北京市高等教育精品教材立项项目

高等学校规划教材

信息安全原理与应用

王　昭　袁　春　编著

陈　钟　审校

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内 容 简 介

本教材涉及密码编码与网络安全从技术到管理的方方面面，以数据机密性、数据完整性、不可否认性、鉴别和访问控制五大类安全服务和安全模型为线索，介绍了信息安全的基本原理。以密码编码与密码分析相结合的思路，比较完整地介绍了密码编码学的基本原理和算法实现，包括：古典密码、现代对称密码、公钥密码和散列函数，并讨论了密码算法实际应用中的一些问题，如密钥长度、密钥管理、硬件加密和软件加密，以及算法应用中曾经出现的教训等。在此基础上，介绍了相关综合应用，包括电子邮件的安全、网络安全协议和数据库安全。在网络安全与系统安全方面讨论了网络入侵与攻击、入侵检测、防火墙和计算机病毒防范。此外也介绍了信息安全的一些标准化情况，包括标准化机构和信息安全的评估标准。本书不仅介绍网络安全的基本原理，更注重理论与实际的结合，在相关章节后附有一些加深理论理解的难易程度不同的思考练习题和实践/实验题。

本书可作为信息类专业高年级本科生和研究生教材，也可以为信息安全、计算机、通信和电子工程等领域研究和开发人员提供有益的帮助和参考。

图书在版编目(CIP)数据

信息安全原理与应用/王昭，袁春编著．—北京：电子工业出版社，2010.1

高等学校规划教材

ISBN 978-7-121-09887-1

Ⅰ．信…　Ⅱ．①王…②袁…　Ⅲ．信息系统－安全技术　Ⅳ．TP309

中国版本图书馆 CIP 数据核字(2009)第 209308 号

策划编辑：冯小贝

责任编辑：李秦华

印　　刷：

装　　订：

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱　邮编　100036

开　　本：787×1092　1/16　印张：21　字数：538 千字

印　　次：2012 年 9 月第 2 次印刷

定　　价：32.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：(010)88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：(010)88258888。

前　言

信息安全是一门跨学科跨专业的综合性学科，它涵盖了非常丰富的内容，涉及数论、密码编码、信息论、通信、网络、编程等多方面的知识，无论是从事管理，还是技术研发的人员，甚至普通的计算机用户，都需要从不同层次和角度了解这方面的基本知识。并且随着信息技术的发展，信息安全新技术新思想不断涌现。此外，它还是一门理论与实际紧密结合的学科。

本书主要是以作者多年来在北京大学讲授信息安全、应用密码学等课程讲义为基础编写而成的。编写中，我们力求做到内容的系统、完整和深入浅出，理论与实际的结合，原理的经典性和技术的先进性。

信息安全问题的解决方案可以分为两类，一类是以密码编码为基础的解决方案，另一类是和密码无关的一些解决方案。本书尽可能全面地涵盖这两类原理和技术，主要内容安排如下：

第 1 章介绍了 ISO 7498—2 定义的 OSI 的五大类安全服务：数据机密性、数据完整性、抗抵赖、鉴别和访问控制。本书以经典的通信安全模型和信息访问安全模型为线索，介绍了这五大类安全服务。

第 2～5 章以密码分析和密码编码相结合的思路，比较完整地介绍了密码编码学的基本原理和算法实现，包括：古典密码、现代对称密码、公钥密码和散列函数。密码算法都以国际上经典的标准或最新的标准为例。在原理介绍的基础上，第 6 章和第 7 章讨论了密码算法实际应用中的一些问题，包括：密钥长度、密钥管理、硬件加密和软件加密，以及算法应用中曾经出现的教训等。

第 8 章、第 10 章和第 11 章介绍了密码编码的相关综合应用，包括鉴别协议、安全电子邮件和网络安全协议，其中的内容都以最新的 RFC 文档和相关文献资料为参考。

第 9 章和第 12～15 章主要讨论了与密码算法无关的安全解决方案，包括访问控制、防火墙技术、黑客攻击与防范技术、计算机病毒防治和入侵检测技术。

第 16 章介绍了信息安全的一些标准化情况，包括信息安全的标准化机构和有关标准。

最后，第 17 章介绍了一个综合应用信息安全有关原理的实例——数据库系统安全。

在相关章节后附有一些加深理论理解的难易程度不同的思考和练习题、实践/实验题，以帮助读者更深入和扎实地掌握相关知识。

根据编者经验，主要内容的课堂讲授需要 50 学时左右，也可根据教学对象和教学目标进行删减，建议根据课程内容再安排一定学时的课外实践/实验。

在本书的编写过程中，查阅和参考了大量文献资料，限于篇幅未能在书后的参考文献中一一列出，在此一并致谢。

本书第 1～16 章由北京大学信息科学技术学院的王昭老师编写，第 17 章由清华大学深圳研究生院的袁春老师编写，全书最后由王昭老师统稿。在编写过程中，得到了解放军某部南湘浩研究员、中科院研究生院翟起滨教授和北京大学信息科学技术学院屈婉玲教授的多次热情指导和帮助。北京大学信息科学技术学院唐礼勇副教授、胡建斌副教授、关志博士、软件与微电子学院沈晴霓副教授、周继军副教授等老师也为本书的编写提供了相关资料和帮

助。北京大学信息科学技术学院的本科生李翔宇、研究生周光明、刘国鹏、陈宇、王永刚、刘勇、金永明、桂尼克参与了书稿的校对、资料收集等工作。本书是北京市精品教材建设立项项目，也得到了电子工业出版社的大力支持。在此表示衷心的感谢。

北京大学信息科学技术学院陈钟教授自始至终关心本书的编写工作。成书后，陈老师又认真审阅了全部书稿，在此致以特别谢意。

信息安全是一个不断发展的领域，由于编者水平有限，书中错误和不当之处在所难免，敬请广大读者和同行专家批评指正，在此先致感谢之意。

为了配合教学，本书还提供与教材配套的电子课件。

编者

2009 年 10 月

目　录

第 1 章　绪论 …… 1
1.1　信息和信息安全的概念 …… 1
1.1.1　信息的定义 …… 1
1.1.2　信息的属性和价值 …… 1
1.1.3　信息安全的含义 …… 2
1.2　信息安全的威胁 …… 2
1.3　安全服务 …… 3
1.3.1　数据机密性 …… 4
1.3.2　数据完整性 …… 4
1.3.3　抗抵赖 …… 4
1.3.4　鉴别 …… 5
1.3.5　访问控制 …… 5
1.3.6　OSI 安全服务的分层配置 …… 5
1.4　信息安全模型 …… 6
1.4.1　通信安全模型 …… 6
1.4.2　信息访问安全模型 …… 6
1.4.3　动态安全模型 …… 7
1.5　信息安全的技术体系 …… 7
1.6　信息安全的政策法规 …… 8
1.6.1　国际信息安全政策法规 …… 8
1.6.2　国内信息安全政策法规 …… 9
1.7　信息安全的相关机构和相关标准 …… 10
1.7.1　国际标准化机构 …… 10
1.7.2　美国的标准化机构 …… 11
1.7.3　信息安全组织机构 …… 12
1.7.4　国内标准制定情况 …… 12
思考和练习题 …… 13

第 2 章　密码学基础 …… 14
2.1　密码学的基本概念和术语 …… 14
2.1.1　消息和加密 …… 14
2.1.2　恺撒密表 …… 15
2.1.3　密码体制 …… 15
2.1.4　密码算法的分类 …… 16
2.1.5　密码分析 …… 17

2.1.6 密码算法的安全性 …… 17
2.2 密码学的历史 …… 18
2.3 古典密码 …… 20
2.3.1 古典密码的数学基础 …… 20
2.3.2 代替密码 …… 22
2.3.3 置换密码 …… 33
2.3.4 古典密码算法小结 …… 33
思考和练习题 …… 34
实践/实验题 …… 34

第 3 章 现代对称密码 …… 35
3.1 乘积密码 …… 35
3.2 对称分组密码的设计原理与方法 …… 36
3.2.1 对称分组密码的三个安全设计原则 …… 36
3.2.2 对称分组密码的两个基本设计方法 …… 37
3.3 数据加密标准 DES …… 37
3.3.1 DES 的产生与应用 …… 37
3.3.2 Feistel 密码结构 …… 38
3.3.3 对 DES 的描述 …… 39
3.3.4 对 DES 的讨论 …… 44
3.4 三重 DES …… 46
3.4.1 双重 DES …… 46
3.4.2 三重 DES …… 47
3.5 高级数据加密标准 AES …… 48
3.5.1 AES 的背景 …… 48
3.5.2 AES 的数学基础 …… 49
3.5.3 对 AES 的描述 …… 52
3.6 分组密码的工作模式 …… 58
3.6.1 电码本(ECB)工作模式 …… 59
3.6.2 密码分组链接(CBC)工作模式 …… 59
3.6.3 密码反馈(CFB)工作模式 …… 61
3.6.4 输出反馈(OFB)模式 …… 62
3.6.5 计数器(CTR)模式 …… 63
3.6.6 不是分组长度整数倍的报文的处理 …… 64
3.6.7 三重 DES 的工作模式 …… 65
3.7 流密码 …… 66
3.7.1 流密码的定义 …… 66
3.7.2 同步流密码 …… 66
3.7.3 密钥流生成器 …… 67
3.7.4 RC4 …… 69

3.7.5 A5 算法 …… 70
3.8 随机数 …… 70
3.8.1 真随机序列产生器 …… 71
3.8.2 伪随机数产生器 …… 72
3.8.3 基于密码编码方法的随机数 …… 73
思考和练习题 …… 73
实践/实验题 …… 73

第 4 章 公钥密码 …… 74
4.1 公钥密码体制的基本原理 …… 74
4.1.1 公钥密码体制的概念 …… 74
4.1.2 公钥密码体制的应用 …… 75
4.1.3 公钥密码体制的思想和要求 …… 76
4.1.4 公钥密码体制的安全性 …… 76
4.2 公钥密码算法的数学基础 …… 77
4.2.1 若干基本定理 …… 78
4.2.2 离散对数难题 …… 80
4.3 Diffie-Hellman 密钥交换算法 …… 81
4.3.1 对 Diffie-Hellman 密钥交换算法的描述 …… 81
4.3.2 对 Diffie-Hellman 密钥交换的攻击 …… 82
4.4 背包算法 …… 82
4.4.1 背包问题和背包算法的思想 …… 82
4.4.2 超递增背包 …… 83
4.4.3 转换背包 …… 83
4.4.4 Merkle-Hellman 公钥算法 …… 83
4.5 RSA 算法 …… 84
4.5.1 RSA 算法描述 …… 84
4.5.2 RSA 实现中的问题 …… 86
4.5.3 RSA 的安全性 …… 87
4.5.4 对 RSA 实现的攻击方法 …… 88
4.6 EIGamal 算法 …… 90
4.7 椭圆曲线密码算法(ECC) …… 91
4.7.1 椭圆曲线的概念 …… 91
4.7.2 有限域上的椭圆曲线 …… 92
4.7.3 椭圆曲线密码算法 …… 93
4.8 密码算法小结 …… 95
思考和练习题 …… 96
实践/实验题 …… 96

第 5 章 消息鉴别和数字签名 …… 97
5.1 消息鉴别 …… 97

5.1.1 鉴别系统模型 …… 97
5.1.2 消息加密 …… 98
5.1.3 消息鉴别码 MAC …… 99
5.1.4 散列函数 …… 100
5.2 散列算法 …… 105
5.3 HMAC …… 106
5.4 数字签名 …… 107
5.4.1 数字签名的功能与特性 …… 107
5.4.2 数字签名方案 …… 109
思考和练习题 …… 113
实践/实验题 …… 113

第 6 章 密码实际应用问题 …… 114
6.1 密码功能的位置 …… 114
6.2 密钥管理 …… 115
6.2.1 密钥的类型 …… 116
6.2.2 密钥的产生和登记 …… 117
6.2.3 密钥的装入 …… 118
6.2.4 密钥的存储和保护 …… 118
6.2.5 密钥的分配 …… 119
6.2.6 密钥的使用控制 …… 124
6.2.7 密钥的撤销和销毁 …… 124
6.2.8 密钥的备份/恢复和更新 …… 124
6.3 密钥的长度 …… 125
6.3.1 对称算法的密钥长度 …… 125
6.3.2 公开密钥密码体制的密钥长度 …… 126
6.3.3 密码体制密钥长度的对比 …… 127
6.4 硬件加密和软件加密 …… 127
6.5 存储数据加密的特点 …… 128
6.6 压缩、纠错编码和加密 …… 128
6.7 文件删除 …… 128
6.8 关于密码的一些教训 …… 129
6.8.1 声称你的算法是“不可攻破的” …… 129
6.8.2 多次使用一次性密码本 …… 129
6.8.3 没有使用最好的可能算法 …… 129
6.8.4 没有正确实现算法 …… 129
6.8.5 在产品中放置了后门 …… 129
思考和练习题 …… 130
实践/实验题 …… 130

第 7 章　公钥管理技术 ……131
7.1　公开密钥基础设施 ……131
7.1.1　PKI 概述 ……131
7.1.2　数字证书 ……133
7.1.3　CA 的组成 ……136
7.1.4　密钥和证书的生命周期 ……137
7.1.5　PKI 信任模型 ……138
7.1.6　PKI 发展中的问题 ……142
7.2　基于身份的密码学 ……142
7.2.1　基于身份的密码学原理 ……142
7.2.2　IBC 的方案 ……143
7.2.3　IBC 的实际问题 ……144
7.3　ECC 组合公钥体制 ……145
7.3.1　CPK 相关概念 ……146
7.3.2　ECC 复合定理 ……146
7.3.3　标识密钥 ……146
7.3.4　密钥的复合 ……147
7.3.5　CPK 数字签名 ……148
7.3.6　CPK 密钥交换 ……148
7.3.7　安全性分析 ……149
7.3.8　ECC CPK 小结 ……150
思考和练习题 ……150
实践/实验题 ……150

第 8 章　鉴别协议 ……151
8.1　鉴别的相关概念 ……151
8.2　密码协议 ……151
8.3　实体鉴别概述 ……152
8.3.1　实体鉴别的基本概念 ……152
8.3.2　实体鉴别和消息鉴别的区别和联系 ……153
8.3.3　实体鉴别实现安全目标的方式 ……153
8.3.4　实体鉴别的分类 ……153
8.3.5　实体鉴别系统的组成 ……153
8.3.6　实现身份鉴别系统的途径和要求 ……154
8.4　鉴别机制 ……154
8.4.1　口令机制 ……155
8.4.2　一次性口令机制 ……157
8.4.3　基于密码算法的鉴别机制 ……158
8.4.4　零知识证明协议 ……159
8.4.5　基于地址的机制 ……159

8.4.6 基于设备的鉴别……159
8.4.7 基于个人特征的机制……160
8.5 鉴别与密钥交换协议设计中的问题……160
8.6 鉴别与交换协议实例……161
8.6.1 CHAP 协议……161
8.6.2 S/KEY 协议……162
8.6.3 Kerberos……163
8.6.4 X.509 鉴别服务……166
思考和练习题……167
实践/实验题……167

第 9 章 访问控制……168
9.1 访问控制的有关概念……168
9.2 自主访问控制……170
9.2.1 访问控制表……170
9.2.2 能力表……171
9.2.3 DAC 的授权管理……172
9.3 强制访问控制 MAC……173
9.3.1 Bell-LaPadula 模型……174
9.3.2 Biba 模型……174
9.4 基于角色的访问控制 RBAC……175
9.4.1 RBAC 的概念和安全原则……175
9.4.2 NIST-RBAC 参考模型……176
9.5 其他访问控制策略……178
9.5.1 使用控制……178
9.5.2 基于任务的访问控制……179
9.5.3 基于属性的访问控制……180
9.6 Windows 2000/XP 的访问控制机制……181
9.7 Linux 系统的访问控制机制……182
思考和练习题……183
实践/实验题……183

第 10 章 安全电子邮件……184
10.1 电子邮件原理……184
10.2 PGP……185
10.2.1 使用 PGP 保护电子通信……186
10.2.2 PGP 的密钥和密钥管理……189
10.2.3 PGP 的其他功能……192
10.3 S/MIME……193
10.3.1 RFC822……193
10.3.2 MIME……193

10.3.3 S/MIME ……194
思考和练习题……194
实践/实验题……195

第 11 章 网络安全协议 ……196
11.1 TCP/IP 基础 ……196
11.1.1 TCP/IP 的历史 ……196
11.1.2 TCP/IP 层次模型……196
11.1.3 IPv4 协议 ……198
11.1.4 IPv6 数据报……200
11.1.5 ARP 协议……201
11.1.6 ICMP 协议 ……202
11.1.7 TCP 协议 ……203
11.1.8 UDP 协议……204
11.1.9 TCP 和 UDP 端口……204
11.2 Internet 安全性途径 ……205
11.3 IP 的安全……206
11.3.1 IPsec 概述 ……206
11.3.2 IPsec 的文档组成 ……207
11.3.3 安全关联……208
11.3.4 鉴别头协议 ……209
11.3.5 封装安全载荷协议 ……211
11.3.6 安全关联组合 ……214
11.3.7 密钥管理……215
11.4 SSL/TLS……215
11.4.1 TLS 的体系结构……216
11.4.2 TLS 的记录协议……216
11.4.3 修改密码规范协议 ……219
11.4.4 警报协议……219
11.4.5 TLS 的握手协议……219
11.4.6 TLS 的实现 ……220
思考和练习题……221
实践/实验题……221

第 12 章 防火墙技术及应用 ……222
12.1 防火墙概述 ……222
12.1.1 防火墙的基本概念 ……222
12.1.2 防火墙的作用和局限性……222
12.1.3 防火墙的安全策略 ……223
12.2 防火墙的体系结构……223
12.2.1 包过滤型防火墙……223

12.2.2 双宿/多宿主机模式……224
12.2.3 屏蔽主机模式……224
12.2.4 屏蔽子网模式……224
12.3 防火墙相关技术……225
12.3.1 静态包过滤防火墙……225
12.3.2 状态监测防火墙……229
12.3.3 应用级网关防火墙……230
12.3.4 电路级网关防火墙……231
12.3.5 深度包检查技术……231
12.3.6 分布式防火墙……232
12.3.7 其他防火墙技术……233
12.4 防火墙的实现和维护……233
12.5 总结和展望……234
思考和练习题……234
实践/实验题……234

第 13 章 黑客攻击与防范技术……235
13.1 认识黑客……235
13.2 攻击的概念和分类……235
13.2.1 攻击方式的分类原则……236
13.2.2 攻击方式分类方法……236
13.2.3 基于多维属性的攻击分类……238
13.3 信息收集技术……241
13.3.1 初始信息的收集……242
13.3.2 网络地址范围的探查……244
13.3.3 查找活动的机器……245
13.3.4 查找开放端口和入口点……246
13.3.5 操作系统辨识……252
13.3.6 针对特定应用和服务的漏洞扫描……253
13.4 口令攻击……253
13.5 欺骗攻击……254
13.5.1 IP 欺骗……254
13.5.2 邮件欺骗……256
13.5.3 TCP 会话劫持……257
13.6 拒绝服务攻击……257
13.6.1 拒绝服务攻击的类型……257
13.6.2 Ping of Death……258
13.6.3 IP 碎片……258
13.6.4 UDP 洪泛……259
13.6.5 SYN 洪泛……259

13.6.6 Smurf……260
13.6.7 Land……261
13.6.8 分布式拒绝服务攻击……261
13.7 缓冲区溢出攻击……262
思考和练习题……263
实践/实验题……263

第 14 章 计算机病毒及其防治……264
14.1 计算机病毒的定义……264
14.2 计算机病毒的基本特征……264
14.3 计算机病毒的分类……265
14.3.1 按照计算机病毒攻击的操作系统分类……265
14.3.2 按照计算机病毒的链接方式分类……266
14.3.3 按照寄生方式和传染途径分类……266
14.3.4 三类特殊的病毒……267
14.4 计算机病毒的命名……268
14.4.1 常用的命名方法……268
14.4.2 国际上对病毒命名的惯例……268
14.5 计算机病毒的发展历程……269
14.5.1 第一阶段……269
14.5.2 第二阶段……269
14.5.3 第三阶段……269
14.5.4 第四阶段……270
14.6 计算机病毒的基本原理……271
14.6.1 计算机病毒的逻辑结构……271
14.6.2 计算机病毒的工作流程……272
14.6.3 计算机病毒存在的理论基础……273
14.7 特洛伊木马……274
14.7.1 木马的定义……274
14.7.2 木马的特性……274
14.7.3 木马的组成……274
14.7.4 木马的类型……275
14.8 计算机病毒防治对策……275
14.8.1 怎样发现计算机病毒……275
14.8.2 计算机病毒防治技术……276
思考和练习题……278
实践/实验题……278

第 15 章 入侵检测技术……279
15.1 入侵检测概述……279
15.1.1 入侵检测的概念……279

15.1.2 入侵检测的起源和发展……280
15.2 入侵检测系统的功能组成……280
15.2.1 信息收集……280
15.2.2 信息分析……281
15.2.3 结果处理……281
15.3 基于主机及基于网络的入侵检测系统……281
15.3.1 基于主机的入侵检测系统……281
15.3.2 基于网络的入侵检测系统……283
15.4 异常检测和误用检测……285
15.4.1 异常检测……285
15.4.2 误用检测……287
15.5 入侵检测的响应……287
15.5.1 针对入侵者的措施……288
15.5.2 对系统的修正……288
15.5.3 收集攻击者的信息……288
15.6 入侵检测的标准化工作……289
15.6.1 通用入侵检测框架 CIDF……289
15.6.2 入侵检测交换格式……290
15.7 入侵防御系统……290
思考和练习题……290
实践/实验题……290

第 16 章 信息安全评估标准……291
16.1 评估标准的发展历程……291
16.2 TCSEC……292
16.2.1 无保护级……293
16.2.2 自主保护级……293
16.2.3 强制保护级……294
16.2.4 验证保护级……295
16.3 信息技术安全评估通用准则(CC)……295
16.3.1 CC 的范围……295
16.3.2 CC 的组成……295
16.4 GB 17859—1999……296
16.5 GB/T 22239—2008……297
16.5.1 GB/T 22239—2008 简介……297
16.5.2 《基本要求》的框架结构……298
16.5.3 《基本要求》的技术要求……298
16.5.4 《基本要求》的管理要求……300
思考和练习题……301

第 17 章 数据库系统的安全……302
17.1 数据库安全基本条件和安全威胁……302

17.2 数据库安全层次 ……302
17.2.1 应用层 ……302
17.2.2 系统层 ……303
17.2.3 数据层 ……303
17.3 安全数据库技术及进展 ……304
17.4 密码学安全数据库 ……306
17.4.1 数据库加密粒度的选择 ……306
17.4.2 基于数据加密的访问控制 ……307
17.4.3 秘密同态加密算法 ……308
17.4.4 在加密数据上实现查询 ……308
17.4.5 次序保留的加密数据库 ……309
17.5 主要商用安全数据库 ……310
思考和练习题 ……311
实践/实验题 ……311
参考文献 ……312

第1章 绪　　论

1.1 信息和信息安全的概念

1.1.1 信息的定义

信息一词，对我们都不陌生，打开电视、翻开报纸或者连上互联网就会接收到大量信息。人们每天都在接收、使用和交流信息。那么什么是信息呢？一个广义的说法是，信息就是消息。一切存在都有信息。人的五官生来就是为了感受信息的，是信息的接收器，它们所感受到的一切，都是信息。对于大量五官不能直接感受的信息，人类正通过各种手段，发明各种仪器来感知和发现它们。

1928 年，L. V. R. Hartley 在《贝尔系统技术杂志》(BSTJ)上发表了一篇题为“信息传输”的论文，在论文中，他把信息理解为选择通信符号的方式，并用选择的自由度来计量这种信息的大小。但他的定义没有涉及信息的内容、价值和统计性质。

1948 年，信息论的创始人美国数学家 C. E. Shannon 发表了一篇题为“通信的数学理论”的论文，以概率论为基础，给出了信息测度的数学公式，明确地把信息量定义为随机不定性程度的减少。Shannon指出，通信系统所处理的信息在本质上都是随机的，可以用统计的方法进行处理，但这一概念同样没有包含信息的内容和价值。

1948 年，控制论的创始人 N. Wiener 出版了专著《控制论：动物和机器中的通信与控制问题》，从控制论的角度出发，认为信息是在人们适应外部世界，并且这种适应反作用于外部世界的过程中，同外部世界进行互相交换内容的名称。虽然这一定义包含了信息的内容与价值，但没有将信息与物质、能量区别开。

信息的定义有很多种，人们从不同侧面揭示了信息的特征与性质，但同时也存在这样或那样的局限性。

1988 年，我国信息论专家钟义信教授在《信息科学原理》一书中把信息定义为：事物运动的状态和状态变化的方式。信息的这个定义具有最大的普遍性。

1.1.2 信息的属性和价值

由于信息是事物运动的状态和状态变化的方式，而事物运动的状态和状态变化的方式是无限的，因此，信息具有无限性。

信息不是有形的自然实体，自身不能独立存在。但为了传播与被利用，它必须依附于各种载体，如图书、期刊、录音带、录像带、光盘等，同样的信息内容可以不同的载体形态出现，所以信息是无形的。

由于事物本身是不断发展变化的，随着时间和空间的推移，信息也会随之变化。此时此地信息资源价值连城，彼时彼地则可能一文不值。所以信息具有时效性。

信息并不因为分享者的人数多寡而使各自得到的信息量增或减，而是可以存储多次和传输利用的；不同的用户可以在同一时间共享同一内容的信息。

在现今的信息化社会，信息也和能源、材料一样成为一种有价值的资产。信息的价值与其属性相关，信息的真实度越高，就越能减少信息利用者的不确定性，其使用价值就越高。

由于事物皆处于运动变化之中，作为反映事物运动状态和方式的信息也在不断变化，如不能及时地使用最新信息，信息的价值就会随其滞后使用的时差而减值。

信息的价值，也来源于信息可以被交流、存储和使用，如果信息不能被交流和使用，也就失去了存在的价值。

1.1.3 信息安全的含义

安全的本意是采取保护，防止来自攻击者有意或无意的破坏。

信息安全是一个随着历史发展内涵不断丰富的概念，在20世纪60年代至70年代，军事通信提出了通信保密的需求，即必须考虑秘密消息在传送途中被除发信者和收信者以外的第三者(特别是敌方)截获的可能性，使截获者即使截获信息的载体(如文本、无线电波等)也无法得知其中的信息内容。那时，信息安全只具有信息保密的含义。到了20世纪80年代至90年代，信息安全就不仅仅是指机密性，它还包含完整性和可用性，俗称CIA。C代表机密性(Confidentiality)，即保证信息为授权者享用而不泄露给未经授权者。I代表完整性(Integrity)，它包含两方面的含义，一是数据完整性，即数据未被未授权篡改或者损坏，二是系统完整性，即系统未被非授权操纵，按既定的功能运行。A代表可用性(Availability)，即保证信息和信息系统随时为授权者提供服务，而不要出现非授权者滥用却对授权者拒绝服务的情况。除了CIA这三个基本方面，信息安全的其他含义还有不可否认性(Non-repudiation)、鉴别(Authentication)、审计(Accountability)、可靠性(Reliability)等。不可否认性，即要求无论发送方还是接收方都不能抵赖所进行的传输。鉴别就是确认实体是它所声明的，它适用于用户、进程、系统、信息等。审计确保实体的活动可被跟踪，可靠性指的是特定行为和结果的一致性。信息安全需求的多样化，决定了信息安全含义的多样性。

一般认为，安全的信息交换应该满足的5个基本特性是：数据机密性、数据完整性、不可否认性、身份真实性和可用性。

上面，我们从安全的特性或者目标方面对信息安全的内涵进行了阐述和解释。在一些教科书上可能会看到计算机安全包括：实体安全、软件安全、运行安全和数据安全的说法。实际上，一个组织要实现安全的目标，还需要在实体、运行、数据、管理等多个层面实现安全。国家标准GB/T 22239—2008《信息系统安全等级保护基本要求》指出信息系统的安全需要从技术和管理两方面来实现，基本技术要求分为5大类：物理安全、网络安全、主机安全、应用安全和数据安全及备份恢复。

1.2 信息安全的威胁

下面我们来分析一下信息为什么会不安全？信息需要存储、共享、交换、传输和使用。假设信息是从源地址流向目的地址，那么正常的信息流向如图1.1所示。

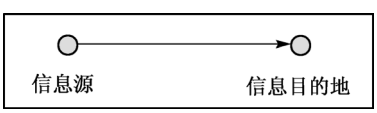


图 1.1 正常的信息流向

信息系统所受到的威胁和攻击多种多样，总体可以分为两类：被动攻击与主动攻击。被动攻击，一般在信息系统的外部运行，对信息网络本身不造成破坏，系统仍可以正常运行，非常难以被检测到，但易于防范。如

窃听或者偷窥、信息内容的泄露(Release of Message Content)、流量分析(Traffic analysis)等。主动攻击，是非法入侵者对数据流的修改，直接进入信息系统内部，往往会影响系统的正常运行，可以被检测到，但难以防范。如伪装(Masquerade)、重放(Replay)、消息篡改(Modification of Message)和拒绝服务(Denial of Service)。

不同的攻击形式威胁到信息安全的不同特性，也可以说，不同的安全威胁导致了相应的信息安全需求。如窃听、业务流分析破坏的是信息的机密性，篡改、重放、旁路、木马等攻击破坏的是信息的完整性，冒充危害的是鉴别特性，抵赖威胁的是不可否认性，拒绝服务、蠕虫病毒、中断等攻击形式对可用性造成了威胁。

中断威胁是使在用信息系统毁坏或者不能使用的攻击，如图1.2所示，破坏了可用性。中断威胁的具体形式有：硬盘等硬件的毁坏，通信线路的切断，文件管理系统的瘫痪等。窃听是一个非授权方介入系统的攻击，非授权方可以是一个人、一个程序或者一台微机，如　　图1.3所示，破坏了机密性。这种攻击包括搭线窃听，文件或者程序的不正当复制。非授权篡改是一个非授权方不仅介入系统而且在系统中“瞎捣乱”的攻击，如图1.4所示，破坏了完整性(Integrity)。这种攻击包括改变数据文件，改变程序使之不能正确执行，修改信件内容等。伪造是一个非授权方将伪造的客体插入系统中，破坏真实性(Authenticity)的攻击，如图1.5 所示。其具体形式有向网络中插入虚假信息，或者在文件中追加记录等。重放是获取有效数据段以重播的方式获取对方信任。在远程登录时，如果一个人的口令不改变，则容易被第三者获取，并用于冒名重放。

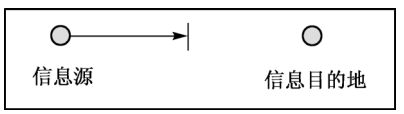


图 1.2　中断威胁

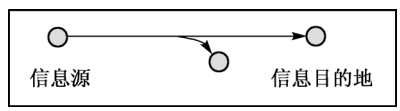


图 1.3　窃听威胁

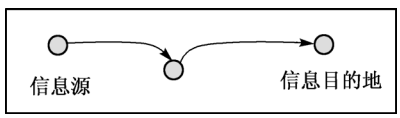


图 1.4　篡改威胁

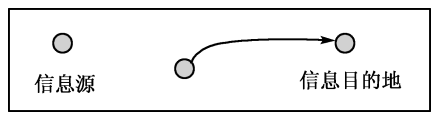


图 1.5　伪造威胁

1.3　安全服务

现实世界中复制品与原件存在不同，对原始文件的修改也总是会留下痕迹，模仿的签名与原始的签名总有差异，可以用铅封来防止文件在传送中被非法阅读或者篡改，用保险柜来防止文件在保管中被盗窃、毁坏、非法阅读或者篡改，采用签名或者图章来表明文件的真实性和有效性，信息安全依赖于物理手段与行政管理。而在数字世界中，复制后的文件跟原始文件没有差别，对原始文件的修改可以不留下痕迹，无法像传统方式一样在文件上直接签名或盖章，不能用传统的铅封来防止文件在传送中被非法阅读或者篡改，难以用传统的保险柜来防止文件在保管中被盗窃、毁坏、非法阅读或者篡改，而信息安全的危害更大，信息的泄密、毁坏所产生的后果更严重，信息安全无法完全依靠物理手段和行政管理。要实现信息安全，只能依赖数学的方法和手段。

1989 年 2 月 15 日国际标准化组织颁布了国际标准 ISO 7498—2，《信息安全体系结构》，确立了基于国际标准开放系统互联(OSI)参考模型七层协议之上的信息安全体系结构。它定

义了五大类安全服务，由参与通信的开放系统的层提供的服务，确保该系统或数据传送具有足够的安全性，分别是数据机密性、数据完整性、抗抵赖、鉴别、访问控制，以及实现这些服务的八类安全机制，分别是加密、数字签名、访问控制、数据完整性、鉴别交换、业务流填充、路由控制、公证，还定义了需要进行的三类安全管理活动。国际电信联盟也于1991年颁布了同样的标准 ITU X.800。

1.3.1 数据机密性

机密性服务是用加密的机制实现的。加密的目的有三种：密级文件经过加密可以公开存放和发送；实现多级安全控制的需要；构建加密通道的需要，防止搭线窃听和冒名入侵。ISO 7498—2 把保密性分为以下四类：

- **连接保密**：即对某个连接上的所有用户数据提供保密。
- **无连接保密**：即对一个无连接的数据报的所有用户数据提供保密。
- **选择字段保密**：即对一个协议数据单元中的用户数据的一些经选择的字段提供保密。
- **信息流机密性**：这种服务提供的保护, 使得通过观察通信业务流而不可能推断出其中的机密信息。

1.3.2 数据完整性

数据完整性是数据本身真实性的证明。数据完整性有两个方面：单个数据单元或字段的完整性、数据单元流或字段流的完整性。ISO 7498—2 把完整性分为以下几类：

- **带恢复的连接完整性**：这种服务为连接上的所有用户数据保证其完整性, 并检测整个服务数据单元序列中的数据遭到的任何篡改、插入、删除或重演，同时试图补救恢复。
- **不带恢复的连接完整性**：与上相同，只是不做恢复。
- **选择字段连接完整性**：这种服务为在一次连接上传送的服务数据单元的用户数据中的选择字段保证其完整性，所取形式是确定这些被选字段是否遭到了篡改、插入、删除或重演。
- **无连接完整性**：这种服务当由某层提供时，对发出请求的上层实体提供完整性保证。这种服务为单个的无连接服务数据单元保证其完整性，所取形式可以是确定一个接收到的服务数据单元是否遭受了篡改。另外，在一定程度上也能提供对重演的检测。
- **选择字段无连接完整性**：这种服务为单位无连接的服务数据单元中的被选字段保证其完整性，所取形式为确定被选字段是否遭受了篡改。

1.3.3 抗抵赖

抗抵赖或者说不可否认性是一种防止源点或终点抵赖的鉴别技术。这种服务可取如下两种形式，或两者之一。一种是有数据原发证明的抗抵赖：为数据的接收者提供数据来源的证据，这将使发送者谎称未发送过这些数据或否认它的内容的企图不能得逞。另一种是有交付证明的抗抵赖：为数据的发送者提供数据交付证据，这将使得接收者事后谎称未收到过这些数据或否认它的内容的企图不能得逞。数字签名是实现抗抵赖服务的机制。

1.3.4　鉴别

鉴别(Authentication)就是确认实体是它所声明的。ISO 7498—2 把鉴别分为下面两种情况：实体鉴别(Entity Authentication)和数据原发鉴别。实体鉴别也称为身份鉴别：某一实体确信与之打交道的实体正是所需要的实体。只是简单地鉴别实体本身的身份，不会和实体想要进行何种活动相联系。数据原发鉴别鉴定某个指定的数据是否来源于某个特定的实体。不是孤立地鉴别一个实体，也不是为了允许实体执行下一步的操作而鉴别它的身份，而是为了确定被鉴别的实体与一些特定数据项有着静态不可分割的联系。这种服务对数据单元的重复或篡改不提供保护。后面我们还会提到一个术语，消息鉴别(Message Authentication)，它指的是一个证实收到的消息来自可信的源点且未被篡改的过程。

1.3.5　访问控制

访问控制(Access Control)是针对越权使用资源的防御措施。它的基本目标是防止对任何资源(如计算资源、通信资源或信息资源)进行未授权的访问。从而使计算机系统在合法范围内使用；决定用户能做什么，也决定代表一定用户利益的程序能做什么。未授权的访问包括：未经授权的使用、泄露、修改、销毁信息以及颁发指令等。它包含两种形式：一是非法用户进入系统，二是合法用户对系统资源的非法使用。这种保护服务可应用于对资源的各种不同类型的访问，例如：使用通信资源；读、写或删除信息资源；处理资源的执行，或应用于对一种资源的所有访问。几种典型的访问控制策略为：自主访问控制(Discretionary Access Control，DAC)、强制访问控制(Mandatory Access Control，MAC)和基于角色的访问控制(Role-Based Access Control，RBAC)。

1.3.6　OSI 安全服务的分层配置

安全服务可以通过不同层的安全机制来实现，ISO 7498—2 的另一个贡献是把这几种服务映射到 OSI 的七层模型当中。根据 OSI 七层模型与 TCP/IP 参考模型的对应关系，我们可以给出表 1.1 的映射关系。

表 1.1　安全服务与 TCP/IP 协议层的关系

安全服务	TCP/IP 协议层			
	网络接口	互联网层	传输层	应用层
对等实体鉴别	—	—	Y	Y
数据源鉴别	—	Y	Y	Y
访问控制服务	—	Y	Y	Y
连接保密性	Y	Y	Y	Y
无连接保密性	Y	Y	Y	Y
选择域保密性	—	—	—	Y
流量保密性	Y	Y	—	Y
有恢复功能的连接完整性	—	—	Y	Y
无恢复功能的连接完整性	—	Y	Y	Y
选择域连接完整性	—	—	—	Y
无连接完整性	—	Y	Y	Y

(续表)

安全服务	TCP/IP 协议层			
	网络接口	互联网层	传输层	应用层
选择域非连接完整性	—	—	—	Y
源发方不可否认	—	—	—	Y
接收方不可否认	—	—	—	Y

1.4 信息安全模型

1.4.1 通信安全模型

经典的通信安全传输模型如图1.6所示，通信一方通过公开信道将消息传送给另一方，要保护信息传输的机密性、真实性等特性的时候，就涉及通信安全。通信的发送方要对信息进行相关的安全变换，可以是加密、签名，接收方再进行相关的逆变换，比如解密、验证签名。双方进行的安全变换通常需要使用一些秘密信息，比如加密密钥、解密密钥。根据上述安全模型，设计安全服务需要完成的四个基本任务是：

(1) 设计一个算法，执行安全相关的转换，算法应具有足够的安全强度。

(2) 生成该算法所使用的秘密信息，也就是密钥。

(3) 设计秘密信息的分布与共享的方法，也就是密钥的分配方案。

(4) 设定通信双方使用的安全协议，该协议利用密码算法和密钥实现安全服务。

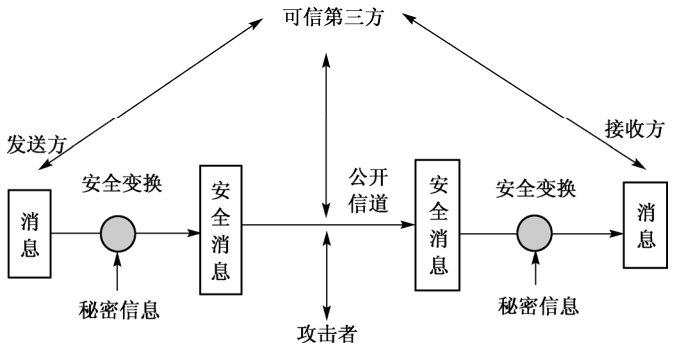


图 1.6 经典的通信安全传输模型

1.4.2 信息访问安全模型

还有一些与安全相关的情形不完全适用于以上模型，William Stallings 给出了如图1.7所示的信息访问安全模型，该模型希望保护信息系统不受到有害的访问。一种有害的访问由黑客引起，他们也可能没有恶意，只是满足于闯入计算机系统，也可能想进行破坏或利用计算机获利。另一种有害的访问，来自恶意软件，比如病毒、蠕虫、木马。对付有害攻击所需要的安全服务则包含鉴别和访问控制两类。

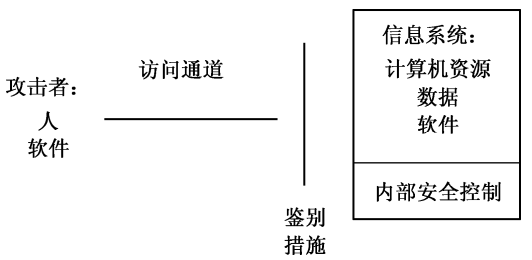


图 1.7 信息访问安全模型

1.4.3　动态安全模型

基于上述模型的安全措施，都属于静态的预防和防护措施，它通过采用严格的访问控制和数据加密策略来提供防护，但在复杂系统中，这些策略是不充分的。这些措施都是以减慢交易为代价的。而且，也不能保证万无一失。由于系统、组织和技术都是发展变化的，攻击也是动态的，动态的安全模型更切合实际需求。在这种形势下，著名的计算机安全公司 Internet Security Systems Inc.提出了 P^2DR（Policy Protection Detection Response）模型，如图 1.8 所示。

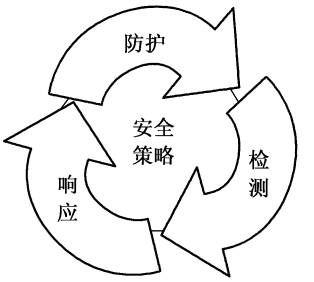


图 1.8　P^2DR 安全模型

在这个模型中，安全策略是模型的核心，具体的实施过程中，策略意味着网络安全要达到的目标。防护包括安全规章、安全配置和安全措施，检测的两种方法有异常检测和误用检测，响应包括报告、记录、反应和恢复等措施。如果把攻击者通过保护措施所需要的时间记为 P_t，检测系统发现攻击和响应的时间分别记为 D_t 和 R_t，就可以给出一个可以量化、可以计算的基于时间的动态模型，如果 $P_t>D_t+R_t$，就可以认为系统是安全的。

通用安全评价准则（Common Criteria for IT security Evaluation, CC）使用威胁、漏洞和风险等词汇定义了一个动态的安全概念和关系模型，如图1.9所示。这个模型反映了所有者和攻击者之间的动态对抗关系，它也是一个动态的风险模型和效益模型。所有者要采取措施，减少漏洞对资产带来的风险，攻击者要利用漏洞，从而增加对资产的风险，所有者采取什么样的保护措施，是和资产的价值有关的，他不可能付出超过资产价值的代价去保护资产，同样，攻击者也不会以超过资产价值的攻击代价进行攻击。

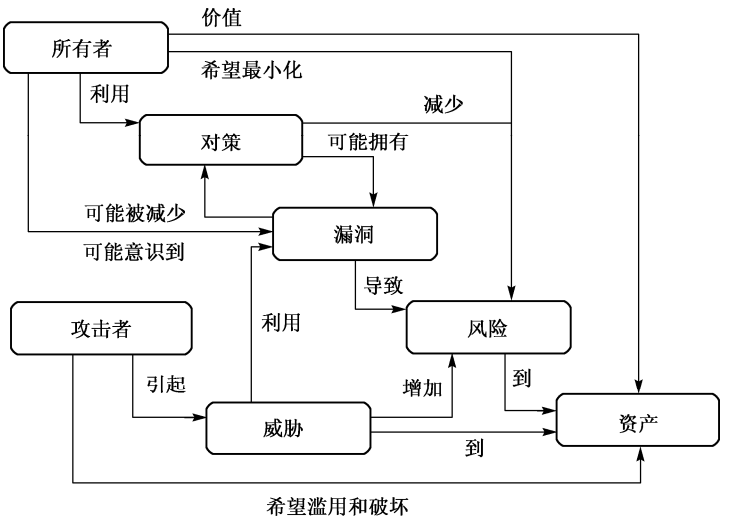


图 1.9　CC 定义的安全概念和关系模型

1.5　信息安全的技术体系

虽然密码算法是信息安全技术的基础，但仅有密码算法是不够的，我国信息安全专家赵战生教授，在《中国信息安全体系机构基本框架与构想》一文中指出，信息安全技术也不是

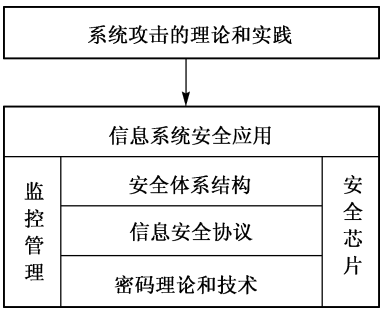


图 1.10 信息安全技术体系

单一的技术，要支持信息系统的安全应用，密码理论和技术是一个核心，安全协议是一个桥梁，安全体系结构是一个基础，安全的芯片是关键，监控管理是保证，攻击和评测的理论和实践是考验，他给出的信息安全技术体系如图1.10所示。

1.6 信息安全的政策法规

人们在不断探索和发展各种技术来满足信息安全需求的同时，逐渐认识到信息安全是一个综合的多层面问题。信息安全保障不仅仅是技术问题，是人、政策和技术三大要素的结合。赵战生教授指出，三个因素：人、技术和政策，它们是有层次关系的，人是打底座的，是根本的；技术是顶端的东西，但是技术是要通过人，通过相应的政策和策略去操作这个技术。一个完整的国家信息安全保障体系应包括信息安全法制体系、组织管理体系、基础设施、技术保障体系、经费保障体系和安全意识教育人才培养体系。一个简单的说法是，要保障信息安全，“三分靠技术、七分靠管理”。足见管理在信息安全中的地位和作用。信息安全管理的原则体现在政府制定的政策法规和机构部门制定的规范制度上。同时，信息安全技术蓬勃发展，形成了一个新的产业，规模化的信息安全产业发展需要技术标准来规范信息系统的建设和使用，生产出满足社会广泛需求的安全产品。在前面对信息安全技术了解的基础上，本节将从政策法规和标准的角度进行介绍，使读者获得关于信息安全的更完整的认识。

1.6.1 国际信息安全政策法规

计算机安全和密码使用是信息安全的两个重要方面，有关的政策法规也因此分为这两个方面，在信息安全的早期阶段，立法和管理的重点集中在计算机犯罪方面，各国陆续围绕着计算机犯罪等问题确立了一些安全法规，之后，立法的热点转移到密码的使用管理方面。

美国的信息技术具有国际领先水平，其安全法规政策也最为完善。早在1987年，美国就再次修订了计算机犯罪法，这部法律在 20 世纪 80 年代末至 90 年代初一直被作为美国各州制定其地方法规的依据。美国现已确立的有关信息安全的法规有：信息自由法、个人隐私法、反腐败行径法、伪造访问设备和计算机欺骗滥用法、电子通信隐私法、计算机欺骗滥用法、计算机安全法和电信法。1998 年 5 月，美国颁发第 63 号总统令，要求行政部门评估国家关键基础设施的计算机脆弱性，并要求联邦政府制定保卫国家免受计算机破坏的详细计划。紧接着于2000年 1 月颁布了《保卫美国计算机空间——信息系统保护国家计划 1.0》，这是一个规划美国计算机安全持续发展和更新的综合方案。可以看出，美国把信息系统保护提高到国家战略计划的高度坚决实施。

俄罗斯于1995年颁布了《联邦信息、信息化和信息保护法》，法规明确界定了信息资源开放和保密的范畴，提出了保护信息的法律责任。2000 年，普京总统批准了《国家信息安全学说》，它是一部纲领性、指导性、顶层性、战略性文件，并不是学说。它明确了俄罗斯联邦信息安全建设的目的、任务、原则和主要内容，第一次明确指出了俄罗斯在信息安全领域的利益、受到的威胁，以及为保障信息安全应采取的首要措施。俄罗斯对信息安全的重视由此可见一斑。

欧洲经济共同体(欧盟的前身)是一个在欧洲范围内具有较强影响力的政府间组织。其成员国从20世纪70年代末到80年代初，先后制定并颁布了各自有关数据安全的法律。

德国政府于1996年夏出台了《信息和通信服务规范法》(即多媒体法)，为电子信息和通信服务的各种利用可能性规定了统一的基本法律框架。该国政府还通过了电信服务数据保护法，并根据需要对刑法法典、治安法、传播危害青少年文字法、著作权法和报价法做了必要的修改和补充。

新加坡在1996年宣布对互联网络实行管制，宣布实施分类许可证制度。它是一种自动取得许可证的制度，目的是鼓励正当使用互联网络，促进其健康发展。

其他国家，如英国、法国、日本等也制定了相应的计算机安全政策法规。

关于密码使用的政策涉及使用密码进行加密和进行数字签名实施证书授权管理两个方面。

美国是最早允许在国内社会使用密码的国家，美国国内，政府、军界、企业和个人为了各自的利益，围绕信息加密政策的争论繁多，主要是密码的使用范围和允许出口的长度。此外，多国出口控制协调委员会(COCOM)、欧盟和国际商务委员会等组织以及英国、法国、德国、意大利、俄罗斯、波兰、澳大利亚、中国香港地区等许多国家和地区也分别制定了自己的信息加密政策。

对于数字签名技术，有关国际组织、各国政府和企业为了各自的利益，很难达成一致观点。1995 年，美国犹他州通过了美国历史上(也是世界历史上)第一部数字签名法。在犹他州的带动下，美国的其他一些州也确立了自己的数字签名法，但是美国联邦政府迟迟没有立法，德国有幸成为第一个以国家名义制定数字签名法的国家。

1.6.2　国内信息安全政策法规

我国建立了如下国家信息安全组织管理体系：

国务院信息化领导小组对Internet安全中的重大问题进行管理协调，国务院信息化领导小组办公室作为 Internet 安全工作的办事机构，负责组织、协调和制定有关 Internet 安全的政策、法规和标准，并检查监督其执行情况。

政府有关信息安全的其他管理和执法部门(如工业和信息化部、国家安全部、公安部、国家保密局、国家密码管理局和国务院新闻办公室等)分别依据其职能和权限进行信息安全的管理和执法活动。

工业和信息化部协调有关部委关于信息安全的工作；公安部主管公共网络安全，即全国计算机系统安全保护工作；国家安全部主管计算机信息网络国际联网的国家安全保护管理工作；国家保密局主管全国计算机信息系统的保密工作；国家密码管理委员会主管密码算法与设备的审批和使用工作；国务院新闻办公室负责信息内容的监察。

我国信息安全管理的基本方针是“兴利除弊，集中监控，分级管理，保障国家安全”。对于密码的管理政策实行“统一领导、集中管理、定点研制、专控经营、满足使用”的发展和管理方针。

相对国外网络立法已成普及之势的情况，我国目前的信息化立法，尤其是信息安全立法，尚处于起步阶段，我国政府和法律界都清醒地认识到这一问题的重要性，正在积极推进这一方面的工作。

我国政府现有的信息安全法规政策可以分为两个层次。一是法律层次，从国家宪法和其

他部门法的高度对个人、法人和其他组织涉及国家安全的信息活动的权利和义务进行规范，例如1997年新《刑法》首次界定了计算机犯罪。二是行政法规和规章层次，直接约束计算机安全和Internet安全，对信息内容、信息安全技术和信息安全产品的授权审批进行规定。其中，第一个层次上的法律主要有宪法、刑法、国家安全法和国家保密法。第二个层次主要包括《中华人民共和国计算机信息系统安全保护条例》(简称《安保条例》)、《中华人民共和国计算机信息网络国际互联网管理暂行规定》(简称《联网规定》)、《中华人民共和国计算机信息网络国际互联网安全保护管理办法》、《电子出版物管理暂行规定》、《中国互联网络域名注册暂行管理办法》和《计算机信息系统安全专用产品检测和销售许可证管理办法》等条例和法规。

1.7 信息安全的相关机构和相关标准

1.7.1 国际标准化机构

1.7.1.1 国际标准化组织和国际电工委员会

国际标准化组织(ISO)和国际电工委员会(IEC)是世界性的标准化专门机构。国家成员体(都是 ISO 或 IEC 的成员国)通过国际组织建立的各个技术委员会参与制定特定技术范围的国际标准。ISO 和 IEC 的各技术委员会在共同感兴趣的领域内进行合作。与 ISO 和 IEC 有联系的其他官方和非官方国际组织也可以参与国际标准的制定工作。

对于信息技术，ISO 和 IEC 建立了一个联合技术委员会，即 ISO/IEC JTC1。由联合技术委员会提出的国际标准草案分发给国家成员体进行表决。发布一项国际标准，至少需要 75%的参与表决的国家成员体投票赞成。

开放系统互连(OSI)基本参考模型(ISO/IEC 7498)是由ISO/IEC JTC1“信息技术”联合技术委员会与 ITU-T 共同制定的，等同文本为 ITU-T 建议 X .200。

1.7.1.2 国际电报和电话咨询委员会

国际电报和电话咨询委员会(CCITT)是一个联合国条约组织，属于国际电信联盟，由主要成员国的邮政、电报和电话部门组成，主要从事涉及通信领域的接口和通信协议的制定，与ISO 密切合作进行国际通信的标准化工作，在数据通信范围内的工作体现于 V 系列和 X 系列建议书。

1.7.1.3 国际信息处理联合会第十一技术委员会

国际信息处理联合会第十一技术委员会(IFIP TC11)是国际上有重要影响的有关信息系统安全的国际组织，每年举行一次计算机安全的国际研讨会，公安部代表我国参加该组织的活动。该组织机构包括安全管理工作组、办公自动化安全工作组、数据库安全工作组、密码工作组、系统完整性与控制工作组、计算机事务处理工作组、计算机安全法律工作组和计算机安全教育工作组。

1.7.1.4 电气和电子工程师学会

电气和电子工程师学会(IEEE)是一个由电气和电子工程师组成的世界上最大的专业性学会，划分成许多部门。1980 年2 月，IEEE 计算机学会建立了一个委员会负责制定有关网络的协议标准(802.1～9)，包括高层接口、逻辑链路控制、CSMA/CD 网、令牌总线网、令牌环网、城域网、宽带技术咨询组、光纤技术咨询组、数据和语音综合网络等标准。

1.7.1.5　欧洲计算机制造商协会

欧洲计算机制造商协会(ECMA)由一些欧洲最大的计算机和技术公司成立，是包括美国在欧洲供应计算机的厂商在内的组织，致力于适用于计算机技术的各种标准的制定和颁布，在 ISO 和 CCITT 中是一个没有表决权的成员。

1.7.1.6　Internet 体系结构委员会(IAB)

Internet 体系结构委员下设两个重要部门：Internet工程特别工作组(IETF)和Internet研究特别工作组(IRTF)。发展到今天，IAB 公布的协议参考草案(Request For Comments, RFC)已经积累到 3000 多个。一个 RFC 文件在成为官方标准前一般至少要经历三个阶段：建议(proposed)标准、草案(draft)标准、因特网标准(Internet Standard)。一份文档必须以建议标准保留 6 个月，而草案标准要保持 4 个月，以提供足够的时间进行修改和评论。

1.7.2　美国的标准化机构

1.7.2.1　美国商务部国家标准技术研究所

美国国家标准和技术研究所(NIST)属于美国商务部的一个机构，现在的工作由 NIST 进行，发布销售给美国联邦政府的设备的信息处理标准。NIST 与 NSA 紧密合作，在 NSA 的指导监督下，制定计算机信息系统的技术安全标准。其工作一般以 NIST 出版物(FIPS PUB)和 NIST 特别出版物(SPEC PUB)等形式发布。所制定的信息安全规范和标准很多，主要涉及访问控制和认证技术、评价和保障、密码、电子商务、一般计算机安全、网络安全、风险管理、电信和联邦信息处理标准等。

该机构比较有影响的工作是制定公布了美国国家数据加密标准 DES，参考了美国、加拿大、英国、法国、德国、荷兰等国制定的信息安全的通用评价准则(CC)，在1993年制定了密钥托管加密标准 EES。

1.7.2.2　美国国家标准学会

美国国家标准学会(ANSI)是由制定标准和使用标准的组织联合组成的非营利的非政府的民办机构，由全美1000多家制造商、专业性协会、贸易协会、政府和管理团体、公司和用户协会组成，是美国自发的制定与计算机工业有关的各种标准的统筹交流组织。

1.7.2.3　美国电子工业协会

美国电子工业协会(EIA)是美国电子公司贸易协会，属于ANSI的成员。它制定了涉及电气和电子领域的400多个标准，该机构比较有影响的工作是建立了数据终端设备和数据通信设备间的接口标准(如 RS232C 等)。

1.7.2.4　美国国防部及国家计算机安全中心

美国国防部(DoD)早在 20 世纪 80 年代就针对计算机安全保密开展了一系列有影响的工作，后来成立的美国国家计算机安全中心(NCSC)接续进行有关的工作。1983年公布了《可信计算机信息系统评价准则》TCSEC，以后 NCSC 又出版了一系列有关可信计算机数据库、可信计算机网络的指南。

1.7.2.5　其他标准

除了上述标准化组织，美国的一些公司也纷纷研究和提出有关规范建议，并根据建议发展产品，试图将建议变为实际的工业标准，其中一些建议或标准归纳如表1.2 所示。

表 1.2 有关公司制定的标准

协 议 名	协议开发者	协 议 内 容
PKCS	RSA 数据安全公司 RSA 实验室在 Apple，Microsoft, DEC, Lotus, Sun 和 MIT 等机构非正式的咨询合作下开发	公开密钥密码标准与 ITU-X.509 标准兼容
SSL	Netsacpe	用于 WWW 上的会话层安全协议
S-HTTP	Enterprise Integration Technologies	基于 WWW，提供保密、认证、完整性和不可否认服务
PTC	Microsoft 和 Visa	保密通信协议，与SSL类似，不同的是，它在客户和服务器之间包含了几个短的报文数据，认证和加密使用不同的密钥，提供了某种防火墙功能
SET	Visa 和 Mastercard	开放网络电子支付协议

1.7.3 信息安全组织机构

目前，国际上信息安全方面的协调机构主要有计算机应急响应小组(CERT/CC)、信息安全问题小组论坛(FIRST)。

计算机应急响应小组是一个信息安全专家技术中心，是设在Carnegie Mellon大学软件工程研究所的联邦资助的研究开发中心，成立于1988年。该组织研究 Internet 的脆弱性、处理计算机安全事件、发布安全警告、研究网络系统的长期变化以及提供安全培训帮助用户提高站点的安全性。CERT/CC 成立后，很多政府、商业和学术机构都组建了信息安全问题小组，但 CERT/CC 始终是这一方面规模最大、最著名和最权威的组织。

信息安全问题小组论坛(FIRST)成立于 1990 年，当时只有 11 个成员，截止到 2001 年，它的成员已超过 90 个，截止到 2003 年 8 月，FIRST 的正式成员达到 151 个，目前成员已超过 200 个。FIRST 的目标是为有效解决安全事件加强各小组间的合作，作为小组之间的信息中介，促进安全技术的共享和研究活动的开展。

美国国内与信息安全事物有关的管理机构主要有国家安全局(NSA)、国家标准技术研究所(NIST)、联邦调查局(FBI)、高级研究计划署(ARPA)和国防部信息局(DISA)。他们有各自授权管理的领域和业务，同时，这些机构通过信息安全管理职责上的理解备忘录和协议备忘录进行合作。此外，美国国会在1987年的计算机安全法案中宣布成立了计算机系统安全及隐私协会——CSSPAB，负责识别与计算机系统安全和隐私相关的管理、技术、政策和物理监护方面的问题。为满足信息技术生产者和使用者进行产品安全测试的需要，美国政府还成立了 NIAP。

1.7.4 国内标准制定情况

我国是国际标准化组织的成员国，我国的信息安全标准化工作在各方面的努力下，正在积极开展之中。国务院授权履行行政管理职能和统一管理全国标准化工作的主管机构是中国国家标准化管理委员会。国家标准化管理委员会下设有 255 个专业技术委员会。1984 年成立了全国信息技术安全标准化技术委员会(CITS)，在国家标准化管理委员会与工业和信息化部的共同领导下负责全国信息技术领域以及与 ISO/IEC JTC1 相对应的标准化工作，下设 24 个分技术委员会和特别工作组。从20世纪80年代中期开始，我国自主制定和视同采用了一批相应的信息安全标准。已颁布的信息技术安全标准涉及信息技术设备的安全、信息处理系统开放系统互联安全体系结构、数据加密、数字签名、实体鉴别、抗抵赖和防火墙安全技术等。

2001 年，我国颁布的国家标准 GB/T 18336 等同采用国际标准 ISO/IEC 15408，即 CC。这些标准的颁布将积极推动我国的信息化建设与发展。

此外，在一些对信息安全要求高的行业和对信息安全管理负有责任的部门，也制定一些信息安全的行业标准和部门标准，如金融、公安等行业和部门。

思考和练习题

(1) 信息安全的 CIA 指的是什么？
(2) 传统世界中的安全与数字世界中的安全有什么差别？
(3) 通信系统的典型攻击形式有哪些？
(4) ISO 7498—2 定义的五大类安全服务是什么？
(5) 简述现有的安全模型有哪些？
(6) ISO 7498—2, TCSEC, DES 分别是由哪个机构制定的？

第 2 章　密码学基础

2.1　密码学的基本概念和术语

自从人类文化诞生以来，就产生了保护敏感信息的愿望。而信息的价值来源于信息可以被交流和使用。我们需要在公开的地方存储信息，使用非隐秘介质交换信息，通过不安全的信道传输信息，就需要某种手段保护信息在存储、交换和传输中的安全，而密码技术正是基于保护敏感信息的需要而产生和发展起来的。密码在今天与我们的生活息息相关，每个人的脑子里大概都记着一堆密码，无论是在ATM机上取钱、拨号上网、登录电子邮箱都要输入密码。密码真的能保证我们的钱财与隐私的安全吗？它又是怎样使我们的信息得到保密和安全的？本章将初步回答这些问题，引导读者步入信息安全的大门。

首先来了解一些有关密码学的基础知识和概念。

2.1.1　消息和加密

消息被称为明文。用某种方法伪装消息以隐藏它的内容的过程称为加密，被加密的消息称为密文，而把密文转变为明文的过程称为解密。图2.1表示了这个过程。密码算法是用于加密和解密的数学函数。

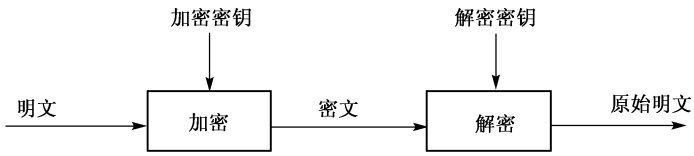


图 2.1　加密和解密

密码学(Cryptology)是研究信息系统安全保密的科学。传统密码学包括密码编码和密码分析两方面的内容。密码编码学(Cryptography)主要研究对信息进行编码，实现对信息的隐蔽。密码分析学(Cryptanalytic)主要研究加密消息的破译或消息的伪造。密码编码技术和密码分析技术是相互依存、相互促进、密不可分的两个方面。随着密码学的发展，这一学科包含了更广泛的内涵，除了编码与破译，而且还包括密码协议、散列函数、安全管理等内容。密码学的进一步发展，又涌现了大量的新概念和新技术，如零知识证明、盲签名、量子密码、混沌密码等。

- 对明文进行加密操作的人员称为加密员或密码员(Cryptographer)。
- 密码算法(Cryptography algorithm)是用于加密和解密的数学函数。
- 密码员对明文进行加密操作时所采用的一组规则称为加密算法(Encryption algorithm)。
- 所传送消息的预定对象称为接收者(Receiver)。
- 接收者对密文解密所采用的一组规则称为解密算法(Decryption algorithm)。
- 加密和解密的操作通常都是在一组密钥的控制下进行的，分别称为加密密钥(Encryption key)和解密密钥(Decryption key)。

下面通过一个例子来解释以上概念。

2.1.2　恺撒密表

公元前 54 年，古罗马杰出的军事家、政治家和作家，共和国末期的独裁者恺撒(Caesar)使用一个简单的代替密码来保护军队和政府的通信，在密码学上称为“恺撒密表”。古罗马文字就是我们英语中所熟知的26个拉丁字母，“恺撒密表”把明文中每一个字母用它在字母表中位置后面的第三个字母代替，如表 2.1 所示。

表 2.1　恺撒密表

明文字母	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
密文字母	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

例如，有一个拉丁文句子：

omnia gallia est divisa in partes tres

(高卢全境分为三个部分)

用恺撒密表加密后就成为密文 RPQLD JDOOLD HVW GLYLVD LQ SDUWHVWUHV。如果不知道其中奥秘，简直不知所云。收信者收到这段密文后要进行解密，解密也用的是恺撒密表，就是把恺撒密表第二行中的每一个字母用它头顶上第一行的相应字母代替。虽然加解密都在用恺撒密表，但加解密时所用的变换是两个互逆的变换。如果用数字 0, 1, 2, …, 25 分别表示字母 a, b, c, …, z，加密相当于模 26 加 3，解密相当于模 26 减 3。除了这种简单代替外，恺撒还把明文的拉丁字母代替为对应的希腊字母。

从以上例子看到，密码的变换规则显然是至关重要的，一旦变换规则被敌方掌握，则无秘密可言，因此变换规则必须严格保密。要提高密码的安全性，不能让敌方轻易破译，就要把变换规则设计得尽量复杂，但变换规则复杂到一定程度就变得难以记忆，需要用文字记录下来，而一有文字记录，其安全性就大打折扣。首先，加密双方需要有变换规则的复本，而复本越多，安全性就越差，文字记录与被保管者的分离，也增加了其被窃取的可能性。如果在把变换规则设计得尽可能复杂的同时，设计出一个或一组“关键词”，根据这个(组)关键词可以推导出变换规则，这个关键词就称为密钥。恺撒密表的密钥就是 3，加密规则是“后移 3”，而解密规则是“前移 3”。

2.1.3　密码体制

图 2.2 给出了保密通信的模型，密码学的目的就是 A 和 B 两个人在不安全的信道上进行通信，而攻击者(破译者)O 不能理解他们通信的内容。

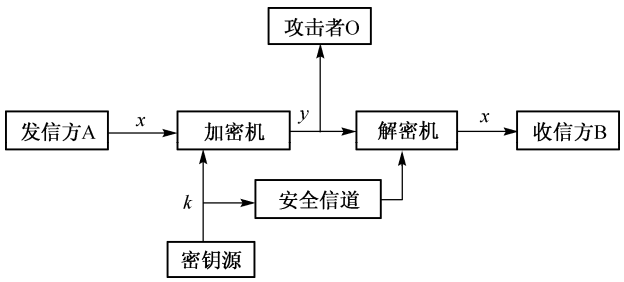


图 2.2　保密通信的模型

通常一个完整密码体制要包括如下 5 个要素 M、C、K、E、D：

- M 是可能明文的有限集，称为明文空间。
- C 是可能密文的有限集，称为密文空间。
- K 是一切可能密钥构成的有限集，称为密钥空间。
- 对于密钥空间的任一密钥 $k \in K$，有一个加密算法 $e_k \in E$ 和相应的解密算法 $d_k \in D$，使得 $e_k : M \to C$ 和 $d_k : C \to M$ 分别为加密解密函数，满足 $d_k(e_k(x)) = x$，这里 $x \in M$。

一个密码体制要是实际可用的，必须满足如下特性：

(1) 每一个加密函数 e_k 和每一个解密函数 d_k 都能有效地计算。

(2) 破译者取得密文后，将不能在有效的时间内破解出密钥 k 或明文 x。

(3) 一个密码系统是安全的必要条件是穷举密钥搜索将是不可行的，即密钥空间非常大。

密码学中的术语“系统或体制”(System)、“方案”(Scheme)和“算法”(Algorithm)本质上是一回事情，本书按作者的习惯交替使用这些术语。

2.1.4 密码算法的分类

密码的发展经历了古典密码、对称密钥密码、公开密钥密码的发展阶段。古典密码是基于字符替换的密码，现在已很少使用了，但是它代表了密码的起源。现在仍在使用的则是对位进行变换的密码算法。密码算法可以按照不同的标准分类如下。

(1) 按照保密的内容分为，受限制的(Restricted)算法和基于密钥(Key-based)的算法。

受限制的算法指的是算法的保密性基于保持变换规则的秘密。虽然这种密码也有密钥，但算法的安全基于对整个变换规则的保密。古典密码就是这类密码。1883 年 Kerchoff 第一次明确提出了密码编码的原则：加密算法应建立在变换规则的公开不影响明文和密钥的安全的基础上。这一原则已得到普遍承认，成为判定密码强度的衡量标准，实际上也成为古典密码和现代密码的分界线。基于密钥的算法指的就是算法的保密性基于对密钥的保密的密码算法。现代密码都属于这类密码。为什么基于密钥的算法比受限制的算法更安全和实用，这里可以给出如下几个理由：

(a) **攻击者总会设法找到算法**：在密码学的历史上，攻击者总是能在没有任何帮助的情况下推导出你的算法。在战争年代，他们可以偷盗密码机，或是通过敲诈、勒索等手段使一些人泄密，甚至可以在没有密码机的情况下，确定它是如何工作的。在现代，如果密码系统是用软件实现的，密码分析者可以使用汇编语言调试器跟踪目标代码得到它，如果密码系统是基于硬件的，工程师就可以打开它，了解它的内部结构。有时，使一个算法保密的时间足够长是可能的，但最终攻击者仍可以找到它。

(b) **更换密钥比更换密码算法更容易**：如果一个算法的安全性基于算法的保密，算法泄露就意味着不得不使用新算法，而设计一个新算法的代价远比更换密钥要大得多。

(c) **公开的算法更安全**：当算法公开后，全世界的密码分析者和计算机工程师就有了检验它的弱点的机会，如果算法是保密的，就没有足够多的分析者来发现它可能存在的弱点，这不意味着它没有弱点，而是仅仅意味着你不知道算法的弱点。

(d) **商业应用的需要**：密码算法公开才能进行标准化，得到更广泛的应用，从而带来经济效益。如果希望算法保密，就只能让有限的信任的人才可以使用它，结果就是无钱可赚。

(2) 基于密钥的算法，按照密钥的特点分为对称密码算法和非对称密码算法。

对称密码算法(Symmetric cipher)的加密密钥和解密密钥相同，或实质上等同，即从一个易于推出另一个，对称密码算法又称为秘密密钥算法或单密钥算法。

非对称密钥密码算法(Asymmetric cipher)的加密密钥和解密密钥不相同，从一个很难推出另一个，非对称密钥算法又称为公钥密码算法(Public key cipher) 、双钥密码算法。

公钥密码算法用一个密钥进行加密，而用另一个进行解密。其中的加密密钥可以公开，又称为公开密钥(Public key)，简称公钥。解密密钥必须保密，又称为私人密钥(Private key)或私有密钥，简称私钥。

(3) 按照明文的处理方法，密码算法又可分为序列密码和分组密码两大类。

序列密码每次加密一位或一字节的明文，也可以称为流密码。序列密码是手工和机械密码时代的主流。它的主要原理是：通过有限状态机产生性能优良的伪随机序列，使用该序列逐比特加密信息流，得到密文序列，算法的安全强度完全决定于它所产生的伪随机序列的好坏。产生好的序列密码的主要途径之一是利用移位寄存器产生伪随机序列。目前要求寄存器的阶数大于 100 阶，才能保证必要的安全。由于序列密码具有实现简单、速度快、错误扩展小等优势，在专用和机密机构中仍保持着优势。分组密码将明文分成固定长度的组，用同一密钥和算法对每一分组加密，输出也是固定长度的密文。

对称密钥密码又可分为分组密码和流密码，多数网络加密应用的就是对称分组密码，如 DES，IDEA，RC6，Rijndael 等。对称流密码一次对一位或一字节加密，一次一密(One-time pad)以及古典密码中的 Vigenére，Vernam 密码可以看做流密码。

公开密钥密码大部分是分组密码，只有概率密码体制属于流密码。公开密钥密码可用于数字签名和身份鉴别，如RSA、ECC和ElGamal算法，每次对一块数据加密，加密解密速度比对称密码慢。

2.1.5　密码分析

假设破译者 O 是在已知密码体制的前提下来破译正在使用的密钥。这个假设称为 Kerckhoff 原则。密码分析必须假设在破译时已经具备了一些已知条件，根据攻击者掌握的信息的多少，最常见的破解类型如下：

唯密文攻击：破译者 O 仅掌握了一些密文，具有密文串 y。

已知明文攻击：破译者 O 具有明文串 x 和相应的密文 y。

选择明文攻击：破译者 O 可获得对加密机的暂时访问，因此他能选择明文串 x 并构造出相应的密文串 y。

选择密文攻击：破译者 O 可暂时接近密码机，可选择密文串 y，并构造出相应的明文串 x。

选择密文攻击主要用于公钥密码体制，有时选择明文攻击和选择密文攻击一起称为选择文本攻击。

上述 4 种攻击的强度按序递增，唯密文攻击显然是比较困难的，但对于密码的编制者来说，设计一个密码能抵御唯密文攻击应该是最低的要求了。如果设计的密码系统能抵抗选择明文攻击，那么它肯定能抵抗唯密文攻击和已知明文攻击。

2.1.6　密码算法的安全性

密码算法的安全性有两种理解：无条件安全(Unconditionally secure)和计算上的安全(Computationally secure)。无条件安全指的是无论破译者有多少密文，他也无法解出对应的

明文，即使他解出了，也无法验证结果的正确性。有一种理想的加密方案，叫做一次一密密码(One-time pad)。它是由 Major Joseph Mauborgne 和 AT&T 公司的 Gilbert Vernam 在 1917 年发明的。一次一密密码本是一个大的不重复的真随机密钥字母集，每个密钥仅对一个消息使用一次，该体制的主要问题是密码本的安全分配和存储问题。除了一次一密之外，所有的加密算法都不是无条件安全的。使用者应尽量挑选满足下列要求的算法：

- 破译的代价超出信息本身的价值。
- 破译的时间超出了信息的有效期。

满足上述两条标准的加密体制是计算上安全的。

可以从以下三方面来衡量攻击方法的复杂性：

- **数据复杂性**(Data complexity)：用做攻击输入所需要的数据量。
- **处理复杂性**(Processing complexity)：完成攻击所需要的时间。
- **存储需求**(Storage requirement)：进行攻击所需要的存储量。

密码分析者攻击密码的两个基本方法是穷举法和分析法。

穷举法(Exhaustive attack method)，又称为强力法(Brute-force method)，完全试凑法(Complete trial-and-error method)。可以采用两种做法，一种是对截获的密文依次用各种可能的密钥破译，一种是对所有可能的明文加密直到与截获的密文一致为止。分析法又分为统计分析法(系统分析法)和确定性分析法，统计分析法通过分析明文和密文的统计规律来破译密码，确定性分析法指针对加密算法的数学依据，通过数学求解的方法来破译密码。

2.2 密码学的历史

密码是一项有着久远历史的技术，它伴随着战争的出现而出现，古代的密写术或隐蔽书写是它的起源。虽然这一技术由来已久，但在 20 世纪 60 年代之前，密码还仅仅是政府机关和军事部门的专利。20 世纪 70 年代以后，计算机科学和技术的发展促使密码学从外交和军事领域走向公开，开始应用于银行等商业部门，形成为一门新的学科。在计算机网络深入普及的今天，密码学成为了一个非常活跃的研究热点，研究密码算法的数家公司都因为互联网的发展而身价倍增。

密码学的发展可以划分为三个阶段：

第一阶段为1949年之前，这一时期的密码学还不是科学，可以说是一门艺术，出现了一些密码算法和加密设备。加密算法属于古典密码，密码算法的基本手段是针对字符的代替(Substitution)和置换(Permutation)，也出现了一些简单的密码分析手段。

公元前1900年，在古埃及，象形文字已经发明，并得到普遍使用，为克努姆霍特普二世撰写碑文的祭司别出心裁，用了一些奇怪的象形符号来代替通常所用的象形文字，这种对标准书写符号的修改是手写密码第一次有记载的使用情况，图2.3是古埃及的原始密码。这个例子蕴涵了一个密码变换的基本思想：代替，一种符号(明文)用另一种符号(密文)代替。

公元前487年，斯巴达人发明了Spartan Scytale，它是斯巴达人用于加解密的一种军事设备，发送者把一条羊皮螺旋形地缠在一个圆柱形棒上。消息沿着棒子的长度方向从左至右书写，写完一行，旋转木棒，再从左至右书写，直至写完，然后把羊皮带从木棒上解下展开。

解密过程就是把羊皮条缠绕在相同直径的木棒上，便可以读出明文。这个例子蕴涵了密码变换的另一个基本思想：置换，按一定规则把明文中的字符变换一个位置，重新排列。假设明文是：start attack at eleven，去掉空格，按行书写，每行至多写五个字符，如下表所示：

s	t	a	r	t
a	t	t	a	c
k	a	t	e	l
e	v	e	n	

当羊皮带解开后，读到的密文为：sakettavatteraentcl。

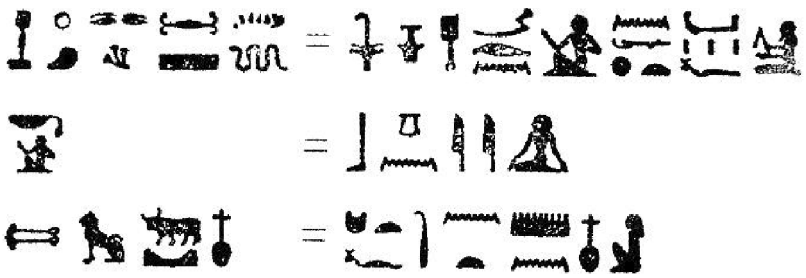


图 2.3　古埃及的原始密码(左方是密文，右方是相应的明文)

第二阶段为1949年至1975年，这一时期，计算机使得基于复杂计算的密码成为可能，数据的安全基于密钥而不是算法的保密。1949 年 Shannon 发表了“保密系统的通信理论”(The Communication Theory of Secret Systems)，该文首先将信息论引入了密码，利用数学方法对信息源、密钥源、接收和截获的密文进行了数学描述和定量分析，提出了通用的秘密钥密码体制模型，奠定了密码学的理论基础，从此密码学成为一门科学。这一时期的一些重要事件有：1967 年 David Kahn 出版了一本专著《破译者》(The Codebreakers)，该书记载了 1967 年之前密码学发展的历史，使许多不知道密码学的人了解了密码学。1971年至1973 年 IBM Watson 实验室的 Horst Feistel 和他的同事们发表了几篇技术报告，这些报告是 DES 算法提出的基础。

第三个阶段为1976年以后，1976年Diffie 和 Hellman 在“密码学的新方向”(New Directions in Cryptography)一文中，提出了不对称密钥密码的思想，首次证明发送端和接收端无密钥传输的保密通信是可能的，但他们在这篇文章中并没有提出一个真正实用的算法，直到 1977 年，Rivest，Shamir 和 Adleman才提出了 RSA 公钥算法，这是第一个真正实用的公钥密码体制。该算法的三个人因此获得了计算机界的诺贝尔奖——图灵奖。1990 年后逐步出现椭圆曲线等其他公钥算法。同一时期，对称密钥密码算法也获得了进一步的发展，1977年DES 正式成为数据加密标准，该算法采用完全公开的加密、解密体制，并批准用于非机密单位和商业上的保密通信，此举使得密码学在商业等民用领域得到广泛应用。随后出现了一些分组加密算法，如 IDEA，RCx，CAST 等。20 世纪 90 年代对称密钥密码进一步成熟，出现了 Rijndael, RC6, MARS, Twofish, Serpent 等算法，2001年Rijndael 成为高级数据加密标准 AES，取代已经变得越来越不安全的 DES。

现代密码学的另一个重要标志是基于计算复杂性理论的密码算法安全性证明。姚期智教授因为在保密通信计算复杂性理论上的重大贡献获得了图灵奖。在密码分析领域，王小云教授对经典散列算法 MD5、SHA-1 等的破解也是近年密码学的重大进展。

2.3 古典密码

古典密码是密码学的渊源，这些密码大都比较简单，可用手工或机械操作实现加解密，现在已很少采用了。然而，了解这些密码的原理，对于理解、分析现代密码和构造新的密码都是十分有益的。我们将从密码编码和密码分析两个方面对古典密码进行介绍。

古典密码分析的两个基本方法是穷举法和统计分析法。

单钥加密的古典算法有简单代换、多表代换、同态代换、多码代换、乘积密码等多种，是计算机出现之前的基于字符的密码。密码变换的基本手段是字符之间互相代替或者相互之间换位。现代的密码算法是对二进制位而不是对字母进行变换。大多数好的算法仍然是代替和换位的组合。

根据密码变换的规则，可以把古典密码分为代替密码(Substitution cipher)和置换密码(Permutation cipher)两大类。代替密码就是明文中的每一个字符被替换成密文中的另一个字符，接收者对密文做反向替换就可以恢复出明文。置换密码，又称换位密码(Transposition cipher)，明文的字母保持相同，但顺序被打乱了。

2.3.1 古典密码的数学基础

在给出古典密码算法之前，先回顾一下算法中用到的一些数学概念。

2.3.1.1 同余

给定任意整数 a 和 q，以 q 除 a，余数是 r，则可以表示为 $a = sq + r, 0 \leqslant r < q$，其中 $s = \lfloor a/q \rfloor$，表示不大于 a/q 的最大整数。定义 **r 为 a mod q 的剩余**，记为 $r \equiv a \bmod q$。若整数 a 和 b 有 $(a \bmod q) = (b \bmod q)$，则称 a 与 b 在 mod q 下**同余**。对于满足 $[r] = \{a | a = sq + r,\ s \in Z\}$ 的整数集称为**同余类**。

对给定的余数 r 和自然数 $q>0$，集合 $\{a |\ a \equiv r \bmod q\}$ 称为 **r 模 q 的剩余类**，它包含除以 q 后余数为 r 的所有整数。

对模 q 的每个剩余类恰好选一个代表，它们构成一个**完全剩余系**。模 q 的**最小非负完全剩余系**是集合 $\{0, 1, 2, \cdots, q-1\}$。

2.3.1.2 模运算

在集合 $\{0, 1, 2, \cdots, q-1\}$ 上可以进行算术运算，就是所谓的**模运算**。

定义加法和乘法运算如下：

模加：$[(a \bmod q) + (b \bmod q)] \bmod q = (a+b) \bmod q$

模减：$[(a \bmod q) - (b \bmod q)] \bmod q = (a-b) \bmod q$

模乘：$[(a \bmod q) \times (b \bmod q)] \bmod q = (a \times b) \bmod q$

模运算有下述性质：

(1) 若 $q|(a-b)$，则 $a \equiv b \pmod q$

(2) $(a \bmod q) = (b \bmod q)$ 意味 $a \equiv b \bmod q$

(3) 对称性，$a \equiv b \bmod q$ 等价于 $b \equiv a \bmod q$

(4) 传递性，若 $a \equiv b \bmod q$ 且 $b \equiv c \bmod q$，则 $a \equiv c \bmod q$

2.3.1.3　逆元

类似普通的加法，在模运算中，每个数也存在**加法逆元**，或者称为**相反数**。一个数 x 的加法逆元 y 是满足 $x+y \equiv 0 \bmod q$ 的数。

例如，$q=8$，$2+6 \equiv 0 \bmod 8$，在模 8 的情况下 2 和 6 就互为加法逆元。

在通常的乘法中，每个数存在**乘法逆元**，或者称为**倒数**。在模 q 的运算中，一个数 x 的乘法逆元 y 是满足 $x \times y \equiv 1 \bmod q$ 的数。但是并不是所有的数在模 q 下都存在乘法逆元。

例如 $3 \times 3 \equiv 1 \bmod 8$，在模 8 的情况下 3 的乘法逆元是 3。

2.3.1.4　Z_q 中的整数模运算性质

定义集合 Z_q 为小于 q 的所有非负整数集合：$Z_q = \{0, 1, 2, \cdots, q-1\}$。该集合也可看做模 q 的余数集合。如果在该集合上实行模运算，Z_q 中的整数保持如下性质：

(1) 交换律：　$(w+x) \bmod q = (x+w) \bmod q$

$(w \times x) \bmod q = (x \times w) \bmod q$

(2) 结合律：　$[(w+x)+y] \bmod q = [w+(x+y)] \bmod q$

$[(w \times x) \times y] \bmod q = [w \times (x \times y)] \bmod q$

(3) 分配律：　$[w \times (x+y)] \bmod q = [(w \times x) + (w \times y)] \bmod q$

(4) 恒等：　$(0+w) \bmod q = w \bmod q$

$(1 \times w) \bmod q = w \bmod q$

(5) 对每一个 $w \in Z_q$，存在 z，使得 $w+z \equiv 0 \bmod q$，z 称为 w 的加法逆元，记为 $-w$。

(6) 若 a 与 q 互素，如果 $(a \times b) \bmod q = (a \times c) \bmod q$，那么 $b \equiv c \bmod q$。

(7) 如果 q 是一个素数，对每一个 $w \in Z_q$，都存在 z，使得 $w \times z \equiv 1 \bmod q$，$z$ 称做 w 的乘法逆元 w^{-1}。

模 26 的最小非负完全剩余系，是模 26 的余数集合为 $\{0, 1, 2, \cdots, 25\}$。可以把字母表与模 26 的余数集合等同，参见表 2.2，在此基础上对字符进行运算。

表 2.2　英文字母与模 26 的剩余之间的对应关系

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

2.3.1.5　逆阵

设 $\boldsymbol{A}=(a_{ij})$ 为一个 $l \times m$ 矩阵，$\boldsymbol{B}=(b_{jk})$ 为一个 $m \times n$ 矩阵，可以定义矩阵乘法 $\boldsymbol{AB}=(c_{ik})$ $(1 \leqslant i \leqslant l,\ 1 \leqslant k \leqslant n)$ 为如下形式：

$$c_{ik} = \sum_{j=1}^{m} a_{ij} b_{jk}$$

矩阵乘法满足结合律，$(\boldsymbol{AB})\boldsymbol{C} = \boldsymbol{A}(\boldsymbol{BC})$，但不满足交换律，$\boldsymbol{AB} = \boldsymbol{BA}$ 并不成立。

$m \times m$ 矩阵中，有一个特殊矩阵，其主对角线的值为 1，其余均为 0，称其为**单位矩阵** $\boldsymbol{I}_m$。如 2×2 单位矩阵具有如下形式：

$$\boldsymbol{I}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$m \times m$ 矩阵 $\boldsymbol{A}$ 的逆矩阵 $\boldsymbol{A}^{-1}$ 如果存在，则应满足 $\boldsymbol{A}\boldsymbol{A}^{-1}=\boldsymbol{A}^{-1}\boldsymbol{A}=\boldsymbol{I}_m$，并且如果 $\boldsymbol{A}^{-1}$ 存在，其一定是唯一的。

设 $\boldsymbol{A}=(a_{ij})$ 为一个 $m \times m$ 矩阵，对 $1 \leqslant i \leqslant m$，$1 \leqslant j \leqslant m$，定义 $\boldsymbol{A}_{ij}$ 为从矩阵 $\boldsymbol{A}$ 删除第 i 行，第 j 列后的新矩阵。$\boldsymbol{A}$ 的行列式的值，一般记为 $\det\boldsymbol{A}$。

$$\det \boldsymbol{A}=\sum_{j=1}^{m}(-1)^{i+j} a_{ij} \det \boldsymbol{A}_{ij}$$

一个实矩阵可逆当且仅当其所对应的行列式的值非零。

矩阵 $\boldsymbol{K}$ 在模 26 情形下存在可逆矩阵的充分必要条件是 $\gcd(\det K, 26)=1$，$\gcd(a, b)$ 表示 a 和 b 的最大公因子。

