为不端者的制裁的场合中，制度上的腐败将仍然是一种很严重的风险。

当发布一个新型的安全产品时，工程师必须时时刻刻考虑销售代理是否在说谎或者犯错误，而且他们是如何这么做的。通用准则应该修复这个问题，但是它们并没有这么做。当伴随着来自被评估列表中的产品发布时，你必须询问这些保护框架是如何被操作的，以及是由谁操作的；是否 CLEF 不诚实或者不合格；政府在幕后采用了哪些压力；以及你的权利是如何通过证书而产生变化的。

例如，如果你使用一个没有被评估过的产品来产生数字签名，那么一个伪造的数字签名将出现，而且某些人将试图使用它来冒充你，你也许希望获得证据，从而让法庭强制发布全部文档给内行的目击者。一份通用准则证书将让法庭很少来进行揭发，从而会严重地损害你的权利。实际上，机构内部人士承认，最主要的问题还是“自信”，也就是让人们接受系统是安全的这种观念，尽管情况并非如此。

一位愤世嫉俗者也许会建议，在商业世界中，这正好是以下这些问题产生的原因。这些问题是为什么产品供应商需要依赖于这些难于理解的状态，以及这些产品为什么被设计用来传递可靠性（例如智能卡），来满足应该付出努力的需求（例如防火墙），或者来给天真的使用者留下深刻印象（例如 PC 访问控制产品），谁对通用准则最为热心。而一位真正坚持的愤世嫉俗者也许会指出，自从苏联解体以来，政府评估机构通过经济间谍来证明它们的存在是正当的，而且通用准则签署国提供了大部分的有趣的目标。一个在世界范围销售的产品的错误的美国评估将危及到 2.5 亿美国人的安全；但是，它也将危及到 4 亿欧洲人和 1 亿日本人的安全，利益的平衡存在于欺骗当中。在诸如英国这样的小国当中，这种平衡更加强烈，这些国家需要保护的国民很少，而攻击的外国人很多。还有，政府评估机构获得的品行积分（和预算）是为了偷取的外国人的秘密，而不是为了外国人根本不想偷取的本地秘密。

然后，一名经济学家不会去相信通用准则的评估结果。也许，我是一名愤世嫉俗的人，但是我趋向于把它们看作是类似于橡胶拐杖的东西。这种设备具有许多用处，从获得法官的同情、从易受骗的政府部门骗取钱财直到疯狂地打击别人（只不过不要试图在它上面增加过

重的负担)!

幸运的是，在23.2.1节中讨论的经济学问题将限制标准被应用到某些领域中，但在这里，那些官方证书，虽然它们是不相关的、错误的或者是虚假的证书，却用来提供一些竞争的有利条件。

23.4 前面的路

Brook在其最著名的书《人月神话（The Mythical Man-Month)》中曾经提到一个引人注目的观点，即没有“银弹”来解决那些过时或者超过预算的软件项目当中的问题［140]。这个问题容易的部分，诸如开发高级语言，这种语言程序员们用起来要比汇编语言好很多，已经被完成了。清除大量编程过程中偶尔出现的复杂性意味着应用内在复杂性被留下了。我在第22章中关于系统开发方法论的叙述中讨论过这个话题；以上的讨论应该使你坚信，应用到保证和评估问题当中的方法是完全一样的。

对于评估和保证的一个更加现实的方法不仅要看产品的技术特征，还要看它在实际应用当中是如何行为的。易用性被通用准则所忽略，但是在真实世界中它是绝对重要的；当英国政府电子邮件系统使用者改变分割时需要重新启动个人电脑，这种做法令使用者很沮丧，以至于他们通过非正式的协议将所有东西都放到公共分割当中，这种做法实际上是浪费了高达9位数的投资（官方机密无疑将继续保护犯罪分子，使他们免于被惩罚)。我在第9章簿记系统中所描述的特征可以被设计用来限制由于人类品德上的弱点而导致的影响，这种特征是很关键的。在大多数的应用当中，我们必须假定人们总是粗心大意的，而且不具备相应的能力，或者甚至是不诚实的。

而且，还很有必要来面对这样一个事实，即大型且特征丰富的程序升级相当频繁。网络经济不能被期望一去不复返。评估和保证方案，例如通用准则、ISO 9001和CMM等，都试图将不断变动且具有竞争力的行业挤压成为官僚政治的紧身衣，从而提供购买者对于产品稳定性的幻想。如果这种稳定性确实存在，那么整个行业都将涌向它；但是人们所能够做到的最好的情况是涌向这些商标，例如20世纪70年代和80年代的IBM以及目前的微软。这些商标的创建和维护包含了巨大的市场约束力；安全只是一个小角色。

到目前为止，我也许已经给你提供了足够多的暗示，来告诉你如何欺骗系统以及如何将一个糟糕的产品冒充为一个安全的产品——至少要经历足够长的时间来让问题变成其他人的问题。在本书余下的部分中，我将假设你很诚实地努力保护系统，而且也希望降低风险，而不是降低应该付出的努力或者某些其他类型的可靠性。在许多情况中，当系统一旦出现故障，受损失的是系统拥有者；我已经引述了许多例子（核武器和控制、付费电视、预付费使用电表等等)，而且他们提供了一些更加有趣的工程事例。

当你真正想要控制一个保护特性时，在设计上遭受具有敌意的评论是必需的。如果这件事情在系统投入使用之前就完成，那么情况将会好一些。在所讨论的一个又一个历史案例当中，攻击者的动机几乎都是非常重要的；那些希望系统通过测试的善意评论与那些就是希望挑剔系统毛病的人所提出的评论比较而言，从本质上说，前者确实是毫无用处的。

23.4.1 半开放设计

做这件事的一种方法是从不同的顾问公司或者大学中雇佣多位专家。另一种方法是使用

多种不同的鉴定团体：我在 21.6.4 节中提到过，在美国，投票系统是如何在每个州中被独立审查的；以及在诸如 VISA 和 SWIFT 这样的组织实施标准之前，银行创建本地支付网络，这些网络中的每一个设计都由其自己的审计人员进行核查。虽然，还没有某种方法是确实可靠的，但是却已经出现了一些遗留下来的、很糟糕的投票和银行支付系统。

另一种已经建立起来的技术，我把它叫做半开放设计。这里，体系结构级的设计被公布，而实现方面的细节问题则无法知道。我曾经举过的例子包括 2.7.1 节中所讨论过的智能卡银行协议，以及第 11 章所提到的核武器和控制系统。

另一种用于半开放设计的方法是使用一个开放的可用软件包，任何人都能够对该包进行试验。当主要的威胁是一种合法攻击时，这种做法就具有相当大的价值了。通过法庭的判决来获得对于电子制表软件，例如 Excel，甚至是由中等规模的公司所提供的计费软件源代码的访问权是不切实际的想法。那些竞争对手的专家们将只能购买一份拷贝，体验一下，并且看看是否可以找出一些漏洞。对于如果以后这个包中出现某些 bug、或者是其他破坏证据特征的情况，你也只能看自己的运气了。

23.4.2　开放源代码

开放源代码是将现有体系结构的开放思想扩展到具体实现的细节层次上。许多安全产品都具有可以被公开使用的源代码，其中最为明显的要算是 PGP 电子邮件加密程序。Linux 操作系统和 Apache Web 服务器也都是开放源代码的，而且许多人依靠这些源代码来保护信息。另外，在政府部门中也是希望能够采用开放源代码的软件产品的。

开放源代码并不是最近才发明出来的新事物。在计算处理的早期岁月里，大多数的系统软件供应商都公开其源代码。这种开放性在 20 世纪 80 年代早期时逐渐衰退，因为在当时，尽管有来自其客户群中严厉的批评，法律上的压力还是迫使 IBM 对于其大型机软件采纳了一种“object-code-only”策略。现在情况似乎又要回到过去了。

在开放源代码的问题上面，有许多持支持意见的呼声，只有很少一部分持反对意见。支持者最主要的论据就是，如果世界上的每一个人都可以检查和使用软件，那么 bug 很容易就可以被发现和修复；Eric Raymond 有一句名言，“放到许多人眼前，bug 将很容易被发现”[634]。如果软件是通过一种合作的方式来维持的话，这种情况尤其如此，例如 Linux 和 Apache 就是这样做的。而且，通过这种方法，向产品中加入后门程序也是十分困难的事情了。

对开放源代码所持反对意见都集中在这样一个事实上面，即一旦软件变得庞大和复杂，那么很少有人或者根本没有动机来对源代码进行学习，因此主要的漏洞要花费很多年才能被发现。一个最近的例子就是在 PGP 版本 5 和版本 6 程序中的 bug，该 bug 可以让攻击者在不具备密钥持有者知识的情况下，可以增加一个额外的密钥契约 [690]，而且，该 bug 在产品中存在了许多年才被发现（问题也许是 PGP 被开发的速度比人们阅读其代码的速度要快；许多功能对于潜在的阅读源码的人来说并不感兴趣；或者目前产品还是商业产品，人们在没有任何报酬的情况下不会对产品进行测试工作）。

在 sendmail 这类产品中，后门“维持密码”在从产品中清除之前也经历了许多年的时间。大家所关心的是那些花费大量时间在公开代码中寻找 bug 或者可利用的特征的攻击者要比那些评论家所花费的时间还要多。实际上，情况可能比这还要糟糕；正如我们在 23.2.4

节中所提到的，不同的测试人员发现不同种类的 bug，因为他们测试的侧重点不同，所以，出现这种情况很有可能，即使该产品经历了 10 000 小时的共同检查，一家国外的情报机构仅仅通过 100 小时的调查也许就发现了一个新的可被利用的系统漏洞。给定一个被引用的可靠性增长模型，这种可能性很容易就可以被计算出来。

其他反对开放源代码的观点包括这样一种发现，就是在那些开放源代码的软件中，增加功能和特征的速度要比那些没有开放源代码的软件慢许多。后者可以开展一些肮脏特征的交互；最好在安全机制希望获得什么样的成果方面取得一致的共识；另外，也还存在许多特定的情况，例如当保护智能卡来抵制各种各样的攻击手段时、在一种内嵌入芯片中的私有加密算法能够强制攻击者在做反向工程时花费更多的努力的场合。

所以，利益平衡点在哪里呢？Eric Raymond 对于开放软件源码经济学具有影响力的分析建议一共有 5 个标准用于判断是否一个产品可能将从开放源代码 [635] 中获取好处。这 5 个标准是：在那些基于公共工程知识而不是私有技术的场合；在那些对于故障很敏感的场合；在那些需要为确认验证获取相应的评论的场合；在那些需要足够关键业务的场合中，使用者们合作来发现和清除 bug；还有在那些其经济需要包括很强网络影响的场合。安全通过了所有这些测试，Kerckhoffs 法则长期存在的本质就是密码系统应该通过以下方法设计，即当对手了解到系统所使用的技术后，仍然不会危急到该系统的安全 [454]。现在，像美国空军这类组织对于开发源代码产生了越来越浓厚的兴趣 [688，689]。

从而，我们可以很合理地得出这样的结论，即当开放源码设计既不是必需的又不是充分的情况时，通常它是会有所帮助的。重要的问题是那些有能力的人需要投入多少努力来检查和测试代码，而不管他们是否告诉你他们所发现的每一件事情。

23.4.3　Penetrate-and-Patch、CERT 和 bugtraq

刺穿和修补（Penetrate-and-Patch）是指在 20 世纪 70 年代到 80 年代期间，对于查找系统中的安全 bug，然后对它们进行修复这个发展过程的一种轻视的叫法；该叫法当时被广泛使用，其实这种叫法并不恰当，因为有更多的 bug 总是被发现。在当时，人们希望使用一些形式化的方法可以创造无 bug 的系统。这类系统已经被设计出来，但是规模过小而且在大部分的应用当中受限，保证中使用的重复方法又被重新使用，伴随这个问题而来的是如何管理它们。

很自然地，会出现一种竞争的环境。美国政府的期望是在诸如操作系统和通信软件这类公共基础产品中出现的漏洞应该首先向当局报告，以便于这些漏洞能够被法律执行过程所利用或者如果必要的话，还可以出于监控的目的而被利用。政府还期望制造商们应该在某些无授权用户使用该漏洞之后，才发布补丁程序。诸如微软这样的公司和从事开发攻击工具的情报机构部门以及那些在许多国家中由国防机构资助的计算机紧急事件响应小组（CERT）共享其源代码和漏洞数据。除此之外，许多人感到 CERT 的响应通常很慢。一种替代的办法就是当 bug 被发现时就公开，就像许多邮件列表中的著名的 bugtraq 一样。

没有一种方法是完全令人满意的。在第一种情况当中，你永远不会知道在你看到漏洞报告以前还有谁看到了该报告；而在第二种情况当中，你知道世界上的任何人都可以看到该报告，而且在补丁出台之前可以使用它们来对你进行攻击。也许，一个更加明智的解决办法是在 [631] 中所建议的方法，在这种方法中，bug 的发现者首先应该报告给软件的维护人。该

维护人将在 48 小时内确认收到了该报告，而不让该 bug 被公布出去；维护人将有 5 天的时间来实际对问题进行处理，这里面还可能通过相互间的协商而进行扩充。最后的 bug 修复应该对 bug 发现者给予信任。在本书出版之前，CERT 已经同意开始使用这种过程，对于供应商来说，设计和测试一个补丁程序仅仅需要 45 天的时间 [174]。

通过这种方法，软件公司都十分愿意来维护这种专门的和总是需要员工维护的 bug 报告设备，而且反过来也将获得足够的时间来在发布补丁前对其进行正确的测试；bug 发现者将获得信任并被写在 CV 上面；使用者将与发布 bug 报告相同的时间获得 bug 补丁；而且系统将很难被某些机构暗中破坏。

23.4.4 教育

也许作为一个理论界的人士，我是有些偏见的，但是我感到关于系统保护的问题和技术需要被更多的人所理解。我看到过一个又一个的实例，在它们当中，错误的机制被使用，或者正确的机制被错误地使用。通常，为了获得正确的保护，要进行 5 次或者 6 次的尝试才可以成功。通过一种见多识广的方法，也许可以变为 2 次或者 3 次。安全专业人士则很不幸，他们不是过于专业，而是把精力都集中在一些细小的方面；就是虽然什么都知道一些，但是却不理解许多更深入的技术问题。但是，抱怨我们对学生的培训问题是很容易的事情，困难的是找出对此我们可以做些什么。本书并不是第一步，而且也一定不是最后一步，但是我希望它可以对你有所帮助。

23.5 小结

有些时候，一个安全工程项目当中最困难的部分就是知道做到何种程度为止。许多评估和保证方法都对此有所帮助。在某种适当的情况下，这种帮助可能是十分巨大的，尤其是对那些刚刚开始创业的公司来说，它们的开发文化还不是很固定，而且正在寻求建立一个良好的工作习惯，从而打造其声誉。但是，它们所能够提供的帮助也有其局限性，而且过分使用因循守旧式的控制工具会有很大的危害。我认为它们有些像盐：在你的锅中加上少许就是一件很不错的事情，但是如果加入过多的盐就会出问题了。

但是，虽然情况有些叫人沮丧，但是也还没有达到必须放弃的程度。随着人们逐渐地获得某些经验，例如什么在起作用、什么遭受攻击和如何遭受攻击等，还有，随着保护需求和机制变为在职工程师们的专业技能中的一部分，事情将变得越来越好。15 年前的典型情况是，安全也许只能在第四次循环时才能够被搞清楚，但是这总还是比永远搞不清楚要好一些。

生活是无序和混乱的。成功意味着要和这种混乱的生活展开竞争。抱怨太多只能最终导致失败。

研究问题

我们需要一些新型的关于管理评估的思路和想法。也许，很可能将应用某些经济学家使用的工具来处理某些有缺陷的信息，这种信息包括从风险价格模型到公司理论很大的范围。如果我们使用一些统计工具来测量和预测故障，那么也将会是有些帮助的。

参考资料

整个行业都致力于提升保证和评估行业的水平，这些都是由大量的税款所保证的。它的热情甚至可以得到宗教信仰的支持和喜爱。不幸的是，还没有哪个地方有足够多的人来编写异教邪说。

结 束 语

我们正处在安全问题如何解决的变化中期。

10年前，大型公司的安全管理员通常是一名退伍军人或者警察，对于他们来说，“计算机安全”问题是一个相对不很重要的专业，可以留给计算机部门去解决，或者偶尔也需要外界专家们帮助解决。而在未来的10年当中，他的职位将被专业系统人员所占据；新的安全管理员将认为门锁和保安是一个相对不很重要的专业，所以可以将它们留给设备管理公司去管理，偶尔也听取一些外界专家们的意见。

10年前，安全技术包括的内容好比是由许多相互之间持怀疑态度的岛屿组成的群岛。密码研究者、操作系统保护人员、防盗报警业，直到那些在钞票上画出古怪图案的化学家们都被看作是岛屿。我们都认为世界最终在我们的掌握之下。在未来的10年当中，安全工程将成为一种已经成型的理论；岛屿间将通过桥梁被联系到一起，而且从业者需要对所有东西都十分熟悉。那些只知道钞票图案而不懂数字水印的人，以及那些只对通信机密机制感兴趣的密码研究者作为雇员来说不会有太大价值。

10年前，信息安全被认为是关于“保密性、完整性和可用性”的机制。在未来的10年当中，这个优先权列表将使用另外一种方式来描述了（正如其已经在许多应用中使用的那样）。安全工程将成为关于保证系统可以具有可事先预计的可靠性，处理所有的恶意攻击，尤其是那些拒绝服务攻击的一门学科。它们必须在错误和灾难面前表现出具有一定的弹性。所以，对人们粗心大意和能力不足的情况要可以容忍，而且其重要程度至少要等价于可以容忍不诚实行为的情况。这将意味着要投入更多的注意力来对待经济和制度上的问题，以及技术上的问题。真实系统提供这种依赖性的方法将比目前的方法更加形式多样：调节应用的安全性策略将成为工程师技术当中很重要的组成部分。

10年前，许多信息安全产品都是政府的专有。它们被那些受宠的增值防卫项目承包人秘密地设计并制造出来，而且制造数量很少。现在，商业上对信息安全产品的使用已经超过了政府使用的数量，在未来10年中，市场的巨大变化将被商业使用者完全的控制和接管。

10年前，面向信息安全的政府政策对于冷战时期大型通信情报网络的效率维护具有很大的贡献。它被秘密地运行，而且与核技术和导弹技术中的非盈利政策采取相同的路线：只有足够的政策才可以被用来防止国外的竞争制造商们的开发工作，而且在任何时候都可以通过审查终端用户和强制出口许可的方法来维持控制机制。目前，已经很明显的是，密码控制几乎和实际的政策需求无关。诸如数据保护、消费者保护甚至在线投票的问题都更加重要。在未来的10年当中，信息保护问题将在政府行为中变得十分普遍，这些政府行为包括税收到市场规划，而且在政策机构的要求之下，许多决定的执行是非常草率的，这些决定将不得不花费一些代价进行修改。

虽然，最大的挑战可能是系统集成和保证。10年前，在安全群岛中不同岛屿上的居民都对他们的产品具有很强的信心。密码研究者们相信，可靠的密码不可能被破译；智能卡制

造商们声称，从他们的芯片当中探测出密钥在物理上是绝对不可能的；而安全印刷人员说，全息图像在没有物理学博士的辅导以及价值2000万美元设备的支持下是不可能被伪造的。在系统一级，也存在许多不适当的自信。银行人士声称他们的自动柜员机绝对不可能出现一例错误。而多级安全操作系统在作为对系统保护问题的解决方案上的销售是供不应求。人们认定一个由发达国家政府许可的实验室所进行的安全评估既真实又具有可信度。这些原先可以很容易确定的事情都将消失。

许多事情都可以让工作变得更加复杂。在内部人士和外界人士之间的差别以前曾成为商务活动的中心，但是随着每样东西都被联系在一起，这种情况很快就消失了。保护过去由少数重大的思想和表述精确的建议所决定，而现在这种决定因素变得很分散，包括许多不精确的和启发式的知识。系统生命周期也在变化：以前，在有限的项目中一个封闭的系统被开发，而现在系统在进化和积累特征方面没有任何限制。在工作性质中的变化更加意义重大：以前的银行主要内部审计人员将记得前30年中的所有欺骗行为，从而防止数据处理部门重复那些错误的出现，但现在短暂雇佣和“永远革命”这样的新型企业文化却使审计人员的记忆变得无效。最后，还存在网络化信息系统的经济学问题：强大的客观因素要求产品投入市场的时间比质量还要重要。

所有这些变化对网络的影响是，对于计算机系统中的信息保护不再是一种科学准则，而是一种工程问题。

21世纪的安全工程师将有责任让系统不断向前发展，同时还要面对那些时时刻刻都在变化着的威胁手段。他们将拥有一个大型的、经常变化的工具箱。他们工作中的很重要的一部分将是跟上技术的发展：理解最新的攻击手段、学会使用新工具、跟上法律和政策的形势变化。像任何工程师一样，他们将需要一个扎实的理论基础；还将不得不理解某些机制的核心思想，例如密码技术、访问控制、信息流、网络和信号检测等。她还需要理解管理的基础问题：如何估计工作、客户的财政状况和商业处理的原理。但是，其中最重要的一点就是具备管理技术以及在开发系统来适应商务需求变化过程中能起到有效作用的能力。和商业人士交流的能力，而不仅仅是和其他工程师们交流，也是至关重要的；经验也将起到很大的作用。

我认为凡是不具有这些技能的人在未来某个时间将会被解雇，或者将会是令别人讨厌。

参考文献

我一直试图在本书的网页 http：//www.ross-anderson.com 处保留一些有用链接。但该网页中有的链接可能已经过期。

[1] M Abadi, "Explicit Communications Revisited: Two New Attacks on Authentication Protocols," in *IEEE Transactions on Software Engineering*, v 23 no 3 (Mar 1997), pp 185–186.

[2] M Abadi, RM Needham, "Prudent Engineering Practice for Cryptographic Protocols," *IEEE Transactions on Software Engineering*, v 22 no 1 (Jan 1996) pp 6–15; also as DEC SRC Research Report no 125 (June 1 1994) at `ftp://gatekeeper.pa.dec.com/pub/DEC/SRC/research-reports/SRC-125.pdf`.

[3] H Abelson, RJ Anderson, SM Bellovin, J Benaloh, M Blaze, W Diffie, J Gilmore, PG Neumann, RL Rivest, JI Schiller, B Schneier, "The Risks of Key Recovery, Key Escrow, and Trusted Third-Party Encryption," in *World Wide Web Journal*, v 2 no 3 (Summer 1997), pp 241–257.

[4] DG Abraham, GM Dolan, GP Double, JV Stevens, "Transaction Security System," in *IBM Systems Journal*, v 30 no 2 (1991), pp 206–229.

[5] A Abulafia, S Brown, S Abramovich-Bar, "A Fraudulent Case Involving Novel Ink Eradication Methods," in *Journal of Forensic Sciences* v 41 (1996), pp 300–302.

[6] N Achs, "VISA Confronts the Con Men," *Cards International* (Oct 20, 1992) pp 8–9.

[7] EN Adams, "Optimizing Preventive Maintenance of Software Products," *IBM Journal of Research & Development*, v 28 no 1 (1984), pp 2–14.

[8] J Adams, "Cars, Cholera, and Cows: The Management of Risk and Uncertainty," in *Policy Analysis*, no 335, Cato Institute, Washington (1999), at `http://www.cato.org/pubs/pas/pa-335es.html`.

[9] J Adams, *Risk*, University College London Press (1995), ISBN 1-85728-067-9.

[10] Y Adini, Y Moses, S Ullman, "Face recognition: The Problem of Compensating for Changes in Illumination Direction," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v 19 no 7 (July 1997), pp 721–732.

[11] C Ajluni, "Two New Imaging Techniques Promise to Improve IC Defect Identification," in *Electronic Design*, v 43 no 14 (July 10, 1995), pp 37–38.

[12] Y Akdeniz, "Regulation of Child Pornography on the Internet" (Dec 1999), at `http://www.cyber-rights.org/reports/child.htm`.

[13] Alliance to Outfox Phone Fraud, http://www.bell-atl.com/security/fraud/.

[14] American Society for Industrial Security, http://www.asisonline.org.

[15] E Amoroso, *Fundamentals of Computer Security Technology*, Englewood Cliffs, NJ: Prentice Hall (1994), ISBN 0-13-10829-3.

[16] J Anderson, *Computer Security Technology Planning Study*, ESD-TR-73-51, U.S. Air Force Electronic Systems Division (1973), http://csrc.nist.gov/publications/history/index.html.

[17] M Anderson, C North, J Griffin, R Milner, J Yesberg, K Yiu, "Starlight: Interactive Link," in *12th Annual Computer Security Applications Conference* (1996) proceedings, published by the IEEE, ISBN 0-8186-7606-XA, pp 55–63.

[18] RJ Anderson, "Solving a Class of Stream Ciphers," in *Cryptologia*, v XIV no 3 (July 1990), pp 285–288.

[19] RJ Anderson, "Why Cryptosystems Fail," in *Communications of the ACM*, v 37 no 11 (Nov 1994), pp 32–40; earlier version at http://www.cl.cam.ac.uk/users/rja14/wcf.html.

[20] RJ Anderson, "Liability and Computer Security: Nine Principles," in *Computer Security—ESORICS 94*, Springer LNCS, v 875, pp 231–245.

[21] RJ Anderson, "Crypto in Europe—Markets, Law, and Policy," in *Cryptography: Policy and Algorithms*, Springer LNCS, v 1029, pp 75–89.

[22] RJ Anderson, "Clinical System Security—Interim Guidelines," in *British Medical Journal*, v 312 no 7023 (Jan 13, 1996), pp 109–111; http://www.cl.cam.ac.uk/ftp/users/rja14/guidelines.txt.

[23] RJ Anderson, "Security in Clinical Information Systems," British Medical Association (1996), ISBN 0-7279-1048-5.

[24] RJ Anderson, "A Security Policy Model for Clinical Information Systems," in *Proceedings of the 1996 IEEE Symposium on Security and Privacy*, pp 30–43; http://www.cl.cam.ac.uk/users/rja14/policy11/policy11.html.

[25] RJ Anderson, "An Update on the BMA Security Policy," in [29], pp 233–250; http://www.cl.cam.ac.uk/ftp/users/rja14/bmaupdate.ps.gz.

[26] RJ Anderson, C Manifavas, C Sutherland, "NetCard—A Practical Electronic Cash Scheme," in *Security Protocols* (1996), Springer LNCS, v 1189, pp 49–57.

[27] RJ Anderson, "The Eternity Service," in *Proceedings of Pragocrypt 96* (GC UCMP, ISBN 80-01-01502-5), pp 242–252.

[28] RJ Anderson (ed), *Proceedings of the First International Workshop on Information Hiding* (1996), Springer LNCS, v 1174.

[29] RJ Anderson (ed), *Personal Medical Information—Security, Engineering and Ethics*, Springer-Verlag (1997), ISBN 3-540-63244-1.

[30] RJ Anderson, "GSM hack—Operator Flunks the Challenge," in comp.risks v 19.48: http://catless.ncl.ac.uk/Risks/19.48.html.

[31] RJ Anderson, "On the Security of Digital Tachographs," in *Computer Security—ESORICS 98*, Springer LNCS, v 1485, pp 111–125; http://www.cl.cam.ac.uk/ftp/users/rja14/tacho5.ps.gz.

[32] RJ Anderson, "Safety and Privacy in Clinical Information Systems," in *Rethinking IT and Health*, J Lenaghan (ed.), IPPR (Nov 1998), (ISBN 1-86030-077-4), pp 140–160.

[33] RJ Anderson, "The DeCODE Proposal for an Icelandic Health Database"; partly published in *Læknabladhidh* (the *Icelandic Medical Journal*), v 84 no 11 (Nov 1998), pp 874–875; full text available from http://www.cl.cam.ac.uk/users/rja14/#Med.

[34] RJ Anderson, "Healthcare Protection Profile—Comments," panel position paper at NISSC 1998; at http://www.cl.cam.ac.uk/ftp/users/rja14/healthpp.pdf.

[35] RJ Anderson, "The Formal Verification of a Payment System," chapter in *Industrial Strength Formal Methods: A Practitioner's Handbook*, MG Hinchey and JP Bowen (eds), Springer Verlag (Sept 1999, 1-85233-640-4), pp 43–52.

[36] RJ Anderson, "How to Cheat at the Lottery (or, Massively Parallel Requirements Engineering)," in *15th Annual Computer Security Application Conference* (1997); proceedings published by IEEE Computer Society, ISBN 0-7695-0346-2, pp xix–xxvii; at http://www.cl.cam.ac.uk/~rja14/lottery/lottery.html.

[37] RJ Anderson, "The Millennium Bug—Reasons Not to Panic," at http://www.ftp.cl.cam.ac.uk/ftp/users/rja14/y2k.html.

[38] RJ Anderson, "Comments on the Security Targets for the Icelandic Health Database," at http://www.cl.cam.ac.uk/ftp/users/rja14/iceland-admiral.pdf.

[39] RJ Anderson, SJ Bezuidenhoudt, "On the Reliability of Electronic Payment Systems," in *IEEE Transactions on Software Engineering*, v 22 no 5 (May 1996), pp 294–301; http://www.cl.cam.ac.uk/ftp/users/rja14/meters.ps.gz.

[40] RJ Anderson, E Biham, LR Knudsen,"Serpent: A Proposal for the Advanced Encryption Standard," submitted to NIST as an AES candidate; a short version of the paper appeared at the AES conference, August 1998; both papers available at [41].

[41] RJ Anderson, E Biham, L Knudsen, "The Serpent Home Page," http://www.cl.cam.ac.uk/~rja14/serpent.html.

[42] RJ Anderson, B Crispo, JH Lee, C Manifavas, V Matyás, FAP Petitcolas, *The Global Internet Trust Register*, MIT Press (1999), (ISBN 0-262-51105-3); http://www.cl.cam.ac.uk/Research/Security/Trust-Register/.

[43] RJ Anderson, MG Kuhn, "Tamper Resistance—A Cautionary Note," in

Proceedings of the Second Usenix Workshop on Electronic Commerce (Nov 1996), pp 1–11; http://www.cl.cam.ac.uk/users/rja14/tamper.html.

[44] RJ Anderson, MG Kuhn, "Low-Cost Attacks on Tamper-Resistant Devices," in *Security Protocols—Proceedings of the 5th International Workshop* (1997), Springer LNCS, v 1361, pp 125–136.

[45] RJ Anderson, MG Kuhn, "Soft Tempest—An Opportunity for NATO," at *Protecting NATO Information Systems in the 21st Century*, Washington, DC, Oct 25–26, 1999.

[46] RJ Anderson, JH Lee, "Jikzi: A New Framework for Secure Publishing," in *Security Protocols 99*, Springer LNCS, v 1976, pp 21–36.

[47] RJ Anderson, RM Needham, "Robustness Principles for Public Key Protocols," in *Advances in Cryptology—Crypto 95*, Springer LNCS, v 963, pp 236–247; http://www.cl.cam.ac.uk/ftp/users/rja14/robustness.ps.gz.

[48] RJ Anderson, RM Needham, "Programming Satan's Computer" in *Computer Science Today*, Springer, Lecture Notes in *Computer Science*, v 1000 (1995), pp 426–441; http://www.cl.cam.ac.uk/ftp/users/rja14/satan.ps.gz.

[49] RJ Anderson, RM Needham, A Shamir, "The Steganographic File System," in *Proceedings of the Second International Workshop on Information Hiding*, Springer LNCS, v 1525, pp 74–84.

[50] RJ Anderson, MR Roe, "The GCHQ Protocol and Its Problems," in *Advances in Cryptology—Eurocrypt 97*, Springer LNCS, v 1233, pp 134–148; http://www.cl.cam.ac.uk/ftp/users/rja14/euroclipper.ps.gz.

[51] CM Andrew, V Mitrokhin, *The Sword and the Shield: The Mitrokhin Archive and the Secret History of the KGB*, New York: Basic Books (1999), ISBN 0-46500310-9.

[52] http://www.anonymizer.com.

[53] JC Anselmo, "U.S. Seen More Vulnerable to Electromagnetic Attack," in *Aviation Week and Space Technology*, v 146 no 4 (July 28, 1997), p 67.

[54] T Appleby, "Chilling Debit-Card Scam Uncovered," in *The Globe & Mail* (Dec 12, 1999), p 1.

[55] U.S. Army, *Electromagnetic Pulse (EMP) and Tempest Protection for Facilities*, Hyattsville, Md: Corps of Engineers Publications Depot (1990).

[56] "ASPECT—Advanced Security for Personal Communications Technologies," at http://www.esat.kuleuven.ac.be/cosic/aspect/index.html.

[57] D Aubrey-Jones, "Internet—Virusnet?" in *Network Security* (Feb 1997), pp 15–19.

[58] D Aucsmith, "Tamper-Resistant Software: An Implementation," in [28], pp 317–333.

[59] D Aucsmith (ed), *Proceedings of the Second International Workshop on Information Hiding* (Portland, Oregon: Apr 1998), Springer LNCS, v 1525.

[60] B Audone, F Bresciani, "Signal Processing in Active Shielding and Direction-Finding Techniques," *IEEE Transactions on Electromagnetic Compatibility*, v 38 no 3 (Aug 1996), pp 334–340.

[61] D Austin, "Barclays Winning Card Fraud War," in *Banking Technology* (Apr 1994), p 5.

[62] D Austin, "Flood warnings," in *Banking Technology* (Jul–Aug 1999), pp 28–31.

[63] "Computer Combat Rules Frustrate the Pentagon," in *Aviation Week and Space Technology*, v 147 no 11 (Sept 9, 1997), pp 67–68.

[64] J Bacon, *Concurrent Systems*, Addison-Wesley (1997), ISBN 0-201-17767-6.

[65] J Bacon, K Moody, J Bates, R Hayton, CY Ma, A McNeil, O Seidel, M Spiteri, "Generic Support for Distributed Applications," in *IEEE Computer* (Mar 2000), pp 68–76.

[66] L Badger, DF Sterne, DL Sherman, KM Walker, SA Haghighat, "Practical Domain and Type Enforcement for UNIX," in *Proceedings of the 1995 IEEE Symposium on Security and Privacy*, pp 66–77.

[67] M Baggott, "The Smart Way to Fight Fraud," *Scottish Banker* (Nov 1995), pp 32–33.

[68] SA Baker, PR Hurst, *The Limits of Trust*, Kluwer Law International (1998), ISBN 9-0411-0639-1.

[69] D Balfanz, EW Felten, "Hand-Held Computers Can Be Better Smart Cards," in *Eighth USENIX Security Symposium* (1999), ISBN 1-880446-28-6, pp 15–23.

[70] J Bamford, *The Puzzle Palace: A Report on NSA, America's Most Secret Agency*, New York: Houghton-Mifflin (1982, 3rd printing, revised edition due out shortly), ISBN 0-395-31286-8.

[71] Bank for International Settlements, *Security and Reliability in Electronic Systems for Payments*, British Computer Society (1982).

[72] Bank for International Settlements, `http://www.bis.org/`.

[73] "Card Fraud: Banking's Boom Sector," in *Banking Automation Bulletin for Europe* (Mar 1992), pp 1–5.

[74] RL Barnard, *Intrusion Detection Systems*, Butterworths (1988), ISBN 0-409-90030-3.

[75] A Barnett, "Britain's UFO Secrets Revealed," in *The Observer* (June 4, 2000); at `http://www.observer.co.uk/uk_news/story/0,6903,328010,00.html`.

[76] J Barr, "The Gates of Hades," in *Linux World* (Apr 2000); at `http://www.linuxworld.com/linuxworld/lw-2000-04/lw-04-vcontrol_3.html`.

[77] R Baskerville, "Information Systems Security Design Methods: Implications for Information Systems Development," in *ACM Computing Surveys*, v 26 (1994), pp 375–414.

[78] PJ Bass, "Telephone Cards and Technology Development as Experienced by GPT Telephone Systems," in *GEC Review*, v 10 no 1 (1995), pp 14–19.

[79] "Great Microprocessors of the Past and Present," at `http://www.cs.uregina.ca/~bayko/cpu.html`.

[80] F Beck, *Integrated Circuit Failure Analysis—A Guide to Preparation Techniques*, New York: John Wiley & Sons, Inc. (1998), ISBN 0-471-97401-3.

[81] J Beck, "Sources of Error in Forensic Handwriting Examination," in *Journal of Forensic Sciences*, v 40 (1995), pp 78–87.

[82] HA Beker, C Amery, "Cryptography Policy," at `http://www.baltimore.com/library/whitepapers/mn_cryptography.html`.

[83] HJ Beker, JMK Friend, PW Halliden, "Simplifying Key Management in Electronic Fund Transfer Point-of-Sale Systems," in *Electronics Letters*, v 19 (1983), pp 442–443.

[84] H Beker, F Piper, *Cipher Systems*, Northwood (1982).

[85] H Beker, M Walker, "Key Management for Secure Electronic Funds Transfer in a Retail Environment," in *Advances in Cryptology—Crypto 84*, Springer LNCS, v 196, pp 401–410.

[86] DE Bell, L LaPadula, "Secure Computer Systems," ESD-TR-73-278, Mitre Corporation; v I and II (Nov 1973), v III (Apr 1974).

[87] M Bellare, J Kilian, P Rogaway, "The Security of Cipher Block Chaining," in *Advances in Cryptology—Crypto 94*, Springer LNCS, v 839, pp 341–358.

[88] M Bellare, P Rogaway, "Optimal Asymmetric Encryption," in *Advances in Cryptology—Eurocrypt 94*, Springer LNCS, v 950, pp 103–113; see also RFC 2437, `http://sunsite.auc.dk/RFC/rfc/rfc2437.html`.

[89] SM Bellovin, "Packets Found on an Internet," in *Computer Communications Review*, v 23 no 3 (July 1993), pp 26–31.

[90] SM Bellovin, "Defending against Sequence Number Attacks," RFC 1948 (May 1996); at `http://sunsite.auc.dk/RFC/rfc/rfc1948.html`.

[91] SM Bellovin, "Debit-Card Fraud in Canada," in `comp.risks`, v 20.69; at `http://catless.ncl.ac.uk/Risks/20.69.html`.

[92] SM Bellovin, "Permissive Action Links," at `http://www.research.att.com/~smb/nsam-160/pal.html`.

[93] SM Bellovin, "ICMP Traceback Messages," Internet draft (Mar 2000), at `http://search.ietf.org/internet-drafts/draft-bellovin-itrace-00.txt`.

[94] SM Bellovin, WR Cheswick, *Firewalls and Internet Security: Repelling the Wily Hacker*, Reading, MA: Addison-Wesley (1994), ISBN 0-201-63357-4.

[95] M Benantar, R Guski, KM Triodle, "Access Control Systems: From Host-Centric to Network-Centric Computing," in *IBM Systems Journal*, v 35 no 1 (1996), pp 94–112.

[96] W Bender, D Gruhl, N Morimoto, A Lu, "Techniques for Data Hiding," in *IBM Systems Journal*, v 35 no 3–4 (1996), pp 313–336.

[97] T Benkart, D Bitzer, "BFE Applicability to LAN Environments," in *Seventeenth National Computer Security Conference* (1994); Proceedings published by NIST, pp 227–236.

[98] F Bergadano, B Crispo, G Ruffo, "Proactive Password Checking with Decision Trees," in *4th ACM Conference on Computer and Communications Security* (1997); Proceedings published by the ACM, ISBN 0-89791-912-2, pp 67–77.

[99] T Berson, G Barksdale, "KSOS: Development Methodology for a Secure Operating System," *AFIPS Conference proceedings* (1979).

[100] K Biba, *Integrity Considerations for Secure Computer Systems*, Mitre Corporation MTR-3153 (1975).

[101] E Biham, A Biryukov, "Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials," in *Advances in Cryptology—Eurocrypt 97*, Springer LNCS, v 1592, pp 12–23.

[102] E Biham, A Shamir, *Differential Cryptanalysis of the Data Encryption Standard*, Springer (1993), ISBN 0-387-97930-1.

[103] E Biham, A Shamir, "Differential Fault Analysis of Secret Key Cryptosystems," in *Advances in Cryptology—Crypto 97*, Springer LNCS, v 1294, pp 513–525.

[104] A Biryukov, A Shamir, D Wagner, "Real-Time Cryptanalysis of A5/1 on a PC," in *Fast Software Encryption* (2000).

[105] Bishop and Bloomfield, "A Conservative Theory for Long-Term Reliability-Growth Prediction," in *IEEE Transactions on Reliability*, v 45 no 4 (Dec 1996), pp 550–560.

[106] DM Bishop, "Applying COMPUSEC to the Battlefield," in *17th Annual National Computer Security Conference* (1994), pp 318–326.

[107] M Bishop, M Dilger, "Checking for Race Conditions in File Accesses," in *Computing Systems USENIX*, v 9 no 2 (Spring 1996), pp 131–152.