设 $\boldsymbol{K}=(k_{ij})$ 为一个定义在 Z_n 上的 $m \times m$ 矩阵，若 $\boldsymbol{K}$ 在 Z_n 上可逆，则有 $\boldsymbol{K}^{-1}=(\det\boldsymbol{K})^{-1}\boldsymbol{K}^{*}$，这里 $\boldsymbol{K}^{*}$ 为矩阵 $\boldsymbol{K}$ 的伴随矩阵。伴随矩阵 $\boldsymbol{K}^{*}$ 其第 i 行，第 j 列的取值为 $(-1)^{i+j}\det\boldsymbol{K}_{ji}$，$\boldsymbol{K}_{ji}$ 为从矩阵 $\boldsymbol{K}$ 删除第 j 行，第 i 列后的新矩阵。

2.3.2 代替密码

一个代替密码由代替规则、密文所用字符、明文中被代替的基本单位三个因素决定。通常假设明文和密文所用的字符相同，根据明文中被代替的基本单位可以把代替密码分为单字母密码和多字母密码。

单字母密码(Monoalphabetic cipher)又称简单代替密码(Simple substitution cipher)，明文的一个字符用相应的一个密文字符代替，而且密文所用的字符与明文所用字符属同一语言系统。单字母密码又可划分为单表代替密码和多表代替密码。就是在对明文消息的代替过程中使用单一代替表还是多个代替表。

多字母密码(Polyalphabetic cipher)的明文中的字符映射到密文空间的字符还依赖于它在上下文中的位置，也即明文的多个字符用相应的多个密文字符代替。

2.3.2.1 单表代替密码

单表代替密码就是将明文字母表 M 中的每个字母用密文字母表 C 中的相应字母来代替。典型的单表代替密码有：移位密码(Shift cipher)、乘数密码(Multiplicative cipher)、仿射密码(Affine cipher)、多项式密码(Polynomial cipher)、密钥短语密码(Key word cipher)。

下面分别对这些典型的单表代替密码进行介绍。

1．移位密码

设 $P=C=K=Z_{26}$，这里 P 代表明文空间。对 $k \in K$，定义

$$y=e_k(x)=x+k \pmod{26} \qquad (y \in C)$$

$$x=d_k(y)=y-k \pmod{26} \qquad (x \in P)$$

当 $k=3$ 时，为恺撒密码。设明文为 cipher，经恺撒密码加密的密文为 FLSKHU。

实际算法为：$\forall x \in P$ 有 $e_3(x)=x+3 \pmod{26}=y$，同时有，$d_3(y)=y-3 \pmod{26}$。

给定加密的消息：PHHW PH DW WKH SDUWB。

由于(a)加解密算法已知；

(b)可能尝试的密钥只有 25 个(不包括 0)。

通过强力攻击可以得到明文：meet me at the party。

移位密码很容易受到唯密文攻击。

2．乘数密码

加密函数取形式为 $e_k(x)=kx \pmod{26}$，$k\in Z_{26}$，要求唯一解的充要条件是 $\gcd(k,\ 26)=1$。该算法描述如下：

设　　$P=C=Z_{26}$，$K=\{k\in Z_{26} \mid \gcd(a,\ 26)=1\}$，

对 $k\in K$，定义

$$e_k(x)=kx \pmod{26}$$

$$d_k(y)=k^{-1}y \pmod{26}，\ (x,\ y\in Z_{26})$$

当 $k=9$，对应的密码表为

表 2.3　$k=9$ 时乘数密码表

明文字母	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
密文字母	A	J	S	B	K	T	C	L	U	D	M	V	E	N	W	F	O	X	G	P	Y	H	Q	Z	I	R

设明文为 cipher，经过加密得到密文为 SUFLKX。

对于乘数密码，当且仅当 a 与 26 互素时，加密变换才是一一映射的，若 a 和 26 不互素，则会有一些明文字母被加密成相同的密文字母，而且不是所有的字母都会出现在密文字母表中。因此 a 的选择有 11 种：$a=3, 5, 7, 9, 11, 15, 17, 19, 21, 23, 25$，可能尝试的密钥只有 11 个。

3．仿射密码

加密函数取形式为 $e_k(x)=ax+b \pmod{26}$，其中 $k\in K$，$K=\{(a, b)\in Z_{26}\times Z_{26} \mid \gcd(a,\ 26)=1\}$。该算法描述为：

设 $P=C=Z_{26}$

对 $k=(a, b)\in K$，定义

$$e_k(x)=ax+b \pmod{26}$$

$$d_k(y)=a^{-1}(y-b) \pmod{26}，\quad (x,\ y\in Z_{26})$$

$q=26$ 时，可能的密钥是 $26\times 12-1=311$ 个。

例如，设 $k=(7, 3)$，注意到 $7^{-1} \pmod{26}=15$，加密函数是 $e_k(x)=7x+3$，相应的解密函数是 $d_k(y)=15(y-3)=15y-19$，易见

$$d_k(e_k(x))=d_k(7x+3)=15(7x+3)-19=x+45-19=x \pmod{26}$$

若明文为 pku ，首先把字母 p，k，u 转换成为数字 15，10，20，然后加密：

$$7\times\begin{bmatrix}15\\10\\20\end{bmatrix}+\begin{bmatrix}3\\3\\3\end{bmatrix}=\begin{bmatrix}4\\21\\13\end{bmatrix}=\begin{bmatrix}E\\V\\N\end{bmatrix}\pmod{26}$$

解密过程为：

$$15\times\begin{bmatrix}4\\21\\13\end{bmatrix}-\begin{bmatrix}19\\19\\19\end{bmatrix}=\begin{bmatrix}15\\10\\20\end{bmatrix}\pmod{26}$$

4．密钥短语密码

仿射密码虽然比移位密码和乘数密码的密钥空间增大了许多，但还是不能抵抗强力攻击。密钥短语密码是以西文单词为密钥的换字表，例如：取university 为密钥，首先找出其中发生

重复的字母，去掉重复字母 i，成为 universty，其次，字母一共 10 个，从第 11 个字母开始，用 universty 按顺序进行代替配置，然后把其余 17 个字母按自然顺序接在后面。这样得到以 university 为密钥的换字表，参见表 2.4。

表 2.4 以 university 为密钥的换字表

明文字母	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
密文字母	J	K	L	M	O	P	Q	W	X	Z	U	N	I	V	E	R	S	T	Y	A	B	C	D	F	G	H

5．任意的单表代替密码算法

当然，也可以构造非自然顺序配置的换字表，明文字母与代替他的密文字母毫无关联，那么整个换字表就是它的密钥，参见表 2.5。

表 2.5 非自然续序配置的换字表

明文字母	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
密文字母	E	O	Z	Q	F	H	P	J	Y	B	N	K	A	X	L	T	S	M	C	W	D	U	I	G	V	R

设 $P=C=Z_{26}$，K 是由 26 个符号 0, 1, …, 25 的所有可能置换组成。对任意 $\boldsymbol{\pi} \in K$，定义 $e_{\pi}(x)=\boldsymbol{\pi}(x)=y$ 且 $d_{\pi}(y)=\boldsymbol{\pi}^{-1}(y)=x$，$\boldsymbol{\pi}^{-1}$ 是 $\boldsymbol{\pi}$ 的逆置换。

置换 $\boldsymbol{\pi}$ 的表示如下：

$$\boldsymbol{\pi}=\begin{bmatrix} 0 & 1 & 2\cdots & 24 & 25 \\ 0' & 1' & 2'\cdots & 24' & 25' \end{bmatrix}$$

任意的单表代替密码的密钥空间 K 很大，置换数目达 $26! \approx 4 \times 10^{26}$，破译者进行穷举搜索是不行的，然而，可用统计的方式破译它。

实际上，移位密码、乘数密码、仿射密码算法都是任意单表代替密码的特例。

6．单表代替密码分析

从以上介绍可以看到，密码设计是利用数学来构造密码。而密码分析除了依靠数学、工程背景、语言学知识外，还要靠经验、统计、测试、眼力、直觉判断力……甚至有时还要靠点运气。密码破译在原则上遵循观察与经验，在方法上采用归纳与演绎，采取的步骤是分析、假设、推测和证实。语言的三大统计特性：语言的频率特征、连接特征和重复特征是进行密码破译的重要依据。

(a)语言的频率特征

在各种语言中，各个字母的使用次数是不一样的，有的偏高，有的偏低，这种现象叫**偏用现象**。对英文的任何一篇文章，一个字母在该文章中出现的次数称为这个字母(在这篇文章中)的**频数**。一个字母的频数除以文章的字母总数，就得到字母的**使用频率**。

美国密码学家 William Friedman 在调查了大量英文资料后，得出英文字母的普遍使用频率，参见表 2.6。

从表中可以看出，不同字母的使用频率是不同的，英文字母 e 的使用频率最高，而 z 等字母的使用频率最低，根据字母的使用频率，可以对它们进行分类，如表 2.7 所示。

除了普遍的使用频率特征外，还有开头结尾的特征。有些文章的开头和结尾受到固定格式的限制，有时文章中间的某些特定部位，某些字母也会出现较高的使用频率。

使用频度最高的前 30 个双字母如表 2.8 所示。

表 2.6　英文字母的普遍使用频率

字　母	频　率	字　母	频　率	字　母	频　率
A	0.0856	J	0.0013	S	0.0607
B	0.0139	K	0.0042	T	0.1045
C	0.0279	L	0.0339	U	0.0249
D	0.0378	M	0.0249	V	0.0092
E	0.1304	N	0.0707	W	0.0149
F	0.0289	O	0.0797	X	0.0017
G	0.0199	P	0.0199	Y	0.0199
H	0.0528	Q	0.0012	Z	0.0008
I	0.0627	R	0.0677		

表 2.7　英文单字母使用频率分类表

分　类	使用频率分类字母集	每个字母约占百分数
I	极高使用频率字母集：e	13%
II	次高使用频率字母集：t,a,o,i,n,s,h,r	6%～9%
III	中使用频率字母集：d,l	4%
IV	低使用频率字母集：c, u,m,w,f,g,y,p,b	1.5%～2.3%
V	次低使用频率字母集：v, k, j, x, q, z	1%

表 2.8　使用频度最高的前 30 个双字母

TH	HE	IN	ER	AN	RE	ED	ON	ES	ST
EN	AT	TO	NT	HA	ND	OU	EA	NG	AS
OR	TI	IS	ET	IT	AR	TE	SE	HI	OF

使用频度最高的前 20 个三字母如表 2.9 所示。

表 2.9　频度最高的前 20 个三字母

THE	ING	AND	HER	ERE	ENT	THA	NTH	WAS	ETH
FOR	DTH	HAT	SHE	ION	INT	HIS	STH	ERS	VER

此外，还有如下一些特征：

THE 的使用频率最高，是 ING 的三倍，若把 the 去掉，t 在第 II 类中排在最后，h 会降为第 III 类，th 和 he 也不是常出现的字母组了。一半的单词以 e, s, d, t 结尾，一半的单词以 t, a, s, w 开头，y 的使用频率 90%都集中在单词的结尾。

(b) 语言的连接特征

语言的第二个特征是连接特征，连接特征有如下几种：

- **后连接**：字母 q 后面除了连接省略号外，几乎百分之百连接着 u。
- **前连接**：有些字母的前面总喜欢连接那么少数几个字母，如字母 x 的前面总是 i 和 e，很少是 o 和 a。
- **间断连接**：在 e 和 e 之间，r 的出现频率最高。

(c) 语言的重复特征

两个字符以上的字符串重复出现的现象，叫做语言的重复特征。例如英语中 th，tion，

tious 等就是重复出现的字符串。1863年，普鲁士军官Kasiski利用这一现象提出了一种重码分析法。

为了说明分析的过程，在这里给出一个例子。假设需要解密的密文是：

QRLLIQQPFVICXPFMTPZWRFNFOTFLPYWIGQPQHICQRGIVAKZWIQIBORGZWPFMQ
PFZWIOGVIGFCHIVYIGQIJIGCFLILCGIBRXHIZWOVQOBCFCXKQPQPFZRPZPOFXRLNZWI
CAPXPZKCZXICQZZOGICVZWIXCFMRCMIOBZWIOGPMPFCXZISZPQJIGKVIQPGCAXIARZ
FOZIQQIFZPCX

这份密文一共 206 个字母，先统计出密文中每个字母使用的次数，将密文字母按频数从多到少排列，参见表 2.10。

表 2.10　密文字母的使用次数

密文字母	I	Z	P	Q	C	F	G	O	X	W	R	V	L
使用次数	28	19	18	17	16	16	12	10	10	9	9	7	6
密文字母	M	B	K	A	H	N	J	Y	T	S	D	E	U
使用次数	5	4	4	4	3	2	2	2	2	1	0	0	0

将这种统计规律与表 2.6 和表 2.7 比较，可以得出的结论是：密文字母中的 I 可能相当于明文中的 e，ZPQCFGOX 可能对应于第 II 类字母集{t, a, o, i, n, s, h, r}中的某一个，WRV 可能对应于第 III 类字母集{d, l}中的某一个，LMBKAH 可能对应于第 IV 类字母集{c, u, m, w, f, g, y, p, b}中的某一个，NJYTSDEU 可能对应于第 V 类字母集{v, k, j, x, q, z}中的某一个。

这里最常见的双字母组合 ZW，出现了 8 次，把 Z 猜测其为 T，W 对应为 H，I 猜测其为 E。密文中的 ZWI 很可能就是 THE。次常见的双字母组合为 WI 和 PF，两者均出现了 6 次，因为已经把 WI 猜测为 HE，使用频度为第三的双字母组合为 IN，所以把 PF 猜测为 IN。

至此，有如下结果：

```
QRLLIQQPFVICXPFMTPZWRFNFOTFLPYWIGQPQH
    e  in e  in  ith n n  n i he  i
ICQRGIVAKZWIQIBORGZWPFMQPFZWIOGVIGFCH
e    e   the e    thin  inthe   e n
IVYIGQIJIGCFLILCGIBRXHIZWOVQOBCFCXKQPQPF
e  e  e e  n e   e    eth      n    i in
ZRPZPOFXRLNZWICAPXPZKCZXICQZZOGICVZWIXCFMRCM
t iti n    the  i it  t e  tt  e  the  n
IOBZWIOGPMPFCXZISZPQJIGKVIQPGCAXIARZFOZIQQIFZPCX
e  the  i in  te ti  e   e i    e  tn te  enti
```

这样一来，QCGOX 可能对应于第 II 类字母集{a, o, s, h, r}中的某一个，字母组合 ZWIQI 与 these 相似，可以把 Q 猜测为 S。字母组合 TPZW 与 with 类似，所以可以把 T 猜测为 W。字母组合 IQQIFZPCX 与 essential 相似，据此把 C 猜测为 A，X 猜测为 L。字母组合 CFCXKQPQ 与 analysis 类似，因此，可以把 K 猜测为 Y。字母组合 CAPXPZK 与 ability 类似，据此把 A 猜测为 B。

至此，分析结果如下：

```
QRLLIQQPFVICXPFMTPZWRFNFOTFLPYWIGQPQH
su  essin ealingwith n n wn i he sis
ICQRGIVAKZWIQIBORGZWPFMQPFZWIOGVIGFCH
easu e bythese  u thin sinthe   e na
```

```
IVYIGQIJIGCFLILCGIBRXHIZWOVQOBCFCXKQPQPF
e  e se e an e a e ul eth  s  analysisin
ZRPZPOFXRLNZWICAPXPZKCZXICQZZOGICVZWIXCFMRCM
tuiti nlu  theabilityatleastt  ea thelanguag
IOBZWIOGPMPFCXZISZPQJIGKVIQPGCAXIARZFOZIQQIFZPCX
e  the  iginalte tis e y esi ablebutn tessential
```

按着，发现字母组合 ZWPFM 与 things 类似，据此把 M 猜测为 G。字母组合 XCFMRCMI 与 language 类似，据此把 R 猜测为 U。字母组合 VICXPFM 与 dealing 类似，据此把 V 猜测为 D。

至此，分析结果如下：

```
QRLLIQQPFVICXPFMTPZWRFNFOTFLPYWIGQPQH
su  essindealingwith n n wn i he sis
ICQRGIVAKZWIQIBORGZWPFMQPFZWIOGVIGFCH
easu edbythese  u thin sinthe  de na
IVYIGQIJIGCFLILCGIBRXHIZWOVQOBCFCXKQPQPF
ed e se e an e a e ul eth ds  analysisin
ZRPZPOFXRLNZWICAPXPZKCZXICQZZOGICVZWIXCFMRCM
tuiti nlu  theabilityatleastt  eadthelanguag
IOBZWIOGPMPFCXZISZPQJIGKVIQPGCAXIARZFOZIQQIFZPCX
e  the  iginalte tis e ydesi ablebutn tessential
```

然后，发现字母组合 QRLLIQQ 与 success 类似，据此把 L 猜测为 C。字母组合 GICV 与 read 类似，据此把 G 猜测为 R。字母组合 LCGIBRX 与 careful 类似，据此把 B 猜测为 F。字母组合 HICQRGIV 与 measured 类似，据此把 H 猜测为 M。字母组合 HIZWOVQ 与 methods 类似，据此把 O 猜测为 O。

继续分析，更进一步把 JIGK 猜测为 very，把 YIGQIJIGCFLI 猜测为 perseverance，把 RFNFOTF 猜测为 unknown，把 ZISZ 猜测为 text。

这样就得到如下的完整明文：

Success in dealing with unknown ciphers is measured by these four things in the order named：perseverance，careful methods of analysis，intuition，luck. The ability at least to read the language of the original text is very desirable but not essential.

其中文意思是：能否成功地破译未知的密码，在于以下 4 个因素，它们依次为：锲而不舍的精神，周密的分析方法，直觉和运气。至少能阅读原文语言的能力是十分需要的，但不是主要的。这段话是帕克• 希特的《军事密码破译手册》的开场白，揭示了密码分析的四要素。

根据以上信息，现在就可以给出这个加密过程所使用的加密变换的明密文字母的对应表，如表 2.11 所示。

表 2.11　明密文字母对应表

明文字母	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
密文字母	C	A	L	V	I	B	M	W	P			N	X	H	F	O	Y		G	Q	Z	R	J	T	S	K

观察到 VWXYZ 在字母中以 4 间隔开，所以把密文字母按从上到下，从左到右的顺序写到一个四行的表格中。

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

字母DEU并没有在密文中出现，但通过观察字母排列的规律，ciphers的意思是密码。其他字母刚好是按字母顺序的一个排列，这样就可以把空缺的字母填到空格中。看到这个密码实际上是在以 ciphers 为密钥字的密钥短语密码的基础上又进行了置换。

单表密码破译的过程实际上是一个假定，推翻，再假定，再推翻，直至破译的过程，可以按照以下步骤进行。

(1) 对密文字母的频数、使用频率和连接次数进行统计。

(2) 根据了解到的密码编制人员的语言修养，以及手中掌握的密文的统计规律，多方比较，对明文的语种和密码种类做出假定。

(3) 将假定语种的字母频率与密文字母频率进行比较。

(4) 首先找出密文中频率最高的字母。

(5) 根据字母的频率特征、连接特征、重复特征，从最明显的单词或字母开始，试探进行。

(6) 对破译的经验进行总结。

简单代替密码由于使用从明文到密文的单一映射，所以明文字母的单字母出现频率分布与密文中相同，即使使用任意单表代替的方法使密钥空间扩大到强力攻击难以进行的程度，仍然可以较容易地通过使用统计分析法进行唯密文的攻击。为了对抗频率分析，出现了多名或同音代替密码、多表代替密码和多字母代替密码，它们都是通过不同的途径试图减少密文所保留的明文的统计特性。

2.3.2.2 多名代替密码

与简单代替密码类似，只是映射是一对多的，每个明文字母可以加密成多个密文字母。例如，A 可能对应于 5, 12, 21, B 可能对应于 7, 9, 18, 27。当对字母的赋值个数与字母出现频率成比例时，密文符号的相关分布会近似于平的，可以挫败频率分析。然而，若明文字母的其他统计信息在密文中仍很明显时，那么同音代替密码仍然是可破译的。

2.3.2.3 多表代替密码

单字母出现频率分布与密文中相同，多表代替密码使用从明文字母到密文字母的多个映射来隐藏单字母出现的频率分布，每个映射是简单代替密码中的一对一映射。多表代替密码是以一系列(两个以上)代换表依次对明文消息的字母进行代换，可以分为非周期多表代替密码和周期多表代替密码。非周期多表代替密码的代换表是非周期的无限序列，例如一次一密密码，对每个明文每次采用不同的代换表。周期多表代替密码的代换表个数有限，重复使用。典型的多表代换密码有：维吉尼亚密码(Vigenére cipher)、博福特密码(Beaufort cipher)、滚动密钥密码(Running-key cipher)、弗纳姆密码(Vernam cipher)、转轮密码机(Rotor machine)。

1．维吉尼亚密码

该密码实际上是贝拉索在 1553 年发明的，19 世纪时被误认为是由法国密码学家维吉尼亚发明的，因此现在称为维吉尼亚密码，它是一种多表移位代替密码。

设 d 为一固定的正整数，d 个移位代换表 $\pi=(\pi_1, \pi_2, \cdots, \pi_d)$ 由密钥序列 $K=(k_1, k_2, \cdots, k_d)$ 给定，第 $i+td$ 个明文字母由表 π_i 决定，即密钥 k_i 决定：

$$e_k(x_{i+td}) = (x_{i+td} + k_i) \bmod q = y$$

$$d_k(y_{i+td}) = (x_{i+td} - k_i) \bmod q = x$$

例如：$q = 26$，x = polyalphabetic cipher，K = RADIO

明文 x=polyalphabeticcipher

密钥 k=RADIORADIORADIORADIO

密文 y=GOOGOCPKIPVTLKQZPKMF

我们称 $k = (k_1, k_2, \cdots, k_d)$ 是长为 d 的密钥字，密钥空间大小为 26^d，即使对于很小的 d，例如当 $d = 5$，$26^5 = 1.1 \times 10^7$，使用穷举密钥法也需要很长时间。

在Vigenére密码中，一个字母可被映射为 d 个可能的字母之一，所以分析起来比单表代换更困难，但算法依然保留了字符频率的某些统计信息，分析的第一步是确定密钥字的长度 d，确定 d 的方法最常用的有两种。一种叫 **Kasiski 测试法**，是普鲁士军官 Kasiski 在 1863 年提出的一种**重码分析法**，另一种方法是 William Friedman 在 1920 年提出的**重合指数法**(Index of coincidence)。

对于明文 To be or not to be that is the question(生存，还是死亡，这是个问题)，使用 Vigenére 密码加密，密钥字取为 run。

密钥：runrunrunrunrunrunrunrunrun

明文：tobeornottobethatisthequestion

密文：**KIOV**IEEIG**KIOVNU**RNVJ**NU**VKHVMGZIA

观察到密文字母串 KIOV 在密文中重复了两次，后一个相当于前一个向后移动了 9 个字母的距离，密文字母串 NU 也在密文中重复了两次，后一个相当于前一个向后移动了 6 个字母的距离，密钥字 RUN 的长度为 3，9 和 6 都是 3 的倍数。KIOV 的重复是因为密钥字母串 RUNR 的重复与明文字母串 tobe 的重复正好合拍。重复明文字母串的距离正好是密钥长度的倍数。合拍现象是意大利的乔瓦·尼·波塔于1602年首先发现的。但他没有很好地利用这个现象，直到 200 年后，Kasiski 抓住了这个现象，所以 Vigenére 密码被安全地使用了 200 年。间距是密钥长度整数倍的相同明文子串对应相同的密文，反过来，密文中两个相同子串对应明文相同的可能性很大，这就是**重码分析法**，这里面有偶然重复和真重复，但它们之间距离的因子可能就是密钥长度。Kasiski 提出，对一份用周期性多表密码加密的密文，确定其中所有重复出现的字母串，计算它们之间的距离，并对这些距离进行因子分解，出现频率较高的因子很可能是密钥的长度。这就是 **Kasiski 实验**。

为了解释重合指数法，先来看一个概率问题，设 x 是一个长度任意的英文字母串，即 $x = x_1x_2x_3\cdots x_m$，其中 x_i 是字母，m 不确定。现在问：从 x 中随机地抽取两个字母，它们为相同字母的概率是多少？答案是 $26 \times (1/26)^2 = 1/26 = 0.0385$。如果 x 是一篇随便选取的英语文章，令 p_1 表示 a 出现的概率，p_2 表示 b 出现的概率，……，p_{26} 表示 z 出现的概率，那么，抽到两个相同字母的概率就是 $p_1^2 + p_2^2 + \cdots + p_{26}^2 = 0.0687$。这说明了随机的字母串与作为语言的明文字母串之间的差别。一个事实是对于单表代替密码加密得到的密文，0.0687 这个数据不变。我们可以把 0.0687 作为标准来检验一个密文是不是单表代替的加密产物。

设 y 是一个长度为 n 密文，即 $y = y_1y_2y_3\cdots y_m$，其中 y_i 是密文字母，同样来求从中抽到两个相同字母的概率是多少？为此，设 N_A 为字母 A 在这份密文中的频数，设 N_B 为字母 B 在这份密文中的频数，依次类推。从 n 个密文字母中抽取两个字母的方式有 $C_n^2 = n(n-1)/2$，而其中 N_A 个 A 组成一对 A 的方式有：

$$C_{NA}^2 = N_A(N_A - 1)/2$$

于是从 y 中抽到两个字母都为 A 的概率为$[N_A(N_A-1)]/[n(n-1)]$，因此，从 y 中抽到两个相同字母的概率为：

$$\frac{\sum_{i=A}^{Z} N_i(N_i-1)}{n(n-1)}$$

这个数据称为这份密文的**重合指数**。根据概率论中的大数定律，如果 y 是用单表加密的，那么当 n 较大时，重合指数很可能接近于 0.0687。假设密钥字的长度是 d，把密文一行一行写在一个有 d 个列的表格里，对各列密文字母串计算相应的重合指数，以此为根据判断 d 的假设是否正确。

通过 Kasiski 实验、重合指数和直觉判断，最终可以确定 d。

2．滚动密钥密码

对于周期代换密码，当密钥的长度 d 和明文一样长时，就成为滚动密钥密码。Vigenére 本人建议密钥与明文一样长。一种设置密钥的方法是先设置一个密钥字，然后用明文来做加密的密钥。下面的例子中 beijing 为密钥字。

密钥：beijingmeetatnineinthe

明文：meetatnineintheevening

密文：NIMCIGTURIBNMUMRZMABUK

因为滚动密钥密码工作在流密码状态，根据密钥字可以把先收到的密文恢复成明文，然后就可以用已收到的密文对应的明文作为密钥字来解密接下来收到的密文。

即使采用这个方案，它也是易受攻击的，因为密钥和明文具有同样的频率分布特征。

3．弗纳姆密码

1918 年，Gillbert Vernam 建议密钥与明文一样长并且与明文没有统计关系，他采用的是二进制数据：

加密：$C_i = P_i \oplus K_i$

解密 $P_i = C_i \oplus K_i$

其中，P_i 是明文的第 i 个二进制位，C_i 是密文的第 i 个二进制位，K_i 是密钥的第 i 个二进制位，密文是明文和密钥逐位异或得到的。

弗纳姆密码的核心是构造和消息一样长的随机密钥。尽管周期很长增加了破解的难度，但是如果密文足够多，或者使用已知的明密文序列，该方案是可以破解的。

4．转轮密码机

加密和解密机的发明和商业化促进了密码学的发展，直到 19 世纪，密码机还是机械的，随着电传打字机的出现，电动密码机开始在保密通信中大显身手。在第二次世界大战中，转轮密码机的使用相当普遍。如日本制造的紫密(Purple)、德国制造的恩尼格玛(Enigma)和瑞典制造的哈格林密码机(Hagelin)，美国军方称为 M-209。

转轮密码机实际上是多表代替密码的一种实现，主要利用机械运动和简单电子线路。它有一个键盘和若干转轮，每个转轮由绝缘的圆形胶板组成，胶板正反两面边缘线上有金属凸块，每个金属凸块上标有字母，字母的位置相互对齐。胶板正反两面的字母用金属连线接通，形成一个代替运算。不同的转轮固定在一个同心轴上，它们可以独立自由转动，每个转轮可选取一定的转动速度。例如，一个转轮可能被导线连通以完成用 F 代替 A，用 U 代替 B，用 L 代替 C，等等。为了防止密码分析，有的转轮密码机还在每个转轮上设定不同的位置号，使得转轮的位置、转轮的数量、转轮上的齿轮结合起来，增大机器的周期。

德国人使用的恩尼格玛实现了一个非常复杂的可变代替密码，它使用一系列的转轮，每个转轮的内部连线各不相同。一个转轮的一个输入-输出连线状态给定了一种代替表，每加密一个字母，转轮转一格，输入-输出连线状态就跟着发生了变化，代替表就相应变一个。恩尼格玛共有 3 个转轮，在每个转轮的边缘上，标记着 26 个英文字母，每个转轮的 26 种内部连线状态可以通过这些字母的 26 种位置来表示。经过巧妙的设计，每次转轮旋转的时候，它都会停留在这 26 种位置之一。假设从转轮的左面“输入一个字母信号”，经过转轮内部特定走向的导线连接后，输出的字母信号也就不再对应刚才输入的字母了。输入的字母K，经过转轮内部的导线，最终从转轮的另一侧输出，并变成了R。在转轮组内，转轮相互接触的侧面之间，都有相对应的电路触点，可以保证转轮组的内部构成通路。于是，输入的字母 K，经过第一个转轮，变成输出字母 R；之后这个 R 进入第二个转轮，假设它又变成了 C；而后，这个C再进入第三个转轮，假设又变成了Y。这样，初始字母K历经层层变换，变成了谁也认不出来的Y。3 个转轮的变换相当于连续使用了 3 个代替表，因此，它们合起来连续加密的总效果就是 3 个转轮各自能力的乘积。每个转轮都有 26 个位置，3 个转轮组合起来，就能生成 26×26×26 = 17 576 种不同的变化。图 2.4 给出了恩尼格玛转轮组的示意图，为了简单，内部只画了三组连线。

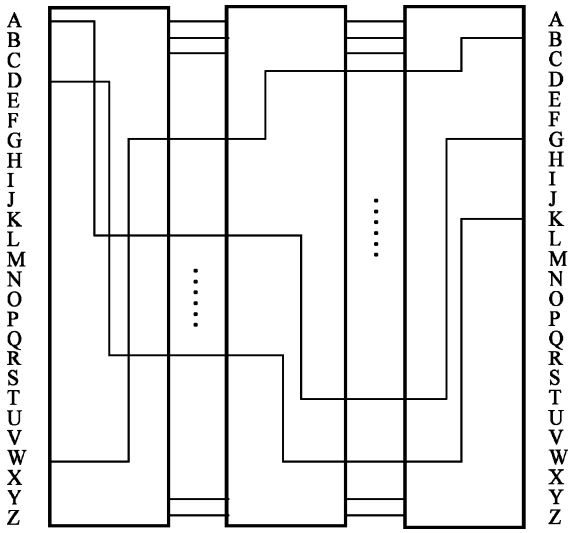


图 2.4　恩尼格玛转轮组的示意图

2.3.2.4　多字母密码

不同于前面介绍的代替密码都是每次加密一个明文字母，多字母代替密码将明文字符划分为长度相同的消息单元，称为明文组，对字符块成组进行代替，这样一来使密码分析更加困难。多字母代替的优点是容易将字母的自然频度隐蔽或均匀化，从而有利于抗击统计分析。因为可以用矩阵变换方便地描述多字母代换密码，有时又称其为矩阵变换密码。典型的多字母密码有 Hill 密码和 Playfair 密码。

1．Playfair 密码

Playfair 密码将明文中的双字母组合作为一个单元对待，并将这些单元转换为密文的双字母组合。Playfair 密码基于一个密钥词构成的 5×5 变换矩阵，I 与 J 视为同一字符。

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

本例中使用的密钥词是 cipher。填充矩阵的方法是先将密钥词(去掉重复字母)从左到右，从上到下填在空格中，再将剩余的字母按字母表的顺序从左到右，从上到下填在剩余的空格中。按成对字母加密的规则：

(1) 如果该字母对中的两个字母是相同的加分隔符，比如 x，例如 balloon 可以先变成 ba lx lo on。

(2) 如果该字母对落在矩阵的同一行，取它们右边的字母来代替，每行最右边的字母由最左边的字母代替，例如 he 变成 EC。

(3) 如果该字母对落在矩阵的同一列，取它们下面的字母来代替，每行最下边的字母由最上面的字母代替，例如 dm 变成 MT。

(4) 如果该字母对落在矩阵的其他位置，取交叉位置的字母代替，例如 kt 变成 MQ，OD 变成 TR。

下表给出了 Playfair 密码的一些加密实例。

明　文	分　组	密　文
balloon	ba lx lo on	db sp gs ug
book	bo ok	sr qg
fill	fi lx lx	ae sp sp

双字母替代比单字母替代能减少明文的结构仍保留在密文中的程度。首先，虽然仅有 26 个字母，Playfair 有 $26\times26=676$ 种字母对组合，因此识别各种双字母组合困难得多，此外，字符出现概率一定程度上被均匀化，基于字母频率的攻击比较困难，但依然保留了相当的结构信息。

2．Hill 密码

Hill 密码是基于矩阵的线性变换，假设 $\boldsymbol{K}$ 是一个 $m\times m$ 矩阵，在 Z_{26} 上可逆，即存在 $\boldsymbol{K}^{-1}$ 使得：$\boldsymbol{K}\boldsymbol{K}^{-1}=\boldsymbol{I}$(在 Z_{26} 上)， 对每一个密钥 $\boldsymbol{K}$，定义

$$e_{\boldsymbol{K}}(x)=x\boldsymbol{K}(\bmod 26) \text{ 和 } d_{\boldsymbol{K}}(y)=y\boldsymbol{K}^{-1}(\bmod 26)$$

这里，明文与密文都是 m 元的向量$(x_1,\ x_2,\ \cdots,\ x_m)$，$(y_1,\ y_2,\ \cdots,\ y_m)$，$Z_{26}$ 为模 26 的最小非负完全剩余系。在这个集合上的可逆矩阵 $\boldsymbol{A}_{m\times m}$ 是指行列式 $\det\boldsymbol{A}_{m\times m}$ 的值$\in Z_{26}^{*}$，它为 Z_{26} 中全体可逆元的集合，即 $Z_{26}^{*}=\{a\in Z_{26}|(a,26)=1\}=\{3,5,7,9,11,15,17,19,21,23,25\}$。

例如，当 $m=2$ 时，明文元素 $x=(x_1,\ x_2)$，密文元素 $y=(y_1,\ y_2)$

$$(y_1,\ y_2)=(x_1,\ x_2)\boldsymbol{K}$$

若 $\boldsymbol{K}=\begin{bmatrix}11 & 8\\ 3 & 7\end{bmatrix}$，可得 $\boldsymbol{K}^{-1}=\begin{bmatrix}7 & 18\\ 23 & 11\end{bmatrix}$

若对明文 hill 加密，它可分成 2 个元素 hi，ll，分别对应于[7　8]，[11　11]，有

$$[7\quad 8]\begin{bmatrix}11 & 8\\ 3 & 7\end{bmatrix}=[77+24\qquad 56+56]\bmod 26=\begin{bmatrix}23 & 8\end{bmatrix}$$

$$[11 \quad 11]\begin{bmatrix}11 & 8\\ 3 & 7\end{bmatrix}=[121+33 \quad 88+77]\bmod 26=\begin{bmatrix}24 & 9\end{bmatrix}$$

于是对 hill 加密的结果为 XIYJ。

为了解密，可计算

$$[23 \quad 8]\begin{bmatrix}7 & 18\\ 23 & 11\end{bmatrix}\bmod 26=[7 \quad 8]$$

$$[24 \quad 9]\begin{bmatrix}7 & 18\\ 23 & 11\end{bmatrix}\bmod 26=[11 \quad 11]$$

因此，得到了正确的明文 hill。

Hill密码完全隐藏了字符(对)的频率信息，采用唯密文攻击希尔密码是很难攻破的。但线性变换的安全性很脆弱，易被已知明文攻击击破。

对于一个 $m\times m$ 的 Hill 密码 $\boldsymbol{K}$，假定有 m 个明文-密文对，明文和密文的长度都是 m，可以把明文和密文对记为：$P_j=(p_{1j},\ p_{2j},\ \cdots,\ p_{mj})$ 和 $C_j=(C_{1j},\ C_{2j},\ \cdots,\ C_{mj})$，

$$C_j=P_j\boldsymbol{K},\ 1\leqslant j\leqslant m$$

定义 $m\times m$ 的方阵 $\boldsymbol{X}=(P_{ij})$，$\boldsymbol{Y}=(C_{ij})$，得到 $\boldsymbol{Y}=\boldsymbol{XK}\bmod 26$，$\boldsymbol{K}=\boldsymbol{X}^{-1}\boldsymbol{Y}\bmod 26$，若 $\boldsymbol{X}$ 不可逆，我们总可以找到一个可逆的 $\boldsymbol{X}$。

假设明文“friday”经过 2×2 的 Hill 密码加密为密文“PQCFKU”，因此，我们有[5 17]$\boldsymbol{K}$=[15 16]，[8 3]$\boldsymbol{K}$=[2 5]，[0 24]$\boldsymbol{K}$=[10 20]。那么由前两个明密文对可得：

$$\begin{bmatrix}15 & 16\\ 2 & 5\end{bmatrix}=\begin{bmatrix}5 & 17\\ 8 & 3\end{bmatrix}\boldsymbol{K}$$

$$\boldsymbol{X}^{-1}=\begin{bmatrix}5 & 17\\ 8 & 3\end{bmatrix}^{-1}=\begin{bmatrix}9 & 1\\ 2 & 15\end{bmatrix}$$

因此，$\boldsymbol{K}=\begin{bmatrix}9 & 1\\ 2 & 15\end{bmatrix}\begin{bmatrix}15 & 16\\ 2 & 5\end{bmatrix}\bmod 26=\begin{bmatrix}7 & 19\\ 8 & 3\end{bmatrix}$

此结果可以由第三个明密文对来验证。

2.3.3　置换密码

在换位密码中，明文的字母保持相同，但顺序被打乱了。由于密文字符与明文字符相同，密文中字母的出现频率与明文中字母的出现频率相同，密码分析者可以很容易地由此进行判别。虽然许多现代密码也使用换位，但由于它对存储要求很大，有时还要求消息为某个特定的长度，因而比较少用。把明文按行写入，按列读出是一种简单的换位方法。置换密码完全保留字符的统计信息，但使用多轮加密可提高安全性。

2.3.4　古典密码算法小结

根据前面的介绍，可以给出如图 2.5 所示的古典密码分类汇总图。

- ✧ 代替密码
  - » 简单代替密码(单字母密码)
    - • 单表代替密码
      - – 移位(shift)密码
      - – 乘数(multiplicative)密码
      - – 仿射(affine)密码
      - – 密钥短语(Key Word)密码
      - – 任意的单表代替密码
    - • 多表代替密码
      - – 维吉尼亚(Vigenére)密码
      - – 博福特(Beaufort)密码
      - – 滚动密钥(running-key)密码
      - – 弗纳姆(Vernam)密码
      - – 转轮密码机(rotor machine)
  - » 多字母密码：Hill, Playfair 密码
- ✧ 置换密码

图 2.5 古典密码分类汇总图

思考和练习题

(1) 什么特性使得加密绝对不可攻破？什么特性将使得加密在实际应用中是不可攻破的？

(2) 密码学的发展可以分为哪几个阶段，各有什么特点?

(3) 密码体制根据密钥和明文处理的方式可以分为哪几类？

(4) 解释古典密码分析的两个基本方法是什么？

(5) 解释实现古典密码的两个基本运算是什么？

(6) 解释密码编码的 Kerchoff 原则是什么，为什么基于密钥保密的算法更安全和实用？

(7) 解释为什么双字母替换密码比单字母替换更安全？

(8) 密文为 c，明文为 m，26 个字母编号为 0～25，加密算法为 $c = 7m + 11 \pmod{26}$，当明文为 hello 时，对应的密文是什么？

(9) 设 π 为集合{1, …, 8}上的置换：

x	1	2	3	4	5	6	7	8
$\pi(x)$	4	1	6	2	7	3	8	5

求出逆置换 π^{-1}。

实践/实验题

(1) 完成通用的仿射变换的加密和解密程序。并对明文 general 使用 $c = 3m + 5 \pmod{26}$ 进行加密，这里设密文为 c，明文为 m。对密文 FMXVEDK 使用 $m = 9c - 19 \pmod{26}$ 进行解密。

(2) 使用穷尽密钥搜索法，破译如下列用移位密码加密的密文：

BEEAKFYDJXUQYHYJIQRYHTYJIQFBQDUYJIIKFUHCQD

(3) 编程给出 Vigenére 密码的代换表。

第 3 章　现代对称密码

3.1　乘积密码

在第 2 章，了解到因为语言的特征，只用代替或置换规则构造的密码是不安全的。因此，可以考虑连续使用两个或两个以上基本密码的方式来增强密码的强度。**乘积密码**就是以某种方式连续执行两个或多个密码以使得到的最后结果从密码编码的角度比其任何一个组成密码都强，能够挫败基于统计分析的密码破译。在现代密码之前，转轮密码机是最普遍的乘积密码，在第二次世界大战中得到广泛应用。

两个加密方法按它们加密的顺序连接进行合成时，要求第一个方法的密文空间与第二个方法的明文空间一致。两个加密方法的合成(乘积加密)通常会形成一个新的加密方法，问题是两类方法的合成是否一定比单个方法更具有抗密码分析的能力？直觉是这样，但这不一定都对，因为第二个方法可以部分或全部抵消第一个方法的作用。

举例如下：设以“BASEDOW’S DISEASE IS CURABLE”为密钥字构造密钥短语密码，代替表为：

```
abcdefghijklmnopqrstuvwxyz
BASEDOWICURLFGHJKMNPQTVXYZ
```

重复两次代替后，代替表为：

```
abcdefghijklmnopqrstuvwxyz
ABNDEHVCSQMLOWIURFGJKPTXYZ
```

发现有 8 个字母，包括高频字母 e 和 a 都没有动。

那么什么样的方法合成是有效的呢？

某些密码体制 M 具有这样的性质，取自 M 的两个加密步的合成仍属于 M，也就是说这一**密码体制形成群**。比如 Z_{26} 上的所有移位代替加密方法形成群 P_{26}，宽度为 24 的所有置换加密形成的群 P_{24}。如果两个加密方法集合的每一个都构成群，且可交换，则它们的乘积加密也构成一个群。一个单字母代替和一个换位的复合是不可交换的，一个正常多字母代替(宽度为 k)和宽度为 k 的换位的合成也不满足交换性。如果合成的方法不仅不满足交换性，而且其中任何两个方法相互独立，则这一合成是有效的。比如换位，执行一次“扩散”，多字母代替，执行一次“混淆”。如果乘积加密不是一个群，它就可以被重复而且其组合复杂度会进一步增加。

两次代替可以构造一个更难以分析的代替，两次置换可以构造一个更难以分析的置换，代替之后再进行一次置换，可以构造一个强度更高的新密码，这是古典密码通往现代密码的桥梁。

3.2 对称分组密码的设计原理与方法

3.2.1 对称分组密码的三个安全设计原则

分组密码将明文消息编码表示后的数字(简称明文数字)序列，划分成长度为 n 的组(可看成长度为 n 的矢量)，每组分别在密钥的控制下变换成等长的输出数字(简称密文数字)序列。

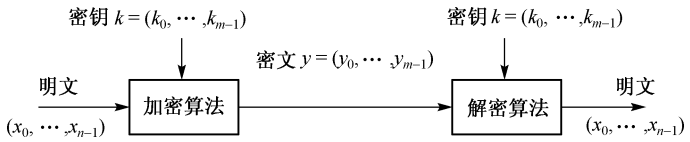


图 3.1 分组密码模型

首先，分组加密算法本质上实现了 n 位明文组和 n 位密文组的一一映射，当 n 较小时，等价于代替变换。例如 $n=3$，表 3.1 给出了一个 3 位分组密码。这是分组密码的最一般形式，能用来定义明密文之间的任意可逆变换。但这种方法有着实际上的困难，对于 $n=3$ 这样较小的分组，类似于古典密码，因为明密文的空间有限，用统计分析方法攻击它是轻而易举的，它的脆弱性来源于分组太少。如果 n 充分大，并且有一个明密文之间的任意可逆变换，那么明文的统计特征将被掩盖。但从实际运行的角度来看，使用大规模分组的任意可逆代替密码是不可行的，一般地，对于 n 位的一般代替分组密码，不同变换的总数为 $(2^n)!$，也就是密钥个数为 $(2^n)!$，表示任一特定代替所需的二进制数字的位数为：

$$\log(2^n!)\approx(n-1.44)2^n=\mathrm{O}(n2^n)(\mathrm{bit})$$

即所需密钥长度达 $n2^n$ 位。

$n=4$ 时，$4\times2^4=64$，需要 64 位的密钥来表示一个任意代替。

$n=64$ 时，表示一个任意代替的密钥的大小将是 $64\times2^{64}=2^{70}\approx10^{21}$ 位。

通过以上分析，可以看出，设计分组密码时，分组长度 n 要足够大，不是为了满足密钥的空间，主要是为了防止明文的穷举攻击，对抗明文的统计分析。此外，任意代替的密钥表的传递和使用也是一个实际的困难。我们将要介绍的数据加密标准 DES 的密钥长度仅为 56 位，高级数据加密标准 AES 的密钥长度为 128/196/256 位。所以，分组密码的设计问题在于找到一种算法，能在密钥的控制之下，从一个足够大而且足够好的代替子集中，简单而迅速地选取出一个代替。

表 3.1 $n=3$ 时的一个 3 位分组密码

(a) 加密表

明文	密文
000	111
001	010
010	110
011	000
100	001
101	100
110	101
111	011

(b) 解密表

密文	明文
111	000
010	001
110	010
000	011
001	100
100	101
101	110
011	111

综上所述，我们给出分组密码的三个安全设计原则：

(1) 分组长度足够大。

(2) 密钥量足够大，能抵抗密钥穷举攻击，但又不能过长，以利于密钥管理。

(3) 由密钥确定代替的算法要足够复杂，能抵抗各种已知攻击。

3.2.2　对称分组密码的两个基本设计方法

在Shannon称为理想密码的密码系统中，密文的所有统计特性都与所使用的密钥独立，他关注的是如何挫败基于统计方法的密码分析，前面所讲的任意代替密码就是这样的，但它是不可能获得实际应用的。抛开求助于理想密码系统，Shannon建议了两种抗统计分析的方法：扩散(Diffusion)和混淆(Confusion)。扩散就是让密文没有统计特征，也就是让明文的统计特征扩散消失到密文的长程统计特性中，以挫败推测出密钥的尝试。做到这一点的方法是让小扰动的影响波及全局，明文的一位影响密文的多位，反过来说，也就是让每个密文比特被多位明文比特影响。扩散增加密文与明文之间关系的复杂性，混淆则增加密钥与密文之间关系的复杂性，同样是为了挫败推测出密钥的尝试。这可以使用一些复杂的代替算法来实现，简单的线性代替函数几乎增加不了混淆。在二进制密码中，对明文进行置换后，再用某个函数作用，重复多次，就可以取得好的扩散效果。

1949 年 Shannon 提出了代替置换网络[Substitution-Permutation(S-P) Networks]的思想，这是构成现代分组密码的基础。S-P 网络基于密码学的两个基本操作，代替被称为 S 盒(S-box)，置换被称为 P 盒(P-box)，它们分别提供了消息的混淆与扩散。

3.3　数据加密标准 DES

1973年，美国国家标准局(NBS)，即现在的国家标准与技术研究所(NIST)，公开征集标准密码算法。IBM公司在 20 世纪 70 年代初开发出的 Lucifer 算法的基础上，改进并提交了建议的 DES 算法。1977 年 1 月 15 日，美国国家标准局将其批准为联邦标准，并设计推出 DES 芯片。DES 开始在商业领域广泛应用。1981 年美国 ANSI 批准 DES 作为私营部门的标准，称为 DEA。1983 年，国际标准化组织 ISO 也将 DES 作为数据加密标准，称为 DEA-1。原先规定使用期为10年，由于新的加密标准没有及时出台，DES实际上活跃了20多年，在保密通信中扮演着十分重要的角色。

3.3.1　DES 的产生与应用

3.3.1.1　DES 的产生

1973 年 5 月 15 日，NBS 开始公开征集标准加密算法，并公布了它的设计要求：

(1) 算法必须提供高度的安全性。

(2) 算法必须有详细的说明，并易于理解。

(3) 算法的安全性取决于密钥，不依赖于算法。

(4) 算法适用于所有用户。

(5) 算法适用于不同应用场合。

(6) 算法必须高效、经济。

(7) 算法必须能被证实有效。

(8) 算法必须是可出口的。

DES 产生过程中的一些标志性事件有：

- 1974 年 8 月 27 日，NBS 开始第二次征集，IBM 提交了算法 LUCIFER，该算法由 IBM 的工程师在 1971 年至 1972 年研制。
- 1975 年 3 月 17 日，NBS 公开了 IBM 提交的算法的全部细节。
- 1976 年，NBS 指派了两个小组对算法进行评价。
- 1976 年 11 月 23 日，DES 被采纳为联邦标准，批准用于非军事场合的各种政府机构。
- 1977 年 1 月 15 日，“数据加密标准”FIPS PUB 46 发布，同年 7 月 15 日开始生效。
- 该标准规定每 5 年审查一次，计划 5 年后采用新标准。
- 最近的一次评估是在 1994 年 1 月，已决定 1998 年 12 月以后，DES 将不再作为联邦加密标准。

3.3.1.2 DES 的应用

DES 应用过程中的一些标志性事件有：

- FIPS PUB 81 (DES 的工作方式)在 1980 年公布。
- FIPS PUB 74(实现和使用 DES 的指南)于 1981 年公布。
- FIPS PUB 112 规定了 DES 用做口令加密的标准。
- FIPS PUB 113 规定了 DES 如何用做计算机数据鉴别。
- 1979 年，美国银行协会批准使用 DES。
- 1980 年，美国国家标准学会(ANSI)赞同 DES 作为私人使用的标准，称为 DEA(ANSI X.392)。
- 1983 年，国际化标准组织 ISO 赞同 DES 作为国际标准，称为 DEA-1。

3.3.2 Feistel 密码结构

前面提到分组加密算法实现了 n 位明密文分组的一一映射，当 n 较小时，等价于代替变换，当 n 较大时，比如 $n=64$，无法表达这样的任意变换。Feistel 结构很好地解决了二者之间的矛盾，其基本思想是用简单算法的乘积来近似表达大尺寸的代替变换。

Feistel 网络是由 Horst Feistel 在设计 Lucifer 分组密码时发明的，并因 DES 的使用而流行。许多分组密码采用了 Feistel 网络，如 Blowfish、RC5 等。

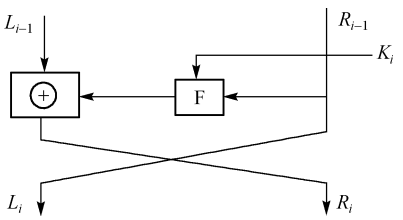


图 3.2 Feistel 结构的一轮

对一个分组长度为 n(偶数)比特的 l 轮 Feistel 网络，它的加密过程如下。给定明文 P，将 P 分成长度相等的左右两半，并分别记为 L_0 和 R_0，从而 $P=L_0R_0$。进行 l 轮完全类似的迭代运算后，再将左边和右边长度相等的两半合并产生密文分组。图3.2给出了Feistel结构的一轮，其加解密过程如下：

- **加密**：$L_i=R_{i-1}$，$R_i=L_{i-1}\oplus F(R_{i-1},K_i)$
- **解密**：$R_{i-1}=L_i$，$L_{i-1}=R_i\oplus F(R_{i-1},K_i)=R_i\oplus$

$F(L_i, K_i)$

其中⊕表示两个比特串的“异或”（XOR），F 是轮函数，K_i 是由种子密钥 K 生成的子密钥。Feistel 分组加密算法具有以下特点：

- **分组长度**：分组长度越大安全性越高，但加解密速度会下降，64 bit 是当时计算条件下较合理的长度。
- **密钥位数**：密钥位数越大安全性越高，但加解密速度下降，现在通常使用的密钥长度是 128 bit。
- **循环次数**：多轮加密可以提高安全性，DES 轮函数的迭代次数是 16 次。
- **子密钥产生算法**：算法越复杂，就越增加密码分析的难度。
- **轮函数**：轮函数越复杂，就越增加密码分析的难度。

此外，分组加密算法还有两个设计上的考虑：

- 能够快速软件实现，包括加密和解密算法。
- 易于分析，便于掌握算法的保密强度以及扩展办法。

对于 DES，还有一些软/硬件实现上的原则。软件实现的要求是使用子块和简单的运算。密码运算在子块上进行，要求子块的长度能自然地适应软件编程，如 8, 16, 32 bit 等。应尽量避免按比特置换，在子块上所进行的密码运算尽量采用易于软件实现的运算。最好是用标准处理器所具有的一些基本指令，如加法、乘法、移位等。硬件实现的要求是加密和解密的相似性，即加密和解密过程的不同应仅仅在密钥的使用方式上，以便采用同样的器件来实现加密和解密，节省费用和体积。尽量采用标准的组件结构，适于在超大规模集成电路中实现。

3.3.3　对 DES 的描述

DES 是一种对二元数据进行加密的算法，数据分组长度为 64 位，密文分组长度也是 64 位。使用的密钥长度为 64 位，其中有效密钥长度为 56 位，有 8 位用于奇偶校验。DES 的解密过程和加密相似，但子密钥的使用顺序正好相反。DES 的整个体制是公开的，系统的安全性完全取决于密钥的保密。

DES 算法的过程示于图3.3中。在一个初始置换 IP 后，明文组被分成右半部分和左半部分，每部分 32 位，以 L_0 和 R_0 表示。然后是16 轮迭代的乘积变换，称为函数 f，将数据和密钥结合起来。16 轮之后，左右两部分再连接起来，经过一个初始逆置换 IP^{-1}，算法结束。

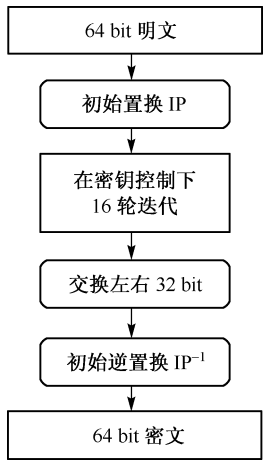


图 3.3　DES 算法的过程

初始置换与初始逆置换在密码意义上作用不大，它们的作用在于打乱原来输入 x 的 ASCII 码字划分关系，并将原来明文的校验位 $x_8, x_{16}, \cdots, x_{64}$ 变成置换输出的一个字节。初始置换 IP 和初始逆置换 IP^{-1} 如表 3.2。

迭代变换是 DES 算法的核心部分，如图3.4 所示。在每轮的开始将输入的 64 bit 数据分成左、右长度相等的两半，将右半部分原封不动地作为本轮输出的 64 bit 数据的左半部分，对右半部分进行一系列的变换，即用轮函数作用于右半部分，然后将所得结果（32 bit

数据)与输入数据的左半部分进行逐位模 2 加，将所得数据作为本轮输出的 64 bit 数据的右半部分。令 i 表示迭代次数，⊕表示逐位模 2 求和，f 为加密轮函数。DES 的加密和解密过程表示如下。加密过程为：

$$L_0R_0 \leftarrow \mathrm{IP}(<64\ \text{bit 输入码}>)$$
$$L_i \leftarrow R_{i-1} \qquad i=1,2,\cdots,16$$
$$R_i \leftarrow L_{i-1} \oplus f(R_{i-1},k_i) \qquad i=1,2,\cdots,16$$
$$<64\ \text{bit 密文}> \leftarrow \mathrm{IP}^{-1}(R_{16}L_{16})$$

表 3.2 初始置换 IP 和初始逆置换 IP^{-1} 表

初始置换 IP								初始逆置换 IP^{-1}							
58	50	42	34	26	18	10	2	40	8	48	16	56	24	64	32
60	52	44	36	28	20	12	4	39	7	47	15	55	23	63	31
62	54	46	38	30	22	14	6	38	6	46	14	54	22	62	30
64	56	48	40	32	24	16	8	37	5	45	13	53	21	61	29
57	49	41	33	25	17	9	1	36	4	44	12	52	20	60	28
59	51	43	35	27	19	11	3	35	3	43	11	51	19	59	27
61	53	45	37	29	21	13	5	34	2	42	10	50	18	58	26
63	55	47	39	31	23	15	7	33	1	41	9	49	17	57	25

解密过程为：

$$R_{16}L_{16} \leftarrow \mathrm{IP}(<64\ \text{bit 密文}>)$$
$$R_{i-1} \leftarrow L_i \qquad i=16,15,\cdots,1$$
$$L_{i-1} \leftarrow R_i \oplus f(R_{i-1},k_i) \qquad i=16,15,\cdots,1$$
$$<64\ \text{bit 明文}> \leftarrow \mathrm{IP}^{-1}(R_0L_0)$$

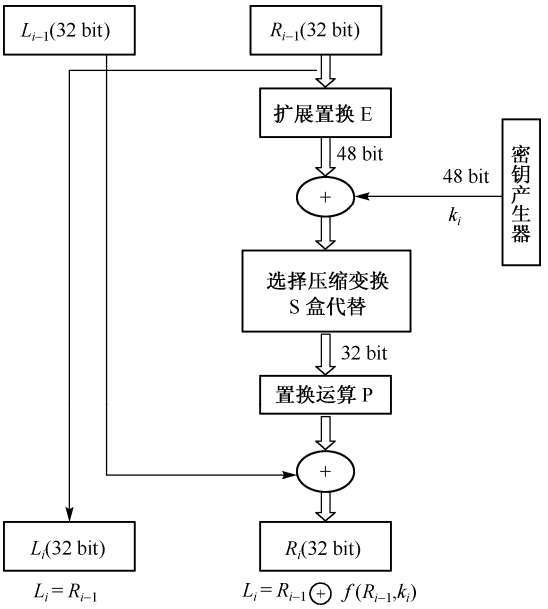


图 3.4 DES 的一轮迭代

从图3.4可以看出，轮函数由选择扩展运算E、与子密钥的异或运算、选择压缩变换 S 和置换运算 P 组成。

扩展置换运算 E 将输入 32 bit 数据扩展为 48 bit 的输出数据，变换表如表3.3所示，它的作用有三个：

(1)产生了与密钥同长度的数据进行异或运算。

(2)它产生了更长的结果，使得在代替运算时能进行压缩。

(3)输入的一位将影响两个替换，所以输出对输入的依赖性将传播得更快。使得明文或密钥的一点小的变动应该使密文发生一个大的变化，这叫**雪崩效应**(Avalanche effect)。

表 3.3　DES 的选择扩展运算 E

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

选择压缩变换将输入的 48 bit 数据自左至右分成 8 组，每组为 6 bit。然后输入 8 个 S 盒，每个 S 盒为一非线性代换，有 4 bit 输出，如图3.5所示。

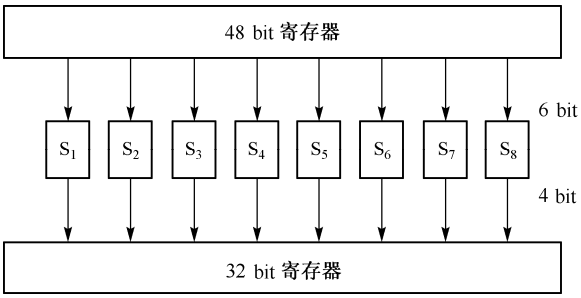


图 3.5　选择压缩变换 S

8 个 S 盒的选择函数关系分别如表 3.4 所示。对每个 S 盒，6 bit 输入中的第 1 和第 6 比特组成的二进制数用于确定 S 盒的行，中间 4 位二进制数用来确定 S 盒的列，对应位置十进制数的 4 位二进制表示作为输出。假定输入是 110011，经过 S_6 盒的输出为 1110。具体变换过程如下：

$$\frac{b_1b_2b_3b_4b_5b_6}{110011} \Rightarrow \frac{\text{行}: b_1b_6 = 11_2 = 3}{\text{列}: b_2b_3b_4b_5 = 1001_2 = 9} \Rightarrow \frac{S_6\text{盒子3行9列}}{\text{值}: 14 = 1110}$$

S盒是许多密码算法的唯一非线性部件，因此，它的密码强度决定了整个算法的安全强度，提供了密码算法所必须的混淆作用。

置换 P 如表 3.5 所示，P 置换的目的是提供雪崩效应，即明文或密钥的一点小的变动都引起密文的较大变化。

表 3.4 S 盒

	行/列	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S_1	0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
	1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
	2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
	3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
S_2	0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
	1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
	2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
	3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
S_3	0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
	1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
	2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
	3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
S_4	0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
	1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
	2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
	3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14
S_5	0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
	1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
	2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
	3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
S_6	0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
	1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
	2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
	3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
S_7	0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
	1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
	2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
	3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
S_8	0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
	1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
	2	1	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
	3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

表 3.5 P 盒

P			
16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

子密钥的产生过程如图3.6所示，给定 64 bit 的密钥 K，用置换选择 1(PC-1)作用，去掉了输入的第 8，16，24，31，40，48，56，64 位，这 8 bit 是奇偶校验位，并重排实际 56 bit 的密钥。将得到的 56 bit 数据分成左、右等长的 28 bit，分别记为 C_0 和 D_0。对1≤ i≤ 16，计算 $C_i = \mathrm{LS}_i(C_{i-1})$ 和 $D_i = \mathrm{LS}_i(D_{i-1})$。LS 表示循环左移。每轮循环左移的位数为：

轮　数	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
左移的位数	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

然后，将每轮 56 bit 数据 C_i D_i 用置换选择 2(PC-2)作用，去掉了第 9，18，22，25，35，38，43，54 位，同时重排了剩下的 48 bit，输出作为 K_i。

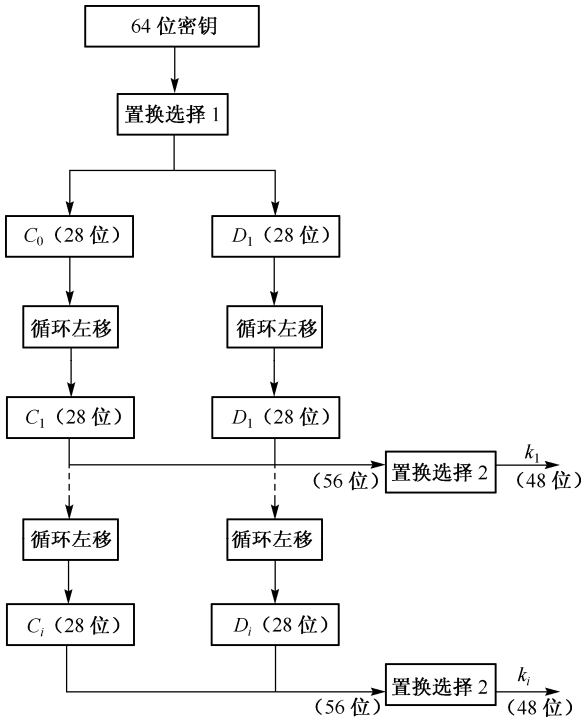


图 3.6　子密钥的产生过程

置换选择 1(PC-1)和置换选择 2(PC-2)如表3.6所示。PC-1 把 64 位 DES 密钥的第 8 的倍数位去掉，缩减为 56 位。PC-2 再从 56 位中选出 48 位。

表 3.6　子密钥产生的置换选择 PC-1 和置换选择 PC-2

PC-1							PC-2					
57	49	41	33	25	17	9	14	17	11	24	1	5
1	58	50	42	34	26	18	3	28	15	6	21	10
10	2	59	51	43	35	27	23	19	12	4	26	8
19	11	3	60	52	44	36	16	7	27	20	13	2
63	55	47	39	31	23	15	41	52	31	37	47	55
7	62	54	46	38	30	22	30	40	51	45	33	48
14	6	61	53	45	37	29	44	49	39	56	34	53
21	13	5	28	20	12	4	46	42	50	36	29	32

DES 的解密算法和加密算法完全相同，只是各子密钥的使用顺序相反，即为 $k_{16}, k_{15}, k_{14}, \cdots, k_2$，$k_1$。算法也是循环右移产生每一圈的子密钥，每次右移的位数为：

轮　数	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
右移的位数	0	1	2	2	2	2	2	2	1	2	2	2	2	2	2	2

3.3.4 对 DES 的讨论

3.3.4.1 弱密钥与半弱密钥

初始密钥被分成两部分，每部分都单独做移位。如果每一部分的每一位都是 0 或都是 1，则每一圈的子密钥都相同。这样的密钥被称为**弱密钥**。基于弱密钥的加解密过程是相同的，加密两次就恢复成明文。DES 存在 4 个弱密钥，以十六进制形式表示，参见表 3.7。

表 3.7 DES 弱密钥

弱密钥值(带奇偶校验)				实际密钥	
0101	0101	0101	0101	0000000	0000000
1F1F	1F1F	0E0E	0E0E	0000000	FFFFFFF
E0E0	E0E0	F1F1	F1F1	FFFFFFF	0000000
FEFE	FEFE	FEFE	FEFE	FFFFFFF	FFFFFFF

有些成对的密钥会将明文加密成相同的密文，即一对密钥中的一个能用来解密由另一个密钥加密的消息，这种密钥称作**半弱密钥**。这些密钥不是产生16 个不同的子密钥，而是产生两种不同的子密钥，每一种出现 8 次。至少有 12 个半弱密钥。其十六进制形式参见表 3.8。

表 3.8 DES 半弱密钥(用十六进制表示)

01FE	01FE	01FE	01FE	与	FE01	FE01	FE01	FE01
1FE0	1FE0	0EF1	0EF1	与	E01F	E01F	F10E	F10E
01E0	01E0	01F1	01F1	与	E001	E001	F101	F101
1FFE	1FFE	0EFE	0EEE	与	FE1F	FE1F	FE0E	FE0E
011F	011F	010E	010E	与	1F01	1F01	0E01	0E01
E0FE	E0FE	F1FE	F1FE	与	FEE0	FEE0	FEF1	FEF1

还有些密钥只产生四种子密钥，每种出现四次。这种密钥称为**半半弱密钥**，共 48 个。弱密钥、半弱密钥、半半弱密钥加起来有 $4+12+48=64$ 个，而可能的密钥数是 2^{56} 个。随机选择密钥，落在这 64 个中的概率很小，而且可以在产生时进行检查，以避免这些密钥。

3.3.4.2 互补密钥

将密钥的 0 换成 1，1 换成 0，就得到该密钥的**补密钥**。

通过对一位的二进制变量 A 和 B 运算规律的观察，对于任何等长的 A 和 B，我们有：$(A \oplus B)' = A' \oplus B,\ A \oplus B = A' \oplus B'$ 。

对于 DES 的一轮来说，如果明文和密钥取补，第一个 XOR 的输入也取补，输出和没有取补时一样，进一步看到对于第二个XOR的输入只有一个取补了，该轮输出是没有取补的输入产生的输出的补。

总体来说，如果用原密钥加密一组明文，则用补密钥可以将明文的补码加密成密文的补码。

设 X' 表示 X 的补码，则

$$\mathrm{E}_k(M) = C$$
$$\mathrm{E}_{k'}(M') = C'$$

这一结论可以用于对 DES 的选择明文攻击，在一个选择明文攻击中，如果选择明文 X，攻击者可以得到 $Y_1 = \mathrm{E}_k[X]$ and $Y_2 = \mathrm{E}_k[X']$，那么穷举式攻击只需要进行 2^{55} 次加密，而不是 2^{56} 次，使测试的密钥减少一半。

注意到 $(Y_2)' = \mathrm{E}_{k'}[X]$，现在选取一个测试密钥 T，计算 $\mathrm{E}_T[X]$，那么有三种可能：

(1) 如果结果是 Y_1，T 是正确的密钥。

(2) 如果结果是 $(Y_2)'$，T' 是正确的密钥。

(3) 如果都不是，就通过一次加密否定了两个基本密钥。

3.3.4.3　DES 的破译

根据攻击者所掌握的信息，可将对对称分组密码的攻击分为以下几类：唯密文攻击、已知明文攻击、选择明文攻击。

攻击的复杂度可以从两个方面衡量，一个是数据复杂度，即实施该攻击所需输入的数据量，一个是处理复杂度，即处理这些数据所需要的计算量。对 DES 最可靠的攻击办法是强力攻击，此外还有一些比强力攻击更有效的攻击方法：差分密码分析、线性密码分析、插值攻击方法、密钥相关攻击方法等。

强力攻击可以用于任何分组密码，攻击复杂度严格依赖于分组长度和密钥长度。具体而言，强力攻击又可以分为以下几种做法：强力密钥搜索攻击、字典攻击、查表攻击和时间-存储攻击。设 k 是密钥的长度，在唯密文攻击下，攻击者依次试用密钥空间中所有的 2^k 个密钥解密密文，直到得到一个有意义的明文。其复杂度平均为 2^{k-1}。字典攻击是攻击者搜集明密文对，把它们编成字典，当截获密文时，查找密文是否在字典中存在，如果 n 是分组长度，攻击者需要 2^n 个明密文对才能在不知道密钥的情况下解密任何消息。查表法是一种选择明文攻击，对给定明文用所有 2^k 个密钥预先算出密文，这样，对于给定的密文就可以从存储表中找出相应的密钥。时间-存储权衡攻击是一种选择明文攻击，由穷尽密钥搜索和查表攻击两种方法混合而成，比穷尽密钥搜索的时间复杂度小，比查表攻击的空间复杂度小。

1990 年，以色列密码学家 Eli Biham 和 Adi Shamir 提出了差分密码分析法，可对 DES 进行选择明文攻击，通过分析明文对的差值对密文对的差值的影响来恢复某些密钥比特。线性密码分析本质上是一种已知明文攻击方法，通过寻找一个给定密码算法的有效的线性近似表达式来破译密码系统。差分密码分析法需要使用2^{47} 对明密文进行选择明文攻击，线性密码分析法需要使用2^{47} 对明密文进行已知明文攻击。由于差分密码分析和线性密码分析所需要的选择(已知)明文数量太大，强力攻击依然是目前实用的攻击，如何改进差分密码分析和线性密码分析的复杂度仍是理论研究的热点。

3.3.4.4　密钥长度的争论

关于 DES 算法的另一个最有争议的问题就是担心实际 56 bit 的密钥长度不足以抵御穷举式攻击，因为密钥量只有 2^{56} 个。

早在 1977 年，Diffie 和 Hellman 已建议制造一个每微秒能测试 100 万个密钥的 VLSI 芯片。每微秒测试 100 万个密钥的机器大约需要一天就可以搜索整个密钥空间。当时，他们估计制造这样的机器大约需要 2000 万美元。

Hellman 提出通过空间和时间的折中，可以加速密钥的寻找过程。他建议计算并存储 2^{56} 个用每种可能密钥加密一段固定明文的结果。估计机器造价 500 万美元。

在 CRYPTO'93 上，Session 和 Wiener 给出了一个非常详细的密钥搜索机器的设计方案，

这个机器基于并行运算的密钥搜索芯片，所以 16 次加密能同时完成。此芯片每秒能测试 5000 万个密钥，用 5760 个芯片组成的系统需要花费 10 万美元，它平均用 1.5 天左右就可找到 DES 密钥。

1997 年1 月28 日，美国的 RSA 数据安全公司在 RSA 安全年会上公布了一项“秘密密钥挑战”竞赛，其中包括悬赏1万美元破译密钥长度为 56 bit 的 DES。美国科罗拉多州的程序员 Verser 从 1997 年 2 月 18 日起，用了 96 天时间，在 Internet 上数万名志愿者的协同工作下，成功地找到了 DES 的密钥，赢得了悬赏的 1 万美元。

1998 年 7 月电子前沿基金会(Electronic Frontier Foundation)使用一台 25 万美元的计算机在 56 小时内破译了 56 bit 密钥的 DES。

1999 年1月 RSA 数据安全会议期间，电子前沿基金会用 22 小时 15 分钟就宣告破解了一个 DES 的密钥。

3.3.4.5 DES 的轮数

对于DES，56 bit 密钥决定了密钥空间是固定的，为什么说轮函数的循环次数越多则安全性越高？为什么 DES 是 16 轮，而不是 32 轮？

一般来说，循环次数越多进行密码分析的难度就越大，循环次数的选择准则是要使已知密码分析的工作量大于简单的穷举式密钥搜索的工作量。对于 16 轮循环的 DES 来说，差分密码分析的运算次数为 $2^{55.1}$，而穷举式搜索要求 2^{55}，前者比后者效率稍低，如果DES 有 15 次循环，那么差分密码分析比穷举式搜索的工作量要小。对 DES 进行差分密码分析，需要 2^{47} 个选择明文，如果仅有已知明文，就需要多一些的明密文对，使运算量增大到 $2^{55.1}$。

3.3.4.6 S 盒的设计原理未知

S 盒(S-Box)是许多密码算法的唯一非线性部件，因此，它的密码强度决定了整个算法的安全强度，提供了密码算法所必须的混乱作用，如何全面准确地度量 S 盒的密码强度和设计有效的 S 盒是分组密码设计和分析中的难题。S盒的设计细节，NSA 和 IBM 都未公开过。Hellman 在1976年指出，在 DES 中仔细选择一些 S 盒，可以使安全性降低。他们以自己设计的 S 盒代替 DES 原来的 S 盒，证明可以做到这一点，且在一定程度上可以隐蔽 S 盒构造上的弱点。在差分分析公开后，1992 年 IBM 公布了S 盒和 P 盒设计准则。从本质上说，希望 S 盒输入向量的任何变动在输出方都产生看似随机的变动。这两种变动之间的关系是非线性的，并难以用线性函数近似。

3.4 三重 DES

自 DES 公布之后，人们了解 DES 的弱点越来越深入，出现了一些对 DES 的改进和代替算法。基本的方式有两种，一种是对 DES 进行复合，强化它的抗攻击能力；另一种是开辟新的方法，即像 DES 那样加解密速度快，又具有抗差分攻击和其他方式攻击的能力。这些算法典型的有三重 DES, IDEA, RC5, RC6, Blowfish, CAST 和 RC2 等。

由于已经证明 DES 不能成为群，于是多重 DES，尤其是三重 DES 还在普遍使用。

3.4.1 双重 DES

最简单的多次 DES 加密形式是用两个密钥进行两次加密，如图 3.7 所示。给定明文 P 和加密密钥 K_1 和 K_2，密文 $C = \mathrm{E}_{K_2}(\mathrm{E}_{K_1}(P))$，解密要求密钥以相反的次序使用，$P = \mathrm{D}_{K_1}(\mathrm{D}_{K_2}(C))$。

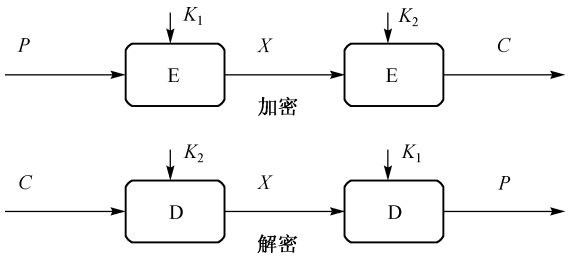


图 3.7　双重 DES

假设对于 DES 和所有 56 bit 密钥，给定任意两个密钥 K_1 和 K_2，都能找到一个密钥 K_3 使得 $\mathrm{E}_{K_2}(\mathrm{E}_{K_1}(P))=\mathrm{E}_{K_3}(P)$。如果这个假设是事实，则 DES 的两重加密或者多重加密都将等价于用一个 56 bit 密钥的一次加密。从直观上看，上面的假设不可能为真。因为 DES 的加密实际上就是完成一个从 64 bit 分组到 64 bit 分组的代替变换，而 64 bit 分组共有 2^{64} 可能的状态，因而可能的代替变换个数为：

$$2^{64}!>10^{34738000000000000000}>10^{10^{20}}$$

另外，DES 的每个密钥确定了一个代替变换，因而总的代替变换个数为 $2^{56}<10^{17}$，直到 1992 年才有人证明了这个结果。

虽然双重DES不同于单次DES，但它可能会受到一种基于观察的中间相遇攻击。根据 $C=\mathrm{E}_{K_2}(\mathrm{E}_{K_1}(P))$，则有 $X=\mathrm{E}_{K_1}(P)=\mathrm{D}_{K_2}(C)$，给定明文密文对$(P, C)$，先用所有$2^{56}$个密钥加密$P$，对结果排序，再用所有$2^{56}$ 个密钥解密C，对结果排序，然后逐个比较两个运算的结果，找出使得 $\mathrm{E}_{K_1}(P)=\mathrm{D}_{K_2}(C)$ 的 K_1 和 K_2。

给定一个明文 P，经双重 DES 加密有 2^{64} 个可能的密文。而双重 DES 所用密钥的长度应是112 位，所以选择密钥有2^{112} 个可能性。于是对一个给定的明文P，把它加密成同样密文的密钥有 $2^{112}/2^{64}=2^{48}$ 种可能。给定两个明密文对，虚警率降为 $2^{48-64}=2^{-16}$。换句话说，对已知两个明文–密文对的中间相遇攻击成功的概率为 $1-2^{-16}$。攻击用的代价也就是加密或解密所用的运算次数，这个数小于等于 4×2^{56}，但需要大量的存储器。

3.4.2　三重 DES

三重DES（Triple DES）使用三（或两）个不同的密钥对数据块进行三次加密，具体有四种模型：

（1）DES-EEE3，使用三个不同密钥，顺序进行三次加密变换。

（2）DES-EDE3，使用三个不同密钥，依次进行加密–解密–加密变换。

（3）DES-EEE2，其中密钥 $K_1=K_3$，顺序进行三次加密变换。

（4）DES-EDE2，其中密钥 $K_1=K_3$，依次进行加密–解密–加密变换。

Tuchman 建议使用双密钥进行加密–解密–加密（EDE）的方案，加密为 $C=\mathrm{E}_{K_1}(\mathrm{D}_{K_2}(\mathrm{E}_{K_1}(P)))$，解密为 $P=\mathrm{D}_{K_1}(\mathrm{E}_{K_2}(\mathrm{D}_{K_1}(C)))$，如图3.8所示，其强度大约和 112 位的密钥强度相当。第二个步骤使用解密并没有密码编码上的考虑，相对于使用加密，它的唯一优点是可以使三重 DES 的用户能够解密原来用 DES 加密的数据。该模式由 IBM 设计，可与常规加密算法兼容，这种替代 DES 的加密较为流行，并且已被采纳用于密钥管理标准 ANSI X9.17 和 ISO 8732。交替使用 K_1 和 K_2 可以抵抗中间相遇攻击。到目前为止，还没有人给出攻击双密钥三重 DES 的有

效方法。对其密钥空间中的密钥进行强力搜索，那么由于空间($2^{112}=5\times10^{33}$)太大，这实际上是不可行的。若用差分攻击的方法，相对于单一 DES 来说复杂性以指数形式增长，要超过 10^{52}。

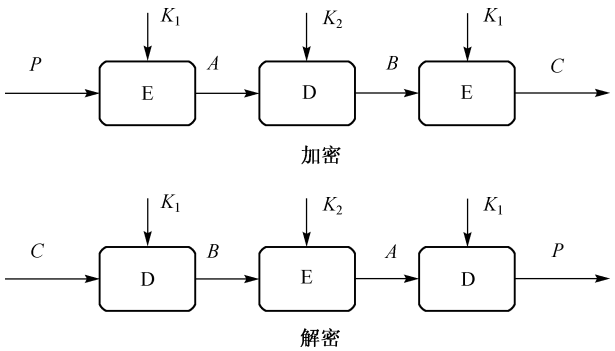


图 3.8 三重 DES

虽然目前还没有针对双密钥三重 DES 的实用攻击方法，但对双密钥三重DES的攻击有一些设想，以这些设想为基础将来可能设计出更成功的攻击技术。许多研究人员感到具有三个密钥的三重 DES 是更好的选择。加、解密过程如下：

$$C=\mathrm{E}_{K_3}(\mathrm{D}_{K_2}(\mathrm{E}_{K_1}(P)))$$
$$P=\mathrm{D}_{K_1}(\mathrm{E}_{K_2}(\mathrm{D}_{K_3}(C)))$$

密钥的有效长度为 168 位，与 DES 的兼容性可以通过令 $K_3=K_2$ 或 $K_1=K_2$ 得到。许多基于 Internet 的应用中用到这种三密钥的 3DES，比如 PGP 和 S/MIME。

3.5 高级数据加密标准 AES

3.5.1 AES 的背景

1997 年 4 月 15 日，美国国家标准和技术研究所(NIST)发起了征集 AES 算法的活动，并成立了专门的 AES 工作组，目的是为了确定一个非保密的、公开披露的、全球免费使用的分组密码算法，用于保护下一世纪政府的敏感信息，并希望成为秘密和公开部门的数据加密标准。1997 年 9 月 12 日，NIST 公布了 AES 算法候选提名的最终要求，AES 的基本要求是：比三重 DES 快而且至少和三重 DES 一样安全，分组长度 128 比特，密钥长度为 128/192/256 比特。1998 年 8 月 20 日，NIST 召开了第一次 AES 候选会议，并公布了 15 个候选算法。1999 年 3 月 22 日举行了第二次 AES 候选会议，会议对 15 个算法在安全性、代价和实现特性等方面进行了评估，从中选出 5 个。入选 AES 的 5 种算法是 MARS、RC6、Serpent、Twofish、Rijndael。2000 年 4 月 13 日，在第三次 AES 候选会议上，对这 5 个候选算法的各种分析结果进行了讨论。2000 年 10 月 2 日，美国商务部部长 Norman Y. Mineta 宣布，经过三年来世界著名密码专家之间的竞争，Rijndael 数据加密算法最终获胜。NIST 指出，之所以最终选择 Rijndael 是因为它是安全性、性能、效率、实现方便性和灵活性的最佳组合。AES 将成为新的公开的联邦信息处理标准(Federal Information Processing Standard, FIPS)——一种用于美国政府组织保护敏感信息的加密算法。美国国家标准技术研究所预测 AES 会被广泛地应用于组织、学院及个人。

3.5.2　AES 的数学基础

先来回顾一下抽象代数的一些基本概念。

3.5.2.1　代数运算

若干个(有限或无限多个)固定事物的全体叫做一个**集合**。组成一个集合的事物叫做这个集合的**元素**(有时简称元)。

一个没有元素的集合叫空集合。

集合 A 和集合 B 的所有共同元素组成的集合叫做 A 和 B 的**交集** $A \cap B$。

由至少属于 A 和 B 之一的一切元素组成的集合叫做 A 和 B 的**并集** $A \cup B$。

令 $A_1, A_2, \cdots, A_n$ 是 n 个集合，由一切从 $A_1, A_2, \cdots, A_n$ 里顺序取出的元素组 $(a_1, a_2, \cdots, a_n)$ $(a_i \in A_i)$ 所做成的集合叫做 $A_1, A_2, \cdots, A_n$ 的积，记为 $A_1 \times A_2 \times \cdots \times A_n$。

假如通过一个法则 ϕ，对于任何一个 $A_1 \times A_2 \times \cdots \times A_n$ 的元素 $(a_1, a_2, \cdots, a_n)$ $(a_i \in A_i)$，都能得到一个唯一的集合 D 的元素 d，那么这个法则 ϕ 叫做集合 $A_1 \times A_2 \times \cdots \times A_n$ 到集合 D 的一个**映射**。元素 d 叫做元素 $(a_1, a_2, \cdots, a_n)$ 在映射 ϕ 之下的**像**；元素 $(a_1, a_2, \cdots, a_n)$ 叫做元素 d 在映射 ϕ 之下的**逆像**。

一个映射常用以下符号来描写

$$\phi: (a_1, a_2, \cdots, a_n) \to \mathrm{d} = \phi(a_1, a_2, \cdots, a_n)$$

一个 $A \times B$ 到 D 的映射叫做一个 $A \times B$ 到 D 的**代数运算**，一个代数运算用“$\circ$”来表示，$(a, b) \to d = \circ(a, b)$，为方便起见，$\circ(a, b)$ 可以写成 $a \circ b$。

假如$\circ$是一个 $A \times A$ 到 A 的代数运算，我们就说集合 A 对于代数运算$\circ$来说是封闭的，也说$\circ$是 A 的二元代数运算。

称一个集合 A 的代数运算$\circ$适合结合律，假如对于 A 的任何三个元素 a，b，c 来说，都有 $(a \circ b) \circ c = a \circ (b \circ c)$。

称一个 $A \times A$ 到 D 的代数运算$\circ$适合交换律，假如对于 A 的任何两个元素 a，b 来说，都有 $a \circ b = b \circ a$。

设 $\odot$，$\oplus$ 是 A 上的两个不同的二元运算，称代数运算 $\odot$ 对 $\oplus$ 适合分配律，假如对于 A 的任何元素 b，a_1, a_2 来说，都有 $b \odot (a_1 \oplus a_2) = (b \odot a_1) \oplus (b \odot a_2)$ 而且 $(a_1 \oplus a_2) \odot b = (a_1 \odot b) \oplus (a_2 \odot b)$。

3.5.2.2　群

一个非空的集合 G 对于一个叫做乘法$\circ$的代数运算来说做成一个**群**，假如

(1) G 对于乘法来说是封闭的。

(2) 结合律成立：$a \circ (b \circ c) = (a \circ b) \circ c$，对于 G 的任意三个元素 a，b，c 都成立。

(3) G 里至少存在一个单位元 e，能让 $e \circ a = a$，$a \circ e = a$，对于 G 的任何元 a 都成立。

(4) 对于 G 的每一个元素 a，在 G 里存在一个逆元 a^{-1} 能让 $a^{-1} \circ a = a \circ a^{-1} = e$。

例如，全体整数的集合 Z 对整数的加法形成一个群，称为整数加群。

一个群叫做**有限群**，假如这个群的元素个数是一个正整数。否则，这个群叫做**无限群**。一个有限群的元素的个数叫做这个群的**阶**。

一个群叫做**交换群**，假如 $a \circ b = b \circ a$，对于 G 里的任何两个元素 a, b 都成立。

若一个群的每一个元素都是 G 的某一个固定元素 a 的乘方，我们把 G 叫做**循环群**，我们也说 G 是由元素 a 所生成的，并且用符号 $G = (a)$ 来表示。a 叫做 G 的一个**生成元**。

例如，整数加群形成一个循环群，它的生成元仅有 1 和–1，为无限循环群。

整数同余类环 Z_n 中的全部元素对同余类加法所形成的群，是一个阶为 n 的循环群。若 a 和 n 互素，则由 a 决定的模同余类$[a]$就是 Z_n 的生成元。事实上，$\forall[b]\in Z_n$，必有一个整数 k 使得 $a\cdot k\equiv b \pmod n$。这样，我们有：

$$k[a]=[a]+[a]+\cdots+[a]=[k\cdot a]=[b]\text{（有 } k \text{ 个}[a]\text{）}$$

整数同余类环 Z_n 的乘法可逆元的全体组成的集合对同余类乘法形成一个群 Z_n^*，这个群是交换群，一般不是循环群，以 Z_{12}^*为例，$Z_{12}^*=\{[1],[5],[7],[11]\}$，它不是循环群。仅当 n 为素数的时候，Z_n^*为循环群。