[108] Wolfgang Bitzer, Joachim Opfer, "Schaltungsanordnung zum Messen der Korrelationsfunktion zwischen zwei vorgegebenen Signalen" [Circuit arrangement for measuring the correlation function between two provided signals]. German Patent DE 3911155 C2, Deutsches Patentamt, November 11, 1993.

[109] J Blackledge, "Making Money from Fractals and Chaos: Microbar," in *Mathematics Today*, v 35 no 6 (Dec 1999), pp 170–173.

[110] RD Blackledge, "DNA versus Fingerprints," in *Journal of Forensic Sciences*, v 40 (1995), p 534.

[111] GR Blakley, "Safeguarding Cryptographic Keys," in *Proceedings of NCC AFIPS* (1979), pp 313–317.

[112] B Blakley, R Blakley, RM Soley, *CORBA Security: An Introduction to Safe Computing with Objects*, Reading, MA: Addison-Wesley (1999), ISBN 0-201-32565-9.

[113] M Blaze, "Protocol Failure in the Escrowed Encryption Standard," in *Second ACM Conference on Computer and Communications Security* (Nov 2–4, 1994), Fairfax, VA: Proceedings published by the ACM ISBN 0-89791-732-4, pp 59–67; at `http://www.crypto.com/papers/`.

[114] M Blaze, SM Bellovin, "Tapping, Tapping on My Network Door," in *Communications of the ACM* (Oct 2000), Inside Risks 124; at `http://www.crypto.com/papers/carnivore-risks.html`.

[115] M Blaze, J Feigenbaum, J Lacy, "Decentralized Trust Management," in *Proceedings of the 1996 IEEE Symposium on Security and Privacy*, pp 164–173.

[116] D Bleichenbacher, "Chosen Ciphertext Attacks against Protocols Based on the RSA Encryption Standard PKCS #1," in *Advances in Cryptology—Crypto 98*, Springer LNCS, v 1462, pp 1–12.

[117] G Bleumer, M Schunter, "Digital Patient Assistants: Privacy vs Cost in Compulsory Health Insurance," in *Health Informatics Journal*, v 4 nos 3–4 (Dec 1998), pp 138–156.

[118] B Blobel, "Clinical Record Systems in Oncology. Experiences and Developments on Cancer Registers in Eastern Germany," in [29], pp 39–56.

[119] JA Bloom, IJ Cox, T Kalker, JPMG Linnartz, ML Miller, CBS Traw, "Copy Protection for DVD Video," in *Proceedings of the IEEE*, v 87 no 7 (July 1999), pp 1267–1276.

[120] ER Block, *Fingerprinting*, Franklin Wells (1970), SBN 85166-435-0.

[121] S Blythe, B Fraboni, S Lall, H Ahmed, U de Riu, "Layout Reconstruction of Complex Silicon Chips," in *IEEE Journal of Solid-State Circuits*, v 28 no 2 (Feb 1993), pp 138–145.

[122] WE Boebert, RY Kain, "A Practical Alternative to Hierarchical Integrity Policies," in *8th National Computer Security Conference* (1985), Proceedings published by NIST, p 18.

[123] BW Boehm, *Software Engineering Economics*, Englewood Cliffs, NJ: Prentice Hall (1981), ISBN 0-13-822122-7.

[124] N Bohm, I Brown, B Gladman, "Electronic Commerce—Who Carries the Risk of Fraud?" *Journal of Information Law & Technology*, v 3 (2000); `http://elj.warwick.ac.uk/jilt/00-3/bokm.html`.

[125] MK Bond, "Attacks on Cryptoprocessor Transaction Sets," *to be submitted to CHES 2001*.

[126] D Boneh, RA Demillo, RJ Lipton, "On the Importance of Checking Cryptographic

Protocols for Faults," in *Advances in Cryptology—Eurocrypt 97*, Springer LNCS, v 1233, pp 37–51.

[127] L Boney, AH Tewfik, KN Hamdy, "Digital Watermarks for Audio Signals," in *Proceedings of the 1996 IEEE International Conference on Multimedia Computing and Systems*, pp 473–480.

[128] V Bontchev, "Possible Macro Virus Attacks and How to Prevent Them," in *Computers and Security*, v 15 no 7 (1996), pp 595–626.

[129] NS Borenstein, "Perils and Pitfalls of Practical Cybercommerce," in *Communications of the ACM*, v 39 no 6 (June 1996), pp 36–44.

[130] E Bovenlander, talk on smartcard security, Eurocrypt 97, reported in [44].

[131] E Bovenlander, RL van Renesse, "Smartcards and Biometrics: An Overview," in *Computer Fraud and Security Bulletin* (Dec 1995), pp 8–12.

[132] C Bowden, Y Akdeniz, "Cryptography and Democracy: Dilemmas of Freedom," in *Liberating Cyberspace: Civil Liberties, Human Rights, and the Internet*, Pluto Press (1999), pp 81–125.

[133] RM Brady, RJ Anderson, RC Ball, *Murphy's Law, the Fitness of Evolving Species, and the Limits of Software Reliability*, Cambridge University Computer Laboratory Technical Report no. 471 (1999).

[134] S Brands, *Rethinking Public Key Infrastructures and Digital Certificates—Building in Privacy*, MIT Press (2000), ISBN 0-262-02491-8.

[135] JT Brassil, S Low, NF Maxemchuk, "Copyright Protection for the Electronic Distribution of Text Documents," in *Proceedings of the IEEE*, v 87 no 7 (July 1999), pp 1181–1196.

[136] M Brelis, "Patients' Files Allegedly Used for Obscene Calls," in *Boston Globe*, (Apr 11, 1995); also in `comp.risks`, v 17 no 7.

[137] DFC Brewer, MJ Nash, "Chinese Wall Model," in *Proceedings of the 1989 IEEE Computer Society Symposium on Security and Privacy*, pp 215–228.

[138] M Briceno, I Goldberg, D Wagner, "An Implementation of the GSM A3A8 Algorithm," at `http://www.scard.org/gsm/a3a8.txt`.

[139] D Brin, *The Transparent Society: Will Technology Force Us to Choose between Privacy and Freedom?* Perseus Press (1999), ISBN 0-73820144-8.

[140] F Brooks, *The Mythical Man-Month: Essays on Software Engineering*, Addison-Wesley (1995), ISBN 0-201-83595-9.

[141] D Brown, "Techniques for Privacy and Authentication in Personal Communications Systems," in *IEEE Personal Communications*, v 2 no 4 (Aug 1995), pp 6–10.

[142] R Buderi, *The Invention That Changed the World*, Simon & Schuster, New York, (1996); ISBN 0-684-81021-2.

[143] H Buehler, interview with Swiss Radio International, (July 4, 1994); at `http://`

www.funet.fi/pub/crypt/mirrors/idea.sec.dsi.unimi.it/rpub.cl.msu.edu/crypt/docs/hans-buehler-crypto-spy.txt.

[144] http://archives.neohapsis.com/archives/bugtraq/.

[145] Bundesamt für Sicherheit in der Informationstechnik (German Information Security Agency), "Schutzmaßnahmen gegen Lauschangriffe" [Protection against bugs], *Faltblätter des BSI*, v 5, Bonn (1997); http://www.bsi.bund.de/literat/faltbl/laus005.htm.

[146] J Bunnell, J Podd, R Henderson, R Napier, J Kennedy-Moffatt, "Cognitive, Associative and Conventional Passwords: Recall and Guessing Rates," in *Computers and Security*, v 16 no 7 (1997), pp 645–657.

[147] Buro Jansen & Janssen, "Making Up the Rules: Interception versus Privacy," (Aug 8, 2000), at http://www.xs4all.nl/~respub/crypto/english/.

[148] M Burrows, M Abadi, RM Needham, "A Logic of Authentication," in *Proceedings of the Royal Society of London A*, v 426 (1989), pp 233–271; earlier version published as DEC SRC Research Report 39, ftp://gatekeeper.pa.dec.com/pub/DEC/SRC/research-reports/SRC-039.pdf.

[149] C Busch, F Graf, S Wolthusen, A Zeidler, "A System for Intellectual Property Protection," Fraunhofer Institute, at http://www.igd.fhg.de/igd-a8.

[150] RW Butler, GB Finelli, "The Infeasibility of Experimental Quantification of Life-Critical Software Reliability," in *ACM Symposium on Software for Critical Systems* (1991), ISBN 0-89791-455-4, pp 66–76.

[151] "Long Distance Phone Scam Hits Internet Surfers," in businessknowhow.com, at http://www.businessknowhow.com/newlong.htm.

[152] California Secretary of State, "A Report on the Feasibility of Internet Voting" (Jan 2000), at http://www.ss.ca.gov/executive/ivote/.

[153] J Calvert, P Warren, "Secrets of McCartney Bank Cash Are Leaked," in *The Express* (Feb 9, 2000), pp 1–2.

[154] JL Cambier, "Biometric Identification in Large Population," in *Information Security Bulletin*, v 5 no 2 (Mar 2000), pp 17–26.

[155] J Camenisch, JM Piveteau, M Stadler, "An Efficient Fair Payment System," in *3rd ACM Conference on Computer and Communications Security* (1996); Proceedings published by the ACM, ISBN 0-89791-829-0, pp 88–94.

[156] LJ Camp, C Wolfram, "Pricing Security," Third Information Survivability Workshop, Boston, (Oct 2000).

[157] D Campbell, "Somebody's Listening," in *The New Statesman* (Aug 12, 1988), pp 1, 10–12; at http://jya.com/echelon-dc.htm.

[158] D Campbell, "Making History: The Original Source for the 1988 First Echelon Report Steps Forward" (Feb 25, 2000); at http://cryptome.org/echelon-mndc.htm.

[159] JC Campbell, N Ikegami, *The Art of Balance in Health Policy—Maintaining*

Japan's Low-Cost, Egalitarian System, Cambridge University Press (1998), ISBN 0-521-57122-7.

[160] D Campbell, P Lashmar, "The New Cold War: How America Spies on Us for Its Oldest Friend—the Dollar," in *The Independent* (July 2, 2000); at `http://www.independent.co.uk/news/World/Americas/2000-07/coldwar020700.shtml`.

[161] JP Campbell, "Speaker Recognition: A Tutorial," in *Proceedings of the IEEE*, v 85 no 9 (Sept 1997), pp 1437–1462.

[162] C Cant, S Wiseman, "Simple Assured Bastion Hosts," in *13th Annual Computer Security Application Conference* (1997); Proceedings published by IEEE Computer Society, ISBN 0-8186-8274-4ACSAC, pp 24–33.

[163] "Dark Horse in Lead for Fingerprint ID Card," *Card World Independent* (May 1994), p 2.

[164] "German A555 Takes Its Toll," in *Card World International* (Dec 1994–Jan 1995), p 6.

[165] "High Tech Helps Card Fraud Decline," in *Cards International*, no 117 (Sept 29, 1994).

[166] "VISA Beefs Up Its Anti-Fraud Technology," in *Cards International*, no 189 (Dec 12, 1997), p 5.

[167] JM Carlin, "UNIX Security Update," at *USENIX Security 93*, pp 119–130.

[168] J Carr, "Doing Nothing Is Just Not an Option," in *The Observer* (June 18, 2000), at `http://www.fipr.org/rip/index.html`.

[169] J Carroll, *Big Blues: The Unmaking of IBM*, New York: Crown Publishers (1993), ISBN 0-517-59197-9.

[170] H Carter, "Car Clock Fixer Jailed for Nine Months," in *The Guardian* (Feb 15, 2000), p 13.

[171] R Carter, "What You Are ... Not What You Have," in *International Security Review Access Control*, Special Issue (Winter 1993–1994), pp 14–16.

[172] S Castano, M Fugini, G Martella, P Samarati, *Database Security*, Reading, MA: Addison-Wesley (1994), ISBN 0-201-59375-0.

[173] Center for Democracy and Technology, `http://www.cdt.org/`.

[174] 'The CERT Coordination Center Vulnerability Disclosure Policy;' http://www.cert.org/faq/vuldisclosurepolicy.html

[175] DW Chadwick, PJ Crook, AJ Young, DM McDowell, TL Dornan, JP New, "Using the Internet to Access Confidential Patient Records: A Case Study," in *British Medical Journal*, v 321 (Sep 9, 2000), pp 612–614; at `http://bmj.com/cgi/content/full/321/7261/612`.

[176] L Chapman, *Your Disobedient Servant*, New York: Penguin Books (1979).

[177] D Chaum, "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms," in *Communications of the ACM*, v 24 no 2 (Feb 1981).

[178] D Chaum, "Blind Signatures for Untraceable Payments," in *Crypto 82*, Plenum Press (1983), pp 199–203.

[179] D Chaum, "The Dining Cryptographers' Problem: Unconditional Sender and Recipient Untraceability," in *Journal of Cryptology*, v 1 (1989) pp 65–75.

[180] D Chaum, A Fiat, M Naor, "Untraceable Electronic Cash," in *Advances in Cryptology—CRYPTO '88*, Springer LNCS, v 403, pp 319–327.

[181] R Chellappa, CL Wilcon, S Sirohey, "Human and Machine Recognition of Faces: A Survey," in *Proceedings of the IEEE*, v 83 no 5 (May 1995), pp 705–740.

[182] HJ Choi, private discussion with author.

[183] B Christianson, et al. (ed), "Security Protocols—5th International Workshop," Springer LNCS, v 1360 (1998).

[184] B Christianson, et al. (ed), "Security Protocols—6th International Workshop," Springer LNCS, v 1550 (1999).

[185] F Church (chairman), "Intelligence Activities—Senate Resolution 21," U.S. Senate, 94th Congress, First Session, at `http://cryptome.org/nsa-4th.htm`.

[186] WS Ciciora, "Inside the Set-Top Box," in *IEEE Spectrum*, v 12 no 4 (Apr 1995), pp 70–75.

[187] D Clark, D Wilson, "A Comparison of Commercial and Military Computer Security Policies," in *Proceedings of the 1987 IEEE Symposium on Security and Privacy*, pp 184–194.

[188] R Clark, *The Man Who Broke Purple*, New York, Little Brown (1977), ISBN 0-316-14595-5.

[189] I Clarke, "The Free Network Project Homepage," at `http://freenet.sourceforge.net/`.

[190] R Clayton, G Davies, C Hall, A Hilborne, K Hartnett, D Jones, P Mansfield, K Mitchell, R Payne, N Titley, D Williams, "LINX Best Current Practice—Traceability," Version 1.0 (May 18, 1999), at `http://www.linx.net/noncore/bcp/traceability-bcp.html`.

[191] S Clough, "Bombings 'Inspired by Atlanta Attack,'" in *Daily Telegraph* (June 6, 2000); at `http://www.telegraph.co.uk:80/`.

[192] FB Cohen, *A Short Course on Computer Viruses*, New York: John Wiley & Sons, Inc. (1994), ISBN 0-471-00769-2.

[193] JL Colbert, PL Bowen, "A Comparison of Internal Controls: COBIT, SAC, COSO and SAS 55/78," at `http://www.isaca.org/bkr_cbt3.htm`.

[194] A Collins, "Court Decides Software Time-Locks Are Illegal," in *Computer Weekly* (Aug 19, 1993), p 1.

[195] D Comer, "Cryptographic Techniques—Secure Your Wireless Designs," in *EDN* (Jan 18, 1996), pp 57–68.

[196] Committee of Sponsoring Organizations of the Treadway Commission (CSOTC), "Internal Control—Integrated Framework" (COSO Report, 1992); from `http://www.coso.org/`.

[197] "Communicating Britain's Future," at http://www.fipr.org/polarch/labour.html.

[198] "Kavkaz-Tsentr Says Russians Hacking Chechen Web Sites"; " 'Information War' Waged on Web Sites over Chechnya," in *Communications Law in Transition* Newsletter, v 1 no 4 (Feb 2000), at http://pcmlp.socleg.ox.ac.uk/transition/issue04/russia.htm.

[199] Computer Emergency Response Team Coordination Center, at http://www.cert.org/.

[200] "Telecoms Fraud in the Cellular Market: How Much Is Hype and How Much Is Real?" in *Computer Fraud and Security Bulletin* (June 1997), pp 11–14.

[201] Computer Privacy Digest, v 17 no 7 (Sept 15, 2000).

[202] JB Condat, "Toll Fraud on French PBX Systems," in *Computer Law and Security Report*, v 10 no 2 (Mar/Apr 1994), pp 89–91.

[203] J Connolly, "Operation Chain Link: The Deployment of a Firewall at Hanscom Air Force Base," *Twelfth Annual Computer Security Applications Conference* (1996); Proceedings published by the IEEE, ISBN 0-8186-7606-X, pp 170–177.

[204] E Constable, "American Express to Reduce the Risk of Online Fraud".

[205] D Coppersmith, "The Data Encryption Standard (DES) and Its Strength against Attacks," IBM report RC 18613 (81421).

[206] Council of Europe, "Convention for the Protection of Individuals with Regard to Automatic Processing of Personal Data," European Treaty Series, no. 108 (Jan 28, 1981); at http://www.privacy.org/pi/intl_orgs/coe/dp_convention_108.txt.

[207] C Cowan, C Pu, D Maier, H Hinton, J Walpole, P Bakke, S Beattie, A Grier, P Wagle, Q Zhang, "StackGuard: Automatic Adaptive Detection and Prevention of Buffer-Overflow Attacks," *7th USENIX Security Conference* (1998), pp 63–77.

[208] JW Coyne, NC Kluksdahl, " 'Mainstreaming' Automated Information Systems Security Engineering (A Case Study in Security Run Amok)," in *Second ACM Conference on Computer and Communications Security* (1994); Proceedings published by the ACM, ISBN 0-89791-732-4, pp 251–257; at http://www.acm.org/pubs/contents/proceedings/commsec/191177/.

[209] L Cranor, "Lorrie Cranor's Electronic Voting Hot List," at http://www.ccrc.wustl.edu/~lorracks/sensus/hotlist.html.

[210] S Craver, "On Public-Key Steganography in the Presence of an Active Warden," in *Proceedings of the Second International Workshop on Information Hiding* (1998), Springer LNCS, v 1525, pp 355–368.

[211] B Crispo, M Lomas, "A Certification Scheme for Electronic Commerce," in *Security Protocols* (1996), Springer LNCS, v 1189, pp 19–32.

[212] W Curtis, H Krasner, N Iscoe, "A Field Study of the Software Design Process for Large Systems," in *Communications of the ACM*, v 31 no 11 (Nov 1988), pp 1268–1287.

[213] J Daemen, L Knudsen, V Rijmen, "The Block Cipher SQUARE," in *Fourth*

International Workshop on Fast Software Encryption, Springer LNCS, v 1267 (1997), pp 149–165; at http://www.esat.kuleuven.ac.be/~rijmen/square/.

[214] "Beating the Credit Card Telephone Fraudsters," in *Daily Telegraph* (Oct 9, 1999), at http://www.telegraph.co.uk:80/.

[215] T Dalrymple, "The Sinister Ethos of the Baying Mob," in *The Sunday Telegraph* (Aug 13, 2000), at http://www.dailytelegraph.co.uk.

[216] M Darman, E le Roux, "A New Generation of Terrestrial and Satellite Microwave Communication Products for Military Networks," in *Electrical Communication* (Q4 1994), pp 359–364.

[217] Data Protection Commissioners of EU and EES countries and Switzerland, two statements, *20th International Conference on Data Protection*, Santiago de Compostela, (Sept 16–18, 1998); at http://www.dataprotection.gov.uk/20dpcom.html.

[218] J Daugman, "High Confidence Visual Recognition of Persons by a Test of Statistical Independence," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v 15 no 11 (Nov 1993), pp 1148–1161.

[219] J Daugman, "Biometric Decision Landscapes," Technical Report No. TR482, University of Cambridge Computer Laboratory.

[220] C Davies, R Ganesan, "BApasswd: A New Proactive Password Checker," in *16th National Computer Security Conference* (1993); proceedings published by NIST, pp 1–15.

[221] DW Davies, WL Price, *Security for Computer Networks*, New York: John Wiley & Sons, Inc. (1984).

[222] G Davies, *A History of Money from Ancient Times to the Present Day*, University of Wales Press (1996); ISBN 0-7083-1351-5; related material at http://www.ex.ac.uk/%7ERDavies/arian/llyfr.html.

[223] H Davies, "Physiognomic Access Control," in *Information Security Monitor*, v 10 no 3 (Feb 1995), pp 5–8.

[224] D Davis, "Compliance Defects in Public-Key Cryptography," in *Sixth USENIX Security Symposium Proceedings* (July 1996), pp 171–178.

[225] D Davis, R Ihaka, P Fenstermacher, "Cryptographic Randomness from Air Turbulence in Disk Drives," in *Advances in Cryptology—Crypto 94*, Springer LNCS, v 839, pp 114–120.

[226] D Dean, EW Felten, DS Wallach, "Java Security: From HotJava to Netscape and Beyond," in *Proceedings of the 1996 IEEE Symposium on Security and Privacy*, IEEE Computer Society Press, pp 190–200.

[227] C Deavours, D Kahn, L Kruh, G Mellen, B Winkel, *Cryptology—Yesterday, Today, and Tomorrow*, Artech House (1987), ISBN 0-89006-253-6.

[228] C Deavours, D Kahn, L Kruh, G Mellen, B Winkel, *Selections from Cryptologia—History, People and Technology*, Artech House (1997), ISBN 0-89006-862-3.

[229] C Deavours, L Kruh, *Machine Cryptography and Modern Cryptanalysis*, Artech House (1985), ISBN 0-89006-161-0.

[230] B Demoulin, L Kone, C Poudroux, P Degauque, "Electromagnetic Radiation of Shielded Data Transmission Lines," in [301], pp 163–173.

[231] I Denley, S Weston-Smith, "Implementing Access Control to Protect the Confidentiality of Patient Information in Clinical Information Systems in the Acute Hospital," in *Health Informatics Journal*, v 4 nos 3–4 (Dec 1998), pp 174–178.

[232] I Denley, S Weston-Smith, "Privacy in Clinical Information Systems in Secondary Care," in *British Medical Journal*, v 318 (May 15, 1999), pp 1328–1331.

[233] DE Denning, "The Lattice Model of Secure Information Flow," in *Communications of the ACM*, v 19 no 5, pp 236–243.

[234] DE Denning, *Cryptography and Data Security*, Addison-Wesley (1982), ISBN 0-201-10150-5.

[235] DE Denning, *Information Warfare and Security*, Addison-Wesley (1999), ISBN 0-201-43303-6.

[236] DE Denning, "Activism, Hacktivism, and Cyberterrorism: The Internet as a Tool for Influencing Foreign Policy," InfowarCon 2000, at `http://www.nautilus.org/info-policy/workshop/papers/denning.html`.

[237] DE Denning, PH MacDoran, "Location-Based Authentication: Grounding Cyberspace for Better Security," in *Computer Fraud and Security Bulletin* (Feb 1996), pp 12–16.

[238] DE Denning, J Schlorer, "Inference Controls for Statistical Databases," in *IEEE Computer*, v 16 no 7 (July 1983), pp 69–82.

[239] DE Denning, *Information Warfare and Security*, Readings, MA: Addison Wesley (1998), ISBN 0-201-43303-6.

[240] Department of Defense, "Department of Defense Trusted Computer System Evaluation Criteria," DoD 5200.28-STD (Dec 1985).

[241] Department of Defense, "A Guide to Understanding Covert Channel Analysis of Trusted Systems," NCSC-TG-030 (Nov 1993).

[242] Department of Defense, "Password Management Guideline," CSC-STD-002-85 (1985).

[243] Department of Defense, "A Guide to Understanding Data Remanence in Automated Information Systems," NCSC-TG-025 (1991).

[244] Department of Defense, "Technical Rationale behind CSC-STD-003-85: Computer Security Requirements," CSC-STD-004-85 (1985).

[245] Department of Justice, "Guidelines for Searching and Seizing Computers" (1994); at `http://www.epic.org/security/computer_search_guidelines.txt`.

[246] Y Desmedt, Y Frankel, "Threshold Cryptosystems," in *Advances in Cryptology—Proceedings of Crypto 89*, Springer LNCS, v 435, pp 307–315.

[247] J Dethloff, "Special Report: Intellectual Property Rights and Smart Card Patents: The Past, the Present, the Future," in *Smart Card News* (Feb 1996), pp 36–38.

[248] W Diffie, ME Hellman, "New Directions in Cryptography," in *IEEE Transactions on Information Theory*, v 22 no 6 (Nov 1976), pp 644–654.

[249] W Diffie, ME Hellman, "Exhaustive Cryptanalysis of the NBS Data Encryption Standard," in *Computer*, v 10 no 6 (June 1977), pp 74–84.

[250] W Diffie, S Landau, *Privacy on the Line—The Politics of Wiretapping and Encryption*, MIT Press (1998), ISBN 0-262-04167-7.

[251] E Dijkstra, "Solution of a Problem in Concurrent Programming Control," in *Communications of the ACM*, v 8 no 9 (1965), p 569.

[252] The Discount Long Distance Digest, at `http://www.thedigest.com/shame/`.

[253] D Dittrich, "Distributed Denial of Service (DDoS) Attacks/Tools," at `http://staff.washington.edu/dittrich/misc/ddos/`; see also `http://www.washington.edu/People/dad/`.

[254] RC Dixon, *Spread Spectrum Systems with Commercial Applications*, New York: John Wiley & Sons, Inc. (1994), ISBN 0-471-59342-7.

[255] H Dobbertin, "Cryptanalysis of MD4," *Journal of Cryptology*, v 11 no 4 (1998), pp 253–270.

[256] B Dole, S Lodin, E Spafford, "Misplaced Trust: Kerberos 4 Session Keys," in *Internet Society Symposium on Network and Distributed System Security*; proceedings published by the IEEE, ISBN 0-8186-7767-8, pp 60–70.

[257] "Dotcom Executives 'More Likely to Have Dark Pasts'," C Daniel, *Financial Times*, (Oct 23, 2000); `http://www.ft.com`.

[258] I Drury, "Pointing the Finger," in *Security Surveyor*, v 27 no 5 (Jan 1997), pp 15–17.

[259] Wim van Eck, "Electromagnetic Radiation from Video Display Units: An Eavesdropping Risk?" in *Computers & Security*, v 4 (1985), pp 269–286.

[260] *The Economist*, "Digital Rights and Wrongs" (July 17, 1999); see `www.economist.com`.

[261] *The Economist*, "Living in the Global Goldfish Bowl," (Dec 18–24, 1999), Christmas special; see `www.economist.com`.

[262] A Edwards, "BOLERO, a TTP project for the Shipping Industry," in *Information Security Technical Report*, v 1 no 1 (1996), pp 40–45.

[263] M Eichin, J Rochlis, "With Microscope and Tweezers: An Analysis of the Internet Virus of November 1988," in *Proceedings of the 1989 IEEE Symposium on Security and Privacy*, pp 326–343.

[264] Electronic Frontier Foundation, `http://www.eff.org`.

[265] Electronic Frontier Foundation, *Cracking DES: Secrets of Encryption Research, Wiretap Politics, and Chip Design* EFF (1998); ISBN 1-56592-520-3; at `http://cryptome.org/cracking-des.htm`.

[266] Electronic Privacy Information Center, http://www.epic.org.

[267] JH Ellis, *The History of Non-Secret Encryption*, at http://www.cesg.gov.uk/about/nsecret/ellis.htm.

[268] C Ellison, B Schneier, "Ten Risks of PKI: What You're Not Being Told about Public Key Infrastructure," in *Computer Security Journal*, v XIII no 1 (2000); also at http://www.counterpane.com/pki-risks.html.

[269] *Enfopol Papiere*, Telepolis archiv special (1998/1999), at http://www.heise.de/tp/deutsch/special/enfo/default.html.

[270] P Enge, T Walter, S Pullen, CD Kee, YC Chao, YJ Tsai, "Wide Area Augmentation of the Global Positioning System," in *Proceedings of the IEEE*, v 84 no 8 (Aug 1996), pp 1063–1088.

[271] EPIC, "Approvals for Federal Pen Registers and Trap and Trace Devices 1987–1998," at http://www.epic.org/privacy/wiretap/stats/penreg.html.

[272] EPIC, "Report of the Director of the Administrative Office of the United States Courts," at http://www.epic.org/privacy/wiretap/stats/1999-report/wiretap99.pdf.

[273] J Epstein, H Orman, J McHugh, R Pascale, M Branstad, A Marmor-Squires, "A High-Assurance Window System Prototype," in *Journal of Computer Security*, v 2 no 2–3 (1993), pp 159–190.

[274] J Epstein, R Pascale, "User Interface for a High-Assurance Windowing System," in *9th Annual Computer Security Applications Conference* (1993); proceedings published by the IEEE, ISBN 0-8186-4330-7, pp 256–264.

[275] T Escamilla, *Intrusion Detection—Network Security beyond the Firewall*, New York: John Wiley & Sons, Inc. (1998), ISBN 0-471-29000-9.

[276] J Essinger, *ATM Networks—Their Organization, Security, and Future*, Elsevier (1987).

[277] A Etzioni, *The Limits of Privacy*, New York: Basic Books (1999), ISBN 0-465-04089-6.

[278] European Parliament, "Development of Surveillance Technology and Risk of Abuse of Economic Information," Luxembourg (Apr 1999), PE 166.184/Part3/4, at http://www.gn.apc.org/duncan/stoa.htm.

[279] European Union, "Directive on the Protection of Individuals with Regard to the Processing of Personal Data and on the Free Movement of Such Data," Directive 95/46/EC, at http://www.privacy.org/pi/intl_orgs/ec/eudp.html.

[280] European Union, "Draft Council Resolution on the Lawful Interception of Telecommunications in Relation to New Technologies" 6715/99 (Mar 15, 1999), at http://www.fipr.org/polarch/enfopol19.html/; for background, see http://www.fipr.org/polarch/.

[281] G Faden, "Reconciling CMW Requirements with Those of X11 Applications," in *Proceedings of the 14th Annual National Computer Security Conference* (1991).

[282] M Fairhurst, "The Hedge End Experiment," in *International Security Review*, no 85 (Summer 1994), p 20.

[283] M Fairhurst, "Signature Verification Revisited: Promoting Practical Exploitation of Biometric Technology," in *Electronics and Communication Engineering Journal*, v 9 no 6 (Dec 1997), pp 273–280.

[284] *Federal Trade Commission v Audiotex Connection, Inc.*, and others, at `http://www.ftc.gov/os/1997/9711/Adtxamdfcmp.htm`.

[285] Federal Trade Commission, "ID Theft: When Bad Things Happen to Your Good Name," at `http://www.consumer.gov/idtheft/`.

[286] Federation of American Scientists, `http://www.fas.org`.

[287] H Federrath, J Thees, "Schutz der Vertraulichkeit des Aufenthaltsorts von Mobilfunkteilnehmern," in *Datenschutz und Datensicherheit* (June 1995), pp 338–348.

[288] P Fellwock (using pseudonym Winslow Peck), "U.S. Electronic Espionage: A Memoir," in *Ramparts*, v 11 no 2 (Aug 1972), pp 35–50; at `http://jya.com/nsa-elint.htm`.

[289] JS Fenton, "Information Protection Systems," PhD thesis, Cambridge University, 1973.

[290] N Ferguson, B Schneier, "A Cryptographic Evaluation of IPSEC," at `http://www.counterpane.com/ipsec.html`.

[291] D Ferraiolo, R Kuhn, "Role-Based Access Controls," in *15th National Computer Security Conference* (1992); proceedings published by NIST, pp 554–563.

[292] PFJ Fillery, AN Chandler, "Is Lack of Quality Software a Password to Information Security Problems?" in *IFIP SEC 94*, paper C8.

[293] "Psychologists and Banks Clash over Merits of Photographs on Cards," in *Financial Technology International Bulletin*, v 13 no 5 (Jan 1996), pp 2–3.

[294] D Fine, "Why Is Kevin Lee Poulsen Really in Jail?" at `http://www.well.com/user/fine/journalism/jail.html`.

[295] B Fischer, talk given at Cryptologic History Symposium, NSA (Oct 1999); reported in *Cryptologia*, v 24 no 2 (Apr 2000), pp 160–167.

[296] S Fischer-Hubner, "Towards a Privacy-Friendly Design and Use of IT-Security Mechanisms," in *17th National Computer Security Conference* (1994); proceedings published by NIST, pp 142–152.

[297] RA Fisher, *The Genetical Theory of Natural Selection*, Oxford: Clarendon Press (1930); 2nd ed., New York: Dover Publications (1958).

[298] J Flanagan, "Prison Phone Phraud (or The RISKS of Spanish)," reporting University of Washington staff newspaper, in `comp.risks`, v 12.47; at `http://catless.ncl.ac.uk/Risks/20.69.html`.

[299] M Fleet, "Five Face Sentence over Notes That Passed Ultraviolet Tests," in *The Daily Telegraph* (Dec 23, 1999), at `http://www.telegraph.co.uk:80/`.

[300] SN Foley, "Aggregation and Separation as Noninterference Properties," in *Journal of Computer Security*, v 1 no 2 (1992), pp 158–188.

[301] Fondazione Ugo Bordoni, Symposium on Electromagnetic Security for Information Protection, Rome, Italy (Nov 21–22, 1991).

[302] S Forrest, SA Hofmeyr, A Somayaji, "Computer Immunology," in *Communications of the ACM*, v 40 no 10 (Oct 1997), pp 88–96.

[303] DS Fortney, JJ Lim, "A Technical Approach for Determining the Importance of Information in Computerized Alarm Systems," in *17th National Computer Security Conference* (1994); proceedings published by NIST, pp 348–357.

[304] The Foundation for Information Policy Research, `http://www.fipr.org`.

[305] B Fox, "How to Keep Thieves Guessing," in *New Scientist* (June 3, 1995), p 18.

[306] B Fox, "Do Not Adjust Your set ... We Have Assumed Radio Control," in *New Scientist* (Jan 8, 2000), at `http://www.newscientist.com/ns/20000108/newsstory6.html`.

[307] B Fox, "The Pirate's Tale," in *New Scientist* (Dec 18, 1999), at `http://www.newscientist.com/ns/19991218/thepirates.html`.

[308] D Fox, "IMSI-Catcher," in *Datenschutz und Datensicherheit*, v 21 no 9 (Sept 1997), p 539.

[309] D Foxwell, "Off-the-Shelf, on to Sea," in *International Defense Review*, v 30 (Jan 1997), pp 33–38.

[310] D Foxwell, M Hewish, "GPS: Is It Lulling the Military into a False Sense of Security?" in *Jane's International Defense Review* (Sept 1998), pp 32–41.

[311] LJ Fraim, "SCOMP: A Solution to the Multilevel Security Problem," in *IEEE Computer*, v 16 no 7 (July 1983), pp 26–34.

[312] E Franz, A Jerichow, "A Mix-Mediated Anonymity Service and Its Payment," in *ESORICS 98*, Springer LNCS, v 1485, pp 313–327.

[313] T Fraser, "LOMAC: Low Water-Mark Integrity Protection for COTS Environments," in *Proceedings of the 2000 IEEE Symposium on Security and Privacy*, IEEE Computer Society Press, pp 230–245.

[314] "Banks Fingerprint Customers to Cut Cheque Fraud," in *Fraud Watch*, no 1 (1997), p 9.

[315] "Chip Cards Reduce Fraud in France," in *Fraud Watch*, no 1 (1996), p 8.

[316] "Counterfeit and Cross-Border Fraud on Increase Warning," in *Fraud Watch*, no 1 (1996), pp 6–7.

[317] "Finger Minutiae System Leaps the 1:100,000 False Refusal Barrier," in *Fraud Watch*, no 2 (1996), pp 6–9.

[318] "Widespread Card Skimming Causes European Concern," in *Fraud Watch*, no 3 (1997), pp 1–2.

[319] P Freiberger, M Swaine, *Fire in the Valley—The Making of the Personal Computer*, New York: McGraw-Hill (1999), ISBN 0-07-135892-7.

[320] M Freiss, *Protecting Networks with Satan*, O'Reilly & Associates (1997), ISBN 1-56592-425-8.

[321] J Frizell, T Phillips, T Groover, "The Electronic Intrusion Threat to National Security and Emergency Preparedness Telecommunications: An Awareness Document," in *17th National Computer Security Conference* (1994); proceedings published by NIST, pp 378–399.

[322] M Frost, "Spyworld: Inside the Canadian & American Intelligence Establishments," Diane Publishing Co (1994), ISBN 0-78815791-4.

[323] AM Froomkin, "The Death of Privacy," in *Stanford Law Review*, v 52, pp 1461–1543, at `http://www.law.miami.edu/~froomkin/articles/privacy-deathof.pdf`.

[324] DA Fulghum, "Communications Intercepts Pace EP-3s," in *Aviation Week and Space Technology*, v 146 no 19 (May 5, 1997), pp 53–54.

[325] S Furber, *ARM System Architecture*, Addison-Wesley (1996), ISBN 0-210-40352-8.

[326] HF Gaines, *Cryptanalysis—A Study of Ciphers and Their Solution*, Dover, ISBN 486-20097-3 (1939, 1956).

[327] M Galecotti, "Russia's Eavesdroppers Come Out of the Shadows," in *Jane's Intelligence Review*, v 9 no 12 (Dec 1997), pp 531–535.

[328] F Galton, "Personal Identification and Description," in *Nature* (June 21, 1888), pp 173–177.

[329] T Gandy, "Brainwaves in Fraud Busting," *Banking Technology* (Dec 1995/Jan 1996), pp 20–24.

[330] S Garfinkel, *Database Nation*, O'Reilly & Associates (2000), ISBN 1-56592-653-6.

[331] S Garfinkel, G Spafford, *Practical UNIX and Internet Security*, O'Reilly & Associates (1996), ISBN 1-56592-148-8.

[332] W Gates, W Buffett, "The Bill & Warren Show," in *Fortune* (July 20, 1998).

[333] General Accounting Office, U.S., "Medicare—Improvements Needed to Enhance Protection of Confidential Health Information," GAO/HEHS-99-140; at `http://www.gao.gov/AIndexFY99/abstracts/he99140.htm`.

[334] E German, "Problem Idents," at http://onin.com/fp/problemidents.html.

[335] A Gidari, JP Morgan, "Survey of State Electronic & Digital Signature Legislative Initiatives," at `http://www.ilpf.org/digsig/digrep.htm`.

[336] D Gifford, A Spector, "The CIRRUS Banking Network," in *Communications of the ACM*, v 28 no 8 (Aug 1985), pp 797–807.

[337] AA Giordano, HA Sunkenberg, HE de Pdero, P Stynes, DW Brown, SC Lee, "A Spread-Spectrum Simulcast MF Radio Network," in *IEEE Transactions on Communications*, v TC-30 no 5 (May 1982), pp 1057–1070.

[338] WN Goetzmann, "Financing Civilization," at `http://viking.som.yale.edu/will/finciv/chapter1.htm`.

[339] J Goguen, J Meseguer, "Security Policies and Security Models," in *Proceedings of*

the 1982 IEEE Computer Society Symposium on Research in Security and Privacy, pp 11–20.

[340] I Goldberg, D Wagner, "Randomness and the Netscape Browser," in *Dr. Dobbs Journal*, no 243 (Jan 1996), pp 66–70.

[341] L Goldberg, "Recycled Cold-War Electronics Battle Cellular Telephone Thieves," in *Electronic Design*, v 44 no 18 (Sept 3, 1996), pp 41–42.

[342] O Goldreich, *"Foundations of Cryptography" (fragments of a book)*, at `http://www.toc.lcs.mit.edu/~oded/homepage.html`.

[343] O Goldreich, "Modern Cryptography, Probabilistic Proofs, and Pseudorandomness," in Springer (1999), ISBN 3-540-64766-X.

[344] D Gollmann, *Computer Security*, New York: John Wiley & Sons, Inc. (1999), ISBN 0-471-97884-2.

[345] D Gollmann, "What Is Authentication?" in *Security Protocols*, Springer LNCS, v 1796 (2000), pp 65–72.

[346] L Gong, *Inside Java 2 Platform Security: Architecture, API Design, and Implementation*, Addison-Wesley (1999), ISBN 0-201-31000-7.

[347] KE Gordon, RJ Wong, "Conducting Filament of the Programmed Metal Electrode Amorphous Silicon Antifuse," in *Proceedings of International Electron Devices Meeting* (Dec 1993); reprinted as pp 6–3 to 6–10, *QuickLogic Data Book* (1994).

[348] J Gough, *Watching the Skies—A History of Ground Radar for the Air Defence of the United Kingdom by the Royal Air Force from 1946 to 1975*, London: Her Majesty's Stationery Office (1993), ISBN 0-11-772723-7.

[349] RM Graham, "Protection in an Information Processing Utility," in *Communications of the ACM*, v 11 no 5 (May 1968), pp 365–369.

[350] FT Grampp, RH Morris, "UNIX Operating System Security," in *AT&T Bell Laboratories Technical Journal*, v 63 no 8 (Oct 1984), pp 1649–1672.