3.5.2.3 环

一个集合 R 叫做一个**环**，假如：

(1) R 对于一个叫做加法“+”的代数运算成为一个交换群。

(2) R 对于一个叫做乘法“∘”的代数运算来说是封闭的。

(3) 这个乘法适合结合律 $a\circ(b\circ c)=(a\circ b)\circ c$，对于属于 R 的任意元素 a, b, c 都成立。

(4) ∘对+的分配律成立，$a\circ(b+c)=a\circ b+a\circ c$，$(b+c)\circ a=b\circ a+c\circ a$。

例如，全体整数构成的集合对于普通加法和乘法来说成为一个环。全体有理数构成的集合对于普通加法和乘法来说成为一个环。

环 R 被称为含有**单位元的环**，是指 R 内含有乘法单位元“1”，使得$\forall a\in R$，有 $a\circ 1=1\circ a=a$。

一个环叫做一个**交换环**，假如 $a\circ b=b\circ a$，对于属于 R 的任意两个元素 a，b 都成立。

例如，剩余类环 Z_n 为整数模 n 剩余类的集合 $Z_n=\{[0],[1],\cdots,[n-1]\}$，它对剩余类的加法和乘法构成一个含有单位元“[1]”的交换环。取大于 1 的正整数 n，则 n 的一切整数倍形成的集合 nZ 对数的加法和乘法形成了一个不含单位元“1”的交换环。

整环：含有乘法单位元“1”而无零因子的交换环称为整环。

若在一个环里，$a\neq 0$，$b\neq 0$ 但 $a\circ b=0$，就说 a 是这个环的一个**左零因子**，b 是这个环的一个**右零因子**。

任何一个整环都至少含有两个元素。恰含有两个元素的整环是存在的，例如 $F_2=\{0, 1\}$ 它对模 2 的加法乘法运算形成一个整环，事实上，它为二元域。

3.5.2.4 有限域 GF(p)

一个环被称为**除环**(或**斜域**)，是指该环的非零元全体对“∘”形成一个群。

一个可交换的除环称为**域**。

例如，F_p 为整数模 p 的剩余类环，p 为素数，可以验证它为域。因为 F_p 中的元素有限，称它为**有限域**；又因为 p 为素数，又称为**素域**。

域首先必是整环；反之则不然。

一个元素个数有限的域称为**有限域**，或者**伽罗瓦域**(Galois field)。

有限域中运算满足交换律、结合律和乘法对加法的分配律。加法的单位元是 0，乘法的单位元是 1，每个非零元素都有一个唯一的乘法逆元。

通俗地讲，域是一个包含两种运算的集合，其中一种运算叫加法，另一种运算叫乘法。如果把加法的逆运算叫减法，把乘法的逆运算叫除法时，域是一个在其元素之间可以自由地

进行加、减、乘、除运算且保持运算结果仍在其内的一个集合。数域只是域中的一种，特点是元素都是数。典型的数域有：有理数域、实数域和复数域。

密码学中用到很多有限域中的运算，因为可以保持数在有限的范围内，且不会有取整的误差。最常用的两个域是 GF(p)和 GF(2^n)。

有限域中元素的个数为一个素数 p，或者一个素数的幂 p^n，记为 GF(p)或 GF(p^n)，其中 p 为素数。GF(p)是整数集合{0, 1, …, p–1}，其运算为模素数 p 的运算。

3.5.2.5　多项式运算和有限域 GF(2^n)

对于一元多项式 $f(x)$，多项式运算可分为三种：使用代数基本规则的普通多项式运算；系数在有限域 GF(p)中，即系数运算是模 p 运算的多项式运算；系数在有限域 GF(p)中，且多项式被定义为模一个 n 次多项式 $m(x)$的多项式运算。

一个一元 n(n≥0)次多项式的表达形式如下：

$$f(x)=a_nx^n+a_{n-1}x^{n-1}+\cdots+a_1x^1+a_0$$

其中，a_i 是某个指定的数集 S 中的元素，该数集叫系数集，且 $a_n\neq 0$，我们称 $f(x)$是定义在系数集 S 上的多项式。

现在考虑这样的多项式，它的系数是域 F 中的元素，这样的多项式集合关于多项式乘法和多项式加法是一个环，称为多项式环，记为 F [x]。对域上的多项式进行多项式运算时，除法是可能的，但并不表示可以整除。一般来说，会产生一个商式和一个余式

$$f(x)=q(x)g(x)+r(x)$$

与整数运算相似，我们可以把余式 $r(x)$写为 $f(x)$ mod $g(x)$，如果余式 $r(x)=0$，可以说 $g(x)$整除 $f(x)$，记为 $g(x)|f(x)$，等价地可以说 $g(x)$是 $f(x)$的一个因式或除式。

域 F 上的多项式 f (x)被称为不可约的，当且仅当 f (x)不能表示为两个基本多项式(两个多项式都在 F 上，次数都低于 $f(x)$)的积。与整数相似，一个不可约多项式称为**素多项式**。

设集合 S 由域 Z_p 上次数小于 n 的所有多项式组成，每一个多项式具有如下形式：$f(x)=a_{n-1}x^{n-1}+a_{n-2}x^{n-2}+\cdots+a_1x^1+a_0$，其中 a_i 在集合{0, 1, 2, …, p–1}上取值。S 中共有 p^n 个不同的多项式。如果定义了合适的运算，那么每一个这样的集合都是一个有限域。定义包含以下几条：

(1) 该运算遵循基本代数规则中的普通多项式运算规则。

(2) 系数运算以 p 为模。

(3) 如果乘法运算的结果是次数大于n–1 的多项式，必将其除以某个次数为 n 的不可约多项式 $m(x)$并取余式。

设 F 为有限域，$m(x)=x^n+m_1x^{n-1}+m_2x^{n-2}+\cdots+m_n$，$m_i\in F$，为 F 上的 n 次不可约多项式，以 $m(x)$为模的同余类环 E=F[x]/<m(x)>可视为 F 上的一个有限域。当多项式的系数为 Z_q 中的元素，q 为一素数，这个域称为 GF(q^n)。有限域 GF(2^3)的元素为：$0, 1, x, x+1, x^2, x^2+1, x^2+x, x^2+x+1$。

实际上所有的加密运算(包括对称密钥和公开密钥算法)都涉及整数集上的算术运算。如果某种算法使用的运算之一是除法，那么就必须定义域上的运算。我们希望整数集中的数与给定的二进制位所能表达的信息一一对应，n 位二进制编码所代表的数的个数为 2^n。

特别地，GF(2^n)中的多项式 $f(x)=a_{n-1}x^{n-1}+a_{n-2}x^{n-2}+\cdots+a_1x^1+a_0$ 可以由它的 n 个二进制系数($a_{n-1}\ a_{n-2}\cdots a_1\ a_0$)唯一地表示。因此，GF($2^n$)中的每一个多项式都可以表示成一个 n 位的二进制数。

在硬件上利用线性反馈移位寄存器可以快速地实现 GF(2^n)上的运算。 GF(2^n)上的运算通常比 GF(p)上的运算快。

3.5.2.6 GF(2^8)的构造

AES 使用有限域 GF(2^8)上的运算，选定 GF(2)上的一个 8 次不可约多项式 $m(x)=x^8+x^4+x^3+x+1$，以 $m(x)$为模的同余类 $F_2[x]/<m(x)>$可以形成一个有限域，GF(2^8)中的多项式 $f(x)=a_7x^7+a_6x^6+\cdots+a_1x^1+a_0$可用一个字节表示为$(a_7, a_6, a_5, a_4, a_3, a_2, a_1, a_0)$，$a_i\in F_2$，于是 GF($2^8$)中两元素的和可以通过相应的两个字节的异或完成：

$$(01010111)\oplus(10000011)=(11010100)$$

对应于

$$(x^6+x^4+x^2+x+1)+(x^7+x+1)=x^7+x^6+x^4+x^2$$

而两个元素的乘法，需要进行字节对应的多项式的乘法，再模 $m(x)=x^8+x^4+x^3+x+1$，例如，两字节(01010111)，(10000011)的乘⊙，相当于$(x^6+x^4+x^2+x+1)\times(x^7+x+1)=x^{13}+x^{11}+x^9+x^8+x^6+x^5+x^4+x^3+1 \pmod{m(x)}=x^7+x^6+1$ (=(11000001))。

另一方面，某字节对应于$f(x)\in F_2[x]/<m(x)>$，求这个字节的逆，就是求$f(x)^{-1}$，可用 Euclid 算法，算出 $u(x)$，$v(x)\in F_2[x]$，使

$$u(x)f(x)+v(x)m(x)=1$$

于是

$$f(x)^{-1}=u(x) \pmod{m(x)}$$

再者，$x\times f(x)=a_7x^8+a_6x^7+\cdots+a_1x^2+a_0x \pmod{m(x)}$，若 $a_7=0$，则乘积相当于将$f(x)$所对应的字节做左移位(left shift)；若 $a_7=1$，则须将乘积再与 $m(x)-x^8=x^4+x^3+x+1$ 所对应的字节(00011011)做异或。

例如 $$f(x)=x^6+x^4+x^2+x+1=(01010111)$$

则 $$x\times f(x) \pmod{m(x)}=(10101110)=x^7+x^5+x^3+x^2+x$$

$$f(x)=x^7+x^5+x^2+x+1=(10100111)$$

则 $$x\times f(x) \pmod{m(x)}=(01001110)+(00011011)=(01010101)=x^6+x^4+x^2+1$$

3.5.3 对 AES 的描述

3.5.3.1 AES 的特点和结构

Rijndael 为 AES 所定义的迭代式分组密码算法，它的分组长度(block length)和密钥长度(key length)是可以各自独立的，当然它们都可以是 128 位、192 位或 256 位。AES 输入-输出分组规定为 128 位，支持 128 位、192 位和 256 位三种密钥长度，有较好的数学理论作为基础，结构简单、速度快。AES 参数取决于密钥长度的选择，表 3.9 给出了 AES 的参数配置。

表 3.9 AES 参数配置

密钥长度(word/byte/bit)	4/16/128	6/24/192	8/32/256
明文分组长度(word/byte/bit)	4/16/128	4/16/128	4/16/128
轮数(Nr)	10	12	14
扩展密钥总长(word/byte)	44/176	52/208	60/240

AES 不属于 Feistel 结构，加密、解密相似但不对称，它采用的是 SP 网络结构，如图 3.9 所示，在这种密码的每一轮中，轮输入首先被一个由子密钥控制的可逆函数 S 作用，然后再对所得结果用置换(或可逆线性变换)P 作用。S 和 P 分别被称为混淆层和扩散层，主要起混淆和扩散作用。与 Feistel 网络相比，每一轮对整个数据分组进行了处理，因而可以得到更快的扩散。

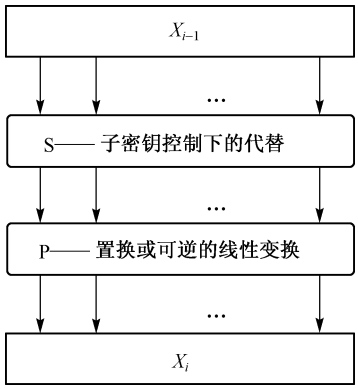
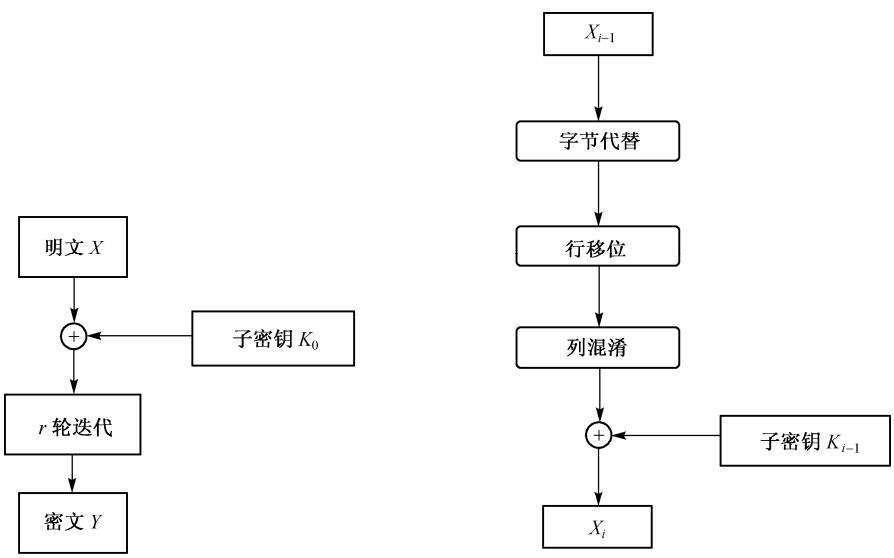


图 3.9　一轮 SP 网络加密过程

图3.10(a)是AES加密算法的框图。第一轮之前，应用了一个密钥加层。它对密码的安全性不做任何贡献。为了使得加密和解密在结构上更为相似，最后一轮的线性混合层与其他的混合层不同。可以证明它不会影响算法的安全性。AES 的每一轮由四个阶段组成，如图3.10(b)所示，分别是：

- **字节代替**：用一个 S 盒完成分组的按字节代替。
- **行移位**：是一个简单的置换。
- **列混淆**：一个利用在域 GF(2^8)上的算术特性的代替。
- **密钥加层**：轮密钥简单地异或到中间状态上。



(a) AES 加密算法框图　　(b) 一轮 AES 迭代结构

图 3.10　AES 算法结构

3.5.3.2 AES 算法流程

假设 State 表示数据，以及每一轮的中间结果，RoundKey 表示每一轮对应的子密钥，算法如下：

第一轮之前执行 AddRoundKey(State，RoundKey)

```
Round(State，RoundKey) {
   ByteSub(State);
   ShiftRow(State);
   MixColumn(State);
   AddRoundKey(State，RoundKey);
}
FinalRound(State,RoundKey) {
   ByteSub(State);
   ShiftRow(State);
   AddRoundKey(State，RoundKey);
}
```

3.5.3.3 字节代替变换

AES 用状态(State)表示输入和输出数据，以及每一轮的中间结果。一个状态可以表成 4 行 N_b 列的一个矩阵，其中 N_b=分组长度/32。密钥(Cipher key)也表示成一个 4 行 N_k 列的矩阵，其中 N_k=密钥长度/32。图3.11给出了当明文分组为128bit，密钥长度也为128bit 时的状态和密钥矩阵。这里，矩阵的每个元素为 1 字节，明文分组和密钥以字节为单位在矩阵中按照从上到下从左到右的顺序排列。

S_{00}	S_{01}	S_{02}	S_{03}
S_{10}	S_{11}	S_{12}	S_{13}
S_{20}	S_{21}	S_{22}	S_{23}
S_{30}	S_{31}	S_{32}	S_{33}

(a)

k_{00}	k_{01}	k_{02}	k_{03}
k_{10}	k_{11}	k_{12}	k_{13}
k_{10}	k_{21}	k_{22}	k_{23}
k_{30}	k_{31}	k_{32}	k_{33}

(b)

图 3.11 (a)状态矩阵；(b)密钥矩阵

字节代替是一个非线性的字节代替，独立地在每个状态字节上进行运算。S 盒输入和输出都是8 bit 的，是由两个可逆变换复合而成，首先在有限域 GF(2^8)中取逆元，零元 00 的逆元规定为 00。其次，将所得逆元再经过下面定义的[GF(2)上的]仿射变换的作用。逆元是为了确保每个值在表中刚好出现一次，仿射变换有助于打破原有模式。完整的代替表如表3.10 所示。例如，字节 21 用第 2 行、第 1 列的 FD 代替。设 S 盒中的输入字节记为 $(x_7, x_6, x_5, x_4, x_3, x_2, x_1, x_0)$，输出字节记为 $(y_7, y_6, y_5, y_4, y_3, y_2, y_1, y_0)$，AES 的字节代替变换可以描述如下：

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

表 3.10　AES 的 S 盒

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

3.5.3.4　行移位变换

虽然 AES 指定分组长度只能为128，但是原来的 Rijndael 算法却允许 128 位、192 位和 256 位的分组。对于 128 位和 192 位的分组，将状态(State)中的第 i 行循环左移$(i-1)$字节，对 256 位的分组，第 3 行和第 4 行额外再移一个字节。具体移位字节见表 3.11。

表 3.11 对应于不同分组长度的行位移量

行 / 分组长度 N_b	0	1	2	3
4	0	1	2	3
6	0	1	2	3
8	0	1	3	4

还是以 $N_b=4$ (128 bit)为例

87	F2	4D	97
EC	6E	4C	90
4A	C3	46	E7
8C	D8	95	A6

→

87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

3.5.3.5　列混淆变换

在列混淆变换中，将状态的列视为有限域 GF(2^8)上的四维向量并且被 GF(2^8)上的一个固定的可逆方阵 $\boldsymbol{A}$ 左乘，设输入状态为$(a_{0j}, a_{1j}, a_{2j}, a_{3j})$，输出状态为$(b_{0j}, b_{1j}, b_{2j}, b_{3j})$，列混淆变换可描述如下：

$$\begin{bmatrix} b_{0j} \\ b_{1j} \\ b_{2j} \\ b_{3j} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} a_{0j} \\ a_{1j} \\ a_{2j} \\ a_{3j} \end{bmatrix}$$

3.5.3.6 轮密钥加变换

用简单的比特异或将一个轮密钥作用在状态上，$b_{ij}=a_{ij}+k_{ij}$，如图3.12所示。轮密钥是通过密钥调度算法从密钥中产生，轮密钥长度等于分组长度。

a_{00}	a_{01}	a_{02}	a_{03}
a_{10}	a_{11}	a_{12}	a_{13}
a_{20}	a_{21}	a_{22}	a_{23}
a_0	a_{31}	$a\,k_{32}$	a_{32}

输入状态

+

k_{00}	k_{01}	k_{02}	k_{03}
k_{10}	k_{11}	k_{12}	k_{13}
k_{20}	k_{21}	k_{22}	k_{23}
k_{30}	k_{31}	k_{32}	k_{33}

子密钥

=

b_{00}	b_{01}	b_{02}	b_{03}
b_{10}	b_{11}	b_{12}	b_{13}
b_{20}	b_{21}	b_{22}	b_{23}
b_{30}	b_{31}	b_{32}	b_{33}

输出状态

图 3.12 轮密钥加

3.5.3.7 密钥扩展算法

轮密钥是通过密钥调度算法从密钥中产生，包括两个组成部分：密钥扩展和轮密钥选取。密钥扩展算法先将密钥扩展成一个扩展密钥，轮密钥按下述方式从扩展密钥中选取：第一个轮密钥由开始 N_b 个字组成，第二个轮密钥由接下来的 N_b 个字组成，如此继续下去。其基本参数如下所述。

轮密钥发生器的输入是初始密钥，长度为 $32\times N_k$ 比特，其中 $N_k=4$，6 或者 8，其输出为各轮的轮密钥。轮密钥只用在 Add RoundKey 中，长度等于明文分组，为 $32\times N_b$ 比特。进行 N_r 轮变换，需要 N_r+1 个轮密钥，所以总共扩展的密钥长为 $32\times N_b\times (N_r+1)$ 比特。将轮密钥的序列表示成 4 行的矩阵，其中每个元素都是一个字节(8 bit)，则矩阵需要有 $N_b\times (N_r+1)$ 列。若设 $N_b=4$，$N_r=10$，且将第 i 列记为一个 32 bit 的字(word) $w[i]$，我们有

k_0	k_4	k_8	k_{12}
k_1	k_5	k_9	k_{13}
k_2	k_6	k_{10}	k_{14}
k_3	k_7	k_{11}	k_{15}

$k_{0,0}$	$k_{0,1}$	$k_{0,2}$	...	$k_{0,42}$	$k_{0,43}$
$k_{1,0}$	$k_{1,1}$	$k_{1,2}$	...	$k_{1,42}$	$k_{1,43}$
$k_{2,0}$	$k_{2,1}$	$k_{2,2}$	...	$k_{2,42}$	$k_{2,43}$
$k_{3,0}$	$k_{3,1}$	$k_{3,2}$	...	$k_{3,42}$	$k_{3,43}$
$w[0]$	$w[1]$	$w[2]$	...	$w[42]$	$w[43]$

密钥扩展对于 $N_k\leq 6$ 和 $N_k>6$ 的情形是不同的。当 $N_k\leq 6$ 时，密钥扩展的伪代码如下：

```
Key Expansion(byte Key[4*Nk], word W[Nb*(Nr +1)])
{
    for ( i=0;i<Nk;i++)
        w[i]={Key[4*i], Key[4*i+1], Key[4*i+2], Key[4*i+3]):
    for(i=Nk; i< Nb*(Nr +1); i++)
    {
        temp=w[i-1];
        if(i%Nk==0)  temp=SubWord(RotWord(temp)+Rcon[i/Nk]);
```

```
            w[i]=w[i-Nk]⊕temp;
        }
    }
```

首轮的 $w[0], w[1], \cdots, w[N_k-1]$就是初始的密钥。32 bit 的 $w[i-1]$可表示为 4 个字节：$a_0 a_1 a_2 a_3$，于是，RotWord$(w[i-1])=a_1 a_2 a_3 a_0$，即将 4 个字节作为左循环移位。SubWord$(a_0 a_1 a_2 a_3)$=S(a_0) S(a_1) S(a_2) S(a_3)，即将 4 个字节分别用 S 盒进行变换。上两个步骤的结果再与轮常量 Rcon[j]异或。每轮的轮常量均不同，定义为 Rcon[j]=(Rc[j], 0, 0, 0)，Rc[j]的值以十六进制表示为：

j	1	2	3	4	5	6	7	8	9	10
Rc[j]	01	02	04	08	10	20	40	80	1B	36

$N_k>6$ 与 $N_k \leq 6$ 时密钥扩展算法的区别在于，当 i 满足 $i-4$ 是 N_k 的整数倍时，在异或之前，要把 AES 的 S 盒作用到 $w[i-1]$的每个字节。

3.5.3.8　AES 的解密

在AES 的解密算法中，使用各变换算法对应的逆变换，尽管在加密和解密过程中密钥扩展的形式是一样的，但解密算法按逆序使用了扩展密钥。其算法伪代码如下：

```
第一轮之前执行 AddRoundKey(State，RoundKey)
Round(State，RoundKey) {
    InvShiftRow(State)；逆向行移位
    InvByteSub(State)；逆向字节代替变换
    AddRoundKey(State，RoundKey)；
    InvMixColumn(State)；逆向列混淆
}
FinalRound(State,RoundKey) {
    InvShiftRow(State)；
    InvByteSub(State)；
    AddRoundKey(State，RoundKey)；
}
```

1．逆向字节代替变换

逆向字节代替变换利用了表 3.12 所给出的逆 S 盒。输入 FD 到逆 S 盒得到输出 21，输入 21 到 S 盒得到输出 FD。S 盒与逆 S 盒互逆，但 S 盒不是自逆的。设逆 S 盒中的输入字节记为 $(x_7, x_6, x_5, x_4, x_3, x_2, x_1, x_0)$，输出字节记为 $(y_7, y_6, y_5, y_4, y_3, y_2, y_1, y_0)$，逆 S 盒变换也可以用如下方式来描述：

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

表 3.12 AES 的逆 S 盒

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
1	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
2	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
3	08	2E	A1	66	28	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
4	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
5	6C	70	48	50	FD	ED	B9	DA	5E	15	46	57	A7	8D	9D	84
6	90	D8	AB	00	8C	BC	D3	0A	F7	E4	58	05	B8	B3	45	06
7	D0	2C	1E	8F	CA	3F	0F	02	C1	AF	BD	03	01	13	8A	6B
8	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	B4	E6	73
9	96	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DF	6E
A	47	F1	1A	71	1D	29	C5	89	6F	B7	62	0E	AA	18	BE	1B
B	FC	56	3E	4B	C6	D2	79	20	9A	DB	C0	FE	78	CD	5A	F4
C	1F	DD	A8	33	88	07	C7	31	B1	12	10	59	27	80	EC	5F
D	60	51	7F	A9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF
E	A0	E0	3B	4D	AE	2A	F5	B0	C8	EB	BB	3C	83	53	99	61
F	17	2B	04	7E	BA	77	D6	26	E1	69	14	63	55	21	0C	7D

2．逆向行移位变换

AES的逆向行移位变换将状态(state)中的后三行执行相反方向的移位操作，第 i 行循环右移(i–1)字节，如第 3 行循环右移 2 字节。

3．逆向列混淆变换

设输入状态为(a_{0j}, a_{1j}, a_{2j}, a_{3j})，输出状态为(b_{0j}, b_{1j}, b_{2j}, b_{3j})，逆向列混淆变换由如下的矩阵乘法定义：

$$\begin{bmatrix} b_{0j} \\ b_{1j} \\ b_{2j} \\ b_{3j} \end{bmatrix} = \begin{bmatrix} 0\text{E} & 0\text{B} & 0\text{D} & 09 \\ 09 & 0\text{E} & 0\text{B} & 0\text{D} \\ 0\text{D} & 09 & 0\text{E} & 0\text{B} \\ 0\text{B} & 0\text{D} & 09 & 0\text{E} \end{bmatrix} \begin{bmatrix} a_{0j} \\ a_{1j} \\ a_{2j} \\ a_{3j} \end{bmatrix}$$

3.6 分组密码的工作模式

由于实际应用中数据格式的多样性和工作模式安全性的需要，人们需要对分组密码的工作模式进行定义。历史上，为了将 DES 应用于实际，美国国家标准局(NBS)在 FIPS PUB 74 和 81，ANSI 在 ANSI X3.106 中，以及 ISO 和 ISO/IEC 在 ISO 9732 ISO/IEC 10116 都对 DES 的工作模式进行了规定。在 ANSI 标准 ANSI X3.106—1983 定义了四种模式。由于新的应用和要求，2001 年 12 月在特别公布的 SP 800—38A 中，NIST 已将推荐模式扩展为 5 个，分别是电码本工作模式［Electronic Codebook(ECB)Operation Mode］、密码分组链接(Cipher Block Chaining, CBC)工作模式、密码反馈(Cipher FeedBack, CFB)工作模式、输出反馈(Output Feedback, OFB)工作模式和计数器(counter, CTR)工作模式。我国的国家标准 GB/T 17964—2008

《信息安全技术——分组密码的工作模式》定义了 7 种工作模式。这些工作模式可以分为两大类，分别是分组密码工作模式和流密码工作模式。在设计工作模式时，主要考虑以下几种性能：安全性、有效性和容错性。

3.6.1　电码本(ECB)工作模式

最简单的模式是电码本模式，它把明文分成同样长度的分组，对每个明文分组独立地进行加密，如图3.13 所示，明文为 q 个分组 $P_1, P_2, \cdots, P_q$ 所组成的序列，每个分组都是 b 位，密钥为 K，对应的密文为 q 个分组 $C_1, C_2, \cdots, C_q$ 组成的序列，每个分组也是 b 位。

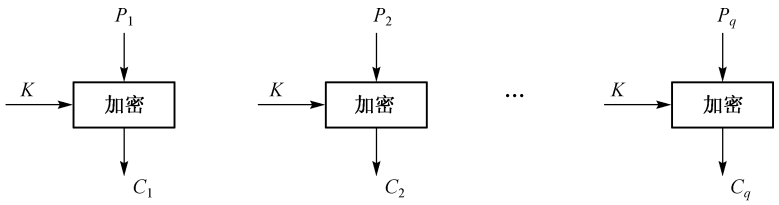
ECB 模式的加密和解密方式可以描述为：

$$C_i = \mathrm{E}_K(P_i)$$
$$P_i = \mathrm{D}_K(C_i) \qquad (i = 1, 2, 3, \cdots, q)$$

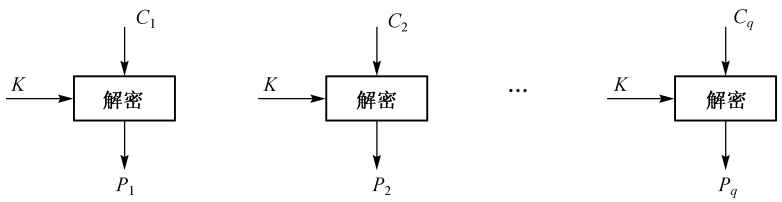
ECB 模式的优点是简单和有效，可以并行实现。对于需要随机访问和处理的文件是方便的，比如数据库的记录，任何一条记录的追加、删除、加密和解密都可以独立于其他记录。

ECB 模式的缺点是不能隐藏明文的模式信息，相同明文总是被加密成相同密文，同样的信息多次出现会造成泄露，另外，用同一密钥加密太多的信息会给攻击者提供统计分析的机会，使安全性降低。此外，对明文的主动攻击是可能的，如果攻击者破译了某些分组，他就可以对信息块进行替换、重排、删除和重放。ECB 模式适合于传输短信息。

在ECB 模式中，如果收到的一个密文分组中的一个或多个位发生差错，只会影响到发生差错的那一个分组的解密。但是如果密文分组被偶然增加或减少一位，就会影响剩余密文的解密。



(a) ECB 加密



(b) ECB 解密

图 3.13　电码本(ECB)工作模式

3.6.2　密码分组链接(CBC)工作模式

为了克服ECB 模式的弱点，我们需要将相同的明文分组加密成不同的密文分组，密码分组链接模式对分组密码加入了反馈机制，每组密文不仅取决于生成它的明文，还和它前面的各组明文相关。如图3.14 所示，当前的明文分组与前一密文组进行异或运算后再进行加密得到当前的密文分组。CBC 模式的加密和解密方式可以描述如下。

对第一个明文分组进行加密：

$$C_1 = \mathrm{E}_K(P_1 \oplus \mathrm{IV})$$

对其他分组加密：

$$C_i = \mathrm{E}_K(C_{i-1} \oplus P_i) \quad (i=2, 3, \cdots, q)$$

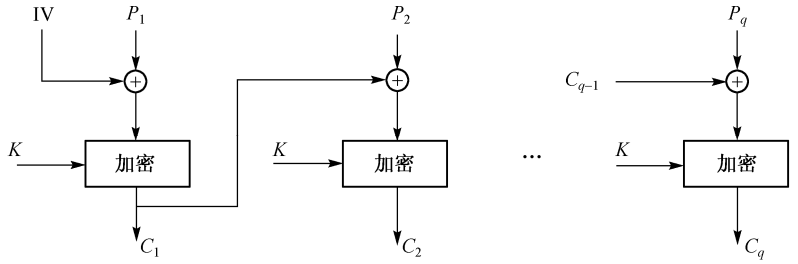
对第一个密文分组进行解密：

$$P_1 = \mathrm{D}_K(C_1) \oplus \mathrm{IV}$$

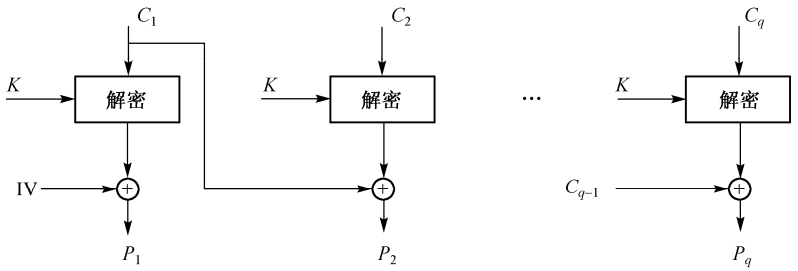
对其他分组解密：

$$P_i = \mathrm{D}_K(C_i) \oplus C_{i-1}$$

CBC 模式没有已知的并行实现算法。能够把相同的明文加密成不同的密文，隐藏明文的模式信息。某些信息使用一个固定的头部，为了防止相同的头部被加密成相同的密文，需要设置共同的初始化向量IV，IV 可以用来改变第一个分组。对明文的主动攻击是不容易的，信息块不容易被替换、重排、删除和重放。适合于传输长度大于分组宽度 b 位的报文，还可以进行用户鉴别，是很多网络加密的标准，如 SSL, IPsec。



(a) CBC 加密



(b) CBC 解密

图 3.14　密码分组链接(CBC)工作模式

在 CBC 方式中，如果收到的一个密文分组的一个或多个位差错会影响对两个分组(即发生差错的分组和随后的分组)的解密。密文错误通常由通信线路和存储介质的故障产生。由小的密文错误而引起大的明文错误的性质称为错误扩散，系统中如出现错误扩散是麻烦的。在 CBC 模式中，由于一组错误密文仅影响它对应的那组明文和随后的一组明文，再后面的明文分组不受影响，均能正确解密，所以该模式是可以自恢复的。虽然CBC 模式可以从位错误中很快恢复，但如果密文中增加或减少一位，那么后面的密文组都会受到影响，解密时将连续出现无意义的信息，因此，使用时需确保各组密码结构的完整性。

3.6.3　密码反馈(CFB)工作模式

DES, AES虽然本质上是分组密码，但是利用密码反馈(CFB)模式或输出反馈(OFB)模式，也可做流密码使用。流密码不需要明文长度是分组长度的整数倍，可以实时操作，密文与明文等长。在CFB模式中，数据能够按照比分组更小的单位进行处理，用户可以使用1位的 CFB一次加密1位数据。图3.15描述了基于 b 位分组密码算法的CFB模式。明文为 q 个变量 $P_1, P_2, \cdots, P_q$ 所组成的序列，每个输入变量都为 s 位，密钥为 K，初始值 IV 为 b 位，R_i 为移位寄存器，宽度为 b 位。

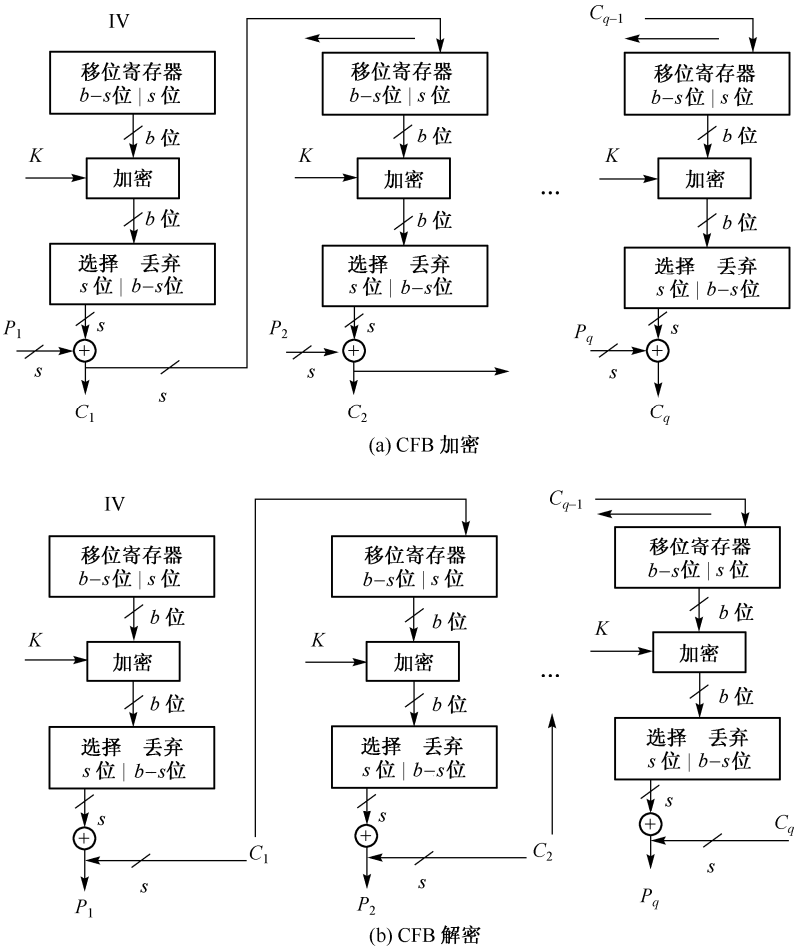


图 3.15　密码反馈(CFB)工作模式

CFB 模式的加密和解密方式可以描述如下：

加密：$C_i = P_i \oplus (\mathrm{E}_K(R_i)$ 的高 s 位)，$R_{i+1} = (R_i \ll s) \,||\, C_i$

解密：$P_i = C_i \oplus (\mathrm{E}_K(R_i)$ 的高 s 位)，$R_{i+1} = (R_i \ll s) \,||\, C_i$

其中 $R_1 = \mathrm{IV}$，$||$是连接函数，$R_i \ll s$ 表示 R_i 左移 s 位。

加密时，加密函数的输入是 b 位的移位寄存器，移位寄存器中放的是初始值 IV，加密函数的输出最左边的 s 位与明文异或得到第一个密文单元 C_1，把 C_1 发送出去，接着移位寄存器左移位，C_1 填入寄存器的最右边。这样，直到所有明文单元被加密。解密时将收到的密文单

元与加密函数的输出异或得到明文单元，这里需要注意的是OFB 的解密使用的也是分组加密函数，而非解密函数。

CFB模式隐藏了明文模式，需要共同的移位寄存器初始值 IV，同时，对于不同的消息，IV 必须唯一。没有已知的并行实现算法。使用 CFB 算法时，明文中 1 位错误会影响所有的密文，但在解密时仅影响错误位。1 位的密文错误首先导致被解密的明文的 1 位错误，其次，该错误被输入到移位寄存器中，又会引起后面的密文错误，直到这个错误被移出寄存器为止。当 $b=128$，$s=8$ 时，1 位密文错误可导致 17 个字节的明文错误，然后系统恢复正常，其后的密文能正确解密。一般而言，s 位 CFB 模式中，1 位密文错误将影响当前和后面的 b/s 个密文的正确解密。

3.6.4 输出反馈(OFB)模式

输出反馈模式和密码反馈模式很相似，与 CFB 模式不同的是，它用加密函数的输出来填充移位寄存器。图 3.16 描述了基于 b 位分组密码算法的 OFB 模式。明文为 q 个变量 $P_1, P_2, \cdots, P_q$ 所组成的序列，每个变量都为 s 位，密钥为 K，初始值 IV 为 b 位，R_i 为移位寄存器，宽度为 b 位。

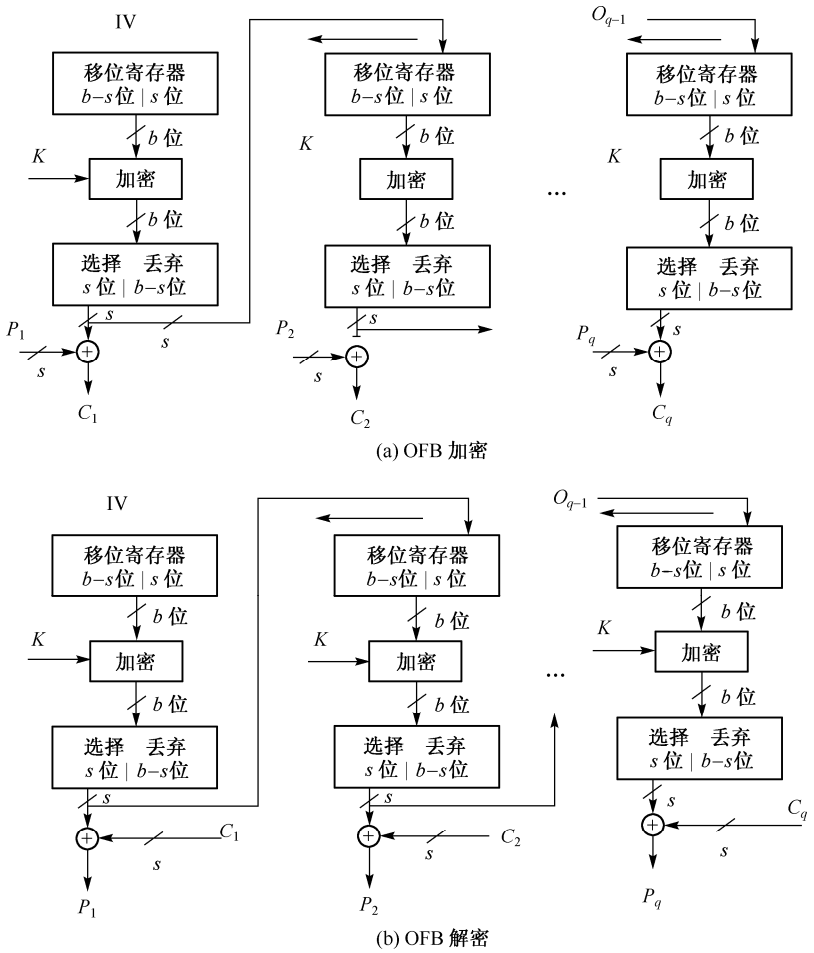


图 3.16 输出反馈(OFB)工作模式

OFB 模式的加密和解密方式可以描述如下：

加密：$C_i = P_i \oplus (\mathrm{E}_K(R_i)$的高 s 位），$R_{i+1} = (R_i \ll s) \parallel (\mathrm{E}_K(R_i)$的高 s 位）

解密：$P_i = C_i \oplus (\mathrm{E}_K(R_i)$的高 s 位），$R_{i+1} = (R_i \ll s) \parallel (\mathrm{E}_K(R_i)$的高 s 位）

其中，R_1=IV，‖是连接函数。

OFB 也是一种将分组密码作为流密码运行的模式，与CFB 模式相同，它也能隐藏明文模式，需要共同的移位寄存器初始值 IV，IV 也应该是唯一的，并且没有已知的并行实现算法。OFB 模式的一个优点是大多数工作可以脱机进行，在密文没到达前计算 $\mathrm{E}_K(R_i)$，当密文到达后只需要进行异或就能得到明文。OFB 模式的传输错误不会扩散，1 位密文的错误只导致对应的 1 位明文的错误。OFB 模式的缺点是，对明文的主动攻击是可能的，如果密文中的某位取反，恢复出的明文相应位也取反。安全性较 CFB 差。

3.6.5　计数器(CTR)模式

尽管计数器模式在2001 年才被写入标准，但是这种模式很早就已经提出来了。与OFB 模式类似，但是加密的是计数器值，而不是密文反馈值，必须对每一个明文使用一个不同的密钥和计数值，如图3.17 所示。

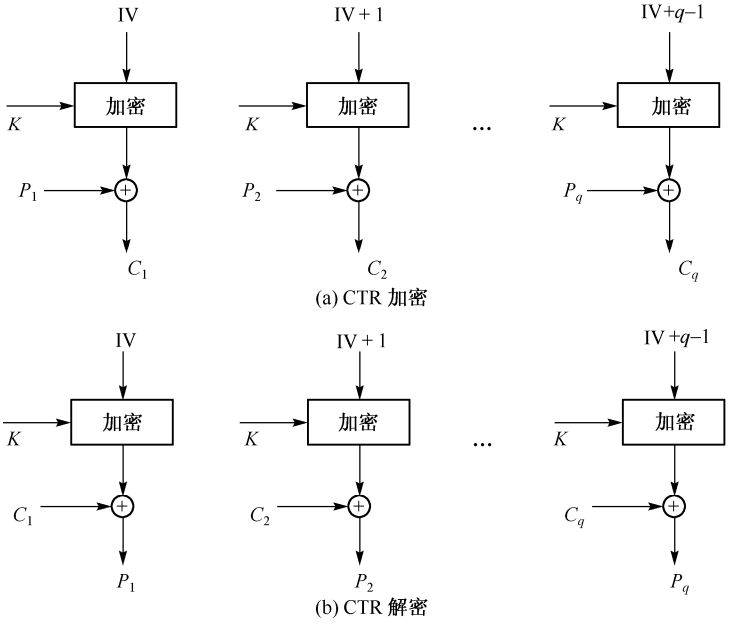


图 3.17　计数器(CTR)工作模式

CTR 模式的加密和解密方式可以描述如下：

加密：$C_i = P_i \oplus O_i$(取 O_i 与 P_i 长度相同的位数），$O_i = E_K$ (IV + i–1)，IV 为计数器的初始值。

解密：$P_i = C_i \oplus O_i$

计数器模式的优点是：

- 高效，这表现在它可以并行实现，每块数据不依赖于前面的数据，可以独立处理；基本加密算法的执行不依赖明密文的输入，如果存储器充足且安全，可以进行预处理，当给出明密文时，只需要进行异或。适用于高速网络加密。

- 它可以随机访问加密的数据分组，每个分组都可以独立加解密。
- 隐藏了明文模式，与 CBC 模式一样安全。
- 可以处理任意长度的消息，但对于每次加密，需要使用不同的密钥的计数器值。
- 从误差传递上讲，一个单元损坏只影响对应单元。

除了上述模式外，常用的分组密码模式还有一些，大多数是上述模式的变形，如分组链接(Block Chaining，BC)模式、传播密码分组链接(Propagating Cipher Block Chaining，PCBC)、带有校验和的密码分组链接(Cipher Block Chaining with Checksum, CBCC)和带有非线性函数的输出反馈模式(Output Feedback with a nonlinear Function, OFBNLF)。BC模式和 OFBNLF 模式被写入了 GB/T 17964—2008。

3.6.6 不是分组长度整数倍的报文的处理

在 ECB 和 CBC 模式中，还涉及一个问题：大多数信息不可能被完整地分成几组，最后一部分通常不够一组，对此，可以采用填充的方式来解决。填充就是采用全 0，全 1 或 0 与 1 组合的方式来填充最后的短块，使其成为一个完整的分组。有多种填充方法。

美国联邦信息处理标准PUB 81建议，如果数据是二进制的，就填入与数据最后一位相反的比特；如果数据块是 ASCII 码，就填入随机字节。将填充位(bit)数目写入最后一个字节，总共凑齐 64 位的分组。填充一般要多于 64 bit。这样在解密后删除填充比特即可，因为知道了填充比特数。

在 RFC2040 中定义了 RC5-CBC-Pad 和 RC5-CTS。CBC-Pad 是用于 RC5 密码的一种分组密码工作模式，CBC-Pad 能够处理任意长度的明文，密文至多比明文多一个分组的长度。填充方法是在每个报文的末尾，增加从 1 到 bb 字节的填充，其中 bb 等于 RC5 以字节为单位的分组大小。填充字节被置为一个代表填充字节数目的字节。如果有 8 字节的填充，则每个字节具有 00001000 的比特模式。

为了进行自动数据处理，通常总是要填充的。即使是分组长度整数倍的报文，也要填充。解密后，最后一个分组的最后一个字节用于确定必须被剥掉的填充的数位，因此不能允许零字节的填充。

填充也不总是合适的，例如，当我们希望使用同样的内存空间来存储明文和对应的密文时，就要求密文必须和明文一样长。RC5-CTS 就是满足这一要求的一种工作模式。图3.18给出了一种ECB 工作模式的填充方式，称为**密文挪用**，即采用部分密文填充最后的短块。图中 P_{n-1} 是最后一组完整的明文(倒数第二组明文)，P_n 是最后的短明文分组，C_{n-1} 是最后一组完整的密文，C_n 是最后的短密文分组。选择 P_{n-1} 加密后的密文与 P_n 同样长度的左边 L 位作为 C_n，用 P_{n-1} 加密后的密文的右边 $b-L$ 位对 P_n 填充到分组长度，对其加密得到 C_{n-1}。同样的思想方法也可以用于 CBC 模式。

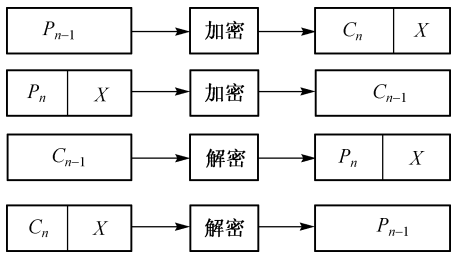
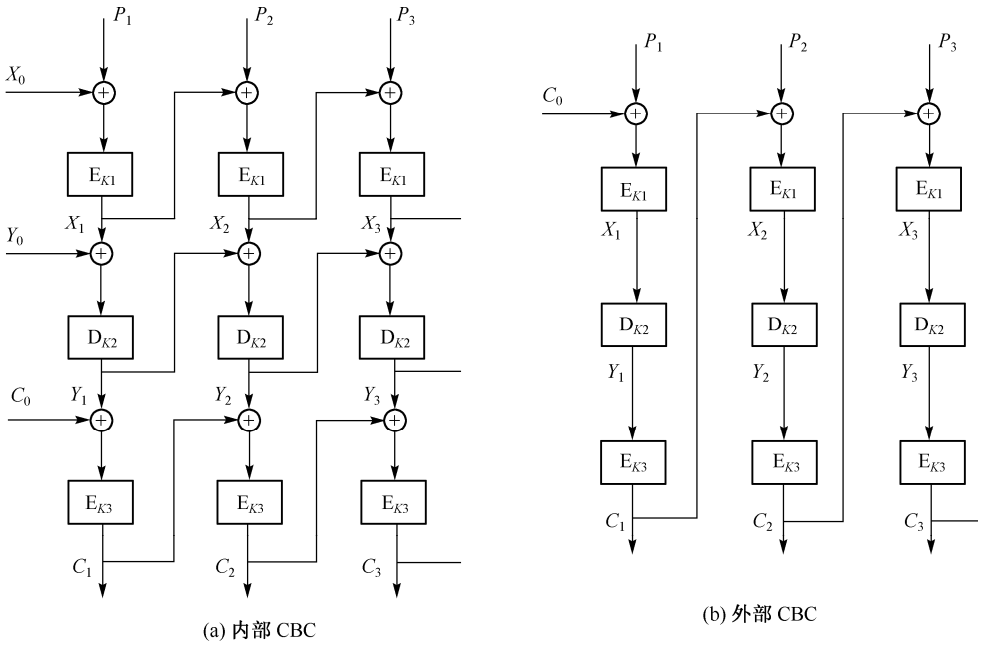


图 3.18 ECB 模式中的密文挪用

3.6.7　三重 DES 的工作模式

这里给出两个可能的三重 DES 加密模式，分别称为内部 CBC 方式(Inner-CBC)和外部 CBC(Outer-CBC)方式。

内部 CBC 用 CBC 方式对整个文件进行三次不同的加密，这需要三个不同的 IV 值 X_0, Y_0, C_0，如图3.19(a)所示。外部 CBC 用 CBC 方式对整个文件进行三重加密，需要一个 IV 值 C_0，如图3.19(b)所示。



(a) 内部 CBC　　(b) 外部 CBC

图 3.19　CBC 模式下的三重 DES

如果用软件实现的话，两种方式的性能是相当的，外部 CBC 比内部 CBC 方式每个分组少两个异或操作，内部CBC 方式可以在交换之前用同一密钥加密多个分组，这两种模式性能的差别比编程方式的正常变化带来的差别要小。

如果用硬件实现的话，从性能上考虑，这两种模式都比单重加密需要更多的资源。然而，如果将内部 CBC 方式制作成加密芯片，其加密并不比单重加密慢，因为三次加密是独立的，每重加密是自反馈的，所以，三块加密芯片可以同时独立地工作，而外部CBC 中反馈是在三次加密之后，这意味着即使使用三块芯片，其处理效率也只有单重加密的 1/3。

令 P_i 表示输入的明文分组流，X_i 表示第一个 DES 的输出，Y_i 是第二个 DES 的输出，C_i 是最后一个 DES 的输出，也就是整个系统的密文。

在外部 CBC 方式中，$X_i = \mathrm{DES}(\mathrm{XOR}(P_i, C_{i-1}))$，$Y_i = \mathrm{DES}(X_i)$，$C_i = \mathrm{DES}(Y_i)$，这里，$C_0$ 是单一的初始向量 IV。如果 P_1 在时刻 $t=0$ 得到(以 DES 操作的时间为测量单位)，X_1 将会在 $t=1$ 得到，Y_1 在 $t=2$ 得到，C_1 在 $t=3$ 得到。$t=1$ 时，第一个 DES 有空闲做更多的工作，但是要做的工作是：$X_2 = \mathrm{DES}(\mathrm{XOR}(P_2, C_1))$，因为 C_1 只有在 $t=3$ 时才能得到，因此 X_2 只有在 $t=4$ 时才能得到，Y_2 在 $t=5$，C_2 在 $t=6$ 时才能得到。

在内部 CBC 方式，我们有：

$$X_i = \mathrm{DES}(\mathrm{XOR}(P_i, X_{i-1}))$$
$$Y_i = \mathrm{DES}(\mathrm{XOR}(X_i, Y_{i-1}))$$
$$C_i = \mathrm{DES}(\mathrm{XOR}(Y_i, C_{i-1}))$$

这里 X_0, Y_0 和 C_0 是三个独立的初始向量 IV。如果 P_1 在 $t=0$ 时得到，X_1 在 $t=1$ 时得到。X_2 和 Y_1 同时在 $t=2$ 时得到，X_3, Y_2 和 C_1 在 $t=3$ 时得到，X_4, Y_3 和 C_2 在 $t=4$ 时得到。因此每一个 DES 操作都会产生一个新的密文分组，与外部 CBC 方式相比，它的吞吐量是内部 CBC 方式的三倍。

从安全性上考虑，简单的加密模式安全性也差一些，Biham 对不同模式进行选择密文差分分析，发现内部 CBC 方式比单重加密安全性好一些，外部 CBC 方式比内部 CBC 方式安全性好一些。如果初始向量 IV 是保密的，内部 CBC 比外部 CBC 方式对于强力攻击来说需要确定更多的数值位，因而更安全。

3.7 流密码

对称密码体制根据对明文的加密方式的不同而分为分组密码和流密码。分组密码的连续明文数字组使用相同的密钥来进行加密。密文组 $y=y_1y_2\cdots y_m$ 通过如下方式对明文组 $x=x_1x_2\cdots x_m$ 加密得到：$y=e_k(x_1)e_k(x_2)\cdots e_k(x_m)$。

流密码的基本思想是利用密钥 k 产生一个密钥流 $z=z_0z_1\cdots z_m$，并使用如下规则加密明文串 $x=x_0x_1x_2\cdots$，得到密文串 $y=y_0y_1y_2\cdots=e_{z0}(x_0)e_{z1}(x_1)e_{z2}(x_2)$。

尽管分组密码也可以工作在流密码的模式，与分组密码相比，流密码的主要优点是速度更快而且需要编写的代码更少。流密码目前主要还是应用在军事、外交、无线通信等领域，虽然也有一些公开的设计和研究成果发表，但大多数还是保密的。目前可以见到的流密码算法有 RC4, A5, SEAL 和 PIKE 等。

3.7.1 流密码的定义

与分组密码类似，可以给出如下定义：

同步流密码是一个六元组(P, C, K, L, E, D)和函数 f，并且满足条件：

(1) P 是可能明文的有限集(明文空间)。

(2) C 是可能密文的有限集(密文空间)。

(3) K 是一切可能密钥构成的有限集(密钥空间)。

(4) L 是一个称为密钥流字母表的有限集。

(5) f 是一个密钥流生成器。f 使用密钥 k 作为输入，产生无限的密钥流 $z=z_0z_1\cdots, z_i\in L$，$i\geqslant 1$。

(6) 对任意 $z_i\in L$，有一个加密算法 $e_{z_i}\in E$ 和相应的解密算法 $d_{z_i}\in D$，并且对每一明文 $x\in P$，e_{z_i}：$P\rightarrow C$ 和 d_{z_i}：$C\rightarrow P$ 分别为加密解密函数，满足 $d_{z_i}(e_{z_i}(x))=x$。

3.7.2 同步流密码

根据加密器中的记忆元件的存储状态 σ_i 是否依赖于输入的明文字符，流密码可进一步分成同步和自同步两种。σ_i 独立于明文字符的叫做**同步流密码**，否则叫做**自同步流密码**。

同步流密码中，密钥流由密钥流发生器 f 产生：$z_i=f(k, \sigma_i)$，$\sigma_i=f_s(k, \sigma_{i-1})$，这里 σ_i 是加密器中的记忆元件(存储器)在时刻 i 的状态，f 是输出函数，f_s 是状态转移函数。

同步流密码中，由于 $z_i=f(k,\ \sigma_i)$ 与明文字符无关，因而密文字符 $y_i=e_{z_i}(x_i)$ 也不依赖于此前的明文字符。因而，可将同步流密码的加密器分成密钥流生成器和加密变换器两个部分。如果与上述变换对应的解密变换为 $x_i=d_{z_i}(y_i)$，则可给出同步流密码的模型，如图3.20所示。

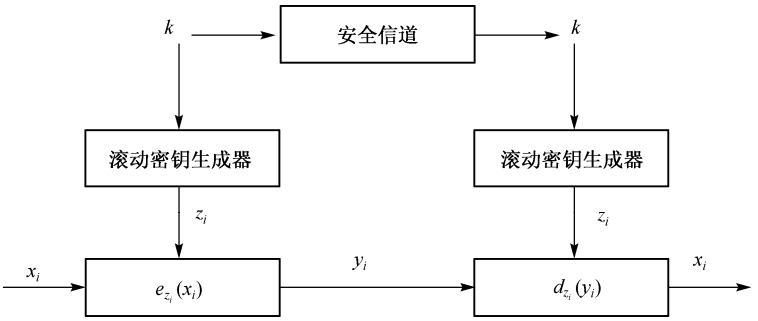


图 3.20　同步流密码体制模型

同步流密码各位之间是真正独立的，因此，它的一个重要优点就是无错误传播，即一位传播错误只影响一位，不影响后面的位。

同步流密码的加密变换 e_{z_i} 可以有多种选择，只要保证变换是可逆的即可。实际使用的数字保密通信系统一般都是二元系统，因而在有限域 GF(2)上的二元加法流密码是目前应用最多的流密码体制，其加密变换可表示为 $y_i=z_i\oplus x_i$，⊕表示模 2 加(即异或)。

自同步流密码中，密钥流的产生过程可用函数描述为：

$$z_i=f(k,\sigma_i),\qquad \sigma_i=f_s(k,\sigma_{i-1},c_{i-1},c_i,c_{i+1},c_{i-n})$$

若每一个密钥字符是由前面 n 个密文字符参与运算推导出来的，如果在传输过程中发生错误，则这一错误就要向前传播 n 个字符。因此，自同步流密码有错误传播现象。在自同步流密码中，密文流和明文流都参与了密钥流的生成，密钥流的分析将复杂化，导致较难从理论上进行详尽的分析。因此，目前用的较多的是同步流密码。

3.7.3　密钥流生成器

流密码的安全强度取决于密钥流生成器所产生的密钥流的性质，当密钥流 $z=z_0z_1\cdots z_i$ 是一个完全随机的非周期序列时，它实际上就实现了一次一密，而实际应用中的密钥流是一个伪随机序列。所以，流密码设计当中最核心的问题是密钥流生成器的设计。设计流密码从两方面进行考虑：一是从系统自身的复杂性度量出发，如输出密钥序列的周期、线性复杂度和随机性等；另一方面从抗已知攻击出发，如线性逼近和统计分析等。目前设计密钥流生成器主要的准则如下：

- 密钥量足够大：因为密钥流的输出取决于输入密钥的值，为防止强力攻击，密钥应该足够长，在目前的软/硬件技术条件下，应不小于 128 位。
- 加密序列的周期足够长：重复的周期越长，密码分析的难度就越大，一般为 2^{128} 或 2^{256}。
- 密钥流应该尽可能地接近于一个真正的随机数流的特征，密钥流的随机特性越好，密文越随机，密码分析就越困难。

为了设计安全的密钥流生成器，必须在生成器中使用非线性变换，Rueppel将这类生成器分成两部分，即驱动部分和非线性组合部分。驱动部分控制生成器的状态序列，并为非线性

组合部分提供统计性能良好的序列。驱动部分可由一个线性反馈移位寄存器组成，而非线性组合部分将驱动部分提供的序列组合成密码学特性良好的序列。

3.7.3.1 一种产生密钥流的方法

下面给出一种产生密钥流的方法。

从密钥$(k_1, k_2, \cdots, k_m)$开始，假定$z_i = k_i$，$1 \leq i \leq m$，利用次数为m的线性递归关系来产生密钥流：

$$z_{i+m} = \sum_{j=0}^{m-1} c_j z_{i+j} \bmod 2$$

这里$c_0, c_1, \cdots, c_{m-1} \in Z_2$是确定的常数。密钥$K$由$2m$个值组成$k_1, k_2, \cdots, k_m, c_0, c_1, \cdots, c_{m-1}$。当$(k_1, k_2, \cdots, k_m) = (0, 0, \cdots, 0)$，则生成的密钥流全为零，这种情况是要避免的。如果常数$c_0, c_1, \cdots, c_{m-1}$选择适当的话，则任意非零初始向量$(k_1, k_2, \cdots, k_m)$都将产生周期为$2^m - 1$的密钥流。

设$m = 4$，按下述线性递归关系产生密钥流：

$$z_{i+4} = (z_i + z_{i+1}) \bmod 2, \quad i \geq 1$$

若初始向量为$(1, 0, 0, 0)$，则可以产生周期为$2^4 - 1 = 15$的密钥流如下：

$$100010011010111\cdots$$

这种密钥流产生器可以使用线性反馈移位寄存器以硬件方式有效实现。

3.7.3.2 反馈移位寄存器

一个反馈移位寄存器(feedback shift register)由两部分组成：移位寄存器和反馈函数。如图3.21所示。

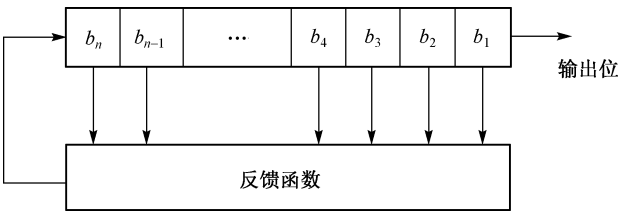


图 3.21 反馈移位寄存器

序列的线性复杂度定义为生成这个序列的最短线性移位寄存器的长度。移位寄存器的周期是指输出序列从开始到重复时的长度。最简单的反馈移位寄存器是线性反馈移位寄存器(Linear Feedback Shift Register, LFSR)，如图3.22 所示，因为容易通过数字硬件实现。

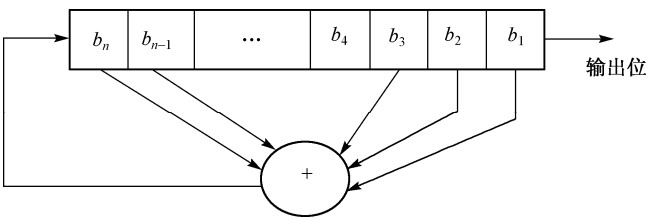


图 3.22 线性反馈移位寄存器

一个n位 LFSR 能够处于$2^n - 1$个内部状态中的一个，理论上，n位 LFSR 在重复之前能

产生 2^n-1 位长的伪随机序列。只有具有特定选择序列的 LFSR 才会遍历所有的 2^n-1 个内部状态，这是最大周期的 LFSR，这样的 LFSR 产生的输出序列称为 ***m* 序列**。

LFSR 用软件实现时运行速度很慢，基于 LFSR 的密码通常都用硬件实现。

3.7.3.3　基于 LFSR 流密码的密码分析

密文以方式 $y_i=(x_i+z_i) \bmod 2$ 产生，利用下述的线性递归关系从初态 $(z_1, z_2, \cdots, z_m)=(k_1, k_2, \cdots, k_m)$ 产生密钥流：

$$z_{m+i}=\sum_{j=0}^{m-1} c_j z_{i+j} \bmod 2, i \geq 1$$

这里 $c_0, c_1, \cdots, c_{m-1} \in Z_2$。

它容易受到已知明文攻击。假定攻击者 Oscar 有了明文串 $x_1 x_2 \cdots x_n$ 和相应的密文串 $y_1 y_2 \cdots y_n$，那么他能计算密钥流 $z_i=(x_i+y_i) \bmod 2$，$1 \leqslant i \leqslant n$，若 Oscar 知道 m 的值，他仅需计算 $c_0, c_1, \cdots, c_{m-1}$。

如果 $n \geq 2m$，就有 m 个未知数的 m 个线性方程

$$(z_{m+1}, z_{m+2}, \ldots, z_{2m})=(c_0, c_1, \ldots, c_{m-1})\begin{bmatrix} z_1 & z_2 & \cdots & z_m \\ z_2 & z_3 & \cdots & z_{m+1} \\ \vdots & \vdots & \ddots & \vdots \\ z_m & z_{m+1} & \cdots & z_{2m-1} \end{bmatrix}$$

$$(c_0, c_1, \ldots, c_{m-1})=(z_{m+1}, z_{m+2}, \ldots, z_{2m})\begin{bmatrix} z_1 & z_2 & \cdots & z_m \\ z_2 & z_3 & \cdots & z_{m+1} \\ \vdots & \vdots & \ddots & \vdots \\ z_m & z_{m+1} & \cdots & z_{2m-1} \end{bmatrix}^{-1}$$

这样一来，如果知道 $2m$ 比特的输出串，很显然能够完全分析出 m 级线性移位寄存器。例如 $m=20$，非重复串的最大长度为 $2^{20}-1$，超过 100 万，而 $2m$ 的值仅为 40。因为线性反馈移位寄存器很容易被分析出来，所以实际应用的流密码必须使用非线性反馈移位寄存器来产生所需的伪随机密钥比特。因为线性反馈移位寄存器容易理解和构造，通常将许多LFSR组合成一个非线性反馈移位寄存器。组合的方法主要有：

(1) 非线性组合若干个 LFSR 的输出。

(2) 非线性过滤一个 LFSR 的输出。

(3) 一个或若干个 LFSR 的输出被用于控制主 LFSR 的时钟。

3.7.4　RC4

RC4 是一个密钥长度从 1 字节到 256 字节(或 8 比特到 2048 比特)可变的流密码，于 1987 年由 Ron Rivest 设计，它保密了 7 年，于 1994 年匿名公开于 Internet 上。RC4 是一个面向字节的流密码，它的密钥流序列独立于明文，RSA 声称 RC4 对线性和差分分析具有免疫力。由于 RC4 是流密码，必须避免重复使用密钥。它是目前应用和影响力都最为广泛的一种流密码算法，被应用于 SSL/TLS，WEP(IEEE802.11)等协议中。

我们可以用下述摸球模型来解释 RC4 算法。假设一个口袋中放有 256 个完全一样的球，分别标有编号 0, 1, 2, …, 255。把所有的球充分混合，随机取出一个球并记下该球的编号，再把取出的球放回袋中。重复这样做，如果重复的次数足够多，每个球出现的概率应该是相

等的，这样将得到一个取出的球的编号序列，它是由整数 0, 1, 2, …, 255 构成的随机序列。当序列足够长时，各个数出现的概率大致相等。RC4 算法正是这一摸球模型的具体实现。

RC4 算法可分为两个阶段，第一个阶段是对一个 256 个字节的状态矢量 ***S*** 的初始化阶段，相当于把编号后的球充分混合。第二个阶段是密钥流的生成阶段，相当于从袋中重复随机取球。

初始化阶段首先对 $S[0]$到 $S[255]$进行线性填充，即 $S[0] = 0, S[1] = 1, \cdots, S[255] = 255$，同时建立临时状态矢量 ***T***，将密钥 K 赋给 ***T***，如果 K 的长度为 256 字节，则直接赋值，如果密钥长度小于 256 字节为 keylen 字节，则循环重复把 K 赋给 ***T***。这一操作的伪代码如下：

```
for i=0 to 255 do
    S[i]=i;
    T[i]=K[i mod keylen]
```

然后用 ***T*** 对 ***S*** 进行初始置换，这一操作的伪代码如下：

```
j=0;
for i = 0 to 255 do
    j = (j + S[i] + T[i]) mod 256;
    Swap S[i] and S[j];
```

初始化阶段结束后，状态矢量 ***S*** 的元素 $S[0], S[1], \cdots, S[255]$变成了整数 0 到 255 之间的随机值。

在密钥流生成阶段中，$S[t]$存放的是取出的球的编号，$t = (S[i]+S[j]) \bmod 256$ 的功能是希望第 t 个球是随机取出的；交换 $S[i]$和 $S[j]$(Swap $S[i]$ and $S[j]$)的目的是每取一次球，就对袋子中的球重新“混和一把”。这一阶段的伪代码如下：

```
i,j=0
While(true)
    i = (i + 1) mod 256;
    j = (j + S[i]) mod 256;
    Swap S[i] and S[j];
    t = (S[i] + S[j]) mod 256;
    k = S[t];
```

根据需要可产生足够长度的密钥流 k，加密时 k 与明文异或产生密文，与密文异或产生明文。

3.7.5 A5 算法

A5 序列密码是欧洲 GSM(Group Special Mobile)标准中规定的加密算法，用于数字蜂窝移动电话的加密，加密从用户设备到基站之间的链路。A5 由三个 LFSR 组成，移位寄存器的长度分别是 19, 22 和 23，三个 LFSR 在时钟控制下运行。A5 算法有两个版本：强安全性的 A5/1 和弱安全性的 A5/2。

3.8 随机数

在结束本章，介绍公钥密码之前，最后再补充介绍一下随机数的产生，因为在 RSA 算法中密钥的产生要用到随机数的产生方法。此外，本书后面将要介绍的其他基于密码学的安全算法和协议也要用到随机数，例如：鉴别过程中，避免重放攻击的非重复值、用于一次性使用的会话密钥。

随机数发生器已经嵌入在大多数的编译器中了，这种随机数发生器产生的随机数可以用在密码中吗？这种随机数发生器对密码来说几乎肯定是不安全的，因为这种随机数发生器并不是完全随机的。之所以在广泛使用这样的伪随机数发生器，是因为大多数的简单应用并不需要真随机数。在计算机上是不可能产生真正的随机数的，因为它只能处于有限的状态，尽管这个数很大，其输出是输入和计算机当前状态的函数。也就是说任何计算机上的随机数发生器都是周期的。著名的计算机科学家冯•诺依曼对此有一个结论，任何人考虑用数学方法产生随机数肯定是不合情理的。

对于随机数序列可以分为如下几种：

伪随机数序列：看起来是随机的序列，它能通过现有的一些随机性统计检验，序列的周期足够长，使得实际应用中相当长的有限序列都不是周期性的。对于随机序列的随机性有两个评价标准，一个是分布的均匀性，即随机数的分布应是一致的，出现频率大致相等；另一个是独立性，即序列中的任何数不能由其他数推导出。分布均匀性检验我们使用χ^2 拟合优度检验法；独立性常用游程检验法检验。密码算法中大量使用了这种似乎随机的随机数序列，如 RSA 算法中的密钥产生过程。

密码学意义上安全的伪随机序列：除了随机性，它还应该是不可预测的。在相互鉴别或会话密钥生成之类的应用中，对随机数的统计随机性要求并不很高，但是要求产生的随机数序列是不可预测的，即使给出序列的算法或硬件以及该序列所有前面的位，也很难预测下一个随机位。

真随机序列：如果序列除了具有与随机序列相同的统计特性，它还具有不能重复产生的特性，那么它就是真正随机的了。如果用完全相同的输入对序列发生器操作两次，你将得到两个不相关的随机序列。

3.8.1　真随机序列产生器

真随机数只能来源于自然界，下面将给出几种真随机序列的例子。

3.8.1.1　RAND 表

1955 年，Rand 公司出版了一本包括 100 万个随机数的书。书中的随机数通过电子转轮产生的基本表再进行随机化获得。随机数字表，以 5 位数字组的形式列出，“10097 32533 76520 …”一行 50 个数字，一页 50 行。行编号从 00000 到 19999。使用该表时首先应寻找一个随机起始位置。一种办法是随机翻到一页，随机选取一个 5 位数，用个位数字模 2 来决定起始行，用右边两位数字模50 来确定起始行中的起始列。每一个被确定起始位置的数字应做上标记，避免重复使用。

3.8.1.2　使用随机噪声

采集大量随机数的最好办法是利用现实世界的自然随机性，这种方法通常需要一个特定的硬件，但仍需要一定的计算机技巧。例如：寻找一个有规律但又随机的事件：如超过某一门限值的大气噪声。测量并记录相邻时间的间隔，如果第一个时间间隔大于第二个时间间隔，输出 1；如果第一个时间间隔小于第二个时间间隔，输出 0。然后重复。

3.8.1.3　使用计算机时钟

如果想要一个单独的随机位，可以取任意一个时钟寄存器的最低位。UNIX 系统中由于存在各种同步机制可能使时钟寄存器的最低位不随机，但这在某些个人计算机上是可以实现的。

这种方法不适于取很多位。例如，如果每个子程序在一个奇数时钟执行，则会得到一个无休止的与上一个程序各位相反的序列，这样一来，结果就不是随机的了。

3.8.1.4 测量键盘反应时间

人们的打字方式有随机和非随机的，非随机方式可用做身份识别。测量两次击键的时间间隔，然后取这些测量的最低有效位。这一技术在 UNIX 终端上不可行，因为击键信息要经过过滤和其他机制方能进入程序，但在个人计算机上是可行的。因此，该技术有一定局限性。

3.8.2 伪随机数产生器

目前计算机所能生成的是伪随机序列发生器(Pseudo Random Sequence Generator)。它看起来是随机的，实际却是周期的，只不过周期足够长，使实际使用的那个序列不是周期的。应该根据所需要的随机序列的长度选择随机序列发生器。评价随机数产生器的一般准则如下：

- 这个函数应该是一个完整周期的产生函数，这个函数在重复之前产生出 0 到 m 之间的所有数。
- 产生的序列应该看起来是随机的。
- 这个函数应该用 32 位算术高效实现。

3.8.2.1 线性同余发生器(Linear Congruential Generator)

到现在为止，使用最广泛的产生伪随机数的方法是Lehmer首先提出的算法，即线性同余法。随机数序列$\{X_n\}$按下面的迭代方程获得：

$$X_{n+1} = (aX_n+c) \bmod m$$

算法中的参数如下：

m	模数	$m > 0$
a	乘数	$0 < a < m$
c	增量	$0 \leqslant c < m$
X_0	种子	$0 \leqslant X_0 < m$

每个随机数满足 $0 \leqslant X_n < m$。m，a，c 的选择对于算法非常关键。m 一般很大，如与给定的计算机能表示的最大整数接近，对于 32 位计算机，$m = 2^{31}-1$，这时取 $c = 0$，a 的取值有很多，但能满足上述准则的只是其中一部分，$a = 7^5 = 16\ 807$ 时满足上述条件，并经过广泛的测试和检验。选定上述参数，改变种子 X_0 就能产生不同的伪随机数序列。这样产生的伪随机数速度快，但缺乏不可预测性。

3.8.2.2 BBS 伪随机数产生器

BBS发生器是现在产生安全伪随机数的普遍方法，它的产生过程如下：首先选择两个大素数 p 和 q，且要求 $p \equiv q \equiv 3 \pmod 4$，接着选择 s 与 n 互素，$n = p \times q$，然后按下列算法产生随机数序列

$$X_0 = s^2 \bmod n$$
$$\text{for} \quad i = 1 \text{ to } \infty$$
$$X_i = (X_{i-1})^2 \bmod n$$
$$B_i = X_i \bmod 2$$

BBS 被称为密码安全伪随机数发生器，它通过了“下一位”测试(next-bit test)，即不存

在多项式时间的算法使得在已知前 k 位的情况下预测出第 k+1 位的概率大于 0.5，BBS 的安全性基于分解 n 的难度。

3.8.3　基于密码编码方法的随机数

3.7 节给出了分组加密的流密码工作模式，在 OFB 和 CTR 模式中，分组加密的输出被作为密钥流与明文进行异或，连续的 b 位输出构成了有着良好统计性质的伪随机序列。

思考和练习题

(1) 为什么使用表 3.1 所示的任意可逆代替密码不实际？
(2) 什么是乘积密码？是否两个相对简单的密码(如代替和换位)的积可将安全提高到较高层次？解释其原因。
(3) 对称分组密码设计的主要指导原则有哪些？实现的手段是什么？
(4) 什么是对称分组密码的雪崩效应?
(5) 对对称分组密码的攻击方式有哪些？
(6) DES 算法中，S 盒和 P 盒的作用是什么？
(7) 对称分组密码的工作模式主要有哪几种，各有什么优缺点？
(8) 什么是 DES 的补密钥？其特点对选择明文攻击有什么影响？
(9) 解释对双重 DES 的中间相遇攻击。
(10) 三重 DES 采用 EDE 的模式有什么好处？
(11) 当明文长度不是分组长度整数倍的时候，进行对称分组加密要进行什么处理？如果明文长度恰好是分组长度整数倍时，需要进行处理吗？
(12) 比较三重 DES 内部 CBC 和外部 CBC 工作模式的异同。
(13) 为什么流密码的密钥不能重复使用？
(14) 密码学上安全的随机数具有什么特性？

实践/实验题

(1) 实现 DES，要求：C 和/或 C++语言，5 种模式(ECB, CBC, CFB, OFB 和 CTR)的加解密，程序可以输入明文文件，输出密文文件，允许选择加解密模式，并进行加解密正确性测试及性能测试。
(2) 实现 AES，要求：C 和/或 C++语言，5 种模式(ECB, CBC, CFB, OFB 和 CTR)的加解密，程序可以输入明文文件，输出密文文件，允许选择加解密模式，并进行加解密正确性测试及性能测试。

第4章　公 钥 密 码

对称算法具有安全性高、加解密速度快的优点，在许多领域得到了广泛的应用，但是它也存在一些缺点：

- 由于系统的保密性主要取决于密钥的安全性，在进行保密通信前，双方必须通过安全的信道传送所用密钥，对于相距较远的用户可能要付出巨大的代价，甚至难以实现。
- 随着网络规模的扩大，密钥的管理成为一个难点。假设有 n 方参与通信，若 n 方都采用同一个对称密钥，一旦密钥被破解，整个体系就会崩溃；若两两分别采用不同的对称密钥，则需要 $n(n-1)/2$ 个密钥，密钥数与参与通信人数的平方成正比，使大系统的密钥管理极为困难。
- 无法解决对消息的窜改、否认等问题。

公钥算法则完全克服了上述缺点。公钥加密算法的出现是现代密码学诞生的标志之一，其加密和解密过程使用不同的密钥，而且，由解密密钥很容易计算出加密密钥，而由加密密钥很难甚至无法计算出解密密钥，加密密钥公开，任何人均可使用加密密钥来加密消息，但只有拥有解密密钥的人才能解密消息。比较著名的公钥密码算法有：RSA、背包密码、McEliece 密码、Rabin、椭圆曲线、EIGamal 算法以及我国学者提出的有限自动机密码。

4.1　公钥密码体制的基本原理

4.1.1　公钥密码体制的概念

1976 年由 Diffie 和 Hellman 在其“密码学新方向”一文中提出了不对称密钥密码的思想，首次证明发送端和接收端无密钥传输的保密通信是可能的，公钥算法具有以下特点：

- 加密与解密由不同的密钥完成，其中 K_U 是公开的，称为**公开密钥**(public key)，K_R 是保密的，称为**私有密钥**(private key)。
  加密：$X \rightarrow Y$: $Y = E_{KU}(X)$
  解密：$Y \rightarrow X$: $X = D_{KR}(Y) = D_{KR}(E_{KU}(X))$
- 知道加密算法，从加密密钥得到解密密钥在计算上是不可行的。
- 两个密钥中任何一个都可以用做加密而另一个用做解密，这一条不是必需的。

$$X = D_{KR}(E_{KU}(X)) = E_{KU}(D_{KR}(X))$$

公钥密码算法可以用于加密、数字签名(身份鉴别)和密钥交换。基于公钥密码算法的加解密过程如图 4.1 所示。其主要步骤如下：

(1) 每一个用户拥有自己的密钥对(K_U，K_R)。

(2) 每一个用户把公钥 K_U 公开，私钥 K_R 保密，每一用户可以拥有其他用户的公钥。

(3) 若 A 要给 B 发送消息，则用 B 的公钥 K_{Ub} 对消息 X 加密，A→B: $Y=E_{KUb}(X)$。

(4) 当 B 收到密文 Y 后，用私钥 K_{Rb} 解密，$D_{KRb}(Y)=D_{KRb}(E_{KUb}(X))=X$，由于只有 B 拥有自己的私钥 K_{Rb}，所有其他接收者都不能解密消息。

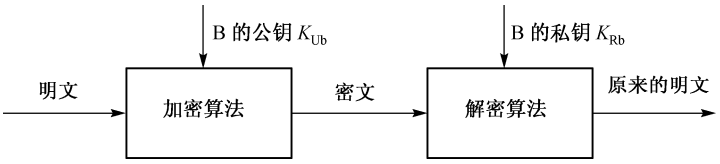


图 4.1　基于公钥密码算法的加解密过程

用公钥密码实现鉴别的过程如图 4.2 所示，主要步骤如下：

(1) 用户 A 用自己的私钥对消息 X 签名，A→ALL: $Y=E_{KRa}(X)$。

(2) 收到签名 Y 的其他人都可以用 A 的公钥 K_{Ua} 对消息的签名进行验证，$D_{KUa}(Y)=D_{KUa}(E_{KRa}(X))=X$，因为只有 A 才拥有能产生 K_{Ua} 能验证的签名的私钥 K_{Ra}，如果验证正确的话，就可以相信消息的确是 A 发出的。

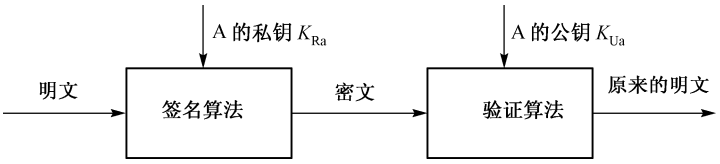


图 4.2　基于公钥密码算法的鉴别过程

如果两次使用公钥密码算法，则可以既实现鉴别，又保证消息的保密性，其主要步骤如下：

(1) 发送方 A 先用自己的私钥 K_{Ra} 对消息进行签名，然后再用接收方 B 的公钥 K_{Ub} 加密，A→B: $Z=E_{KUb}(D_{KRa}(X))$。

(2) 接收方 B 收到密文，先用自己的私钥 K_{Rb} 解密，再用对方的公钥 K_{Ua} 验证签名的正确性，$E_{KUa}(D_{KRb}(Z))=X$。

4.1.2　公钥密码体制的应用

一般地，公钥密码体制的应用可以分为三类：

- **加解密**：发送方用接收方的公钥加密，接收方用自己的私钥解密。
- **数字签名**：发送方用自己的私钥对消息签名，接收方用发送方的公钥验证签名。我们将在第 5 章讨论数字签名的特性和相关问题。
- **密钥交换**：通信双方交换会话密钥。有多种方法可用于密钥交换。

有些算法可用于上述三种应用，而一些算法只适用于其中的一两种应用。表 4.1 列出了一些典型的公钥算法及其应用。

表 4.1　公钥密码体制的应用

算　法	加/解密	数 字 签 名	密 钥 交 换
RSA，ECC	是	是	是
Diffie-Hellman 密钥交换	否	否	是
DSS	否	是	否

4.1.3 公钥密码体制的思想和要求

公钥密码体制涉及到三方：发送方、接收方、攻击者，涉及到数据有：公钥、私钥、明文和密文。Diffie和Hellman并没有给出具体的公钥密码体制，但对这种体制提出了一些要求。公钥算法的基本要求包含安全性和计算可行性两方面，分述如下：

- 用户 B 产生一对密钥(公钥 K_{Ub}，私钥 K_{Rb})是计算可行的。
- 已知 B 的公钥 K_{Ub} 和明文 X，发送方 A 产生密文 Y 是计算可行的。
- 接收方 B 利用其私钥 K_{Rb} 来解密密文 Y 是计算可行的。
- 对于攻击者，利用 B 的公钥 K_{Ub} 来推断其私钥 K_{Rb} 是计算不可行的。
- 已知公钥 K_{Ub} 和密文 Y，恢复明文 X 是计算不可行的。
- 加密和解密的顺序可交换，这一条对于算法不是必需的。

从数学上可以使用单向陷门函数来表示以上要求，单向陷门函数是满足下列条件的函数 f：

(1)给定 x，计算 $y=f_k(x)$ 是容易的。

(2)给定 y，计算 x 使 $x=f_k^{-1}(y)$ 是不可行的。

(3)存在 k，已知 k 时，对给定的任何 y，若相应的 x 存在，则计算 $x=f_k^{-1}(y)$ 是容易的。

通常，容易是指一个问题可以在输入长度的多项式时间内得到解决。不可行的定义比较模糊，一般而言，若解决一个问题所需要的时间比输入规模的多项式时间增长更快，则称问题是不可行的。

而严格单向函数是不能用于加解密的，一个单射函数 $f: X \rightarrow Y$ 称为是**严格单向函数**，如果下述条件成立：存在一个有效的方法，对所有的 $x \in X$ 可计算 $f(x)$，但不存在一个有效的办法，对所有的 $y \in Y$ 由 $y=f(x)$ 计算 x。现在还无法证明严格单向函数的存在。

4.1.4 公钥密码体制的安全性

4.1.4.1 公钥密码体制的安全目标和攻击类型

公钥密码体制用于加密时，其安全性按照目标可以分为三种：

- **单向性**(One-way, OW)：指没有密钥时密文不能恢复相应的明文，这是一个加密机制最基本的要求。
- **不可区分性**(Indistinguish ability, IND)：指给定的两个明文，加密者随机地选择其中一个进行加密，攻击者无法从密文中知道是对哪个明文的加密。
- **非延展性**(Non-malleability，NM)：指给定一个密文，攻击者的目的是构造与已给密文相关的新密文，而安全性要求攻击者无法构造与已给密文相关的新密文。

以上安全性概念依次加强，NM 比 IND 安全性强，IND 比 OW 安全性强。

按照攻击者模型可分为：

- **选择明文**(Chosen plain text attack, CPA)**攻击**：攻击者可以适当地选择明文，获得相应的密文。在公钥密码中，攻击者拥有公钥，可以随便加密，进而实现选择明文攻击。为了抵抗选择明文攻击，可以通过对报文附加随机比特来实现。
- **明文校验攻击**(Plain text check attack, PCA)：攻击者可以对一个明文密文对 (m, c)，回答 c 是否由 m 加密得来。

- **选择密文攻击**(Chosen cipher text attack，CCA)：攻击者可以选择密文获得相应的解密。选择密文攻击可以进一步分为非适应性和适应性两种。适应性选择密文攻击强度最高。

公钥密码和对称方案一样，进行强力攻击是计算上不可行的。公钥密码体制的安全性基于一些陷门单向函数，只是计算上不可行，要求使用非常大的数，通常要大于 512 位，因此比对称方案计算速度慢。由于公钥密码体制的安全性是指计算上的安全性，安全性的理论基础是复杂性理论。关于复杂性理论，我们给出以下一些概念的简单介绍。

4.1.4.2　算法复杂性

问题是需要回答的一般性提问，通常含有若干个未定参数或自由变量。对问题的描述包括两部分，一部分是给定所有的未定参数的一般性描述，另一部分是陈述“答案”或“解”必须满足的性质。

算法是求解某个问题的一系列具体步骤，通常可理解为求解某个问题的通用计算机程序。一个算法的复杂性由该算法所需求的最大时间(T)和存储空间(S)度量。由于算法用于同一问题的不同规模实例所需求的时间和空间往往不同，所以总是将它们表示成问题实例的规模 n 的函数，n 表示描述该实例所需的输入数据长度。