[351] RD Graubart, JL Berger, JPL Woodward, "Compartmented Mode, Workstation Evaluation Criteria, Version 1," Mitre MTR 10953 (1991); also published by the Defense Intelligence Agency as Document DDS-2600-6243-91.

[352] J Gray, P Helland, P O'Neil, D Shasha, "The Dangers of Replication and a Solution," in *SIGMOD Record*, v 25 no 2 (1996), pp 173–182.

[353] J Gray, P Syverson, "A Logical Approach to Multilevel Security of Probabilistic Systems," in *Distributed Computing*, v 11 no 2 (1988), pp 73–90.

[354] T Greening, "Ask and Ye Shall Receive: A Study in Social Engineering," in *SIGSAC Review*, v 14 no 2 (Apr 1996), pp 9–14.

[355] A Griew, R Currell, *A Strategy for Security of the Electronic Patient Record*, Aberystwyth: Institute fo Health Informatics, University of Wales (Mar 1995).

[356] D Grover, *The Protection of Computer Software—Its Technology and Applications*, Cambridge: British Computer Society/Cambridge University Press (1992), ISBN 0-521-42462-3.

[357] D Gruhl, W Bender, "Information Hiding to Foil the Casual Counterfeiter," in

Proceedings of the Second International Workshop on Information Hiding (Portland, Oregon, Apr 1998), Springer LNCS, v 1525, pp 1–15.

[358] LC Guillou, M Ugon, JJ Quisquater, "The Smart Card—A Standardized Security Device Dedicated to Public Cryptology," in [702], pp 561–613.

[359] C Gülcü, G Tsudik, "Mixing E-mail with Babel," in *Proceedings of the Internet Society Symposium on Network and Distributed System Security* (1996); proceedings published by the IEEE, ISBN 0-8186-7222-6, pp 2–16.

[360] R Gupta, SA Smolka, S Bhaskar, "On Randomization in Sequential and Distributed Algorithms," in *ACM Computing Surveys*, v 26 no 1 (Mar 1994), pp 7–86.

[361] J Gurnsey, *Copyright Theft*, Aslib (1997), ISBN 0-566-07631-4.

[362] P Gutman, "Secure Deletion of Data from Magnetic and Solid-State Memory," in *Sixth USENIX Security Symposium Proceedings* (July 1996), pp 77–89.

[363] P Gutman, "Software Generation of Practically Strong Random Numbers," in *Seventh USENIX Security Symposium Proceedings* (Jan 1998), pp 243–257.

[364] S Haber, WS Stornetta, "How to Time-Stamp a Digital Document," in *Journal of Cryptology*, v 3 no 2 (1991), pp 99–111.

[365] S Haber, WS Stornetta, "Secure Names for Bit-Strings," in *4th ACM Conference on Computer and Communications Security*; proceedings published by the ACM, ISBN 0-89791-912-2CCS 97, pp 28–35.

[366] W Hackmann, "Asdics at War," in *IEE Review*, v 46 no 3 (May 2000), pp 15–19.

[367] "Chris Carey Arrested in New Zealand," in *Hack Watch News* (Jan 1, 1999), at `http://www.iol.ie/~kooltek/legal.html`.

[368] N Hager, *Secret Power—New Zealand's Role in the International Spy Network*, Craig Potton Publishing (1996), ISBN 0-908802-35-8.

[369] PS Hall, TK Garland-Collins, RS Picton, RG Lee, *Radar*, Brassey's New Battlefield Weapons Systems and Technology Series, v 9, ISBN 0-08-037711-4.

[370] H Handschuh, P Paillier, J Stern, "Probing Attacks on Tamper-Resistant Devices," in *Cryptographic Hardware and Embedded Systems—CHES 99*, Springer LNCS, v 1717, pp 303–315.

[371] R Hanley, "Millions in Thefts Plague New Jersey Area," in *The New York Times* (Feb 9, 1981), p A1.

[372] R Hanson, "Can Wiretaps Remain Cost-Effective?" in *Communications of the ACM*, v 37 no 12 (Dec 1994), pp 13–15.

[373] MA Harrison, ML Ruzzo, JD Ullman, "Protection in Operating Systems," in *Communications of the ACM*, v 19 no 8 (Aug 1976), pp 461–471.

[374] A Hassey, M Wells, "Clinical Systems Security—Implementing the BMA Policy and Guidelines," in [29], pp 79–94.

[375] Health and Safety Executive, Nuclear Safety Reports at `http://www.hse.gov.uk/nsd/`, especially "HSE Team Inspection of the

Control and Supervision of Operations at BNFL's Sellafield Site," http://www.hse.gov.uk/nsd/team.htm.

[376] N Heintze, "Scalable Document Fingerprinting," in *Second USENIX Workshop on Electronic Commerce* (1996), ISBN 1-880446-83-9, pp 191–200.

[377] Herodotus, *Histories*, Book 1, 123.4, Book 5 35.3, and Book 7 239.3.

[378] "Interview with David Herson—SOGIS," (Sept 25, 1996), in *Ingeniørennet*, at http://www.ing.dk/redaktion/herson.htm.

[379] A Herzberg, M Jokabsson, S Jarecki, H Krawczyk, M Yung, "Proactive Public Key and Signature Systems," *4th ACM Conference on Computer and Communications Security* (1997), pp 100–110.

[380] RA Hettinga, "Credit Card Fraud Higher, Credit Card Fraud Lower," in nettime (Mar 22, 2000), at http://www.nettime.org/nettime.w3archive/200003/msg00184.html.

[381] M Hewish, "Combat ID Advances on All Fronts," in *International Defense Review*, v 29 (Dec 1996), pp 18–19.

[382] Hewlett-Packard, "IA-64 Instruction Set Architecture Guide," at http://devresource.hp.com/devresource/Docs/Refs/IA64ISA/index.html.

[383] HJ Highland, "Electromagnetic Radiation Revisited," in *Computers & Security*, v 5 (1986), pp 85–93, 181–184.

[384] HJ Highland, "Perspectives in Information Technology Security," in *Proceedings of the 1992 IFIP Congress, Education and Society*, IFIP A-13, v II (1992), pp 440–446.

[385] TF Himdi, RS Sandhu, "Lattice-Based Models for Controlled Sharing of Confidential Information in the Saudi Hajj System," in *13th Annual Computer Security Applications Conference*, San Diego, CA (Dec 8–12, 1997); proceedings published by the IEEE Computer Society, ISBN 0-8186-8274-4; pp 164–174.

[386] J Hoffman, "Implementing RBAC on a Type-Enforced System," in *13th Annual Computer Security Applications Conference*, San Diego, CA (Dec 8–12, 1997); proceedings published by the IEEE Computer Society, ISBN 0-8186-8274-4; pp 158–163.

[387] P Hollinger, "Single Language for Barcode Babel," in *Financial Times* (July 25, 2000), p 15.

[388] C Holloway, "Controlling the Use of Cryptographic Keys," in *Computers and Security*, v 14 no 7 (1995), pp 587–598.

[389] DI Hopper, "Authorities Sue Adult Web Sites," in *The Washington Post* (Aug 23, 2000); at http://www.washingtonpost.com/.

[390] G Horn, B Preneel, "Authentication and Payment in Future Mobile Systems," in *ESORICS 98*, Springer LNCS, v 1485, pp 277–293; journal version in *Journal of Computer Security*, v 8 no 2–3 (2000), pp 183–207.

[391] JD Horton, R Harland, E Ashby, RH Cooper, WF Hyslop, DG Nickerson, WM

Stewart, OK Ward, "The Cascade Vulnerability Problem," in *Journal of Computer Security*, v 2 no 4 (1993), pp 279–290.

[392] JD Howard, "An Analysis of Security Incidents on the Internet 1989–1995," PhD thesis (1997), Carnegie Mellon University, at http://www.cert.org/research/JHThesis/Start.html.

[393] D Howell, "Counterfeit Technology Forges Ahead," in *The Daily Telegraph* (Mar 22, 1999), at http://www.telegraph.co.uk:80/.

[394] N Htoo-Mosher, R Nasser, N Zunic, J Straw, "E4 ITSEC Evaluation of PR/SM on ES/9000 Processors," in *19th National Information Systems Security Conference* (1996), proceedings published by MST, pp 1–11.

[395] Q Hu, JY Yang, Q Zhang, K Liu, XJ Shen, "An Automatic Seal Imprint Verification Approach," in *Pattern Recognition*, v 28 no 8 (Aug 1995), pp 251–266.

[396] G Huber, "CMW Introduction," in *ACM SIGSAC*, v 12 no 4 (Oct 1994), pp 6–10.

[397] IBM, *'IBM 4758 PCI Cryptographic Coprocessor—CCA Basic Services Reference and Guide*, Release 1.31 for the IBM 4758-001, available through http://www.ibm.com/security/cryptocards/.

[398] "Role of Communications in Operation Desert Storm," *IEEE Communications Magazine*, Special Issue, v 30 no 1 (Jan 1992).

[399] *IEEE Carnahan Conference*, at http://www.carnahanconference.com/.

[400] *IEEE Electronics and Communications Engineering Journal*, v 12 no 3 (June 2000), special issue on UMTS.

[401] *IEEE Spectrum*, special issue on nuclear safekeeping, v 37 no 3 (Mar 2000).

[402] IFCI, "Real Cases," at http://risk.ifci.ch/Realcases.htm.

[403] "Ex-Radio Chief 'Masterminded' TV Cards Scam," in *The Independent* (Feb 17, 1998); see also, "The Sinking of a Pirate," *Sunday Independent* (Mar 1, 1998).

[404] Intel Corporation, *Intel Architecture Software Developer's Manual, Volume 1: Basic Architecture*, Order number 243190 (1997).

[405] "New England Shopping Mall ATM Scam Copied in UK," in *Information Security Monitor*, v 9 no 7 (June 1994), pp 1–2.

[406] "Pink Death Strikes at US West Cellular," in *Information Security Monitor*, v 9 no 2 (Jan 1994), pp 1–2.

[407] Information Systems Audit and Control Association, "Control Objectives for Information and related Technology," at http://www.isaca.org/cobit.htm.

[408] Information Systems Audit and Control Association, "Exam Preparation Materials," available from ISACA, at http://www.isaca.org/cert1.htm.

[409] International Atomic Energy Authority (IAEA), "The Physical Protection of Nuclear Material and Nuclear Facilities," INFCIRC/225/Rev. 4, at http://www.iaea.org/worldatom/program/protection/index.shtml.

[410] International Electrotechnical Commission, *Digital Audio Interface*, IEC 60958, Geneva (Feb 1989).

[411] I Jackson, personal communication with the author.

[412] L Jackson, "BT Forced to Pay Out Refunds after Free Calls Fraud," in *The Sunday Telegraph* (Feb 9, 1997); at `http://www.telegraph.co.uk:80/`.

[413] G Jagpal, "Steganography in Digital Images," undergraduate thesis, Selwyn College, Cambridge University (1995).

[414] AK Jain, R Bolle, S Pankanti, *Biometrics—Personal Identification in Networked Society*, Kluwer (1991), ISBN 0-7923-8346-1.

[415] AK Jain, L Hong, S Pankanti, R Bolle, "An Identity-Authentication System Using Fingerprints," in *Proceedings of the IEEE*, v 85 no 9 (Sept 1997), pp 1365–1388.

[416] S Jajodia, W List, G McGregor, L Strous (eds), *Integrity and Internal Control in Information Systems, Volume 1: Increasing the Confidence in Information Systems*, Chapman & Hall (1997), ISBN 0-412-82600-3.

[417] M Jay, "ACPO's Intruder Policy—Underwritten?" in *Security Surveyor*, v 26 no 3 (Sept 1995), pp 10–15.

[418] N Jefferies, C Mitchell, M Walker, "A Proposed Architecture for Trusted Third-Party Services," in *Cryptography: Policy and Algorithms*, Springer LNCS, v 1029, pp 98–104; also appeared at the Public Key Infrastructure Invitational Workshop at MITRE, VA (Sept 1995) and PKS '96 in Zürich (Oct 1, 1996).

[419] A Jerichow, J Müller, A Pfitzmann, B Pfitzmann, M Waidner, "Real-Time Mixes: A Bandwidth-Efficient Anonymity Protocol," in *IEEE Journal on Special Areas in Communications*, v 16 no 4 (May 1998), pp 495–509.

[420] John Young Architect, `http://www.jya.com`.

[421] K Johnson, "One Less Thing to Believe In: Fraud at Fake Cash Machine," in *The New York Times* (May 13, 1993), p 1.

[422] RG Johnson, ARE Garcia, "Vulnerability Assessment of Security Seals," in *Journal of Security Administration*, v 20 no 1 (June 1997), pp 15–27; `http://lib-www.lanl.gov/la-pubs/00418796.pdf`; more at `http://pearl1.lanl.gov/seals/`.

[423] P Jones, "Protection Money," in *Computer Business Review*, v 4 no 12 (Dec 1996), pp 31–36.

[424] RV Jones, *Most Secret War*, Wordsworth Editions (1978, 1998), ISBN 1-85326-699-X.

[425] RV Jones, "*Reflections on Intelligence*," Octopus (1989), ISBN 0-7493-0474-X.

[426] A Jøsang, K Johannesen, "Authentication in Analogue Telephone Access Networks," in *Pragocrypt 96;* proceedings published by CTU Publishing House, Prague, ISBN 80-01-01502-5; pp 324–336.

[427] *Dorothy Judd v Citibank*, 435 NYS, 2d series, pp 210–212, 107 Misc.2d 526.

[428] D Kahn, *The Codebreakers*, New York: Macmillan (1967).

[429] D Kahn, *Seizing the Enigma*, New York: Houghton Mifflin (1991), ISBN 0-395-42739-8.

[430] D Kahn, "Soviet Comint in the Cold War," in *Cryptologia*, v XXII no 1 (Jan 1998), pp 1–24.

[431] M Kam, G Fielding, R Conn, "Writer Identification by Professional Document Examiners," in *Journal of Forensic Sciences*, v 42 (1997), pp 778–786.

[432] M Kam, G Fielding, R Conn, "Effects of Monetary Incentives on Performance of Nonprofessionals in Document Examination Proficiency Tests," in *Journal of Forensic Sciences*, v 43 (1998), pp 1000–1004.

[433] MS Kamel, HC Shen, AKC Wong, RI Campeanu, "System for the Recognition of Human Faces," in *IBM Systems Journal*, v 32 no 2 (1993), pp 307–320.

[434] MH Kang, IS Moskowitz, "A Pump for Rapid, Reliable, Secure Communications," in *1st ACM Conference on Computer and Communications Security* (Nov 3–5, 1993), Fairfax, VA; proceedings published by the ACM, ISBN 0-89791-629-8, pp 118–129.

[435] MH Kang, JN Froscher, J McDermott, O Costich, R Peyton, "Achieving Database Security through Data Replication: The SINTRA Prototype," in *17th National Computer Security Conference* (1994), pp 77–87.

[436] MH Kang, IS Moskowitz, DC Lee, "A Network Pump," in *IEEE Transactions on Software Engineering*, v 22 no 5 (May 1996), pp 329–338.

[437] MH Kang, IS Moskowitz, B Montrose, J Parsonese, "A Case Study of Two NRL Pump Prototypes," in *12th Annual Computer Security Applications Conference*, San Diego, CA, (Dec 9–13, 1996); proceedings published by the IEEE, ISBN 0-8186-7606-X, pp 32–43.

[438] MH Kang, JN Froscher, IS Moskowitz, "An Architecture for Multilevel Secure Interoperability," in *13th Annual Computer Security Applications Conference*, San Diego, CA, (Dec 8–12, 1997); proceedings published by the IEEE Computer Society, ISBN 0-8186-8274-4; pp 194–204.

[439] CS Kaplan, "Privacy Plan Likely to Kick Off Debate," in *The New York Times* (July 28, 2000), at `http://www.nytimes.com/`.

[440] PA Karger, VA Austell, DC Toll, "*A New Mandatory Security Policy Combining Secrecy and Integrity*," IBM Research Report RC 21717 (97406) (Mar 15, 2000).

[441] F Kasiski, *Die Geheimschriften und die Dechiffrier-Kunst*, Berlin: Mittler & Sohn (1863).

[442] KASUMI Specification, ETSI/SAGE, v 1 (Dec 23, 1999), at `http://www.etsi.org/dvbandca/`.

[443] S Katzenbeisser, FAP Petitcolas, *Information Hiding—Techniques for Steganography and Digital Watermarking*, Artech House (2000), ISBN 1-58053-035-4.

[444] C Kaufman, R Perlman, M Speciner, *Network Security—Private Communication in a Public World*, Prentice Hall 1995, ISBN 0-13-061466-1.

[445] DT Keitkemper, SF Platek, KA Wolnik, "DNA versus Fingerprints," in *Journal of Forensic Sciences*, v 40 (1995), p 534.

[446] J Kelsey, B Schneier, D Wagner, "Protocol Interactions and the Chosen Protocol Attack," in *Security Protocols—Proceedings of the 5th International Workshop* (1997), Springer LNCS, v 1361, pp 91–104.

[447] J Kelsey, B Schneier, D Wagner, C Hall, "Cryptanalytic Attacks on Pseudorandom Number Generators," in *Fifth International Workshop on Fast Software Encryption* (1998), Springer LNCS, v 1372, pp 168–188.

[448] R Kemmerer, "Shared Resource Matrix Methodology: An Approach to Identifying Storage and Timing Channels," in *IEEE Transactions on Computer Systems*, v 1 no 3 (1983), pp 256–277.

[449] R Kemmerer, C Meadows, J Millen, "Three Systems for Cryptographic Protocol Analysis," in *Journal of Cryptology*, v 7 no 2 (Spring 1994), pp 79–130.

[450] R Kemp, N Towell, G Pike, "When Seeing Should Not Be Believing: Photographs, Credit Cards and Fraud," in *Applied Cognitive Psychology*, v 11 no 3 (1997), pp 211–222.

[451] MG Kendall, B Babington-Smith, "Randomness and Random Sampling Numbers," part 1 in *Journal of the Royal Statistical Society*, v 101, pp 147–166; part 2, in *Supplement to the Journal of the Royal Statistical Society*, v 6 no 1, pp 51–61.

[452] JO Kephardt, SR White, "Measuring and Modeling Computer Virus Prevalence," in *Proceedings of the 1993 IEEE Symposium on Security and Privacy*, pp 2–15.

[453] JO Kephardt, SR White, DM Chess, "Epidemiology of Computer Viruses," in *IEEE Spectrum*, v 30 no 5 (May 1993), pp 27–29.

[454] A Kerckhoffs, "La Cryptographie Militaire," in *Journal des Sciences Militaires* (Jan 9, 1883), pp 5–38; at `http://www.cl.cam.ac.uk/users/fapp2/kerckhoffs/`.

[455] PJ Kerry, "EMC in the New Millennium," in *Electronics & Communication Engineering Journal*, v 12 no 2, pp 43–48.

[456] D Kesdogan, H Federrath, A Jerichow, "Location Management Strategies Increasing Privacy in Mobile Communication," in *12th International Information Security Conference* (1996), Samos, Greece; proceedings published by Chapman & Hall, ISBN 0-412-78120-4, pp 39–48.

[457] J Kilian, P Rogaway, "How to Protect DES against Exhaustive Key Search," in *Advances in Cryptology—Crypto 96*, Springer LNCS, v 1109, pp 252–267.

[458] J King, "Bolero—A Practical Application of Trusted Third-Party Services," in *Computer Fraud and Security Bulletin* (July 1995), pp 12–15.

[459] Kingpin, "iKey 1000 Administrator Access and Data Compromise," in *bugtraq* (July 20, 2000), at `http://www.L0pht.com/advisories.html`.

[460] DV Klein, "Foiling the Cracker: A Survey of, and Improvements to, UNIX

Password Security," *Proceedings of the USENIX Security Workshop*, Portland, OR: USENIX Association (Summer 1990); http://www.deter.com/unix/.

[461] RL Klevans, RD Rodman, *Voice Recognition*, Artech House (1997), ISBN 0-89006-927-1.

[462] HM Kluepfel, "Securing a Global Village and Its Resources: Baseline Security for Interconnected Signaling System #7 Telecommunications Networks," in *First ACM Conference on Computer and Communications Security* (1993); proceedings published by the ACM, ISBN 0-89791-629-8, pp 195–212; later version in *IEEE Communications Magazine*, v 32 no 9 (Sept 1994), pp 82–89.

[463] N Koblitz, *A Course in Number Theory and Cryptography*, Springer Graduate Texts in Mathematics, no 114 (1987), ISBN 0-387-96576-9.

[464] ER Koch, J Sperber, *Die Datenmafia*, Rohwolt Verlag (1995), ISBN 3-499-60247-4.

[465] M Kochanski, "A Survey of Data Insecurity Devices," in *Cryptologia*, v IX no 1, pp 1–15.

[466] P Kocher, "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems," in *Advances in Cryptology—Crypto 96*, Springer LNCS, v 1109, pp 104–113.

[467] P Kocher, "Differential Power Analysis," in *Advances in Cryptology—Crypto 99*, Springer LNCS, v 1666, pp 388–397; a brief version was presented at the rump session of Crypto 98.

[468] KJ Koelman, "A Hard Nut to Crack: The Protection of Technological Measures," in *European Intellectual Property Review* (2000), pp 272–288; at http://www.ivir.nl/Publicaties/koelman/hardnut.html.

[469] S Kokolakis, D Gritzalis, S Katsikas, "Generic Security Policies for Health Information Systems," in *Health Informatics Journal*, v 4 nos 3–4 (Dec 1998), pp 184–195.

[470] O Kömmerling, MG Kuhn, "Design Principles for Tamper-Resistant Smartcard Processors," in *USENIX Workshop on Smartcard Technology;* proceedings published by USENIX (1999), ISBN 1-880446-34-0, pp 9–20.

[471] A Kondi, R Davis, "Software Encryption in the DoD," in *20th National Information Systems Security Conference* (1997); proceedings published by NIST, pp 543–554.

[472] BJ Koops, "Crypto Law Survey," at http://cwis.kub.nl/~frw/people/koops/lawsurvy.htm; see also his thesis "The Crypto Controversy: A Key Conflict in the Information Society," The Hague: Kluwer Law International (1999), ISBN 90-411-1143-3.

[473] DP Kormann, AD Rubin, "Risks of the Passport Single Signon Protocol," in *Computer Networks* (July 2000); at http://avirubin.com/vita.html.

[474] H Krawczyk, M Bellare, R Canetti, "HMAC: Keyed-Hashing for Message Authentication," RFC 2104 (Feb 1997); at http://www.faqs.org/rfcs/rfc2104.html.

[475] HM Kriz, "Phreaking recognized by Directorate General of France Telecom," in *Chaos Digest* 1.03 (Jan 1993).

[476] I Krsul, EH Spafford, "Authorship Analysis: Identifying the Author of a Program," in *Computers and Security*, v 16 no 3 (1996), pp 233–257.

[477] MG Kuhn, "Cipher Instruction Search Attack on the Bus-Encryption Security Microcontroller DS5002FP," in *IEEE Transactions on Computers*, v 47 no 10 (Oct 1998), pp 1153–1157.

[478] MG Kuhn, RJ Anderson, "Soft Tempest: Hidden Data Transmission Using Electromagnetic Emanations," in *Proceedings of the Second International Workshop on Information Hiding* (Portland, Apr 1998), Springer LNCS, v 1525, pp 126–143.

[479] MG Kuhn, private communication with the author.

[480] R Kuhn, P Edfors, V Howard, C Caputo, TS Philips, "Improving Public Switched Network Security in an Open Environment," in *Computer* (Aug 1993), pp 32–35.

[481] "L0phtCrack 2.52 for Win95/NT," at `http://www.l0pht.com/l0phtcrack/`.

[482] RJ Lackey, DW Upmal, "Speakeasy: The Military Software Radio," in *IEEE Communications Magazine*, v 33 no 5 (May 1995), pp 56–61.

[483] J Lacy, SR Quackenbush, A Reibman, JH Snyder, "Intellectual Property Protection Systems and Digital Watermarking," in *Proceedings of the Second International Workshop on Information Hiding* (Portland, OR: Apr 1998), Springer LNCS, v 1525, pp 158–168.

[484] Lamarr/Antheil Patent Story Home Page, `http://www.ncafe.com/chris/pat2/index.html`; contains U.S. patent no 2,292,387 (HK Markey et al., Aug 11, 1942).

[485] G Lambourne, *The Fingerprint Story*, Harrap (1984), ISBN 0-245-53963-8.

[486] L Lamport, "Time, Clocks and the Ordering of Events in a Distributed System," in *Communications of the ACM*, v 21 no 7 (July 1978), pp 558–565.

[487] L Lamport, R Shostack, M Peace, "The Byzantine Generals' Problem," in *ACM Transactions on Programming Languages and Systems*, v 4 no 3 (1982), pp 382–401.

[488] B Lampson, "A Note on the Confinement Problem," in *Communications of the ACM*, v 16 no 10 (Oct 1973), pp 613–615.

[489] P Lamy, J Martinho, T Rosa, MP Queluz, "Content-Based Watermarking for Image Authentication," in *Proceedings of the Third International Workshop on Information Hiding* (1999), Springer LNCS, v 1768, pp 187–198.

[490] S Landau, S Kent, C Brooks, S Charney, D Denning, W Diffie, A Lauck, D Miller, P Neumann, D Sobel, "Codes, Keys, and Conflicts: Issues in U.S. Crypto Policy," *Report of the ACM U.S. Public Policy Committee* (June 1994).

[491] R Landley, "Son of DIVX: DVD Copy Control," Motley Fool, `http://www.fool.com/portfolios/rulemaker/2000/rulemaker000127.htm`.

[492] P Landrock, "Roles and Responsibilities in BOLERO," in *TEDIS EDI Trusted Third Parties Workshop* (1995); proceedings published as ISBN 84-7653-506-6, pp 125–135.

[493] CE Landwehr, AR Bull, JP McDermott, WS Choi, "A Taxonomy of Computer Program Security Flaws, with Examples," U.S. Navy Report NRL/FR/5542–93-9591 (Nov 19, 1993).

[494] D Lane, "Where Cash is King," in *Banking Technology* (Oct 1992), pp 38–41.

[495] J Leake, "Workers Used Forged Passes at Sellafield," in *Sunday Times* (Apr 2, 2000), p 6.

[496] HC Lee, RE Guesslen (eds), *Advances in Fingerprint Technology*, Elsevier (1991), ISBN 0-444-01579-5.

[497] AK Lenstra, HW Lenstra, "The Development of the Number Field Sieve," in *Springer Lecture Notes in Mathematics*, v 1554 (1993), ISBN 0-387-57013-6.

[498] NG Leveson, *Safeware—System Safety and Computers*, Addison-Wesley (1994), ISBN 0-201-11972-2.

[499] A Lewcock, "Bodily Power," in *Computer Business Review*, v 6 no 2 (Feb 1998), pp 24–27.

[500] O Lewis, "Re: News: London Mailbomber Used the Net," post to `ukcrypto` mailing list (June 5, 2000), archived at `http://www.cs.ucl.ac.uk/staff/I.Brown/archives/ukcrypto/` and `http://www.chiark.greenend.org.uk/mailman/listinfo/ukcrypto`.

[501] "Minister Backs Phone Crime Initiative," *Lewisham Community News*, at `http://www.lewisham.gov.uk/templates/community/commdetails.cfm?file=2000071200.txt`.

[502] CC Lin, WC Lin, "Extracting Facial Features by an Inhibiting Mechanism Based on Gradient Distributions," in *Pattern Recognition*, v 29 no 12 (Dec 1996), pp 2079–2101.

[503] R Linde, "Operating Systems Penetration," *National Computer Conference*, AFIPS (1975), pp 361–368.

[504] JPMG Linnartz, "The 'Ticket' Concept for Copy Control Based on Embedded Signalling," *Fifth European Symposium on Research in Computer Security* (ESORICS 1998), Springer LNCS, v 1485, pp 257–274.

[505] JPMG Linnartz, M van Dijk, "Analysis of the Sensitivity attack against Electronic Watermarks in Images," in [59], pp 258–272.

[506] B Littlewood, "Predicting Software Reliability," in *Philosophical Transactions of the Royal Society of London*, A327 (1989), pp 513–527.

[507] WF Lloyd, *Two Lectures on the Checks to Population*, Oxford University Press (1833).

[508] Lockheed Martin, "Covert Surveillance Using Commercial Radio and Television Signals," at `http://silentsentry.external.lmco.com`.

[509] L Loeb, *Secure Electronic Transactions—Introduction and Technical Reference*, Artech House (1998), ISBN 0-89006-992-1.

[510] PA Loscocco, SD Smalley, PA Muckelbauer, RC Taylor, SJ Turner, JF Farrell, "The Inevitability of Failure: The Flawed Assumption of Security in Modern

Computing Environments," in *20th National Information Systems Security Conference;* proceedings published by NIST (1998), pp 303–314.

[511] WW Lowrance, "Privacy and Health Research," Report to the U.S. Secretary of Health and Human Services (May 1997).

[512] M Ludwig, *The Giant Black Book of Computer Viruses*, American Eagle Publishers (1995), ISBN 0-929408-10-1.

[513] AP Lutzker, "Primer on the Digital Millennium—What the Digital Millennium Copyright Act and the Copyright Term Extension Act Mean for the Library Community," Association of Research Libraries, at `http://www.arl.org/info/frn/copy/primer.html`.

[514] M Lyu, *Software Reliability Engineering*, IEEE Computer Society Press (1995), ISBN 0-07-039400-8.

[515] B Macq, "Special Issue: Identification and Protection of Multimedia Information," *Proceedings of the IEEE*, v 87 no 7 (July 1999).

[516] W Madsen, "Airline Passengers to Be Subject to Database Monitoring," in *Computer Fraud and Security Bulletin* (Mar 1997), pp 7–8.

[517] W Madsen, "Crypto AG: The NSA's Trojan Whore?" in *Covert Action Quarterly* (Winter 1998), at `http://www.mediafilter.org/caq/cryptogate/`.

[518] W Madsen, "Government-Sponsored Computer Warfare and Sabotage," in *Computers and Security*, v 11 (1991), pp 233–236.

[519] M Maes, "Twin Peaks: The Histogram Attack on Fixed-Depth Image Watermarks," in *Proceedings of the Second International Workshop on Information Hiding* (1998), Springer LNCS, v 1525, pp 290–305.

[520] K Maguire, "Muckraker Who Feeds Off Bins of the Famous," in *The Guardian* (July 27, 2000), at `http://www.guardianunlimited.co.uk/Labour/Story/0,2763,347535,00.html`.

[521] S Maguire, *Debugging the Development Process*, Redmond, WA: Microsoft Press, ISBN 1-55615-650-2 (1994), p 50.

[522] D Maio, D Maltoni, "Direct Gray-Scale Minutiae Detection in Fingerprints," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v 19 no 1 (Jan 1997), pp 27–40.

[523] L Marks, *Between Silk and Cyanide—A Codemaker's War 1941–1945*, New York: HarperCollins (1998), ISBN 0-68486780-X.

[524] D Martin, "Internet Anonymizing Techniques," in `;login:` Magazine, (May 1998); at `http://www.usenix.org/publications/login/1998-5/martin.html`.

[525] B Masuda, "Reducing the Price of Convenience," *International Security Review*, no 82 (Autumn 1993), pp 45–48.

[526] M Matsui, "Linear Cryptanalysis Method for DES Cipher," in *Advances in Cryptology—Eurocrypt 93*, Springer LNCS, v 765, pp 386–397.

[527] M Matsui, "New Block Encryption Algorithm MISTY," in *Fourth International*

Workshop on Fast Software Encryption (1997), Springer LNCS, v 1267, pp 54–68.

[528] Gospel according to St. Matthew, Chapter 7, verse 3.

[529] R Matthews, "The Power of One," in *New Scientist* (Oct 7, 1999), pp 26–30; at http://www.newscientist.com/ns/19990710/thepowerof.html.

[530] V Matyás, "Protecting the Identity of Doctors in Drug Prescription Analysis," in *Health Informatics Journal*, v 4 nos 3–4 (Dec 1998), pp 205–209.

[531] D Mazières, MF Kaashoek, "The Design, Implementation, and Operation of an Email Pseudonym Server," in *Proceedings of the 5th ACM Conference on Computer and Communications Security* (1998), http://www.pdos.lcs.mit.edu/~dm.

[532] J McCormac, "*European Scrambling Systems—The Black Book*," version 5, Waterford University Press, Ireland (1996), ISBN 1-873556-22-5.

[533] D McCullagh, "U.S. to Track Crypto Trails," in *Wired* (May 4, 2000), at http://www.wired.com/news/politics/0,1283,36067,00.html; statistics at http://www.uscourts.gov/wiretap99/contents.html.

[534] D McCullough, "A Hook-up Theorem for Multi-Level Security," in *IEEE Transactions on Software Engineering*, v 16 no 6 (June 1990), pp 563–568.

[535] K McCurley, Remarks at IACR General Meeting, *Crypto 98*, Santa Barbara, CA: (Aug 1998).

[536] AD McDonald, MG Kuhn, "StegFS: A Steganographic File System for Linux," in [613], pp 463–477.

[537] G McGraw, EW Felten, *Java Security*, New York: John Wiley & Sons, Inc. (1997), ISBN 0-471-17842-X.

[538] I McKie, "Total Vindication for Shirley McKie!" (June 23, 2000), at http://onin.com/fp/mckievindication.html.

[539] J McLean, "The Specification and Modeling of Computer Security," in *Computer*, v 23 no 1 (Jan 1990), pp 9–16.

[540] J McLean, "Security Models," in *Encyclopedia of Software Engineering*, New York: John Wiley & Sons, Inc. (1994).

[541] J McLean, "A General Theory of Composition for a Class of 'Possibilistic' Properties," in *IEEE Transactions on Software Engineering*, v 22 no 1 (Jan 1996), pp 53–67.

[542] J McNamara, "The Complete, Unofficial TEMPEST Information Page," at http://www.eskimo.com/~joelm/tempest.html.

[543] B McWilliams, "Sex Sites Accused of Gouging Visitors with Phone Scam," in InternetNews.com (Apr 7, 2000), at http://www.internetnews.com/bus-news/print/0,,3_337101,00.html.

[544] AJ Menezes, PC van Oorschot, SA Vanstone, *Handbook of Applied Cryptography*, CRC Press (1997); ISBN 0-8493-8523-7; also available online at http://www.cacr.math.uwaterloo.ca/hac/.

[545] CG Menk, "System Security Engineering Capability Maturity Model and Evaluations: Partners within the Assurance Framework," in *19th National Information Systems Security Conference* (1996), pp 76–88.

[546] J Mercer, "Document Fraud Deterrent Strategies: Four Case Studies," in *Optical Security and Counterfeit Deterrence Techniques II* (1998), IS&T (The Society for Imaging Science and Technology) and SPIE (The International Society for Optical Engineering), v 3314, ISBN 0-8194-2754-3, pp 39–51.

[547] TS Messergues, EA Dabish, RH Sloan, "Investigations of Power Analysis Attacks on Smartcards," in *USENIX Workshop on Smartcard Technology;* proceedings published by USENIX (1999), ISBN 1-880446-34-0, pp 151–161.

[548] CH Meyer and SM Matyas, *Cryptography: A New Dimension in Computer Data Security,* New York: John Wiley & Sons, Inc. (1982).

[549] R Meyer-Sommer, "Smartly Analyzing the Simplicity and the Power of Simple Power Analysis on Smartcards," in *Workshop on Cryptographic Hardware and Embedded Systems* (2000); Springer LNCS, v 1965, ISBN 3-540-41455-X, pp 78–92.

[550] J Micklethwait, A Wooldridge, *The Witch Doctors—What the Management Gurus Are Saying, Why It Matters and How to Make Sense of It,* New York: Random House (1997), ISBN 0-7493-2645-X.

[551] A Midgley, "R.I.P. and NHSNet," post to `ukcrypto` mailing list (July 1, 2000), archived at `http://www.cs.ucl.ac.uk/staff/I.Brown/archives/ukcrypto/`.

[552] J Millen, "A Resource Allocation Model for Denial of Service Protection," in *Journal of Computer Security,* v 2 nos 2–3 (1993), pp 89–106.

[553] B Miller, "Vital Signs of Security," in *IEEE Spectrum* (Feb 1994), pp 22–30.

[554] ML Miller, IJ Cox, JA Bloom, "Watermarking in the Real World: An Application to DVD," in *Sixth ACM International Multimedia Conference* (1998); workshop notes published by GMD—Forschungszentrum Informationstechnik GmbH, as v 41 of GMD Report, pp 71–76.

[555] K Mitnick, Congressional testimony, as reported by Associated Press (Mar 3, 2000); see also `http://www.zdnet.com/zdnn/stories/news/0,4586,2454737,00.html` and `http://news.cnet.com/category/0-1005-200-1562611.html`.

[556] B Moghaddam, A Pentland, "Probabilistic Visual learning for Object Representation," in *IEEE Transactions on Pattern Analysis and Machine Intelligence,* v 19 no 7 (July 1997), pp 696–710.

[557] F Mollet, "Card Fraud Nets Esc6 billion," in *Cards International* (Sept 22, 1995), p 3.

[558] E Montegrosso, "Charging and Accounting Mechanisms" (3G TR 22.924, v 3.1.1), from Third-Generation Partnership Project, at `http://www.3gpp.org/TSG/Oct_status_list.htm`.

[559] R Morris, "A Weakness in the 4.2BSD UNIX TCP/IP Software," Bell Labs

Computer Science Technical Report no. 117 (Feb 25, 1985); at http://www.cs.berkeley.edu/~daw/security/seq-attack.html.

[560] R Morris, Invited talk, *Crypto 95*.

[561] R Morris, K Thompson, "Password Security: A Case History," in *Communications of the ACM*, v 22 no 11 (Nov 1979), pp 594–597.

[562] DP Moynihan, *Secrecy—The American Experience*, New Haven, CT: Yale University Press (1999), ISBN 0-300-08079-4.

[563] P Mukherjee, V Stavridou, "The Formal Specification of Safety Requirements for Storing Explosives," in *Formal Aspects of Computing*, v 5 no 4 (1993), pp 299–336.

[564] T Mulhall, "Where Have All the Hackers Gone? A Study in Motivation, Deterrence, and Crime Displacement," in *Computers and Security*, v 16 no 4 (1997), pp 277–315.

[565] S Mullender (ed), *Distributed Systems*, Addison-Wesley (1993), ISBN 0-201-62427-3.

[566] JC Murphy, D Dubbel, R Benson, "Technology Approaches to Currency Security," in *Optical Security and Counterfeit Deterrence Techniques II* (1998), IS&T (The Society for Imaging Science and Technology) and SPIE (The International Society for Optical Engineering), v 3314, ISBN 0-8194-2754-3, pp 21–28.

[567] E Murray, "SSL Server Security Survey," at http://www.meer.net/~ericm/papers/ssl_servers.html.

[568] K Murray, "Protection of Computer Programs in Ireland," in *Computer Law and Security Report*, v 12 no 3 (May/June 1996), pp 57–59.

[569] RFH Nalder, *History of the Royal Corps of Signals*, Royal Signals Institution (1958).

[570] Napster, http://www.napster.com.

[571] M Nash, R Kennett, "Implementing Security Policy in a Large Defense Procurement," in *12th Annual Computer Security Applications Conference*, San Diego, CA (Dec 9–13, 1996); proceedings published by the IEEE, ISBN 0-8186-7606-X; pp 15–23.

[572] National Information Infrastructure Task Force, "Options for Promoting Privacy on the National Information Infrastructure" (Apr 1997), at http://www.iitf.nist.gov/ipc/privacy.htm.

[573] National Institute of Standards and Technology, archive of publications on computer security, http://csrc.nist.gov/publications/history/index.html.

[574] National Institute of Standards and Technology, "Common Criteria for Information Technology Security," Version 2.0/ISO IS 15408 (May 1998), http://www.commoncriteria.org.

[575] National Institute of Standards and Technology, "Data Encryption Standard (DES)," FIPS 46; Draft FIPS 46-3, incorporating upgrade to triple DES, at http://csrc.nist.gov/encryption/.

[576] National Institute of Standards and Technology, "Security Requirements for Cryptographic Modules" (Jan 11, 1994), at http://www.itl.nist.gov/fipspubs/0-toc.htm#cs.

[577] National Institute of Standards and Technology, "SKIPJACK and KEA Algorithms," (June 23, 1998), at http://csrc.nist.gov/encryption/skipjack-kea.htm.

[578] National Institute of Standards and Technology, "Escrowed Encryption Standard," FIPS 185 (Feb 1994).

[579] National Institute of Standards and Technology, "SCSUG Smart Card Protection Profile" (draft, v 2.0, May 2000), at http://csrc.nist.gov/cc/sc/sclist.htm.

[580] National Research Council, *Cryptography's Role in Securing the Information Society*, National Academy Press (1996), ISBN 0-309-05475-3.

[581] National Research Council, *For the Record: Protecting Electronic Health Information*, National Academy Press (1997), ISBN 0-309-05697-7.