算法复杂性用“大O”的符号来表示，它表示算法复杂性的数量级。$f(n)=O(g(n))$意味着存在常数 c 和 n_0，使得对一切$n\geq n_0$，有 $f(n)\leq c|g(n)|$。通常按时间(或空间)复杂性对算法进行分类。一个输入数据规模为 n 的算法被称为是：

- **线性的**：如果运行时间是 $O(n)$。
- **多项式的**：如果运行时间是 $O(n^t)$，其中 t 是一个常数。
- **指数的**：如果运行时间是 $O(t^{h(n)})$，其中 t 是一个常数，$h(n)$是一个多项式。

一般地，一个可以在多项式时间内解决的问题被认为是可解的，而需要比多项式时间更长的时间，尤其是指数时间求解的问题在实践中往往被认为是不可解的。

4.1.4.3　问题复杂性

问题复杂性理论利用算法复杂性理论作为工具，将大量典型的问题按求解代价进行分类。

图灵机是一种具有无限读、写能力的有限状态机。图灵机分为确定型和非确定型两种。确定型是指图灵机的每一步操作的结果是唯一确定的。所谓非确定型图灵机的每一步操作结果及下一步操作都允许有多种选择，不是唯一确定的。

一个问题的最坏情况下的复杂性由在图灵机上解此问题的最难实例所需要的最小时间与空间决定，即解此问题的最有效算法所需的时间与空间来度量。在确定型图灵机上用多项式时间可解的问题，称为**易处理问题**。易处理判定问题的全体称为**确定型多项式时间可解判定问题类**，记为 P。在非确定型图灵机上用多项式时间可解的判定问题，称为**非确定型多项式时间可解判定问题**，简称 NP 问题。NP 问题的全体称为**非确定性多项式时间可解判定问题类**，记为 NP。对于 NP 问题的求解一般分为猜测和验证阶段。目前还没有人证明 P≠NP。背包问题(knapsack)，又称子集和问题(subset sum)，属于 NP 问题。

4.2　公钥密码算法的数学基础

数论中的一些定理和概念在设计公钥算法时是必不可少的，所以，在介绍公钥密码前，先回顾一下初等数论的一些定理和概念。

4.2.1 若干基本定理

4.2.1.1 算术基本定理

任意整数 $a>1$ 都可以唯一地因子分解为：

$$a = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_t^{\alpha_t}$$

其中 $p_1 > p_2 > \cdots > p_i$ 都是素数而且每一个 $\alpha_i > 0 \quad (i = 1, 2, 3, \cdots)$。

4.2.1.2 中国剩余定理(CRT)

《孙子算经》是我国南北朝时期一部重要的数学著作，书中有一个“物不知数”问题，也称为“孙子问题”：今有物，不知其数， 三三数之剩二，五五数之剩三，七七数之剩二，问物几何？这就是要求解同余方程组：

$$\begin{cases} x \equiv 2 \bmod 3 \\ x \equiv 3 \bmod 5 \\ x \equiv 2 \bmod 7 \end{cases}$$

的正整数解。

明代著名的大数学家程大位，在他所著的《算法统宗》中，对于“孙子问题”的解题方法，还编出了四句歌诀：三人同行七十稀，五树梅花廿一枝，七子团圆正半月，除百零五便得知。意思是：一个数用 3 除，除得的余数乘 70；用 5 除，除得的余数乘 21；用 7 除，除得的余数乘 15。最后把这些乘积加起来再减去 105 的倍数，就知道这个数是多少。运用这一歌诀来解答这道“物不知数”问题，便是

$$70 \times 2 + 21 \times 3 + 15 \times 2 = 233$$
$$233 - 105 - 105 = 23$$

所以这些物品最少有 23 个。不过，这一歌诀只能解答用 3、5、7 作除数的题目。

“孙子定理”给出这类题目的一般解答形式，国际上称为中国剩余定理(Chinese Remainder Theory，CRT)，其实质是给出了一种求解线性同余方程组的方法，是数论中最重要的定理之一。该定理说明某一范围内的整数可以通过它对两两互素的整数取模所得的余数来重构。

中国剩余定理有多种表示形式，其中一种常用表示形式为：设自然数 $m_1, m_2, \cdots, m_r$ 两两互素，并记 $N = m_1 m_2 \cdots m_r, N = m_i M_i, i = 1, 2, \cdots, r$，则同余方程组

$$\begin{cases} x \equiv b_1 (\bmod m_1) \\ x \equiv b_2 (\bmod m_2) \\ \cdots \\ \cdots \\ x \equiv b_r (\bmod m_r) \end{cases}$$

在模 N 同余的意义下有唯一解

$$x \equiv M_1' M_1 b_1 + M_2' M_2 b_2 + \cdots + M_r' M_r b_r (\bmod N)$$
$$\text{其中} M_i' M_i \equiv 1 (\bmod m_i), i = 1, 2 \cdots, r$$

证明：由于 $\gcd(m_i, m_j) = 1$，$(1 \leqslant i, j \leqslant r, i \neq j)$，得 $\gcd(M_i, m_i) = 1$，故对每一 M_i，有

一 M_i' 存在，使得

$$M_i'M_i \equiv 1(\bmod m_i),\ i=1,2\cdots,r$$

另外，$N=m_iM_i$，因此，$m_j|M_i$，$i\neq j$，故

$$M_1'M_1b_1+M_2'M_2b_2+\cdots+M_r'M_rb_r \equiv M_i'M_ib_i \equiv b_i(\bmod m_i)$$

为方程组的解。

若 x_1，x_2 是适合同余方程组的任意两个整数，则

$$x_1 \equiv x_2(\bmod m_i),\ i=1,2,\cdots,r$$

由于 $\gcd(m_i, m_j)=1$，于是，解在模 N 同余的意义下唯一。

4.2.1.3　Fermat 定理

Fermat 小定理：若 p 是素数，a 是正整数且不能被 p 整除，则

$$a^{p-1} \equiv 1 \bmod p$$

证明：考虑集合$\{1, 2, \cdots, p-1\}$，对每个数乘以 a，得到集合$\{a \bmod p, 2a \bmod p, \cdots, (p-1)a \bmod p\}$，对于 p，后者两两不同且都在 1 与 $p-1$ 之间，因此两个集合相同，于是：

$$\begin{aligned}(p-1)! &= 1\times 2\times \cdots \times (p-1)\\ &\equiv [(a \bmod p)\times(2a \bmod p)\times\cdots\times((p-1)a \bmod p)] \bmod p\\ &\equiv [a\times 2a\times\cdots\times(p-1)a] \bmod p\\ &\equiv [a^{p-1}\times(p-1)!] \bmod p\end{aligned}$$

注意到$(p-1)!$与 p 互素，因此定理成立。

推论：p 是素数，a 是任意正整数，则：$a^p \equiv a \bmod p$。

4.2.1.4　Euler 定理

Euler 函数 $\phi(n)$定义为小于 n 且与 n 互素的正整数个数。特别地，有如下几种情况：

- 若 n 是素数，$\phi(n)=n-1$。
- 若 n 的因子分解为 $n=\prod p_i^{\alpha_i}$，$a_i>0$，p_i 互不相同，则 $\phi(n)=\prod p_i^{\alpha_i}\times(1-1/p_i)=n(1-1/p_1)(1-1/p_2)\cdots(1-1/p_n)$，$p_1, p_2, \cdots, p_n$ 是 n 的素数因子。
- 若 $\gcd(m, n)=1$，则 $\phi(mn)=\phi(m)\phi(n)$，特别地，若 $p\neq q$ 且都是素数，$\phi(pq)=(p-1)(q-1)$。

例如，$20=2\times2\times5$，有两个素数 2 和 5，这样，$\phi(20)=20(1-1/2)(1-1/5)=8$，即 20 中有 8 个整数与 20 是互素的，即它们没有 2 或 5 为因子，这 8 个整数分别为：1, 3, 7, 9, 11, 13, 17, 19。

Euler 定理：若 a 与 n 为互素的正整数，则：$a^{\phi(n)} \equiv 1 \bmod n$。

证明：设 $R=\{x_1, x_2, \cdots, x_{\phi(n)}\}$为所有小于 n 且与 n 互素的正整数，考虑集合。

$S=\{(ax_1 \bmod n), (ax_2 \bmod n), \cdots, (ax_{\phi(n)} \bmod n)\}$，$(ax_i \bmod n)$与 n 互素，$(ax_i \bmod n)$两两不等，由$(ax_i \bmod n) = (ax_j \bmod n)$可以得出 $x_i \bmod n = x_j \bmod n$，$S$ 有 $\phi(n)$个元素，故 S 也是所有小于 n 且与 n 互素的正整数，因此 $S=R$，从而

$$\prod x_i = \prod(ax_i \bmod n) \equiv (\prod(ax_i)) \bmod n \equiv (a^{\phi(n)}\prod x_i) \bmod n$$

注意到 x_i 与 n 互素，从而得到结论。

推论：若 $n=pq$，$p\neq q$ 都是素数，k 是任意整数，则
$m^{k(p-1)(q-1)+1}\equiv m \bmod n$，对任意 $0\leq m\leq n$。

证明：若 $m=0$ 或 n，结论是显然的；若 m 与 n 互素，注意到 $\phi(n)=(p-1)(q-1)$，由 Euler 定理可得到结论；否则 m 必定是 p 或者 q 的倍数，不妨设 $m=sp$，则 $0<s<q$ 为正整数，m 与 q 互素，由 Euler 定理得到：

$$m^{\phi(q)}\equiv 1 \bmod q$$
$$(m^{\phi(q)})^{k\phi(p)}\equiv 1 \bmod q$$
$$m^{k(p-1)(q-1)}=tq+1，t\text{ 是整数}$$

等式两边乘以 $m=sp$，得到：

$$m^{k(p-1)(q-1)+1}=(tq+1)sp=tspq+sp\equiv m \bmod n$$
$$m^{k(p-1)(q-1)+1}\equiv m \bmod n，\text{对任意 } 0\leq m\leq n$$

这个推论对于证明 RSA 算法的有效性非常有用。

4.2.2 离散对数难题

4.2.2.1 有限循环群上的离散对数

对于普通的正实数，对数函数是指数函数的反函数。对模运算也是类似的。设 G 是一个阶为 p 的有限循环群，g 是它的生成元，则 G 的元素可表示为：$G=\{1, g, g^2, \cdots, g^{p-1}\}$，由此可见，对 G 的任何元素 y，一定存在某一个正整数 x，$0\leqslant x\leqslant p-1$，使得 $y=g^x \bmod p$，这里，称整数 x 是群 G 上元素 y 关于生成元 g 的**离散对数**。

离散对数难题(Discrete Logarithm Problem, DLP)是：在 G 上，对于方程 $y=g^x \bmod p$，已知 g, x, p，计算 y 是容易的，已知 y, g, p，计算 x 是困难的。

密码学中常用的是三个有限群上的离散对数，这三个有限群是：

- 有限域 GF(p)上的乘法群。
- 有限域 GF(2^n)上的乘法群。
- 有限域 F 上的椭圆曲线群。

4.2.2.2 有限域 GF(p)上的离散对数

设 p 是一个给定的素数，现考虑有限域 GF(p) = {0, 1, 2, …, p–1}上非零元组成的乘法群 $Z_p^*=\{1, 2, \cdots, p-1\}$。

由 Euler 定理，若 a 与 n 为互素的正整数，则 $a^{\phi(n)}\equiv 1 \bmod n$。考虑 $a^m\equiv 1 \bmod n$，满足上式的最小指数 m 称为 ***a*(mod *n*)的阶**，***a* 属于(mod *n*)的指数**，***a* 的生成周期长度**。如果最小的 $m=\phi(n)$，那么 a 被称为 n 的**原根**(Primitive root)。并不是所有整数都有原根，只有 2, 4, p^a, $2p^a$ 这样形式的整数才有原根，其中 p 为奇素数。

取 $p=13$，有限域 GF(13)={0, 1, 2, …, 12}上的非零元组成乘法群：$Z_{13}^*=\{1, 2, \cdots, 12\}$。这一乘法群是一个有限循环群。可以验证，元素 2, 6, 7, 11 中的每一个都能生成 Z_{13}^*，因此，这 3 个元素都可以作为它的生成元。这 4 个元素也是素数 13 的原根。表 4.2 给出了整数模 13 的幂，表中带阴影的部分表明了该行整数在计算模幂时重复出现的幂结果，也就是我们说的阶和生成周期长度。从表中可以看出，2 模 13 的阶为 12，而 12 模 13 的阶为 2。

如果 a 是素数 p 的原根，则数 $a \bmod p, a^2 \bmod p, \cdots, a^{p-1} \bmod p$ 是不同的并且包含 1 到 $p-1$ 的整数的某种排列，也即

$$\{a \bmod p, a^2 \bmod p, \cdots, a^{p-1} \bmod p\} = \{1, 2, \cdots, p-1\} = Z_p^*$$

因此，乘法群 Z_p^* 是一个有限循环群。

表 4.2　整数模 13 的幂

a	a^2	a^3	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}
1	1	1	1	1	1	1	1	1	1	1	1
2	4	8	3	6	12	11	9	5	10	7	1
3	9	1	3	9	1	3	9	1	3	9	1
4	3	12	10	1	4	3	12	10	1	4	3
5	12	8	1	5	12	8	1	5	12	8	1
6	10	8	9	2	12	7	3	5	4	11	1
7	10	5	9	11	12	6	3	8	4	2	1
8	12	5	1	8	12	5	1	8	12	5	1
9	3	1	9	3	1	9	3	1	9	3	1
10	9	12	3	4	1	10	9	12	3	4	1
11	4	5	3	7	12	2	9	8	10	6	1
12	1	12	1	12	1	12	1	12	1	12	1

在有限域 GF(p)的乘法群 Z_p^* 上的离散对数问题是，给定任意整数 b，求 x，使 $a^x = b \bmod p$。对任意的整数 b 和素数 p 的原根 a ，我们可以找到唯一的指数 x 满足：

$$b \equiv a^x \bmod p \qquad 0 \leqslant x \leqslant (p-1)$$

x 称为 b 以 $a(\bmod p)$ 为底数的指数(**离散对数**)，记为 $x = \log_a b \bmod p$，$\text{ind}\, a, p(b)$。

4.3　Diffie-Hellman 密钥交换算法

Diffie-Hellman密钥交换算法是第一个公钥方案，其安全性建立在有限域中计算离散对数的困难性的基础上。Diffie-Hellman密钥交换算法不能用于交换任意信息，它允许两个用户可以通过公开信道安全地建立一个秘密信息，用于后续的通信过程，该秘密信息仅为两个参与者知道。该算法由Diffie 和Hellman 于 1976 年公开提出，虽然现在知道早在 1970 年，英国政府通信总部(GCHQ)通信电子安全局(Communications-Electronics Security Group，CESG)的 James Ellis 已经秘密地提出了这一概念。该算法在一些商业产品中得到应用，在美国的专利于 1997 年 4 月 29 日到期。

4.3.1　对 Diffie-Hellman 密钥交换算法的描述

假定 A, B 双方选择素数 p 以及 p 的一个原根 a，算法的步骤如下：

(1)用户 A 选择一个随机数 $X_a < p$，计算 $Y_a = a^{Xa} \bmod p$。

(2)用户 B 选择一个随机数 $X_b < p$，计算 $Y_b = a^{Xb} \bmod p$。

(3)每一方保密 X 值，而将 Y 值交换给对方。

(4)用户 A 计算出 $K = Y_b^{Xa} \bmod p$。

(5)用户 B 计算出 $K = Y_a^{Xb} \bmod p$。

(6) 这两种计算的结果是相同的，最后双方获得一个共享密钥 $a^{XaXb} \bmod p$。素数 p 以及 p 的原根 a 可由一方选择后发给对方。

由于 X_a 和 X_b 是私有的，攻击者只能通过公开的 p，a，Y_a 和 Y_b 来进行攻击，这样他就必须先计算离散对数：$X_a = \text{ind}a, p(Y_a)$ 或 $X_b = \text{ind}a, p(Y_b)$，然后才能像用户 A 或 B 那样计算出密钥 K。

4.3.2 对 Diffie-Hellman 密钥交换的攻击

虽然攻击者试图计算出用户的私钥是困难的，但Diffie-Hellman密钥交换仍然可以受到一种叫做中间人攻击的攻击，如图4.3所示。中间人攻击的步骤如下：

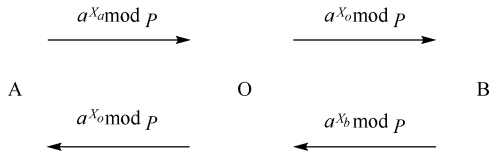


图 4.3 对 Diffie-Hellman 密钥交换的中间人攻击

(1) 双方选择素数 p 以及 p 的一个原根 a(假定 O 知道)。

(2) A 选择 $X_a<p$，计算出 $Y_a = a^{Xa} \bmod p$，并把 Y_a 发给 B。

(3) O 截获 Y_a，选择 X_o，计算出 $Y_o = a^{Xo} \bmod p$，冒充 A 把 Y_o 发给 B。

(4) B 选择 $X_b<p$，计算出 $Y_b = a^{Xb} \bmod p$，并把 Y_b 发给 A。

(5) O 截获 Y_b，冒充 B 把 Y_o 发给 A。

(6) A 进行如下计算，$(Y_o)^{Xa} \equiv (a^{Xo})^{Xa} \equiv a^{XoXa} \bmod p$。

(7) B 进行如下计算，$(Y_o)^{Xb} \equiv (a^{Xo})^{Xb} \equiv a^{XoXb} \bmod p$。

(8) O 完成如下计算，$(Y_a)^{Xo} \equiv a^{XaXo} \bmod p$，$(Y_b)^{Xo} \equiv a^{XbXo} \bmod p$。

在这种攻击方式中，O 无法计算出 A 和 B 期望获得的密钥 $a^{XaXb} \bmod p$，但最后 O 与 A 和 B 分别建立了共享密钥 $a^{XaXo} \bmod p$ 和 $a^{XbXo} \bmod p$，同时，O 永远必须实时截获并冒充转发，否则会被发现。

4.4 背包算法

第一个推广的公开密钥加密算法是 Ralph Merkle 和 Martin Hellman 开发的背包算法，它只能用于加密。背包算法的安全性基于背包问题，这是一个NP完全问题。尽管目前已经有攻击基本 Merkle-Hellman 体制的多项式时间算法，但该算法表示了如何将 NP 完全问题用于公开密钥算法，从这个意义上讲，它还是值得研究的。

4.4.1 背包问题和背包算法的思想

背包问题可以描述如下：给定重量分别为 $a_1, a_2, \cdots, a_n$ 的 n 个物品，装入一个背包中，要求重量等于一个给定值。那么，究竟是那些物品？这个问题可以形式化地描述为一个 0–1 背包问题：给定一个正整数 S 和一个背包向量 $\boldsymbol{A} = (a_1, \cdots, a_n)$，其中 a_i 是正整数，问是否存在满足方程 $S = \sum a_i x_i$ 的二进制向量 $\boldsymbol{X} = (x_1, \cdots, x_n)$？这是一个 NP 完全问题，因为对于给定的子集易于验证其和是否为 S。然而，找到一个子集使其和为 S 要困难得多，因为有 2^n 个可能的

子集，试验所有子集的时间复杂性为 $T=O(2^n)$，目前的算法解决这个问题所需要的时间与 n 呈指数增长。

背包问题用于公钥密码学的基本做法是假设明文为 X，S 为密文。奥妙在于有两类背包，一类可以在线性时间内求解，另一类则不能。容易的背包可以修改成难解的背包，公开密钥使用难解的背包，可以很容易地用来加密，但不能解密，私钥使用易解的背包，可以很容易地解密消息，不知道私钥的人只能求解难解的背包问题。

4.4.2　超递增背包

容易解的背包问题是：如果物品重量的列表为一个超递增序列，则此背包问题容易解决。超递增序列是满足下列特征的序列：其中每个元素都大于前面所有元素的和。我们把满足下列条件的背包 $a_i > \sum a_j\ (j = 1, 2, \cdots, i-1)$，也称为简单背包。超递增背包的求解过程为：从最大的 a_i 开始，如果 S 大于这个数，则减去 a_i，记 x_i 为 1，否则记 x_i 为 0，如此下去，直到最小的 a_i。

例如，背包序列{2, 3, 6, 13, 27, 52}，求解 70 的背包，结果为{2, 3, 13, 52}，所以，密文 70 对应的明文为 110101。

4.4.3　转换背包

我们把简单背包用做私钥，如何产生相应的公钥呢？具体的转换做法为：

选择一个整数 $m > \sum a_i\ (i = 1, 2, \cdots, n)$，然后选择一个与 m 互素的整数 w，进行计算 $a_i' = wa_i \pmod m\ (i = 1, 2, \cdots, n)$，这里的 a_i' 是伪随机分布的，这样得到的背包是非超递增背包，即一般的背包。一般的背包问题是一个困难的问题，没有已知的快速算法。要确定一个元素是否在背包中的一般方法是依次测试所有可能的解，直到找到正确的解。

4.4.4　Merkle-Hellman 公钥算法

Merkle-Hellman 公钥算法的加解密过程可以描述如下。

加密：

- 将明文分为长度为 n 的块 $X=(x_1, x_2, \cdots, x_n)$。
- 然后用公钥 $A'=(a_1', a_2', \cdots, a_n')$，将明文变为密文 $S=\mathrm{E}(X)=\sum_{i=1}^{n} a_i' x_i$。

解密：

- 先计算 $S' = w^{-1}S \bmod m$。
- 再求解简单背包问题 $S'=\sum_{i=1}^{n} a_i x_i$。

选择一个简单背包作为私钥，例如{2, 3, 6, 13, 27}。令 $w=31$，$m=105$，进行下列计算：

$$2 \times 31 \bmod 105 = 62$$
$$3 \times 31 \bmod 105 = 93$$
$$6 \times 31 \bmod 105 = 81$$
$$13 \times 31 \bmod 105 = 88$$
$$27 \times 31 \bmod 105 = 102$$

这样就得到一个一般背包{62, 93, 81, 88, 102}，我们将它作为公钥。

假设要加密一个二进制消息，首先将它分成长度与背包序列等长的块。1表示元素在背包中，0 表示不在，对每个明文，计算对应的密文。如消息为 01100 11010 10111，用前面的背包加密。

明文	0	1	1	0	0
背包	62	93	81	88	102

密文　93＋81＝174

01100 对应于 93＋81＝174

11010 对应于 62＋93＋88＝243

10111 对应于 62＋81＋88＋102＝333

则密文为 174, 243, 333。

消息的合法接收者知道私钥{2, 3, 6, 13, 27}与 w 和 m 的值。解密时，先计算 w^{-1}，满足 $w(w^{-1})=1 \bmod m$，可以使用扩展的 Euclid 算法，求得 $w^{-1}=61$。

然后进行解密计算

$174\times61 \bmod 105=9=3+6$，对应于 01100

$243\times61 \bmod 105=18=2+3+13$，对应于 11010

$333\times61 \bmod 105=48=2+6+13+27$，对应于 10111

因此，消息=01100 11010 10111。

一个只有 5 个元素的背包序列，即使不是超递增的，也不难解。实际的背包应该包括至少 250 个元素。超递增背包中的每项应该有 100 到 200 位长，模应该有 200 到 400 位长。实际实现时，用随机序列发生器来产生这些值。这样的背包才能抵抗强力攻击。

4.5 RSA 算法

继 Merkle-Hellman 背包算法之后，出现了第一个成熟的公开密钥算法 RSA。1977 年由 Ron Rivest, Adi Shamir 和 Len Adleman 发明，并于 1978 年公布。它是一种分组加密算法，明文和密文在 0～n–1 之间，n 是一个正整数。RSA 算法是应用最广泛的公钥密码算法，只在美国申请了专利，且已于 2000 年 9 月到期。

4.5.1 RSA 算法描述

4.5.1.1 设计 RSA 算法的思路

大素数相乘和因子分解可以被看成一个单向函数，也就是说根据大素数 p 和 q 计算出 $N=pq$，比已知 N，把 N 分解为 $N=pq$ 容易得多。我们是否能够基于这样一个观察构造一个公钥算法？

设 p 和 q 为两个大素数，$N=pq$，考虑这样的乘法群 $Z_N^*=Z_{pq}^*$，它包含 1 到 $pq-1$ 之间所有与 p 和 q 都互素的整数，那么这个群的大小为 $\phi(pq)=(p-1)(q-1)=N-(p+q)+1$，对于任意 $x\in Z_{pq}^*$，根据 Euler 定理有 $x^{(p-1)(q-1)}=1 \bmod N$。

因为有限域 GF(p)上的幂运算用软件实现的时候有高效的算法，我们希望用求幂运算来加密。令 e 是一个整数，$1<e<(p-1)(q-1)$，那么当 e 满足什么条件时，$m\to m^e$ 是乘法群 Z_{pq}^* 上的一一映射？答案是当 e 与 $(p-1)(q-1)$ 互素时，$m\to m^e$ 是乘法群 Z_{pq}^* 上的一一映射。因为 gcd

$(e,(p-1)(q-1))=1$，那么 e 存在一个模 $(p-1)(q-1)$ 的乘法逆元 d，$ed=1+k(p-1)(q-1)$。令 $c_i=m_i^e$

$$c_i^d=(m_i^e)^d=m_i^{ed}=m_i*m_i^{k(p-1)(q-1)}$$
$$=m_i*1(\bmod n)=m_i(\bmod n)$$

4.5.1.2　对 RSA 算法的描述

RSA 算法对明文以分组为单位进行加密，每个二进制分组的值均小于 n，也就是说分组的大小必须小于或等于 $\lceil \log_2 n \rceil$ 位，实际应用中，分组的大小为 k 位，$2^k<n\leqslant 2^{k+1}$，公开密钥为 $\{e,\ n\}$，其中 n 是两素数 p 和 q 的乘积，推荐 p 和 q 等长，e 为与 $(p-1)(q-1)$ 互素的数，私有密钥为 $\{d,\ p,\ q\}$，$d=e^{-1}\bmod(p-1)(q-1)$，对于明文分组 m 和密文分组 c，加解密过程如下。

加密：$c=m^e \bmod n$

解密：$m=c^d \bmod n=(m^e)^d \bmod n=m^{ed} \bmod n=m \bmod n$

RSA 算法的实现步骤如下：

(1) 用户 B 产生两个大素数 p 和 q，$p\neq q$；

(2) B 计算 $n=pq$，n 的 Euler 函数 $\phi(n)=(p-1)(q-1)$；

(3) B 选择随机数 e，$(0<e<\phi(n))$，使 $\gcd(e,\phi(n))=1$；

(4) B 使用扩展的 Euclid 算法计算 $d\equiv e^{-1} \bmod \phi(n)$；

(5) 选定算法的公钥：$K_U=\{e,n\}$，私钥：$K_R=\{d,p,q\}$。

选择好公私钥后，RSA 算法的加解密步骤如下：

将明文划分成块，使得每个明文报文 m 的长度 k 满足 $0<m<n$。

加密 m 时，计算 $c=m^e(\bmod n)$

解密 c 时，计算 $m=c^d(\bmod n)$

选择好公私钥后，RSA 算法的签名步骤如下：

将明文划分成块，使得每个明文报文 m 的长度 k 满足 $0<m<n$。

签名 m 时，计算 $y=\mathrm{Sig}(m)=m^d(\bmod n)$。

验证签名时，计算 $m'=y^e(\bmod n)$，判断其是否与原来的 m 相同。

下面通过一个例子说明用 RSA 算法加密与解密的过程。

(1) 选取素数 p 和 q，$p=47$，$q=61$，p 和 q 是秘密的。

(2) 计算 $n=pq=2867$，n 是公开的。

(3) $\phi(n)=(p-1)(q-1)=46\times 60=2760$，$\phi(n)$ 是秘密的。

(4) 选取公开密钥 e，使 $\gcd(e,\phi(n))=1$，$e=1223$ 是公开的。

(5) 计算秘密密钥 d，使 $ed\equiv 1 \bmod(\phi(n))$，$d=167$ 是秘密的。

这里，明文=“RSA ALGORITHM”，$n=2867$，$e=1223$，$d=167$，把明文用两位十进制数字表示，空白 =00，A=01，B=02，…，Z=26，再将要保护的明文信息分成一连串十进制数的数据块，每个数据块的值不超过 $n-1$，结果如下：

1819　0100　0112　0715　1809　2008　1300

利用加密变换公式 $C=m^e \bmod n$，分别对每一个数据块进行加密产生相应的密文块，如 $C=1819^{1223} \bmod 2867=2756$。

$e=1223=2^{10}+2^7+2^6+2^2+2^1+2^0=1024+128+64+4+2+1$，表示为二进制串就是 10011000111。

密文 $C=1819^{1223} \pmod{2867}$

$=1819^{1024} \cdot 1819^{128} \cdot 1819^{64} \cdot 1819^{4} \cdot 1819^{2} \cdot 1819^{1} \pmod{2867}$

$=2756$

类似地，可以得到整个明文对应的密文序列：

2756 2001 0542 0669 2347 0408 1815

4.5.2 RSA 实现中的问题

4.5.2.1 如何计算 $m^e \bmod n$

RSA 的加密和解密都要计算某整数的模 n 整数次幂，如果先计算出幂再对 n 取模，中间结果会非常大。利用模算术的下述性质进行模幂运算，可对中间结果取模，避免以上问题。

$$(a \times b) \bmod n = [(a \bmod n) \times (b \bmod n)] \bmod n$$

此外还应考虑到幂运算的效率，以 a^{16} 为例，若直接计算，需要进行15次乘法：$a^{16}=a \times a \times a \times a \times a \times a \times a \times a \times a \times a \times a \times a \times a \times a \times a \times a$

如果重复计算每个中间结果的平方，a^2, a^4, a^8, a^{16}，那么只需要 $4=\log_2 16$ 次乘法就可以计算出 a^{16}。

更一般地，“平方乘”算法是一种计算模幂的有效算法。如果要计算 m^e，将 e 的二进制表示为 $e_k e_{k-1} \cdots e_0$，则 $e=\sum_{e_i \neq 0} 2^i$，因此

$$m^e \bmod n = \left[\prod_{e_i \neq 0} m^{2^i} \right] \bmod n = \left(\prod_{e_i \neq 0} \left[m^{(2^i)} \bmod n \right] \right) \bmod n$$

这样就可以构造一个计算 $c = m^e \bmod n$ 的算法，伪代码如下：

```
c=1;
for i = k downto 0 do
    c = c^2 mod n
    if e_i = 1 then c = mc mod n
return c
```

4.5.2.2 密钥产生

要实现RSA算法，还存在一个比较难的问题，就是密钥的产生。这包括两个问题：如何找到足够大的素数 p 和 q？如何选择 e 或 d，并计算另外一个？

首先考虑如何挑选大素数 p 和 q。为了防止攻击者通过穷举方法找到 p 和 q，p 和 q 必须是足够大的素数。目前没有产生任意大素数的有用技术，通常的做法是随机选取一个需要数量级的奇数，并检验这个数是否是素数。直接判断一个整数是否为素数是困难的，传统使用试除法判断一个数是否是素数，即依次用比该数平方根小的素数进行除法运算，这种方法只对小数有操作性。还有一种办法是根据所有素数满足的特性进行统计素性检测，但是一些伪素数也满足此特性。

素性检测有很多种方法，一般是采用概率性的检验方法，即通过检验的奇数将以一定的概率是素数，当设置检验的次数足够多时，得到的“素数”不是真正的素数可能性会相当小。当然检验需要的时间也相应地变长。实际中应用最多的是 Miller-Rabin 算法。它基于这样一个数学命题：如果 p 是奇素数，则方程 $x^2 \equiv 1 \bmod p$ 只有两个解 $x \equiv \pm 1 \bmod p$。这是因为

$$x^2 \equiv 1 \bmod p \Rightarrow p|(x^2-1) \Rightarrow p|(x+1)(x-1) \Rightarrow p|(x+1)，或者\ p|(x-1)$$
$$\Rightarrow x+1=kp，或者\ x-1=jp, k, j\ 是整数$$
$$\Rightarrow x=kp-1，或者\ x=jp+1$$
$$\Rightarrow x \equiv 1 \bmod p，或者\ x \equiv -1 \bmod p$$

若方程 $x^2 \equiv 1 \bmod p$ 存在的解不是 $x \equiv \pm 1$，则 p 不是素数。

函数 WITNESS(a, n)判定 n 是否为素数，a 是某些小于 n 的整数，当返回值为TRUE表示 n 一定不是素数，返回值为 FALSE 表示 n 可能是素数。随机选择 $a<n$，计算 s 次，如果每次都返回 FALSE，则这时 n 是素数的概率为$(1-1/2^s)$。令 $b_k\ b_{k-1}\cdots b_0$ 为$(n-1)$的二进制表示，WITNESS(a, n)的伪代码如下：

```
d=1;
for i=k downto 0 do
    x=d;
    d=d^2 mod n
    if d=1 and x≠1 and x≠n-1 then return TRUE ; //(x^2? ≡ 1mod n)
    if b_i=1 then d=ad mod n;        //(a^(n-1) mod n)
if d≠1 then return TRUE;
return FALSE;
```

选取素数的一般过程如下：

(1) 随机选取一个奇素数 n。

(2) 随机选取一个整数 $a<n$。

(3) 执行概率素数判定测试[如：用 WITNESS(a, n)]。如果 n 没有通过检验，舍弃 n 并转到步骤 1。

(4) 如果 n 通过了足够多次的检验，接受 n，转到步骤 2。

Miller-Rabin 算法又称为“强伪素数检测”(Strong Pseudo-Prime test)，和其他大部分素性检测算法一样，它检验一个给定整数是否素数的过程，是涉及一个挑选出来的整数 n 和一个随机选取的整数 a 的计算过程，如果 n 没有通过这一次检验，那么 n 肯定不是素数，如果 n 通过了多次检验，那么有相当大的信心相信 n 是一个素数。随机选取大约需要用 $\ln n/2$ 的次数，如要找一个 2^{200} 数量级的素数，需要测试 $\ln 2^{200}/2=70$ 次。因为只是在生成密钥对时才用到，慢一点还可忍受。

确定素数 p 和 q 以后，只需选取 e，满足 $\gcd(e, \phi(n))=1$，使用扩展 Euclid 算法计算 $d=e^{-1} \bmod \phi(n)$。

4.5.3　RSA 的安全性

RSA 的安全性基于加密函数 $e_k(x)=x^e \pmod n$ 是一个单向函数，如图4.4 所示，所以对攻击的人来说求逆计算不可行。而 Bob 能解密的陷门是分解 $n=pq$，求得 $\phi(n)=(p-1)(q-1)$，从而用欧几里得算法解出解密私钥 d。攻破 RSA 与分解 n 是多项式等价的。然而，这个猜想至今没有给出可信的证明。

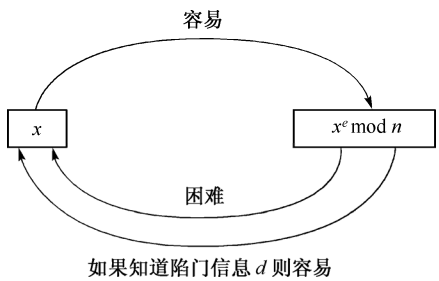


图 4.4　RSA 算法的单向陷门函数

对 RSA 的攻击可能有如下三种形式：

- 强力攻击(穷举法)：尝试所有可能的私有密钥。
- 数学分析攻击：有多种数学方法，实质都是试图分解两个素数的乘积。
- 对 RSA 实现的攻击。

若要使 RSA 安全，p 与 q 必为足够大的素数，使分析者没有办法在多项式时间内将 n 分解出来。建议选择 p 和 q 大约是 100 位的十进制素数。国际数字签名标准 ISO/IEC 9796 中规定 n 的长度为 512 比特。至 1996 年，建议使用 768 bit 的模 n。现在建议模 n 为是一个 1024 比特或更高比特的数。为了抵抗现有的整数分解算法，对 RSA 模 n 的素因子 p 和 q 还有如下要求：

(1) $|p-q|$很大，通常 p 和 q 的长度相同。

(2) $p-1$ 和 $q-1$ 分别含有大素因子 p_1 和 q_1。

(3) p_1-1 和 q_1-1 分别含有大素因子 p_2 和 q_2。

(4) $p+1$ 和 $q+1$ 分别含有大素因子 p_3 和 q_3。

满足以上条件的素数被称为强素数。

表 4.3 给出了因子分解的最新进展。开始人们使用十进制位数描述被分解的数，如表 4.3(a)中的 RSA-129 表示 129 位的十进制数，后来使用二进制位数描述被分解的数，如表 4.3(b)中的 RSA-640 表示二进制长度为 640 比特的数。

表 4.3 因子分解的最新进展

(a)

被分解的数(十进制位数)	分 解 时 间
RSA-100	Apr. 1991
RSA-110	Apr. 1992
RSA-120	Jun. 1993
RSA-129	Apr. 1994
RSA-130	Apr. 10, 1996
RSA-140	Feb. 2, 1999
RSA-150	Apr. 16, 2004
RSA-155	Aug. 22, 1999
RSA-160	Apr. 1, 2003
RSA-200	May 9, 2005

(b)

被分解的数(二进制位数)	分 解 时 间
RSA-576	Dec. 3, 2003
RSA-640	Nov. 4, 2005
RSA-704	open
RSA-768	open
RSA-896	open
RSA-1024	open
RSA-1536	open
RSA-2048	open

4.5.4 对 RSA 实现的攻击方法

对RSA的具体实现存在一些攻击方法，但不是针对基本算法的，而是针对协议的。主要有以下几种方式：对RSA的选择密文攻击、对RSA的公共模攻击、对RSA的小加密指数攻击、对RSA的小解密指数攻击和计时攻击。

4.5.4.1 对 RSA 的选择密文攻击

这里给出对 RSA 的选择密文攻击的三个例子。

例 1，E 监听 A 的通信，收集由 A 的公开密钥加密的密文 c，E 想知道消息的明文 m，使 $m=c^d \bmod n$。

它首先选择随机数 r，使 $r<n$，然后用 A 的公开密钥 e 计算：

$$x=r^e \bmod n$$
$$y=xc \bmod n$$
$$t=r^{-1} \bmod n$$

如果 $x=r^e \bmod n$，则 $r=x^d \bmod n$。现在 E 让 A 对 y 签名，即解密 y，A 向 E 发送 $u=y^d \bmod n$，而 E 计算：

$$tu \bmod n = r^{-1}y^d \bmod n = r^{-1}x^d c^d \bmod n = c^d \bmod n = m$$

则 E 得到了 m。

例 2，Mallory 选取任意一个数值 x，计算 $y=x^e \bmod n$。然后，他计算 $m=ym' \bmod n$，并将 m 发给 Trent，并让 Trent 对它签名。Trent 回送 $m^d \bmod n$，现在 Mallory 计算

$$(m^d \bmod \mathrm{n}) \cdot (x^{-1} \bmod n) = ((ym')^d \bmod n) \cdot (x^{-1} \bmod n) = (x^e m')^d \bmod n \cdot (x^{-1} \bmod n)$$

$$= (x^{ed}(m')^d) \bmod n \cdot (x^{-1} \bmod n) = x(m')^d x^{-1} \bmod n = (m')^d \bmod n$$

它等于对 m' 的签名。

例 3，E 想产生 A 对 m_3 签名。他产生两个消息 m_1 和 m_2，满足 $m_3=m_1m_2 \pmod n$，如果 E 能让 A 分别对 m_1 和 m_2 签名，则可以计算：

$$m_3{}^d = (m_1{}^d \bmod n)(m_2{}^d \bmod n)$$

注意，不要用RSA对陌生人的随机文件签名，签名前先使用一个散列函数，就可以防止这种攻击。

4.5.4.2　对 RSA 的公共模攻击

一种可能的 RSA 实现方法是给每个人相同的 n，但指数 d 和 e 不同。这导致的问题是，如果相同的消息曾用两个不同的指数加密，而这两个指数是互素的，则明文可以不用任何一个解密密钥来恢复。

令 m 为明文消息，两个加密密钥为 e_1, e_2，两个密文消息为 c_1, c_2

$$c_1 = m^{e_1} \bmod n$$
$$c_2 = m^{e_2} \bmod n$$

由于 e_1 和 e_2 互素，所以可以用扩展 Euclid 算法找到 r，s 使 $re_1+se_2=1$，这样，$c_1^r \times c_2^s \bmod n = (m^{e_1})^r \times (m^{e_2})^s \bmod n = m^{re_1+se_2} \bmod n = m \bmod n$。

为了防止这种攻击，注意，不要让一群用户共享一个模 n。

4.5.4.3　对 RSA 的小加/解密指数攻击

为了提高加密速度，通常取 e 为特定的小整数，如 EDI 国际标准中规定 $e = 2^{16}+1$，ISO/IEC9796 中甚至允许取 $e=3$。这时加密速度一般比解密速度快 10 倍以上。$e=2^{16}+1$ 优于 $e=3$ 之处在于它能够抵抗对 RSA 的小加密指数攻击。

使用一个较小的 e 值，进行 RSA 加密会很快，但也不安全。如果用相同 e 值的不同公开密钥加密 $e(e+1)/2$ 个线性相关的消息，则系统是可破的。如果有少于这些的消息或消息不相关，则无问题。比如，消息为 m_j，使用同样的指数 e，模数分别为 $q_1, q_2, \cdots, q_s$(两两互素)，则密文为 $m_j^e \bmod q_1$，$m_j^e \bmod q_2$，$\cdots$，$m_j^e \bmod q_s$，根据中国剩余定理，$m'=m_j^e \bmod q_1 q_2 \cdots q_s$ 可以计算出来，对于较小的 e，可以解出 m_j。

例如，对同一消息 m，使用相同的小加密指数 $e=3$ 加密，同时发给三个的不同的人，这三个人的模数分别为 q_1，q_2，q_3，则有：

$$c_1 = m^3 \bmod q_1$$
$$c_2 = m^3 \bmod q_2$$
$$c_3 = m^3 \bmod q_3$$

利用中国剩余定理可以解下列方程组：

$$x \equiv c_1 \bmod q_1$$
$$x \equiv c_2 \bmod q_2$$
$$x \equiv c_3 \bmod q_3$$

得到 $x=m^3$，再通过求 x 的三次方根可得到 m。

对于这一攻击的解决办法是加密前将消息与随机值混合，并保证 m 与 n 有相同的长度。此外，使用较小的 d 会产生穷尽解密攻击的可能，注意，应选择一个大的 d 值。

4.5.4.4 计时攻击

计时攻击是一种全新的攻击手段，属于选择密文的攻击，也适用于攻击其他公钥算法，发明于 20 世纪 90 年代，是基于加密程序运行时间的攻击。在 RSA 的加解密过程中涉及到计算 $m=c^d$，$d=d_k d_{k-1}\cdots d_0$（二进制表示）。

$m = 1$；
for $i = k$ downto 0 do
  $m = m^2 \bmod n$
  if $d_i = 1$ then $m = cm \bmod n$
return m

观察到若 $d_i=1$，执行 $m=cm \bmod n$，否则不执行，有少数的 c 和 m 值，上述模乘速度很慢，若攻击者观察到解密算法的执行速度很慢，则可以认为该位是 1，否则是 0，从左到右逐个确定 d_i。

对抗计时攻击的对策有三种，第一种是使用恒定的幂运算时间，虽然简单，但会降低算法性能。第二种是加一个随机延迟，但是仍然可被攻击者攻击成功。第三种办法称为盲化，在执行幂运算之前先将密文乘上一个随机数。

盲化的 $m=c^d \bmod n$ 的实现如下：

(1) 产生一个 0 到 $n-1$ 之间的随机数 r。

(2) 计算 $c'=c(re) \bmod n$，其中 e 是公开指数。

(3) 用通常的 RSA 实现计算 $m'=(c')^d \bmod n$。

(4) 计算 $m=m' r^{-1} \bmod n$。

可以证明结果是正确的，盲化操作增加的开销是 2% ～10%。

4.6 ElGamal 算法

ElGamal算法既可以用于加密，也可以用于签名，其安全性依赖于有限域上计算离散对数的难度。要产生一对密钥，首先选择一素数 p，整数模 p 的一个原根 g，随机选取 x，g 和 x 都小于 p，然后计算：

$$y=g^x \bmod p$$

公开密钥是 y，g，p，其中 g，p 可以为一组用户共享，私有密钥是 x。

将明文信息 M 表示成$\{0, 1,\cdots, p-1\}$范围内的数，加密时，秘密选择随机数 k，计算：

$$a=g^k \bmod p$$
$$b=y^k M \bmod p$$

(a, b)作为密文。注意，密文长度是明文的两倍，信息有扩张。

解密时，计算：

$$M = b/a^x \bmod p$$

$$a^x \equiv g^{kx} \bmod p,\ b/a^x \equiv y^k M/a^x \equiv g^{xk} M/g^{xk} \equiv M \bmod p$$

下面的例子说明 ElGamal 算法的加解密过程。

① **生成密钥**：使用者 Alice 选取素数 p =19 及 Z_{19}^*的生成元 g =2，Alice 选取私钥 x =10 并计算：

$$g^x \bmod p = 2^{10} \bmod 19 = 17$$

A 的公钥是 $p=19$，$g=2$，$g^x=17$

② **加密**：为加密消息 $m=16$，Bob 选取一个随机整数 $k=9$ 并计算：

$$a = 2^9 \bmod 19 = 18$$

$$b = 16 \times 17^9 \bmod 19 = 16$$

Bob 发送 a，b 给 Alice

③ **解密**：Alice 计算：

$$a^{-x} \equiv 18^{p-1-x} \equiv 18^8 \equiv 1 \pmod{19}$$

$$M \equiv b/a^x \equiv ba^{-x} \equiv 16 \times 1 \equiv 16 \pmod{19}$$

攻击 ElGamal 加密算法等价于解离散对数问题，要使用不同的随机数 k 加密不同的信息。假设用同一个 k 加密两个消息 m_1，m_2，所得到的密文分别为 (a_1, b_1) (a_2, b_2)，则 $b_1/b_2 = m_1/m_2$，故当 m_1 已知，m_2 可以很容易地计算出来。

4.7　椭圆曲线密码算法(ECC)

数学家对椭圆曲线的研究始于 19 世纪中叶，积累了丰富而深厚的理论。1985 年，Neal Koblitz 和 Victor Miller 分别独立地提出了椭圆曲线密码体制。多数公钥密码如 RSA，Diffie-Hellman 算法等都使用非常大的数或多项式，给密钥和信息的存储和处理带来很大的运算量。椭圆曲线是一个代替，可以用更小的尺寸得到同样的安全性，密钥长度为 160 位的椭圆曲线密码系统可以具有与密钥长度为1024位的 ElGamal 或 RSA 密码系统相当的安全性。所以近年来，椭圆曲线密码系统在商业应用领域也开始得到越来越多的关注，ANSI, IEEE, ISO 和NIST 都制定了 ECC 标准草案。ElGamal 密码体制能够在任何离散对数难处理的有限群中实现，ECC 算法就是基于椭圆曲线群上的离散对数难题。

4.7.1　椭圆曲线的概念

域 K 上的椭圆曲线 E 由 Weierstrass 方程定义：

$$E:\ y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \tag{4.1}$$

其中 a_1，a_2，a_3，a_4，$a_6 \in K$ 且 $\Delta \neq 0$，Δ 是 E 的判别式，具体定义此处省略。密码学上通常使用如下简化形式的椭圆曲线:

$$y^2 \equiv x^3 + ax + b \tag{4.2}$$

其中 a，$b \in K$，曲线的判别式是 $\Delta = -16(4a^3 + 27b^2)$，$\Delta \neq 0$ 确保椭圆曲线是“光滑”的，即曲线的所有点都没有两个或两个以上不同的切线。图4.5 给出两个实数域上椭圆曲线的例子。满足

式(4.2)的所有点(x, y)和一个被称为无穷远点(point at infinity)或零点(zero point)的元素O组成点集$E(a, b)$。点集$E(a, b)$按照一定运算法则构成一个加法群，群上的加法规则可以用几何方法说明如下：

- 若曲线中的三点在一条直线上，则其和为O。
- O用做加法的单位元，因而$O=-O$，对于椭圆曲线上的任何一点P，$P+O=P$。
- 一条垂直的线与曲线相交于两个x坐标相同的点P_1, P_2，则$P_1+P_2+O=O$，于是$P_1=-P_2$。因而一个点的逆元是与其有着相同x坐标和相反的y坐标的点，如图4.5(b)所示。
- 要对具有不同x坐标的两个点Q与R进行相加，先在它们之间画一条直线并求出第三个交点P_1，容易看出这种交点是唯一的，除非这条直线在Q或者R处是曲线的切线，这时分别取$P_1=Q$或者$P_1=R$，那么注意到$Q+R+P_1=O$，因此有$Q+R=-P_1$，如图4.5(a)所示。
- 一个点Q的两倍是，找到它的切线与曲线的另一个交点S，于是$Q+Q=2Q=-S$。

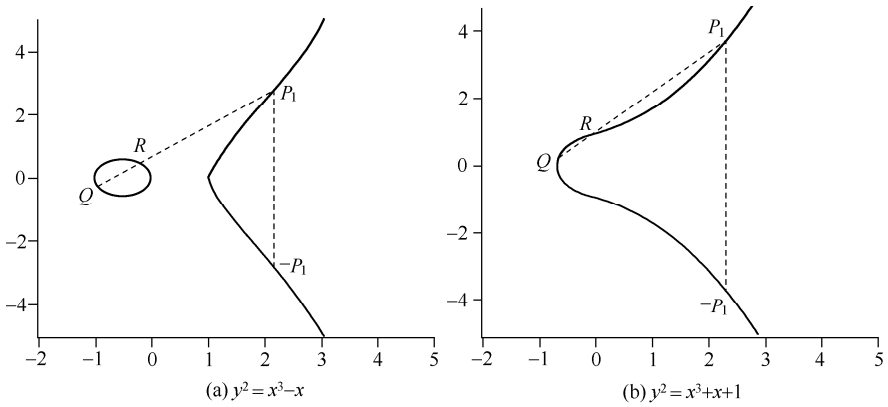


图 4.5 椭圆曲线的例子

显然，根据定义，前面定义的加法满足交换律和结合律。而一个点的倍乘定义为：

$$kP=\underbrace{P+P+P\cdots+P}_{k\text{个}P}$$

也称由P和k计算kP的运算为椭圆曲线多倍点运算(Point multiplication)或椭圆曲线标量乘法(Scalar multiplication)。

4.7.2 有限域上的椭圆曲线

密码学上关心的是一种受限形式的椭圆曲线，这种椭圆曲线定义在一个有限域上，不用实数域上的椭圆曲线做加密，原因是显而易见的，计算机本身是离散的，做实数运算会有舍入误差，而且可能溢出。椭圆曲线密码使用系数与变量定义在有限域的曲线，通常使用的有两类：定义在素数域GF(p)上的$E_p(a, b)$与定义在GF(2^n)上的二元曲线$E_{2^n}(a, b)$。讨论较多的是定义在素数域GF$(p)=\{0, 1, 2, \cdots, p-1\}$上的椭圆曲线，此时变元和系数均在素数域GF$(p)$中取值，定义如下

$$y^2\equiv x^3+ax+b \bmod p \tag{4.3}$$

其中p是奇素数，且$4a^3+27b^2\neq 0 \bmod p$。

满足以上方程的小于 p 的非负整数对 (x, y) 加上无穷远点 O 组成集合 $E_p(a, b)$。$E_p(a, b)$ 通过如下方式计算，针对所有的 $0 \leqslant x < p$，计算 $x^3 + ax + b \bmod p$，确定是否可以求出有效的 y，如果是有效的 y，则得到曲线上的点 (x, y)，其中 $x, y < p$。对于任意点 $P, Q \in E_p(a, b)$，有如下运算规则：

- $P + O = P$
- 如果 $P = (x, y)$，则 $P + (x, -y) = O$，$(x, -y)$ 点是 P 的逆元，记为 $-P$，并且 $-P$ 也在 $E_p(a, b)$ 中。
- 如果 $P = (x_1, y_1)$，$Q = (x_2, y_2)$，则 $P + Q = (x_3, y_3)$ 为
  $x_3 = (\lambda^2 - x_1 - x_2) \bmod p$
  $y_3 = (\lambda(x_1 - x_3) - y_1) \bmod p$
  其中，如果 $P \neq Q$，则 $\lambda = ((y_2 - y_1)/(x_2 - x_1)) \bmod p$
  如果 $P = Q$，则 $\lambda = ((3x_1^2 + a)/(2y_1)) \bmod p$
- 乘法的定义为重复相加。如 $3P = P + P + P$。

例如，取 $p = 29$，$a = 4$，$b = 20$，$4a^3 + 27b^2 = 11056 \neq 0 \bmod p$，定义在素数域 GF(29) 上的椭圆曲线 $y^2 = x^3 + 4x + 20$ 在 GF(29) 上的解为：

(0, 7)，(0, 22)，(1, 5)，(1, 24)，(2, 6)，(2, 23)，(3, 1)，(3, 28)，(4, 10)，(4, 19)，(5, 7)，(5, 22)，(6, 12)，(6, 17)，(8, 10)，(8, 19)，(10, 4)，(10, 25)，(13, 6)，(13, 23)，(14, 6)，(14, 23)，(15, 2)，(15, 27)，(16, 2)，(16, 27)，(17, 10)，(17, 19)，(19, 13)，(19, 16)，(20, 3)，(20, 26)，(24, 7)，(24, 22)，(27, 2)，(27, 27)

这些点加上无穷远点 O 组成一个群 $E_{29}(4, 20)$。

在 $E_{29}(4, 20)$ 中，取点 $P_1 = (x_1, y_1) = (2, 6)$，有 $-P_1 = (x_1, -y_1) = (2, -6)$，而 $-6 \bmod 29 = 23$，因此 $-P_1 = (2, 23)$，该点也在 $E_{29}(4, 20)$ 中。

取点 $P_1 = (x_1, y_1) = (2,6)$，$P_2 = (x_2, y_2) = (4, 10)$，$P_1 + P_2 = (x_3, y_3)$ 可计算如下：

$$\lambda = ((y_2 - y_1)/(x_2 - x_1)) \bmod p = ((10 - 6)/(4 - 2)) \bmod 29 = (4/2) \bmod 29 = 2 \bmod 29$$
$$x_3 = (\lambda^2 - x_1 - x_2) \bmod p = (2^2 - 2 - 4) \bmod 29 = 27 \bmod 29$$
$$y_3 = (\lambda(x_1 - x_3) - y_1) \bmod p = (2 \times (2 - 27) - 6) \bmod 29 = 2 \bmod 29$$

因此，$P_1 + P_2 = (27, 2)$。

取点 $P_1 = (x_1, y_1) = (2, 6)$，$2P_1 = (x_3, y_3)$ 可计算如下：

$$\lambda = ((3x_1^2 + a)/(2y_1)) \bmod 29 = ((3 \times 2^2 + 4)/(2 \times 6)) \bmod 29$$
$$= (16/12) \bmod 29 = (4/3) \bmod 29 = 11 \bmod 29$$
$$x_3 = (\lambda^2 - x_1 - x_1) \bmod 29 = (11^2 - 2 - 2) \bmod 29 = 1 \bmod 29$$
$$y_3 = (\lambda(x_1 - x_3) - y_1) \bmod 29 = (11 \times (2 - 1) - 6) \bmod 29 = 5 \bmod 29$$

因此，$2P_1 = (1, 5)$。

4.7.3　椭圆曲线密码算法

4.7.3.1　椭圆曲线群上的离散对数难题

ECC 是基于椭圆曲线离散对数难题(Discrete logarithm problem on elliptic curve, ECDLP)的密码体制。我们对有限域 GF(p) 的乘法群上的离散对数难题(Discrete logarithm problem, DLP)

和 ECDLP 进行了一些比较，如表 4.4 所示，可以把 ECC 的加类比于模乘，ECC 的重复相加类比于模幂，定义与 GF(p) 的乘法群上离散对数类似的 ECDLP 难题：

定义于有限域上的椭圆曲线对于加法构成了一个群 $E_p(a, b)$，G 是 $E_p(a, b)$ 一个阶为素数 n 的点，令 $Q = kG$，其中 Q，G 属于 $E_p(a, b)$，$0 < k < n$，

给定 k，G，容易计算 Q

给定 Q，G，难以解出 k

实际应用中，k 的值非常大，从而使穷举攻击不可行。

表 4.4 DLP 和 ECDLP 比较

	DLP	ECDL
所用集合	GF(p) 的乘法群 Z_p^*	GF(p) 上的椭圆曲线群 E
基本运算	Z_p^* 上的乘法	E 上点的加法
主要运算	模幂运算	标量乘法
生成元	生成元 $g \in Z_p^*$	E 上基点 G
运算的模	素数 p	素数 p
群的阶	$p-1$	n
私钥	整数 $a \pmod p$	整数 $a \pmod p$
公钥	$w = g^a \in Z_p^*$	E 上的点 $W = aG$

4.7.3.2 基于 ECDLP 的 D-H 密钥交换

利用椭圆曲线可以进行类似的 D-H 密钥交换。首先选择一曲线 $E_p(a, b)$，其次，选择 $E_p(a, b)$ 中的元素 $G = (x_1, y_1)$，使得 G 的阶 n 是一个大素数，G 的阶是指满足 $nG = O$ 的最小 n 值。用户 A 和 B 之间的密钥交换过程如下：

(1) A 和 B 分别选取一个小于 n 的整数 $K_{RA} < n$ 和 $K_{RB} < n$ 作为私钥。

(2) 然后分别计算其公钥：$K_{UA} = K_{RA} \times G$，$K_{UB} = K_{RB} \times G$。

(3) A 和 B 分别秘密地计算其共享密钥：$K = K_{RA} \times K_{UB}$，$K = K_{RB} \times K_{UA}$。

两种计算结果是相同的，A 和 B 获得了同样的密钥 $K = K_{RA} \times k_{RB} \times G$。

取 $p = 211$，$E_p(0, -4)$，这等价于曲线 $y^2 = x^3 - 4$，$G = (2, 2)$，可以算得 $241G = O$。

A 的私钥是 $K_{RA} = 121$，因此 A 的公钥是 $K_{UA} = 121(2, 2) = (115, 48)$。B 的私钥是 $K_{RB} = 203$，因此 B 的公钥是 $K_{UB} = 203(2, 2) = (130, 203)$。它们共享的密钥是 $121(130, 203) = 203(115, 48) = (161, 169)$。

4.7.3.3 基于 ECDLP 的加解密

基于ECDLP的加解密方法有多种，这里介绍一种最简单的基于ECDLP的加解密方法。首先选择一曲线 $E_p(a, b)$，其次，选择 $E_p(a, b)$ 中的元素 G，使得 G 的阶 n 是一个大素数，G 的阶是指满足 $nG = O$ 的最小 n 值。用户 A 和 B 之间的加解密过程如下：

(1) **密钥计算**：用户 A 秘密选择整数 n_A，计算 $K_{UA} = n_A G$，然后公开 (p, a, b, G, K_{UA})，K_{UA} 为公钥，n_A 为私钥。

(2) **加密**：用户 B 要把消息 M 发给 A，先把消息 M 变换成为 $E_p(a, b)$ 中一个点 P_m，然后，选择随机数 r，计算密文 $C_m = \{rG, P_m + rK_{UA}\}$，如果 r 使得 rG 或者 rK_{UA} 为 O，则要重新选择 r。

(3) **解密**：用户 A 对密文 C_m 解密，$(P_m+rK_{UA})-n_A(rG)=P_m+rn_AG-n_ArG=P_m$。

与有限域 GF(p)上的 ElGamal 加密算法相同，加密信息有扩张。

取 $p=751$，$E_p(-1, 188)$，这等价于曲线 $y^2=x^3-x+188$，$G=(0, 376)$。假设 B 希望发送一个报文给 A，这个报文被编码为椭圆曲线的点 $P_m=(562, 201)$，B 选择随机数 $r=386$，A 的公开密钥是 $K_{UA}=(201, 5)$，我们有 $rG=386(0, 376)=(676, 558)$，$P_m+rK_{UA}=(562, 201)+386(201, 5)=(385, 328)$。因此 B 发送密文{(676, 558), (385, 328)}。

4.7.3.4　椭圆曲线密码体制的安全性

经过密码学研究人员20多年的努力，椭圆曲线密码已经得到了长足的发展。各种基于椭圆曲线的加密算法和密钥交换机制被提出并实现，椭圆曲线密码算法的标准化工作也正在进行中。IEEE 的 P1363 定义了椭圆曲线的公钥系统。无线通信的 WAP 协议中的 WTLS 层，也可以选择使用椭圆曲线密码系统。RSA 和椭圆曲线密码分别基于不同的数学难题，相对来说，椭圆曲线密码能用更少的密钥位来获得更高的安全性，而且加密速度比RSA 要快，使它在许多计算资源受限制的环境里得到广泛的应用。目前已知求解椭圆曲线离散对数问题的最快方法为 Pollard rho 方法，表 4.5 比较了这种方法与使用扩展数域筛法(GNFS)分解整数来破解 RSA 算法的效率。两者都可以用于加解密，密钥交换和数字签名。

表 4.5　ECC 和 RSA 性能比较

ECC 密钥长度 (n 的比特位)	RSA 密钥长度 (模 n 的比特位)	MIPS 年
106	512	10^4
132	768	10^8
160	1024	10^{12}
211	2048	10^{20}
320	5120	10^{36}
600	21 000	10^{78}
1200	120 000	10^{168}

4.8　密码算法小结

公钥密码虽然具有一些单钥算法无法比拟的优点，但关于公钥算法的以下认识也是不对的：

- 公开密钥密码算法更安全。决定算法安全性的不是体制，从前面可以看到算法的安全性和密钥长度相关。要得到和对称算法同样的安全性，公钥算法需要更大的密钥长度。
- 公开密钥密码使对称密钥密码过时了。公开密钥密码算法计算量大，加密数据的速率较低。实际应用中人们通常将对称密码和公钥密码结合在一起使用，比如：利用 DES 或者 AES 来加密信息，而采用 RSA 来传递会话密钥。
- 公钥的分发是简单和一目了然的。公钥的分发虽然不需要保密，但需要保证公钥的真实性。这也需要相应的技术。

对称密码算法运算速度快、密钥短，可用于多种用途，比如随机数产生和Hash 函数的构造，历史悠久，但面临密钥管理的困难。使用对称密码的通信双方是平等的，所以不能为发送者提供保护。

非对称密码算法只需保管私钥，可以相当长的时间保持不变，需要的密钥数目较小，但运算速度慢，密钥尺寸大，出现的历史很短。使用非对称密码通信的双方是不平等的，因为加密消息和验证签名的人不能解密同一信息和产生同样的签名。

思考和练习题

(1) 简要说明 Diffie-Hellman 密钥交换。

(2) 应用 RSA 算法对下列情况进行加解密。

(a) p=3，q=11，e=7，M=5

(b) p=5，q=11，e=3，M=9

(c) p=7，q=11，e=17，M=8

(3) 设 RSA 算法的 n=35，e=5，密文 C=10，对应的明文 M 是什么？

(4) 使用 Fermat 定理计算 $5^{302} \bmod 31$。

(5) 尽可能全面地给出对称密码算法和非对称密码算法特点的异同分析。

(6) 在 ElGamal 算法中，为什么要使用不同的随机数 k 来加密不同的信息？

(7) 对于椭圆曲线 $E_{11}(1, 6)$，即 $y^2 = x^3 + x + 6 \mod 11$，考虑点 $G = (2, 7)$，已知私钥 n=7，求

(a) 公钥 K_U;

(b) 已知明文 P_m= (10, 9)，并选择随机数 r=3，确定密文 C_m。

(8) 椭圆曲线群 $E_p(a, b)$ 定义的加法运算规则是什么？

实践/实验题

用 C 语言实现 RSA 算法，包括素数生成，加密和签名。进行加解密正确性测试及性能测试。

第 5 章　消息鉴别和数字签名

在第1章提到安全的信息交换应该满足的5个基本性质是：机密性、完整性、可用性、鉴别和不可否认性。这 5 种基本性质，主要是针对通信系统面临的不同威胁提出的。前面介绍的对称密码和公钥密码体制，主要是为了对抗窃听和业务流分析等形式的威胁，解决信息的机密性问题。而实际的信息系统和通信网络还可能受到消息篡改、冒充和抵赖等形式的威胁和攻击。攻击者可以对从发送方发送到接收方的消息进行下列形式的篡改：插入、删除或修改消息的内容，插入、删除或重组消息序列，延迟或重放消息。如何防止第三方冒充其中一方发送消息或者篡改所传送的消息，注意篡改并不一定需要破解密文，第三方可能只是把之前获得的一段消息插入到当前传输的消息中，也可能就是加入一些随机串，幸运的情况下可以在解密后发现被篡改过，可是仍然有无法发现篡改痕迹的可能，这属于信息安全的完整性问题。攻击者也可以假冒信息源向网络中插入消息，如何防止第三方冒充其中一方发送消息，这属于信息安全的真实性问题。发送者可以对自己发送过的消息进行否认，如何防止通信的另一方进行欺骗——想象一下在股票交易中，A 给他的经纪人 B 发了一个消息，要求 B抛售他持有的所有股票，事后很不幸地亏损了，这时A否认他给B发过这样的消息——这属于信息安全中的不可否认问题。虽然有方案用纯粹的加密方法——公钥或者对称密钥密码算法来解决这些问题，可惜它们并不如想象得那么好用。本章将要介绍的消息鉴别和数字签名技术，就是为了对抗这些威胁，从而保证信息完整性、实现实体鉴别和抗否认的基本方法和技术。

5.1　消息鉴别

消息鉴别(Message Authentication)是一个证实收到的消息来自可信的源点且未被篡改的过程。鉴别的主要目的有两个。第一，验证信息的发送者是真正的，而不是冒充的，此为信源识别；第二，验证信息的完整性，在传送或存储过程中未被篡改、重放或延迟等。

5.1.1　鉴别系统模型

图5.1给出了一个单纯鉴别系统的模型。鉴别编码器和鉴别译码器可抽象为**鉴别函数**。一个安全的鉴别系统，需满足下列要求：意定的接收者能够检验和证实消息的合法性、真实性和完整性；消息的发送者和接收者不能抵赖；除了合法的消息发送者，其他人不能伪造合法的消息。为了达到以上目的，通常的做法是先选好恰当的鉴别函数，该函数产生一个鉴别标识，然后在此基础上，给出合理的鉴别协议(Authentication Protocol)，使接收者完成消息的鉴别。

用于产生鉴别符的鉴别函数分为以下三类：

(1) **消息加密函数**：用完整消息的密文作为对消息的鉴别。

(2) **消息鉴别码(Message Authentication Code, MAC)函数**：是一个密钥控制的公开函数，对变长的报文产生一个固定长度的数值。

(3) **散列函数(Hash Function)**：一个散列函数是一个公开的函数，以一个变长的报文作为

输入，并产生一个固定长度的散列码作为输出，该输出有时也称为报文摘要。

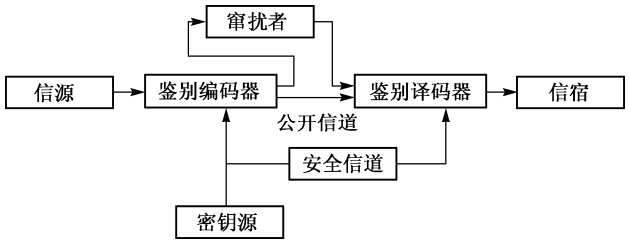


图 5.1 一个单纯鉴别系统模型

5.1.2 消息加密

消息自身的加密可以作为一个鉴别的度量，在对称密钥和公开密钥两种模式下有所不同。

5.1.2.1 对称加密

如果用对称密钥加密，在提供机密性保护的同时，的确可以提供一定程度的内容和来源的真实性。这基于通信双方有共享密钥K_{AB}，第三方没有该密钥的前提。如果用户 B 收到 A 发给他的密文，他能够用与 A 共享的密钥 K_{AB}解密的话，他相信消息的确来自 A，传输中没有被更改，因为不知道他们共享密钥K_{AB}的人无法产生可以用K_{AB}加密的密文。人可以直接判断收到的密文是否被解密为有意义的明文，但是如何让机器自动确定这一点呢？分组加密算法相当于一个数学映射，对于任意的 b 位输入都可以产生 b 位输出，如果没有额外信息的话，机器无法知道解密出来的明文是否为可懂的明文。一种解决办法是强制明文有某种结构，例如，可以在加密前对每个消息附加一个错误检测码，也称为**帧校验序列**(Frame Check Sequence，FCS)或校验和。在接收端，机器对解密后的消息应用同样的校验函数，若得到的校验和与收到的一致，则认为消息是真实的。事实上，在要发送的信息中加入任何类型的结构信息都会增强鉴别的能力，如在 TCP 协议中，为了防止 TCP 数据报被篡改，TCP 数据报的头部包含校验和，并且对于给定连接，连续的TCP 数据报是按顺序编号的，防止攻击者进行延时、删除、重放等形式的篡改。

加密与FCS 执行的顺序很重要，先计算FCS 再加密被称为内部差错控制，由于攻击者不知道密钥，很难产生解密后校验和正确的密文。先加密再计算 FCS 被称为外部差错控制，攻击者就可以构造具有正确校验和的消息，虽然攻击者在不知道密钥的情况下，他也不知道构造的密文解密后的明文是什么，但可以造成混淆从而破坏通信。

5.1.2.2 公钥加密

如果使用公开密钥加密，前提是知道对方的公钥，各自的私钥保密。用公钥对信息加密提供机密性，不能保证发送方的真实性，任何用户都可以假冒 A 用B 的公钥对消息加密。用私钥对信息数字签名，不提供机密性，如果其他用户收到用户 A 签名的消息，并且可以用 A 的公钥验证签名的正确性的话，他就可以相信消息的确来自 A，因为只有 A 拥有能产生正确签名的私钥，并且信息在传输中没有被篡改。如果既要提供机密性，又要提供鉴别，用户 A 就需要先用自己的私钥签名，再用用户 B 的公钥加密，一次通信中就要进行四次复杂的加解密。

5.1.3　消息鉴别码 MAC

5.1.3.1　消息鉴别码的需求

使用一个密钥生成一个固定大小的小数据块，并加入到消息中，称为**消息鉴别码**(MAC)或**密码校验和**(Cryptographic checksum)。它由如下形式的函数产生：

$$\mathrm{MAC}=\mathrm{C}(K, M)$$

其中 M 是一个变长的消息，K 是收发双方共享的密钥，$\mathrm{C}(K, M)$ 是定长的鉴别符。消息和MAC被一起发送给接收方。接收方对收到的消息用相同的密钥进行相同的计算得出新的MAC，与接收到的MAC相比较，如果两者相等，接收者可以确信消息 M 未被改变，并且消息来自所声称的发送者，如果消息中包含顺序码(如 X.25，TCP 中使用的序列号)，则接收者可以保证消息的正常顺序。MAC函数类似于加密函数，但不需要可逆性。因此在数学上比加密算法被攻击的弱点要少。

对称加密可以提供鉴别，且已被广泛应用，为什么还要使用单独的消息鉴别码呢？因为，机密性与真实性是两个不同的概念，从根本上讲，信息加密提供的是机密性而非真实性。首先，加密运算的代价很大，公钥算法代价更大；其次，鉴别函数与加密函数的分离能提供功能上的灵活性，可以把加密和鉴别功能独立地实现在通信的不同传输层次；再次，能够在目标系统中延长对消息的保护时间，因为加密只保证传输中的机密性，一旦消息到达接收者的机器并被解密后，就不能保护消息不被修改；此外，某些信息只需要真实性，不需要机密性，比如：广播的信息，信息量大，难以使用加密；政府/权威部门的公告和网络管理信息等只需要保证真实性。

MAC 的基本用法有三种：

(1) MAC 直接附加在消息之后

$$\mathrm{A}\rightarrow\mathrm{B}：M \,\|\, \mathrm{MAC}_K(M)$$

MAC 基于 A 和 B 共享的密钥 K 生成。如果 B 对收到的消息生成的 MAC 与收到的 MAC 相同，他可以确认消息一定来自 A，且没有被篡改。这个过程没有提供对消息的保密。

(2) MAC 直接附加在消息之后，并对整体进行加密

$$\mathrm{A}\rightarrow\mathrm{B}: \mathrm{E}_{K_2}[M \,\|\, \mathrm{MA\,C}_{K_1}(M)]$$

这里密钥 K_1 用于生成鉴别码，提供对消息的鉴别，消息的机密性由 A 和 B 共享的密钥 K_2 提供。

(3) 先对消息加密，再对密文生成鉴别码

$$\mathrm{A}\rightarrow\mathrm{B}: \mathrm{E}_{K_2}[M] \,\|\, \mathrm{MA\,C}_{K_1}(\mathrm{E}_{K_2}[M])$$

这种方法也同时提供保密和鉴别，一般来说，将 MAC 附加于明文之后更好一些。

5.1.3.2　基于对称分组密码的消息鉴别码

消息鉴别码最初主要是基于对称分组密码算法而设计的，一个使用广泛的数据鉴别算法(Data Authentication Algorithm)，也被称为 CBC-MAC(密文分组链接消息鉴别码)，定义于标准 ANSI X9.9，FIPS PUB 113 和 ISO/IEC 9797 中，它使用分组长度为 b 位的对称分组密码算法的 CBC(Cipher Block Chaining)工作模式对消息进行加密，并取最后一个密文分组最左边的 M 位作为 MAC 值，M 的大小可由通信双方约定，如图 5.2 所示。首先将数据按 b 位分组，假设消

息被划分为 N 个分组：$D_1, D_2, \cdots, D_N$，如果消息不是分组长度的整数倍，最后一个数据块用一个 1 和若干个 0 向右填充。运用加密算法 E 和密钥 k，消息鉴别码(MAC)的计算如下：

$$O_1 = \mathrm{E}_k(D_1)$$
$$O_i = \mathrm{E}_k(D_i \oplus O_{i-1}) \qquad (1 < i < N)$$

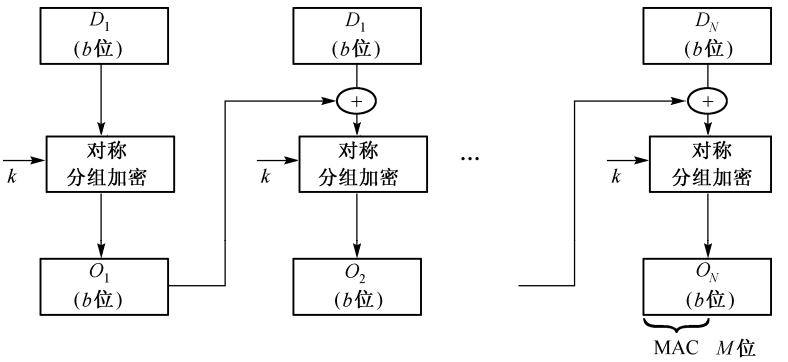


图 5.2 CBC-MAC

基于 CFB 工作模式下的对称分组密码算法也可以构造 MAC，如图 5.3 所示。

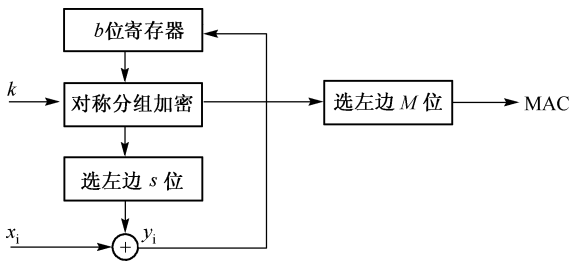


图 5.3 基于分组密码 CFB 模式下的 MAC

在散列函数提出之前，基于 DES 的 MAC 获得广泛应用，但 MAC 需要对全部数据进行加密，速度很慢，由此导致了直接设计散列函数对给定数据生成鉴别码的研究。

5.1.4 散列函数

5.1.4.1 散列函数的用途

散列值 h 由下述形式的函数 H 生成：

$$h = \mathrm{H}(M)$$

H 的输入为任意长度的消息 M，输出为一个固定长度的散列值 h，称为**消息摘要**(Message Digest)。这个散列值是消息 M 所有位的函数。它能够提供错误检测能力，因为消息中的任何一位或多位的变化都将导致该散列值的变化。消息摘要函数在不同的场合有不同的名称，它的音译是**哈希函数**。因为它可以像指纹一样代表一串数据，也被称为**数字指纹**(Digital finger print)。因为它对任意长度的输入产生固定长度的输出，也被称为压缩(Compression)函数或紧缩(Contraction)函数。当它用于数据篡改的检验时，也被称为**数据鉴别码** DAC(Data Authentication Code)或**篡改检验码** MDC(Manipulation Detection Code)。

散列函数的两个主要用途是消息的完整性检测和数字签名。数字签名是一种给数字形式存储的消息签名的方法，我们在下一节进行介绍。

图5.4给出了使用Hash函数进行完整性检测的模型。发送者要发送的明文消息为x，他应用散列函数 H 生成 x 的消息摘要 H(x)，把消息 x 和消息摘要 H(x)的密文 y 通过公开信道同时发送给接收者，接收者对接收到的消息 x' 生成消息摘要 H(x')，与通过对 y 解密得到的消息摘要 H(x)进行比较，如果两者相等，就可以确认 $x'=x$，消息 x 在传输中没有被篡改。

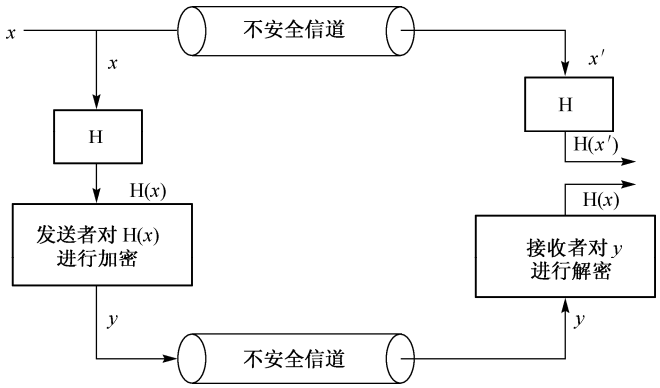


图 5.4　使用 Hash 函数进行完整性检测的模型

将消息摘要用于消息鉴别的具体方法，可以总结为以下 6 种。

(1)用对称密码对消息及附加在其后的消息摘要加密

$$\mathrm{A}\rightarrow\mathrm{B}：\mathrm{E}_K[M\|\mathrm{H}(M)]$$

这种方式提供了机密性，因为消息 M 用 A 和 B 共享的密钥 K 加密。如果 B 使用密钥 K 对接收到的密文解密，再对得到的消息生成消息摘要，如果与解密得到的原始消息摘要相同，他可以确认消息一定来自 A，且没有被篡改。

(2)用对称密码只对消息摘要加密

$$\mathrm{A}\rightarrow\mathrm{B}：M\|\mathrm{E}_K[\mathrm{H}(M)]$$

这种方法只提供真实性保护，消息摘要 H(M)用 A 和 B 共享的密钥 K 加密。如果 B 对接收到的消息生成消息摘要，与解密得到的原始消息摘要相同，他可以确认消息一定来自A，且没有被篡改。

(3)仅用发送方的私钥对消息摘要签名

$$\mathrm{A}\rightarrow\mathrm{B}：M\|\mathrm{E}_{\mathrm{KRa}}[\mathrm{H}(M)]$$

这种方法提供真实性保护和数字签名。首先，同方法(2)一样，该方法提供真实性保护，如果B能用A的公钥验证签名的正确性，他可以确认消息一定来自A，且没有被篡改。其次，因为只有 A 能生成这样的签名，所以该方法也提供了不可否认保护。

(4)用发送方的私钥对消息摘要签名，再对全部消息加密

$$\mathrm{A}\rightarrow\mathrm{B}：\mathrm{E}_K[M\|\mathrm{E}_{\mathrm{KRa}}[\mathrm{H}(M)]]$$

该方法既提供对消息的保密，又提供数字签名，比较常用。

(5)该方法使用散列函数，但不使用加密函数来进行消息鉴别

$$\mathrm{A}\rightarrow\mathrm{B}：M\|\mathrm{H}(M\|S)$$

因为双方共享秘密值S，B也可以计算散列值，并验证正确性，而不知道秘密值的攻击者不能伪造和篡改消息。

(6) 如果对整个消息和消息摘要加密，方法(5)在提供消息鉴别的同时，也可以提供机密性

$$A \rightarrow B: E_K[M||H(M||S)]]$$

上述方法中，方法(5)避免了对加密函数的使用，引起了人们的极大关注。避免使用加密函数的原因有：加密软件很慢；加密硬件的开销很大；加密是对大长度数据进行优化的；加密算法可能受专利保护；加密算法可能受出口的限制。

5.1.4.2 对散列函数的要求

在第 4 章介绍了基于 RSA 的数字签名算法，这样的签名算法存在以下弱点：

- 假设用户 B 的公钥 $K_{Ub}=e$，私钥 $K_{Rb}=d$，任何人能通过对某一给定的 y 计算 $x=E_{KUb}(y)=y^e \bmod n$，伪造一个 B 对随机消息 x 的签名，因为 $y=Sig_{KRb}(x)=(y^e)^d=y^{ed} \bmod n$。
- 如果消息 x_1，x_2 的签名分别是 y_1 和 y_2，则拥有 x_1，x_2，y_1，y_2 的任何人可伪造 B 关于消息 x_1x_2 的签名 y_1y_2，因为 $Sig_{KRb}(x_1x_2)=(x_1x_2)^d \bmod n=x_1^d x_2^d \bmod n=(x_1^d \bmod n)(x_2^d \bmod n)=Sig_{KRb}(x_1)Sig_{KRb}(x_2)$。
- 签名者每次只能签 $\log_2 n$ 比特长的消息，获得同样长的签名。如果消息很长，需要逐组签名。然而，这样做，速度慢，签名长。

解决以上问题的办法是引入可公开的散列函数。它将取任意长度的消息做自变量，结果产生规定长度的消息摘要。例如，使用数字签名标准 DSS，一个消息摘要为 160 bit，对消息摘要进行签名，结果如下：

消息	X	任意长
消息摘要	$z=H(x)$	160 bit
签名	$y=Sig_k(z)$	320 bit(签名一个消息摘要)

使用散列函数进行数字签名的好处有：由于把对原始消息的签名替换成了对消息的短的消息摘要的签名，提高数字签名的速度；无须泄露签名所对应的消息，可将签名泄露，如对消息 x 的签名是 $y=Sig_k(z)$，其中 $z=H(x)$，可将 (z, y) 公开，而保密 x；可将签名变换和加密变换分开，在 OSI 的不同层提供消息的完整性和机密性。

用以鉴别的散列函数，能否减弱鉴别方案的安全性？这个问题是要分析的。签名的对象由完整消息变成消息摘要，这就有可能出现伪造。下面分析可能的伪造方式。

伪造方式一：Oscar 以一个有效签名 (x, y) 开始，此处 $y=Sig_k(H(x))$。首先他计算 $z=H(x)$，并企图找到一个 x' 满足 $H(x')=H(x)$。若他做到这一点，则 (x', y) 也将被识别为有效签名。为防止这一点，要求函数 H 具有无碰撞特性。

散列函数 H 称为是**弱无碰撞的**，是指对给定消息 $x \in X$，在计算上几乎找不到不同于 x 的 $x' \in X$ 使 $H(x)=H(x')$。

伪造方式二：Oscar 首先找到两个消息 $x \neq x'$，满足 $H(x)=H(x')$，然后 Oscar 把 x 给 Bob 且使他对 x 的摘要 $H(x)$ 签名，从而得到 y，那么 (x', y) 就是一个有效的伪造。为防止这一点，要求函数 H 具有强无碰撞特性。

散列函数 H 被称为是**强无碰撞的**，是指在计算上几乎不可能找到相异的 x 和 x'，使得 $H(x)=H(x')$。

这里，强无碰撞自然含弱无碰撞。因为很显然，如果找不到任意的 (x, y) 满足条件的话，给定 x 也一定找不到。

伪造方式三：在散列函数的用法(5)中，秘密值S本身并不发送，如果散列函数不是单向的，攻击者截获到 M 和 H(M||S)，然后通过某种逆变换获得 M||S，这样攻击者就可以得到 S。为防止这一点，要求函数 H 具有单向性。

称散列函数 H 为单向的，是指计算散列函数 H 的逆函数 H^{-1} 在计算上不可行。

综上所述，我们认为散列函数需要满足如下性质：

(1) H 可以作用于一个任意长度的数据块。

(2) H 产生一个固定长度的输出。

(3) 对任意给定的 x，H(x) 的计算相对容易，无论是软件还是硬件实现。

(4) 对任意给定散列码 h，找到 x 满足 H(x) = h 具有计算不可行性，这被称为**单向性**(**抗原像攻击**)。

(5) 对任意给定的数据块 x，找到满足 H(y) = H(x) 的 $y \neq x$ 具有计算不可行性，这被称为**弱无碰撞性**(**抗第二原像攻击**)。

(6) 找到任意数据对 (x, y)，满足 H(x) =H(y) 是计算不可行的，这被称为**强无碰撞性**。

前三条要求散列函数具有计算可行性，第 4 条是单向性质，即给定消息可以产生一个散列码，而给定散列码不可能产生对应的消息。第 5 条性质是保证给定一个消息，不能找到与之散列码相同的另外一个消息，即防止伪造。第 6 条则保证了一种被称为生日攻击的方法无法奏效。

近年来，随着散列函数分析技术的进步，出现了一些新的攻击方式，如长消息的第二原像攻击和选择前缀的原像攻击，因此，NIST 在全球范围内征集新的散列函数时，增加了抗长度扩展攻击(Length-extension attack)的安全属性：给定 H(M)和 M 的长度，对于任意的消息 M'，即使当 M 未知，可以直接利用 H(M)来计算 H($M \| M'$) 是困难的。

理想情况下，还有一些更强的条件和要求，如对于攻击者来说找到两个具有大体相似的消息摘要的消息是不可能的，或者给定消息摘要，从中推出任何有用的消息也是不可能的。因此，密码学意义上的散列函数还应该具有尽量多的随机函数特性。一些校验和算法，如 CRC32 及其他循环冗余校验序列就不适合作为密码学意义上的散列函数。

5.1.4.3　散列函数的安全性

散列函数是一种多对一函数，理论上一定存在碰撞。假定使用 64 位的散列码，是否安全？如果传输的是加密的散列码和不加密的报文 M，对手需要找到 M'，使得 H(M')=H(M)，就可以使用替代报文来欺骗接收者。一种基于生日悖论的攻击可能做到这一点。

与生日悖论相关的生日问题是：一个教室中，最少应有多少学生，才使至少有两人具有相同生日的概率不小于 1/2？一年有 365 天，如果有 366 人的话，至少两人的生日相同。直觉上这个数字不会小，而概率结果与人的直觉是相违背的，从这个意义上，它被称为悖论。实际上只需 23 人，即任找 23 人，其中两人具有相同生日的概率至少为 1/2。

一个与生日问题类似的散列函数问题是：给定一个散列函数和散列值 H(x)，假定 H 有 n 种可能的输出，如果 H 有 k 个随机输入，k 必须为多大才能使至少存在一个输入 y，使得 H(y) = H(x) 的概率大于 0.5。

对于一个输出值 h 为 m 比特的散列函数，共有 $n = 2^m$ 种可能的输出，在理想状态下，对任何输入，h 应均匀地在整数 0 到 2^m–1 之间取值，也就是说，散列函数输出恰好是给定输出值的概率是 1/n。因此，任取一个 y，H(y) =H(x) 的概率为 1/n，反过来 H(y) ≠ H(x) 的概率为 1–(1/n)。如果产生 k 个随机值 y，他们的消息摘要都不等于 H(x) 的概率等于每个个体不匹配概率的乘积，

即$[1-(1/n)]^k$，这样，至少有一个匹配的概率为$1-[1-(1/n)]^k \approx 1-[1-(k/n)]=k/n$。若要概率等于0.5，只需$k=n/2$。若要概率等于1，只需$k=n$。

因此，我们有如下结论：对长度为m位的散列码，共有2^m个可能的散列值，给定$H(x)$，若要找到一个y，使得$H(y)=H(x)$的概率为0.5，只需要k的值为：$k=2^{m-1}$。若要上述概率为1，只需要k的值为：$k=2^m$。

更进一步，有下面的结论：对长度为m位的散列码，共有2^m个可能的散列值，若要使任意的x，y有$H(x)=H(y)$的概率为0.5，只需$k=2^{m/2}$。

基于上述结论，攻击者可以进行这样一种攻击，因为它建立在生日悖论的基础之上，被称为生日攻击。

(1) 攻击者产生一个有效消息的$2^{m/2}$种变化，每一个消息具有同样的意义。

(2) 攻击者也产生一个期望的欺骗性消息的$2^{m/2}$种变化。

(3) 比较两组消息，找到具有同样散列值的消息对，根据生日悖论，找到的概率大于0.5。

(4) 让用户签署有效的文件，然后用欺骗性消息进行替代，因为两者的消息摘要相同，因此对有效文件的签名也是对欺骗性消息的有效签名。

一个生日攻击的例子是：

(1) A准备两份合同M和M'，一份B会同意，一份让他转让他的全部财产而被他拒绝。

(2) A对M和M'各构造32处微小变化，比如在某处添加一个空格，改变某处的行间距或字间距，这种变化不会改变合同的原意，选择进行或不进行这32处修改，可以得到2^{32}个保持原意但有微小变化的合同文本，分别产生2^{32}个64位散列值。

(3) 根据前面的结论，超过0.5的概率能找到一个M和一个M'，它们的散列值相同。

(4) A提交M，经B审阅后产生64位散列值并对该值签名，返回给A。

(5) A用M'替换M。

综上，可以得出结论，散列函数要满足单向性、弱无碰撞性和强无碰撞性，散列值必须足够长。

5.1.4.4 散列函数的构造

散列函数的构造方法有三种，一种是基于数学难题的构造方法，这种方法构造的散列函数计算速度慢，不实用。第二种是利用对称密码体制来设计散列函数，第三是直接设计。

用对称加密算法构造散列函数的一种做法是把消息M分成长度相同的分组，$M=(M_1, M_2, \cdots, M_t)$，$H_0$为初始值，$H_i=f(M_i, H_{i-1})$，例如$H_i=E_{M_i}(H_{i-1})$，散列值为$H_t$。这种方法速度慢，且有许多这样的散列函数被证明是不安全的，其安全性与E的安全性无关。下面的四种可能是安全的：

① $H_i=E_{H_{i-1}}(M_i)\oplus M_i$

② $H_i=E_{H_{i-1}}(M_i)\oplus M_i\oplus H_{i-1}$

③ $H_i=E_{H_{i-1}}(M_i\oplus H_{i-1})\oplus M_i$

④ $H_i=E_{H_{i-1}}(M_i\oplus H_{i-1})\oplus M_i\oplus H_{i-1}$

1989年，Merkle和Damgård分别独立地提出一种与分组密码类似的通用迭代结构，它利用输入消息长度是固定的定长压缩函数来构造散列函数，人们称这种结构为Merkle-Damgård结构，如图5.5所示。该结构的具体做法是：

(1) 把原始消息M分成L个固定长度的块Y_{i-1}，$1\leqslant i\leqslant L$。

(2) 对最后一块填充并使其包含消息 M 的长度。

(3) 设定初始值 CV_0。

(4) f 为压缩函数，m 为散列码的长度，b 为输入块的长度，用初始向量 CV_0 与第一块报文计算散列值产生 CV_1，然后由 CV_{i-1} 与第 i–1 块计算 CV_i，$CV_i = f(CV_{i-1}, Y_{i-1})$，$1 \leq i \leq L$。

(5) 最后一个 CV_i 为消息的散列值，$H(M) = CV_L$。

如果压缩函数具有抗碰撞迭代的能力，那么迭代散列函数也具有抗碰撞的能力，具体散列函数之间的不同之处在于计算 CV 的函数 f。

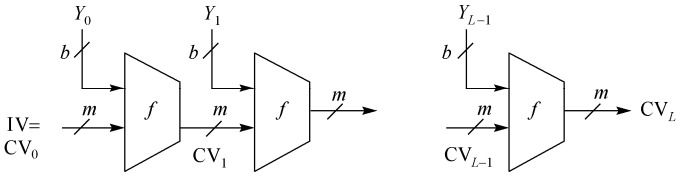


图 5.5　设计散列算法的 Merkle-Damgård 结构

目前，国际上给出了针对迭代结构的一些新的攻击方式，如比特追踪法、多碰撞攻击、长消息的第二原像攻击等，并证明 Merkle-Damgård 结构不是可证明安全的，并提出了一些改进的 Merkle-Damgård 结构，如宽轨道、HAIFA、Sponge 等。对散列函数的结构研究正成为国际研究的热点。

5.2　散列算法

1990 年，Ron Rivest 采用 Merkle-Damgård 结构第一个直接设计了散列算法 MD4。与分组密码相似的进展，强力攻击能力的提高，导致了算法的改进。分组密码从 DES 发展到 AES，散列算法也从 MD4 和 MD5 发展到 SHA-1、RIPEMD-160 以及 SHA-2 系列。1992 年，Ron Rivest 在 MIT 设计了 MD5 (RFC 1321)，在 SHA-1 之前，MD5 是最主要的散列算法。它的输入是任意长度的消息，输出是 128 位的消息摘要，以 512 位输入数据块为单位进行处理。安全散列算法 (SHA) 是由 NIST 设计，并于 1993 年作为联邦信息处理标准 FIPS 180 发布，修订版 (FIPS 180-1) 发布于 1995 年，通常称为 SHA-1。SHA-1 是美国 DSA 签名方案 (Internet RFC 3174) 使用的标准算法，它的输入是最大长度为 2^{64}–1 位的消息，输出是 160 位的消息摘要，以 512 位数据块为单位进行处理。RIPEMD-160 是著名的密码工程欧洲 RIPE (RACE Integrity Primitives Evaluation, RIPE) 工程的结果，最早为 128 位 RIPEMD，后改进成为 160 位消息摘要，它的基础是 MD5，输入是最大长度为 2^{64}–1 位的消息，输出是长度为 160 位的消息摘要，以 512 位数据块为单位进行处理。

2001 年 5 月 30 日，NIST 发布了修订版本 FIPS 180-2，增加了三个附加的散列算法，SHA-256, SHA-384 和 SHA-512，统称为SHA-2 系列。与 AES 密码的安全性相兼容，结构与细节类似于 SHA-1。SHA 算法的参数比较参见表 5.1。表中安全性是指对输出长度为 m 比特的散列函数的生日攻击产生碰撞的工作量为 $2^{m/2}$。

基于 MD4 算法的框架结构，人们设计的一系列散列算法，包括 MD5，HAVAL，RIPEMD，SHA-0，SHA-1，SHA-2 等，被统称为 MD4 系列的散列算法。根据消息分布或扩展方式的不同，MD4 系列的散列函数又可分为 MDx 类和 SHAx 类。2004 年 8 月 17 日，在美国加州圣 •巴巴拉召开的国际密码学会议 (Crypto'2004) 上，我国密码学家王小云教授宣布了包括 MD4，

MD5，HAVAL-128 以及 RIPEMD 在内的碰撞实例，立即引起密码学界、信息安全界和民众的轰动。该结果被认为是近年来密码学界最具突破性的结果，攻击的具体方法——比特追踪法正式发表在 2005 年的欧洲密码会上，它建立了对 MD4 类散列算法进行碰撞攻击的新理论，开创了散列函数研究的新高潮。在此之前，SHA-1 嵌入诸如加密软件(PGP)和安全套接层协议(SSL)等使用广泛的程序中。王小云教授等人公布的成果，证明MD5 已被破解，不能再被实际使用。对于 SHA-1 算法，碰撞攻击所需要的运算次数从原来设计时估算的 2^{80} 次降为 2^{61} 次。这些工作迫使美国国家标准与技术研究所(NIST)不得不采取一系列应对措施，2005 年，NIST 宣布了逐步停止使用 SHA-1 的意图，到 2010 之后使用 SHA-2 系列的版本。在 2010 之后，SHA-1 应该只用于 HMAC、随机数的产生和密钥推导函数(Key Derivation Functions, KDF)等。2006 年，NIST 开始启动为期 6 年的公开征集新散列算法标准 SHA-3 的计划。目前，有 14 个候选算法进入了 SHA-3 公开征集的第二轮评估。

表 5.1 SHA 参数比较

单位：比特	SHA-1	SHA-256	SHA-384	SHA-512
消息摘要长度	160	256	384	512
消息长度	$<2^{64}$	$<2^{64}$	$<2^{128}$	$<2^{128}$
分组长度	512	512	1024	1024
字长	32	32	64	64
步数	80	64	80	80
安全性	80	128	192	256

我国的《电子签名法》于 2005 年 4 月 1 日刚刚开始实施，在法律上确定了电子签名的有效性。电子签名的主体算法是 RSA，ECC 和 DSA 等公钥算法，而 MD5 和 SHA-1 等摘要算法是辅助算法，但起重要作用。算法的破解不会影响电子签名的法律效力，只是表明某种具体的电子签名算法需要替换，技术需要更新。

5.3 HMAC

散列算法设计出来后，人们希望使用散列函数而不是分组密码构造MAC，因为散列函数一般速度快，没有出口限制。由于散列函数不依赖于秘密密钥，不能直接用于MAC，最早提议基于下述办法构造基于密钥的散列函数：

$$\text{KeyedHash} = \text{Hash}(\text{Key}||\text{Message})$$

但由于发现了上述方法的一些弱点，最终导致了HMAC 的出现。HMAC 作为 RFC 2104 并在 SSL 中使用。

RFC 2104 给出了 HMAC 的设计目标：

- 无须修改地使用现有的散列函数。
- 当出现新的散列函数时，要能轻易地替换。
- 保持散列函数的原有性能不会导致算法性能的降低。
- 使用和处理密钥的方式简单。
- 对鉴别机制的安全强度容易分析，与散列函数有同等的安全性。

HMAC 可描述如下：

$$\text{HMAC}_K = \text{H}[K^+ \oplus \text{opad}) \,||\, \text{H}(K^+ \oplus \text{ipad}) \,||\, M]]$$

上式中的符号定义如下：

H——嵌入散列函数(MD5, SHA-1 等)

M——消息(包括散列函数所需填充位)

Y_i——M 的第 i 个数据块，$0 \leq i \leq L-1$

L——M 的数据块数

b——数据块的位数

m——嵌入散列函数产生的散列码长度位数

K——保密密钥，如果密钥长度大于 b，则密钥送入散列函数，形成一个 m 位的密钥；推荐长度大于等于 m

K^+——K 在左部添加 0 使得其长度为 b 位

ipad——00110110 重复 $b/8$ 次

opad——01011010 重复 $b/8$ 次

具体产生步骤如下：

(1) 对密钥 K 左边补 0 以产生一个 hash 用块 K^+。

(2) K^+每个字节与 ipad(00110110)作 XOR 以产生 $S_1=K^+ \oplus$ ipad。

(3) 对 $(S_1||M)$ 进行 hash。

(4) K^+每个字节与 opad(01011010)作 XOR 以产生 $S_0=K^+ \oplus$ ipad。

(5) $\mathrm{HMAC}_K = \mathrm{H}[S_0 \| \mathrm{H}(S_1 \| \mathrm{M})] = \mathrm{H}[K^+ \oplus \mathrm{opad}) \| \mathrm{H}[(K^+ \oplus \mathrm{ipad}) \| M]]$。

5.4　数字签名

消息鉴别能保护通信双方免受任何第三方的攻击。然而，它无法防止通信双方的互相攻击。如本章开头讨论到的，A 与B 关于股市投资失败的争执，还有一种可能就是A 伪造不同的消息，声称它来自于B，他只需要简单地生成一个报文，并附上由A 和B 共享的密钥生成的鉴别码就可以了。在这种情况下，发送方与接收方之间存在欺骗或抵赖，因此除了消息鉴别之外还需要别的东西，最常见的解决方案就是采用数字签名。

5.4.1　数字签名的功能与特性

5.4.1.1　手写签名和数字签名

政治、军事、外交等活动中签署文件，商业活动中签订契约或合同，以及日常生活中人们对书信或刷卡购物的凭单签字，传统上都是采用手写签名或印鉴。签名起到认证、核准和生效的作用。手写签名具有如下特性：签名是可信的，接收者相信签名者慎重签署了该文件，确保文件内容的真实可信；签名是不能伪造的，每个人的书写习惯有不同的特征，通过笔迹可以判断是否是其本人的签名；签名是不可重用的，签名只对所签署的文件有效；签名后的文件是不能更改的，为了保证修改是有效的，财务上规定对凭单任何地方的修改都必须签名，考卷批阅规定对考卷任何批阅的修改处也必须签名；签名是不可否认的，签名者无法否认自己的签名，因为他人模仿签名者的笔迹伪造的签名与原签名还是有差别的，可以被识别。

通信和网络技术的产生和发展，为人们提供了通过互联网进行通信、购物、汇款、办公的方便，随之而来的是电子信息世界的行为确认问题，与手写签名对应，数字签名应运而生。作为与手写签名有着同样目的的数字签名(Digital signatures)提供了如下功能：

- 签名者事后不能否认自己的签名。
- 接收者能验证签名，而任何其他人都不能伪造签名。
- 在有争议时，可由第三方进行验证。
- 对签名的作者、日期和时间、签名时刻消息的内容提供了验证。

从上面的解释和说明中，我们看出数字签名也能对消息的完整性提供保证和证明，此外它还提供了消息鉴别之外的附加功能——不可否认性。

手写签名和数字签名的主要差别有：

(1) **签署文件方面**：手写签名与被签的文件在物理上不可分割，数字签名与所签文件的“绑定”是通过数学变换来实现的。

(2) **验证方面**：手写签名通过与一个真实的手写签名相比较来验证，相对而言，这种方法不是绝对准确的。数字签名通过公开的验证算法来验证，任何人都可以验证签名的正确性，通过安全的算法和协议保证签名是不可伪造的。

(3) **“复制”方面**：手写签名不易复制，任何复制品都与原件有差别，而在电子信息世界里，得到数字签名的复制是非常容易的。

5.4.1.2 数字签名的定义

与加密和消息鉴别类似，可以给出数字签名的一个正式定义：一个签名方案是一个满足下列条件的五元组(P, A, K, S, V)：

(1) P 是所有可能消息组成的一个有限集合。

(2) A 是由所有可能的签名组成的一个有限集合。

(3) K 为密钥空间，它是由所有可能密钥组成的一个有限集合。

(4) 对每一个 $k\in K$，有一个签名算法 $\mathrm{Sig}_k \in S$ 和一个相应的验证算法 $\mathrm{Ver}_k \in V$。对每一个消息 $x\in P$ 和每一个签名 $y\in A$，每一个 Sig_k：$P\rightarrow A$ 和 Ver_k：$P\times A\rightarrow\{\text{true, false}\}$都是满足下列条件的函数：

$$\mathrm{Ver}_k(x,y)=\begin{cases}\text{true} & y=\mathrm{Sig}_k(x)\\ \text{false} & y\neq\mathrm{Sig}_k(x)\end{cases}$$

由 $x\in P$ 和 $y\in A$ 组成的数据对(x, y)称为签名消息。

5.4.1.3 数字签名的设计要求

为了设计一个安全的签名，需要考虑对签名方案的可能攻击。把可能的攻击分为下面几种：

- 唯密钥攻击(Key-only attack)：攻击者 Oscar 拥有 Alice 的公钥，即验证函数 Ver_k 。
- 已知消息攻击(know message attack)：Oscar 拥有一系列以前由 Alice 签名的消息(x_1, y_1)，(x_2, y_2)，其中 x_i 是消息，y_i 是 Alice 对消息的签名。
- 选择消息攻击(Choose message attack)：Oscar 请求 Alice 对一个消息列表签名。

攻击者对签名方案可能有如下的攻击目的：

- 完全破译(Total break)：攻击者 Oscar 可以确定 Alice 的私钥，即签名函数 sig_k，因此能对任何消息产生有效签名。
- 选择性伪造(Selective forgery)：攻击者能以某一不可忽略的概率对另外某个人选择的消息产生一个有效的签名，该消息不是以前 Alice 曾经签名的消息。

- 存在性伪造(Existential forgery)：攻击者至少能够为一则消息产生一个有效的签名，该消息不应该是以前 Alice 曾经签名的消息。

我们以 RSA 签名方案为例说明可能的攻击类型。

Oscar 能通过对某一 y 计算 $x = E_{KUa}(y)$ 伪造一个 Alice 对随机消息 x 的签名，因为 $y = Sig_{KRa}(x)$。这是一种唯密钥攻击的存在性伪造。

如果 Osacr 拥有 Alice 对消息 x_1，x_2 的签名分别是 y_1 和 y_2，则 Oscar 可伪造 Alice 关于消息 $x_1x_2 \bmod n$ 的签名 $y_1y_2 \bmod n$，因为 $Sig_{KRa}(x_1x_2) = Sig_{KRa}(x_1)\ Sig_{KRa}(x_2) \bmod n$。这是一种已知消息攻击的存在性伪造。

假定 Oscar 要伪造消息 x 的签名，Oscar 找到 x_1，$x_2 \in Z_n$，使 $x \equiv x_1x_2 \bmod n$。他请求 A 对 x_1，x_2 签名，签名结果分别是 y_1 和 y_2。$y_1y_2 \bmod n$ 是消息 $x_1x_2 \bmod n$ 的签名。这是一种选择消息攻击的选择性伪造。

考虑到以上的攻击，我们给出对数字签名的设计要求：

- 签名必须是依赖于被签名信息的一个位串模式，类似于笔迹签名与被签文件的不可分离性。
- 签名必须使用某些对发送者是唯一的信息，以防止双方的伪造与否认，类似于笔迹签名的独特性。
- 必须相对容易生成该数字签名。
- 必须相对容易识别和验证该数字签名。
- 伪造该数字签名在计算复杂性意义上具有不可行性，既包括对一个已有的数字签名构造新的消息，也包括对一个给定消息伪造一个数字签名，类似笔迹签名不可模仿性。
- 在存储器中保存一个数字签名副本是现实可行的，显然签名不能太长。

5.4.2　数字签名方案

根据接收者验证签名的方式可以将数字签名分为直接数字签名(Direct digital signature)和仲裁数字签名(Arbitrated digital signature)。从计算能力上，可将数字签名分为无条件安全的数字签名和计算上安全的数字签名。现有的数字签名大部分是计算上安全的，所谓计算上安全是指任何伪造者伪造签名是计算上不可行的。Chaum 和 Roijakkers 提出了第一个无条件安全的数字签名，理论上它能代替计算上安全的数字签名，但在实际应用中不太有效，因为这种签名需要一个复杂的交互密钥生成协议，而且签名很长。根据签名者在一个签名方案中能签的消息的数量，可将数字签名分为一次性的数字签名和多次性的数字签名。一次性数字签名方案只能签一个消息，如果签两个或以上的不同消息，伪造者就能伪造签名，类似于一次一密的方案。此外，还有一些具有特殊性质的数字签名。本节介绍一些有代表性的数字签名方案。

5.4.2.1　RSA 数字签名方案

RSA 算法中，公开密钥为$\{e, n\}$，其中 n 是两素数 p 和 q 的乘积，私有密钥为$\{d, p, q\}$，一种直接的数字签名方案为：

对消息 M 签名，计算 $y = \mathrm{Sig}(M) = M^d \pmod n$

验证签名，计算 $M' = y^e \pmod n$，判断其是否与原来的 M 相同。

这种签名方案有如下缺点：速度慢；信息量大，签名和明文一样长；第三方仲裁时必须暴露明文信息；且存在以下漏洞：$E_{KRa}(x \times y) \equiv E_{KRa}(x) \times E_{KRa}(y) \bmod n$。这一漏洞可以通过对

原始消息的消息摘要的签名替换直接对消息的签名来避免，先做摘要 $h_m = \mathrm{H}(M)$，再对 h_m 签名 $S_A = \mathrm{E}_{\mathrm{KRa}}(h_m)$，散列函数的无碰撞性保证了签名的有效性，如图5.6所示。

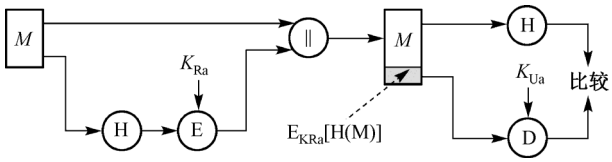


图 5.6 RSA 签名方案

本质上签名提供真实性(Authentication)，加密提供机密性(Confidentiality)。如果想要同时提供机密性和真实性，就需要同时实现签名和加密。当用户 A 向用户 B 发送消息时，有两种实现的方式：一种是先签名，后加密，$\mathrm{E}_{\mathrm{KUb}}\{M \parallel \mathrm{Sig}_{\mathrm{KRa}}(M)\}$；另一种是先加密，后签名，$\{\mathrm{E}_{\mathrm{KUb}}(M) \| \mathrm{Sig}_{\mathrm{KRa}}(\mathrm{E}_{\mathrm{KUb}}(M))\}$。后一种方法存在如下争议：

- 发生争议时，B 需要向仲裁者提供自己的私钥。
- 安全漏洞：攻击者 E 截获消息，把 $\mathrm{Sig}_{\mathrm{KRa}}(\mathrm{E}_{\mathrm{KUb}}(M))$ 换成 $\mathrm{Sig}_{\mathrm{KRe}}(\mathrm{E}_{\mathrm{KUb}}(M))$，让 B 以为该消息来自 E。
- 保存信息多：除了 M, $\mathrm{Sig}_{\mathrm{KRa}}(\mathrm{E}_{\mathrm{KUb}}(M))$，还要保存 $\mathrm{E}_{\mathrm{KUb}}(M)$，因为 K_{Ub} 可能过期。

5.4.2.2 数字签名标准 DSS/DSA

ElGamal于1985年提出了一个基于离散对数问题的数字签名体制，通常称为ElGamal 数字签名体制，它很大程度上为Diffie-Hellman密钥交换算法的推广和变形。该方案是特别为签名的目的而设计的。DSS(Digital Signature Standard)是由 NIST 于 1991 年提出的，1994 年 12 月 1 日被美国 NIST 采纳作为数字签名标准(FIPS 186)。DSS 使用 SHA 作为散列函数，并提出了一种新的数字签名技术，即数字签名算法 DSA(Digital Signature Algorithm)。

在数字签名算法 DSA 中，发送方先利用安全散列算法产生报文的消息摘要，然后将消息摘要和一个专用于该签名的随机数k作为签名函数的输入，该签名函数还依赖于发送方的私有密钥(K_{Ra})和一个全局公开密钥(K_{UG})，事实上是一组相关联的参数。生成的签名由两个分量组成，记为s和r。接收方将计算所收到报文的散列码，该散列码和签名作为验证函数的输入。验证函数同时还依赖于全局公开密钥以及与发送方私钥配对的发送方公钥。如果签名是有效的，验证函数的输出等于签名分量 r。如图5.7所示。

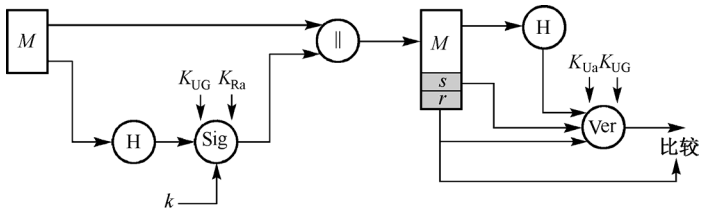


图 5.7 DSS 数字签名体制

DSA 算法的计算步骤如下：

(1) **全局公开密钥的选择**：首先选择一个160 bit 的素数 q，接着，选择一个素数 p，要求 p 的长度 L 在 512 到 1024 bit 之间，且 L 为 64 的倍数，$(p-1)$能被 q 整除。最后，选择 g 等于 $h^{(p-1)/q} \bmod p$，其中 h 是大于 1 小于$(p-1)$的整数，要求 g 大于 1。

(2) **选择用户私钥公钥对**：有了上面的全局公开密钥，用户就能选择一个私有密钥并产生一个公开密钥。私有密钥 x 必须是 1 到 $(q-1)$ 之间的随机数或者伪随机数，公开密钥 $y=g^x \bmod p$。给定 x 计算 y 是简单的，然而给定公开密钥 y，要确定 x 需要计算以 g 为底的模 p 的离散对数，这在计算上是不可行的。

(3) **生成签名**：用户为每个报文 M 选择随机数 k，其中 $0<k<q$，计算两个分量 r 与 s，$r=(g^k \bmod p) \bmod q$，$s=[k^{-1}(\mathrm{H}(M)+xr)] \bmod q$，签名为 (r,s)，发送签名 (r,s) 和消息 M。

(4) **验证签名**：接收方先根据收到的 s' 计算 $w, w=(s')^{-1} \bmod q$，然后根据 w 和收到的 r'，报文的散列值，y, q, g 等值计算验证签名值 v

$$u_1=[\mathrm{H}(M')w] \bmod q, u_2=(r')w \bmod q$$

$$v=[(g^{u_1}y^{u_2}) \bmod p] \bmod q$$

将 v 与接收到的 r 比较，若相等则接受签名。

DSS 具有如下特点：DSS 的签名比验证快得多，NIST 认为这对安全没有影响；DSS 不能用于加密或者密钥分配，只能用于数字签名；若 p 为 512 位，q 为 160 位，而 DSS 只需要两个 160 位，即 320 位。

DSS 使用中要注意避免以下问题：

(1) 当用户产生的签名 $s=0$，会泄露私钥

$$s=0=k^{-1}[\mathrm{H}(M)+xr) \bmod q$$

$$x=-\mathrm{H}(M)r^{-1} \bmod q$$

(2) 用户产生的 $s^{-1} \bmod q$ 要存在，要求 $s\neq 0 \bmod q$，如果得到的 $s\equiv 0 \bmod q$，接收者可拒绝该签名，要求重新构造该签名。实际上，$s\equiv 0 \bmod q$ 的概率非常小，为 2^{-160}。

(3) 不能将签名所使用的随机数 k 泄露出去。如果签名中所使用的随机数 k 泄露了，那么任何知道的人可由方程 $\mathrm{s}=[k^{-1}(\mathrm{H}(M)+xr)] \bmod q$ 求出：

$$x=[sk-\mathrm{H}(M)]r^{-1} \bmod q$$

一旦 x 被知道，攻击者就可以任意伪造签名。

(4) 不要使用同一个 k 签两个不同的消息，如果使用同一个 k 签两个不同的消息，会使攻击者计算私钥变得容易。

5.4.2.3　直接数字签名和仲裁数字签名

直接数字签名是接收者直接验证签名的方式，这种验证模式依赖于发送方私有密钥的保密。当发送方要抵赖发送过某一消息时，可能会声称其私有密钥丢失或被窃，从而他人伪造了他的签名。通常需要采用与私有密钥安全性相关的行政管理控制手段来制止或至少是削弱这种情况，但威胁在某种程度上依然存在。一种改进的方式是要求被签名的信息包含一个时间戳(日期与时间)，并将已暴露的密钥报告给一个授权中心。但是用户 U 的某些私有密钥确实在时间 T 被窃取，敌方可以伪造 U 的签名及早于或等于时间 T 的时间戳。为了避免以上问题，在数字签名体制中引入仲裁者。通常的做法是所有从发送方 U 到接收方 V 的签名消息首先送到仲裁者 A，A 对消息及其签名进行一系列测试，以检查其来源和内容，然后将消息加上日期并与已被仲裁者验证通过的指示一起发给V。仲裁者在这一类签名模式中扮演敏感和关键的角色。所有的参与者必须极大地相信这一仲裁机制工作正常。

5.4.2.4 不可否认的数字签名方案

不可否认的数字签名(Undeniable digital signature)方案是由 Chaum 和 van Antwerpen 在 1989 年提出的，其中最主要的特征是没有签名者的合作，签名就不能得到验证。从而防止了由她签署的电子文档资料没有经过她的同意而被复制和分发的可能性。不可否认签名的真伪性是通过接收者和签名者执行一个协议来推断的，这个协议称为否认协议。为了防止用户 A 否认一个之前的签名，一个不可否认签名与一个否认协议结合。用户 U 对签名进行验证，通过这种方式用户 U 能证明一个签名是伪造签名。不可否认的数字签名方案由三部分组成：签名算法、验证协议和否认协议。

5.4.2.5 一次性数字签名方案

如果一个签名方案仅给一则消息签名时是安全的，则签名方案是一次性签名方案，当然可以进行若干次验证。Lamport 数字签名方案就是一个一次性数字签名，它可以基于任何单向函数来构造，它的缺点是签名信息比较长。如果单向函数基于离散对数问题来构造，当 p 是 1024 位，则签名信息扩大 1024 倍。如果基于对称密码体制来构造，当密钥是 128 位，则签名信息扩大 128 倍。由于 Lamport 方案的签名太长，不太实用，改进方案之一是 Bos-Chaum 签名方案，使得签名的长度大约降低 50%。

5.4.2.6 群签名方案

群签名(Group digital signature)方案允许群中各个成员以群的名义匿名地签发消息，它具备下列三个特性：只有群内成员能代表那个群为消息签名；接收者能验证签名是来自于该群的合法签名；接收者不能确认签名者是群内的哪一个人，需要时，可借助于群成员或者可信机构找到签名者。群数字签名方案由三个算法组成：签名算法、验证算法和识别算法。群签名方案的应用例子有投标、公司打印机管理等。对于公司的打印机管理，群是公司所有员工的集合，通常情况下，每个员工都可以匿名地用群签名使用打印机，当出现打印机被滥用的行为时，无须签名者的合作，可以查出滥用的员工。Chaum 和 van Heijst 在1991年提出了一种群签名方案。

5.4.2.7 盲签名方案

所谓盲签名(Blind digital signature)是指签名人员只是完成对文件的签名，并不了解所签文件的内容。它要求：消息内容对签名者不可见；签名被接收者泄露后，签名者无法追踪签名。它应用在某些需要参加者匿名性的密码协议中，如电子货币、电子选举。如果不考虑顾客的隐私权，银行能追踪顾客的每一笔开支，顾客不能隐瞒把钱交给了谁，购买了什么东西，这种电子货币称为可追踪电子货币，利用现有的密码技术是很容易实现的。我们在实际生活中用现金购物，商家并不知道是谁购买了这个商品。如果电子货币也具有同样的特性，银行不知道顾客把钱付给了谁，购买了什么东西的话，这种电子货币称为不可追踪的电子货币。盲签名过程是先对消息进行盲变换，再对变换后的消息(称为盲消息)签名，接收者收到签名后进行逆盲变换。绝对的盲签名并不实用，加入“切选”(Cut and choose)技术的签名方案，是有实际需求的。一个基于 RSA 的盲签名方案如下：

B 的公钥 K_{Ub}，私钥是 K_{Rb}，f 是单向函数，如散列函数，A 需要 B 为他盲签消息 x。

(1) **对消息 x 进行盲变换**：A 随机选择 k，计算 $x' = f(x)\mathrm{E}_{KUb}(k) \bmod n$，把 x' 给 B。

(2) **B 对盲消息 x' 签名**：B 对 x' 进行签名得到 $y'=\mathrm{Sig}_B(x')=\mathrm{E}_{\mathrm{KRb}}(x')$，把 y' 给 A。

(3) **A 对签名逆盲变换**：A 计算 $y=y'/k \bmod n=\mathrm{E}_{\mathrm{KRb}}(f(x)\mathrm{E}_{\mathrm{KUb}}(k))/k \bmod n=\mathrm{E}_{\mathrm{KRb}}(f(x)) \bmod n$。

上面的方案中，随机数 k 使得签名者无法知道签名的内容，f 的单向性增加了安全性。

思考和练习题

(1) 消息鉴别是为了对付哪些方式的攻击？

(2) 产生消息鉴别符的鉴别函数有哪几种？

(3) 对称加密和错误检测码一起执行时，应按哪种顺序执行？说明其原因。

(4) 安全散列函数应该满足哪些性质？

(5) 是否可以利用散列函数构造分组密码，如何构造？

(6) 手写签名与数字签名有哪些异同？

(7) 数字签名应该满足哪些要求？

(8) 数字签名具有哪些性质？

(9) 对数字签名的主要攻击类型有哪些？请举例说明。

(10) 在 DSA 中，如果计算结果的 $s=0$，必须重新计算签名，为什么？

(11) 在 DSA 中，如果用于产生数字签名的 k 泄露，会出现什么问题？

(12) 在 DSA 中，为什么不能使用同一个 k 签名两个不同的消息？

实践/实验题

(1) 用 C 语言实现 DSA 程序。

(2) 用 C 语言实现基于 RSA 的盲签名方案。

第 6 章　密码实际应用问题

前面的章节给读者介绍了有关密码学的基本算法和原理，本章主要讨论密码算法实际使用过程中的一些问题。有效的安全性，不仅取决于强大的算法，还要求保障密码使用环节上的安全。这涉及密码功能的逻辑位置的选择，以及密钥的产生、存储和分配等密钥管理问题。

6.1　密码功能的位置

数据加解密可在网络OSI 7层协议的多个层上实现，从密码技术应用的逻辑位置看，数据链路层以下的加密称为**链路加密**，网络层以上的高层协议加密称为**端对端的加密**。

链路加密对两个相邻网络节点之间的数据传输提供安全保护。加密在数据加密设备(DEE)中实现，两个DEE 设备被安置在它们各自节点的数据终端设备(DTE)和数据通信设备(DCE)之间，并装配了同样的密钥。消息在被传输之前进行加密，每一个节点对接收到的消息进行解密，然后再使用下一个链路的密钥对消息进行加密传输。在到达目的节点之前，一条消息可能要经过许多通信链路的传输。

端对端的加密提供数据传输中一端到另一端的全程保密。由源主机或终端对消息进行加密，消息经过中间节点的传输，最后只在终点才进行解密。源主机和目的主机共享一个密钥。

一般来说，网络节点间的通信信息包括用户之间要交换的数据以及消息头部(报头)。消息报头通常包含路由信息，例如：源 IP 地址、目的 IP 地址、报文的序列号、消息的类型等。对于应用层加密，则只加密 TCP 信息片中的用户数据，传输层、网络层、链路层的信息头都以明文的形式出现。如果在传输层上实现加密，那么对于单个的端对端连接，用户数据和 TCP 头部被加密。因为路由的需要，IP 头信息还是明文。但是，当消息通过网关时，TCP 连接被终止并开始一段新的连接，所以网关被看成IP 层的目的地，消息在网关处被解密，但是如果下一段网络还是基于TCP/IP的，传输之前用户数据和TCP 头部还要再次被加密。对于链路层的加密，除了链路头及附加信息外，所有的数据都被加密，只是在路由器和网关中才暂时恢复为明文。不同加密策略的加密内容如图6.1所示，阴影表示被加密。

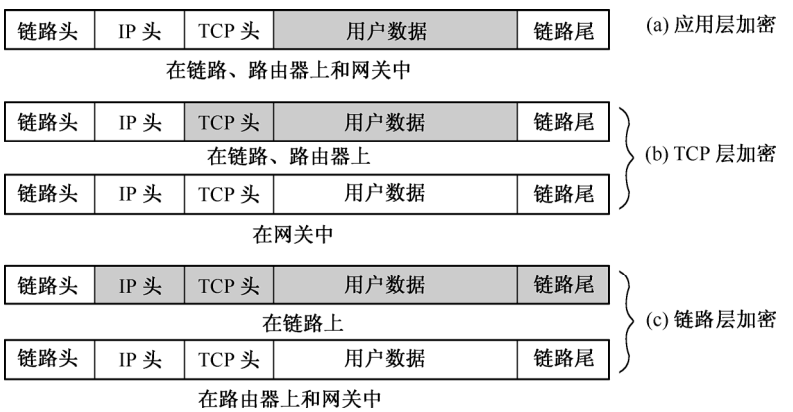


图 6.1　不同加密策略的加密内容

由于采用端到端的方式，消息头部是明文，数据通信可能遭到信息流分析。这在军事冲突中有明显的意义，通过信息流分析，可以得到如下几种信息：传输双方的标识，传输双方的联系频率，消息格式、消息长度或者消息的频繁交换对信息重要性的暗示，与传输双方的某些谈话相关的事件。信息流的分析是容易防止的，但要以牺牲效率为代价，传输填充是一种有效的对策，即在没有消息传送时，产生随机数加密传送。

从成本、灵活性和安全性来看，端对端的系统对那些要求有很多逻辑链路被保护的系统具有最大的吸引力。所以，下一节我们主要针对端到端系统来讨论密钥产生、存储和分配等密钥管理问题。用链路加密，密码设备是对实际链路的所有传输起作用。相反，端对端加密用于两个进程之间的“逻辑链路”信道。例如，对于网络层加密，可识别并被保护的实体数与网络末端系统数相对应。任何一个末端系统都可以进行加密数据的交换，只需要与对方末端系统共享一个密钥。在端对端加密的条件下，每对用户之间有一个虚的密码通道。

然而，对于某些网络和远距离处理设备的配置来说，链路加密比端到端的加密有着更大的吸引力。在用链路加密时，每个物理链路只用1个密钥，在保护的链路数目较少时，就只要求为数不多的链路加密设备。使用链路加密，用户一般并不知道消息正在接受密码保护，即密码功能由网络提供，对用户是透明的。另外，大多数链路加密设备，通常用硬件在物理层实现，不会显著导致传输性能的降低。

在某些情况下，同时使用两种方式，可以达到更多的安全性。

6.2　密钥管理

1883 年，Auguste Kerckhoffs 第一次阐明了密码系统的设计原则，密码体制应该是公开的，数据的安全性应仅仅依赖密钥的选择。如果密钥管理有薄弱环节，密码算法的强壮性就会减为零。所有的密码技术都依赖于密钥管理(Key management)，密码系统的设计者必须回答下列问题：系统中哪些节点要求密钥？如何将密钥装配到节点中，密钥的更换按照什么频率？系统在何处产生密钥？如何保护数据和密钥？这就是密钥管理的相关问题。密钥的管理本身是一个很复杂的课题，而且是保证安全性的关键点。密钥管理方法因所使用的密码体制(对称密码体制和公钥密码体制)而异。本章重点讨论对称算法的密钥管理(公开密钥的管理本章只做简单介绍，第 7 章会进行详细介绍)。

任何密钥都有一定的生存周期，也就是授权使用该密钥的周期。原因如下：

- 如果基于同一密钥加密的数据过多，这就使攻击者有可能拥有大量的同一密钥加密的密文，从而有助于他们进行密码分析。
- 如果我们限制单一密钥的使用次数，那么，在单一密钥受到威胁时，也只有该密钥加密的数据受到威胁，从而限制信息的暴露。
- 对密钥使用周期的考虑，也是考虑到一个技术的有效期，随着软硬件技术的发展，56 位密钥长度的 DES 已经不能满足大部分安全的需要，而被 128 位密钥长度的 AES 所代替。
- 此外，对密钥生存期的限制也限制了计算密集型密码分析攻击的有效时间。

一个密钥的生存期主要经历以下几个阶段：产生(可能需要登记)、分配、使用、更新/替换、撤销和销毁。《开放系统互联 安全体系结构》(GB/T 9387.2—1995 及 ISO 7498—2—1989)给出的密钥管理的定义是：在一种安全策略指导下密钥的产生、存储、分配、删除、归档及

应用。也就是说，密钥管理通常都是在一个特定安全策略的上下文环境下提供的，安全策略揭示了系统的威胁和风险，一般地说，包括密钥管理在技术和管理方面要遵守的程序，系统中相关实体的权限和责任，以及为支持安全事件的调查需要保留的记录类型。

密钥管理的目的是为了维持系统中各实体之间的密钥关系，以抗击各种可能的威胁，包括密钥的泄露、秘密密钥或公开密钥的身份真实性丧失和未经授权使用。

密钥管理处理密钥自产生到最终销毁的整个过程中的有关问题，包括系统的初始化，密钥的产生、存储、备份/恢复、装入、分配、保护、更新、泄露、撤销和销毁等内容。其中密钥的分配和存储是最棘手的。下面分别进行描述。

6.2.1 密钥的类型

我们把保护数据的密钥，叫做数据加密密钥(Data Encrypting Key)，当它保护的是两个通信终端用户的一次通话或交换数据时，也称为会话密钥(Session Key)。当它用于加密文件时，称为文件密钥(File key)。

数据加密密钥可由用户自己提供，也可由系统根据用户的请求产生，它们是被另一个密钥加密从而得到保护的。用于对会话密钥或文件密钥进行加密时采用的密钥称为密钥加密密钥(Key Encrypting Key)，又称辅助(二级)密钥(Secondary Key)或密钥传送密钥(Key Transport Key)。通信网中的每个节点都分配有一个这类密钥。

一个终端只要求存储一个密钥加密密钥，它称为终端主密钥。终端主密钥的保密是通过把它存储在一个叫做密码设施的保护区域里实现的。密码设施是一个安全部件，它包含通常的密码算法，以及放置少量的密钥和数据参数的存储器。只能通过一个不受侵害的接口对其进行存取。密钥变换和数据加密的基本操作都是通过密码设施完成的。

主机处理器里存储的密钥加密密钥的保密是通过在主机主密钥(Host Master Key)控制下的加密实现的。对主机主密钥的保护类似于终端主密钥，它存在于一个主机的密码设施中。

对于能够进行输入-输出操作的一些终端而言，终端用户是人。基本密钥(Base Key)，又称初始密钥(Primary Key)、用户密钥(User Key)，是由用户选定或由系统分配给用户的，可在较长时间(相对于会话密钥)内由一对用户所专用的密钥。

在公钥体制下，还有公开密钥、秘密密钥、签名密钥之分。

虽然密钥都承担着保护数据的功能，但各个密钥的生存期还是存在差别的。根据密钥的使用期限，可以把它们划分为长期密钥和短期密钥。长期密钥通常指主密钥、密钥加密密钥和用于完成密钥协定的密钥。短期密钥通常指数据密钥和用于一次会话的会话密钥。在通信环境中，会话密钥仅在两个节点要求交换数据时才存在，密钥的生存期是大约几分钟，也许一小时，但很少多于一天。加密的文件在节点之间传输时，密钥存在几天或几周，但用户保护档案数据的密钥可能存在几年，或者和文件保留的时间一样长。

一个会话密钥只在一次通信会话期间使用，会话完成时，该密钥就被删除或者重写，因此，每次会话都会使用不同的密钥进行通信，减少了单个密钥加密的数据量。

在数据库环境中，密码系统动态地采纳密钥，类似于会话密钥，把它赋给一个文件去使用，最终，不同的文件使用不同的文件密钥加密。

上述概念确定了一个密钥或者密钥保护的层次，大量的数据是用为数较少的动态产生的“数据加密密钥”来保护的，“数据加密密钥”是用为数更少的“密钥加密密钥”或“终端主

密钥”来保护的，而“密钥加密密钥”则是由“主机主密钥”来保护的。因此，只有少量的密钥以明文形式存储在密码设施内部，而其他的密钥以加密的形式存在于密码设施的外部，这个层次可由图6.2说明。

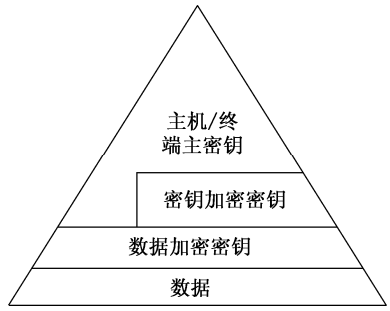


图 6.2　密钥保护的层次

6.2.2　密钥的产生和登记

密钥的产生可以是手工的，也可以是以自动化的方式产生。选择密钥方式的不当会影响安全性，比如选择不同的密钥产生方式，密钥空间是不同的，如表 6.1。当采用 4 个小写英文字母做密钥时，密钥空间仅为 4.6×10^5，如果采用每秒测试 100 万个密钥的硬件进行攻击，只需要 0.5 秒，而当采用任意的 8 个 8 位 ASCII 码字符做密钥时，密钥空间为 1.8×10^{19}，其搜索时间为 580 000 年，抵抗穷举攻击的时间大大增加。其次，差的选择方式易受字典式攻击。用户选择自己的密钥时，往往选择容易记忆的字符或者数字串，而攻击者也正是抓住了人们选择密钥上的这一心理特点进行攻击。攻击者并不按照数字或者字母顺序去尝试所有的密钥，首先尝试可能的密钥，例如英文单词、名字等。一个叫 Daniel Klein 的人，使用此法破译了40%的计算机口令。他采用如下方法来构造字典：

- 用户的姓名、首字母、账户名等与用户相关的个人信息。
- 从各种数据库得到的单词，如地名、山名、河名、著名人物的名字、著名文章或者小说的名字等。
- 数据库单词的置换，如数据库单词的大写置换，字母 o 变成数字 0，或者单词中的字母颠倒顺序。
- 如果被攻击者为外国人，可以尝试外国文字。
- 尝试一些词组。

表 6.1　不同产生方式时的密钥空间

	4 字节	5 字节	6 字节	7 字节	8 字节
小写字母(26)	4.6×10^5	1.2×10^7	3.1×10^8	8.0×10^9	2.1×10^{11}
小写字母 + 数字(36)	1.7×10^6	6.0×10^7	2.2×10^9	7.8×10^{10}	2.8×10^{12}
字母数字字符(62)	1.5×10^7	9.2×10^8	5.7×10^{10}	3.5×10^{12}	2.2×10^{12}
印刷字符(95)	8.1×10^7	7.7×10^9	7.4×10^{11}	7.0×10^{13}	6.6×10^{15}
ASCII 字符(128)	2.7×10^8	3.4×10^{10}	4.4×10^{12}	5.6×10^{14}	7.2×10^{16}
8 位 ASCII 字符(256)	4.3×10^9	1.1×10^{12}	2.8×10^{14}	7.2×10^{16}	1.8×10^{19}

一个好的密钥应该具有什么特点呢？好的密钥是某些自动过程产生的真随机位串，而真随机位串往往没有意义，不容易记忆，实际操作过程中建议采用以下原则或方式：

- 与特定算法相关的弱密钥和敏感密钥将被删除，例如 DES 有 16 个弱密钥和半弱密钥。
- 公钥体制的密钥必须满足一定的数学关系。
- 选用易记难猜的密钥，如较长短语的首字母，词组用标点符号分开，或者使用单向散

列函数将一个任意长的短语转换成一个伪随机串。例如，容易记忆的短语为，I like playing football and basketball，首字母为ILPFAB。如把“computer”用符号%分开，成为“com%put%er”。

- 不同等级的密钥，因为其重要性不同，也要选用不同的产生方式。
- 主机主密钥对系统中存储的所有密钥提供加密保护，可能在相当长的时间保持不变，其安全性至关重要，故要保证其完全随机性、不可重复性和不可预测性。可用投硬币、骰子，噪声发生器，随机数表等方法产生。
- 在一个有n个通信节点的系统中，为了得到较好的安全性，在每对节点之间都要使用不同的密钥加密密钥，需要的数量为$n(n-1)/2$，在节点数目较多时，需要的量非常大，可采用一些安全算法或者伪随机数发生器由机器自动产生。
- 会话密钥可利用密钥加密密钥及某种算法(加密算法，单向函数等)产生。

必须在安全环境中产生密钥以防止对密钥的非授权访问。密钥生产形式现有两种，一种是由中心(或分中心)集中生产，也称有边界生产；另一种是由个人分散生产，也称无边界生产。表6.2从两种生产方式的生产者、用户数量、特点、安全性和适用范围进行了对比。

如果是用户自己产生的密钥，那就需要进行密钥的登记。密钥登记指的是将产生的密钥与特定的使用捆绑在一起，例如，用于数字签名的密钥，必须与签名者的身份捆绑在一起。这个捆绑必须通过某一授权机构来完成。

表 6.2 两种密钥生产方式的对比

方 式	代 表	生 产 者	用 户 数 量	特 点	安 全 性	适 用 范 围
集中式	传统的密钥分发中心 KDC 和证书分发中心 CDC 等方案	在中心统一进行	生产有边界，边界以所能配置的密钥总量定义，其用户数量受限	密钥的认证协议简洁	交易中的安全责任由中心承担	网络边界确定的有中心系统
分散式	由个人生产		密钥生产无边界，其用户数量不受限制	密钥变量中的公钥必须公开，需经第三方认证	交易中安全责任由个人承担	无边界的和无中心系统

6.2.3 密钥的装入

为了安全的原因，主机主密钥可直接或间接装入，装入时需要有电磁屏蔽，装入后不能再读出，但可间接验证。直接装入是在密钥的进入地点和密码设施中的密码存储区之间用硬线连接，密钥的进入可借助于按钮开关、拨码盘等来完成。间接装入是首先把密钥读入主机处理器的主存储器中，然后用置主密钥的操作把密钥写入永久性的存储区中。

终端主密钥可直接或间接装入，装入时需要有电磁屏蔽，装入后不能再读出，可联机或者间接验证。可以借助开关或者号码盘，或者密钥加载装置把终端主密钥置位，或是在键盘上输入。联机校验的方法是和主机处理器建立一个通信会话。

会话密钥是随机产生的，不存在装入问题。

6.2.4 密钥的存储和保护

密钥除了需要保证其机密性外，所有密钥的完整性也需要保护，因为一个入侵者可能修改或替代密钥，从而危及机密性服务。

除了公钥密码系统中的公钥外，所有的密钥需要保密。

对于对称密码算法的密钥和非对称密码算法的私钥，如果只涉及单用户，密钥存储问题是最简单的，最安全的存储方法是将密钥记忆在脑子里，而不存储在任何系统中，用户每次在使用时才输入到系统中。而在实际中，无法记忆的密钥最安全的方法是将其放在物理上安全的地方。当无法用物理的办法对一个密钥进行安全保护时，必须用其他的方法来保护。一种方法是将一个密钥分成若干部分，委托给若干个不同的人或是放到若干个不同的地方，密钥的分存可以采用秘密共享的门限方案；另一种方式是通过机密性(例如，用另一个密钥加密)和/或完整性服务来保护。例如，可把密钥存储在 ROM芯片中或智能卡中，使用时再将其插入到计算机中，ROM 密钥的提取需要用户输入口令。具体使用时，还可将密钥分解成两部分，一部分存储到终端上，一部分作为 ROM 密钥。只要两者不同时丢失，密钥就不会泄露。美国的STH-III安全电话就是以这种方式工作的。除了极少数密钥(主机主密钥)以明文存储于有严密物理保护的密码器中，其他密钥都被(主密钥)加密后存储。通过机密性保护密钥，除了前面提到的密钥分层保护的方式外，也可以两种密码算法结合使用，比如用DES算法来加密保存RSA算法的私钥。

非对称密码算法的公钥存储可采用下列几种方式。一种是将所有公钥存储在专用媒体(软盘、芯片等)一次性发放给各用户，用户在本机中就可以获得对方的公钥，协议非常简单，又很安全。计算机黑客的入侵破坏，也只能破坏本机而不影响其他终端。这种形式只有在 KDC 等集中式方式下才能实现。第二种是用对方的公钥建立密钥环各自分散保存(如 PGP)。第三种是将各用户的公钥存放在公用媒体中。后两种都需要解决密钥传递技术，以获取对方的公钥。第三种还要解决公用媒体的安全技术，即数据库的安全问题。

6.2.5　密钥的分配

这里的密钥分配指的是在 n 个对等实体中，任意两个实体之间进行保密通信所需会话密钥的获得过程。

6.2.5.1　基于对称密码体制的密钥分配

对称密码体制的主要商业应用起始于 20 世纪 80 年代早期，特别是在银行系统中，采纳了DES标准和银行工业标准ANSI数据加密算法(DEA)。实际上，这两个标准所采用的算法是一致的。随着DES的广泛应用带来了一些研究课题，比如如何管理DES密钥。从而导致了ANSI X9.17 标准的发展，该标准于1985年完成，是有关金融机构密钥管理的一个标准。ANSI X9.17 是一个三层密钥层次结构：

(1) 主密钥(KKM)，通过手工分配。

(2) 密钥加密密钥(KK)，通过在线分配。

(3) 数据密钥(KD)，用于加密传输的数据。

其中，KKM 保护 KK 的传输，用 KK 保护 KD 的传输。主密钥是通信双方长期建立密钥关系的基础，必须通过安全信道进行分配，可以采用如下两种方式：

(1) 直接面对面分发或通过可靠信使递送。

(2) 将密钥分割成几部分分别传送，图6.3表示可以把密钥分成几部分，分别用信使、电话、信件等方式传送，在接收端把密钥组合起来。密钥分割也可以采用秘密共享的门限方案，把密钥分割成 n 份，使用其中的 $k(k<n)$ 份或 k 份以上就可以组合出密钥。

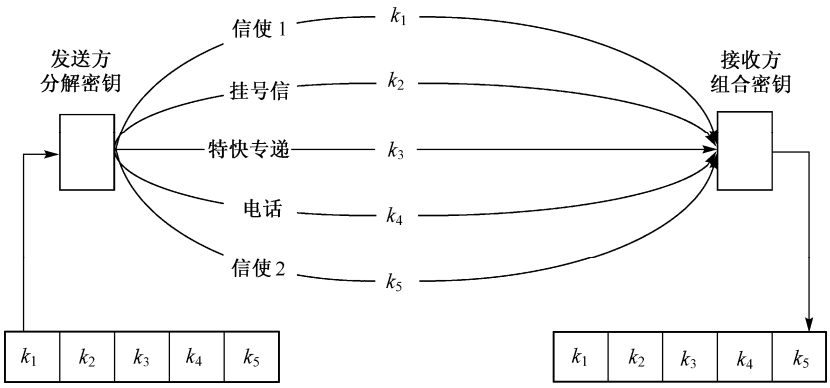


图 6.3 采用密钥分割的方式分发密钥

典型地，对称密钥的分配有两种方案，集中式和分布式。集中式分配是引入一个中心服务器(通常称为密钥分配中心或KDC)，在这个体系中，团体中的任何一个实体与中心服务器共享一个密钥。需要存储的密钥数量和团体的人数差不多，KDC 接受用户的请求，为用户提供安全的密钥分配服务。但是中心服务器必须随时都是在线的，如果服务器掉线，实体间的通信就不能进行了。服务器不仅成为通信的关键和攻击的焦点，而且是整个通信的瓶颈。典型的集中式分配方案的代表是Kerboros协议。分布式方案是指网络中的主机具有相同的地位，它们之间的密钥分配取决于它们之间的协商，比较著名的有 Diffie-Hellman 密钥交换协议，但Diffie-Hellman密钥交换协议没有提供鉴别机制，不能抵抗中间人攻击。两种体制的比较参见表 6.3。

表 6.3 两种密钥分配体制的对比

名 称	特 点	缺 点	代 表
集中式	集中式分配是引入一个中心服务器，通常称为密钥分配中心或 KDC，在这个体系中，团体中的任何一个实体与中心服务器共享一个密钥。在这样的系统中，需要存储的密钥数量和团体的人数差不多，KDC 接受用户的请求，为用户提供安全的密钥分配服务	动态分发时，中心服务器必须随时都是在线的	Kerboros 协议
分布式	网络中的主机具有相同的地位，它们之间的密钥分配取决于它们之间的协商	但 Diffie-Hellman 密钥交换协议没有提供鉴别机制，不能抵抗中间人攻击	比较著名的有 Diffie-Hellman 密钥交换协议

密钥分发技术有两种：静态分配和动态分配。静态分配是一种由中心以离线方式预分配的技术，是“面对面”的分发，安全性好，是长期沿用的传统密钥管理技术。静态分配只能以集中式机制存在，且必须解决密钥的存储技术。动态分配是“请求-分发”的在线分发技术，其分发及时，成本相对较低。有中心和无中心的机制都可以采用，需要有专门的协议支持。两种技术的对比参见表 6.4。

表 6.4 两种密钥分配技术的对比

名 称	特 点	优 点	缺 点
静态分配	是一种由中心以离线方式预分配的技术，是“面对面”的分发	安全性好，是长期沿用的传统密钥管理技术	必须解决密钥的存储技术
动态分配	是“请求-分发”的在线分发技术	分发及时，成本相对较低	需要有专门的协议的支持

静态配置是一种事先的预配置。在密钥管理中心(KMC)和各所属用户之间建立信任关系的基础上，密钥统一由 KMC 生成、分发、更换的集中式(Centralized)的管理体制。常见的模式有点对点配置、单层星状配置和多层星状配置。在点对点配置之下，如果要实现端对端的保密，两两用户之间需要共享一个密钥，n 个用户，需要 $n(n-1)/2$ 个共享密钥，为了解决大规模用户情况下，用户密钥急剧增加的困难，提出了星状配置。单层星状配置是网络密钥配置的基本形式。其密码网是由一个中心和n个终端构成的星状网。星状网的密钥配置在中心和终端(用户)之间用多个点对点的密钥实现。很多业务处理系统是星状结构，如银行的支付系统。在电子商务中商家和客户之间也构成星状结构，即 B2C。单层星状配置可直接推广到树形多层星状网中。网内是单层星状配置，而网间又构成新的二层星状网，各星状网保持各自的独立性。因为各星状网的业务相对独立，其密钥管理也相对独立，因此，可以成立分中心，分别管理。这对密钥管理带来很大方便，一个网的密钥配置和更新不影响其他网。

动态分配是一种请求分发的方式，在中心化的密钥管理方式中，由一个可信赖的联机服务器作为密钥分配中心(KDC)或密钥传递中心(KTC)。图 6.4 给出了其密钥分配的过程。在图 6.4(a)中，第一步，用户 A 向 KDC 请求为他和 B 的通信分发密钥；第二步，KDC 产生密钥 K，并把密钥 K 用 KDC 分别和 A、B 共享的密钥加密后发给 A；第三步，A 把收到的用 KDC 和 B 共享的密钥加密的密钥 K 转发给 B。图 6.4(b)与(a)不同的是，第二步，KDC 产生密钥 K，并把密钥 K 用 KDC 和 A 共享的密钥加密后发给 A；第三步，KDC 把密钥 K 用 KDC 和 B 共享的密钥加密后发给 B。在 KDC 分配密钥的方式中，A 和 B 的通信密钥 K 由 KDC 产生。在图 6.4(c)中，第一步，用户 A 产生密钥 K，并用 A 和 KTC 共享的密钥加密后发给 KTC；第二步，KTC 把密钥 K 用 KTC 和 B 共享的密钥加密后发给 A；第三步，A 把收到的用 KTC 和 B 共享的密钥加密的密钥 K 转发给 B。图 6.4(d)与(c)不同的是，第二步，KTC 把密钥 K 用 KTC 和 B 共享的密钥加密后发给 B。在 KTC 传递密钥的方式中，用户 A 和 B 的通信密钥 K 由一方用户产生。引入可信第三方的好处是：每一方只需要和可信第三方维护一个长期密钥。同时它也带来了如下缺点：中央节点成为安全的焦点和性能的瓶颈，中央节点的可靠性要求高，要求在线服务。

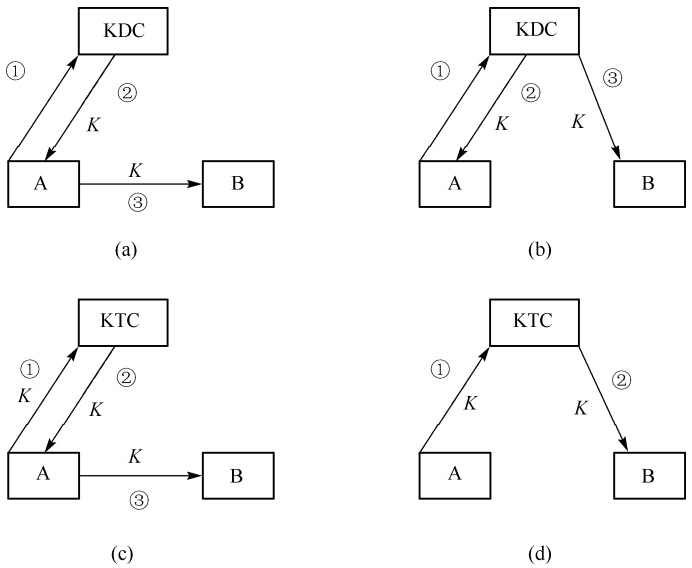


图 6.4　动态密钥的分配过程

对称密码具有一些很好的特性，如运行占用空间少，加解密速度比较快，但在密钥的分配上存在以下困难。

- 需要进行密钥的交换。对称密码依赖于这样一个事实，即双方通信之前，必须共享一些信息(密钥)，密钥必须通过一条安全的通道进行传输。这在某些情况下极为困难。
- 规模复杂。一个 n 个用户的团体，如果两两之间要相互通信，每个用户需至少保持 $n-1$ 个密钥，整个团体需要的密钥数量是 $n(n-1)/2$ 个，当 $n=1000$ 时，这个团体需要拥有的密钥量接近50万个，为了防止同一密钥加密太多的数据，需要定期更换，情况就更加复杂了。
- 未知实体通信。在两个没有任何接触的实体进行通信时，需要介绍人，这是对称密码和非对称密码都会碰到的问题，但解决方案截然不同。

6.2.5.2 基于公钥密码体制的密钥分配

公钥密码分配方案有多种，可以归纳为：公开宣布、公开可以得到的目录、公开密钥管理机构和公开密钥证书。

公开宣布的方法虽然很方便，但是一个主要的缺点是任何人都可以伪造这样一个公开告示。

利用动态的公开目录来维护公钥，可改善公钥的安全性。公开目录必须由可信实体或机构来运行，这种方式比个人当众公布要安全一些，但仍然有弱点。如果目录机构的私钥被泄露，那么，攻击者可以代替所有公钥，窃听所有来往信件。另一个弱点是，目录记录的损害和混乱招致系统瘫痪。

为了实现公钥的更安全的分发，应该采用比目录更严谨的公钥分发措施。一种方法是设置中心机构来维系公钥的动态目录。每一个参与者知道中心机构的公钥，而只有中心机构才掌握私钥。这种分发方式中心容易成为瓶颈，而且易受干扰和损害。

公钥证书是Kohnfelder于1978 年提出的，现在获得了广泛的应用。简单地说，公钥证书是由可信第三方 CA 签名的用户的公钥值(当然也包括一些其他信息)，是为了在庞大的用户群体中确保所有公钥的真实性和安全性，是 PKI 的重要组成部分。各种主流 Web 服务器(IIS, Apache)、浏览器(IE, Netscape)、E-mail 客户端几乎都支持 PKI 技术。在这种体系中，所有的安全功能(签名与签名验证、加密与解密)都可以离线实施。当然，一个实际运作的 PKI 系统并不是绝对离线，特别是证书作废列表(CRL)必须在线提供。随着 PKI 体系的庞大化，其离线优势可能逐步丧失。总体来说，PKI技术具有标准化、支持广泛、离线认证、应用层次灵活等诸多优点，技术上已经趋于成熟。

公钥技术在以下两方面优于对称技术：首先公钥密码算法的加密密钥——公钥不需要保密，不像对称密码算法的加解密密钥都需要保密，从而在一定程度上简化了密钥管理。此外，非对称密码体制的公钥分配，不需要联机的可信服务器。

6.2.5.3 Diffie-Hellman 密钥分配方案

人工手动分配密钥存在效率低、成本高、每个用户要存储与所有用户通信的密钥、安全性差的问题。人们希望可以采用机器自动分配的方式，要求自动分配密钥的协议能满足：任何两个用户能独立计算他们之间的秘密密钥，传输量小，存储量小，任何一个(或多个)用户不能计算出其他用户之间的秘密密钥。一个满足此要求的著名协议是 Diffie-Hellman 密钥交

换协议，双方选择素数 p 以及 p 的一个原根 a，然后按如下步骤进行：

(1) 用户 U 随机选择 $X_u \in Z_p$，计算 $a^{Xu} \bmod p$ 并发给 V。

(2) 用户 V 随机选择 $X_v \in Z_p$，计算 $a^{Xv} \bmod p$ 并发给 U。

(3) U 计算 $(a^{Xv} \bmod p)^{Xu} \bmod p = a^{XuXv} \bmod p$。

(4) V 计算 $(a^{Xu} \bmod \mathrm{p})^{Xv} \bmod p = a^{XuXv} \bmod p$。

最后，双方获得共享密钥 $(a^{XuXv} \bmod p)$，其基本模式参见图6.5。该协议必须结合实体鉴别才能使用，否则容易受到中间人攻击。它的两种改进协议是：端-端协议(Station-To-Station protocol，STS)和 MTI 协议。

假设每个用户的签名和验证算法记为 Sigu 和 Veru，可信中心的签名和验证算法记为 $\mathrm{Sig_{TA}}$ 和 $\mathrm{Ver_{TA}}$，每个用户有一个证书 C(U)= (ID(U)，Veru，$\mathrm{Sig_{TA}}$(ID(U)，Veru))，这里 ID(U) 是用户的身份信息。一个简化的端对端协议按如下步骤进行：

(1) 用户 U 随机选择 $X_u \in Z_p$，计算 $a^{Xu} \bmod p$ 并发给 V；

(2) 用户 V 随机选择 $X_v \in Z_p$，计算 $a^{Xv} \bmod p$，　$K=(a^{Xu})^{Xv} \bmod p = a^{XuXv} \bmod p$ 以及 $S_V = \mathrm{Sig_V}(a^{Xu} \bmod p \| a^{Xv} \bmod p)$，并把(C(V)，$a^{Xv} \bmod p, S_V$)发给 U；

(3) U 计算 $K=(a^{Xv})^{Xu} \bmod p = a^{XuXv} \bmod p$ ，并使用 Verv 验证 S_V，使用 $\mathrm{Ver_{TA}}$ 验证 C(V)；然后计算 $S_U = \mathrm{Sig_U}(a^{Xu} \bmod p \| a^{Xv} \bmod p)$，再把(C(V)，$S_U$)发给 V；

(4) V 使用 Verv 验证 S_U，使用 $\mathrm{Ver_{TA}}$ 验证 C(U)。

数字签名使得中间人攻击无法奏效，共享密钥 K（$= a^{XuXv} \bmod p$）也得到了验证。图 6.6 为 STS 协议的基本模式。

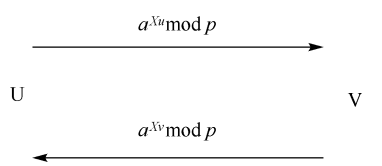


图 6.5　Diffie-Hellman 密钥交换协议的基本模式

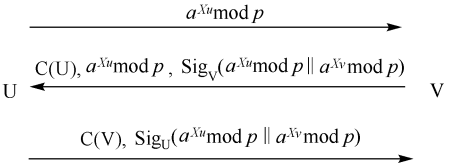


图 6.6　STS 协议的基本模式

MTI 协议是Matsumoto，Takashima 和Imai 通过修改Diffie-Hellman 密钥交换协议而得到，它不需要计算数字签名，但需要数字证书，是一个两段协议(two-pass protocol)，而STS 是三段协议。

假设：p 是素数，a 是 p 的原根，p 和 a 都公开。用户 U 的标识信息为 ID(U)，秘密指数 $X_u \in [0, p-2]$，公开值 $Y_u = a^{Xu} \bmod p$，数字证书是 C(U)=(ID(U), Y_u, $\mathrm{Sig_{TA}}$(ID(U)，Y_u))。

(1) U 随机选择 $R_u \in Z_p$，计算 $S_u = a^{Ru} \bmod p$ 并把(S_u，C(U))发给 V。

(2) V 随机选择 $R_v \in Z_p$，计算 $S_v = a^{Rv} \bmod p$ 并把(S_v，C(V))发给 U。

(3) U 从 C(V) 中获得 Y_v，并计算 $K = S_v^{Xu} Y_v^{Ru} \bmod p$。

(4) V 从 C(U) 中获得 Y_u，并计算 $K = S_u^{Xv} Y_u^{Rv} \bmod p$。

$$S_v^{Xu} Y_v^{Ru} = (a^{Rv})^{Xu} (a^{Xv})^{Ru} = a^{XuRv + XvRu} \bmod p$$

$$S_u^{Xv} Y_u^{Rv} = (a^{Ru})^{Xv} (a^{Xu})^{Rv} = a^{XuRv + XvRu} \bmod p$$

计算密钥需秘密指数 X_u, X_v，从而使得中间人攻击失效。图6.7 为 MTI 协议的基本模式。

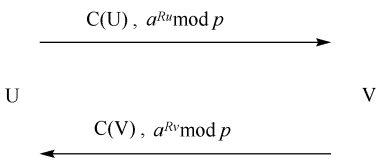


图 6.7　MTI 协议的基本模式

6.2.6 密钥的使用控制

密钥层次结构的概念和自动密钥分配技术的使用极大地减少了必须由人工方式管理的密钥的数量。对于自动分配的密钥的使用方式施加某些限制也是必要的。考虑到密钥加密密钥和数据加密密钥的重要性不同，用于签名和加密的公私钥以及用于消息加密和消息鉴别的秘密密钥具有不同的安全要求，如果对它们误用会带来一定危险，这就需要确保打算用于一种目的的密钥不能和用于另一种目的的密钥混用，将密钥值和密钥的合法使用范围绑定在一起。与密钥的使用联系在一起的信息有：

(1) 密钥的拥有者。

(2) 有效期限(期望的使用时间)。

(3) 密钥的标识符。

(4) 密钥所对应的特定算法。

(5) 特定的使用环境和用户。

(6) 密钥产生、注册和认证的实体。

(7) 完整性检查。

如果控制信息易于操作，可以使用公钥证书或加密技术提供完整性。保证不同用途的密钥通过密码手段隔离的技术称为**密钥隔离**。密钥隔离的简单方式是使用密钥标签(Tag)，密钥标签是一个位向量或结构性区域，与密钥加密绑定在一起，因为标签的长度有限，并且不能采用明文形式传送，限制了它的灵活性和功能。还有一种被称为控制向量的方案，可以提供更大的灵活性。

6.2.7 密钥的撤销和销毁

当与密钥有关的系统已经迁移，或者怀疑一个特定的密钥已受到威胁，密钥的使用目的已经改变(如提高安全级别)，就必须进行密钥的撤销。

密钥的销毁指清除一个密钥的所有踪迹。

一个密钥的值在被停止使用后可能还要保密一段时间，例如，一条加密数据流包含的信息可能仍然需要保密一段时间。为此，使用的任何密钥的秘密性都需要保持到所保护的信息不再需要保密为止。

在密钥的使用活动终结后，安全地销毁所有敏感密钥的副本是十分重要的。例如，必须使得一个攻击者通过观察旧的数据文件、存储的内容或抛弃的设备确定旧密钥值是绝不可能的。

6.2.8 密钥的备份/恢复和更新

假设某单位的一些秘密资料经加密后保存在负责人 A 那里，A 很尽职尽责，从未将密钥泄露给其他人，这些资料无疑是安全的，但如果 A 意外死亡，这些文件如何恢复呢？这里可以采用以下两个解决方案。

1．使用秘密共享协议

当 A 生成一个密钥时，他把它分成 n 份，每一份分给不同的职工，任何一个职工单独不能生成 A 的密钥，但只要把其中的 $k(k<n)$ 份合在一起，便可恢复。这样，当 A 出现意外时，单位领导可以召集 k 个持有部分密钥的人，恢复他的密钥。

2．使用智能卡暂存

A 出差时，他可以把密钥存储在智能卡上，交给领导，领导可以使用智能卡，但无法读出密钥的内容。同时，当 A 返回时，他可以查看别人是否用了他的卡以及使用了多少次。

有时加密密钥需要每天更新，一种简单的做法是从旧的密钥生成新的密钥。虽然如果攻击者知道了旧密钥及更新方法，他也可以导出新密钥，但他不可能容易地通过传输过程获得新密钥。

6.3　密钥的长度

密码算法的安全性与采用对称还是非对称体制没有关系，其安全性与密钥长度相关。密钥长度越长，安全性就越高，但同时计算代价也在增加。密钥长度的选择原则是在安全性和计算代价之间的折中，既保证系统的安全性，又不至于开销太高。

6.3.1　对称算法的密钥长度

假设算法的保密强度是足够的，除了穷举攻击外，没有更好的攻击方法，穷举攻击是已知明文攻击。穷举攻击的复杂度为：密钥长为 n 位，则有 2^n 种可能的密钥，因此需要 2^n 次计算。表 6.5 给出了不同密钥长度的情况下，假设每秒测试 100 万个密钥，穷举攻击所需要的时间。

表 6.5　穷举攻击的运算量

密 钥 位 数	试 验 次 数	时间(年)
8	2^8	
56	2^{56}	2285
64	2^{64}	585 000
128	2^{128}	10^{25}
2048	2^{2048}	10^{597}

可见，当密钥长度足够长时，将使穷举攻击的时间太长而无法成功。

密钥的搜索速度由计算机资源确定。串行搜索下，由计算机的计算时间，即时间复杂性决定。并行搜索由空间复杂性和各计算机的计算能力决定。增大空间复杂性可减少时间复杂性。

在第 3 章曾提到，Session 和 Wiener 在 CRYPTO’93 上给出的密钥搜索芯片每秒能测试 5000 万个密钥，用 5760 个芯片组成的系统需要花费 10 万美元，它平均用 1.5 天左右就可找到 DES 密钥。如果使用 10 个这样的系统将花费 100 万美元，但可将平均搜索时间降到 3.5 小时左右。

一种统计观点是，某种计算资源破译需要一年时间，在一个月内破译的可能性为 8%，若穷举需要一个月时间，则一小时内破译的可能性是 0.14%。

从上述讨论得知，如果没有特殊的硬件和整体的并行机制，穷举攻击是难以成功的。除了使用特殊硬件之外，还可以采用其他方法。

在 1997 年 RSA 安全年会上破译 DES 密钥的挑战赛中，美国科罗拉多州的程序员 Verser 用了 96 天时间，在 Internet 上数万名志愿者的协同工作下，成功地找到了 DES 的密钥。Verser 实际上是利用互联网发动了一场对 DES 穷举攻击的人民战争，这种方式无须花费什么代价建立专门的攻击硬件。研究表明，计算机在 70%～90%的时间内是空闲的，如果攻击者编制一

种利用计算机空闲时间进行攻击的病毒，原理上应是可以成功的，只是这种方式比较慢，构成的威胁并不大，但是随着计算机计算能力的提高，会使上述攻击变得更有效。

确定密钥长度时，设计者应从所保存信息的价值、信息保密的时间、信息的攻击者及其使用的设备和资源的情况等几个方面考虑，表 6.6 给出了对各种信息的保密时间的估计。

表 6.6 不同信息的保密时间

信息类型	保密时间
战略军事信息	数分钟/小时
企业短期营利信息	几天/周
企业长期营利信息	几年
商业秘密	几十年
外交秘密	65 年以上
美国统计数据	100 年

6.3.2 公开密钥密码体制的密钥长度

公开密钥密码体制是基于数学难题的，主要是大数因子分解和有限域中求解离散对数这类难题。

RSA 算法的密钥长度取决于因数分解的时间。表 6.7 给出了使用通用数域筛法和特殊数域筛法分解因数时间(单位为 MIPS 年)。

表 6.7 使用通用数域筛法和特殊数域筛法分解因数时间

所需 MIPS 年 / 十进位 / 分解方法	512	768	1024	1280	1536	2048
通用数域筛法	300 000	2×10^{8}	3×10^{11}	1×10^{14}	3×10^{16}	3×10^{20}
特殊数域筛法	<200	100 000	3×10^{7}	3×10^{9}	2×10^{11}	4×10^{14}

因此，在选择公开密钥密码算法时，既要看系统需要保证的安全性，又要看密钥的生存期，还要看因数分解的能力。根据不同情况下的信息的不同价值，并假设计算机的计算能力每 5 年提高 9 倍，Bruce Schneier 给出的公开密钥长度建议值如表 6.8 所示。Ron Rivest 给出的密钥长度建议值比较乐观，如表 6.9 所示。

表 6.8 Bruce Schneier 公开密钥长度建议值

年度	对于个人	对于公司	对于政府
1995	768	1280	1536
2000	1024	1280	1536
2005	1280	1536	2048
2010	1280	1536	2048
2015	1536	2048	2048

表 6.9 Ron Rivest 的乐观的密钥长度建议值

年度	较小值	平均值	较大值
1990	398	515	1289
1995	405	542	1399

（续表）

年　度	较 小 值	平 均 值	较 大 值
2000	422	572	1512
2005	439	602	1628
2010	455	631	1754
2015	472	661	1884
2020	489	677	2017

6.3.3　密码体制密钥长度的对比

对称密码体制和公开密钥密码体制均有各自的优缺点，对称密码算法运算速度快、密钥短，密钥管理困难(分发、更换)，通信方是平等的(不能为发送者提供保护)。非对称密码算法，只需保管私钥，可以相当长的时间保持不变，需要的密钥数量较小，运算速度慢，密钥尺寸大，通信方是不平等的，因为加密消息和验证签名的人不能解密同一信息和产生同样的签名。因此，目前的保密系统采用了两种密码体制结合的方式，在两种或两种以上密码体制混用的系统中，密钥长度如何确定呢？

根据安全的“木桶原理”，一个安全系统经常在最薄弱的地方被攻破。因此，在设计基于密码的安全系统时，要使用安全性能相当的密码算法。否则，攻击者就会从安全性较弱的密码算法进行攻击。表 6.10 给出了 NIST 推荐的多种密码算法的安全性对比关系。可以看到，3072 bit 的 RSA 或 DSA, 128 bit 的对称加密与 256 bit 的 ECC 的加密强度相同。

表 6.10　NIST 公布的密码安全性对比关系

对比加密强度						
	比特位大小					
对称算法(密钥比特位)	56	80	112	128	192	256
Hash 算法(消息摘要的长度)	160		256		384	512
MAC(MAC 的长度)	64	160	256		384	512
RSA/DSA(模数比特位)	512	1024	2048	3072	7680	15 360
ECC(*n* 的比特位)	112	160	224	256	384	512

6.4　硬件加密和软件加密

尽管软件加密在今天已经非常流行，硬件加解密还是商业或军事上的主流，这基于以下原因。

首先是速度问题。加密算法有许多针对位的操作，而这些操作又不是由运算器直接运行的。此外，加密运算是计算密集性的，占用计算机的主处理器去执行加密任务显然不是有效的做法，使用专用硬件实现，可以使系统更快。

第二个原因是安全性，运行在一般计算机上的加密算法没有物理保护，容易被攻击者修改。硬件加密装置可以被安全地封装起来，防止被篡改。

第三个原因是硬件易于安装，不需要使用计算机。如果需要加密通信链路，只需要在电话、传真或Modem中安装一个专用硬件，比在计算机中放置相应软件更容易；即使在计算机环境下，使用硬件加密可对用户透明，而软件实现要做到这一点，需要写入操作系统底层，这是不容易实现的。

任何加密算法都可用软件实现，它的缺点是速度慢、造价高、安全性差，优点是使用灵活、修改方便、可移植性好。采用软件加密时，密钥管理的手段必须可靠，密钥和明文应在加密后删除。

6.5 存储数据加密的特点

用于保护存储数据的加密，与用于保护传输数据的加密不同，数据存放的时间很长，可能几年才需要解密，因此，密钥需要安全、长期地存储。此外，对于存储数据的加密，可能还会存在以下问题：

(1) 数据可能在另外的盘上、计算机上以明文形式出现，为攻击者提供了“已知明文攻击”的机会。

(2) 数据库的应用中，数据小于加密分组的长度，造成数据扩展。

(3) I/O 设备的速度要求高速的加、解密运算。

(4) 密钥的管理更复杂，因为不同的人需要访问不同的文件或同一文件的不同部分。

6.6 压缩、纠错编码和加密

将压缩算法和加密算法结合使用效果更好。一方面，密码分析依赖于明文中的数据冗余，而压缩明文会减少这种冗余。另一方面，加密运算比较耗时间，压缩明文会减少加密时间。

有一种编码叫纠错码，它具有错误纠正的能力，如果传输中有一位或少数位被改变，纠错码具有足够的冗余信息可以确定并纠正错误位。如果用户打算在传输的信息中增加纠错编码，纠错码应在加密之后。当纠错码用在加密之后时，它显然具有错误纠正的能力。如果使用内部差错控制，把纠错码放在加密之前，当传输中的一个比特或少量的比特发生改变，由于雪崩效应，解密操作会使相应明文的多位发生改变，使得差错控制由于误差太多而无法完成纠错功能。压缩、纠错编码和加密的正确顺序如图6.8 所示。

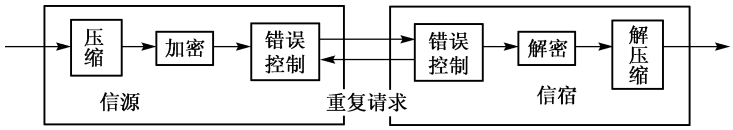


图 6.8 压缩、纠错编码和加密的顺序

6.7 文件删除

大多数计算机删除文件时，只是删除文件的索引，此外，虚拟存储器意味着计算机可以在任何时候往硬盘上读、写数据。要想彻底删除数据只有使用多次物理写入的办法。

- 美国 NCSC[NCSC-TG-0257]建议以一定格式的随机数重写至少 3 次。第 1 次用 00110101…，第2次用 11001010…，第3次用随机数 10010111。
- 一些商用软件使用3次重写，第1次全1，第2次全0，第3次用 0 和 1 相间的数字
- Schneier 建议用7次，第1次用全1，第2次用全0，后5次用密码学上的安全伪随机数。
- 偶尔对未用空间重写，或将文件与硬盘未用部分进行交换。

NIST 对电子隧道显微镜的研究表明即使多次重写也是不够的，对存储介质进行物理损毁才是最安全的办法。

6.8　关于密码的一些教训

回顾一下这些密码使用过程中的小故事，对于我们正确使用密码，获得有效的安全性，是很有意义的。下面的这些错误，有很多甚至是著名的公司和人物所犯的。

6.8.1　声称你的算法是“不可攻破的”

1930年，德国官员声称他们的密码机 Enigma 是不可攻破的，第二次世界大战时，Luftwaffe 最高指挥部给战场指挥官发布了一条消息，保证 Enigma 是不可攻破的。但是我们怎么知道这样一条消息被送出了呢？因为在它发出不久以后，英国的密码破译者就截获并解密了它。

1977 年，RSA 的作者为 RSA 算法的公开发表准备了一道挑战性的题目，选择了一个 129 位的合数 n，加密了一条消息，为任何能够分解 n 并破译出原文的人提供100美元，文章发表在 8 月的《科学美国人》杂志上。为了证明他们的确有秘密钥匙，三位教授还提供了一个数字签字，签字用公开密钥解密后，为“FIRST SOLVER WINS ONE HUNDRED DOLLARS”（首位破译者可以赢得 100 美元）。他们声称破译需要 40 quadrillion（百万之四次方）年，但是仅仅 17 年后，他就有幸再次读到自己加密的消息“THE MAGIC WORDS ARE SQUEAMSIH OSSIFRAGE”。

6.8.2　多次使用一次性密码本

20 世纪 30 年代到 20 世纪 40 年代，前苏联使用一次性密码本加密消息，1942 年前苏联密码中心偶然打印了一份密码本的副本，1943 年美国的密码分析家发现了这一问题，从而能够破解前苏联 1942 年到 1948 年间发送的消息。

1998 年，微软发布了PPTP协议，使用 RC4 连接客户与服务器端，但两个方向却使用同样的密钥。

6.8.3　没有使用最好的可能算法

现在有很多加密算法，有些是免费的，有些不是。微软设计操作系统 NT 时，提出了一个保护口令的方式“LANMAN”，涉及对口令做消息摘要，该算法存在明显缺陷，破译者可以几秒钟内攻破。后来微软修正了这一错误，但在下一个改进版本中仍然使用的是容易被攻破的 MD4 算法，虽然，当时已经有了 MD5 算法。

6.8.4　没有正确实现算法

Sun 公司发布 JDK1.1 时，使用了 DSA，其中涉及随机数 k，为了保证私钥不被泄露，不能使用同一个 k 签名两个不同的消息，但是该程序却使用了一个固定的 k，虽然 JDK1.1.2 对此进行了修正，但计算一次签名需要 4～5 秒，而正常的签名只需要几毫秒。

6.8.5　在产品中放置了后门

Clipper 是美国国家安全机构为语音和其他信息的加密而开发出来的一款芯片。然而它还带有一个公开的陷门来支持政府对任何使用Clipper 芯片的计算机进行窃听，并预先约定政府拥有窃听许可。但该芯片终因侵犯公众隐私，没有获得成功。

思考和练习题

(1) 为什么所有的密钥都有生存期？密钥的生存期主要有哪些阶段？
(2) 对称加密算法的秘密密钥的保护方式有哪些？
(3) 请举例说明集中式和分布式密钥分配体制的优缺点。
(4) 请说明在中心化的密钥管理方式中，KDC 和KTC 作用的异同，可信第三方有哪几种工作模式？
(5) 请说明 Diffie-Hellman 密钥交换协议的步骤，它会受到什么攻击，如何改进？
(6) 为什么要进行密钥隔离？
(7) 如果采用 RSA 算法来传递 DES 算法的密钥，合适的 RSA 算法的模 n 长度应为多少？
(8) 纠错码、压缩和加密如果要共同用于通信系统，合适的顺序是什么？为什么？
(9) 如何进行彻底的文件删除？
(10) 你认为本章的哪些内容对你日常安全使用密码有帮助？请总结说明。

实践/实验题

收集一些密码实际使用中的经验教训，可以是自己的，也可以是别人的。

第 7 章　公钥管理技术

公钥密码能够比对称密码提供更多的功能，或以更简单的方式提供机密性、完整性等服务。公钥密码能够提供数字签名功能，从而实现不可否认服务，公钥体制允许远程的两个实体之间通过协商获得所需要的对称加密密钥。如果 Alice 以前和 Bob 没有任何意义上的接触，她也能在一个公开的地方查阅到Bob的公钥，但她如何确信她所找到的是Bob的公钥呢？公钥的分发虽然不需要保密，但需要保证公钥的真实性。本章将探讨如何实现公钥密码应用中的公钥分发？介绍公开密钥管理的若干方法和途径。

7.1　公开密钥基础设施

7.1.1　PKI 概述

公开密钥基础设施(Public Key Infrastructure，PKI)的基本定义十分简单，所谓 PKI 就是一个用公钥概念和技术实施和提供安全服务的具有普适性的安全基础设施。PKI 提供的主要安全服务有：实体鉴别、完整性、机密性和抗抵赖。PKI 是一种标准的密钥管理平台，它通过第三方的可信任机构——认证中心(Certification Authority，CA)，生成用户的公钥证书，把用户的公钥和用户的其他标识信息(如电子信箱、手机号码等)绑定在一起，从而能够为所有采用加密和数据签名等密码服务的网络应用提供所必须的密钥和证书管理。一个有效的 PKI 系统必须是安全和透明的，用户在获得加密或数字签名服务时，不需要详细地了解 PKI 是怎样管理证书和密钥的。PKIX(Public Key Infrastructure using X.509)工作组给 PKI 的定义为：“是一组建立在公开密钥算法基础上的硬件、软件、人员和应用程序的集合，它应具备产生、管理、存储、分发和废止证书的能力”。一个典型、完整、有效的 PKI 系统如图 7.1 所示，包括认证机构(CA，也称为认证中心)、证书注册机构(Registration Authority，RA，也称为注册中心)、密钥和证书管理系统、PKI 应用接口系统、PKI 策略和信任模型。