[582] National Security Agency, "The NSA Security Manual," at http://www.cl.cam.ac.uk/ftp/users/rja14/nsaman.tex.gz.

[583] P Naur, B Randell, "Software Engineering—Report on a Conference," NATO Scientific Affairs Division, Garmisch (1968).

[584] R Neame, "Managing Health Data Privacy and Security," in [29], pp 225–232.

[585] GC Necula, P Lee, "Safe, Untrusted Agents Using Proof-Carrying Code," in *Mobile Agents and Security*, ISBN 3-540-64792-9, pp 61–91.

[586] RM Needham, "Denial of Service: An Example," in *Communications of the ACM*, v 37 no 11 (Nov 1994), pp 42–46.

[587] RM Needham, "Naming," in [565], pp 318–327.

[588] RM Needham, "The Hardware Environment," in *Proceedings of the 1999 IEEE Symposium on Security and Privacy*, IEEE Computer Society Press, p 236.

[589] RM Needham, MD Schroeder, "Using Encryption for Authentication in Large Networks of Computers," in *Communications of the ACM*, v 21 no 12 (Dec 1978), pp 993–999.

[590] P Neumann, *Computer-Related Risks*, Addison-Wesley (1995), ISBN 0-201-55805-X.

[591] J Newton, "Countering the Counterfeiters," in *Cards International* (Dec 12, 1994), p 12.

[592] J Newton, "Organised Plastic Counterfeiting," Her Majesty's Stationery Office (1996), ISBN 0-11-341128-6.

[593] Nuclear Regulatory Commission, www.nrc.gov.

[594] AM Odlyzko, "The History of Communications and Its Implications for the Internet," at http://www.research.att.com/~amo/doc/networks.html.

[595] AM Odlyzko, "Smart and Stupid Networks: Why the Internet Is Like Microsoft,"

ACM netWorker (Dec 1998), pp 38–46, at http://www.acm.org/networker/issue/9805/ssnet.html.

[596] N Okuntsev, *Windows NT Security*, R&D Books (1999), ISBN 0-87930-473-1.

[597] R Oppliger, *Internet and Intranet Security*, Artech House (1998), ISBN 0-89006-829-1.

[598] Organization for Economic Cooperation & Development, "Guidelines for the Protections of Privacy and Transborder Flow of Personal Data," OECD Doc. No C(80)58 (1981), at http://www.oecd.org//dsti/sti/it/secur/prod/PRIV-EN.HTM.

[599] J Osen, "The Cream of Other Men's Wit: Plagiarism and Misappropriation in Cyberspace," in *Computer Fraud and Security Bulletin* (Nov 1997), pp 13–19.

[600] S Pancho, "Paradigm Shifts in Protocol Analysis," in *Proceedings of the 1999 New Security Paradigms Workshop*, ACM (2000), pp 70–79.

[601] DJ Parker, "DVD Copy Protection: An Agreement At Last? Protecting Intellectual Property Rights in the Age of Technology," in *Tape/Disc Magazine* (Oct 1996), http://www.kipinet.com/tdb/tdb_oct96/feat_protection.html.

[602] DJ Parker, *Fighting Computer Crime—A New Framework for Protecting Information*, New York: John Wiley & Sons, Inc. (1998), ISBN 0-471-16378-3.

[603] B Patterson, letter to *Communications of the ACM*, v 43 no 4 (Apr 2000), pp 11–12.

[604] LC Paulson, "Inductive Analysis of the Internet Protocol TLS," in *ACM Transactions on Computer and System Security*, v 2 no 3 (1999), pp 332–351; also at http://www.cl.cam.ac.uk/users/lcp/papers/protocols.html.

[605] TP Pedersen, "Electronic Payments of Small Amounts," in *Security Protocols* (1996), Springer LNCS, v 1189, pp 59–68.

[606] A Perrig, "A Copyright Protection Environment for Digital Images," Diploma thesis, École Polytechnique Fédérale de Lausanne (1997).

[607] P Pesic, "The Clue to the Labyrinth: Francis Bacon and the Decryption of Nature," in *Cryptologia*, v XXIV, no 3 (July 2000), pp 193–211.

[608] R Petersen, "UCITA Update," at http://www.arl.org/info/frn/copy/petersen.html.

[609] I Peterson, "From Counting to Writing," MathLand Archives, http://www.maa.org/mathland/mathland_2_24.html.

[610] FAP Petitcolas, RJ Anderson, MG Kuhn, "Attacks on Copyright Marking Systems," in *Proceedings of the Second International Workshop on Information Hiding* (1998), Springer LNCS, v 1525, pp 219–239.

[611] FAP Petitcolas, RJ Anderson, MG Kuhn, "Information Hiding—A Survey," in *Proceedings of the IEEE*, v 87 no 7 (July 1999), pp 1062–1078.

[612] H Petroski, *To Engineer Is Human*, New York: Barnes and Noble Books (1994), ISBN 1-56619502-0.

[613] A Pfitzmann, *Proceedings of the Third International Workshop on Information Hiding* (1999), Springer LNCS, v 1768.

[614] B Pfitzmann, "Information Hiding Terminology," in *Proceedings of the First International Workshop on Information Hiding* (1996), Springer LNCS, v 1174, pp 347–350.

[615] GE Pickett, "How Do You Select the 'Right' Security Feature(s) for Your Company's Products???," in *Optical Security and Counterfeit Deterrence Techniques II* (1998), IS&T (The Society for Imaging Science and Technology) and SPIE (The International Society for Optical Engineering), v 3314, ISBN 0-8194-2754-3, pp 52–58.

[616] RL Pickholtz, DL Schilling, LB Milstein, "Theory of Spread-Spectrum Communications—A Tutorial," in *IEEE Transactions on Communications*, v TC-30 no 5 (May 1982), pp 855–884.

[617] RL Pickholtz, DB Newman, YQ Zhang, M Tatebayashi, "Security Analysis of the INTEL-SAT VI and VII Command Network," in *IEEE Proceedings on Selected Areas in Communications*, v 11 no 5 (June 1993), pp 663–672.

[618] D Polak, "GSM Mobile Network in Switzerland Reveals Location of Its Users," in *Privacy Forum Digest*, v 6 no 18 (Dec 31, 1997), at `http://www.vortex.com/privacy/priv.06.18`.

[619] Politech mailing list, at `http://www.politechbot.com/`.

[620] B Pomeroy, S Wiseman, "Private Desktops and Shared Store," in *Computer Security Applications Conference*, Phoenix, AZ (1998); proceedings published by the IEEE, ISBN 0-8186-8789-4, pp 190–200.

[621] B Preneel, PC van Oorschot, "MDx-MAC and Building Fast MACs from Hash Functions," in *Advances in Cryptology—Crypto 95*, Springer LNCS, v 963, pp 1–14.

[622] RS Pressman, *Software Engineering: A Practitioner's Approach*, New York: McGraw-Hill (5th ed, 2000), ISBN 0-073-65578-3.

[623] G Price, "The Interaction between Fault Tolerance and Security," Technical Report no 214, Cambridge University Computer Laboratory.

[624] WR Price, "Issues to Consider When Using Evaluated Products to Implement Secure Mission Systems," in *Proceedings of the 15th National Computer Security Conference*, National Institute of Standards and Technology (1992), pp 292–299.

[625] H Pringle, "The Cradle of Cash," in *Discover*, v 19 no 10 (Oct 1998); at `http://www.discover.com/oct_issue/cradle.html`.

[626] C Prins, "Biometric Technology Law," in *The Computer Law and Security Report*, v 14 no 3 (May/Jun 1998), pp 159–165.

[627] D Pritchard, *The Radar War—Germany's Pioneering Achievement 1904–1945*, Wellingborough (1989), ISBN 1-85260-246-5.

[628] The Privacy Exchange, `http://www.privacyexchange.org/`.

[629] Public Lending Right (PLR), at http://www.writers.org.uk/guild/Crafts/Books/PLRBody.html.

[630] Public Record Office, "Functional Requirements for Electronic Record Management Systems," (Nov 1999), at http://www.pro.gov.uk/recordsmanagement/eros/invest/reference.pdf.

[631] Rain Forest Puppy, "Issue Disclosure Policy V1.1," at http://www.wiretrip.net/rfp/policy.html.

[632] W Rankl, W Effing, *Smartcard Handbook*, New York: John Wiley & Sons, Inc. (1997), ISBN 0-471-96720-3; translated from the German *Handbuch der Chpkarten*, Carl Hanser Verlag (1995), ISBN 3-446-17993-3.

[633] ES Raymond, "The Case of the Quake Cheats," (Dec 27, 1999), at http://www.tuxedo. org/~esr/writings/quake-cheats.html.

[634] ES Raymond, "The Cathedral and the Bazaar," at http://www.tuxedo.org/~esr/writings/cathedral-bazaar/.

[635] ES Raymond, "The Magic Cauldron," (June 1999), at http://www.tuxedo.org/~esr/writings/magic-cauldron/magic-cauldron.html.

[636] SM Redl, MK Weber, MW Oliphant, *GSM and Personal Communications Handbook*, Artech House (1998), ISBN 0-89006-957-3.

[637] MG Reed, PF Syverson, DM Goldschlag, "Anonymous Connections and Onion Routing," in *IEEE Journal on Special Areas in Communications*, v 16 no 4 (May 1998), pp 482–494.

[638] C Reiss, "Mystery of Levy Tax Phone Calls," *The Evening Standard* (July 5, 2000), p 1; also at http://www.thisislondon.com/.

[639] MK Reiter, "A Secure Group Membership Protocol," in *IEEE Transactions on Software Engineering*, v 22 no 1 (Jan 1996), pp 31–42.

[640] MK Reiter, MK Franklin, JB Lacy, RA Wright, "The Omega Key Management Service," in *3rd ACM Conference on Computer and Communications Security* (1996), pp 38–47.

[641] M Reiter, AD Rubin, "Anonymous Web Transactions with Crowds," in *Communications of the ACM*, v 42 no 2 (Feb 1999), pp 32–38.

[642] J Reno, http://www.cnn.com/2000/US/05/25/security.breaches.01/index.html.

[643] MA Rice, AJ Sammes, *Command and Control: Support Systems in the Gulf War*, Brassey's (1994), ISBN 1-8575 3-015-2.

[644] D Richardson, *Techniques and Equipment of Electronic Warfare*, Salamander Books, ISBN 0-8601-265-8.

[645] LW Ricketts, JE Bridges, J Miletta, *EMP Radiation and Protective Techniques*, John Wiley & Sons, Inc. New York (1975), ISBN 0-471-010403-6.

[646] M Ridley, "The Red Queen: Sex and the Evolution of Human Nature," New York: Viking Books (1993), ISBN 0-1402-4548-0.

[647] V Rijmen, *The block cipher Rijndael*, at http://www.esat.kuleuven.ac.be/~rijmen/rijndael/.

[648] RL Rivest, A Shamir, "PayWord and MicroMint: Two Simple Micropayment Schemes," in *Security Protocols* (1996), Springer LNCS, v 1189, pp 69–87.

[649] RL Rivest, A Shamir, L Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," in *Communications of the ACM*, v 21 no 2 (Feb 1978), pp 120–126.

[650] AR Roddy, JD Stosz, "Fingerprint Features—Statistical Analysis and System Performance Estimates," in *Proceedings of the IEEE*, v 85 no 9 (Sept 1997), pp 1390–1421.

[651] DE Ross, "Two Signatures," in comp.risks, v 20.81: http://catless.ncl.ac.uk/Risks/20.81.html.

[652] M Rowe, "Card Fraud Plummets in France," *Banking Technology* (May 1994), p 10.

[653] WW Royce, "Managing the Development of Large Software Systems: Concepts and Techniques," in *Proceedings IEEE WESCON* (1970), pp 1–9.

[654] A Rubin, "Bugs in Anonymity Services," bugtraq (Apr 13, 1999); at http://www.securityportal.com/list-archive/bugtraq/1999/Apr/0126.html.

[655] HH Rubinovitz, "Issues Associated with Porting Applications to the Compartmented Mode Workstation," in *ACM SIGSAC*, v 12 no 4 (Oct 1994), pp 2–5.

[656] RA Rueppel, *Analysis and Design of Stream Ciphers*, Springer-Verlag (1986), ISBN 0-387-16870-2.

[657] RA Rueppel, "Criticism of ISO CD 11166 Banking: Key Management by Means of Asymmetric Algorithms," in *Proceedings of 3rd Symposium of State and Progress of Research in Cryptography*, Rome: Fondazione Ugo Bordoni (1993), pp 191–198.

[658] R Ruffin, "Following the Flow of Funds," in *Security Management* (July 1994), pp 46–52.

[659] J Rushby, B Randell, "A Distributed Secure System," in *IEEE Computer*, v 16 no 7 (July 1983), pp 55–67.

[660] D Russell, GT Gangemi, "Computer Security Basics," Chapter 10: *TEMPEST*, O'Reilly & Associates (1991), ISBN 0-937175-71-4.

[661] DR Safford, DL Schales, DK Hess, "The TAMU Security Package: An Ongoing Response to Internet Intruders in an Academic Environment," in *USENIX Security 93*, pp 91–118.

[662] JD Saltzer, MD Schroeder, "The Protection of Information in Computer Systems," in *Proceedings of the IEEE*, v 63 no 9 (Mar 1975), pp 1278–1308.

[663] RG Saltman, "Assuring Accuracy, Integrity, and Security in National Elections: The Role of the U.S. Congress," in *Computers, Freedom, and Privacy* (1993); at http://www.cpsr.org/conferences/cfp93/saltman.html.

[664] T Sammes, B Jenkinson, *Forensic Computing—A Practitioner's Guide*, Springer (2000), ISBN 1-85233-299-9.

[665] P Samuelson, "Copyright and Digital Libraries," in *Communications of the ACM*, v 38 no 4 (April 1995).

[666] P Samuelson, "Intellectual Property Rights and the Global Information Economy," in *Communications of the ACM*, v 39 no 1 (Jan 1996), pp 23–28.

[667] P Samuelson, "The Copyright Grab," at `http://uainfo.arizona.edu/~weisband/411_511/copyright.html`.

[668] D Samyde, JJ Quisquater, "S.E.M.A. Electromagnetic Analysis," *presented at the rump session of Eurocrypt 2000.*

[669] RS Sandhu, S Jajodia, "Polyinstantiation for Cover Stories," in *Computer Security—ESORICS 92*, LNCS, v 648, pp 307–328.

[670] SANS Institute, "Consensus List of the Top Ten Internet Security Threats," v 1.22 (June 19, 2000); at `http://www.sans.org/`.

[671] G Sandoval, "Glitches Let Net Shoppers Get Free Goods," in *CNET News.com* (July 5, 2000); at `http://news.cnet.com/news/0-1007-200-2208733.html`.

[672] PF Sass, L Gorr, "Communications for the Digitized Battlefield of the 21st Century," in *IEEE Communications*, v 33 no 10 (Oct 1995), pp 86–95.

[673] M Schaefer, "Symbol Security Condition Considered Harmful," in *Proceedings of the 1989 IEEE Symposium on Security and Privacy*, pp 20–46.

[674] RR Schell, "Computer Security: The Achilles' Heel of the Electronic Air Force?" in *Air University Review*, v 30 no 2 (Jan-Feb 1979), pp 16–33.

[675] RR Schell, PJ Downey, GJ Popek, "Preliminary Notes on the Design of Secure Military Computer Systems," Electronic Systems Division, Air Force Systems Command (Jan 1, 1973), MCI-73-1; at `http://seclab.cs.ucdavis.edu/projects/history/papers/sche73.pdf`.

[676] DL Schilling, *Meteor Burst Communications: Theory and Practice*, New York: John Wiley & Sons, Inc. (1993), ISBN 0-471-52212-0.

[677] DC Schleher, *Electronic Warfare in the Information Age*, Artech House (1999), ISBN 0-89006-526-8.

[678] D Schmandt-Besserat, *How Writing Came About*, University of Texas Press (1996); ISBN 0-29277-704-3, `http://www.dla.utexas.edu/depts/lrc/numerals/dsb1.html`.

[679] ZE Schnabel, "The Estimation of the Total Fish Population in a Lake," in *American Mathematical Monthly*, v 45 (1938), pp 348–352.

[680] PM Schneider, "Datenbanken mit genetischen Merkmalen von Straftätern," in *Datenschutz und Datensicherheit*, v 22 (June 1998), pp 330–333.

[681] B Schneier, *Applied Cryptography*, New York: John Wiley & Sons, Inc. (1996); ISBN 0-471-12845-7.

[682] B Schneier, "Why Computers Are Insecure," in `comp.risks` v 20.67; at `http://catless.ncl.ac.uk/Risks/20.67.html`.

[683] B Schneier, *Secrets and Lies: Digital Security in a Networked World*, New York: John Wiley & Sons, Inc. (2000); ISBN 0-471-25311-1.

[684] B Schneier, D Banisar, *The Electronic Privacy Papers—Documents on the Battle for Privacy in the Age of Surveillance*, New York: John Wiley & Sons, Inc. (1997); ISBN 0-471-12297-1.

[685] M Schnyder, "Datenfluesse im Gesundheitswesen," in *Symposium für Datenschutz und Informationssicherheit*, Zuerich (Oct 1998).

[686] RA Scholtz, "Origins of Spread-Spectrum Communications," in *IEEE Transactions on Communications*, v TC-30 no 5 (May 1982), pp 822–854.

[687] MD Schroeder, "Cooperation of Mutually Suspicious Subsystems in a Computer Utility," MIT PhD Thesis (Sept 1972); also available as Project MAC Technical Report MAC TR-104, `http://hdl.handle.net/ncstrl.mit_lcs/MIT/LCS/TR-104`.

[688] CJ Seiferth, "Opening the Military to Open Source," in *COTS Magazine* (Nov-Dec 1999), at `http://www.rtcgroup.com/cotsjournal/cotsj111200/cots111200.html`.

[689] CJ Seiferth, "Adoption of Open Licensing," in *COTS Magazine* (Nov-Dec 1999), at `http://www.rtcgroup.com/cotsjournal/cotsj111200/cots111200.html`.

[690] R Senderek, "Key-Experiments—How PGP Deals with Manipulated Keys," at `http://senderek.de/security/key-experiments.html`.

[691] D Senie, "Changing the Default foı Directed Broadcasts in Routers," RFC 2644, at `http://www.ietf.org/rfc/rfc2644.txt`.

[692] A Shamir, "How to Share a Secret," in *Communications of the ACM*, v 22 no 11 (Nov 1979), pp 612–613.

[693] MI Shamos, "Electronic Voting—Evaluating the Threat," in *Computers, Freedom, and Privacy* (1993); at `http://www.cpsr.org/conferences/cfp93/shamos.html`.

[694] CE Shannon, "A Mathematical Theory of Communication," in *Bell Systems Technical Journal*, v 27 (1948), pp 379–423, 623–656.

[695] CE Shannon, "Communication Theory of Secrecy Systems," in *Bell Systems Technical Journal*, v 28 (1949), pp 656–715.

[696] C Shapiro, H Varian, *Information Rules*, Boston: Harvard Business School Press (1998), ISBN 0-87584-863-X; see `http://www.inforules.com`.

[697] D Sherwin, "Fraud—The Unmanaged Risk," in *Financial Crime Review*, v 1 no 1 (Fall 2000), pp 67–69.

[698] PW Shor, "Algorithms for Quantum Computers," in *35th Annual Symposium on the Foundations of Computer Science* (1994); proceedings published by the IEEE, ISBN 0-8186-6580-7, pp 124–134.

[699] O Sibert, PA Porras, R Lindell, "An Analysis of the Intel 80x86 Security Architecture and Implementations," in *IEEE Transactions on Software Engineering*, v 22 no 5 (May 1996), pp 283–293.

[700] GJ Simmons, "The Prisoners' Problem and the Subliminal Channel," in *Proceedings of CRYPTO '83*, Plenum Press (1984), pp 51–67.

[701] GJ Simmons, "How to Insure That Data Acquired to Verify Treaty Compliance Are Trustworthy," in *Proceedings of the IEEE*, v 76 no 5 (1988); reprinted as a chapter in [702]).

[702] GJ Simmons (ed), *Contemporary Cryptology—The Science of Information Integrity*, IEEE Press (1992), ISBN 0-87942-277-7.

[703] GJ Simmons, "A Survey of Information Authentication," in [702], pp 379–439.

[704] GJ Simmons, "An Introduction to Shared Secret and/or Shared Control Schemes and Their Application," in [702], pp 441–497.

[705] GJ Simmons, invited talk at the *1993 ACM Conference on Computer and Communications Security*, Fairfax, VA (Nov 3–5, 1993).