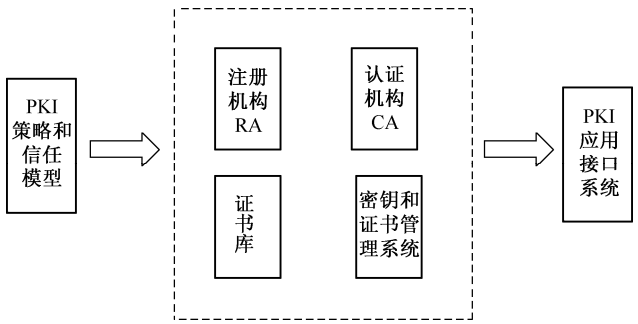


图 7.1　典型 PKI 系统组成

7.1.1.1 注册中心 RA

尽管注册功能可以直接由 CA 来实现，专门用一个单独的机构来实现注册功能也是很有必要的，它的主要目的就是分担 CA 的功能，作为 CA 和最终用户之间的接口，增强 CA 系统的安全性。例如，一个 PKI 域的终端用户很可能在地理位置上是分散的，为了终端用户的方便，需要建立多个本地 RA（Local RA）代理接收注册信息。RA 与 CA 功能的分离，也使 CA 能够以一个离线实体的身份来运作，减少受到外部攻击的可能性。RA 的主要功能包括：接收和验证新注册人的注册信息，对证书申请者的信用度、申请证书的目的、身份的真实可靠性等问题进行审查，确保证书与身份绑定的正确性；代表最终用户生成密钥对；接收和授权密钥备份和恢复请求；接收和授权证书吊销请求；按需分发或恢复硬件设备，如令牌。

7.1.1.2 认证中心 CA

CA 是证书的签发机构，它是PKI的核心。CA对任何一个主体的公钥进行公证，通过签发证书把主体与公钥绑定。它的主要功能包括：验证并标识证书申请者的身份；确保CA用于签名证书的非对称密钥的质量, 为了防止被破译，CA用于签名的私钥长度必须足够长并且私钥必须由硬件卡产生，私钥不能从卡内读出；确保整个签名过程和签名私钥的安全性；证书材料信息（如公钥证书序列号、CA等）的管理；确保证书主体标识的唯一性，防止证书主体名字的重复；确定并检查证书的有效期限，保证不使用过期或已作废的证书；发布并维护撤销证书列表（Certificate Revocation List, CRL），证书在有效期之内由于某种原因需要作废、终止使用时，通过证书撤销列表CRL 来实现，CRL 一般存放在目录系统中，以供交易时在线查询，防止交易风险；对整个证书签发过程做日志记录，以备发生争端时，提供公正依据，参与仲裁；向申请人发通知。

其中最重要的是CA 自己私钥的管理，它必须确保其高度的机密性，防止他方利用其私钥伪造证书。CA 自己的公钥证书在网上公开，通过自签名的方式，保证 CA 公钥的真实性。其他人可以利用CA 的公钥验证 CA 的数字签名，保证了证书（实质是持有者的公钥）的合法性和权威性。

7.1.1.3 证书库

证书库是一种网上公共信息库，用于存储CA 已签发证书及公钥、撤销证书及公钥，可供公众进行开放式查询。一般来说，查询的目的有两个：一是获得对方的公钥；二是验证对方的公钥证书是否已经作废，即该证书是否已经不再被使用。证书库可以是关系数据库，也可以是目录（目录服务器）。通常用做 PKI 组成部分的证书库是目录，有时是 X.500 的目录，更常见的是 LDAP（Light Directory Access Protocol，轻量级目录访问协议）目录。

7.1.1.4 密钥和证书管理系统

密钥和证书管理涉及密钥备份与恢复、自动密钥更新和建立密钥历史档案。

如果用户的解密私钥丢失，密文无法解密，就会造成数据丢失。为了防止这一问题，提供密钥的备份和自动恢复是非常重要的，CA 将在安全的数据库中存放解密私钥的备份。

当证书被颁发时，会被赋予一个固定的有效期，当证书接近过期时，必须颁发新的公钥证书，这称为密钥更新。

每次更新密钥后，“旧”的密钥和证书都应该存档，以便将来恢复用旧密钥加密的数据或者用旧密钥验证以前的签名。经过若干时间以后，每一个用户都会拥有多个旧证书和一个

当前证书。这些旧证书及相应的私钥就组成了用户密钥和证书的历史档案。密钥历史档案的管理也是 PKI 自动完成的。

7.1.1.5　应用接口系统

一个完整的 PKI 还必须提供良好的应用接口系统，使得各种应用能够安全、透明地与 PKI 交互，具有可扩展性和互操作性。PKI 应提供对文件传送、文件存储、电子邮件、电子表单、Web 等多种应用需要的安全服务的支持，尽可能地向上层应用屏蔽密码服务的实现细节，向用户屏蔽复杂的安全解决方案，使密码服务对用户而言简单易用，并且便于机构完全控制其信息资源。为了满足系统不断发展的需要，证书库和 CRL 要有良好的可扩展性。不同企业、不同单位的PKI实现可能是不同的，必须支持多环境、多操作系统的PKI的互操作性。

7.1.1.6　PKI 策略和信任模型

一个完整的 PKI 还包括认证策略的制定(包括遵循的技术标准、各 CA 之间的上下级或同级关系、安全策略、安全程度、服务对象、管理原则和框架等)、认证规则、运作制度的制定、所涉及的各方法律关系内容以及技术的实现等。建立一个全世界所有用户都信赖的全球性 PKI 是不现实的，比较可行的方案是：建立多个独立运行的 PKI，为不同地理环境和行业的用户团体服务。这就涉及不同 PKI 管理的用户之间的交叉认证问题，已经提出的 PKI 信任模型有：层次模型、网状模型、桥 CA 模型等。

7.1.2　数字证书

7.1.2.1　数字证书的格式与内容

数字证书的概念是Kohnfelder 于 1978 年提出的。数字证书，就是公钥证书，是一段包含用户身份信息、用户公钥信息以及一个可信第三方认证机构 CA 的数字签名的数据。CA 的数字签名可以确保证书也就是用户公钥信息的真实性。

证书可以放在一个非安全的容器中。只要用户信任CA，无论从何处获得证书，只要使用 CA 的公钥验证 CA 对证书的签名是正确的，就可以放心地使用证书中的公钥。

X.509 证书已被广为接受，应用于许多网络安全领域，其中包括 IPsec, SSL/TLS, SET 和 S/MIME。X.509 证书格式的版本到目前为止一共有三个，通常用 V1, V2 和 V3 表示。1988 年国际电联 ITU-T 在 X.500 标准的基础上制订了 X.509 vl 标准。1993 年新版本的 X.509 v2 标准充实了原先的版本，细化了证书、CA 等重要的概念，并修改了上一版的 CRL 格式。1997 年制订了X.509 v3 标准，2000 年发布了 X.509 v4，证书格式版本未变，只是增加了公钥证书扩展项和对属性证书的描述，CRL 格式也没有变化，保持为 v2。这种证书扩展项可以保存任何类型的附加数据。图7.2给出了 X.509 证书结构，以及一个 X.509 v3 格式的证书示例。X.509 证书各个字段的含义如下：

- **证书版本号(Version number)**：说明证书的版本号，现在合法的版本有 1, 2 和 3，分别代表 X.509 版本 l，版本 2 和版本 3。
- **证书序列号(Serial number)**：是由证书签发者分配给证书的唯一数字标识符。当证书被撤销时，实际上是将此证书的序列号放入由 CA 签发的 CRL 中，这也是序列号唯一的原因。

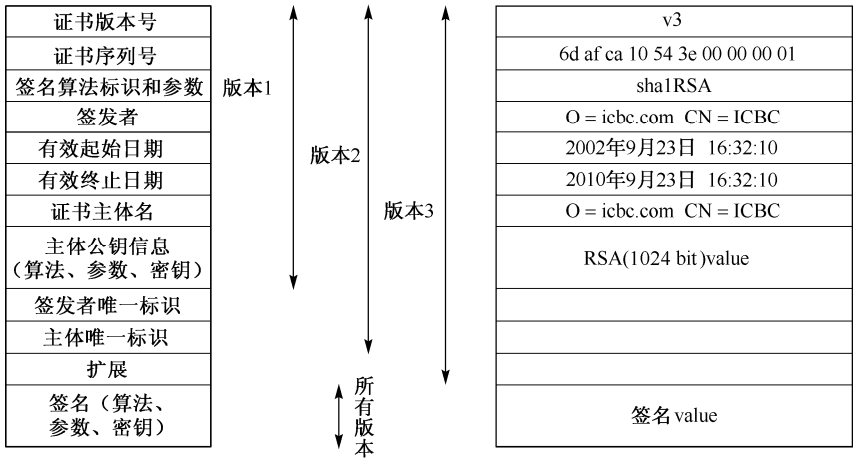


图 7.2 X.509 证书结构及示例

- **签名算法标识和参数(Signature algorithm identifier)**：签名算法标识用来指定由 CA 签发证书时所使用的数字签名算法，包含公开密钥算法和散列算法，由对象标识符加上相关参数组成。例如，对象标识符 SHA-1 RSA 就用来说明该数字签名是利用RSA 对 SHA-1 生成的消息摘要签名。
- **签发机构名(Issuer name)**：符合 X.500 标准的签发该证书的 CA 实体的名称。包括国家、省市、地区、组织机构、单位部门和通用名。
- **有效期(Period of validity)**：是CA授权维持证书状态的时间间隔，由证书开始生效的日期和时间(Not valid before)和失效的日期和时间(Not valid after)这两个日期表示。他们分别由 UTC 时间或一般的时间表示，在 RFC2459 中有详细的时间表示规则。每次使用证书时，需要检查证书是否在有效期内。
- **证书主体名(Subject name)**：指定证书持有者的X.500 唯一名字。包括国家、省市、地区、组织机构、单位部门和通用名，还可包含 E-mail 地址等个人信息等。此字段必须是非空的，除非使用了其他的名字形式。
- **主体公钥信息(Subject's public key information)**：此域包含两个重要信息：证书持有者的公开密钥的值；公开密钥使用的算法标识符，此标识符包含公开密钥算法和参数。该项是必须说明的。
- **签发者唯一标识(Issuer unique identifier)**：签发者唯一标识在第 2 版加入证书定义中。此域用在当同一个 X.500 名字用于多个认证机构时，用 1 比特字符串来唯一标识签发者的 X.500 名字，是隐含且可选的，实际中此项用得较少，并且不被 RFC2459 推荐使用。
- **主体唯一标识符(Subject unique identifier)**：主体唯一标识符在第 2 版的标准中加入 X.509 证书定义。此域用在当同一个 X.500 名字用于多个证书持有者时，用 1 比特字符串来唯一标识证书持有者的 X.500 名字，是隐含且可选的，实际中此项用得较少，并且不被 RFC2459 推荐使用。
- **签名(Issuer's signature)**：证书签发机构对证书上述内容的签名值。包含证书发行者对该证书的签名、用于证书签名的算法标识符和所有相关参数。

- **扩展域**：为证书提供了携带附加信息和证书管理的能力。每一个扩展字段包括三个域：扩展类型、扩展值和关键状态指示。扩展类型字段定义扩展值的语法和句法。扩展值字段包含扩展域的实际数据，它由扩展类型字段所描述。关键状态指示是一个标识位，只有两个状态：关键状态和非关键状态。通知一个使用证书的应用程序在不能识别一个扩展字段时，如果忽略一个扩展域是否安全。非关键状态表示可以忽略此类扩充，若为关键，则证书必须在此扩充类下应用才安全。证书扩展项主要有三类：密钥及其策略信息、证书路径约束、证书主体及签发者属性信息。

7.1.2.2　数字证书的类型

实际应用中，数字证书可以颁发给服务器、计算机、个人、电子邮件地址、企业、软件等。根据证书的持有者，数字证书可以分为如下类型：CA证书、服务器证书、个人证书、企业证书、安全电子邮件证书、安全电子交易(Secure Electronic Transaction，SET)证书、代码签名(Object Signing)证书和 WAP 证书等。

CA 是可信的第三方机构，负责验证证书申请者的身份并签发证书，其自身的证书由上层 CA 签发或由其自签(Self-Signing)。CA 证书主要用于验证由其签发证书的真实性和完整性。

个人证书，提供证书持有者的个人身份信息、公钥及CA的签名，用于在网络中标识证书持有者的个人身份。比如开展网上业务的银行 CA 给客户签发 SSL 客户证书，允许客户使用证书访问银行应用服务器进行在线交易。服务器首先根据客户证书对其身份进行验证，并且确定客户是否有权访问其账户信息。

服务器证书，提供服务器信息、公钥及CA的签名，用于在网络中标识服务器软件的身份。比如，SSL服务器证书，与SSL客户证书相对应，用于SSL连接时客户对服务器进行身份鉴别。

安全电子邮件证书，提供证书持有者的电子邮件地址、公钥及CA的签名，用于电子邮件的加密和签名。

SET 证书，用于解决安全电子交易中各方的身份验证问题。为了保证 Internet 上电子交易的安全性，防范交易及支付过程中的欺诈行为，除了在数据传输过程中采用加密等措施外，还必须建立一种信任及信任验证机制，使交易及支付各方能够确认其他各方的身份。SET 证书是电子交易各方的身份标识，是他们在 Internet 上进行信息交换及商务活动的身份证明。

代码签名证书，可以帮助用户验证软件发行商的真实身份以及该软件的完整性。随着ActiveX控件、Java Applet和脚本程序的广泛使用，在未经许可的情况下访问或修改本地计算机中文件的可能性在不断地增加。因此，一些软件发行商对在Internet上发布的软件程序进行签名，声明软件是该公司正式发布的产品。用户在下载这些具有代码签名的软件时，可以确信软件是否来自其签发者，并且在签发之后没有被篡改。

WAP 证书是由于 WAP 协议的出现而产生的新证书类型。WAP 把 Internet 的一系列协议规范引入到无线网络中，为无线接入和无线电子商务提供了一个安全便捷的平台。WAP中的身份鉴别通过 WAP 网关证书和 WAP 客户证书实现。WAP 网关证书类似于SSL/TLS服务器证书，用于网关向移动终端进行身份鉴别。同样地，WAP客户证书也类似于SSL/TLS客户证书，区别在于前者定义了两种证书格式：WTLS MIMI-Certificate和X.509。WTLS MIMI-Certificate 在 X.509 的基础上进行了一系列优化处理，更适合于移动终端处理。

7.1.3 CA 的组成

一个典型的CA系统包括安全服务器、注册机构(RA)服务器、认证机构(CA)服务器、LDAP目录服务器和数据库服务器等，如图7.3所示。

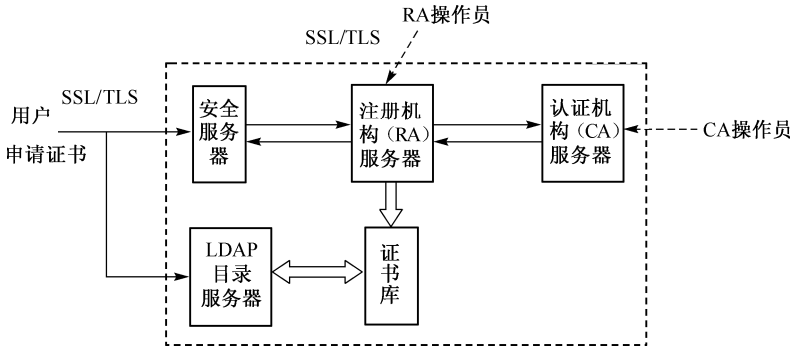


图 7.3 CA 的组成

(1) **安全服务器**：安全服务器面向普通用户，是通过Web服务器建立的站点，可以为用户提供每日 24 小时的安全服务，包括证书申请、浏览、撤销和下载等。安全服务器与用户的通信采取安全信道方式(如SSL/TLS方式)，用户首先得到安全服务器的证书(由 CA 颁发)，然后用户与服务器之间的所有通信均以安全服务器的公钥加密传输。用户把浏览器生成的公钥和个人信息提交给安全服务器。安全服务器将用户的申请信息传送给 RA 服务器。

(2) **RA服务器**：RA 服务器面向 RA 操作员。一方面，RA 操作员利用自己的浏览器与 RA 服务器建立SSL/TLS安全通信，核对用户信息，如果同意用户申请证书的请求，对证书的申请信息签名，并转发给CA服务器。另一方面，RA 服务器向 LDAP 服务器和安全服务器转发 CA服务器颁发的数字证书和证书撤销列表。在某些小型的认证中心中，RA服务器可与 CA 服务器合并。

(3) **CA 服务器**：CA服务器是整个认证中心的核心，负责证书的签发。CA首先产生自身的私钥和公钥(密钥长度至少为1024位)，然后生成数字证书，且将数字证书传输给安全服务器。CA还负责为操作员、安全服务器以及 RA服务器生成数字证书。CA操作员查看用户的详细信息，并且验证RA操作员的签名，如果通过，则同意用户的证书请求，颁发证书。CA服务器是整个结构中最为重要的部分，存有CA的私钥以及发行证书的脚本文件，出于安全的考虑，应将CA服务器与其他服务器隔离，任何通信采用人工干预的方式，确保认证中心的安全。

(4) **LDAP 服务器**：LDAP服务器提供目录浏览服务，负责将注册机构传输过来的用户信息以及数字证书加入到服务器上。这样其他用户通过LDAP服务器就能够得到其他用户的数字证书。

(5) **数据库服务器**：数据库服务器用于认证机构中数据(如密钥和用户信息等)、日志和统计信息的存储和管理。实际的数据库系统应采用多种措施，如磁盘阵列、双机备份和多处理器等方式，以维护数据库系统的安全性、稳定性、可伸缩性和高性能。

7.1.4　密钥和证书的生命周期

图7.4 说明了密钥/证书生命周期的不同阶段，具体描述如下所述。

7.1.4.1　初始化

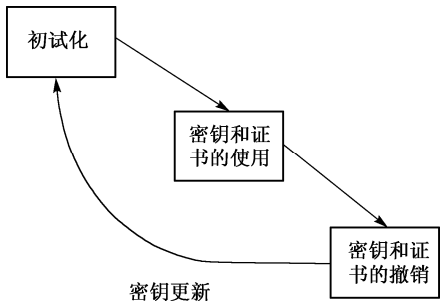


图 7.4　密钥和证书的生命周期

这个阶段由终端实体注册、密钥对产生、证书创建、密钥证书分发组成，必要的话也包含密钥备份。终端实体注册是单个用户或进程的身份被建立的过程，是在线执行的。用户的密钥有两种产生的方式：一种是用户自己生成密钥对，私钥以安全的方式保存在本地，将公钥以安全的方式传送给 CA，该过程必须保证用户公钥的真实性。一种是CA 或其他可信实体集中产生，然后将公钥证书及私钥以安全的方式传送给用户，该过程必须确保密钥对的机密性、完整性和来源真实。通常集中式模型产生加密密钥，签名密钥则在本地产生。无论密钥在哪里产生，证书创建的职责由 CA 完成。证书的分发有几种方式：证书可以电子邮件的方式发送给申请者；也可以由 CA 操作员通过邮件告诉用户申请成功的证书的序列号和下载证书的网址，用户自己下载，服务器要求用户必须用申请证书时的浏览器登录指定网址；也可以从一个证书库(目录服务器)处获得。

7.1.4.2　密钥和证书的使用

密钥和证书分发成功，就进入使用阶段，在这个阶段，用户可以进行证书检索、证书验证、密钥的备份和恢复、密钥更新。证书检索的需求来自加密和签名的需要。证书验证包括证书完整性检查、证书是由一个可信CA 颁发的、证书的有效期是适当的和证书按照预期的策略使用。当用户证书生成的同时，解密密钥就被CA 备份并存储起来。当需要恢复时，用户向 CA 提出申请，CA 就会为用户自动进行恢复。用于签名的私钥为确保其唯一性，是不能够做备份和恢复的。由于证书的有效期是有限的，而且由于某种原因在有效期内也可能作废，比如密钥丢失、用户的个人身份信息发生改变、CA 对用户不再信任或者用户对该CA 不再信任等各种情况。为此，证书和密钥必须要更换。由于用户可能会忘记自己的证书过期时间，PKI 本身自动完成密钥和证书的更新就十分重要。当有效期将满时，CA 会自动启动更新程序，生成一个新证书来代替原来的旧证书，同时将旧证书列入证书撤销列表，并通知用户。

7.1.4.3　密钥和证书的撤销

密钥和证书的管理以撤销阶段来结束。证书撤销可以是因为证书的有效期自然过期，也可以在自然过期前即时取消，即时取消可能基于许多因素，比如，怀疑相应的私钥泄露、作业状态的变化等。即使密钥过期了，可靠和安全地存储它们也是必需的，这被称为**密钥历史**。尽管同样的密钥对可以被同时用于加密和签名，最好还是使用不同的密钥对。签名的密钥和加密的密钥有不同的生命周期，解密的私钥和验证用的公钥在过期之后仍应保留，以能解密用旧密钥加密的数据和验证以前的签名。签名的私钥在过期之后应该被销毁，加密的公钥在过期之后应该被撤销。这样可以保证**前向安全**，也就是一个应用的密钥泄露不会影响到其他应用的安全性。

7.1.5 PKI 信任模型

要实现各 PKI 体系间的互联互通，最为可行的办法是在多个独立运行的 CA 之间实行交叉认证，交叉认证提供了一种解决CA之间互相信任的机制。交叉认证是建立在信任模型之上的。因此选择合适的信任模型和确定最终用户的信任点是 PKI 设计的重要环节。信任模型主要阐述了以下几个问题：一个 PKI 用户能够信任的证书是怎样被确定的? 这种信任是怎样被建立的? 在一定的环境下，这种信任如何被控制?

7.1.5.1 PKI 信任模型相关概念

为了描述信任模型，先了解与信任相关的概念是必要的。与信任相关的概念包括信任、信任域、信任锚、信任关系和信任路径等。

(1) **信任(Trust)**：在 X.509 的 2000 年版中，信任是这样定义的：一般说来，如果一个实体假定另一个实体会严格并准确地按照它所期望的那样行动，那么它就信任该实体。其中的实体是指在网络或者分布式环境中具有独立决策和行动能力的终端、服务器或者智能代理等。信任包含了双方的一种关系以及对该关系的期望，而期望是一个主观的概念，可以用信任水平(信任度)来描述。

(2) **信任域(Trust domain)**：信任域是指公共控制下或者服从于一组公共策略的系统集，简单来说就是信任的范围。识别信任域及其边界对构建PKI很重要。信任域可以按照行业和地理界限来划分。例如，我国构建的CFCA 国家金融认证中心、CTCA 中国电信认证中心、海关 CA 等都是大行业或政府部门建立的行业型 CA。

(3) **信任锚(Trust anchor)**：简单地讲，信任锚就是PKI 体系中的信任起点。在信任模型中，当可以确定一个实体身份或者有一个足够可信的身份签发者证明该实体的身份时，才能做出信任该实体身份的决定，这个可信的身份签发者称为信任锚。

(4) **信任关系**：在公钥基础设施中，当两个认证机构中的一方给对方的公开密钥或双方给互相的公开密钥颁发证书时，两者之间就建立了信任关系。信任关系可以是双向的也可以是单向的，多数情况下采取双向的形式，即某实体相信另外一个实体，那么另一个实体也相信它。

(5) **信任路径(Trust path)**：在一个实体需要确认另一个实体身份时，它先需要确定信任锚，再由信任锚找出一条到达待确认实体的各个证书组成的路径，该路径称为信任路径。信任通过信任路径进行传递。证书用户要找到一条从证书颁发者到信任锚的路径可能需要建立一系列的信任关系。

7.1.5.2 PKI 信任模型评价

从技术的可实现角度和适用的环境和范围出发，评价一个 PKI 信任模型所要涉及的内容很多，其中最重要的有以下几个方面：

(1) **信任域扩展的灵活度**：信任域能否扩展，扩展是否容易，扩展数量有没有限制是多级信任模型应考虑的首要问题。

(2) **信任路径的长度**：信任路径的长度和证书的数量与信任的管理直接相关，信任路径越短，信任路径构建越容易，构建的信任度越高。

(3) **信任路径构建的难易度**：即从现有的信任关系中找出一条从信任的发起方或者发起方的根 CA 到信任的目的方或者目的方的根 CA 的满足要求的信任路径的难易程度，

信任路径构建越容易，需要的运算量以及必需的信息量就越少。反之，证书路径的构建和验证就很困难。

(4) **信任关系的可靠度**：由于信任在传递的过程中会发生衰减，如何保证建立的信任关系的可靠度对一个信任模型来说非常重要。

(5) **证书管理的难易程度**：一个信任模型运行所需要的证书数量越多，管理证书的难度就越大。

(6) **应用范围和具体实施的难度**：信任模型对各种规模应用的支持能力决定了该信任模型适用的范围。

7.1.5.3　PKI 信任模型

目前，已经应用的信任模型包括严格层次(Hierarchical)结构模型、网状信任模型、Web 模型和以用户为中心的信任模型等。

7.1.5.4　严格层次结构模型

严格层次结构模型是最初被提出来的 PKI 信任模型。它可以描绘为一棵倒转的树，如图7.5所示，根在顶上，树枝向下伸展，树叶在下面，根代表根 CA(root CA)，是整个信任环境的信任锚。在根 CA 的下面是零层或多层中间 CA(也称做子CA，Subordinate CA)，树叶通常称做终端实体或者终端用户(End user)。其最简单的形式是由一个 CA 组成单级证书层次结构。多级层次结构包含多个带有明确定义的父子关系的 CA，下级子 CA 由它们的父 CA 颁发证书，而最终用户的证书由子 CA 签发。检查最终用户的证书有效性需要验证从根 CA 到最终用户之间的信任路径上的所有证书。信任关系是单向的，上级 CA 可以而且必须认证下级 CA，而下级 CA 不能认证上级 CA。

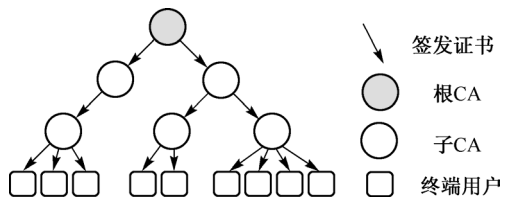


图 7.5　严格层次结构信任模型

严格层次模型的优点是：

(1) 其结构与许多组织或单位的结构相似，容易规划。

(2) 增加新的认证域容易，该信任域可以直接加到根 CA 的下面，也可以加到某个子 CA 下，这两种情况都很方便，容易实现。

(3) 证书路径唯一，容易构建，路径长度相对较短。

(4) 所有的人只要知道根 CA 的证书，就可以验证其他用户的公钥，证书策略简单。

(5) 信任建立在一个严格的层次机制之上，建立的信任关系可信度高。

严格层次模型的缺点是：

(1) 风险集中，根 CA 私钥的泄露或者根 CA 系统的故障将导致整个体系的破坏。

(2) 所有机构共享一个根 CA 是不现实的，许多应用领域并不需要如此严谨的架构。

7.1.5.5　网状信任模型

在网状信任模型中，信任锚的选取不是唯一的，终端实体通常都选择直接给自己发证的 CA 作为信任锚。如果任意两个CA之间都存在着交叉认证，则这种模型就称为严格网状信任模型，如图7.6所示。

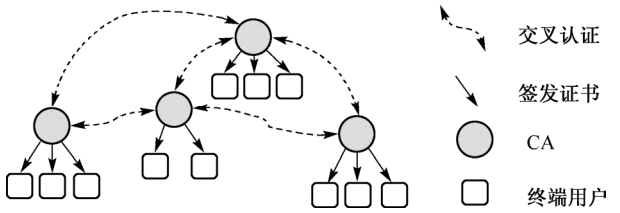


图 7.6 网状信任模型

网状信任模型的优点有：

(1) 网状模型适用于通信机构间对等关系的情况下。

(2) 具有更好的灵活性，因为存在多个信任锚，单个 CA 安全性的削弱不会影响到整个 PKI。

(3) 增加新的认证域更为容易，只需新的根 CA 在网中至少向一个CA 发放过证书，用户不需要改变信任锚。

(4) 由于信任关系可以传递，从而减少颁发的证书个数，使证书管理更加简单容易，能够很好地应用到企业之间。

网状信任模型的缺点是信任路径构建复杂，因为存在多种选择，使得路径发现比较困难，甚至出现死循环。证书路径较长，信任的传递引起信任的衰减，信任关系处理复杂度增加。

7.1.5.6 混合信任模型

混合信任模型把严格层次模型和网状模型结合起来，具有一个或多个单层结构和一个或多个多层结构，如图7.7所示。

混合信任模型结合严格层次模型和网状模型的优点，每个用户都把各自信任域的根CA 作为信任锚，同一信任域内的认证优点完全与严格层次信任模型相同，不同信任域间的用户相互认证时，只需将另一信任域的根 CA 证书作为信任锚即可。尽管可能存在多条证书路径，但信任路径的构造简单，信任路径的长度可以只比严格层次信任模型多一，当非根CA之间相互认证时，证书路径还会更短。

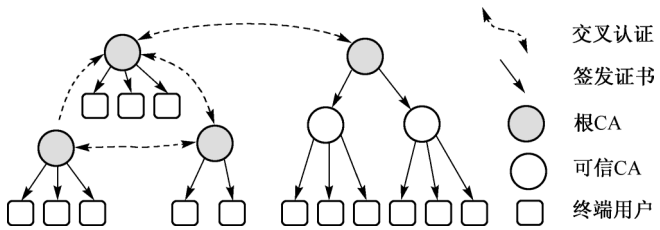


图 7.7 混合信任模型

7.1.5.7 桥 CA 信任模型

桥CA 信任模型也叫中心辐射式信任模型。每一个根CA 都与单一的用做相互连接的处于中心地位的 CA 进行相互交叉认证。这个处于中心地位的 CA 被叫做中心 CA，也叫桥 CA。它被设计成用来克服层次模型和网状模型的缺点和连接不同的 PKI 体系。任何结构类型的 PKI 都可以通过桥CA连接在一起，实现彼此间的信任，每一个单独的信任域都可以通过桥CA 扩展到整个PKI 体系中。这种配置的吸引力在于对 n 个根 CA 来说，完全连接时仅需要 n 个交叉认证。

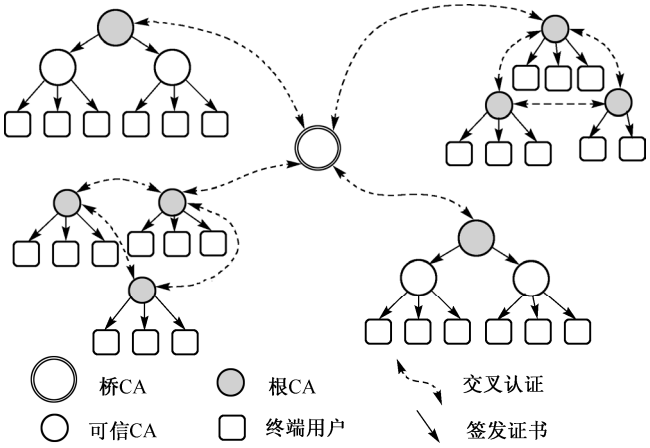


图 7.8　桥 CA 模型

7.1.5.8　Web 模型

Web模型的名字来源于WWW，依赖于流行的Web浏览器。在这种模型中，许多CA的证书被预先安装在浏览器上。一般来说，最终用户和依赖方最初都默认是信任这些CA并把它们作为证书检验的根，同时用户可以根据自己的实际应用增加、删除或更新这些 CA 证书。这种模型更接近严格层次结构信任模型。各个嵌入的根CA并不被浏览器厂商显式认证，而是物理地嵌入软件来发布，作为对CA 名字和它的密钥的安全绑定。但是由于各个根CA 是浏览器厂商内置的，浏览器厂商隐含认证了这些根CA。这样，浏览器厂商就成为了事实上的隐含的根 CA。Web 信任模型可以通过图7.9来表示。

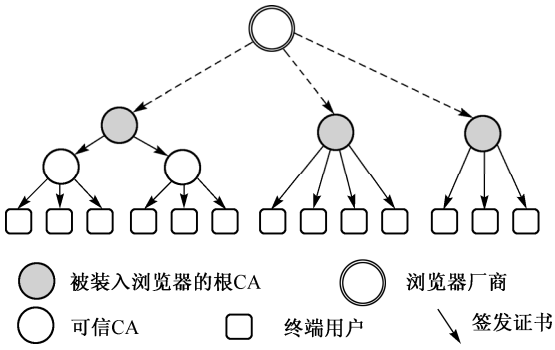


图 7.9　Web 信任模型

Web 信任模型的优点是：方便简单，操作性强，对终端用户的要求较低。用户只需简单的信任嵌入的各个根 CA。但它存在很多不足：

(1) 安全性较差，如果这些根CA中有一个是坏的，即使其他根CA仍然完好，安全性也将被破坏。目前没有实用的机制来撤销嵌入到浏览器中的根密钥，用户也难以查出到底哪一个根 CA 是坏的。

(2) 根 CA 与终端用户信任关系模糊，终端用户与嵌入的根 CA 间交互十分困难。

(3) 扩展性差，根 CA 预先安装，难以扩展。

7.1.6 PKI 发展中的问题

为了保证公钥的真实性，PKI通过引入可信第三方CA，以公钥证书的形式绑定公钥和标识，并形成了解决网络安全问题的一套完整解决方案。但任何一种技术都不是完美的，PKI也是如此。随着 PKI 的使用发展，在实际应用中 PKI 日益凸显出很多问题，列举如下。

证书的使用复杂。首先，需要对用户证书进行管理，包括签发、发布、获取、验证、存储、撤销等，流程较为复杂；其次，在使用证书过程中，需要对证书进行验证，一般要求有一个在线的证书目录能够为用户随时提供证书下载服务，并能够提供证书状态的查询，这在某些情况下可能无法做到；再次，如果用户通信的对象较多，那么用户必须在本地存储和管理这些证书，这也在一定程度上增加了用户使用证书的复杂性。

证书撤销问题难以解决，给证书管理增加了难度。当用户发生私钥泄露或证书损坏等意外情况，必须将此证书作废。据1994年的MITRE报告，证书撤销要占PKI总耗费的 90%。目前较为广泛应用的是 CRL 和在线证书状态查询协议 OCSP（On-line Certificate Status Protocol）两种方法。OCSP 要求发布方是可信第三方，只能在线使用。CRL 是目前应用最广泛的撤销机制，但是存在时间延迟的问题，在两次CRL发布时间之间，有可能发生新的证书撤销事件，因此用户得到的查询结果可能是不正确的。并且CRL 随着证书数量的增加，规模也会变得庞大，下载时占用大量带宽。针对 CRL 的问题，先后提出了增量 CRL（Delta CRL）、分段 CRL（Segment CRL）和重复颁发 CRL（Over Issued CRL）等改进方法，但是由于存在使用复杂等缺陷，应用并不广泛，没有达到很好的效果。在容易发生证书变更的环境中，证书撤销会大大影响 PKI 的工作，给证书管理带来了很大的问题。

PKI 的核心是 CA，大规模密钥管理的问题一般是采用物理上增加 CA 的方法，各个 CA 的用户之间还存在交叉认证和信任管理的问题，比较复杂。

PKI部署一般需要包括CA, KMC, LDAP等部分，部署较为复杂，管理和使用需要一定的知识基础。

7.2 基于身份的密码学

7.2.1 基于身份的密码学原理

PKI使用数字证书将密钥和用户身份进行绑定，由此带来证书的管理、存储和计算上的代价。为了解决这个问题，Shamir 在 1984 年美洲密码学年会上提出了基于身份的密码学（Identity-Based Cryptography，IBC）。在 IBC 中，公钥不再是无意义的二进制数据或大整数，而可以是任意有含义的字符串，用户自己的姓名、邮件地址、电话和身份证号天然地带有用户身份信息，因此可以将这些数据直接作为公钥，无须第三方证书的证明，也就去除了使用数字证书带来的管理、存储和计算上的麻烦。

在Shamir提出的机制中，系统中每个用户都有一个身份，身份可以是用户的姓名、电子邮件地址、电话等信息（如alice，alice@abc.com，0101234567等），用户的公钥可以被任何人根据其身份计算出来，用户向一个称为私钥产生器的可信第三方（Shamir 提到的 trusted Key Generation Center，KGC）认证自己的身份并获得私钥，这样就无须使用服务器保存每个用户的公钥证书，从而减少了证书管理的开销，也不需要使用目录服务管理用户的证书和公钥信息。

IBC 与 PKI 最根本的区别在于密钥产生算法的不同。IBC 使用主密钥、用户公钥、用户私钥三个密钥，公钥是预先选定的，可以是任意比特串，通常使用姓名、电子邮件地址等用户身份标识。密钥产生过程可表示为：

私钥为 F(主密钥，公钥)。

IBC 的出发点在于简化传统 PKI 体制中繁杂的证书管理过程，期望使系统实现变得更为简单。事实上，IBC 有如下三个特点：

- 用户公钥就是其身份(或者由其身份直接导出)。
- 不需要目录服务发布公钥。
- 消息加密和验证签名仅仅需要使用用户的身份。

7.2.2　IBC 的方案

IBE(基于身份的加密机制)、IBS(基于身份的签名机制)和 IBAKA(基于身份带认证的密钥协商机制)是 IBC 研究的主要几个方面，本小节简单介绍 IBE 和 IBS 的实现机制。

7.2.2.1　IBE 机制

IBE 的方案有很多，但概括起来基本上由四个算法组成：系统初始化(Setup)、私钥提取(Private Key Extraction)、加密和解密，如图7.10所示。在该系统中需要包含一个第三方可信机构 TA(Trusted Authority)，该机构主要完成系统初始化、保存系统主密钥、生成用户私钥等功能，该TA也被称为可信的私钥产生中心(Private Key Generator，PKG)。IBE 方案的基本结构和算法步骤描述如下：

(1) **系统初始化**：产生系统参数和主密钥(master key)。系统参数包括明文空间 M、密文空间 C、密码散列函数等，计算系统公钥，将主密钥保密，其他作为系统公开参数。

(2) **私钥提取**：输入用户身份 ID、主密钥、系统公开参数，输出与公钥 ID 对应的私钥 d_{ID}。

(3) **加密**：输入系统参数、公钥 ID 和明文 M，返回对应的密文 C。

(4) **解密**：输入系统参数、私钥 d_{ID} 和密文 C，返回对应的明文。

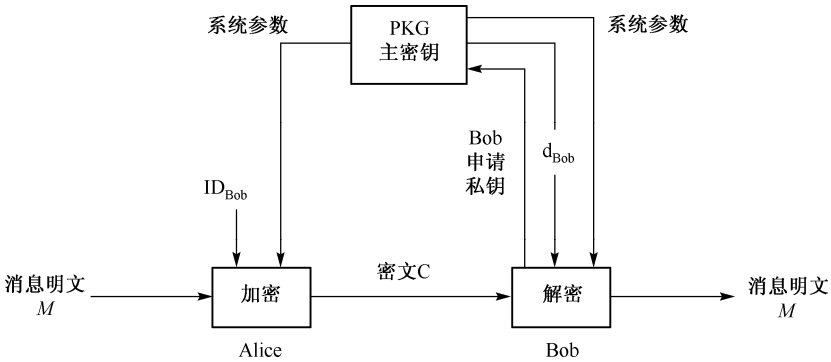


图 7.10　IBE 方案基本过程

同基于证书的加密过程相比，由于身份标识即是公钥，不再需要从证书目录处下载证书，也不需要验证证书的有效性。加密和解密算法分别由加密、解密消息的用户运行。在实际应用中，公钥 ID 就是用户公开的身份标识符，例如一个 E-mail 地址user@abc.com，将这个身份标识符送到 PKG 后，PKG 产生对应的私钥并分发给用户。需要注意的是，这里：

- 用户需要向 PKG 证明他是该 ID 字符串的合法拥有者。
- PKG 生成的私钥必须通过安全信道传送给用户，以保证私钥的机密性。

7.2.2.2 IBS 机制

与 IBE 类似，基于身份的签名机制也需要包含一个可信机构 PKG(Private Key Generator)，整个机制也包括四个算法：系统初始化(Setup)、私钥提取(Private Key Extraction)、签名(Sign)和验证(Verify)。

(1) **系统初始化**：产生系统参数和主密钥(master key)。系统参数包括明文空间 M、密文空间 C、密码散列函数等，计算系统公钥，将主密钥保密，其他作为系统公开参数。

(2) **私钥提取**：输入系统参数、主密钥和用做公钥的任意字符串ID，输出与公钥ID对应的私钥 d_{ID}。

(3) **签名**：输入待签的消息 M，系统公开参数和用户私钥，生成签名。

(4) **验证**：输入签名数据、系统公开参数和用户身份，输出验证结果。

签名过程和验证过程如图7.11所示。

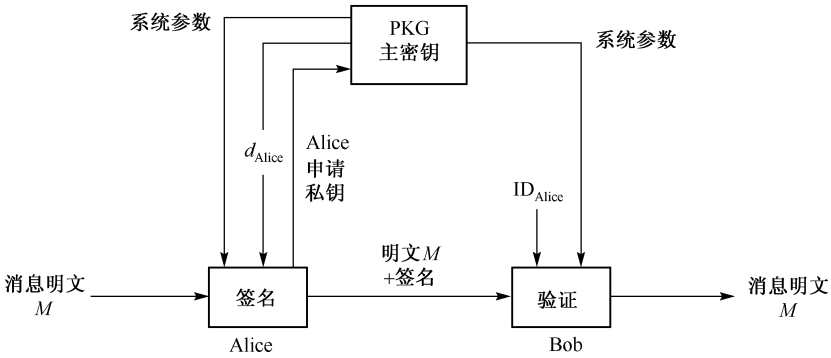


图 7.11 签名过程和验证过程

7.2.3 IBC 的实际问题

仅仅有算法是不够的，IBC 算法应用于实际时，还面临一些具体问题，最主要的两个是密钥托管和密钥撤销问题。

7.2.3.1 密钥托管

密钥托管(Key escrow)是指用户的私钥由第三方生成或托管在第三方。IBC使用了可信第三方作为用户私钥产生中心(PKG)，PKG 能够解开它所管辖范围内所有人的加密数据，所以密钥托管是 IBC 的一个固有问题。在很多有中央管理机构的应用中，IBE 具有密钥托管能力是一个好的性质；而在数字签名中通常要避免密钥托管，因为被托管的密钥无法产生具有法律效力的数字签名。针对密钥托管问题，有很多文献提出了各种解决机制。

7.2.3.2 密钥撤销

密钥撤销也是 IBC 中的一个重要问题，对于传统 PKI 来讲，密钥和用户身份是独立的，只能通过证书来绑定，通过设置证书有效期、发布 CRL 以及查询 OCSP 等办法可以对用户的

密钥进行撤销。但是在 IBC 中，用户的密钥是和用户身份标识直接关联的，是天然就绑定在一起的，因此，一旦用户密钥发生(或者怀疑发生)泄露，用户密钥的撤销问题就是一个大问题，因此，即使用户密钥能撤销，用户的身份怎么能撤销？又如何撤销？在 IBC 中，解决密钥撤销问题一般有几种办法：

- **将时间段附在身份标识后**：将该密钥的有效时间段附在用户身份标识后，作为新的身份标识，犹如给用户设定一个定期的身份标识。比如为某用户 Bob 设置 2007 年的身份标识：“Bob2007”，这样，Bob 只能在 2007 年使用这个身份标识和对应的密钥。这类办法可以在一定程度提供定期的密钥撤销能力，但是还不好提供更细粒度的密钥撤销能力，更不能提供即时撤销(Instant Revoke)能力。
- **使用 LDAP, OCSP**：参考传统的 PKI 的办法，也使用 LDAP 和 OCSP，为用户提供已撤销身份标识列表和查询身份标识有效状态。
- **使用SEM**：即将用户私钥分成两部分，一部分在用户手中，另外一部分在一个半可信机构(Semi-trusted Security Mediator, SEM)中，在用户需要使用私钥时(解密、签名)需要 SEM 的协助才能完成。这样，在需要撤销用户时，只需要控制 SEM 即可。

7.3　ECC 组合公钥体制

基于标识的组合公钥体制(IBCPK-Identity Based Combined Public Key, CPK)是具有我国自主知识产权的公开密钥管理体制。为了解决大规模网络环境下密钥的生产与分发问题，我国著名密码专家南湘浩教授1999年提出了CPK算法，2003年公布，2006年获得中国专利。

如何解决大规模密钥的存储，正向的思维是对大量的密钥进行压缩，CPK 算法采用逆向思维，由较小的因子矩阵生成大量的密钥。

公钥组合算法现在有两种：一种是利用 RSA 算法体制实现的多重公钥(Lapped Public Key, LPK)，另一种是利用离散对数(DLP)和椭圆曲线(ECC)实现的组合公钥。下面对ECC CPK 进行介绍。

2003 年南湘浩教授在其所编著的《网络安全技术概论》中以论著形式首次提出基于椭圆曲线的组合公钥算法，此为 ECC CPK 的 1.0 版本。在一般的公钥体制中，各用户的公钥是直接公布的，有多少用户，就公布多少个公钥，而在 ECC CPK 体制中，各用户的公钥不直接公布，而只公布公开种子矩阵，公私钥通过组合公钥算法从种子矩阵中生成，采用映射算法将标识映射到种子矩阵。2006 年南湘浩教授在其所著的《CPK 标识认证》一书中，对组合公钥算法进行了改进。2008 年提出了 ECC CPK 的 2.0 版本。2.0 版本是标识密钥和随机密钥复合的公钥体制，因此也称双因子复合型组合公钥体制(Two Factor Combined Public Key, TF-CPK)。标识密钥(Identity-Key)由实体的标识通过组合矩阵生成。组合矩阵(Combining matrix)由密钥管理中心(KMC)定义，并以文件形式公布，提供标识和密钥的绑定关系。随机密钥(Random-Key)是由系统定义的随机序列，与标识密钥复合产生一阶复合密钥，随机序列消除私钥变量之间的线性关系。标识密钥和随机密钥均由密钥管理中心负责定义，称为集中式 CPK。2009 年 6 月南湘浩教授又给出了组合公钥体制标准的 3.0 版本，将 2.0 版本中定义的随机密钥分为伴随密钥和分割密钥，并给出了数字签名和加密的方法。下面给出 CPK3.0 版本的算法描述。

7.3.1 CPK 相关概念

组合公钥密码体制(CPK-Cryptosystem)是在椭圆曲线密码(ECC)上构建的基于标识的密码体制。CPK 的组合密钥可由标识密钥和伴随随机密钥复合，或由标识密钥和分割密钥复合而成，组合密钥(Combined-Key)用(csk，CPK)标记。

在组合公钥(CPK)体制中，密钥由标识密钥、伴随密钥、分割密钥组成。

标识密钥(Identity-Key)由实体的标识通过组合矩阵生成。**组合矩阵**由密钥管理中心(KMC)定义，分为**私钥矩阵**和**公钥矩阵**。私钥矩阵用于私钥的生产，需要保密；而公钥矩阵用于公钥的生成，需要公布。组合矩阵为标识和密钥的绑定关系提供证明。标识密钥对用(isk, IPK)标记。其中 isk 是标识私钥，IPK 是标识公钥。

伴随密钥(Accompany Key)是由 KMC 为各个实体定义的随机密钥，伴随密钥对用(ask, APK)标记。**分割密钥**(Separting Key)由KMC 统一定义，每个实体有一对分割密钥，用(ssk, SPK)标记。

7.3.2 ECC 复合定理

设椭圆曲线 E：$y^2=(x^3+ax+b) \bmod p$，其参数为 $T=\{a, b, G, n, p\}$，其中 p 是奇素数，变元和系数 a，b 均在有限域 Z_P 中取值；G 是椭圆曲线 $E_p(a, b)$ 上的基点，用 $G=(x_G, y_G)$ 标记；n 是素数，是基点 G 的阶。假定私钥 K_R 为任一小于 n 的整数 r，那么公钥 K_U 为椭圆曲线 E 上的一个点 rG。组合公钥体制采用有限域 Z_P 上的椭圆曲线密码。

ECC 复合定理如下：在椭圆曲线密码ECC 中，任意多对公、私钥，其私钥之和与公钥之和构成新的公、私钥对。

如果，私钥之和为：$(K_{R1}+K_{R1}+\cdots+K_{Rm})=(r_1+r_2+\cdots+r_m) \bmod n=r$

则对应公钥之和为：$(K_{U1}+K_{U2}+\cdots+K_{U\,m})=R$(点加)

那么，r 和 R 刚好形成新的公、私钥对。

因为，$R=(K_{U1}+K_{U2}+\cdots+K_{U\,m})=r_1G+r_2G+\cdots+r_mG=(r_1+r_2+\cdots+r_m)\,G$

$=r\,G$

7.3.3 标识密钥

7.3.3.1 组合矩阵

组合矩阵分为私钥矩阵和公钥矩阵。矩阵大小均为 32×32。私钥矩阵由互不相同的小于 n 的随机数构成，矩阵中的元素标记 r_{ij}，私钥矩阵记为 skm。

$$\mathrm{skm}=\begin{bmatrix} r_{1,1} & r_{1,2} & r_{1,2} & \cdots & r_{1,32} \\ r_{2,1} & r_{2,2} & r_{2,3} & \cdots & r_{2,32} \\ r_{3,1} & r_{3,2} & r_{3,3} & \cdots & r_{3,32} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ r_{32,1} & r_{32,1} & r_{32,3} & \cdots & r_{32,32} \end{bmatrix}$$

公钥矩阵由私钥矩阵派生，即 $r_{ij}G=(x_{ij}, y_{ij})=R_{ij}$，公钥矩阵记为 PKM。

$$\mathrm{PKM}=\mathrm{skm}\cdot\boldsymbol{G}=\begin{bmatrix} r_{1,1}G & r_{1,2}G & r_{1,3}G & \cdots & r_{1,32}G \\ r_{2,1}G & r_{2,2}G & r_{2,3}G & \cdots & r_{2,32}G \\ r_{3,1}G & r_{3,2}G & r_{3,3}G & \cdots & r_{3,32}G \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ r_{32,1}G & r_{32,2}G & r_{32,3}G & \cdots & r_{32,32}G \end{bmatrix}=\begin{bmatrix} R_{1,1} & R_{1,2} & R_{1,3} & \cdots & R_{1,32} \\ R_{2,1} & R_{2,2} & R_{2,3} & \cdots & R_{3,32} \\ R_{3,1} & R_{3,2} & R_{3,3} & \cdots & R_{3,32} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ R_{32,1} & R_{32,2} & R_{32,3} & \cdots & R_{32,32} \end{bmatrix}$$

7.3.3.2　标识到矩阵坐标的映射

标识私钥 isk 和标识公钥 IPK 的计算在 KMC 进行，根据实体标识在组合矩阵每一列选取一个元素计算得到。标识到所选组合矩阵元素坐标的映射是通过对标识的散列变换实现的。将散列输出调整成长度为 190 比特的映射序列 YS，以每 5 比特构成 $w_0, w_1, \cdots, w_{37}$ 的字符串，用于决定被选元素的列坐标与行坐标。

$$\mathrm{YS}=\mathrm{HASH}(\mathrm{ID})=w_0, w_1, w_2, \cdots, w_{32};\ w_{33}\cdots, w_{37}$$

其中，w_0 的值 u 指示列的起始坐标，以后的列坐标是在前列坐标加 1 实现，w_1～w_{32} 依次指示行坐标。w_{33}～w_{37} 指示标识对应的分割密钥在分割密钥表中的位置。

7.3.3.3　标识密钥的组合计算

标识私钥 isk 的计算以有限域 Z_n 上的加法实现，设 $r[w_i, (u+i)\bmod 32]$表示私钥矩阵 skm 中行坐标为 w_i、列坐标为$(u+i)\bmod 32$ 的元素。实体 A 的私钥为：

$$\mathrm{isk_A}=\sum_{i=1}^{32} r\ [w_i, (u+i)\bmod 32]\bmod n$$

标识公钥 IPK 的计算以椭圆曲线 $E_p(a,b)$上的点的加法实现，设 $R[w_i, (u+i)\bmod 32]$表示公钥矩阵 PKM 中行坐标为 w_i、列坐标为$(u+i)\bmod 32$ 的元素，按照复合定理，对应公钥为：

$$\mathrm{IPK_A}=\sum_{i=1}^{32} R\ [w_i, (u+i)\bmod 32]\qquad (\text{点加})$$

设矩阵大小为 $m*n$，那么存储量为 $m*n$，可组合出的密钥数量为 m^n。当 $m=32$，$n=32$ 时，$m*n=1024$，而 $m^n=2^{160}\approx 10^{48}$，从而实现了由较小的组合矩阵生成大量的密钥。组合公钥的思想可以推广到私钥之和(或积)与对应的公钥之和刚好构成新的公私钥对的密码体制中，如 RSA、ElGamal 等。

7.3.4　密钥的复合

7.3.4.1　标识密钥与伴随密钥的复合

实体 A 随机定义一对伴随密钥 $\mathrm{ask_A}$, $\mathrm{APK_A}$，由 KMC 生成组合私钥：

$$\mathrm{csk1_A}=(\mathrm{isk_A}+\mathrm{ask_A})\bmod n$$

KMC 删除伴随私钥 $\mathrm{ask_A}$，并对伴随公钥签名：

$$\mathrm{SIG}_{\mathrm{isk_A}}(\mathrm{APK_A})=\mathrm{sign1}$$

KMC 将实体 A 的组合私钥 $\mathrm{csk1_A}$，伴随公钥 $\mathrm{APK_A}$ 以及 sign1 记入 CPK-card。

组合公钥将由依赖方计算：

$\mathrm{CPK1_A}=\mathrm{IPK_A}+\mathrm{APK_A}$(由签名方 A 提供 $\mathrm{APK_A}$ 和 sign1)

7.3.4.2 标识密钥与系统密钥的复合

KMC 统一生成分割密钥(ssk, SPK)表，分割公钥 SPK 表以文件形式公布。

实体 A 的组合私钥 $csk2_A$ 由 KMC 计算：$csk2_A=(isk_A+ssk_A) \bmod n$，将组合私钥 $csk2_A$ 记入 CPK-card，并删除分割私钥 ssk_A。

组合公钥将由依赖方计算：$CPK2_A = IPK_A + SPK_A$。

其中，ssk_A 和 SPK_A 在分割密钥表中的位置由序列 YS 中的 w_{33}～w_{37} 指示。

7.3.5 CPK 数字签名

7.3.5.1 利用伴随密钥签名

签名：设在 Alice 的 ID 卡中具有组合私钥 $csk1_A$，伴随公钥 APK_A，

那么 Alice 的签名：$SIG_{csk1_A}(TAG) = sign$；

将(sign, APK_A, sign1)提供给验证方。

其中 TAG 是标识域、时间域或字符串。

验证：设验证方收到签名码(sign, APK_A, sign1)；

验证方计算 Alice 的标识公钥：$HASH(ID) \rightarrow IPK_A$；

验证方验证 Alice 的伴随公钥：$SIG^{-1}_{IPK_A}(APK_A) = sign1'$；

验证方计算 Alice 的组合公钥：$CPK1_A = IPK_A + APK_A$；

验证方验证 Alice 的签名：$SIG^{-1}_{CPK1_A}(TAG) = sign'$。

7.3.5.2 利用分割密钥签名

签名：设在 Alice 的 ID 卡中具有组合私钥 $csk2_A$；

那么 Alice 的签名：$SIG_{csk2_A}(TAG) = sign$；

将签名码 sign 发送给验证方。

验证：验证方计算 Alice 的公钥：$CPK2_A = IPK_A + SPK_A$；

其中，IPK_A，SPK_A 查表获得。

验证 Alice 签名：$SIG^{-1}_{CPK2_A}(TAG) = sign'$。

7.3.6 CPK 密钥交换

7.3.6.1 利用分割密钥加密

下面给出了在CPK 体制下，用对方公钥来加密传送对称加密的会话密钥并进行加解密的过程，这里利用对方的分割公钥和标识公钥复合得到组合公钥。

加密：Alice 通过 Bob 的标识求出 w_{33}～w_{37}，查找 Bob 的分割公钥 SPK_B；

Alice 根据 Bob 的标识和公钥矩阵计算 Bob 的标识公钥 IPK_B；

Alice 计算 Bob 的公钥：$CPK2_B = IPK_B + SPK_B$；

Alice 选择随机数 r，计算：$r \cdot CPK2_B = \beta$ 和 $r \cdot G = key$；

Alice 加密：$E_{key}(data) = code$；

Alice 将 code 和 β 发送给 Bob。

解密：Bob 用自己的私钥计算出 key：

$$csk2_B^{-1}\beta = csk2_B^{-1}(r\ CPK2_B) = csk2_B^{-1}(r\ csk2_B\ G) = r\ G = key$$

Bob 解密：$D_{key}(code) = data$。

7.3.6.2　利用伴随密钥加密

下面给出了使用伴随密钥时双方如何交换公钥的方案，在有些在线通信中，如 C/S 通信，需要证明双方的真实性，可以通过认证获得对方的公钥。

请求：客户端发送认证信息

$$SIG_{csk1_c}(ID_c) = sign_c, APK_c, sign1_c$$

响应：服务器端发送认证信息

$$SIG_{csk1_s}(ID_s) = sign_s, APK_s, sign1_s$$

这样通信双方都具有了对方的公钥。

7.3.7　安全性分析

CPK 以有限域上椭圆曲线群的数学原理为理论基础，攻击者要从其公开密钥逆求其私有密钥，其难度相当于解决椭圆曲线离散对数的难题。

CPK1.0 是由标识密钥单独构成的 CPK 体制，即 IPK = CPK，isk = csk。由于标识私钥是由用户标识和私钥矩阵通过线性关系产生的，不同的用户可合谋起来，利用其私钥和标识，建立一组求解私钥矩阵的线性方程组，称这种攻击为共谋攻击，参与建立方程组的不同用户数为共谋数。如果共谋数不超过矩阵的变量数，方程无解，称为理论安全，如果实际环境中矩阵变量数大于可能产生的共谋数，方程仍然无解，称为实际安全。在本体制中，组合矩阵的大小不恒定，因此需要在共谋数和矩阵变量数之间进行权衡。此体制的优点是一个体制可以同时做数字签名和密钥交换，而且数字签名长度短，缺点是组合矩阵占用很大存储空间。组合矩阵变量数(允许共谋的数)和存储空间的关系如表7.1所示(密钥长度 160 bit)。

表 7.1　组合矩阵变量数(允许共谋的数)和存储空间的关系

变量数(共谋数)	公钥的简化存储(只存 *x*)	公钥全变量存储(存 *x* 和 *y*)
128 000　(10 万)	2.56 MB	5.124 MB
512 000　(50 万)	10.24 MB	20.48 MB
1 024 000 (100 万)	20.48 MB	40.96 MB

在 CPK3.0 中，分割公钥 SPK 以文件形式公布，存放于本机媒介中，文件大小随用户数量的增加而增大。由于分割密钥为互相独立的随机数序列，而且每一个实体一个密钥，因此，各实体的私钥之间不存在线性关系，共谋威胁不会扩大到共谋参与者以外。表7.2给出了分割密钥的用户数量和存储空间的关系。

表 7.2　分割密钥的用户数量和存储空间的关系

用 户 数 量	*x* 的公布量	(*x*, *y*) 的公布量
128 000	2.56 MB	5.12 MB
512 000	10.24 MB	20.48 MB
1 024 000	20.48 MB	40.96 MB
2 048 000	40.96 MB	81.92 MB

此外，为了防止共谋的攻击者在获取足够数量私有密钥的基础上实施对种子密钥矩阵的逆求，CPK采用了芯片反解剖、软件防护及私钥加密存放等技术，使存放于芯片的私有密钥得到强有力的保护。

7.3.8 ECC CPK 小结

CPK 的突出优势在于：它通过有限种子密钥组合的方法，产生数量庞大的公、私钥对；它基于实体的标识生成公钥解决了实体与公开密钥的“捆绑”，无须可信第三方的认证；它采用私有密钥的集中生产与分发方式，便于管理与建立网络上的秩序。CPK 算法密钥管理简捷、高效、经济，适用于各种大型专用网，如 VPN、国防网、行政网、金融支付网的集中式密钥管理。

PKI、IBC 和 CPK 三者有相通之处，但技术路线不同，PKI 是通过第三方间接证明的体系，而 IBE、CPK 是将标识作为公钥的直接证明的体系。IBE 的数字签名和密钥交换采用与现有公钥算法不同的算法，而 CPK 则采用的是相同的算法。

思考和练习题

(1) 什么是 PKI？一个典型的 PKI 包含哪些组件？
(2) 为什么要设置独立的 RA 和 CA？是否可以把两者合二为一？有什么不同？
(3) CA 的基本功能有哪些？
(4) 数字证书中存放了哪些信息？有什么作用？
(5) 简述数字证书的签发过程。
(6) 有哪些类型的数字证书？
(7) 解释 PKI 信任模型中的下列概念：信任、信任域、信任锚、信任路径。
(8) 如何评价 PKI 的信任模型？
(9) PKI 的信任模型主要有哪几种？各有什么特点？
(10) PKI 发展中遇到了哪些问题？
(11) IBC 体制的原理是什么，你能设想出基于 IBC 体制的哪些应用？
(12) IBC 实际应用中遇到的主要问题有哪些，各有什么解决思路？
(13) ECC CPK 的公私钥对是如何生成的？
(14) ECC 组合公钥算法的数学原理是什么？
(15) 什么是对 CPK 的共谋攻击？
(16) CPK 的理论安全性和实际安全性是什么？

实践/实验题

基于开放源码构建一个小型集中式 CA 系统。

第8章 鉴 别 协 议

在第 1 章中，提到国际标准ISO 7498—2《信息安全体系结构》定义了5大类安全服务，分别是数据机密性、数据完整性、不可否认性、鉴别和访问控制。前面的几章，已经对数据机密性、数据完整性、不可否认性服务进行了介绍，本章将介绍第 4 类安全服务——鉴别。

8.1 鉴别的相关概念

在本章正式开始之前，先来解释一些容易混用的基本概念。

- **身份和标识**：身份和标识都是由一个词Identity而来。在物理世界中，身份主要指一个人的社会地位和法律地位，行为主体往往是在法律意义上具有行为能力的人。因此，身份认证、身份鉴别都是指人而言，所用的技术也是照片和指纹等生物技术。因为人与人的区别，当然主要体现在生物特征上。但是在日常生活中，如果都用生物特征来区别，会有很多不便，于是便产生了主体的逻辑替代物：姓名、称谓、个人标识(身份证)号。在网络世界中，身份的含义发生了很大变化，主体不完全是人，进程也可以是主体，以标识的名义进行动作，成为行为主体，通常通过标识的识别达到身份识别的目的。
- **认证、鉴别和识别**：认证对应的英文是Certification，它代表一种资质的证明，比如公安机关对公民的身份进行审查，如果审查通过就给公民颁发相应的资质证明，身份证或者护照。学校对学生的身份进行审查，如果审查通过就发给相应的证件，学生证或者图书证。鉴别对应的英文是Authentication，它代表一种真伪的证明，是一种一对一的匹配，结果只有两个，真或者假。学校门卫对学生的证件进行检查，判断学生身份的真伪，如果审查通过，就让其进校门或者进图书馆。识别对应的英文是 Identification，含义是对不同的东西进行区分，是一种一对多的匹配。对应的身份认证、身份鉴别和身份识别三个术语的含义有联系也有区别。身份鉴别通过回答“他是他自称的这个人吗”来辨别身份的真伪。身份识别通过回答“他是谁”来确定用户身份。由于一些使用习惯，三个词之间也形成了一些混用。对此，本书加以说明，读者可以根据这三个词的上下文使用环境分析其具体含义。

8.2 密码协议

图1.1给出了信息安全的技术体系，前面的几章介绍了信息安全的核心技术：密码理论和技术，内容涉及密码算法、密码算法提供的服务和密钥的管理。在此之上的技术层次是安全协议或者说是密码协议。

首先给出协议的概念，协议指的是双方或多方通过一系列规定的步骤来完成某项任务。从这个定义中，可以看出协议的含义包含三方面内容：第一，协议自始至终是有序的过程，

每一步骤必须依次执行；第二，协议至少需要两个参与者；第三，通过执行协议必须完成某项任务。此外，协议还有一些默认的其他特点：协议的每一方必须事先知道此协议及要执行的步骤；协议涉及的每一方必须同意遵守协议；协议必须是非模糊的，也就是说协议的每一步必须含义明确，不能有二义性；协议必须是完整的；每一步的操作要么是由一方或多方进行计算，要么是在各方之间进行消息传递。

协议有三种类型：仲裁协议、裁决协议和自动执行协议。

- 在仲裁协议中，仲裁者是在完成协议的过程中，值得信任的公正的第三方，“公正”意味着仲裁者在协议中没有既得利益，与参与协议的任何人也没有特别的利害关系。“值得信任”表示协议中的所有人都接受这一事实，即仲裁者所说的都是真实的，所做的都是正确的。现实生活中，律师、银行和公正人担任的都是仲裁者的角色。
- 裁决协议包括两个低级的子协议：一个是非仲裁子协议，执行协议的各方每次想要完成的，另一个是裁决子协议，仅在例外的情况下，即有争议的时候才执行，这种特殊的仲裁者叫裁决人。法官是现实生活中的裁决者。
- 自动执行协议指的是协议本身就保证了公平性，不需要仲裁者来完成协议，也不需要裁决者来解决争端。如果协议的一方试图欺骗另一方，那么另一方能够立刻检测到欺骗，并停止执行该协议。

使用密码的具有安全性功能的协议称为**安全协议**或**密码协议**。根据协议的功能，密码协议主要有以下三类：

- **密钥建立协议**(Key establishment protocol)：在通信双方或多方之间建立共享秘密。
- **鉴别协议**(Authentication protocol)：向一个实体提供另一个通信实体身份的某种程度的确认。
- **鉴别的密钥建立协议**(Authenticated key establishment protocol)：与另一个身份已被或可被证实的实体之间建立共享秘密。

对密码协议的攻击，从攻击目标上分为攻击协议使用的密码算法和密码技术与攻击协议本身。从攻击方式上分为被动攻击与主动攻击。被动攻击指与协议无关的人能窃听协议的一部分或全部。主动攻击指改变协议以便对自己有利，如假冒、删除、代替和重放等。

8.3 实体鉴别概述

对于特定的信息系统资源，为了保证机密性和完整性，应该只有经过授权的合法用户才能访问，这里问题的关键就在于如何正确地鉴别用户的真实身份。作为保障信息系统安全的第一道重要防线，实体鉴别技术用于鉴别用户身份，防止未经授权的非法访问，是网上用户进行通信、交易等活动的基础。实体鉴别也被称为**身份鉴别**。

8.3.1 实体鉴别的基本概念

简单地说，**实体鉴别**(Entity Authentication)就是确认实体是它所声明的。实体鉴别的需求来自某一成员(**声称者**)提交一个主体的身份并声称它是那个主体，目的是使别的成员(**验证者**)获得对声称者所声称的事实的信任。

8.3.2　实体鉴别和消息鉴别的区别和联系

在国际标准ISO 7498—2《信息安全体系结构》中定义的 5 大安全服务之一 —— 鉴别，指的是通信中的对等实体和数据来源的鉴别。数据原发鉴别对数据单元的来源提供确证，对数据单元的重复或篡改不提供保护。

实体鉴别一般都是实时的，在实体鉴别中，身份由参与某次通信连接或会话的参与者提交。这种服务在连接建立或在数据传送阶段的某些时刻提供，使用这种服务可以确信，仅仅在使用时间内，一个实体此时没有试图冒充别的实体，或没有试图将先前的连接做非授权重演。消息鉴别一般不提供时间性。验证者可以在任何时间对消息的完整性和来源进行验证。

实体鉴别用于一个特定的通信过程，即在此过程中需要提交实体的身份，它只是简单地鉴别实体本身的身份，不会和实体想要进行何种活动相联系。消息鉴别除了鉴定某个指定的数据是否来源于某个特定的实体，还要对数据单元的完整性提供保护。它不是孤立地鉴别一个实体，也不是为了允许实体执行下一步的操作而鉴别其身份，而是为了确定被鉴别的实体与一些特定数据项有着静态的不可分割的联系。

数字签名主要用于证实消息的真实来源，但数字签名服务中也包含对消息内容的保证。

在身份鉴别中，消息的语义是基本固定的，但一般不是“终生”的，比如用于身份验证的口令字会定期更改，签字却是长期有效的。

8.3.3　实体鉴别实现安全目标的方式

实体鉴别可以对抗假冒攻击的危险，这是最重要的安全服务之一，它提供了关于某个实体身份的保证，所有其他的安全服务都依赖于该服务。

作为访问控制服务的一种必要支持，访问控制服务的执行依赖于确知的身份，而访问控制服务直接对机密性、完整性、可用性及合法使用目标提供支持。

当它与数据完整性机制结合起来使用时，可以作为提供数据起源认证的一种可能方法。

作为对责任原则的一种直接支持，例如，在审计追踪过程中做记录时，提供与某一活动相联系的确知身份。

8.3.4　实体鉴别的分类

实体鉴别可以分为本地和远程两类。本地多用户鉴别指实体在本地环境的初始化鉴别，也就是说，作为实体个人和设备物理接触，不和网络中的其他设备通信，需要用户进行明确的操作。远程用户鉴别指连接远程设备、实体和环境的实体鉴别。通常采用将本地鉴别结果传送到远程的方式，而不直接与远程设备或实体进行鉴别，前一种方式更安全和易用。

实体鉴别可以是单向的也可以是双向的。单向鉴别是指通信双方中只有一方向另一方进行鉴别。双向鉴别是指通信双方相互进行鉴别。

8.3.5　实体鉴别系统的组成

实体鉴别系统由以下几部分组成。

一方是出示证件的人，称为示证者 P(Prover)，又称声称者(Claimant)。声称者提交一个主体的身份并声称他是那个主体。

另一方为验证者 V(Verifier)，检验声称者提出的身份的正

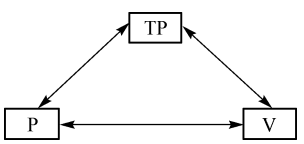


图 8.1　实体鉴别系统的组成

确性和合法性，决定是否满足要求。

第三方是可信赖者 TP(Trusted third party)，参与调解纠纷。

系统的攻击者，可以窃听或伪装声称者骗取验证者的信任。

8.3.6 实现身份鉴别系统的途径和要求

实现身份鉴别的途径可以是以下三种途径之一或它们的组合：

(1) **所知**：英文是 What do you know，也就是只有该用户唯一知道的一些知识，如密码、口令等。

(2) **所有**：英文是 What do you have，也就是只有该用户唯一拥有的一些物品，如身份证、护照、信用卡、钥匙等。

(3) **个人特征**：英文是 What are you made of，指的是人体自身的独特特征(生物特征)以及个人动作方面的一些特征。如指纹、笔迹、声纹、手形、血型、视网膜、虹膜、DNA 等。

要设计一个身份鉴别系统，从安全性和计算效率上有以下要求：

- 验证者正确识别合法申请者的概率极大化，是对一个身份鉴别系统的首要要求。这是考虑到一个身份鉴别系统在实际应用中可能会存在这样一种误判——错误拒绝，指系统将合法用户当成非法用户(假冒者)而拒绝。我们希望设计的鉴别系统把合法用户识别为合法用户的概率极大化，也就是正确接受率极大化。而正确接受率 =1-错误拒绝率，错误拒绝率，又称**拒识率**(False Rejection Rate, FRR)，表示授权人不被正确承认的程度。也就是要求拒识率极小化。
- 不具有可传递性(Transferability)。验证者不可能重用示证者提供给他的信息来伪装示证者，而成功地骗取其他人的验证，从而得到信任。
- 攻击者伪装成申请者欺骗验证者成功的概率要小到可以忽略的程度。身份鉴别系统误判的另一种情况是错误接受，指系统将非法用户(假冒者)当成合法用户而接受。通常用错误接受率，又称**误识率**(False Acceptance Rate, FAR)，表示未授权者被错误承认的程度。而正确拒绝率 =1-错误接受率，也就是希望误识率极小，系统正确拒绝未授权用户的概率极大化。
- 从协议的效率出发，还希望系统计算高效、通信有效。因为协议每一步的操作要么是由一方或多方进行计算，要么是在各方之间进行消息传递。
- 鉴别协议如果涉及密码算法，要求秘密参数能安全存储。

此外，交互识别、第三方的实时参与、第三方的可信赖性和可证明的安全性也是一些身份鉴别系统的设计要求，但不是必需的。

要设计一个实用的身份鉴别系统要从安全水平、系统通过率、用户可接受性和成本等方面综合考虑。

8.4 鉴别机制

在电子世界中，鉴别机制的基本类别如图8.2所示。基于所知的机制目前有三种类型：静态口令(Fixed password)机制、一次性口令(One-time Password，OTP)机制和基于密码学的挑战-应答协议(Cryptographic challenge-response protocol)的机制。基于所有的机制在电子世界中

主要包含基于地址和基于设备的鉴别机制。此外，基于个人特征的鉴别机制也是目前的一个研究热点。

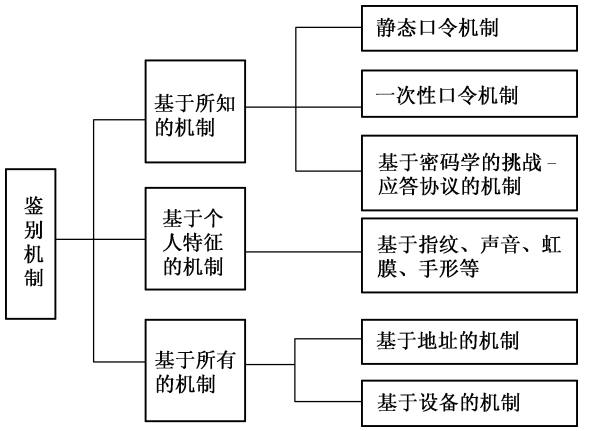


图 8.2　鉴别机制的基本类别

8.4.1　口令机制

本节将介绍口令机制，并讨论它的脆弱性，以及对抗这些脆弱性的措施。口令或通行字机制是最广泛研究和使用的身份鉴别方法。口令通常是长度为 5～8 个字符的字符串，选择口令的原则是易记、难猜、抗分析能力强。常规的口令方案涉及不随时间变化的口令，提供所谓的**弱鉴别**(Weak authentication)。口令系统有许多脆弱点，最严重的是口令外部泄露和口令猜测，此外还有线路窃听、危及验证者和重放等威胁。

1．对付口令外部泄露的措施

口令的外部泄露指由于用户或管理员的疏忽或其他原因导致未授权者得到口令。例如，用户为了防止忘掉口令而将口令写到一个不安全的地方，比如把银行卡的口令写在背面，把计算机的口令写在纸条上，把纸条贴在计算机上，或者在一个未受保护的文件中存储口令。下列措施可以有助于防止口令的外部泄露：

(a) 对用户或者系统管理者进行教育、培训，增强他们的安全意识。

(b) 建立严格的组织管理办法和执行手续。

(c) 确保口令定期改变。

(d) 保证每个口令只与一个人有关。

(e) 确保输入的口令不显示在终端上。

(f) 使用易记的口令，不要写在纸上。

2．对付口令猜测的措施

口令猜测也是一个严重的脆弱性，下列措施有助于防止口令猜测攻击：

(a) 对用户或者系统管理者进行教育、培训，增强他们的安全意识。

(b) 严格限制非法登录的次数；比如一些网站的邮件系统规定如果10分钟内非法登录超过 12 次，账户就会被锁定 24 小时。

(c) 口令验证中插入实时延迟，比如一些系统除了要求你输入口令之外，还要求输入一个系统回显的字符串，这可以限制强力攻击的测试频率。

(d) 限制口令的最小长度，至少 6～8 字节以上，这可以增大强力攻击的测试数量。

(e) 防止使用与用户特征相关口令，因为攻击者很容易想到从用户相关的一些信息来猜测，比如生日。

(f) 确保口令定期改变。

(g) 及时更改预设口令。

(h) 使用随机数发生器产生的口令比使用用户自己选择的口令更难猜，但会有记忆的问题。

避免口令外部泄露的措施可能会与避免口令猜测的措施有一些冲突，需要设计者和管理者折中考虑。

3．对付线路窃听的措施

如果一个攻击者对通信线路进行窃听，就可能获得合法登录者输入的口令，冒充合法用户进行假冒攻击。我们可以使用单向函数对付这种窃听口令攻击，如图8.3所示。设 f 为单向函数，用户的标识是id，他的正确口令是 p，在验证系统中保存的是用户的标识 id 和口令 p 单向变换后的值$q=f(p)$，用户在鉴别终端输入用户输入标识id和口令 p'，终端计算$q'=f(p')$，并将 q' 和 id 发送给验证者，验证者比较标识为 id 的用户的 q 和q'，如果两者一致，则认为输入口令正确，确认用户的身份。即使攻击者窃听通信线路得到q'，因为函数f的单向性，他也难以猜测出相应的 p。一种拨号用户鉴别协议CHAP在传输用户口令信息时就使用了单向变换的方式，协议的具体内容本章后面将会介绍。

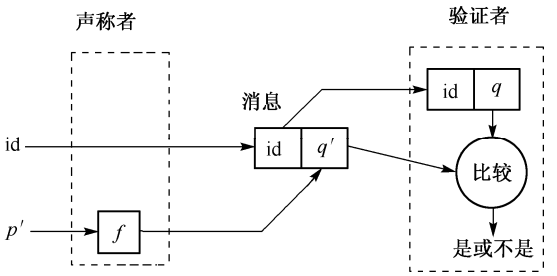


图 8.3　一种对付线路窃听的机制

4．对付危及验证者的措施

如果攻击者获得了对验证者上存储的口令文件的访问，知道了标识为 id 的用户对应的口令，他就可以在线路上产生一个鉴别请求，假冒标识为 id 的用户。为了对付这种脆弱性，口令一般不明文存储，而是存储口令单向变换后的值。为了保护口令，使用单向函数显然比使用可逆的加密变换优越。

但攻击者可对口令文件进行字典攻击，它构造一张口令散列值 q 与口令 p 对应的表，表中的p 尽最大可能包含所期望的值，就能以很高的概率获得用户的某些口令。随机串(salt)是使这种攻击变得困难的一种办法。也就是说在口令后使用随机数，令 $q=f(p||\text{salt})$。改进后的方案如图8.4所示。即使不同用户选择了同样的口令，由于每个用户的 salt 不同，所存储的口令单向变换后的值也不同，这个方法只能保护在多台计算机上使用相同口令或在同一计算机上使用同一口令的不同用户。因为口令密文和salt的明文及用户 ID 一起保存在口令文件中，对于一个用户，加salt并不能增加对其口令穷尽搜索的次数，但是却增加了字典攻击的复杂性，因为针对每一个用户口令需要存储 2^t 种不同的单向变换值，t 为 salt 的位数。所以，使用 salt 的唯一目的是防止预计算，防止字典攻击。即使 salt 不是保密的，仍然可以达到这个目的。

UNIX 系统中的口令存储就采用了这种加 Salt 的办法，它使用函数crypt( )来保证系统密

码的完整性，这一函数完成被称为单向加密的功能，它基于DES算法，同时通过加入 12 位(现在已经有多种长度的 salt值，可供选择)的 salt 改变了标准 DES 的扩展置换，使可能的口令数量增加了 $2^{12}=4096$ 倍。每个用户选择长度小于 8 位的可打印字符作为口令，选择每个 ASCII 码的 7 位组成 56 位的密钥，如果不够就补 0，对 64 位的全 0 分组重复加密 25 次，最后，将 64 位的输出和 12 位的 Salt 转换为 11 个字符序列。使用 salt可以达到三个目的：防止口令文件中出现相同口令；无须用户额外记住两个字符，就能增加口令长度；阻止了用硬件实现 DES，对口令进行强力破解。

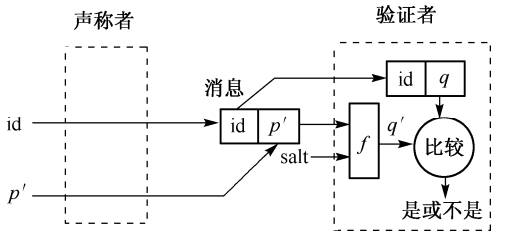


图 8.4　对付危及验证者的机制

8.4.2　一次性口令机制

上面的机制都不能抵抗对通信线路的主动攻击——重放攻击。一次性口令机制确保在每次鉴别中所使用的口令不同，以对付重放攻击。Alfred J. Menezes 在《应用密码学手册》中把不使用密码算法的一次性口令机制称为**近似的强鉴别**(Towards strong authentication)，他也把静态口令机制和不使用密码算法的一次性口令机制称为非密码的机制(Non-cryptographic)，并给出如下确定口令的方式：

- **两端秘密共享一串随机口令**：在该串的某一位置保持同步，此方案有两端需要维护秘密随机串表的缺点。
- **顺序更新一次性口令**：使用旧的口令把新的口令加密传输过去。

现在的一次性口令机制通常都需要用户拥有某种计算设备。首先，在用户和远程服务器之间建立一个秘密——相当于传统口令技术当中的“口令”，在此被称为“秘密通行短语”(Secure Pass Phrase，SPP)。实现一次性口令的机制主要有两种方式：第一种是采用询问-应答方式，用户登录时，系统随机提示一条信息，用户根据这一信息连同其通行短语言共同产生一个口令字，用户输入这个口令字，完成一次登录过程，如贝尔实验室于1994 年公开发布的S/KEY 协议，后面详细介绍；第二种是采用时间(或事件)同步机制，即根据这个同步时钟(或事件)信息连同其通行短语共同产生一个口令字。多家安全公司推出了基于密钥和时间双因素的身份鉴别系统，用户登录口令随时间变化，口令一次性使用，无法预测。图8.5给出了一种动态口令卡的原理。动态口令卡是发给每个用户的动态口令发生器，通过某种非线性迭代算法，以时间为参数，每隔不到一分钟产生一个一次性使用的“动态口令”，用户把“动态口令”传输到服务器，服务器对此“动态口令”进行逆向变换，如果得到的时间参数与服务器时间在误差范围内的话，就确认此用户。事件同步以询问-应答方式为基础，将单向的前后相关序列作为系统的质询信息，如 Secure Computing 公司的 SafeWord 产品。

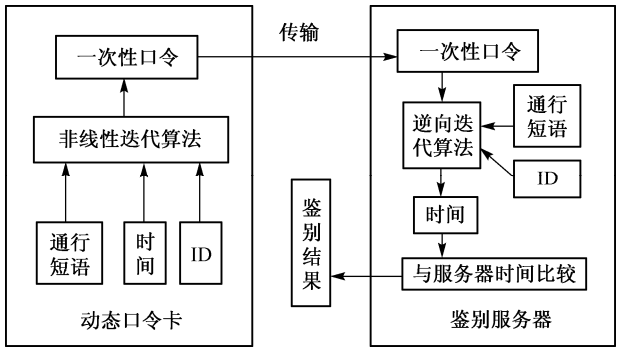


图 8.5 一种一次性口令机制

8.4.3 基于密码算法的鉴别机制

强鉴别(Strong authentication)指基于密码学的询问-应答(Challenge-response)协议实现的身份鉴别，询问-应答协议的思想是一个实体向另一个实体证明他知道有关的秘密知识，但不向验证者提供秘密本身。这通过对一个时变的询问提供应答来实现，应答通常依赖于实体的秘密和询问。询问通常是一个实体随机和秘密地选择的一个数值。这类机制根据采用的密码算法的不同可以分为如下三类：基于对称密码算法的鉴别机制、基于公开密码算法的机制和基于密码校验函数的机制，图 8.6 为简化的基于密码算法的鉴别机制示意图。

基于对称密码算法的鉴别依靠一定协议下的数据加密处理。通信双方共享一个密钥，通常存储在硬件中，该密钥在询问-应答协议中处理或加密信息交换。基于对称密码算法的鉴别机制可以没有可信第三方，也可以有可信第三方——通常称为 KDC(Key Distribution Center)或 AS(Authentication Server)，由这个第三方来实现通信双方的身份鉴别和密钥分发，Kebreros 是此类鉴别协议中比较完善、较具优势的协议，得到了广泛的应用。

在采用公开密码算法的机制中，声称者要通过证明他知道某秘密签名密钥来证实身份。通过使用他的秘密签名密钥签署某一消息来完成。消息可包含一个非重复值以抵抗重放攻击。要求验证者有声称者的有效公钥，声称者有仅由自己知道和使用的秘密签名私钥。获得有效公钥的途径可以采用第 7 章介绍的数字证书或 CPK 等方式。

在采用密码校验函数的机制中，待鉴别的实体通过表明它拥有某个秘密鉴别密钥来证实其身份。可由该实体以其秘密密钥和特定数据作输入，使用密码校验函数获得密码校验值来达到。密码校验值可由拥有该实体的秘密鉴别密钥的任何实体来校验。声称者和验证者共享秘密鉴别密钥，应仅为这两个实体以及他们的信任方所知。

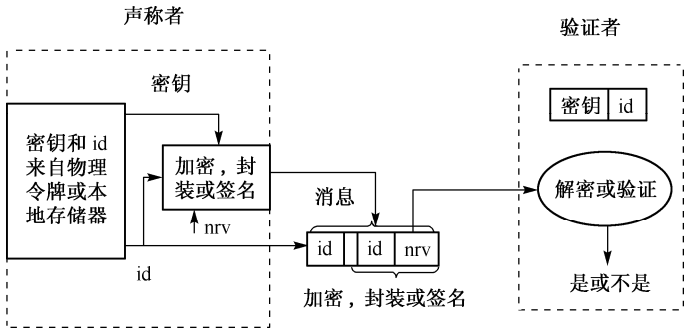


图 8.6 简化的基于密码算法的鉴别机制示意图

8.4.4　零知识证明协议

零知识证明(Zero knowledge proof)技术可使信息的拥有者无须泄露任何信息就能够向验证者或者任何第三方证明它拥有该信息。即当示证者 P 掌握某些秘密信息，P 想方设法让验证者 V 相信他确实掌握那些信息，但又不想让V也知道那些信息。零知识证明分为两大类：最小泄露证明(Minimum disclosure proof)和零知识证明。零知识证明需满足下列条件：

(1)P 几乎不可能欺骗 V，如果 P 知道证明，他可以使 V 以极大的概率相信他知道证明；如果 P 不知道证明，则他使得 V 相信他知道证明的概率几乎为零。

(2)V 几乎不可能知道证明的知识，特别是他不可能向别人重复证明过程。

(3)V 无法从 P 那里得到任何有关证明的知识。

满足前两个条件就是最小泄露证明。

零知识证明最通俗的例子就是图 8.7 所示的山洞问题。图中的山洞里 C, D 两点之间有一扇上锁的门，P 知道打开门的咒语，按照下面的协议 P 就可以向 V 证明他知道咒语但不需要告诉 V 咒语的内容：

(1)V 站在 A 点。

(2)P 进入山洞，走到 C 点或 D 点。

(3)当 P 消失后，V 进入到 B 点。

(4)V 指定 P 从左边或右边出来。

(5)P 按照要求出洞(如果需要通过门，则使用咒语)。

(6)P 和 V 重复步骤(1)至(5)n 次。

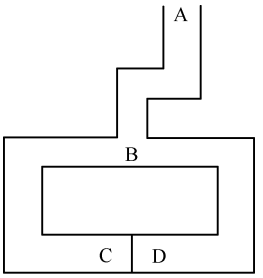


图 8.7　山洞问题

如果 P 知道咒语，他一定可以按照 V 的要求正确走出山洞 n 次。如果 P 不知道咒语，他需要猜测，每次猜对的概率是0.5，猜对 n 次的概率是0.5^n，当 n 足够大时，这个概率接近于零。

在网络身份鉴别中，已经提出了一些零知识证明协议，如FFS方案、FS方案和GQ方案。一般地，验证者颁布大量的询问给声称者，声称者对每个询问计算一个回答，而在计算中使用了秘密信息。大部分技术要求传输的数据量较大，并且要求一个更复杂的协议。

8.4.5　基于地址的机制

基于地址的机制假定声称者的可鉴别性是以呼叫的源地址为基础的。在大多数的数据网络中，呼叫地址的辨别都是可行的。在不能可靠地辨别地址时，可以用一个呼叫-回应设备来获得呼叫的源地址。一个验证者对每一个主体都保持一份合法呼叫地址的文件。这种机制最大的困难是在一个临时的环境里维持一个连续的主机和网络地址的联系。地址的转换频繁、呼叫-转发或重定向引起了一些主要问题。基于地址的机制自身不能被作为鉴别机制，但可作为其他机制的有用补充。

8.4.6　基于设备的鉴别

设备的物理特性用于支持鉴别“某人拥有某东西”，但通常要与一个口令或PIN结合使用。这种设备应具有存储功能，通常有键盘、显示器等界面部件，更复杂的能支持一次性口令，甚至可嵌入处理器和自己的网络通信设备(如智能卡)。

这种设备通常还利用其他密码鉴别方法。目前支持鉴别的设备有 PDA, WLAN 网卡，DRM 设备和 RFID 等。

8.4.7 基于个人特征的机制

生物特征识别(Biometric identification)技术是一门以生物测定技术为基础，信息技术为手段，将21世纪生物和信息两大热门技术融为一体的科学。它是利用图像处理、模式识别和计算机视觉的方法对人类本身所具有的独一无二的生理特征和行为特征(统称为生物特征)进行可靠地、有效地分析和描述，通过判断这些描述的一致性从而实现自动身份确认的一类技术。生物特征具有“人各有异、终生不变、随身携带”等特点，因而同传统的基于所知和所有的机制相比，生物特征识别技术具有稳定、便捷、不易伪造等优点，近年来引起了国际学术界、企业界以及政府有关部门的广泛关注。生物特征识别技术主要有：指纹识别、声音识别、虹膜识别、手形识别等。这些技术的使用对网络安全协议不会有重要的影响。

8.5 鉴别与密钥交换协议设计中的问题

本节介绍在设计鉴别协议时特别需要注意的问题，并给出抵抗这些攻击的具体设计策略。如果用于连接完整性服务的密钥被在线建立，那么事实证明将鉴别和密钥交换功能组合在一个协议中是重要的，这就是鉴别和密钥交换协议，是最常用的协议，该协议使得通信各方互相鉴别各自的身份，然后交换会话密钥。鉴别和密钥交换协议的核心问题有两个：保密性和时效性。为了防止伪装和暴露会话密钥，基本鉴别与会话密码信息必须以保密形式通信。这就要求预先存在保密或公开密钥供实现加密使用。第二个问题也很重要，因为涉及防止消息重放攻击。针对不同验证者的重放，可以在发送消息的同时发送验证者的标识符。针对同一验证者的重放对策是使用非重复值。非重复值有以下几类：

(1) **序列号**：对付重放攻击的一种方法是在鉴别交换协议中使用一个序数给每一个消息报文编号。仅当收到的消息序数顺序合法时才接受。如 TCP/IP 协议中的每一个 TCP 数据报都有一个序列号。但这种方法的困难是要求双方必须保持上次消息的序号。

(2) **时间戳**：用户 A 接受一个新消息仅当该消息包含一个时间戳，该时间戳在 A 看来，足够接近A所知道的当前时间；这种方法要求不同参与者之间的时钟需要同步。在网络环境中，特别是在分布式网络环境中，时钟同步并不容易做到。一旦时钟同步失败，要么协议不能正常服务，影响可用性，造成拒绝服务；要么放大时钟窗口，造成攻击的机会。时间窗大小的选择应根据消息的时效性来确定。

(3) **随机值**：该随机值不可预测、不重复。当用户A 期望从用户B获得一个消息，首先发给B一个随机值(challenge)，B收到这个值之后，对它做某种变换，并送回去，A收到B的应答(response)，验证 B 是否真的收到了 A 发给他的随机值，该随机值是否是重放。在有的协议中，这个challenge 也称为nonce，可能明文传输，也可能密文传输。变换的方式可以是用密钥加密，说明B知道这个密钥，也可以是简单运算，比如增一，说明B知道这个随机值。询问/应答方法不适应非连接性的应用，因为它要求在传输开始之前握手的额外开销，这就抵消了无连接通信的主要特点。

在理论上，相互鉴别可通过组合两个单向鉴别交换协议来实现。然而，这种组合需要被仔细地考察，因为有可能这样的组合易受窃听和重放攻击。另外，设计协议消息数比两倍相

应单向交换协议消息数少得多的相互鉴别交换协议是可能的。因此，由于安全性和性能的原因，相互鉴别交换协议必须为此目的而特别地进行设计。

8.6　鉴别与交换协议实例

8.6.1　CHAP 协议

挑战握手鉴别协议(Challenge-Handshake Authentication Protocol，CHAP)是一种拨号用户鉴别协议，完成对 PPP(Point to Point Protocol)链接的身份鉴别。CHAP 主要适用于 NAS(Network Access Server)对来自于 PSTN 或 ISDN 的电路交换连接、拨入连接或专有连接的身份鉴别。

挑战握手鉴别协议(CHAP)采用提问/应答方式进行鉴别，通过验证方与被验证方之间的三次握手周期性地鉴别访问者的身份，在初始链路建立时完成，可以在链路建立之后的任何时候重复进行。验证方周期地验证登录和访问请求，一旦检测到，就生成和发送一个随机数(Challenge)给被验证者，被验证者据此生成一个单向加密的摘要值作为应答(Response)传给验证方，验证方根据收到的应答来判断用户身份合法性。CHAP 成功鉴别的前提是验证双方共享同样的秘密值和单向加密(Hash)算法，并且共享的密钥秘密值不是通过该链路发送的。实际验证中，验证方在发出随机数的同时，会和被验证方一起以共享的秘密值和随机数为输入计算消息摘要，并把二者计算的结果汇总、比较，若相等，则认可该次访问，反之予以拒绝，如图8.8所示。

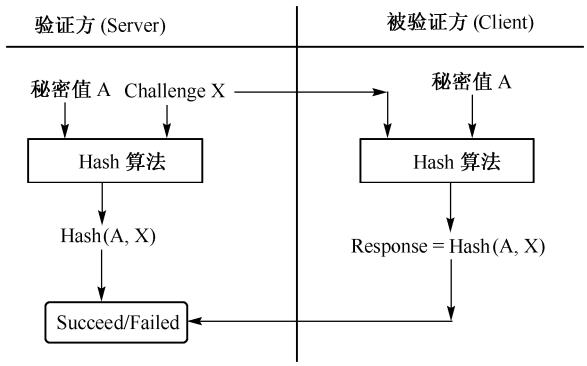


图 8.8　CHAP 鉴别原理

CHAP 协议的优点是：

(1) 协议简单、易于实现。

(2) CHAP 协议虽然为单向鉴别协议，如果需要双向鉴别，可以将CHAP 按两个相反的方向执行两次，两次执行可以使用同一个共享密钥，但是更好的做法是使用两个不同的共享密钥。

(3) 通过不断地改变鉴别标识符和提问消息的值来防止重放(Playback)攻击。

(4) 利用周期性的提问防止通信双方在长期会话过程中被攻击，限制了对单个攻击的暴露时间，验证方控制挑战的频度。

CHAP 协议的缺点是：

(1) CHAP 要求秘密值以明文形式存在。

(2) CHAP 鉴别基于共享秘密值，给密钥管理带来巨大不便，不适合于大规模用户鉴别。

8.6.2 S/KEY 协议

S/KEY 协议出现的背景是为了避免以下的攻击：搭线监听网络上传输的口令及用户 ID，并在适当的时候利用这些信息，登录计算机阅读或修改用户信息。

S/KEY 协议由贝尔实验室于 1994 年公开发布并在 RF1760 中定义，是基于 MD4, MD5 的一次性口令生成机制，主要用于防止重放攻击。

S/KEY 协议的安全性依赖于一个单向函数 f。其原理如下：为建立这样的系统，A 输入一随机数 R，计算机计算 $f(R)$，$f(f(R))$，$f(f(f(R)))$，…，共计算 100 次，计算得到的数为 $x_1, x_2, x_3, \cdots x_{100}$，A 打印出这样的表，随身携带，计算机将 x_{101} 存在 A 的名字旁边。A 第一次登录，输入 x_{100}，计算机计算 $f(x_{100})$，并将它与 x_{101} 比较。比较通过，计算机存储 x_{100}，删除 x_{101}。以后按 i 递减的顺序依次输入 x_i，计算机计算 $f(x_i)$，并将它与 x_{i+1} 比较。

实际的 S/KEY 协议由三个组成部分(如图8.9所示)：

(1) **客户端程序**：为端用户提供登录程序，并在得到服务器质询值时，获取用户私钥，并调用口令计算器形成本次鉴别口令，然后发送给服务器程序。

(2) **口令计算器**：负责产生本次口令。

(3) **服务器程序**：验证用户口令。

在整个过程中，用户的私钥不会暴露在网络上。

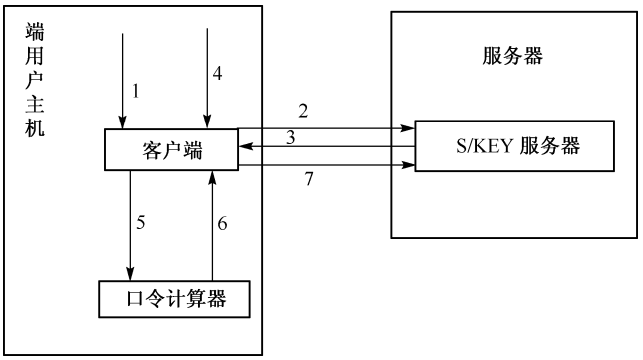


图 8.9 S/KEY 协议三个组成部分的关系

图 8.9 中 S/KEY 协议的鉴别步骤如下：

(1) 用户登录。

(2) 客户端向服务器发出登录请求。

(3) 服务器向客户端发出 S/KEY 质询。

(4) 客户端要求用户输入私钥。

(5) 私钥与服务器发出的质询被送入计算器。

(6) 计算器产生本次口令。

(7) 客户端将口令传给服务器，服务器对之进行验证。

S/KEY 协议的口令生成过程如下：S/KEY 协议要求用户的私钥不少于 8 字节(64 bit)。它把种子和密钥黏结，根据登录服务器的提示的次数，多次使用安全散列函数，进行 MD4/MD5

散列，再将输出分为两部分异或，产生 64 位的输出，并按照可读的形式显示 64 位的一次性口令，根据包含 2^{11} 个单词的字典，对应 6 个英文单词。

8.6.3　Kerberos

Kerberos 协议是 20 世纪 80 年代 MIT 的 Athena 项目的一部分，它的名字取自希腊神话，Kerberos 是希腊神话故事中一种三个头的狗，还有一个蛇形尾巴，是地狱之门的守卫。这里 Kerberos 意指有三个组成部分的网络之门的保卫者，“三头”包括：鉴别(Authentication)、清算(Accounting)和审计(Audit)，但目前只实现了鉴别功能。前三个版本仅用于内部，第4版得到了广泛的应用，第 5 版于1989 年开始设计，1993 年确定，并成为 Internet 标准草案(RFC 1510)，第 5 版主要是修补了第 4 版中的一些安全漏洞。

8.6.3.1　Kerberos 的设计动机

在一个开放的分布式网络环境中，当用户通过工作站访问服务器上提供的服务时，服务器应能够限制非授权用户的访问并能够鉴别对服务的请求，同时也需要服务器向用户证明自己的身份。第一个有关 Kerberos 的公开发表的报告列举了如下的 Kerberos 需求：

- **安全**：网络窃听者不能获得必要信息以假冒其他用户；并且，Kerberos 应足够强壮以至于潜在的攻击者无法找到它的脆弱连接。
- **可靠**：Kerberos 应高度可靠，并且应借助于一个分布式服务器体系结构，使得一个系统能够备份另一个系统。
- **透明**：理想情况下，除了输入口令以外用户应感觉不到认证的发生。
- **可伸缩**：系统应能够支持大数量的客户和服务器。

Kerberos 协议的做法不是为每一个服务器构造一个身份鉴别协议，而是提供一个中心鉴别服务器 AS(Authentication Server)，提供用户到服务器和服务器到用户的鉴别服务。Kerberos 采用对称加密算法，其中第 4 版要求使用 DES 算法。

8.6.3.2　Kerberos 模型

在 Kerberos 协议中，参与方包括客户 C、鉴别服务器 AS(Authentication Sever)，票据许可服务器 TGS(Ticket-granting Server)和应用服务器 V，交互的步骤有 6 步，如图8.10所示。

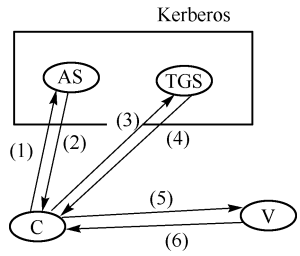


图 8.10　Kerberos 协议的步骤

在 Kerberos 协议中，每一个被鉴别的个体客户 C 称为安全个体(Principal)，有一个名字(Name)和口令(Password)。

鉴别服务器AS和票据许可服务器TGS是KDC(Key Distribution Center)，提供票据(Ticket)和临时的会话密钥。鉴别服务器 AS 只有一个，票据许可服务器 TGS 却可以有多个。

Kerberos鉴别过程使用两类凭证：票据和鉴别码(Authenticator)。票据包括客户的标识、会话密钥、时间戳，以及其他一些信息，客户C可以用它来向服务器 V 证明自己的身份，票据中的大多数信息都被加密，密钥为服务器的密钥。鉴别码(Authenticator)是一个记录，包含一些最近产生的信息，产生这些信息需要用到客户和服务器之间共享的会话密钥，用于防止重放攻击。

8.6.3.3 Kerberos 版本 4

Kerberos 版本 4 的鉴别过程包含 6 个步骤：

(1) C→AS：$ID_c \| ID_{tgs} \| TS_1$

(2) AS→C：$E_{Kc}[K_{c,tgs} \| ID_{tgs} \| TS_2 \| Lifetime_2 \| Ticket_{tgs}]$

$Ticket_{tgs} = E_{Ktgs}[K_{c,tgs} \| ID_c \| AD_c \| ID_{tgs} \| TS_2 \| Lifetime_2]$

(3) C→TGS：$ID_v \| Ticket_{tgs} \| Authenticator_c$

(4) TGS→C：$E_{Kc,tgs}[K_{c,v} \| ID_v \| TS_4 \| Ticket\ Ticket_v]$

$Ticket_{tgs} = E_{Ktgs}[K_{c,tgs} \| ID_c \| AD_c \| ID_{tgs} \| TS_2 \| Lifetime_2]$

$Ticket_v = E_{Kv}[K_{c,v} \| ID_c \| AD_c \| ID_v \| TS_4 \| Lifetime_4]$

$Authenticator_c = E_{Kc,tgs}[ID_c \| AD_c \| TS_3]$

(5) C→V：$Ticket_v \| Authenticator_c$

(6) V→C：$E_{Kc,v}[TS_5+1]$　　(用于相互鉴别)

$Ticket_v = E_{Kv}[K_{c,v} \| ID_c \| AD_c \| ID_v \| TS_4 \| Lifetime_4]$

$Authenticator_c = E_{Kc,v}[ID_c \| AD_c \| TS_5]$

上述过程可以分为三个阶段，第一阶段是鉴别服务交换，目的是为了获得票据许可票据(ticket-granting ticket)，包含步骤(1)和步骤(2)。在消息(1)中客户 C 向鉴别服务器 AS 请求票据许可票据，ID_c 告诉 AS 客户端的用户标识；ID_{tgs} 告诉 AS 用户请求访问 TGS；TS_1 让 AS 验证客户端的时钟是否与 AS 的时钟同步。消息(2)返回票据许可票据 $Ticket_{tgs}$，是客户端用来访问 TGS 的票据。该票据基于用户口令 K_c 加密，使得 AS 和客户端可以验证口令，并保护消息(2)。$K_{c,tgs}$ 是一个会话密钥，由 AS 产生，可用于在 AS 与客户端之间信息的安全交换，而不必共用一个永久的密钥。ID_{tgs} 标识这个票据是为 TGS 制作的。TS_2 告诉客户端该票据签发的时间。$Lifetime_2$ 告诉客户端该票据的有效期。

第二阶段是服务许可票据的交换，目的是获得访问应用服务器的许可票据，包含步骤(3)和步骤(4)。在消息(3)中，客户端向TGS请求服务许可票据，ID_v 告诉 TGS 用户要访问服务器 V；$Ticket_{tgs}$ 向 TGS 证实该用户已被AS鉴别；$Authenticator_c$ 由客户端生成，包含用户 C 的标识 ID_c、网络地址 AD_c 和时间戳TS_3，仅能使用一次，用于说明用户已经得到会话密钥$K_{c,tgs}$，也即证明使用 $K_{c,tgs}$ 的用户必为 C。在消息(4)中，TGS 返回服务许可票据 $Ticket_V$ 和客户与服务器的会话密钥 $K_{c,v}$，该消息用仅由 C 和 TGS 共享的密钥 $K_{c,tgs}$ 加密，ID_v 确认该票据是为服务器 V 签发的；TS_4 告诉客户端该票据签发的时间；$Ticket_v$ 是客户端用以访问服务器 V 的票据；$Ticket_{tgs}$ 可重用，从而用户不必重新输入口令；$K_{c,tgs}$ 可用于解密 $authenticator_c$，ID_c 指明该票据的正确主人。

第三阶段是客户与应用服务器的鉴别交换，目的是请求服务，包含步骤(5)和可选的步骤(6)。在消息(5)中，客户端向服务器申请服务，客户端用$Ticket_v$ 向服务器V证明该用户通过了AS的鉴别，$Authenticator_c$ 由客户端生成，包含用户C的标识 ID_c、网络地址 AD_c 和时间戳 TS_5，仅能使用一次，用于说明用户已经得到会话密钥 $K_{c,v}$，也即证明使用 $K_{c,v}$ 的用户必为 C。消息(6)用于服务器对用户表明此消息只能由服务器 V 生成，实现双向鉴别。

8.6.3.4 Kerberos 版本 5

Kerberos 版本 5 的消息交换过程如下：

(1) C→AS：$Options \| ID_c \| Realm_c \| ID_{tgs} \| Times \| Nonce_1$

(2) AS→C：

$Realm_c||ID_c||Ticket_{tgs}||E_{Kc}[K_{c,tgs}||Times||Nonce_1||Realm_{tgs}||ID_{tgs}]$

$Ticket_{tgs} = E_{Ktgs}[Flags||K_{c,tgs}||Realm_c||ID_c||AD_c||Times]$

(3) C→TGS：$Options||ID_v||Times||Nonce_2||Ticket_{tgs}||Authenticator_c$

(4) TGS→C：$Realm_c||ID_c||Ticket_v||E_{Kc,tgs}[K_{c,v}||Times||Nonce_2||Realm_v||ID_v||]$

$Ticket_{tgs} = E_{Ktgs}[Flags||K_{c,tgs}||Realm_c||ID_c||AD_c||Times]$

$Ticket_v = E_{Kv}\ [Flags||K_{c,v}||Realm_c\ ||ID_c||AD_c||Times]$

$Authenticator_c = E_{Kc,tgs}[ID_c||Realm_c||TS_1]$

(5) C→V：$Options||Ticket_v||Authenticator_c$

(6) V→C：$E_{Kc,v}[TS_2||Subkey||Seq\#]$

$Ticket_v = E_{Kv}[Flags||K_{c,v}||Realm_c||ID_c||AD_c||Times]$

$Authenticator_c = E_{Kc,v}[ID_c||Realm_c||TS_2||Subkey||Seq\#]$

其中 Realm 标识用户所属的域，Options 用于请求在返回的票据中设置指定的位，Times 用于设置时间，包括起始时间、过期时间和请求更新的过期时间，nonce 是一个临时交互号，用于防止重放。

在上述版本 5 中步骤(2)和步骤(4)对提供给客户端的票据不再进行二次加密。对客户服务器鉴别交换也进行了一些改进，消息(5)中，客户端可以请求选择双向鉴别选项，增加了 Subkey 和 Sequence 两个新域。Subkey 可用于保护一个特定的应用会话，Sequence 是一个可选域，说明消息的序号。

Kerberos 版本 5 改进了 Kerberos 版本 4 的其他一些局限和不足，能更好地防止重放攻击，提高了口令猜测的复杂度，简化了域间认证，使用混合加密算法，避免了对Internet协议的依赖性，增加了票据的生命周期，允许向前鉴别，即服务器为了完成客户请求的服务而请求其他服务器协作的能力。

8.6.3.5　Kerberos 协议模型的分析

可以证明 Kerberos 协议模型功能的正确性，它具有以下优点：

(1) Kerberos 协议引入了可信第三方，使所需的共享密钥分配和管理变得十分简单；AS 担负鉴别工作，减轻应用服务器的负担；安全相关数据的集中管理和保护，从而使攻击者的入侵很难成功。

(2) Kerberos 协议引入票据的概念，使 AS 的鉴别结果和会话密钥安全地传送给应用服务器；在生存期内可重用，降低用户口令的使用频度，更好地保护口令提高方便性；减轻 AS 的负担，提高鉴别系统的效率。

(3) Kerberos 协议使用时间戳防止对票据(Ticket)和鉴别码(Authenticator)的重放攻击。

Kerberos 协议模型也具有以下弱点：

(1) Kerberos 协议模型依赖于安全的时间服务。实现较好的时钟同步往往是很困难的；攻击者误导系统时间并进行重放攻击是有可乘之机的。

(2) 两种版本都容易受到猜测口令攻击，脆弱口令容易受到攻击，协议模型未对口令提供额外的保护。

(3) 鉴别系统本身的程序完整性很难保证，存在篡改登录程序的威胁。

(4) 密钥存储问题，口令及会话密钥无法安全存放于典型的计算机系统中。

综上所述，Kerberos 协议是一个基于口令的利用对称密码技术建立起来的鉴别协议，可适用于分布式网络环境。

8.6.4 X.509 鉴别服务

OSI目录检索服务标准X.500首先公布于1988年，该标准中包括了一部分陈述鉴别的标准，即 ISO/IEC 9594—8 或 ITU-T X.509 建议。根据分析的结果，1993 年和 1995 年分别对 X.509 建议做了微小修改。

在X.509中规定了几个可选的鉴别过程，所有这些过程都是基于公钥加密的，假设双方互相知道对方的公钥。获得对方的证书，也就知道了对方的公钥。一方可以通过访问CA提供的目录获得对方的证书，也可以由对方将自己的证书放在初始报文里。然后，就可以通过比较简洁的协议进行相互鉴别。X.509的三种鉴别过程分别是单向鉴别、双向鉴别和三向鉴别。

8.6.4.1 单向鉴别

单向鉴别只需要一次通信。用户 A 向用户 B 发送一条消息，证明自己的身份。消息内容如下：

$$\mathrm{A}\rightarrow\mathrm{B}:\mathrm{E}_{\mathrm{KRa}}\{t_{\mathrm{A}},r_{\mathrm{A}},ID_{\mathrm{B}},\mathrm{sgnData},\mathrm{E}_{\mathrm{KU_b}}[K_{\mathrm{ab}}]\}$$

其中时间戳 t_{A} 包含**起始**时间和终止时间，用来防止报文的延迟传送；r_{A} 是一个临时交互号，在一个有效期内是唯一的，B 需要保存 r_{A} 直到有效期结束；$\mathrm{ID_B}$ 是 B 的身份信息；sgnData 是签名数据；最后一项是用 B 的公钥加密的会话密钥。$\mathrm{E_{KRa}}\{\}$表示整个报文用 A 的私钥签名，就可以完成对用户 A 的身份鉴别。由于使用了时间戳，需要时钟同步。

8.6.4.2 双向鉴别

双向鉴别在单向鉴别的基础上增加了一次通信，过程如下：

$$\mathrm{A}\rightarrow\mathrm{B}：\mathrm{E}_{\mathrm{KRa}}\{t_{\mathrm{A}},r_{\mathrm{A}},\mathrm{ID_B},\mathrm{sgnData},\mathrm{E}_{\mathrm{KU_b}}[K_{\mathrm{ab}}]\}$$

$$\mathrm{B}\rightarrow\mathrm{A}：\mathrm{E}_{\mathrm{KRb}}\{t_{\mathrm{B}},r_{\mathrm{B}},\mathrm{ID_A},r_{\mathrm{A}},\mathrm{sgnData},\mathrm{E}_{\mathrm{KU_a}}[K_{\mathrm{ba}}]\}$$

应答消息包含 A 发送的临时交互号，以及其他信息，第二条消息同样要由 B 签名。

8.6.4.3 三向鉴别

三向鉴别又在最后增加了一个 A 向 B 的应答。过程如下：

$$\mathrm{A}\rightarrow\mathrm{B}：\mathrm{E}_{\mathrm{KRa}}\{t_{\mathrm{A}},r_{\mathrm{A}},\mathrm{ID_B},\mathrm{sgnData},\mathrm{E}_{\mathrm{KU_b}}[K_{\mathrm{ab}}]\}$$

$$\mathrm{B}\rightarrow\mathrm{A}：\mathrm{E}_{\mathrm{KRb}}\{t_{\mathrm{B}},r_{\mathrm{B}},\mathrm{ID_A},r_{\mathrm{A}},\mathrm{sgnData},\mathrm{E}_{\mathrm{KU_a}}[K_{\mathrm{ba}}]\}$$

$$\mathrm{A}\rightarrow\mathrm{B}：\mathrm{E}_{\mathrm{KRa}}\{r_{\mathrm{B}}\}$$

最后一条报文包含对临时交互号 r_{B} 的签名，这样就无须检查时间戳，因为每一端都可以通过检查返回的临时交互号来探测重放攻击。所以三向鉴别不需要时钟同步，在没有同步条件时，必须采用这种方法。

与Kerberos 协议相比，X.509 鉴别交换协议有一个很大的优点：X.509 不需要物理上安全的在线服务器，因为一个证书包含了一个认证授权机构的签名。X.509 双向交换拥有 Kerberos 的缺陷，即依赖于时戳，而 X.509 三向交换克服了这一缺陷。

思考和练习题

(1) 实现身份鉴别的途径有哪三种？
(2) 消息鉴别和身份鉴别的区别和联系是什么？
(3) 如何对抗口令的泄露和猜测攻击？
(4) 在鉴别和交换协议中如何对抗重放攻击？
(5) UNIX 系统中的口令存储中采用了加 salt 的办法，有什么好处？
(6) 什么是弱鉴别、强鉴别和近似的强鉴别？
(7) Kerberos 鉴别协议如何实现口令的非明文传输?
(8) Kerberos 鉴别协议的优缺点有哪些？
(9) 零知识证明的原理是什么？

实践/实验题

(1) 编写完成下列功能的程序：口令的加 salt 存储和鉴别。
(2) 编写实现 S/KEY 协议基本过程的程序，要求根据提示的次数完成口令的输入和鉴别。
(3) 请调研 3～5 家网上银行的账户鉴别机制，分析其安全性，比较其异同。

第9章 访 问 控 制

当用户的身份通过鉴别之后，是否可以任意使用系统内的资源呢？答案是否定的，UNIX系统的用户有如下类型：特殊的用户、一般的用户、做审计的用户和作废的用户，系统需要为不同的用户分配不同的权限。特殊的用户是系统管理员，具有最高级别的特权，可以访问任何资源，并具有任何类型的访问操作能力。一般的用户是最大的一类用户，他们的访问操作受到一定限制，由系统管理员分配。做审计的用户负责整个安全系统范围内的安全控制与资源使用情况的审计。作废的用户是被系统拒绝的用户。如何使各用户在授权范围内合法使用资源呢？访问控制是用户身份得到鉴别之后，信息系统的另一道保护措施，也就是国际标准 ISO 7498—2 定义的第 5 类安全服务。

9.1 访问控制的有关概念

简单地说，访问控制是一种针对越权使用资源的防御措施。它的基本目标是防止对任何资源(如计算资源、通信资源或信息资源)进行未授权的访问。从而使计算机系统在合法范围内使用；决定用户能做什么，也决定代表一定用户利益的程序能做什么。未授权的访问包括：未经授权的使用、泄露、修改、销毁信息以及颁发指令等，它可能是非法用户进入系统，也可能是合法用户对系统资源的非法使用。

访问控制对安全服务的其他方面有着直接的支持，从访问控制的定义，不难看出，访问控制对机密性、完整性起直接的作用。对于可用性，访问控制通过对以下信息的有效控制来实现：谁可以颁发影响网络可用性的网络管理指令；谁能够滥用资源以达到占用资源的目的；谁能够获得可以用于拒绝服务攻击的信息。通过对机密性、完整性的支持，访问控制可以保护存储在某些机器上的个人信息或重要信息的保密性，维护机器内系统的完整性，减少病毒感染的机会。

访问控制包括三个要素：主体、客体和授权。访问控制通过某种途径显式地准许或限制主体对客体的访问能力及范围。

- **客体**(Object)是需要保护的资源，又称为目标(Target)。凡是可以被操作的信息、设备等资源都可以被认为是客体。如磁盘、远程终端、数据库中的数据、应用资源、信息管理系统的事务处理及其应用等。
- **主体**(Subject)或称为发起者(Initiator)，是一个可以访问该资源的主动实体，通常指用户或代表用户执行的程序。
- **授权**(Authorization)规定主体可对客体执行的动作。例如，操作系统设置的用户对文件的访问模式(Access mode)有：读-复制(Read-copy)、写-删除(Write-delete)、运行(Execute)和拒绝访问(Null)。授权策略是用于确定一个主体是否能对客体拥有访问能力的一套规则，是访问控制的核心。

主客体的关系是相对的。当用户启动 Office 应用程序时，用户是主体，Office 应用程序是客体，当 Office 应用程序打开文件 A 时，Office 应用程序是主体，文件 A 是客体。

访问控制与其他安全措施之间的关系，可以用图9.1来说明。当用户身份经过鉴别之后，负责实施系统安全策略的软硬件组合体——**引用监控器**(Reference monitor)执行访问控制机制，对用户访问某项资源的请求做出允许或者拒绝的决策。用户访问资源的行为可以被审计机制记录下来。

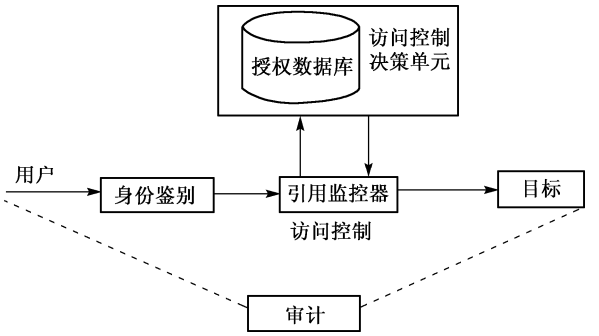


图 9.1　访问控制与其他安全措施的关系

访问控制系统可以分为两部分，如图9.2所示。一部分完成访问控制的决策功能，一部分完成访问控制的实施功能，分别对应对访问控制策略(Access control policy)和访问控制机制(Access control mechanisms)。**访问控制策略**在系统安全策略级上表示授权，是对访问如何控制，如何做出访问决定的高层指南。**访问控制机制**是访问控制策略的软硬件低层实现。访问控制机制与访问控制策略独立，同一安全机制可支持不同的安全策略。安全策略之间没有更好的说法，只是一种可以比一种提供更多的保护，应根据应用环境灵活使用。

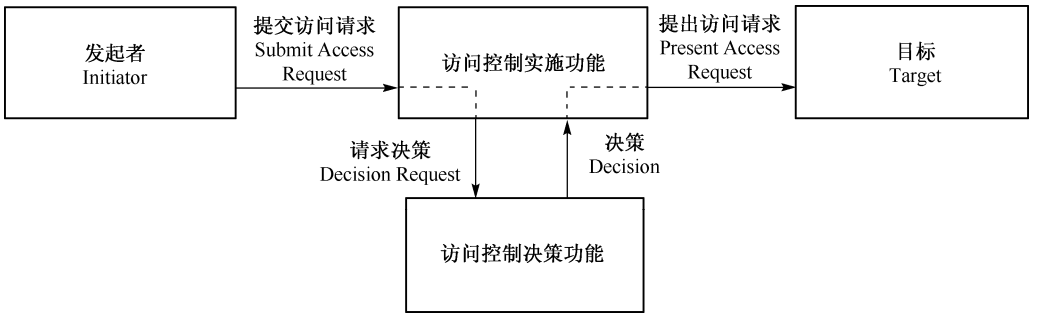


图 9.2　访问控制系统的基本组成

根据应用环境的不同，访问控制主要有以下三种：网络访问控制、操作系统访问控制和应用程序访问控制。访问控制机制应用在网络环境中，主要是限制用户可以建立什么样的连接以及通过网络传输什么样的数据，这就是传统的网络防火墙。通常，操作系统借助访问控制机制来限制对文件及系统设备的访问，主流的操作系统均提供不同级别的访问控制功能。访问控制也可以嵌入应用程序(或中间件)中，提供更细粒度的数据访问控制或者限制对应用程序的不同功能模块具有不同的操作权限。当访问控制需要基于数据记录或更小的数据单元实现时，应用程序将提供其内置的访问控制模型。例如，大多数数据库(如Oracle)都提供独立于操作系统的访问控制机制。OA 系统也可以设计成不同用户具有不同的操作使用权限。

根据访问控制策略，访问控制模型可以分为：自主访问控制、强制访问控制、基于角色的访问控制、基于使用的访问控制、基于任务和工作流的访问控制、基于属性的访问控制等。下面各小节将分别进行介绍。

9.2 自主访问控制

自主访问控制 DAC 也称为基于身份的访问控制(Identity Based Access Control)或任意访问控制。它的特点是根据主体的身份及允许访问的权限进行决策。自主是指具有某种访问能力的主体能够自主地将访问权的某个子集授予其他主体。因为其灵活性高，被大量采用。Linux, UNIX, Windows NT 或 Server 版本的操作系统都提供自主访问控制的功能，允许系统中信息的拥有者按照自己的意愿去决定，谁可以以何种方式访问该客体，对于信息的拥有者来说是自主的。

任何访问控制策略最终均可被模型化为访问矩阵形式：行对应于主体，列对应于客体，每个矩阵元素规定了主体对相应的目标被准予的访问模式。访问控制矩阵模型最早由 Butler Lampson 于1971年提出，Graham 和 Denning 对它进行了改进。表 9.1 是访问控制矩阵的示例，R, W, Own 分别表示读、写和拥有。访问控制矩阵模型简单、易懂、通用性强，但在大型系统中，访问矩阵非常庞大，需要消耗大量存储空间，由于每个主体访问的客体有限，这种矩阵通常是稀疏的，因此，访问控制矩阵可以表示成<主体，客体，权限>形式的三元组，被称为授权关系表，但是搜索这么大数量的三元组效率太低了。两种流行的实现机制是**访问控制表**(Access Control List，ACL)和**能力表**(Capabilities List，CL)。访问控制矩阵按列看是访问控制表内容，按行看是访问能力表内容。

表 9.1 访问控制矩阵

客体 / 主体	文件 1	文件 2	文件 3
Alice	R, W, Own	R	R, W
Bob	R	R, W, Own	
John	R, W	R	R, W, Own

9.2.1 访问控制表

访问控制表对应于访问控制矩阵中一列的内容，每个客体附加一个可以访问它的主体及相应权限的明细表。设 S 表示系统中主体的集合，R_S 表示权限的集合，访问控制表是有序对的集合 $\mathrm{ACL}=\{(S_i, R_{S_i}) \mid S_i \in S, R_{S_i} \in R\}$。与表 9.1 对应的文件 1 的访问控制表是

$$\mathrm{ACL}(\text{文件 1})=\{(\text{Alice}, \{\text{R, W, Own}\}), (\text{Bob}, \{R\}), (\text{John}, \{\text{R, W}\})\}$$

如图9.3 所示。基于身份的访问控制策略和基于角色的访问控制策略都可以用 ACL 来实现。它的优点是控制粒度比较小，适用于被区分的用户数比较小的情况，并且这些用户的授权情况相对比较稳定的情形。

基于访问控制表的主体 s 对客体的访问控制方式如图9.4所示，主体 s 提交了对某一客体 o 的访问请求 r，引用监控器查看该客体的访问控制表，ACL 中包含了可以访问该客体的主体的身份和权限，据此判断主体 s 是否可以访问客体 o，并且是否具有访问权限 r。

基于身份的策略可以基于个人或者基于组来实现。当一组用户对于一个客体具有同样的访问权限，可以把访问矩阵中的多个行压缩为一个行。实际使用时，先定义组的成员，对用户组进行授权，同一个组可以被重复使用，组的成员可以改变。基于组的策略在表示和实现

上更容易和更有效。在基于个人的实现方式中，对于系统中每一个需要保护的客体，要为其附加一个访问控制表，表中包括主体标识符(ID)和对该客体的访问模式。在基于组的实现方式中，将属于同一部门或工作性质相同的人归为一组(Group)，分配组名 GN，访问按照组名判断，通配符“*”可以代替任何组名或者主体标识符。ACL 条目的形式为(user, group, rights)。假设对文件 1 有如下访问控制表：

ACL(文件 1)={(*, job1, {R, X}), (Alice, job1, {R, W, X}), (Bob, *, {R}), (*, *, *N*)}

(*, job1, {R, X})表示属于 job1 组的所有主体都对文件 1 具有读(R)和执行(X)的权利；(Alice, job1, {R, W, X})表示只有 job1 组中的主体 Alice 才额外具有写(W)的权限；(Bob, *, {R})表示无论属于哪一组中的 Bob 都可以读文件 1；(*, *, *N*)说明所有其他的主体，无论属于哪个组，都不具备对文件 1 的任何访问权限。基于组的实现方式可以大大缩小访问控制表，并且满足访问控制的需要。

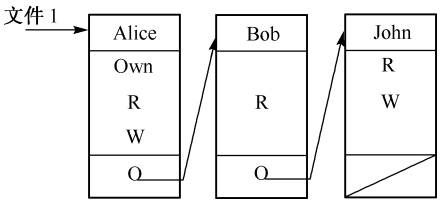


图 9.3　与表 9.1 对应的文件 1 的访问控制表(ACL)

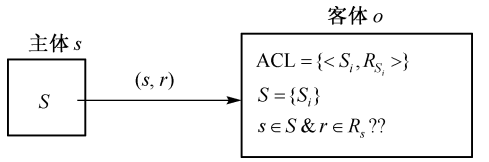


图 9.4　基于 ACL 的访问控制方式

使用访问控制表的一个前提是需要设置一个隐含的、或者显式的默认策略，这可以方便用户的使用，同时也在很大程度上避免了文件泄露的可能。例如，全部权限否决，除非指定特定的主体对特定客体的权限。对主体的权限分配应遵循最小特权原则，要求最大限度地限制每个用户实施授权任务所需要的许可集合。在不同的环境下，默认策略不尽相同。例如，在公开的布告板环境中，所有用户都可以得到所有公开的信息，对于特定的用户，有时候需要提供显式的否定许可，比如，对于违纪的内部员工，禁止访问内部一些信息。

利用访问控制表，能够很容易地判断出对于特定客体，哪些主体可以访问并有哪些访问权限，但在查询特定主体能够访问的客体时，需要遍历查询所有客体的访问控制表。同样很容易撤销主体对特定客体的访问，只要把该主体的权限从客体的访问控制表中删除。所有权限中最重要的是拥有权，具有拥有权的所有者可以修改 ACL，如果拥有权不能控制权限的授予，则权限的撤销就会变得更为复杂。

ACL 是一种成熟且有效的访问控制实现方法，许多通用的操作系统使用访问控制表来提供访问控制服务。例如UNIX系统利用访问控制表的简略方式，允许以少量工作组的形式实现访问控制表，而不允许单个的主体出现，这样可以使访问控制表很小而能够用几位就可以和文件存储在一起。另一种复杂的访问控制表应用是利用一些访问控制包，通过它制定复杂的访问规则限制何时和如何进行访问，而且这些规则根据用户名和其他用户属性的定义进行单个用户的匹配应用。

9.2.2　能力表

能力表对应于访问控制矩阵中一行的内容，与ACL 相反，是以主体为索引建立访问权限表，表中规定了该用户可访问的文件名及访问权限。设O表示系统中客体的集合，R_O表示权

限的集合，能力表 CL 是有序对的集合 $CL=\{(O_i,R_{O_i})\mid O_i\in O,R_{O_i}\in R_O\}$。与表 9.1 对应的 Alice 的能力表是 CL(Alice)={(文件 1，{R, W, Own})，(文件 3，{R, W})}，如图 9.5 所示。

基于能力表的主体 *s* 对客体的访问控制方式如图9.6所示，主体 *s* 提交了对某一客体 *o* 的访问请求*r*，引用监控器查看主体出示的能力表，能力表中封装了主体可以访问的客体的身份和权限，据此判断主体 *s* 是否可以访问客体 *o*，并且是否具有访问权限 *r*。

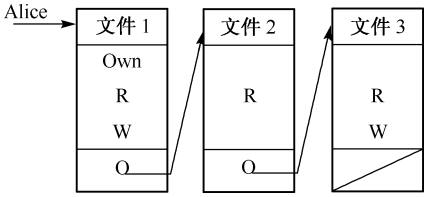


图 9.5 与表 9.1 对应的 Alice 的能力表(CL)

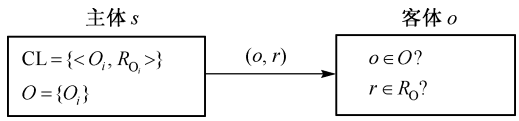


图 9.6 基于能力表的主体 *s* 对客体 *o* 的访问的控制方式

利用能力关系表可以很方便查询一个主体的所有授权访问。相反，检索具有授权访问特定客体的所有主体，则需要遍历所有主体的能力表。在能力表系统中，要撤销一个客体的访问权就要废除所有对该客体有访问权的能力表，理论上，这种操作的开销是不可接受的，实际常使用其他的替代方法。主体具有的能力相当于一个凭证，比如“入场券”，是用户登录系统时，由操作系统赋予的一种权限标记。如果主体的能力包括“转授”访问权限，具有这种能力的主体就可以把自己的能力复制传递给其他主体。

9.2.3 DAC 的授权管理

授权的管理决定谁能被授权修改允许的访问权限，用**访问许可**(access permission)来描述，是决定主体对客体具有何种访问能力的。访问许可定义了改变访问模式的能力或向其他主体传送这种能力的能力。在 DAC 模式下，访问许可有如下管理方式：

- **集中式管理**：只有单个的管理者或组可以对用户进行访问控制授权和授权撤销。
- **分级式管理**：一个中心管理者把管理责任分配给其他管理员，这些管理员再对用户进行访问授权和授权撤销。分级式管理可以根据单位组织结构实行，其控制关系与部门的组织结构对应，容易获得用户单位的认可。在图 9.7 所示的分级架构中，最高领导或者系统管理员具有最高控制权，可以修改系统中所有对象的ACL，最底层的成员对任何客体都不具有访问许可。这种方式的缺点是一个客体有多个主体对其有控制权，发生问题后有责任界定问题。

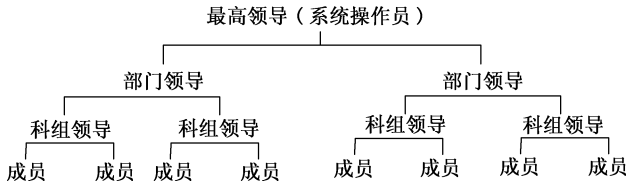


图 9.7 访问许可的分级架构

- **所属权管理**：对每个客体设置一个拥有者(通常是客体的生成者)。拥有者是唯一有权修改客体访问控制表的主体，拥有者对其客体具有全部控制权，但无权将控制权转授

给其他主体。这种方式下，仍然需要一个能够修改系统中所有客体ACL的系统管理员，以解决客体的拥有者离开组织而无法删除客体的问题。

- **协作式管理**：对于特定系统资源的访问不能由单个用户授权决定，而必须要其他用户的协作授权决定。
- **分散式管理**：在分散管理中，客体所有者可以把管理权限授权给其他用户。

由于 DAC 对用户提供灵活和易行的数据访问方式，能够适用于许多系统环境，所以 DAC 被大量采用，尤其在商业和工业环境的应用上。然而，DAC 提供的安全保护容易被非法用户绕过而获得访问，因为信息在移动过程中其访问权限关系会被改变。例如，用户 A 可将其对目标 *O* 的访问权限传递给用户 B，从而使不具备对 *O* 访问权限的 B 可访问 *O*。因而 DAC 不能够对系统资源提供充分的保护，不能抵御特洛伊木马的攻击。如果用户的一个合法程序中隐藏了木马，就可能造成敏感信息的泄露，木马程序可以修改文件的访问权限或者对敏感信息进行复制。对安全性要求更高的系统来说，仅采用DAC机制就不够了，强制访问控制机制是一个更好的选择。

9.3　强制访问控制 MAC

强制访问控制(Mandatory Access Control，MAC)也被称为基于规则的访问控制(Rule Based Access Control)，它的特点是访问控制取决于能用算法表达的并能在计算机上执行的策略。策略给出资源受到的限制和对实体的授权，对资源的访问取决于对实体的授权策略而非简单地取决于实体的身份，比如还取决于信息流动的控制。在自主访问控制中，访问控制负责实体本身的访问管理，但并不管理实体中的信息和它所涉及的内容，缺乏对信息流动的保护，因此造成了信息的泄露。信息流控制规定信息可以流通的有效途径，是对访问控制的重要补充。两种最基本的信息流策略模型是保证保密性的Bell-LaPadula模型和保证完整性的Biba模型。

在强制访问控制中，系统包含主体集 *S* 和客体集 *O*，每个主体集 *S* 中的主体 *s* 及客体集 *O* 中的客体 *o*，都属于一个固定的安全类 SC(Security Classes)，安全类 SC=<*L*, *C*>包括两个部分：有层次的安全级别 *L* 和无层次的安全范畴 *C*。有层次的安全级别 *L* 构成一个偏序关系。例如，在军事信息系统中，所有的信息被划分为无密级(Unclassified)、秘密级(Secret)、机密级(Confidential)和绝密级(Top Secret)四个从低到高的安全级别。在一个企业内部，有财务部、人事部和技术部等不同部门，只有本部门的人才有权访问本部门的信息。

主客体的安全级别是由安全标签来表示的，是限制在目标上的一组安全属性信息项。在访问控制中，一个安全标签隶属于一个用户、一个目标、一个访问请求或传输中的一个访问控制信息。在处理一个访问请求时，目标环境比较请求上的标签和目标上的标签，应用策略规则(如 Bell-Lapadula 规则)决定是允许还是拒绝访问。

为了保证信息流的单向性，主体对客体的访问方式有以下四种。

- **下读(read down)**：即主体的安全级别大于等于客体的安全级别时，允许主体对客体的读操作。
- **上写(write up)**：即主体的安全级别小于等于客体的安全级别时，允许主体对客体的写操作。
- **下写(write down)**：即主体的安全级别大于等于客体的安全级别时，允许主体对客体的写操作。

- **上读(read up)**：即主体的安全级别小于等于客体的安全级别时，允许主体对客体的读操作。

9.3.1 Bell-LaPadula 模型

Bell-LaPadula(BLP)模型是最早的一种计算机多级安全模型，结合了自主访问控制和强制访问控制，它建立的访问控制原则可以用下列两项简单表示：

- **简单安全特性(无上读)**：仅当 $l(o) \leqslant l(s)$ 且 s 对 o 具有自主型读权限时，s 可以读取 o。
- ***-特性(无下写)**：仅当 $l(s) \leqslant l(o)$ 且 s 对 o 具有自主型写权限时，s 可以写 o。

其中 s 对 o 具有自主型读(写)权限，就对应于自主访问控制模型中的读(写)权限，换句话说，如果没有强制型控制，s 就可以读(写)o。BLP 模型的信息流规则如图 9.8 所示。

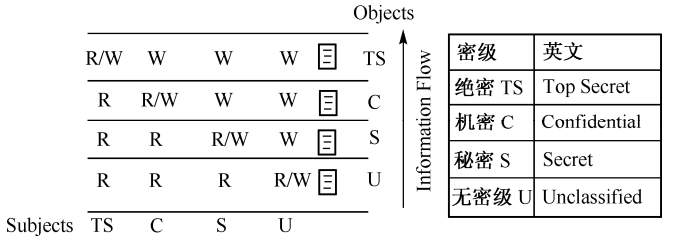


图 9.8 BLP 模型的信息流规则

应用 Bell-LaPadula 模型实现机密性保护的一个例子是利用防火墙来实现单向访问机制，如图9.9所示。不允许敏感数据从内部网络(例如，其安全级别为“机密”)流向 Internet(安全级别为“公开”)，所有的内部数据被标志为“机密”。防火墙提供“无上读”功能来阻止Internet对内部网络的访问，提供“无下写”功能来限制机密数据流出，例如，阻止任何外发的邮件。

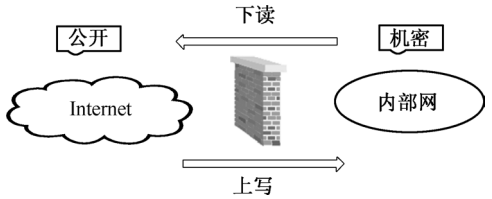


图 9.9 应用 Bell-LaPadula 模型的例子

9.3.2 Biba 模型

Biba 模型模仿BLP 模型的信息保密性级别，定义了信息的完整性级别。在信息流的流动方向方面不允许从级别低的进程到级别高的进程，也就是说，主体只能向比自己安全级别低的客体写入信息，从而阻止低安全级别用户修改高安全级别的信息，保证信息的完整性。在 Biba 模型中，系统包含主体集合 S，客体集合 O 和一个完整性集合 I，每个主体集 S 中的主体 s 及客体集 O 中的客体 o，都属于一个固定的完整性级别 i，这些级别是有序的，它遵循以下原则：

- **无上写**：当且仅当 $i(s) \geqslant i(o)$，$s \in S$ 可以写入 $o \in O$。
- **无下读**：当且仅当 $i(o) \geqslant i(s)$，$s \in S$ 可以读取 $o \in O$。

Biba模型在应用中的一个例子是对 Web 服务器的访问过程，如图9.10所示。定义Web 服

务器上发布的资源安全级别为“秘密”，Internet上用户的安全级别为“公开”，依照Biba 模型，Web 服务器上数据的完整性将得到保障，Internet 上的用户只能读取服务器上的数据而不能更改它，因此，任何“POST”操作将被拒绝。

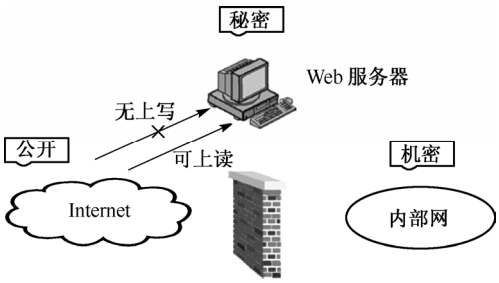


图 9.10　基于 Biba 模型对 Web 服务器的访问保护

在强制访问控制中，允许的访问控制完全是根据主体和客体的安全级别决定。其中主体(用户、进程)的安全级别是由系统安全管理员赋予用户，而客体的安全级别则由系统根据创建它们的用户的安全级别决定。因此，强制访问控制的管理策略是比较简单的，只有安全管理员能够改变主体和客体的安全级别。

自主访问控制具有配置粒度小的优点，但配置的工作量大，效率低，强制访问控制配置粒度大，缺乏灵活性，但可以提供更高的安全性。

9.4　基于角色的访问控制 RBAC

9.4.1　RBAC 的概念和安全原则

基于角色的访问控制(Role-Based Access Control，RBAC)于1992年首次提出，是与现代商业环境相结合的产物。在实际的组织中，为了完成组织的业务工作，需要设置不同的职位，职位既表示一种业务分工，也代表不同的责任和权利。例如，在一个银行系统中，有出纳员、分行管理者、系统管理员、顾客、审计员等职位，在 RBAC 模型中，可以根据职位定义相应的角色。基于角色的访问控制是一个复合的规则，可以被认为是 DAC 和 MAC 的变体。它的基本思路是管理员创建角色，给角色分配权限，给用户分配角色，角色所属的用户可以执行相应的权限。角色起源于 UNIX 系统或别的操作系统中组的概念，每个角色与一组用户和有关的动作相互关联，角色中所属的用户可以有权执行这些操作，而用户组属于具体的实现机制，是用来实现角色的工具。RBAC 与传统的访问控制的差别在于增加一层间接性，带来了灵活性，如图9.11所示。一个公司的职员可能会变化较多，但是公司的职能却很少变化，职位，即对应的角色是相对稳定的，用户通过角色访问资源，建立这样一种映射关系，大大提高了管理的效率，减少了授权管理的复杂性，降低了管理开销。

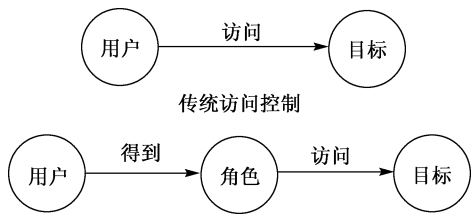


图 9.11　RBAC 与传统访问控制的差别

在给用户分配角色和给角色分配权限时，RBAC模型遵循三条公认的安全原则：最小权限、责任分离(separation of duties)和数据抽象原则。

最小权限原则要求用户所拥有的权力不能超过他执行工作时所需的权限。在 RBAC 中，依据企业或组织内的规章制度、岗位工作内容，确定执行该项工作的最小权限集，然后将用户限制在这些权限范围之内。

责任分离是多个互斥的角色合作完成重要工作时的原则，分为静态责任分离和动态责任分离两种。**静态责任分离**的原则是只有当一个角色与用户所属的其他角色彼此不互斥时，该角色才能授权给该用户。**动态责任分离**的原则是只有当一个角色与一个主体的任何一个当前活跃角色都不互斥时，该角色才能成为该主体的另一个活跃角色。

数据抽象指的是在 RBAC 中可以定义抽象的权限，而不仅仅是操作系统中的读、写、执行等。RBAC 支持数据抽象的程度与 RBAC 模型的实现细节有关。

9.4.2 NIST-RBAC 参考模型

国外的许多机构从 1990 年开始就在为定义 RBAC 标准而工作，并且推动这项技术的研究和开发。Ravi Sandhu 等人于 1996 年提出了著名的 RBAC96 模型，1997 年，他们更进一步提出了一种分布式 RBAC 管理模型 ARBAC97，2001 年 8 月，NIST 发表了 RBAC 建议标准。其中，NIST-RBAC 参考模型最为业界认可。该标准包括三个从简单到复杂的模型：基本 RBAC（Core RBAC），分级 RBAC（Hierarchical RBAC）和有约束的 RBAC（Constrained RBAC）模型。

9.4.2.1 基本 RBAC 模型

基本 RBAC 模型如图9.12所示，包括 5 个基本数据元素：用户（Users）、角色（Roles）、目标（Objects，OBS）、操作（operations，OPS）和许可权（permissions，PRMS）。用户就是一个可以独立访问计算机系统中的数据或者用数据表示的其他资源的主体，可以是人、设备或进程等。操作是一个可执行的动作，RBAC 的操作类型依赖于其实现系统的类型，在文件系统中，操作可能包括浏览、复制、打印等；在数据库系统中，操作可能包括插入、删除、更新等。许可权是对被保护目标执行操作 OPS 的许可。用户委派（User assignment relations，UA）是用户与角色之间的一个二元关系，表示用户被委派了一个角色。权限分配（Permission assignment relations，PA）是角色与权限之间的一个二元关系，表示给角色分配权限。用户与角色、权限与角色的映射关系是多对多的映射关系。一个用户可以有许多角色，一个角色也可以被分配给很多用户。同一个权限可以分配给许多角色，单一的角色可以赋予很多权限。用户建立会话，通过会话激活角色，一个用户通过会话对应多个角色，每一个会话对应一个用户。会话-角色（Session_roles）是用户与激活的角色集合之间的映射。会话由用户控制，允许动态激活或者取消角色，实现最小特权。应避免同时激活所有角色，会话和用户的分离可以解决同一用户多账号带来的问题，如审计、计账等。

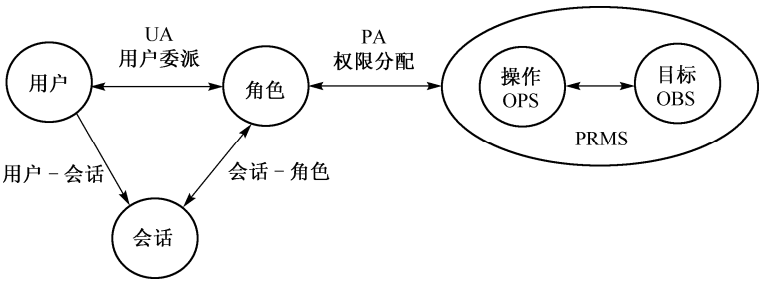


图 9.12 基本 RBAC 模型

9.4.2.2　等级 RBAC 模型

等级 RBAC 模型在基本 RBAC 模型的基础上定义了角色等级(Role Hierarchy, RH)，如图 9.13 所示。角色的结构化分层是反映一个组织授权和责任的自然方式。角色等级定义了角色之间的继承关系，如果角色 r_2 的所有权限都是 r_1 的权限，我们就说角色 r_1 继承(Inherit)角色 r_2。

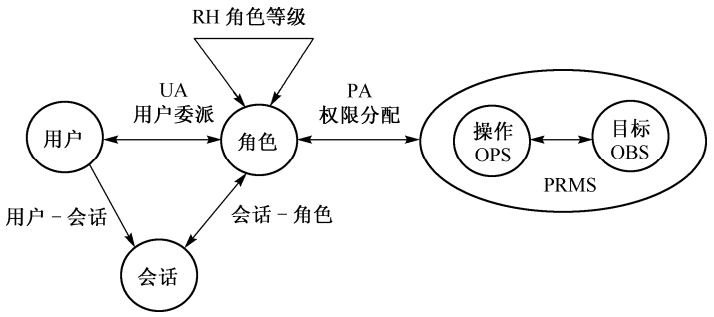


图 9.13　等级 RBAC 模型

9.4.2.3　有约束的 RBAC

有约束的 RBAC 模型中，增加了一种称为责任分离的约束关系，用于解决利益的冲突，防止用户超越权限。RBAC 允许两种责任分离：静态责任分离(Static Separation of Duty Relations，SSD)和动态责任分离(Dynamic Separation of Duty relations，DSD)。在静态责任分离中，对用户分配的角色进行约束，也就是当用户被分配给一个角色时，禁止其成为第二个互斥的角色。SSD 定义了一个用户角色分配的约束关系，一个用户不可能同时分配 SSD 中的两个角色。SSD-RBAC 模型如图9.14所示。与SSD类似，DSD 也要限制一个用户的许可权。SSD 直接在用户的许可空间进行约束，DSD 通过对用户会话过程进行约束，使用户在不同的时间拥有不同的权限，对最小特权提供支持。DSD 允许用户被授予不产生利益冲突的多个角色。DSD-RBAC 模型如图 9.15 所示。

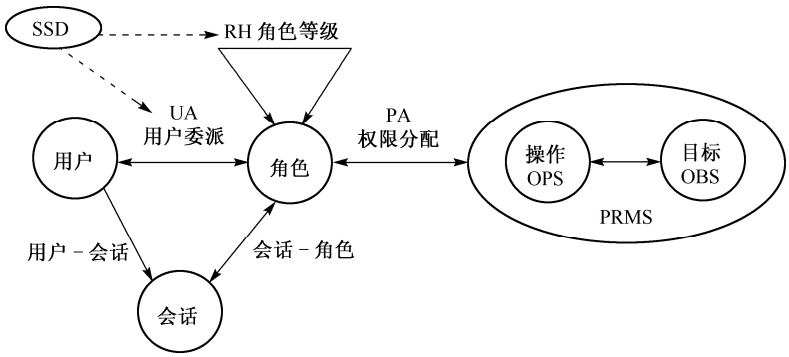


图 9.14　SSD-RBAC 模型

RBAC 模型具有以下优势：

- **便于授权管理**：例如，系统管理员需要修改系统设置等内容时，必须有几个不同角色的用户到场方能操作，从而保证了安全性。此外，基于角色的访问控制提供了类似自

主访问控制的许多管理策略。而且，管理权限的委托代理是角色访问控制管理的重要特点，在以上的两种访问控制的管理策略中都不存在。

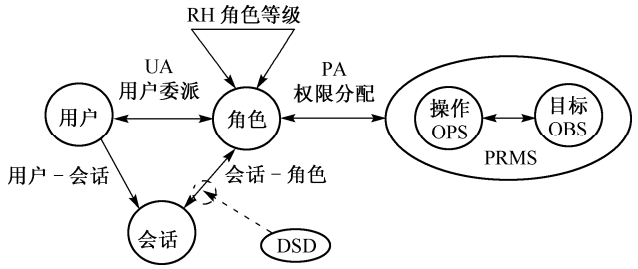


图 9.15 DSD-RBAC 模型

- **便于根据工作需要分级**：例如，企业财务部门与非财务部门的员工对企业财务的访问权就可由财务人员这个角色来区分。
- **便于赋予最小特权**：例如，即使用户被赋予高级身份时也未必一定要使用特权，以便减少损失，只有必要时方能拥有特权。
- **便于任务分担**：不同的角色完成不同的任务。
- **便于文件分级管理**：文件本身也可分为不同的角色，如信件、账单等，由不同角色的用户拥有。

基于角色的访问控制策略的陈述易于被非技术的组织策略者理解，同时也易于映射到访问控制矩阵或基于组的策略陈述。同时具有基于身份策略的特征和基于规则的策略的特征。

9.5 其他访问控制策略

在过去30多年中，访问控制一直以访问矩阵模型为基础，访问矩阵的核心概念就是向主体授予某种权限，使主体可以以某种模式访问客体。访问控制主要有三类：自主访问控制(DAC)、强制访问控制(MAC)以及基于角色的访问控制(RBAC)，它们已经不能满足新的应用需求，因此研究人员又提出了使用控制、基于任务(Task-Based)的访问控制、基于属性的访问控制、信任管理(Trust Management)和数字版权管理(DRM)等新概念和方法，这些都从不同方面对访问控制进行了一定的扩展。

9.5.1 使用控制

2002 年，GMU 的 Jaehong Park 和 Ravi Sandhu 提出了一种新的访问控制——使用控制(Usage Control，UCON)，并把它称为下一代的访问控制，其中使用(Usage)的含义为对客体权限的使用。使用控制的核心模型是一系列 ABC 模型(也称为 ABC 模型族)，它们是围绕着三个决策性因素：授权(Authorization)，义务(Obligation)和条件(Condition)而建立的。和传统的访问控制一样，授权基于主体和客体的属性；义务是指主体为了获取或维持对客体的访问权限必须完成的动作，例如在许可协议上点击“ACCEPT”等；条件是指环境或系统因素，例如系统负载等。除了授权、义务以及条件这三个因素，ABC 还集成了可变属性以及继续执行(Ongoing enforcement)，因而 ABC 涵盖了 DAC, MAC, RBAC 等传统访问控制策略以及信任管理、DRM 等新概念和方法。它是第一个对访问控制矩阵做了系统而全面扩展的模型。

传统访问控制的共性就是授权决策是基于主体和客体的属性以及请求的权限。主体和客体的属性包括身份、能力或性质。尽管这种传统访问控制在很多信息系统中得到应用，今天的信息系统产生了新的需求，例如，一个用户在下载某公司产品的白皮书之前被要求必须在许可协议上点击“ACCEPT”或者填写表单，在这种情况下，使用决策不是基于主体/客体的属性而是必需动作的完成。这种决策因素也称为“义务”。

在某些情况下，由于环境或系统状态需要对资源的访问和使用进行一定的限制。例如，某些资源只能在一定的位置或业务时间使用。这种决策因素称为“条件”。

除了授权、义务和条件之外，信息系统的安全决策还需要考虑两个性质：持续性(Continuity)和可变性(Mutability)。在传统访问控制中，授权是在访问客体之前进行(pre)。可以对授权进行扩展，在对客体进行访问(使用)时根据使用需求仍可进行授权(Ongoing)，这就是持续性。在某些信息系统中，如DRM，属性会由于主体的动作而发生改变。在传统访问控制中，很少对这种性质的可变性进行讨论。如果属性是可变的，属性的更新发生在使用前(pre)、使用中(Ongoing)或使用后(post)。

使用控制(UCON)集成了授权、义务和条件，并包含了持续性和可变性，对传统的访问控制进行了扩展，使其成为了符合现代应用需求的访问控制策略。

9.5.2　基于任务的访问控制

基于任务的访问控制(TBAC)模型是一种以任务为中心的，并采用动态授权的主动安全模型。该模型的基本思想是：授予用户的访问权限，不仅仅依赖主体，客体，还依赖于主体当前执行的任务，任务的状态。当任务处于活动状态时，主体拥有访问权限；一旦任务被挂起，主体拥有的访问权限就被冻结；如果任务恢复执行，主体将重新拥有访问权限；任务处于终止状态时，主体拥有的权限马上被撤销。TBAC 适用于工作流、分布式处理、多点访问控制的信息处理以及事务管理系统中的决策制定，但最显著的应用还是在安全工作流管理中。

1．任务

工作流是将各种纷繁复杂的商务过程按工程化的要求重新调整，以便进行计算机管理和控制。工作流包括活动(Activity)、控制流(Control flow)、主体(Subject)、数据项(Date item)和数据流(Data flow)这 5 个要素。而在这几个要素中，活动是最基本的组成部件。为了描述活动，需指定能执行该活动的主体，以及执行时所需要的数据项和(或)创建的数据项，而这些活动之间的逻辑顺序是由控制流给定的。TBAC中的任务，可以是工作流的活动，也可以是若干个活动的组成。

任务具有嵌套性。一个任务可由若干个子任务组成，子任务还可以继续分解，直到任务不能再分解为止。不能再分解的任务叫做原子任务。

正如操作系统中的进程一样，任务也有其生命周期。每个任务实例包括 5 个状态：静止态、活动态、挂起态、终止态和夭折态，如图 9.16 所示。任务实例被创建，就处于静止态；被激活，就处于活动态；执行时由于某种原因暂停执行，就处于挂起态；如果恢复执行，又重新处于活动态；如果任务顺利完成，就处于终止态；假设执行任务的条件不能得到满足，任务无法执行，中途夭折，那么就处于夭折态。

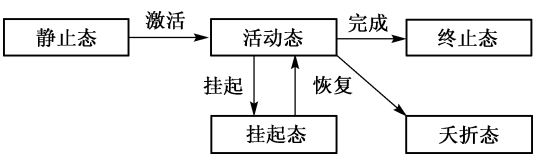


图 9.16　任务的生命周期

2．最小特权原则

在 TBAC 中，可以根据组织内的规章制度、职员的分工、任务等设计主体执行任务所需拥有的权限。但是主体真正得到权限的时间是在任务执行时，也即是说：主体由于执行一个任务，需要访问某客体，如果该任务当前状态不是活动态，主体所要求的权限将拒绝授予。

3．责任分离

责任分离是用来形成多人控制策略的安全原则，本质上要求两个或多个人负责完成某个处理。从理论上来讲，因为要求多人共谋，减少了欺诈犯罪的潜在危险。对于某些特定的操作集，某一个角色或用户不能单独完成所有这些操作。“责任分离”可以有静态和动态两种实现形式。静态责任分离的意思是：当一个任务与主体所拥有的其他任务彼此不互斥时，这个任务才能授权给该主体执行。动态责任分离是指当一个任务与主体的任何一个当前活动任务都不互斥时，该任务才能成为该主体的另一个活动任务。

9.5.3 基于属性的访问控制

依赖主体属性授权，是为陌生双方建立信任关系的一种有效方法，并基于此提出了基于属性的访问控制(ABAC)方法。在ABAC中，利用相关实体(如主体、资源、环境)的属性而不只是身份作为授权的基础。比如公司的门禁系统的访问控制策略可为：只有属于该部门的员工，并且时间在 6:00～24:00 之间的时候，员工才可以执行开门操作。这种基于属性的方法尤其适合于开放和分布式系统中的授权和访问控制。

1．基于属性的授权模型

属性是指与安全相关的某些特性。基于访问控制的目的，可将属性分为主体属性、资源属性和环境属性。主体是对资源采取操作的实体，如用户、应用、进程等；主体有定义其身份和特性的属性，包括主体的身份、角色、能力、年龄、邮政编码、IP 地址、雇员职位、已验证的 PKI 证书等。资源是被主体操作的实体，如数据、服务、系统设备等，资源属性包括资源的身份、位置(URL)、大小、值等，这些属性可从资源的“元数据”中获取；环境属性是与事务处理关联的属性，它通常与身份无关，但适用于授权决策，如时间、日期、系统状态、安全级别等。

ABAC 的基本观点是不直接在主体和客体之间定义授权，而是利用他们的属性作为授权决策的基础。基本的 ABAC 授权模型如图 9.17 所示。

在 ABAC 基本授权模型中，主体和客体均用一组属性和对应的属性值表示。许可(Permission)由客体描述器(Object Descriptor)和操作(Operation)组成，授权是在主体描述器(Subject Descriptor)与客体描述器之间定义的，主体描述器或客体描述器由关于主体或客体的属性条件组成，如“年龄 >30”等。

除了主体属性和客体属性外，在许多情况下，访问还需要受到一定环境和系统状态的约束。例如，只有在工作日或在特定地点才能访问某个数字资源。系统负荷很重时，只有高级用户才能得到它提供的服务。授权系统需要检查目前环境和系统的状态，这种决策因素被称为“环境上下文(环境属性)”。因此，此模型可进一步扩展。

2．基于属性的访问控制架构

基于属性的访问控制基本框架如图 9.18 所示。

其中各模块的主要功能如下：

属性机构(Attribute Authority，AA)：各个属性机构负责建立和管理各自的主体属性、

资源属性和环境属性。AA 本身可以存储也可以不存储属性，如可以从 LDAP 目录抽取属性。

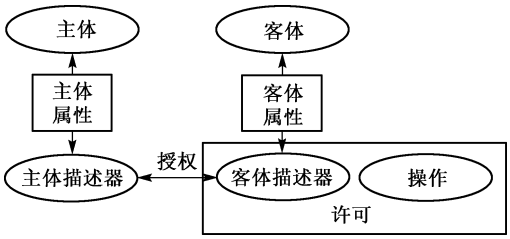


图 9.17　基本的 ABAC 授权模型

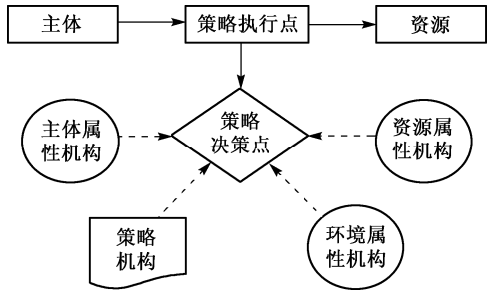


图 9.18　ABAC 访问控制框架

策略执行点(Policy Enforcement Point，PEP)：负责建立一个基于主体、资源、环境的属性的授权请求，并发送授权请求给 PDP；它还要执行 PDP 的决策，允许或拒绝对资源的访问请求。

策略决策点(Policy Decision Point，PDP)：负责利用策略规则集来判断主体的访问请求是否满足要求，以便决定是允许还是拒绝请求，并将决策结果返回给PEP。当属性没有出现在访问请求中时，它还要负责联系相关的 AA，以抽取需要的属性值。

策略机构(Policy Authority，PA)：负责建立和管理访问控制策略，供 PDP 使用。策略中主要包含访问资源所需的决策规则、条件和其他约束。

从上述 ABAC 授权模型及访问控制框架可以看出，ABAC 中所要解决的主要问题和研究内容有：属性的表示、基于属性授权时所采用的策略语言、敏感属性的保护、属性的分布式存储与发现、实际系统的实现等。ITU-T X.509 v4 中所提供的属性证书和原有公钥证书属性拓展字段可以作为上述方法的实现基础，比如属性证书中加入代表角色的属性字段等。

9.6　Windows 2000/XP 的访问控制机制

本节将在访问控制基础理论的基础上介绍实际操作系统中访问控制的实现机制。在 Windows 2000 中，引入了账户(User Accounts)这个概念，所有的用户模式代码在一个用户账户的上下文中运行。账户定义了 Windows 中一个用户所必要的信息，包括口令、安全 ID(SID)、组成员关系、登录限制等。同时，Windows 2000 中还引入了账户组，比如 universal groups, global groups, local groups 等账户组，每个账户组定制了不同的权限，账户组的任何账户自动具备该账户组的所有权限。安全 ID(SID)是系统自动生成的一个时间和空间唯一、全局唯一的 48 位数字。为了保证 SID 的唯一性，在生成它们的时候使用一个公式，结合计算机名、当前时间和当前用户模式线程使用 CPU 时间的总量。比如：S-1-5-21-1504001333-1204557612-4512547891-500，SID 带有前缀 S，它的各个部分之间用连字符隔开，第一个数字(本例中的1)是修订版本号，第二个数字是颁发机构代码(对 Windows 2000 来说，总是为 5)，然后是 4 个子颁发机构代码(本例中是 21 和后面三个 10 位长的数字串)和一个相对标识符(本例中是 500)。RID 对所有的计算机和域来说都是一个常数。例如，带有 RID 500 的 SID 总是代表本地计算机的真正的 Administrator 账户，RID 501 是 Guest 账户。

在 Windows 2000/XP 中，系统把所有的资源都称为对象，比如文件、目录、注册表键、内核对象、同步对象、私有对象(如打印机等)、管道、内存等。所有对对象的访问都要通过

安全子系统的检查，系统中的所有对象都被保护起来。对象的安全描述符(Security Descriptor，SD)包含了与一个安全对象有关的安全信息，它包含如下几个部分：

(1)客体所有者的安全标识符(SID)。

(2)自主访问控制表 DACL(Discretionary Access Control List)：是该对象的访问控制表，由对象的所有者控制。

(3)系统访问控制表SACL(System Access-Control List)：定义该对象上的哪些操作应被审计，由系统管理员控制。

(4)一组访问标记。

安全访问令牌(Security Access Token)是对一个进程或者线程的安全环境的完整描述，它包括以下主要信息：

(1)用户账户的 SID。

(2)所有包含该用户的安全组的 SID。

(3)特权：该用户和用户组可以调用的一组对安全性敏感的系统服务。

(4)默认的所有者，如果该进程创建了另一个对象，该域决定谁是新对象的所有者。

(5)默认的自主访问控制表(DACL)：用于保护用户创建的所有对象的初始表。

安全访问令牌是一个基本的安全单元，每个进程都具有一个。

给定客体，确定有哪些主体能够访问它，如何访问？给定主体，确定它能访问哪些客体，如何访问？是访问控制主要关注的两个问题。在Windows 2000操作系统中，用户对系统资源的访问权限，如对某文件的读、写控制，更关注前一问题，采用了ACL机制，如对象SD 中的DACL。用户对整个系统做的事情，如关掉系统、往系统中添加设备，更改系统时间，更关注哪些用户拥有这些特权，属于 CL 机制，如进程的安全访问令牌中定义的特权。

当用户登录成功后，系统为用户生成一个进程，本地安全权威为用户创建访问令牌，包括用户名、所在组、安全标识等信息。此后用户每新建一个进程，都将访问令牌复制作为该进程的访问令牌。当用户或者用户生成的进程要访问某个对象时，安全引用监控器将用户/进程的访问令牌中的SID与对象安全描述符中的自主访问控制表进行比较，从而决定用户是否有权访问对象。

9.7 Linux 系统的访问控制机制

Linux系统将设备和目录都看做文件，有很多与普通文件相同的操作，这为简洁的访问控制奠定了基础。

主体对文件具有三种权限：

- **读(r)**：用户可以读文件。
- **写(w)**：用户可以创建或修改文件。
- **执行(x)**：用户可以运行可执行程序。

系统将所有用户分为四类：

- **根用户(root)**：在系统里具有最大的权利，几乎可以控制整个系统，独有很多重要的能力。
- **所有者(owner)**：文件的所有者。一般可以读写执行文件，比其他用户具有更高的权

限。多数情况下是文件的创建者，但并不完全如此。文件所有者的id用 UID 表示。root 的 UID 是 0。

- **组(user group)**：所有者所在组，一个用户可以同时在多个组中。组可以按不同的需求设定，便于设置访问权限。文件所有者所在用户组的 id 用 GID 表示。
- **其他用户(other user)**：不属于前三类的用户。

Linux 文件系统安全模型与两个属性相关，一个是文件的所有者，一个是访问权限。每个文件和其创建者的 UID 和 GID 关联，一个进程通常被赋予其父进程的 UID 和 GID。

Linux 系统中每个文件有 10 位权限位，含义如下：

- **第 1 位**：d(目录)，b(块系统设备)，c(字符设备)，.(普通文件)。
- **第 2～4 位**：所有者的读、写、执行权限。
- **第 5～7 位**：所有者所在组的读、写、执行权限。
- **第 8～10 位**：其他用户的读、写、执行权限。

例如某目录文件的权限为(drwxrwxrwx)，表示所有用户可读可写可执行。

思考和练习题

(1) 什么是访问控制，访问控制包含哪几个要素？
(2) 什么是自主访问控制？什么是强制访问控制？各有什么优缺点？
(3) 自主访问控制的实现机制有哪几类，有什么区别？
(4) Bell-LaPadula 模型的基本思想是什么，举例说明。
(5) Biba 模型的基本思想是什么，举例说明。
(6) Windows 2000/XP 采用了怎样的访问控制机制？
(7) 基于角色的访问控制的基本安全原则有哪些？

实践/实验题

设计并实现一个简单的支持基于属性的访问控制模型的软件。比如文档资料的访问控制策略可为：只有属于该部门的员工，并且时间在 6:00～24:00 之间的时候，其 IP 地址是 192.168.*.*时，才可以执行读操作。

第 10 章　安全电子邮件

前面几章介绍了密码算法和 5 大类安全服务，在此基础上，本章探讨基于密码算法实现安全保护的实际应用实例：安全电子邮件。

E-mail是Internet上最大的应用之一，也是一个广泛的跨平台、跨体系结构的分布式应用。由于 E-mail 属于个人或机构在网络上的绝对隐私，其安全性受到高度的关注。本章所讲的安全的电子邮件主要是解决身份鉴别和保密性问题，并不能解决垃圾邮件、病毒在邮件中的传播、邮件服务器的入侵等问题，这些问题还需要其他的防御措施和机制。一些典型的安全电子邮件解决方案有：PGP, S/MIME, PEM（Privacy Enhanced Mail）, MOSS（MIME Object Security Services）。PEM 是由美国 RSA 实验室开发的安全电子邮件的早期标准，仅支持一些固定的算法：MD5, RSA 和 DES，但允许以后补充其他算法。PEM 只支持安全的文本信息，此外，PEM 指定了一个单一的证书层次结构，所有的 CA 都要信任同一个根 CA，这大大限制了 PEM 的发展。MOSS 针对 PEM 的不足做了一些改进，它改变了 PEM 只支持文本信息的局面，支持了 MIME。但 MOSS 有很多的执行选项，往往被认为是一种框架而不是一个规范。本章重点对两种应用广泛的安全电子邮件标准 PGP 和 S/MIME 进行介绍。

10.1　电子邮件原理

邮件服务器通常被划分为三个模块：邮件分发代理（Mail Delivery Agent，MDA）、邮件传送代理（Mail Transfer Agent，MTA）和邮件用户代理（Mail User Agent，MUA），这三个部分之间的界限并不十分明确。有时一个程序模块可能既包含了 MDA 功能同时又能实现 MTA 功能，有时又是 MTA 与 MUA 功能组合在一起。

MUA 程序是用来阅读和发送邮件的程序。MUA 并不接收邮件，只是显示用户邮箱中已经存在的邮件。很多 MUA 程序还允许用户创建不同的邮件夹来存储邮件。好的MUA向用户隐藏整个邮件系统的复杂性。一些常见的用户代理程序有：文本终端的 binmail，图形文本终端的 pine，X Windows 系统终端的 kmail 程序，Rand 公司的 Mail Handler（MH），Netscape Messenger 和 Microsoft Outlook Express 等。

MDA程序从MTA程序接收邮件，然后决定怎样分发这些邮件，要么发到本地用户的邮箱，要么发到由本地用户指定的某个地点。MDA的主要功能是在本地邮件服务器上将邮件分发给用户，主要功能包括：自动邮件过滤，使邮件信息能够被自动分捡到不同的邮件文件夹；自动邮件回复，对所收邮件进行自动回复；邮件自动触发，根据邮件信息的不同启动不同的程序。开放源码 MDA 程序有：同时是 MUA 的文本终端的 binmail、可进行灵活配置的 procmail。

MTA负责处理所有接收和发送的邮件。如果目的主机是远程的邮件服务器，本地MTA必须同这个远程的MTA建立通信链路来传递邮件。对MTA程序要求有：安全性，MTA 程序使用普通用户运行，确保黑客即使破坏了邮件软件，也不至于控制整个服务器；配置简单；处理迅速。UNIX 环境下三种使用最广泛的 MTA 是 Sendmail, Qmail, Postfix。

E-mail 首先是由 MUA 创建的存在用户磁盘上的一个文件。MUA 将其送到邮件分发器

MDA，邮件分发器再将其交给邮件传输代理 MTA，MTA 通过网络把邮件传送到最终的 MTA，MDA 从 MTA 接收邮件，然后将其放入接收者的邮箱(另一个磁盘文件)。互联网上电子邮件传递简图如图10.1所示。

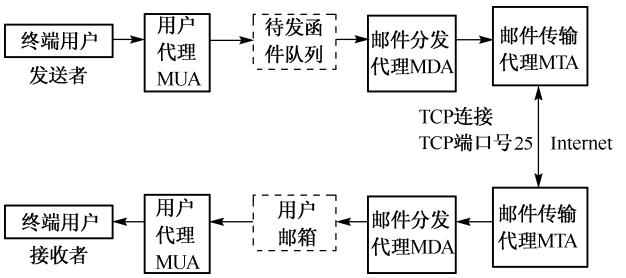


图 10.1　互联网上电子邮件传递简图

两个基本的电子邮件协议是邮件传输代理协议(MTA Protocol)和邮件用户代理协议(MUA Protocol)。邮件传输代理协议用于支持两个远程主机间传送邮件，在互联网上使用最广泛的邮件传送协议是简单邮件传送协议 SMTP。RFC 821 定义了 SMTP 协议，RFC 822 定义了在两个 MTA 之间采用 RFC821 标准传输邮件的格式，此外还有扩展的简单邮件传输协议 ESMTP。MUA 协议的目的是允许用户从他们的邮箱中读取邮件，最简单的是邮局协议(Post Office Protocol，POP3 见 RFC 1939，v3)和交互邮件访问协议 IMAP(Internet Message Access Protocol 参见 RFC 2060，v4rev1)。

UNIX 系统通常把 E-mail 从MTA传到本地 Client 文件系统的文件中，在客户机器上使用 elm, pine 和 xmail 等阅读 E-mail。因此，UNIX 系统的用户名和口令控制对客户邮箱的访问，于是E-mail系统的安全性部分地依赖于用户账号的安全性。也可以把E-mail存储在邮件服务器而不是在客户端的机器上，客户端与邮件服务器交互的两个公共协议是 POP 和 IMAP。也可以使用Browser来收发电子邮件，如直接使用Web服务器提供的Web Mail服务，如 mail.gmail.cn、mail.139.com 等。

10.2　PGP

PGP的英文全称是 Pretty Good Privacy，提供可用于电子邮件和文件存储的保密与鉴别服务。作者是 Phil Zimmermann，他从 20 世纪 80 年代中期开始编写 PGP，到 1991 年完成了第一个版本。在此之后 PGP 成为自由软件，经过许多人的修改和完善，PGP 逐渐成熟，已成为 Internet 标准文档(RFC2044 和 RFC3156)。PGP 得到广泛应用，其成功的主要原因在于：

(1) 提供免费版本，且可用于多种平台，如 Windows, UNIX, Linux, Mac OSX，商业版本使用户得到很好的技术支持。

(2) 选用生命力和安全性已经得到公众认可的算法。目前最新版本的 PGP9.12.0 采用的公钥算法有 RSA 算法(签名和加密密钥均支持 1024～4096 位)，DSS 算法(1024～3072 位)，Diffie-Hellman 密钥交换算法(1024～4096 位)；对称密码算法有 CAST-128、IDEA(128 位)、3DES(168 位)、AES-128, AES-192, AES-256；散列算法有 SHA-2-256, SHA-2-384, SHA-2-512, SHA-1(160 位), RIPEMD-160, MD5(128 位)。PGP 软件在 5.0 版本以前使用 MD5，从 PGP 5.0 开始向 SHA-1 发展，保留 MD5 的目的之一是出于兼容性的考虑。

(3) 具有广泛的可用性，适用于公司和个人。

(4) 不由政府或标准化组织控制。

10.2.1 使用 PGP 保护电子通信

PGP 使用加密以及校验的方式，提供了多种功能和工具，保护电子邮件、文件、磁盘以及网络通信的安全。这里我们重点介绍PGP提供的安全电子邮件功能。PGP采用公开密钥与对称密钥加密相结合的方式提供电子邮件的安全性。PGP 采用的对称加密技术，所使用的密钥称为“会话密钥”，每次加密时，PGP都会随机产生会话密钥，用来直接加密报文，公开密钥加密技术中的公钥和私钥，则用来加解密会话密钥，并通过它间接地保护报文的内容。PGP把公钥和私钥存放密钥环文件中，允许任意多个密钥环，但用户至少有以下两个默认的密钥环文件，这就是：

secring.skr：私钥环文件，存放用户的私钥。

pubring.pkr：公钥环文件，存放用户和其他人的公钥。

PGP 在多处需要用到口令，它主要起到保护私钥的作用，私钥是百位以上且无规律的数字，用户直接记忆十分困难，PGP 把私钥加密后存入私钥环，用户使用口令提取私钥。PGP用于保护邮件时主要在两处需要用户输入口令：需要解开收到的加密邮件时；需要为文件或邮件签名时。此外，PGP 还可以对磁盘上的文件进行对称加密，这时也需要用户输入一个口令，每个密文文件可以有一个单独的口令。PGP 并不在磁盘上存储用户口令，而是用口令加解密，如果口令输入错误，解密必然失败，PGP 的口令允许任意长度，这也为普通用户选择口令带来了很大的方便。