[706] GJ Simmons, "Subliminal Channels: Past and Present," in *European Transactions on Telecommunications*, v 5 no 4 (July/Aug 1994), pp 459–473.

[707] GJ Simmons, "The History of Subliminal Channels," in *IEEE Journal on Selected Areas in Communications*, v 16 no 4 (April 1998), pp 452–462.

[708] DR Simon, "Anonymous Communication and Anonymous Cash," in *Advances in Cryptology—Crypto 96*, Springer LNCS, v 1109, pp 61–73.

[709] WA Simpson, "Electronic Signatures Yield Unpleasant Surprises," (June 23, 2000), at `http://cryptome.org/esigs-suck.htm`.

[710] A Sipress, "Tracking Traffic by Cell Phone Maryland, Virginia to Use Transmissions to Pinpoint Congestion," in *The Washington Post* (Dec 22, 1999), p A1, at `http://www.washingtonpost.com/`.

[711] KS Siyan, J Casad, J Millecan, D Yarashus, P Tso, J Shoults, *Windows NT Server 4—Professional Reference*, New Riders Publishing (1996).

[712] SP Skorobogatov, "Low Temperature Remanence in Static RAM" (*to appear*).

[713] Smartcard Developer Association, `http://www.scard.org/gsm/`.

[714] "Plastic Card Fraud Rises in the UK," in *Smart Card News*, v 6 no 3 (Mar 1997), p 45.

[715] RE Smith, "Constructing a High-Assurance Mail Guard," in *17th National Computer Security Conference* (Oct 11–14, 1994), Baltimore, MD; proceedings published by NIST, pp 247–253.

[716] RM Smith, "Problems with Web Anonymizing Services," (Apr 15, 1999), at `http://www.tiac.net/users/smiths/anon/anonprob.htm`.

[717] SP Smith, H Perrit, H Krent, S Mencik, JA Crider, MF Shyong, LL Reynolds, *Independent Technical Committee Review of the Carnivore System—Draft report*, U.S. Department of Justice Contract No. 00-C-0238 IITRI, CR-022-216 (Nov 17, 2000), at `http://cryptome.org/carnivore.rev.htm`.

[718] S Smith, S Weingart, "Building a High-Performance, Programmable Secure Coprocessor," IBM Technical report RC 21102, available at `http://www.ibm.com/security/cryptocards/`.

[719] Peter Smulders, "The Threat of Information Theft by Reception of Electromagnetic Radiation from RS-232 Cables," in *Computers & Security*, v 9 (1990), pp 53–58.

[720] A Solomon, "A Brief History of PC Viruses," in *Computer Fraud and Security Bulletin* (Dec 1993), pp 9–19.

[721] A Solomon, Seminar given at Cambridge University Computer Laboratory (May 30, 2000).

[722] P Sommer, "Intrusion Detection and Legal Proceedings," in *Recent Advances in Intrustion Detection* (RAID) (1998), at `http://www.zurich.ibm.com/~dac/Prog_RAID98/Full_Papers/Sommer_text.pdf`.

[723] South West Thames Regional Health Authority, "Report of the Inquiry into the London Ambulance Service" (1993), at `http://www.cs.ucl.ac.uk/staff/A.Finkelstein/las.html`.

[724] E Spafford, "The Internet Worm Program: An Analysis," in *Computer Communications Review*, v 19 no 1 (Jan 1989), pp 17–57.

[725] EH Spafford, "OPUS: Preventing Weak Password Choices," in *Computers and Security*, v 11 no 3 (1992), pp 273–278.

[726] "Tip von Urmel," in *Spiegel Magazine*, no 38 (Sept 11, 1995).

[727] J Spolsky, "Does Issuing Passports Make Microsoft a Country?" at `http://joel.editthispage.com/stories/storyReader$139`.

[728] "Your Car Radio May Be Revealing Your Tastes," in *St. Petersburg Times* (Jan 31, 2000), at `http://www.sptimes.com/News/013100/Technology/Your_car_radio_may_be.shtml`.

[729] T Standage, *The Victorian Internet*, Phoenix Press (1999), ISBN 0-75380-703-3.

[730] F Stajano, personal communication with the author.

[731] F Stajano, RJ Anderson, "The Resurrecting Duckling: Security Issues in Ad-Hoc Wireless Networks," in *Security Protocols—7th International Workshop*, Springer LNCS, v 1796, pp 172–182.

[732] F Stajano, RJ Anderson, "The Cocaine Auction Protocol—On the Power of Anonymous Broadcast," in [613], pp 434–447.

[733] WA Steer, "VideoDeCrypt," at `http://www.ucl.ac.uk/~ucapwas/vdc/`.

[734] P Stein, P Feaver, *Assuring Control of Nuclear Weapons*, University Press (1987) quoted in [92].

[735] J Steiner, BC Neuman, JI Schiller, "Kerberos: An Authentication Service for Open Network Systems," in *USENIX* (Winter 1988); version 5 in "RFC 1510: The Kerberos Network Authentication Service (V5)"; at `http://sunsite.utk.edu/net/security/kerberos/`.

[736] N Stephenson, *Snow Crash*, New York: Bantam Doubleday Dell (1992), ISBN 0-553-38095-8.

[737] FA Stevenson, "Cryptanalysis of Contents Scrambling System," at `http://www.derfrosch.de/decss/`.

[738] DR Stinson, *Cryptography—Theory and Practice*, CRC Press (1995); ISBN 0-8493-8521-0.

[739] "Watching Them, Watching Us—UK CCTV Surveillance Regulation Campaign," at http://www.spy.org.uk/.

[740] R Strehle, *Verschlüsselt—Der Fall Hans Bühler*, Werd Verlag (1994), ISBN 3-85932-141-2.

[741] K Stumper, "DNA-Analysen und ein Recht auf Nichtwissen," in *Datenschutz und Datensicherheit*, v 19 no 9 (Sept 1995), pp 511–517.

[742] Suetonius (Gaius Suetonius Tranquillus), *Vitae XII Caesarum, translated into English as History of Twelve Caesars*, by Philemon Holland, 1606; Nutt (1899).

[743] D Sutherland, "A Model of Information," in *9th National Computer Security Conference (1986)*, pp 175–183.

[744] L Sweeney, "Weaving Technology and Policy Together to Maintain Confidentiality," in *Journal of Law, Medicine, and Ethics*, v 25 nos 2–3 (1997), pp 98–110.

[745] S Tendler, N Nuttall, "Hackers Run Up £1m Bill on Yard's Phones," in *The London Times* (Aug 5, 1996); at http://www.the-times.co.uk/.

[746] K Thompson, "Reflections on Trusting Trust," in *Communications of the ACM*, v 27 no 8 (Aug 1984), pp 761–763; at http://www.acm.org/classics/sep95/.

[747] J Ticehurst, "Barclays Online Bank Suffers Another Blow" (Aug 11, 2000), at http://www.vnunet.com/News/1108767.

[748] AZ Tirkel, GA Rankin, RM van Schyndel, WJ Ho, NRA Mee, CF Osborne, "Electronic Watermark," in *Digital Image Computing, Technology, and Applications* (DICTA 93) McQuarie University (1993), pp 666–673.

[749] JW Toigo, *Disaster Recovery Planning for Computers and Communication Resources*, New York: John Wiley & Sons, Inc. (1996), ISBN 0-471-12175-4.

[750] C Tomlinson, "*Rudimentary Treatise on the Construction of Locks*," (1853) excerpt at http://www.deter.com/unix/papers/treatise_locks.html.

[751] Transactional Records Access Clearinghouse, "TRACFBI," at http://trac.syr.edu/tracfbi/index.html.

[752] M Trombly, "VISA Issues 10 'Commandments' for Online Merchants," in *Computerworld* (Aug 11, 2000), at http://www.computerworld.com/cwi/story/0,1199,NAV47_STO48487,00.html.

[753] JD Tygar, BS Yee, N Heintze, "Cryptographic Postage Indicia," in *Concurrency and Parallelism, Programming, Networking, and Security*, Springer-Verlag, (Dec 1996), pp 378–391, at http://buffy.eecs.berkeley.edu/~tygar/recommend.html.

[754] R Uhlig, "BT Admits Staff Could Have Fiddled System to Win Concorde Trip," in *The Daily Telegraph* (July 23, 1997), at http://www.telegraph.co.uk:80/.

[755] ukcrypto mailing list, at http://www.chiark.greenend.org.uk/mailman/listinfo/ukcrypto.

[756] Underwriters' Laboratories, http://www.ul.com.

[757] J Ungoed-Thomas, A Lorenz, "French Play Dirty for £1bn Tank Deal," in *The Sunday Times* (Aug 6, 2000), p 5.

[758] United Kingdon Government, "e-commerce@its.best.uk," at http://www.e-envoy.gov.uk/2000/strategy/strategy.htm.

[759] United States Code—U.S. Federal Law, online for example at http://www4.law.cornell.edu/uscode/.

[760] United States Court of Appeals, District of Columbia Circuit, *United States Telecom Association v. Federal Communications Commission and United States of America*, No. 99-1442 (Aug 15, 2000), at http://pacer.cadc.uscourts.gov/common/opinions/200008/99-1442a.txt.

[761] United States Senate Select Committee on Intelligence, *CIA Office of Inspector General Investigations Staff Report on the Improper Handling of Classified Information by John M. Deutch*, 106th Congress, at http://intelligence.senate.gov/igreport.pdf.

[762] UPI newswire item, Oklahoma distribution (Nov 26, 1983), Tulsa, OK.

[763] L van Hove, "Electronic Purses: (Which) Way to Go?" in *First Monday*, v 5 no 7 (June 2000), at http://firstmonday.org/issues/issue5_7/hove/.

[764] P van Oorschot, M Wiener, "Parallel Collision Search with Application to Hash Functions and Discrete Logarithms," *Second ACM Conference on Computer and Communications Security*; proceedings published by the ACM, ISBN 0-89791-732-4, pp 210–218.

[765] R van Renesse, *Optical Document Security* (2nd ed), Artech House (1997), ISBN 0-89006-982-4.

[766] R van Renesse, "Verifying versus Falsifying Banknotes," in *Optical Security and Counterfeit Deterrence Techniques II* (1998), IS&T (The Society for Imaging Science and Technology) and SPIE (The International Society for Optical Engineering), v 3314, ISBN 0-8194-2754-3, pp 71–85.

[767] H van Vliet, *Software Engineering—Principles and Practice*, 2nd ed, New York: John Wiley & Sons, Inc. (2000), ISBN 0-471-97508-7.

[768] R van Voris, "Black Box Car Idea Opens Can of Worms," in *Law News Network* (June 4, 1999), at http://www.lawnewsnetwork.com/stories/A2024-1999Jun4.html.

[769] G Vanneste, J Degraeve, "Initial Report on Security Requirements," in [56].

[770] HR Varian, *Intermediate Microeconomics—A Modern Approach*, 5 ed, New York: W. W. Norton (1999), ISBN 0-393-97370-0.

[771] HR Varian, "Managing Online Security Risks," in *The New York Times* (June 1, 2000); at http://www.nytimes.com/library/financial/columns/060100econ-scene.html.

[772] V Varadharajan, N Kumar, Y Mu, "Security Agent-Based Distributed Authorization: An Approach," in *20th National Information Systems Security Conference;* proceedings published by NIST (1998), pp 315–328.

[773] S Vaudenay, "FFT-Hash-II Is Not Yet Collision-Free," in *Laboratoire d'Informatique de l'Ecole Normale Supérieure report LIENS-92-17.*

[774] W Venema, "Murphy's Law and Computer Security," in *USENIX Security 96*, pp 187–193.

[775] B Vinck, "Security Architecture" (3G TS 33.102 v 3.2.0), from *Third-Generation Partnership Project*, at `http://www.3gpp.org/TSG/Oct_status_list.htm`.

[776] B Vinck, "Lawful Interception Requirements"(3G TS 33.106 v 3.0.0), from *Third-Generation Partnership Project*, at `http://www.3gpp.org/TSG/Oct_status_list.htm`.

[777] VISA International, "Integrated Circuit Chip Card—Security Guidelines Summary," version 2 draft 1 (Nov 1997).

[778] A Viterbi, "Spread-Spectrum Communications—Myths and Realities," in *IEEE Communications Magazine*, v 17 no 3 (May 1979), pp 11–18.

[779] PR Vizcaya, LA Gerhardt, "A Nonlinear Orientation Model for Global Description of Fingerprints," in *Pattern Recognition*, v 29 no 7 (July 1996), pp 1221–1231.

[780] D Wagner, B Schneier, J Kelsey, "Cryptanalysis of the Cellular Message Encryption Algorithm," in *Advances in Cryptology—Crypto 95*, Springer LNCS, v 1294, pp 527–537.

[781] D Wagner, "Cryptanalysis of Some Recently Proposed Multiple Modes of Operation," in *Fifth International Workshop on Fast Software Encryption* (1998), Springer LNCS, v 1372, pp 254–269.

[782] DA Wagner, SM Bellovin, "A 'Bump in the Stack' Encryptor for MS-DOS Systems," in *Proceedings of the Internet Society Symposium on Network and Distributed System Security* (1996); proceedings published by the IEEE, ISBN 0-8186-7222-6, pp 155–160.

[783] D Wagner, I Goldberg, M Briceno, "GSM Cloning," at `http://www.isaac.cs.berkeley.edu/isaac/gsm-faq.html`; see also `http://www.scard.org/gsm/`.

[784] D Wagner, B Schneier, "Analysis of the SSL 3.0 Protocol," in *Second USENIX Workshop on Electronic Commerce* (1996), pp 29–40; at `http://www.counterpane.com`.

[785] M Waldman, AD Rubin, LF Cranor, "Publius: A Robust, Tamper-Evident, Censorship-Resistant, Web Publishing System," in *9th USENIX Security Symposium* (2000), pp 59–72.

[786] M Walker, "On the Security of 3GPP Networks," invited talk at Eurocrypt 2000, at `http://www.ieee-security.org/Cipher/ConfReports/2000/CR2000-Eurocrypt.html`.

[787] G Walsh, "Review of Policy Relating to Encryption Technologies" (1996), at http://www.efa.org.au/Issues/Crypto/Walsh/.

[788] KG Walter, WF Ogden, WC Rounds, FT Bradshaw, SR Ames, DG Shumway, "Models for Secure Computer Systems," Case Western Reserve University, Report No. 1137 (July 31, 1973, revised Nov 21, 1973).

[789] KG Walter, WF Ogden, WC Rounds, FT Bradshaw, SR Ames, DG Shumway, "Primitive Models for Computer Security," Case Western Reserve University, Report No. ESD-TR-74-117 (Jan 23, 1974); at http://www.dtic.mil.

[790] E Waltz, *Information Warfare—Principles and Operations*, Artech House (1998), ISBN 0-89006-511-X.

[791] W Ware, "Security Controls for Computer Systems: Report of Defense Science Board Task Force on Computer Security," Rand Report R609-1, The RAND Corporation, Santa Monica, CA (Feb 1970); at http://csrc.nist.gov/publications/history/index.html.

[792] SD Warren, LD Brandeis, "The Right to Privacy," *Harvard Law Review*, series 4 (1890), pp 193–195.

[793] M Weaver, "Developer Tortured by Raiders with Crowbars," *Daily Telegraph* (Oct 31, 1997).

[794] W Webb, "High-Tech Security: The Eyes Have It," in *EDN* (Dec 18, 1997), pp 75–78.

[795] SH Weingart, "Physical Security for the μABYSS System," in *Proceedings of the 1987 IEEE Symposium on Security and Privacy*, IEEE Computer Society Press, pp 52–58.

[796] SH Weingart, SR White, WC Arnold, GP Double, "An Evaluation System for the Physical Security of Computing Systems," in *Sixth Annual Computer Security Applications Conference* (Dec 3–7, 1990), Tucson, AZ; proceedings published by the IEEE (1990), pp 232–243.

[797] L Weinstein, "IDs in Color Copies—A PRIVACY Forum Special Report," in *Privacy Forum Digest*, v 8 no 18 (Dec 6, 1999), at http://www.vortex.com/privacy/priv.08.18.

[798] C Weissman, "Security Controls in the ADEPT-50 Time-Sharing System," in *AFIPS Conference Proceedings*, v 35, 1969 Fall Joint Computer Conference, pp 119–133.

[799] C Weissman, "BLACKER: Security for the DDN, Examples of A1 Security Engineering Trades," in *Proceedings of the 1992 IEEE Symposium on Security and Privacy*, pp 286–292.

[800] G Welchman, *The Hut Six Story*, New York: McGraw-Hill (1982), ISBN 0-07-069180-0.

[801] A Westfeld, A Pfitzmann, "Attacks on Steganographic Systems," in *Proceedings of the Third International Workshop on Information Hiding* (1999), Springer LNCS, v 1768, pp 61–76.

[802] AF Westin, "Data Protection in the Global Society" (1996 conference report), at http://www.privacyexchange.org/iss/confpro/aicgsberlin.html.

[803] A Whitten, "Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0," in *Eighth USENIX Security Symposium*, proceedings ISBN 1-880446-28-6, pp 169–183.

[804] MV Wilkes, RM Needham, *The Cambridge CAP Computer and Its Operating System*, Elsevier North Holland (1979).

[805] J Wilkins, *Mercury; or the Secret and Swift Messenger: Shewing, How a Man May, with Privacy and Speed Communicate His Thoughts to a Friend at Any Distance*, London: Rich Baldwin (1694).

[806] FW Winterbotham, *The Ultra Secret*, New York: Harper & Row (1974).

[807] K Wong, "Mobile Phone Fraud—Are GSM Networks Secure?" in *Computer Fraud and Security Bulletin* (Nov 1996), pp 11–18.

[808] CC Wood, "Identity Token Usage at American Commercial Banks," in *Computer Fraud and Security Bulletin* (Mar 1995), pp 14–16.

[809] JPL Woodward, "Security Requirements for System High and Compartmented Mode Workstations," Mitre MTR 9992, Revision 1 (1987); also published by the Defense Intelligence Agency as document DDS-2600-5502-87.

[810] B Wright, "The Verdict on Plaintext Signatures: They're Legal," in *Computer Law and Security Report*, v 14 no 6 (Nov/Dec 1994), pp 311–312.

[811] B Wright, *The Law of Electronic Commerce: EDI, Fax and Email*, New York: Little Brown (1991); 4th ed, (with supplement) 1994.

[812] JB Wright, "Report of the Weaponization and Weapons Production and Military Use Working Group," Appendix F to the Report of the Fundamental Classification Policy Review Group, U.S. Department of Energy Office of Scientific and Technical Information (1997), http://www.osti.gov/opennet/app-f.html.

[813] MA Wright, "Security Controls in ATM Systems," in *Computer Fraud and Security Bulletin* (Nov 1991), pp 11–14.

[814] P Wright, *Spycatcher—The Candid Autobiography of a Senior Intelligence Officer*, Australia: William Heinemann (1987), ISBN 0-85561-098-0.

[815] JX Yan, A Blackwell, RJ Anderson, A Grant, "The Memorability and Security of Passwords—Some Empirical Results," University of Cambridge Computer Laboratory Technical Report no 500; at http://www.cl.cam.ac.uk/ftp/users/rja14/tr500.pdf.

[816] JX Yan, S Early, R Anderson, "The XenoService—A Distributed Defeat for Distributed Denial of Service," Third Information Survivability Workshop (Oct 2000), at http://www.cl.cam.ac.uk/users/rja14/.

[817] T Ylönen, "SSH—Secure Login Connections over the Internet," in *USENIX Security 96*, pp 37–42.

[818] KS Yoon, YK Ham, RH Park, "Hybrid Approaches to Fractal Face Recognition Using the Hidden Markov Model and Neural Network," in *Pattern Recognition*, v 31 no 3 (1998), pp 283–293.

[819] G Yuval, "Reinventing the Travois: Encryption/MAC in 30 ROM Bytes," in *Fourth International Workshop on Fast Software Encryption* (1997), Springer LNCS, v 1267, pp 205–209.

[820] MC Zari, AF Zwilling, DA Hess, KW Snow, CJ Anderson, D Chiang, "Personal Identification System Utilizing Low Probability of Intercept (LPI) Techniques for Covert Ops," in *30th Annual IEEE Carnahan Conference on Security Technology* (1996), pp 1–6.

[821] Zero Knowledge Systems Inc., `http://www.zeroknowledge.com/`.

[822] MW Zior, "A Community Response to CMM-Based Security Engineering Process Improvement," in *18th National Information Systems Security Conference* (1995), pp 404–413.

[823] M Zviran, WJ Haga, "A Comparison of Password Techniques for Multilevel Authentication Mechanisms," in *The Computer Journal*, v 36 no 3 (1993), pp 227–237.