PGP 的安全电子邮件的实际操作由 5 种服务组成：数字签名、消息加密、数据压缩、邮件兼容性和分段。

1．数字签名

假设所选的公钥算法为 RSA-2048，散列算法为 SHA-256，对称加密算法为 AES-128，PGP进行数字签名的内部过程如图10.2所示。签名时，用户输入口令，口令经过散列函数作用后，用其中的 128 位作为密钥，解密从私钥环取出的用AES-128加密的RSA私钥，恢复出明文的RSA 私钥。另一方面，PGP 使用散列算法 SHA-256 生成发送报文 M 的消息摘要 H，再用发送者的 RSA 私钥对 H 签名，并与 M 连接。最后对整个报文压缩后发送。

接收方对接收到的文件解压缩，然后从公钥环取出发送者的公钥解密并恢复散列码H，对报文 M 生成一个新的散列码，与 H 比较。如果一致，则报文 M 被接收者确认。该过程的原理如图10.3所示。

数字签名中 RSA 算法的安全强度使接收方相信只有拥有私钥的人才可以生成正确的签名，确保了发送方的身份，SHA-256 的强度使接收方确认其他人不可能伪造与该散列值匹配的消息，保证了签名的有效性。此外，签名也可以采用 DSS/SHA-256 的替代方案。

2．消息加密

图10.4为PGP发送加密邮件的内部过程的示意图。PGP随机数发生器产生仅供一次性使用的128位会话密钥，使用该会话密钥对压缩后的报文进行加密，生成密文文件。另一方面，PGP 根据用户输入的收信人标识信息，从公钥环找出收信人的公钥，使用接收方的公钥对会话密钥再做加密。最后，PGP 把加密后的会话密钥和密文合并在一起，形成一个新的文件，发送给对方。

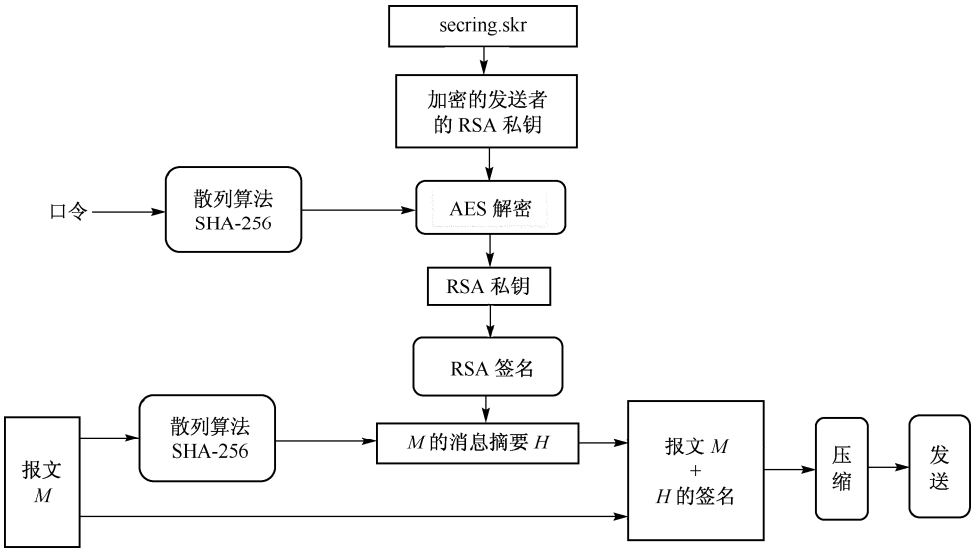


图 10.2　PGP 的数字签名过程

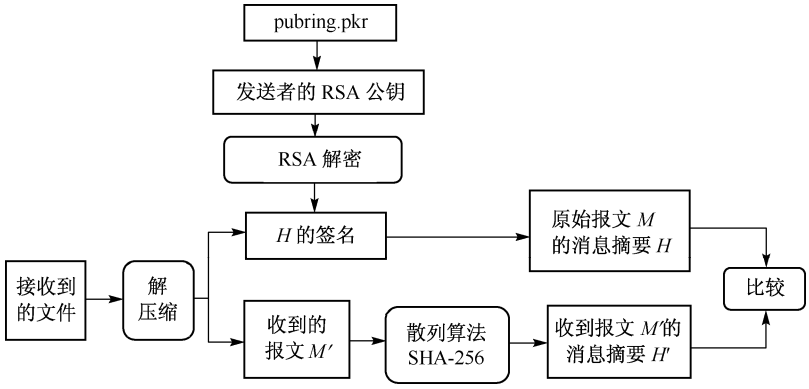


图 10.3　PGP 数字签名的验证过程

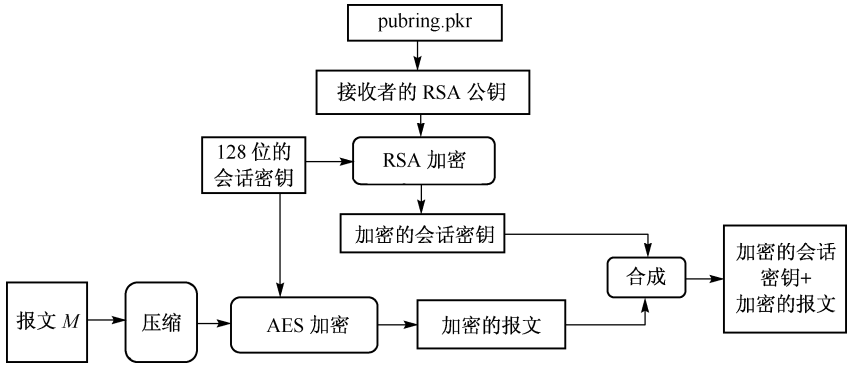


图 10.4　PGP 发送加密邮件的过程

PGP 接收密文邮件后的解密过程原理如图10.5所示。PGP 把接收到的密文分成两部分，一部分是经 RSA 算法加密的会话密钥，另一部分是经 AES 算法加密的密文，接着，PGP 要

求用户输入口令，用户输入的口令经过散列函数作用后得到128位的AES解密密钥，此密钥用于解密从密钥环中取出的经加密的RSA私钥，得到明文的RSA私钥用于解密加密的会话密钥，然后再用明文的会话密钥解密密文，最后还需要进行解压缩的操作。

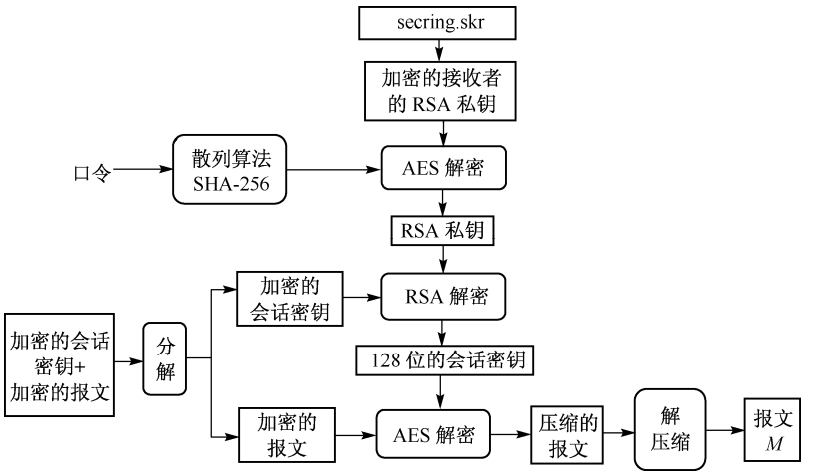


图 10.5 PGP 接收加密邮件的过程

PGP 的加密采用了对称加密算法和公钥加密算法的结合，可以缩短加密时间，用公钥算法解决了会话密钥的分配问题，不需要专门的会话密钥交换协议，由于邮件系统的存储-转发的特性，用握手方式交换密钥也不太可能。PGP 每次加密都使用一个随机的会话密钥，即使同一文件两次加密发送的密文也是不同的，进一步增强了保密强度。此外，PGP 提供了 Diffie-Hellman 算法的一种变体可替换 RSA 算法进行加密。

当用户希望把加密的邮件发送给多个收信人时，当然可以把同一信件内容依次发送，但这过于烦琐。PGP 系统允许用户一次性地为多个收信人加密发送同一通信内容，产生的加密文件可以被其中任何一个人解开。这时，PGP 不是用不同的会话密钥生成不同的密文，这样会使得密文加倍，而是从公钥环中取出所有接收人的公钥，用它们对会话密钥分别进行加密，得到多个加密的会话密钥。每个收件人在收到加密邮件后，用自己的私钥解开用自己的公钥加密的会话密钥，进而解开通信密文。

两种服务都需要时，发送者先用自己的私钥签名，然后用会话密钥加密，再用接收者的公钥加密会话密钥。

3．数据压缩

PGP 把压缩的位置发生在签名之后、加密之前。在压缩之前生成签名，这一方面可以使验证签名时无须压缩，另一方面也因为采用的是动态的压缩算法，它会根据运行速度产生不同的压缩比，从而导致不同的压缩格式。在加密前压缩的原因是压缩的报文信息冗余度小，增加密码分析的难度。此外，压缩有利于节省邮件存储的空间，节约邮件传输的时间。

最新版的 PGP 软件支持的压缩算法有：Bzip2, ZLIB 和 Zip。

4．邮件兼容

PGP 对部分或者全部报文进行了加密处理，加密处理后的字节流，为任意的 8 位字节流。然而，某些邮件系统只允许ASCII码字符组成的块，所以 PGP 提供了将 8 位二进制流转换为可打印的ASCII码字符的功能。为此，PGP采用了Radix-64转换方案。该方案将导致消息大小增加 33%。实际上压缩可以补偿 Radix-64 转换导致的膨胀。

5．数据分段

电子邮件工具常常限制最大消息长度，一般限制在最大50 000字节，更长的消息要进行分段，每一段分别邮寄。为了适应这个限制，PGP 自动将长消息分段并在接收时自动恢复。会话密钥和签名只需出现在第一段的段首。

10.2.2　PGP 的密钥和密钥管理

PGP 使用四种类型的密钥：对称加密的一次性会话密钥、公钥、私钥和基于口令短语的对称密钥。

1．对称加密的一次性会话密钥

PGP 的会话密钥是个伪随机数，它是基于 ANSI X.917 的算法由随机数生成器产生的。以 CAST-128 为例，ANSI X.917 中的 3DES 被替换成了 CAST-128。输入包括一个 128 位的以前产生的会话密钥和两个 64 位的数据块作为加密的输入，使用 CFB 方式，CAST-128 产生两个 64 位的加密数据块，这两个数据块的结合构成 128 位的会话密钥。作为明文输入的两个 64 位数据块，是从一个 128 位的随机数流中导出的，这些数据是用户击键产生的，击键的时间和内容用来产生随机流。

2．密钥标识符

当一个用户有多个公钥/私钥对时，接收者如何知道发送者是用了哪个公钥来加密会话密钥？一种简单的方案是将公钥与消息一起传送。但一个RSA的公钥可以长达几百个十进制数，造成了很大的空间浪费，为了解决这个问题，可以将一个标识符与一个公钥关联，对一个用户来说做到一一对应就可以，这样就只需要传递较短的密钥标识。定义公钥标识 KeyID 包括 64 个有效位：$KU_a \bmod 2^{64}$，即公开密钥的低 64 位。

至此，可以给出PGP 传递消息的格式，消息由三个部分组成：报文部分、签名(可选)和会话密钥部分(可选)，如图 10.6 所示。报文部分包括实际存储或传输的实际数据、文件名以及说明创建时间的时间戳。

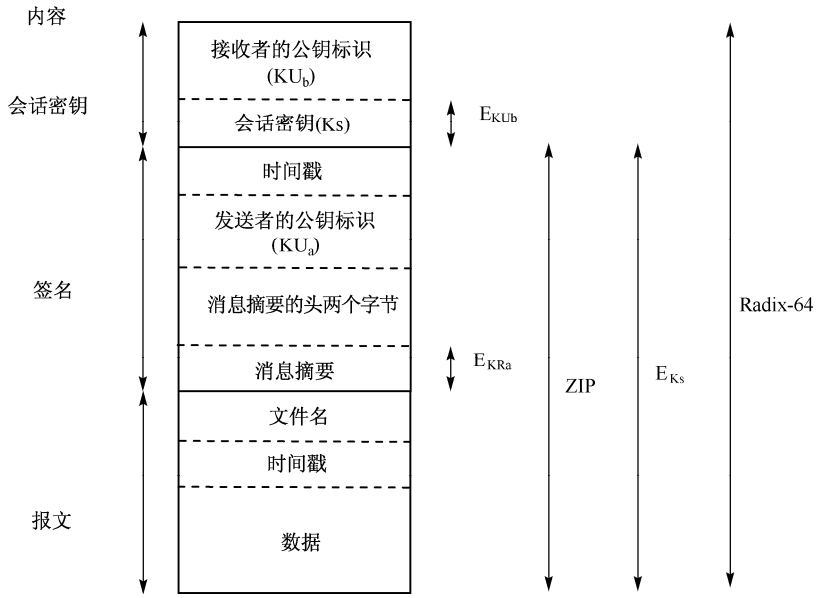


图 10.6　PGP 报文(A 发送给 B)的一般格式

签名部分包括签名的消息摘要，由计算签名的时间戳和数据部分得到；消息摘要的前两个字节，可以使接收方判断是否使用了正确的公钥验证消息摘要；发送者公开密钥的密钥 ID 和签名构造的时间戳，签名时间戳用于防止重放攻击。

会话密钥部分包括加密的会话密钥和接收者公开密钥的密钥 ID。

整个消息使用 Radix-64 转换编码。

3．密钥环

PGP 为了管理密钥ID 在每个用户创建了一对数据结构，一个用来存储该用户所拥有的公钥/私钥对，称为私钥环，一个用来存储该用户所知道的其他用户的公钥，称为公钥环，如图10.7所示。

私钥环

时 间 戳	公 钥 标 识	公 钥	加密的私钥	用 户 标 识
•	•	•	•	•
T_i	$KU_i \bmod 2^{64}$	KU_i	$E_{H(P_i)}[KR_i]$	用户 i
•	•	•	•	•
•	•	•	•	•

公钥环

时 间 戳	公 钥 标 识	公 钥	所有者信任度	用 户 标 识	密钥合法性	签 名	签名信任度
•	•	•	•	•	•	•	•
T_i	$KU_i \bmod 2^{64}$	KU_i	$trust\_flag_i$	用户 i			
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•

图 10.7　PGP 的密钥环结构

(a) 私钥环

私钥环中所包含的数据项有时间戳、公钥标识、公开密钥、私有密钥以及用户标识。私有密钥可以通过用户标识或公钥标识进行查找。此外，为了保证私有密钥的安全性，PGP 采用如下方式保存私有密钥：

(i) 用户选择一个口令用于加密私钥。

(ii) 当系统用公钥算法如 RSA 生成一个新的公钥/私钥对时，要求用户输入口令，对该口令使用散列函数生成散列码后，销毁该口令。PGP可以接受任意长度的口令，为用户选择高质量的口令创造了条件。

(iii) 系统用散列码的一些位作为密钥，用某种对称加密算法如CAST-128 加密私钥，然后销毁这个散列码，并将加密后的私钥存储到私钥环中。

当用户要访问私钥环中的私钥时，必须提供口令，PGP 将取出加密后的私钥，将口令生成散列码，解密私钥。

(b) 公钥环

公钥环中所包含的主要数据项有时间戳、公钥标识、公开密钥和用户标识。公钥环可以通过用户标识或公钥标识进行查找。

4．公钥的管理

由于 PGP 重在广泛地应用于正式或非正式环境，没有建立严格的公钥管理模式。如果用户 A 的公钥环上有一个从 BBS 上获得用户 B 发布的公钥，但已被用户 C 替换，这时就存在两种风险。C 可以向 A 发信并冒充 B 的签名，A 以为是来自 B；A 与 B 的任何加密消息 C 都可以读取。为了防止 A 的公钥环上包含错误的 B 的公钥，有若干种方法可用于降低这种风险。

(a) 物理上得到 B 的公钥。如直接面对面复制，这种方式可靠，但有局限性。

(b) 通过电话验证公钥。B 将其公钥 E-mail 给A，A 可以用PGP对该公钥生成一个消息摘要，并以十六进制显示，称为密钥的“指纹”。然后 A 打电话给 B，让 B 在电话中对证“指纹”。如果双方一致，则该公钥被认可。

(c) 数字证书。一种办法是从双方都信任的个体 D 处获得 B 的公钥，D 是介绍人，生成一个签名的证书，其中包含 B 的公钥，密钥生成时间。另一种办法是从一个信任的 CA 中心得到 B 的公钥。

PGP支持两种形式的证书，PGP证书和X.509证书。PGP证书包括以下信息：PGP的版本号、证书持有者的公钥、证书持有者的信息(如名字、用户标识、E-mail 地址、照片等)、证书持有者的自签名、证书的有效期、倾向的对称加密算法。与X.509证书不同的是PGP证书可以有多个签名。

在 PGP 中使用元介绍人(Meta-introducer)和可信介绍人(trusted introducers)与 X.509 环境中的 Root CA 和 CA 相对应。因为不可能由一个 CA 来证明所有的证书，所以存在三种信任模型，直接信任(Direct Trust)、分层信任(Hierarchical Trust)和 Web 信任。在直接信任模型中，用户之间直接验证密钥，不通过可信介绍人。分层信任与 X.509 中的层次信任模型相对应。Web 信任基于从旁观者角度和信息越多越好的思想，是一种累计的信任模型，它可以是直接信任，可以是某种形式的信任链，也可以通过多个介绍人建立信任关系。

尽管 PGP 没有包含任何建立认证权威机构或建立信任体系的规范，但它提供一种利用信任关系的方法，将信任关系与公钥联系起来。每个公钥在公钥环中有三个相关的属性域：

- **密钥合法性字段(Key legitimacy field)**：表明公钥的合法性或者有效性，PGP 对“此用户公钥是合法的”的信任程度；信任级别越高，这个用户 ID 与该公钥的绑定越强。这个字段是由 PGP 计算的。
- **签名信任度字段(Signature trust field)**：每一个公钥都有一个或者多个签名，这是公钥环主人收集到的、能够认证该公钥的签名。每一个签名与一个Signature trust field 关联，表明这个 PGP 用户对“签名人对公钥签名”的信任程度。Key legitimacy field 是由多个Signature trust field 导出的。
- **所有者信任度字段(Owner trust field)**：表明该公钥被用于签名其他公钥证书时的信任程度。这个信任程度是由用户给出的。最新版本的PGP9.12.0给出如下的信任级别：绝对(Implicit)、可信的(Trusted)、部分的(Marginal)、完全不(None)信任。

假设正在处理用户 A 的公钥环，则信任处理的过程可以描述如下：

(a) 当 A 向公钥环中插入一个新公钥时，PGP 必须为该公钥的所有者设定一个信任值，

也即为所有者信任度字段trust_flag赋值。如果 A 的公钥也出现在私钥环中，该信任域自动设为绝对(Implicit)信任。否则，PGP询问用户，让用户给出信任级别。用户可选：可信的(Trusted)、部分的(Marginal)或者完全不(None)信任。

(b) 当新公钥进入时，可能有一个或多个签名跟随其后。许多签名可以以后再加入。当一个签名插入到一个条目中时，PGP 查找该公钥环，看签名者是否是已有公钥的所有者。如果是，则这个公钥所有者的 Owner Trust 值被赋予该签名的 Signature trust 字段。否则，赋予 None 值。

(c) 密钥合法性字段的值根据该条目中多个签名合法性字段来计算。如果至少有一个签名的签名信任度值为Implicit，则该值设为完全有效。否则，PGP 计算一个信任值的权重和。

值得指出的是一个密钥如果被认为是可信任的，并不就意味着由它所签名的其他密钥同样可信任。

最后，由于密钥泄露或定时更新的需要，公开密钥的注销是必要的。通常的注销方法是由私钥所有者签发一个密钥注销证书，私钥所有者应尽可能快和尽可能广地散布这个证书，以使得潜在的有关人员更新他们的公钥环。

10.2.3 PGP 的其他功能

PGP 除了提供安全的电子邮件功能之外，逐渐还增添了一些其他功能，列举如下所述。

1．用 PGP 保护文件

PGP ZIP 提供对文件的加密压缩保护功能。在使用上用 PGP 加密文件没有任何特殊之处，用户首先选择一个加密文件的口令，然后对明文文件进行加密得到密文文件。是否同时进行压缩是 PGP ZIP 对文件加密时的一个可选项。文件加密也可以按如下几种方式进行：使用接收者的公钥加密、使用口令加密、只对文件签名不加密以及把文件加密成不依赖 PGP 的自解密文件。需要说明的是 PGP 在接收到用户输入的加密口令之后，并不是把它直接用于加密文件，而是让口令先通过一个散列函数的作用后，用得到的散列值作为对称加密的密钥，对文件进行加密。加密完成后，PGP 不会在任何地方存储用户的口令和中间过程所产生的密钥。

2．Shredder

PGP粉碎工具(PGP Shredder)可以永久地删除那些敏感的文件和文件夹，而不会遗留任何的数据片段在硬盘上。PGP 自由空间粉碎器(PGP Shred Free Space Assistant)可以再次清除已经被删除的文件实际占用的硬盘空间。PGP通过使用随机数多次重写的方法来达到以上目的，你可以设置重写的次数。PGP 建议个人使用的话，可以设置 3 次重写，商业上使用的话，设置 10 次重写，军事上使用的话设置 18 次重写，为了达到极大的安全性，你还可以设置 26 次重写。

3．PGP 全盘加密

PGP 全盘加密(PGP Whole Disk Encryption)，使用对称加密算法加密硬盘的每个扇区，所有的文件，包括操作系统文件、应用文件、数据文件、交换文件、自由空间、临时文件都被加密。在系统启动之后，PGP WDE提示你输入正确的口令，当你访问加密数据的时候，它就被解密了，在你往硬盘上写数据之前，PGP WDE 要对其进行加密处理。

此外，PGP 还提供即时消息安全保护、网络共享、PGP 虚拟磁盘(PGP Virtual Disk)保存和保护等功能。

10.3　S/MIME

S/MIME 是对 MIME 电子邮件格式的安全扩展，在 MIME 协议的基础上，S/MIME 增加了以下两类安全特性：数字签名和加密。S/MIME 使用 X.509 证书以及 PKCS 标准建立信任模型，由客户端的实现和用户来决定。S/MIME 更像商用或组织使用的工业标准，PGP 更面向个体用户选用，同样已成为Internet标准。RFC 为 S/MIME 定义了许多文档，重要的有：RFC3850, RFC3851, RFC3369 和 RFC3370。

10.3.1　RFC822

RFC822 定义了一种十分简单的邮件格式，这种格式的邮件只能包含纯文本信息，而且只能是ASCII码字符。RFC822将邮件消息视为信封和内容两部分，其中信封部分是为完成传输和投递所需的信息，内容部分是发往收方的具体消息内容。该标准仅适用于邮件消息的内容部分。根据此标准，内容部分有很多报头字段，这些字段又可能被用于产生信封。一个消息包括若干行的报头和无限制的正文。报头在前，正文在后，中间用空行分开。报头通常是由关键字、冒号和关键字取值构成，最常使用的关键字有 From, To, Subject, Date，分别表示邮件的发方、收方、主题、日期。例如：

```
From: "Wang Zhao" <wangzhao@infosec.pku.edu.cn>
To: "Wang Zhao" <wangzhao@infosec.pku.edu.cn>
Subject: hello
Date: Wed, 22 Apr 2009 18:03:38 +0800
```

10.3.2　MIME

MIME 协议克服了 RFC822 存在的问题和限制。定义了 5 个新的报头域，分别是：MIME-Version（MIME 版本），Content-Type（内容类型），Content-Transfer-Encoding（内容转换编码），Content-ID（内容标识）和 Content-Description（内容描述）。MIME-Version 这个参数的值必须为 1.0，表示报文符合 RFC2045 和 2046 的要求。Content-ID 在多个上下文中，用来唯一标识 MIME 实体。Content-Transfer-Encoding 定义了两种数据编码方式，quoted-printable 转换编码和 Base-64 转换编码。Content-Description 是对正文对象的文本描述，当对象是不可读时（如音频）这会很有用。其中只有 Content-Type 字段是必需的，该字段定义了 7 种内容基本类型，分别是 text, message, image, video, audio, application, multipart。这 7 种基本类型又分为 15 种子类型。

文本（text）类型主要的子类型是纯文本（plain text），就是简单的由 ASCII 码字符或 ISO 8859 字符组成的字符串。Enriched（丰富的）子类型允许更大的格式灵活性。

多部分（Multipart）类型指示报文正文包含了多个独立的部分。Content-type 字段包含了称为边界符（Boundary）的参数，定义了在报文的不同部分之间的分隔符。在报文正文的每个部分，可以存在可选的普通 MIME 报头。

Message type 提供了许多 MIME 的重要特性。RFC822 子类型指示报文正文是完整的报文，包括报头和正文。尽管使用了这种子类型名，但封装的报文可以不仅仅是 RFC822 报文，而是任何 MIME 报文。

Application type 指的是其他种类的数据，典型的是邮件应用程序不理解的二进制数据或信息。

10.3.3 S/MIME

具体来说，S/MIME 提供了以下的功能：

- **数据封装(Enveloped data)**：这一功能允许用对称密码加密一个 MIME 消息中的任何内容类型，然后用一个或多个接收者的公钥加密对称密钥。接着将加密的数据、加密的对称密钥以及接收者的公钥标识符等封装在一起。
- **数据签名(Signed data)**：发送者对选定的内容计算消息摘要，然后用签名者的私钥加密，再对内容和签名进行 Base-64 编码。此时邮件内容只能被具有 S/MIME 功能的接收方处理。
- **数据透明签名(Clear-signed data)**：只有数字签名部分使用 Base-64 进行编码。因此没有 S/MIME 功能的接收者可以看到报文的内容，但是不能验证签名。
- **数据的签名和封装(Signed and enveloped data)**：这一功能允许签名已加密的数据或者加密已签名的数据。

S/MIME 使用的密码算法有：消息摘要算法、签名算法、加密会话密钥的公钥加密算法和加密消息的对称加密算法。建议使用的消息摘要算法是SHA，但也支持MD5，签名的首选算法是 DSS，加密会话密钥的首选算法是 ElGamal。此外，RSA 算法也能用于签名和加密会话密钥。加密消息的建议算法是 3DES。

S/MIME 在 MIME 消息格式的基础上增加了签名、加密等安全功能，使用了两个新的内容类型 multipart 和 application，application 的所有子类型都使用 PKCS（Public Key Cryptography Specifications）来标记。数据加密和签名的过程与 PGP 类似。

S/MIME 使用第 3 版X.509 公钥证书，其公钥管理采用X.509 证书层次结构和 PGP 的 Web 信任的一种混合方式。和 PGP 一样，S/MIME 中管理者和（或）用户都必须配置每个客户端的信任密钥列表和证书撤销列表。也就是说对签名的验证和对消息的签名都是通过本地维护证书实现的，但证书由 CA 签发。VeriSign 是最为广泛使用的、支持S/MIME 的证书颁发机构，它颁发的 X.509 证书称为 VeriSign Digital ID，其中包括证书拥有者（即用户）的公钥、用户的名称或别名、证书的有效期、序列号、颁发机构的名称、颁发机构的签名，此外还可以包括用户提供的信息，如地址、电子邮件地址、基本注册信息（国籍、邮编、年龄、性别等）。VeriSign 按安全性要求，将证书分为三类，用户可在线向 VeriSign 的 Web 站点申请。

思考和练习题

（1）PGP 采用了哪些服务来保证电子邮件的安全性？
（2）PGP 为什么在压缩前生成签名？
（3）PGP 有几种密钥类型？
（4）S/MIME 和 PGP 的密钥管理有什么差别？
（5）你认为 PGP 可能的安全漏洞有哪些？

实践/实验题

(1) 任意使用一种 PGP 工具，熟悉其使用后，进行密钥对的生成，使用 PGP 发送加密和签名的邮件，练习公钥和私钥的导出和保存。

(2) 理解PGP的信任模型，要求按选课名单分成若干组，各组同学分别产生自己的公私钥对，将公钥自签名后，发给组内其他同学，每个同学收到别的同学的公钥之后，在确认后(如何确认，说明确认办法)，可以签名。也可以不对公钥确认。若签名，对公钥的拥有者的签名分别给予不同的信任级别。然后把签了名的公钥证书随意转发给其他同学。通过这种相互介绍的方式建立自己的公钥环。

第 11 章　网络安全协议

11.1　TCP/IP 基础

11.1.1　TCP/IP 的历史

1957 年美国国防部成立了高级研究计划署(Advanced Research Project Agency，ARPA)，1969 年，ARPA 更名为 DARPA，同年，DARPA 启动了一项计算机互联计划 ARPAnet，1973 年，DARPA 的 Bob Kahn 和 Stanford 大学的 Vint Cerf 领衔，组织研制一种能使各种计算机网络互相通信的技术，研究成果就是TCP/IP协议规范，1980年前后，TCP/IP开始应用在 ARPAnet 上。从 1982 年起，加州 Berkeley 大学推出了内含 TCP/IP 的 UNIX BSD，在一定程度上推动了TCP/IP 的研发。1983年，ARPAnet分裂为两部分：ARPAnet和纯军事用的MILNET，TCP/IP 成为 ARPAnet 的核心通信协议。1985 年，美国国家科学基金会(National Science Foundation，NSF)采用 TCP/IP 建设 NSFnet，NSFnet 于 1990 年 6 月彻底取代了 ARPAnet 而成为 Internet 的主干网。到 20 世纪 90 年代，TCP/IP 已发展成为因特网的主要协议。

11.1.2　TCP/IP 层次模型

TCP/IP 代表传输控制协议(TCP)和网际协议(IP)，除了这两个重要协议外，还有许多相关的协议和工具，它们组合在一起共同构成了 TCP/IP 协议集(协议栈)，如图11.1所示。

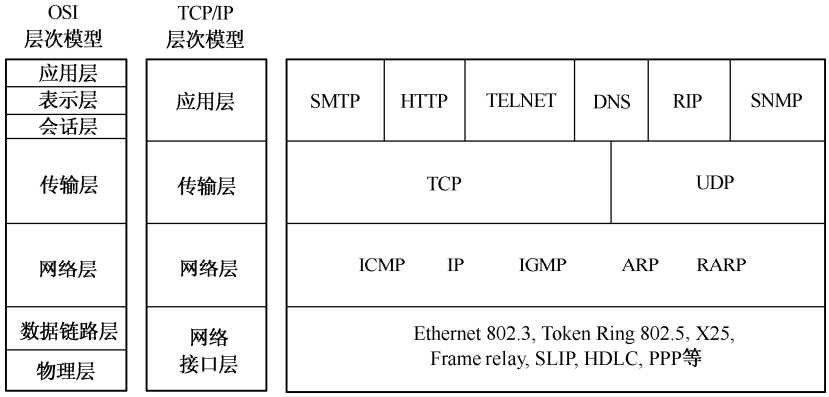


图 11.1　TCP/IP 协议栈

TCP/IP 层次模型中的网络接口层(Network Interface Layer)相当于 OSI 模型中的物理层(Physical Layer)加上数据链路层(Data Link Layer)，物理层规定了有关物理设备通过传输介质进行互联的描述和规定，为信息流提供物理传输通道，在物理层比特流被转换成传输介质易于传输的电、光等信号。数据链路层确保在物理层上可以建立一个进行数据通信的数据链路，负责完成封装帧以及将数据帧无差错地传输。每一种物理硬件都存在自己特有的通信协议，支持特有的数据链路方式，如以太网上支持的以太协议等。TCP/IP 层次模型中的网络接口层综合了这两个层次的功能，负责从上层接收IP 数据报，并把数据报处理成数据帧发送出去，

或者从网络上接收数据帧，抽出 IP 数据报，并把数据报交给 IP 层。数据链路层的协议主要有 Ethernet 802.3, Token Ring 802.5, X.25, SLIP, HDLC, PPP 和帧中继等。

TCP/IP 层次模型中的网络层(Internet layer)相当于OSI模型中的网络层(Network Layer)，网络层的主要功能是路由。网络层的协议主要包括：IP(Internet Protocol)协议、Internet 控制报文协议(Internet Control Message Protocol，ICMP)、地址解析协议(Address Resolution Protocol，ARP)、反向地址解析协议 RARP(Reverse ARP)等。IP 层的服务是无连接的、不可靠的，服务的可靠性交给了上层的 TCP 协议来保证。

TCP/IP 层次模型中的传输层(Transport Layer)对应于 OSI 模型中的传输层，它包含一个面向连接的、可靠的传输控制协议(Transmission Control Protocol，TCP)，一个面向无连接的不可靠的用户数据报协议(User Datagram Protocol，UDP)，基于 UDP 协议的应用的可靠性必须由应用层来提供。

TCP/IP层次模型中传输层的上面是应用层(Application Layer)。应用层为用户提供调用和访问网络上的各种应用程序的接口，并向用户提供各种标准的应用程序及相应的协议，是网络与用户应用软件之间的接口。该层协议可分为三类。第一类是面向连接的，如远程登录 Telnet、文件传输协议(File Transmission Protocol，FTP)、SMTP、超文本传输协议(Hyper Text Transfer Protocol)等；第二类是面向无连接的，如简单网络管理协议 SNMP、路由信息协议(Routing Information Protocol，RIP)等；第三类是既面向连接又面向无连接的，如域名解析协议 DNS 等。OSI 模型的会话层、表示层和应用层都和 TCP/IP 的应用层对应。会话层(Session Layer)允许不同机器上的用户建立会话关系，会话层的服务之一是管理对话，令牌管理(Token Management)和同步(Synchronization)都是与会话有关的服务。表示层以下的各层只关心可靠地传输比特流，而表示层关心的是所传输信息的语法和语义。如果通信双方用不同的数据表示方法，它们就不能相互理解，表示层可以屏蔽这种不同。

在TCP/IP 的每一层中，数据包分为报头和有效载荷，报头是与该层相关的控制信息，有效载荷是从上一层传下来的数据。发送时，每一层把上一层的数据包作为有效载荷，加上本层的报头信息，交给下一次层处理，图11.2 给出了以太网环境下用户数据经过TCP/IP 协议栈的封装过程。

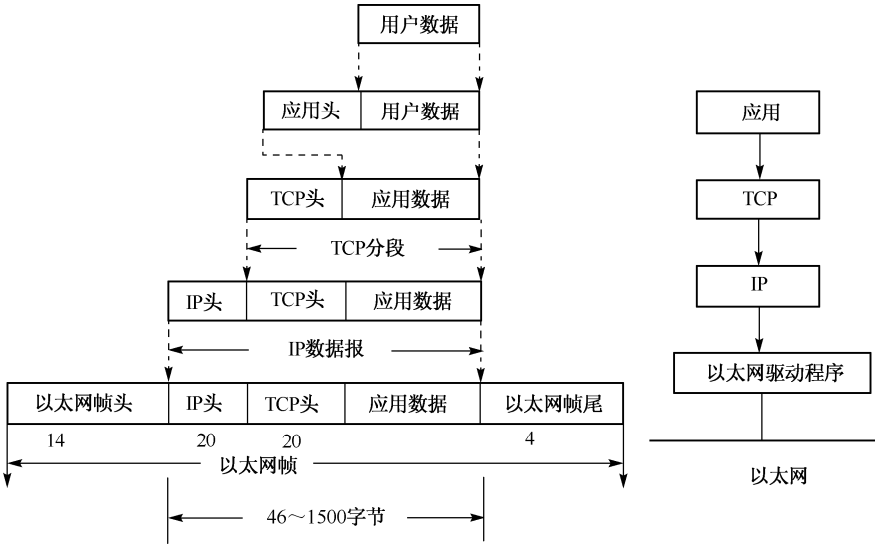


图 11.2　用户数据经过 TCP/IP 协议栈的封装过程

传输层将应用层的数据流截成分组，并加上TCP报头形成TCP分段，送交网络层。在网络层给TCP分段加上IP报头，生成一个IP数据报，并将IP数据报送交数据链路层。数据链路层给IP数据报加上以太网的帧头和帧尾，生成以太网帧，将数据帧发往目的主机或IP路由器。在目的主机，执行相反的过程，逐层去掉报头信息，得到用户数据。

11.1.3 IPv4 协议

IP协议是TCP/IP协议族中至关重要的组成部分，但它提供的是一种不可靠、无连接的数据报传输服务。不可靠(Unreliable)指的是不能保证一个IP数据报成功地到达其目的地。传输错误的处理办法是扔掉该数据报，向其发送者传送一个ICMP消息。无连接(Connectionless)表示IP并不维护关于连续发送的数据报的任何状态信息。每个数据报单独处理，在传送过程中可能出现错序。

11.1.3.1 IPv4 编址技术

在互联网中，通过IP层软件提供一种通用的地址格式，在统一管理下进行分配，确保一个地址对应一台主机。通称IP层所用的地址为互联网地址或IP地址。在Windows平台下可以使用命令ipconfig来查看本机的IP地址。IPv4规定地址总长32比特，分为5类，如图11.3所示。

A类地址中，第一个8位字节表示网络号，其余三个8位字节用来标识主机号，A类地址的第一个字节的数值范围是0～127。B类地址中，前两个8位字节表示网络号，其余两个8位字节用来标识主机号，B类地址的第一个字节的数值范围是128～191。C类地址使用前三个8位字节表示网络号，只有一个8位字节用来标识主机号，C类地址的第一个字节的数值范围是192～223。A类、B类和C类被称为基本类，用于主机地址，D类地址用于组播，允许发送到一组计算机，这时，一组主机必须共享一个组播地址，任意发送到组播地址的数据包副本都会发送到这组主机的每一台。D类地址的第一个字节的数值范围是224～239。E类地址属于保留地址。E类地址的第一个字节的数值范围是240～255。

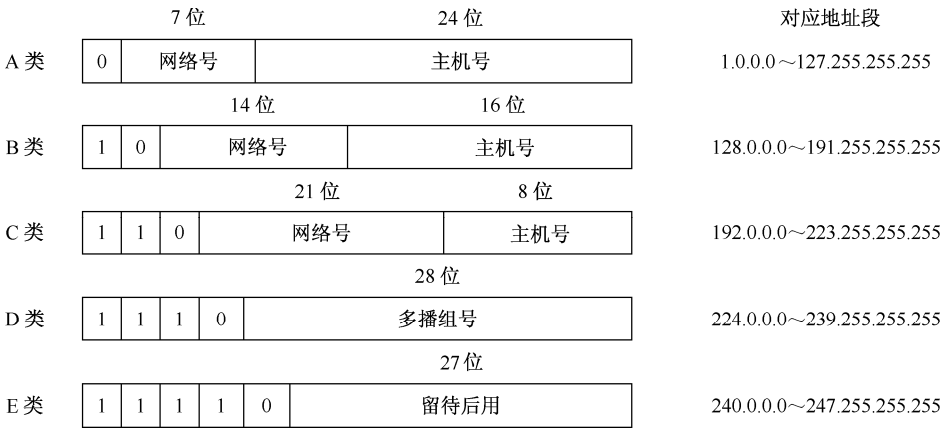


图 11.3 IP地址的分类

IP地址分类方案把32位地址空间分成了大小不同的类，各类包含不同的网络数和主机数。比如，A类地址的网络号用7位表示，可以容纳的最大网络数为$2^7=128$，主机号用24位表示，可以容纳的最大主机数为$2^{24}=16\,777\,216$。因为一些地址作为特殊用途保留使用，实际容纳的主机数或网络数没有那么多，表11.1列出了剔除一些特殊地址后，A～C类地址实际剩余的地址数。即使如此，我们看到A类和B类网络主机数量也是非常巨大，实际上大多

数情况下都不需要一个单网上有这么多计算机。为了解决这个问题，引入了子网的概念，从 IP 地址表示主机号的字段部分“借”出一些位来定义子网，进一步细化 A 类和 B 类地址，图 11.4 给出了一个子网化的 B 类地址。那么子网 ID(SubnetID)和主机 ID(HostID)如何区分呢？子网掩码可以完成这一功能。子网掩码由 32 位组成，其中与网络 ID(NetID)和子网 ID 对应的部分全被置 1，与主机号对应的部分全被置 0，子网掩码与 IP 地址进行“与”操作，就可以提取出 IP 地址的网络号与子网号部分。网络号与子网号可以通过地址类型进行分离。地址分类和子网划分，实际上是为了减小路由表，提高寻径的效率。有了地址分类，就不需要为网络中每一个主机保存一份路由表，只要保存一份去往该子网的路径即可。

表 11.1　三类 IP 地址类中包含的网络数和每类网络中所包含的主机数目

地　址　类	二进制位数	网　络　数	二进制位数	主　机　数
A	7	126	24	16 777 214
B	14	16 384	16	65 534
C	21	2 097 152	8	254

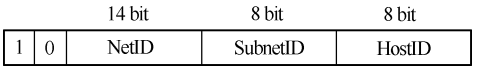


图 11.4　一个子网化的 B 类地址

11.1.3.2　特殊 IP 地址

IP 地址采用点分十进制形式表示，每个地址表示为 4 个以小数点隔开的十进制整数，每个整数对应一个字节。例如：162.105.30.1。有一些 IP 地址有特殊的用途，不分配给 Internet 上的计算机，这样的地址有：

(1) 局域网保留地址，只用于内部通信，不能进行路由。这样的地址包括 A 类地址段的 10.0.0.0～10.255.255.255；B 类地址段的 172.16.0.0～172.31.255.255；C 类地址段的 192.168.0.0～192.168.255.255。

(2) 主机号全“0”全“1”的地址在 TCP/IP 协议中有特殊含义，不能用做一台主机的有效地址。主机号的所有位都为“0”的地址表示网络本身；127.*. *. *保留做回路(Loopback)测试，大多数系统使用 127.0.0.1；全零(0.0.0.0)地址对应于当前主机。

(3) 广播地址包含三类。有限广播地址：255.255.255.255，它只向本地网络广播，在不知道子网掩码时使用。网络直接广播地址主机ID的各位全为1，如 A 类网络的直接广播地址是 netid.255.255.255，netid 是A类网络的ID。子网直接广播地址是除了特定子网 ID 外，主机 ID 各位全为“1”的地址，表示向某网络内所有主机发送信息，需要知道子网掩码。

(4) 169.254. *. *。如果你的主机使用 DHCP 功能获得一个 IP 地址，当 DHCP 服务器发生故障，或者响应时间太长超过系统规定的时间时，在 Windows 2000 以前的系统中，则自动配置成“IP 地址：0.0.0.0”、“子网掩码：0.0.0.0”的形式；而对于 Windows 2000 以后的操作系统，自动配置成“IP 地址：169.254. *. *”，“子网掩码：255.255.0.0”的形式。

11.1.3.3　IPv4 数据报

TCP/IP 协议使用 IP 数据报(IP Datagram)来命名网络层的数据包，每个 IP 数据报由一个头部和紧跟其后的数据组成。图11.5给出了 IPv4 数据报的组成。IP 数据报头部从 4 位的版本

号开始，目前取值为 4。接下来是 4 位的首部长度，取值范围为 5～15，单位为字节。服务类型，指定传输的优先级、传输速度、可靠性和吞吐量等内容。报文总长度用两个 8 位组表示，指示数据报的总长度，包含报头和数据区，最大长度为 65 535 字节。16 位报文标识字段唯一标识一个数据报，如果数据报分段，则每个分段的标识都一样。标志字段为 3 位，最高位未使用，定义为 0，其余两位分别是 DF(不分段)和 MF(更多分段)的指示位，如 DF 为 1 表示不分段，DF 为 0 表示分段。段偏移量，以 8 个字节为单位，指出该分段的第一个数据字在原始数据报中的偏移位置。生存时间，取值 0～255，以秒为单位，每经过一个路由节点减 1，为 0 时被丢弃。协议字段指明包含在该数据报中的数据的协议类型，1 为 ICMP，4 为 IP，6 为 TCP，17 为 UDP 等。首部校验和字段，每通过一次网关都要重新计算该值，用于保证IP首部的完整性。选项字段长度可变，提供某些场合下需要的控制功能，IP 首部的长度包括控制、保留、调试和测量选项，必须是 4 字节的整数倍，如果选项长度不是 4 的整数倍，必须填充数据。

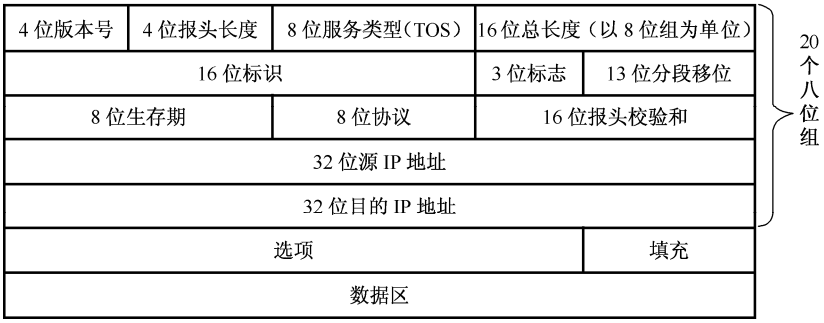


图 11.5　IPv4 数据报的组成

每一种硬件技术都规定了一个数据帧所能携带的最大数据量，这一限制称为最大传输单元(Maximum Transmission Unit, MTU)，因而每一个数据报必须小于或等于一个网络的 MTU，否则无法进行封装。当一个数据的尺寸大于发送网络的MTU时，会将数据报分成若干较小的部分，称为分段(Fragment)，然后再将每段独立发送。图11.6表示一个 IP 数据报被分成两段，每段携带原始数据的一部分。

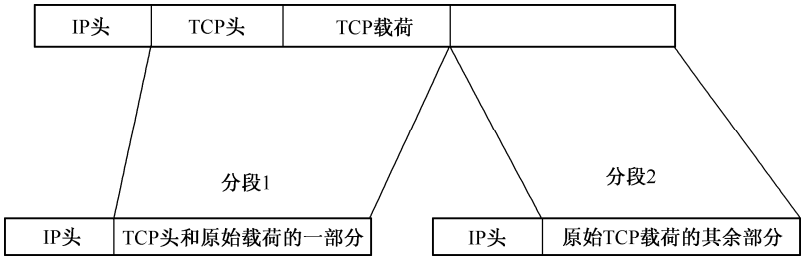


图 11.6　IP 数据报分段过程

11.1.4　IPv6 数据报

IPv6 保留了IPv4的许多成功的特征，比如，IPv4 和IPv6 都是无连接的。一个 IPv6 数据报开始于一个基本头部，后面跟着零个或者多个扩展头部，然后是数据区，如图11.7所示。

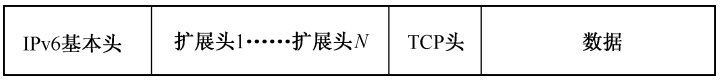


图 11.7　IPv6 数据报的通用格式

IPv6 的基本头部包含的信息比 IPv4 少，但尺寸是 IPv4 的两倍大，其格式如图11.8 所示。与 IPv4 不同的是，有效载荷只包含所携带数据的大小，不包括头部长度。跳数限制对应于 IPv4 的生存期字段，跳数限制被源端设置为某个最大值，网络节点转发一次就减1，当该值减为 0 时，就丢弃数据包。通信量类型用来标记和区分不同优先级别的 IPv6 数据包，流标签用于主机标记需要特殊处理的数据包。下一个首部用于指定基本头后面的信息类型，可以是一个 IPv6 扩展头，或者是上层协议的类型。

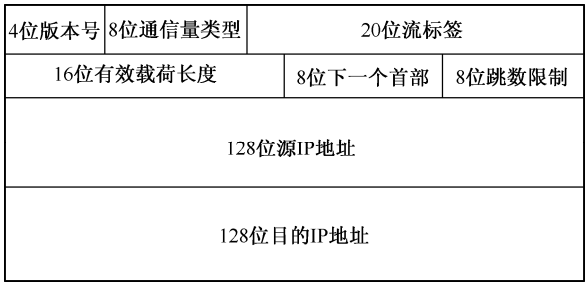


图 11.8　IPv6 的基本头部格式

11.1.5　ARP 协议

在局域网中，现在用得最多的是以太网协议。每个以太网卡都有唯一的以太网物理地址，也称为硬件地址或MAC地址。ARP解决发送的IP数据包的目的以太地址如何确定的问题。以太地址有 6 个字节，是生产厂商根据分配给它的地址空间直接烧结在网卡上的。ARP 地址翻译是通过查地址翻译表来实现的。

ARP 表由 ARP 协议来完成，工作的过程根据需时再取的原则。当 ARP 查询以太地址时，发送一个ARP包到网络广播地址，每台计算机的ARP模块检查自己的IP地址是否和ARP 包内的 IP 地址相同，如果一致，则返回一个响应，否则，丢弃该 ARP 包。UNIX, Windows 操作系统可以使用命令ARP–a 来查询 ARP 表。表 11.2 是一个 ARP 请求包的例子，包含发送方的 IP 地址和对相应硬件地址的请求，表 11.3 是一个 ARP 响应包的例子，包含发来的 IP 地址和响应的硬件地址。

表 11.2　ARP 请求包示例

硬件地址类型	协议地址类型	
硬件地址长度	协议地址长度	请求
发送者以太地址	00-15-58-12-2F-C3	
发送者 IP 地址	162.105.30.195	
目标以太地址	空	
目标 IP 地址	162.105.30.121	

表 11.3　ARP 响应包示例

硬件地址类型	协议地址类型	
硬件地址长度	协议地址长度	响应
发送者以太地址	00-00-21-F7-17-88	
发送者 IP 地址	162.105.30.121	
目标以太地址	00-15-58-12-2F-C3	
目标 IP 地址	162.105.30.195	

11.1.6 ICMP 协议

ICMP（Internet Control Message Protocol）本身是 IP 的一部分，ICMP 报文用于报告在传输报文过程中发生的各种情况，在 IP 协议栈中必须实现。它的特点是：其控制能力并不用于保证传输的可靠性，它本身也不是可靠传输的，并不用来反映 ICMP 报文的传输情况。

ICMP 报文直接包含在 IP 数据报的净荷数据中，IP 头中协议类型为 1，如图11.9所示。ICMP报文的第一个字节代表ICMP报文的类型，它决定了后续数据的格式，第二字节代码域记录了报文的种类，例如目标不可达，其后是两字节的校验和，后续数据域记录了进一步的细节，视报文类型而有所不同。

图 11.9 ICMP 数据包的格式

ICMP 定义了三种差错报文、两种控制报文，还有四对请求/应答报文，如表 11.4 所示。三种差错报文分别是目标不可达报文、超时报文、参数出错报文。两种控制报文分别是源抑制报文、重定向报文。四对请求/应答报文是回应请求和回应应答报文、时间戳请求和时间戳应答报文、地址掩码请求和地址掩码应答报文、信息请求和信息应答报文。

表 11.4 ICMP 报文类型

类　型	描　述
0	回应请求（Echo Reply）
3	目标不可达（Destination Unreachable）
4	源抑制（Source Quench）
5	重定向（Redirect）
8	回应请求（Echo Request）
11	超时（Time Exceeded）
12	参数出错（Parameter Problem）
13	时间戳请求（Timestamp Request）
14	时间戳应答（Timestamp Reply）
15	信息请求（Information Request）（过时）
16	信息应答（Information Reply）（过时）
17	地址掩码请求（Address Mask Request）
18	地址掩码应答（Address Mask Reply）

回应请求和回应应答报文用于检测目的站的可达性与状态，主机或路由器向指定目的站发送 ICMP Echo 请求报文，请求报文包含一个可选的数据区；收到 Echo 请求报文的机器应立即回应一个 Echo 应答报文，应答报文包含了请求报文中数据的副本。Ping 是 Packet Internet Groper 的缩写，该命令是用来判断远程设备可访问性最常用的方法，它的基本原理就是发送 ICMP Echo 消息，然后等待 ICMP Echo Reply 消息。

当路由器无法转发或投递 IP 数据报时，向源端发回一个目的站不可达报文，并丢弃该数据报。目标不可达报文说明不可达的具体原因：如网络不可达、主机不可达、协议不可达、端口不可达等。

11.1.7　TCP 协议

传输控制协议(TCP)是传输层协议，它从上层协议接收任意长度的报文，并提供面向连接的传输服务。一个应用程序必须首先请求一个到目的地的连接，然后使用这一连接来传输数据。TCP 为了确保服务的可靠使用了各种机制。除了在每一分段提供校验和，还建立了报文的确认和重发机制。TCP 接收数据流，并分成段，然后将这些段送给 IP，因 IP 为无连接的，TCP 为每个段提供顺序同步。下面我们介绍 TCP 分段格式和 TCP 连接的建立和终止机制。

11.1.7.1　TCP 分段格式

TCP/IP 把 TCP 层的数据包称为 TCP 分段(TCP Segment)，图11.10给出了 TCP 分段的格式。

<table>
<tr><td colspan="3">16位源端口号</td><td>16位目的端口号</td></tr>
<tr><td colspan="4">32位序号</td></tr>
<tr><td colspan="4">32位确认序号</td></tr>
<tr><td>4位首部长度</td><td>保留(6位)</td><td>6个标志位</td><td>16位窗口大小</td></tr>
<tr><td colspan="3">16位校验和</td><td>16位紧急指针</td></tr>
<tr><td colspan="4">选项</td></tr>
<tr><td colspan="4">数据区</td></tr>
</table>

图 11.10　TCP 分段格式

源端口号是 16 位字段，表示应用程序发送数据所用的端口，目的端口号是 16 位字段，表示数据包将交付的接收端端口，源和目的主机的 IP 地址加上端口号构成一个 TCP 连接。序号是32 位字段，当数据分段传输时，标记各分段如何组装；确认号为希望接收的下一个数据字的序号。首部长度为 4 位字段，以 4 个字节为单位，首部的基本长度为20个字节。保留字段 6 位，没有使用，置为 0。

6 个标志位是 URG, ACK, PSH, RST, SYN 和 FIN 标志。如果使用了紧急指针，URG 置为 1，紧急指针为当前序号到紧急数据位置的偏移量。ACK 为1 表示确认序号有效，为 0 表示该 TCP 数据包不包含确认信息。PSH 为1 表示是带有 PUSH 标志的数据，接收到数据后不必等缓冲区满再发送。RST 用于连接复位，也可用于拒绝非法数据或连接请求。SYN 用于建立连接，发起连接请求时 SYN＝1，ACK＝0；响应连接请求时 SYN＝1，ACK＝1。FIN 用于释放连接，表示发送方已经没有供发送的数据。

窗口大小表示在确认字节后还可以发送的字节数，用于流量控制。校验和覆盖了整个数据包，包括对数据包的首部和数据，用于数据包的完整性检查。选项字段可以是任意长度，常见的选项是 MSS(Maximum Segment Size)，指明最大分段长度。

11.1.7.2　TCP 连接的建立和终止

TCP 连接的建立过程是一个三次握手，在此过程中双方要互报初始序列号(Initial Sequence Number, ISN)，以保证数据包的接收顺序和发送顺序一致。TCP 通过包头的两个标记位 SYN 和 ACK 来完成连接建立的过程，如图11.11 所示。

第一步，主机 A 向主机 B 发送连接请求数据包，数据包的序号为 ISN_A，SYN＝1，ACK＝0，表示这是一个主动发起请求的数据包。第二步，主机B 收到这个请求包后，记下主机A 的初始序列号 ISN_A，对主机 A 的连接请求发送响应数据包，主机 B 发送的数据包的序号为 ISN_B，

SYN=1，ACK=1，表示这是一个响应请求的数据包，确认序号为 ISN_A+1，表示主机 B 期望从 A 收到的下一个数据包的序号。第三步，主机 A 收到主机 B 的响应数据包后，记下主机 B 的初始序列号 ISN_B，给主机 B 发送响应数据包，表示收到了主机 B 的响应，该数据包的序号为 ISN_A+1，确认序号为 ISN_B+1，表示主机 A 期望从 B 收到的下一个数据包的序号。至此，双方建立了 TCP 连接，可以在两个方向传送数据流了。我们可以注意到，主机 A 和 B 的序号是分别单独编号的。

TCP 通过包头的两个标记位 FIN 和 ACK 来完成连接终止的过程，如图11.12所示。主机 A 通过发送 FIN=1 的数据包来关闭本方数据流，通知主机 B，已经没有数据要发送，但还可以继续接收主机 A 发送的数据，直到对方关闭那个方向的数据流，连接就关闭。终止一个连接要经过四次握手。这是因为一个 TCP 连接是全双工的，即数据在两个方向上能同时传递，因此每个方向必须单独地进行关闭。关闭的原则就是当一方完成它的数据发送任务后就发送一个 FIN 来终止这个方向的连接。

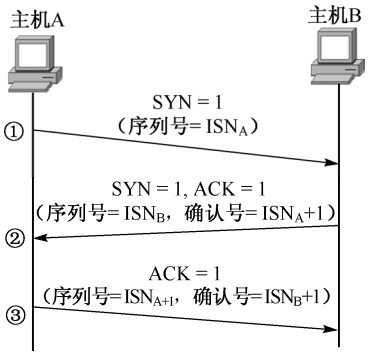


图 11.11　TCP 连接的三次握手

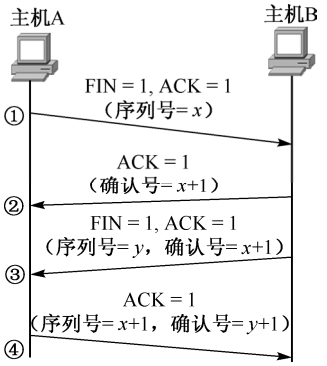


图 11.12　TCP 连接的关闭过程

11.1.8　UDP 协议

与 TCP 协议相反，用户数据报协议（UDP）协议是无确认的数据报服务，只是简单地接收和传输数据，UDP 比 TCP 传输数据快，主要用于可靠性高的局域网中。UDP 的数据包格式如图11.13所示。

16位源端口号	16位目的端口号
16位的UDP长度	16位的UDP校验和
数据区	

图 11.13　UDP 的数据包格式

11.1.9　TCP 和 UDP 端口

如果信息交换的时候只是简单地定义目的 IP 地址，虽然信息可以到达计算机，但是却不知道交给哪个进程去处理。TCP 和 UDP 使用端口解决了这一问题。由于 TCP/IP 传输层的 TCP 和 UDP 两个协议是两个完全独立的软件模块，TCP 和 UDP 的端口号的编号也是独立的，都是 0～65 535。端口号 0～1023，称为**公认端口**（Well known ports），由互联网分配号码机构（Internet Assigned Numbers Authority，IANA）集中分配，与因特网上常见的一些服务相对应。表11.5 和表11.6 分别给出了TCP 和UDP 的一些公认端口。端口号1024～65 535 称为**注册端口**

(Registered Ports)，可以被系统中的任意用户进程使用。系统管理员可以“重定向”端口，一种常见的技术是把一个端口重定向到另一个地址。例如默认的 HTTP 端口是 80，不少人将它重定向到另一个端口，如 8080。

表 11.5　TCP 公认端口

端　口　号	描　　述
0	保留
20	FTP-data
21	FTP-command
23	Telnet
25	SMTP
53	DNS
80	WWW
88	Kerberos
139	NetBIOS-SSN

表 11.6　UDP 公认端口

端　口　号	描　　述
0	保留
49	login
53	DNS
69	TFTP
80	WWW
110	POP3
137	NetBIOS-NS
139	NetBIOS-SSN
161	SNMP

11.2　Internet 安全性途径

力求简单高效的设计初衷使TCP/IP 协议集的许多安全因素未得以完善，安全缺陷的一些表现有：TCP/IP 协议数据流采用明文传输，TCP/IP 协议以 IP 地址作为网络节点的唯一标识，此举并不能对节点上的用户进行有效的身份鉴别，此外，协议本身的一些特点也被利用实施网络攻击。

为了实现Internet安全性，从原理上来说可以实现在TCP/IP协议的任意一层。无论在协议的什么层次实现安全性，下面这些安全服务是必须提供的：机密性、完整性、身份鉴别、抗否认、访问控制、相应的密钥管理和抗重放。但在不同层次实现安全性有着不同的特点。

应用层安全必须在终端主机上实施，以用户为背景执行，便于实施强大的基于用户的身份鉴别和访问控制。应用层的数据加密，数据在链路、路由器和网关中都是密文状态，只有到用户的主机上才恢复成明文，减小数据受到威胁的机会。应用可以自由扩展，应用程序对数据有着充分的理解。想要区分一个具体文件的不同安全性要求，那就必须借助于应用层的安全性。提供应用层的安全服务实际上是最灵活的处理单个文件安全性的手段，缺点是必须针对每个应用设计一套安全机制，要对每个应用或者应用协议分别进行修改。为了保证应用层数据的安全，应该加强数据的备份和恢复措施，对资源的有效性进行控制，资源包括各种数据和服务。应用层安全协议有 PGP, PEM, S/MIME, SHTTP, SSH 等，如图11.14所示。

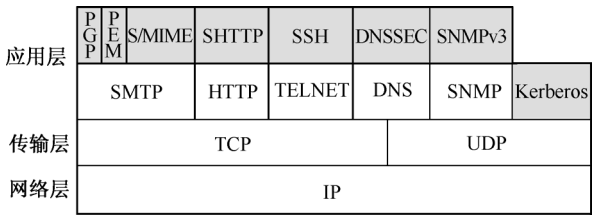


图 11.14　应用层安全协议

在传输层实现安全机制，不会强制要求每个应用都在安全方面做出相应改进，缺点是为提供由具体用户决定的服务，通常假定只有一名用户使用系统，与应用级安全类似，只能在端系统实现，应用程序仍需要修改，才能要求传输层提供安全服务。传输层的安全协议有SSL/TLS，如图11.15所示。

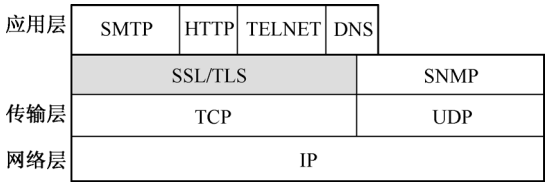


图 11.15　传输层安全

网络层安全的优点是密钥协商的开销被大大削减了，因为只需要在网络节点之间协商密钥，其上的多种传输协议和应用程序可共享网络层提供的密钥管理架构，对应用程序的改动要少得多，能很容易构建 VPN。缺点是因为缺乏用户背景，很难解决“抗抵赖”之类的问题。网络层的安全协议是 IPsec，如图11.16所示。

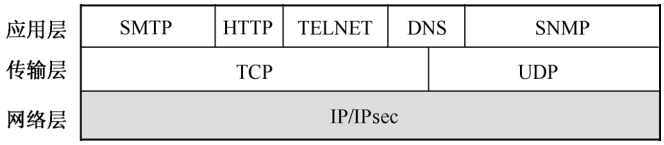


图 11.16　网络层安全

在网络接口层实现安全的最大的好处在于速度，因为硬件实现加解密比软件实现要快得多，缺点是不易扩展，在 ATM（自动柜员机）上得到广泛应用。

11.3　IP 的安全

人们在使用中逐渐发现了 IPv4 的许多缺陷，比如，缺乏对通信双方身份真实性的鉴别能力；缺乏对传输数据完整性和机密性保护的机制；IP 的分组和重组机制不完善；由于IP 地址可软件配置， IP 地址的表示不需要真实并确认真假，以及基于源 IP 地址的鉴别机制，IP 层存在：业务流被监听和捕获、IP 地址欺骗、信息泄漏和数据项篡改、IP 包伪造、Ping Flooding 和 Ping of Death 等大量的攻击。

11.3.1　IPsec 概述

IPsec 的设计目标是为 IPv4 和 IPv6 提供可以互操作的、基于密码的高质量的安全，IPsec 提供的安全服务有：访问控制、无连接完整性、数据源鉴别、重放的检测和拒绝、机密性（通过加密）、有限通信流的机密性。

1994 年 IETF 专门成立 IP 安全协议工作组，来制定和推动一套称为IPsec的IP安全协议标准。1995 年 8 月公布了一系列关于 IPsec 的建议标准。1996 年，IETF 公布了下一代 IP 的标准 IPv6，把鉴别和加密作为必要的特征，IPsec 成为其必要的组成部分。1999 年年底，IETF 安全工作组完成了 IPsec 的扩展，在 IPsec 协议中加上 ISAKMP（Internet Security Association and

Key Management Protocol)协议，密钥分配协议 IKE, Oakley。ISAKMP, IKE 和 Oakley 支持自动建立加密和鉴别信道以及密钥的自动安全分发和更新。2005 年年底，对 IPsec 所采用和支持的算法等特性又进行了新的修订，公布了一系列新的 IPsec 标准。

IPsec 可以为在 LAN, WAN 和 Internet 上的通信提供安全性，如一个公司的各分支办公机构通过 Internet 互联、终端用户通过 Internet 获得对某站点的安全远程访问等。IPsec 的主要特征是可以支持 IP 层所有流量的加密和/或鉴别。因此可以增强所有分布式应用的安全性。IPsec 可以以端到端(end-end)、端到路由(end-router)和路由到路由(router-router)的方式实现，分别为主机到主机、主机到路由设备和路由设备之间提供安全通信，常用于在两个网络之间建立虚拟私有网(VPN)。

11.3.2　IPsec 的文档组成

IPsec 包含三个系列的文档，第一系列文档重要的有 1995 年发布的 RFC1825, RFC1826, RFC1827, RFC1828, RFC1829。第二系列的文档包括 1998 年发布的 RFC2401～RFC2412，重要的有 RFC2401, RFC2402, RFC2406 和 RFC2408。最新系列的文档发布于 2005 年，包括 RFC4301～RFC4309，这些文档分别为：IP 安全结构、鉴别头(AH)协议、载荷安全封装(ESP)协议、加密和鉴别算法、密钥管理协议及其支持的算法，如图11.17所示。

RFC4301 是“IP 协议安全架构”(Security Architecture for the Internet Protocol)，用于代替过时的RFC2401。该文档不是一个关于Internet的安全架构，它只是解释通过一些密码和协议的安全机制实现 IP 层的安全。文档描述了实现 IPsec 的系统要求，系统的基本要素，以及这些要素如何在 IP 环境中配置，IPsec 协议提供的安全服务等内容。

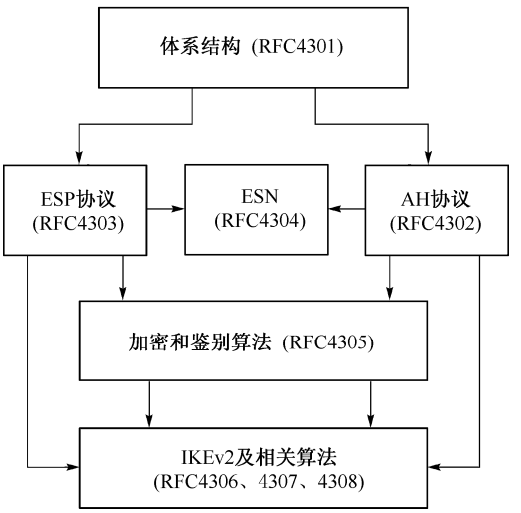


图 11.17　IPsec 文档的组成

IPsec 包含了两个安全协议，鉴别头(Authentication Header，AH)协议和封装安全载荷(Encapsulating Security Payload ，ESP)协议。IPsec 在 IPv6 中是强制的，在 IPv4 中是可选的。这两种情况下都是采用在主 IP 报头后面接续扩展报头的方法实现的。AH 是鉴别的扩展报头，ESP 是实现加密和鉴别(可选)的扩展报头。RFC4302 描述了“IP 鉴别头(IP Authentication Header)”，代替过时的 RFC2402，用于提供无连接完整性、数据源鉴别和抗重放服务。RFC4303

描述了“IP 封装安全载荷”(IP Encapsulating Security Payload)，代替过时的 RFC2406，用于提供机密性、数据源鉴别、无连接完整性、抗重放和有限通信流机密性服务。

RFC4305 是“ESP 和 AH 的密码算法实现要求”(Cryptographic Algorithm Implementation Requirements for ESP and AH)，用于代替过时的 RFC2404 和 RFC2405。为了保证各厂商独立实现的产品之间的互操作性，规定一个至少强制(must)实现的算法是必要的，该文档定义了一些 AH和ESP强制实现的算法，以及应该(should)实现的算法，这些算法未来可能成为强制实现的算法。

AH和 ESP协议都使用了序列号来检测重放，RFC4304 是对 ISAKMP 的参数规定的扩展，对于一个特定的安全关联，允许传统的 32 位序列号和扩展的 64 位序列号。

RFC4306 描述了“第二版的 IKE 协议”(Internet Key Exchange(IKEv2) Protocol)，这一版的IKE规范组合了以前独立的三个文档，包括ISAKMP(RFC 2408)，IKE(RFC 2409)，Internet 解释域(DOI，RFC 2407)，并代替它们。IKEv2 提供了一种协商用于任何关联的算法的机制，为了保证各厂商独立实现的产品之间的互操作性，同样要规定强制实现的算法，RFC4307 规定了 IKEv2 必须强制实现的密码算法以及应该实现的密码算法，应该实现的算法将来可能发展成强制算法。

11.3.3 安全关联

安全关联(Security Association，SA)是 IPsec 的基础，AH 和 ESP 都用到 SA，IKE 的一个主要任务就是建立和维护 SA。一个**关联**就是一个对通信流提供安全服务的单向连接，安全服务由 AH 或 ESP 或两者同时提供，如果需要双向安全交换，则需要两个 SA，每个方向一个。

每个 SA 通过三个参数来标识：安全参数索引(Security Parameters Index，SPI)、对方 IP 地址、安全协议标识(AH 或 ESP)。

SA 与 IPsec 系统中实现的两个数据库有关：安全策略数据库(SPD)和安全关联数据库(SAD)。

一些必须列入安全关联数据库的数据如下：

- 安全参数索引(Security Parameter Index，SPI)：一个由接收方选择的唯一标识一个 SA 的 32 位数值，SPI 用于系统从 SAD 查找合适的 SA 处理数据包。
- 序列号计数器(Sequence Number Counter)：一个 64 位的计数器，用于产生 AH 或 ESP 头的序列号。默认情况下，序号是 64 位，也可以通过协商选择 32 位。
- 序列号溢出标志(Sequence Counter Overflow)：一个指示序列号溢出是否产生审计事件的标志，阻止在该 SA 下继续传输数据包，或者允许重置。
- 抗重放窗口(Anti-Replay Window)：一个 64 位计数器或位图，用于判断进入的 AH 或 ESP 数据包是否是重放。
- AH 信息：AH 的鉴别算法、密钥等。
- ESP 加密算法信息：ESP 加密算法、密钥、模式、初始值等。
- ESP 完整性算法信息：ESP 完整性算法、密钥等。
- ESP 组合模式(加密和完整性)算法、密钥等。
- SA 的生存期(Lifetime of this SA)：一个时间周期，指示超过一定时间后一个 SA 必须终止，或者被一个新的 SA 代替。

- IPsec 协议模式(IPsec protocol mode)：隧道(Tunnel)或传输(Transport)，指示 AH 或 ESP 的哪一种模式被用于这一 SA。
- 状态分段检查标志(Stateful Fragment Checking Flag)：指示对这一 SA 是否使用状态分段检查。
- Path MTU：最大传输单元和迟滞变量(Aging Variables)。
- 隧道头的源 IP 地址和目的 IP 地址，只有隧道模式的 IPsec 协议才使用。

11.3.4　鉴别头协议

AH协议用于提供无连接完整性、数据源鉴别和抗重放服务，它既可以单独使用，也可以与ESP协议联合使用，它提供的完整性服务与ESP协议提供的完整性服务的差别在于保护的数据范围不同。鉴别头试图保护IP 数据报尽可能多的字段，那些在传输过程中要发生变化的字段就只能被排除在外。

11.3.4.1　鉴别头格式

AH 头如图11.18所示，由如下域组成：

- 邻接头(Next Header)：跟在AH之后下一个报头的类型，标识被传送数据所属的协议，其值由 IANA 定义的 IP 协议号决定，如 4 表示 IPv4，6 表示 TCP。
- 载荷长度(Payload Length)：以 32 位字为单位的 AH 的长度减 2。例如完整性算法产生 96 位的鉴别数据，这一区域就是 4(3 个 32 位的固定区域加上 3 个 32 位字的鉴别数据，减 2)。
- 安全参数索引(Security Parameters Index ，SPI)：是一个任意的 32 位的值，接收方用于标识一个流入数据包所绑定的安全关联 SA。
- 序列号：用来避免重放攻击。为了支持高速 IPsec 实现，应该提供一个新的选项，作为对现在 32 位序列号的扩展，但必须通过 SA 管理协议的协商使用扩展的序列号。
- 鉴别数据(Authentication Data)：可变长度的域，包含针对这个数据包的完整性校验值(Integrity Check Value，ICV)。这个值必须是 32 位的整数倍。

8位邻接头	8位载荷长度	保留
32位安全参数索引SPI		
序列号		
鉴别数据（可变长度）		

图 11.18　IPsec 鉴别头

11.3.4.2　鉴别头位置

鉴别头可以用两种方式实现：传输模式或者隧道模式。

在 IPv4 的传输模式中，AH 插入到 IP 头之后，IP 载荷之前，或者任何一个已经插入的 IPsec 头之前，如图11.19(b)所示。在 IPv6 的传输模式中，AH 被看做端对端的载荷，因此它必须放在逐跳、路由、分段扩展头之后，目的地址作为可选扩展头出现在AH头之前或者之后，由

特定的语义决定，如图11.20(b)所示。在传输模式中，鉴别头保护的是除原始 IP 头可变域之外的内容。

在隧道模式中，内部 IP 头携带着最终的 IP 源地址和目的地址，外部 IP 头包含的是 IPsec 对端的地址，比如安全网关的地址。混合的内部和外部 IP 版本是允许的，比如外部是 IPv6，内部是 IPv4。在隧道模式中，鉴别头保护的是全部内部 IP 数据报，包括完整的 IP 头。隧道模式 AH 的位置，相对于外部 IP 头来说，与传输模式是一致的，图11.19(c)给出了 IPv4 隧道模式下 AH 头的位置，图11.20(c)给出了 IPv6 隧道模式下 AH 头的位置。

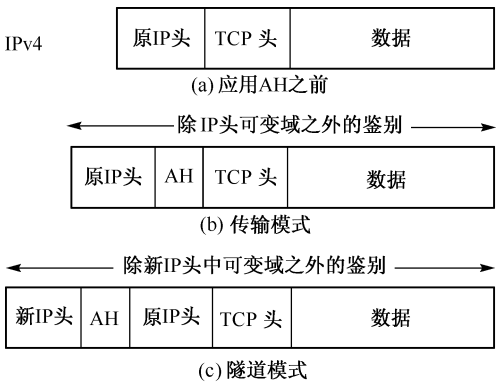


图 11.19　IPv4 传输模式与隧道模式 AH 的位置和鉴别范围

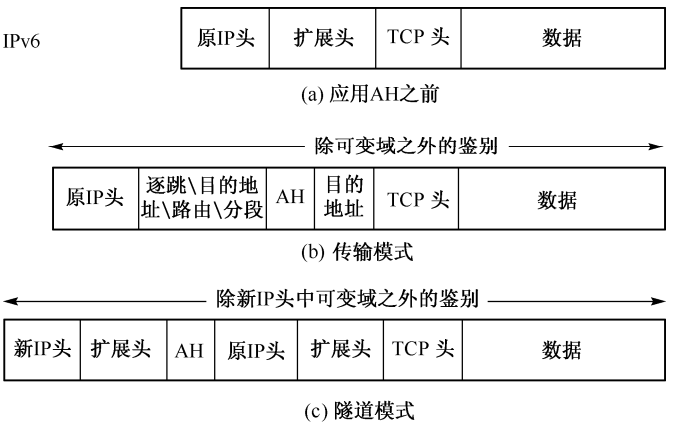


图 11.20　IPv6 传输模式与隧道模式 AH 的位置和鉴别范围

11.3.4.3　完整性算法

计算完整性校验值的完整性算法由SA商定，可以是基于对称加密算法或者是基于单向函数的MAC码。RFC4305 规定必须实现的完整性算法是 HMAC-SHA1-96[RFC2404]，应该实现的完整性算法是AES-XCBC-MAC-96[RFC3566]，可以实现HMAC-MD5-96[RFC2403]。HMAC-SHA1-96 和 HMAC-MD5-96 都使用了基于 SHA-1 或者 MD5 的 HMAC 算法，计算全部 HMAC 值，然后取前 96 位。AES-XCBC-MAC-96 算法是使用 1 个 1 和多个 0 填充的基本 CBC-MAC 的变体，AES-XCBC-MAC-96 的计算使用 128 位密钥长度的 AES。假设消息 M 由 n 个分组组成，$M[1]\cdots M[n]$，分组 $M[1]\cdots M[n-1]$是 128 位，分组 $M[n]$的尺寸为 1～128 位之间，AES-XCBC-MAC-96 的计算步骤如下：

(1) 从 128 位的密钥 K 按如下方式推出 3 个 128 位的密钥 K_1, K_2 和 K_3：

$$K_1 = \mathrm{E}_K(0x01010101010101010101010101010101)$$

$$K_2 = \mathrm{E}_K(0x02020202020202020202020202020202)$$

$$K_3 = \mathrm{E}_K(0x03030303030303030303030303030303)$$

(2) 定义 $X[0]$ = 0x00000000000000000000000000000000

(3) 对于每一个分组 $M[i]$，$i = 1 \cdots n-1$，计算 $X[i] = E_{K1}(M[i] \oplus X[i-1])$

(4) 对于分组 $M[n]$

(a) 如果 $M[n]$是 128 位，$X[n] = E_{K2}(M[n] \oplus X[n-1])$。

(b) 如果 $M[n]$小于 128 位，用 1 个 1 和多个 0“10…00”把 $M[n]$补足 128 位，计算 $X[n] = E_{K3}(M[n] \oplus X[n-1])$。

最后，取最左边的 96 位作为鉴别数据。

AH 的 ICV 根据如下部分进行计算：

- AH 头之前的 IP 头或者扩展头传输中不变的部分，或者可以预测的部分。
- AH 头不含鉴别数据的部分。
- AH 头之后传输中不变的部分。

如果某个域在传输过程中被修改的话，计算 ICV 时该域要被置为 0。如果该域是变化的，但是对于接收者是可预测的，计算 ICV 时就把那个值插入到域中。不去掉变化的域，而是用0进行填充，能够保证域的长度不改变。对于IPv4 来说，在有宽松和严格路由选项的情况下，目的地址就是可变但可预测的；版本号、报头长度、总长度、标识、协议、源地址和没有宽松和严格路由选项的目的地址是不变的部分；标志(Flag)、分段偏移、生存期和包头校验和就是可变的部分。对于IPv6 来说，不可变的部分是版本、载荷长度、下一个首部、源地址和没有路由扩展头时的目的地址。可变但可预测的部分是带路由扩展头时的目的地址。我们看到，源和目的地址都是受保护的，可防地址欺骗。

11.3.4.4　AH 数据包处理过程

对于流出(Outbound Packet)的包，需要查找相应的SA，为其产生抗重放的序列号，计算ICV，最后进行分段处理。

对于接收到的包(Inbound Packet)首先要进行分片装配，接下来可以根据 SPI、协议域(AH 或 ESP)、目的地址和源地址在 SAD 中查找对应的 SA，然后使用一个滑动窗口来检查序列号的重放，最后对 ICV 进行验证。如果都验证正确，数据报就被作为有效数据。

11.3.5　封装安全载荷协议

封装安全载荷协议提供保密功能，包括报文内容的机密性和有限的通信量的机密性，也提供无连接的完整性服务。ESP 以三种方式提供这些服务：

- **只提供机密性服务**：这只是标准规定的一种可以实现的方式，不是必需的，只提供机密性服务，而由更高层来实现完整性服务，可以得到更高的性能。
- **只提供完整性服务**：这是标准规定的一种必须实现的方式，ESP提供的鉴别服务是一个比 AH 提供的鉴别服务更有吸引力的选择，它处理速度快，并且遵守许多实现中的流水线技术。

- **同时提供机密性和完整性服务**：这也是标准规定的一种必须实现的方式。

它将需要保密的用户数据进行加密后再封装到一个新的 IP 包中，ESP 只鉴别 ESP 头之后的信息，加密算法和鉴别算法由 SA 指定，ESP 也有两种应用模式：传输模式和隧道模式。只有当 ESP 应用在隐藏源地址和目的地址的方式下才提供通信流的机密性，比如安全网关之间的隧道模式。

11.3.5.1 ESP 格式

图11.21 说明了 ESP 数据包的格式，它包含如下域：

- **安全参数索引(SPI)**：是任意32 位的数值，用于接收方判断进入的数据包与哪个SA 绑定，这个域是必需的。对于每一个IPsec 保护的进入数据包，要按照 SA 标识匹配最长的优先原则在SAD 中查找其SA，也即按照{SPI，目的地址，源地址}，{SPI，目的地址}，{SPI}的顺序进行匹配。如果找到了相应的匹配项，就使用相应项处理进入的数据包，否则丢弃数据包并记录。
- **序列号**：是一个无符号的 32 位域，包含一个由发送方不断增 1 的计数器。当一个 SA 被建立后，发送方和接收方的计数器被初始化为 0。
- **载荷数据**：其结构取决于加密算法和模式。密文的覆盖范围是载荷数据和 ESP 尾部。
- **填充域**：填充的需要来自以下几个方面：加密算法要求明文是分组的整数倍，因此需要把明文长度(包括载荷数据、填充、填充长度和邻接头域)扩展到需要的长度；除了加密算法的要求，ESP 格式要求密文数据是4 字节的整数倍；填充也可以隐藏载荷的实际长度，提供通信流量的保护。
- **填充长度**：它的有效范围是 0～255 字节。
- **邻接头**：这个 8 位的域也是强制的，它指示包含在载荷中的数据类型。
- **鉴别数据**：该域的计算覆盖ESP 头、载荷、ESP尾部。该域的长度是任意的。ESP 尾部由填充、填充长度和邻接头域组成。

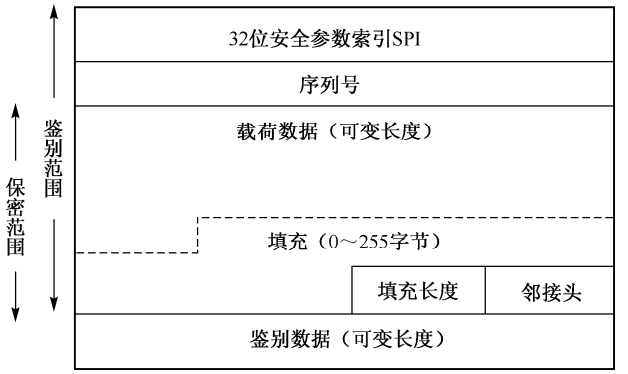


图 11.21 IPsec ESP 格式

11.3.5.2 ESP 头的位置

ESP 的两种应用模式为：传输模式和隧道模式。

在传输模式中，ESP 头被插入到 IP 头之后和下一层协议之前，如 TCP, UDP 和 ICMP。图11.22(a)及图11.23(a)分别给出了在 IPv4 和 IPv6 中传输模式 ESP 的格式。在 IPv4 中，ESP 头被插入到原 IP 头之后。在 IPv6 中，ESP 被看做端对端的载荷，出现在逐跳(hop-by-hop)、

路由和分段扩展头之后。目的地址扩展头根据语义可以出现在 ESP 头之前、之后或同时都出现，因为 ESP 只保护 ESP 头之后的区域，目的地址出现在 ESP 头之后是更期望的做法。

在隧道模式中，内部 IP 头携带着真正的 IP 源地址和目的地址，外部 IP 头携带的是 IPsec 对端的 IP 地址，例如安全网关的 IP 地址。ESP 保护完整的内部 IP 数据包，包括完整的内部 IP 头。ESP 的位置，相对于外部 IP 头来说，与传输模式相同。图11.22(b)及图11.23(b)分别给出了在 IPv4 和 IPv6 中隧道模式 ESP 的格式。

传输模式中，加密的数据范围是原始的上层协议信息和 ESP 尾部，隧道模式中，加密的数据范围包括原始的整个 IP 数据报和 ESP 尾部。鉴别的数据范围是 ESP 头和所有密文。

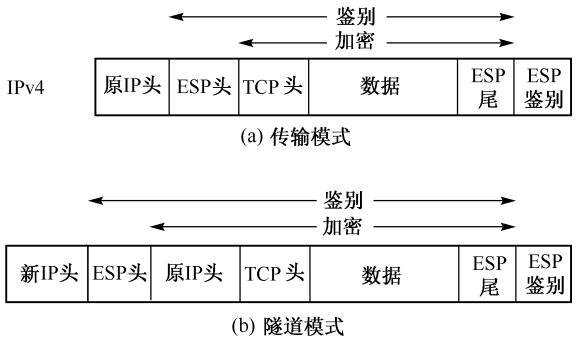


图 11.22　IPv4 ESP 传输模式和隧道模式的位置

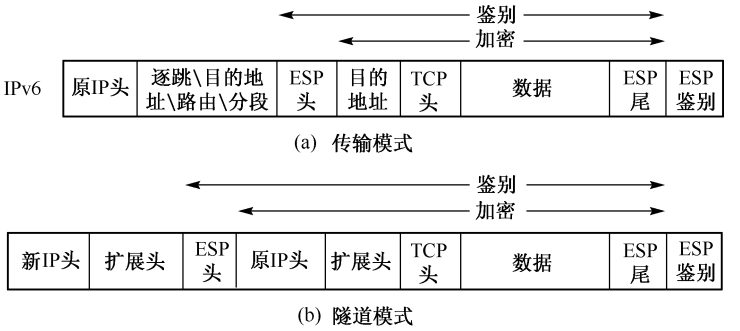


图 11.23　IPv6 ESP 传输模式和隧道模式的位置

11.3.5.3　ESP 的算法和实现要求

ESP 的支持的加密和鉴别算法及其实现要求参见表 11.7。表中的一些符号的含义如下：

- MUST(必须)或者 REQUIRED(要求)，SHALL 这些词，表示该定义是规范绝对要求的。
- MUST NOT 或者 SHALL NOT，表示该定义是规范绝对禁止的。
- SHOULD(应该)或者RECOMMENDED(推荐)，表示在一些特殊的情况下，存在忽视某些项的有效理由，但在做出另一个选择之前要完全理解和慎重权衡。
- SHOULD NOT 或者 NOT RECOMMENDED 表示在一些特殊的情况下，存在接受特定行为的理由，但在实现这种行为时要完全理解和慎重权衡。
- MAY (可以)或者 OPTIONAL，表示该项是一个选择项。
- SHOULD+这个词的含义与 SHOULD 相同，但表示该算法将来可能会被提升为一个 MUST 的算法。

- SHOULD–这个词的含义与SHOULD相同，但表示该算法将来可能会被降低为一个可以(MAY)实现的算法或者更差的情况。
- MUST–这个词的含义与 MUST 相同，但表示该算法将来可能不再会是强制的。

表 11.7 ESP 的算法和实现要求

实 现 要 求	加 密 算 法	完整性算法
MUST	NULL	HMAC-SHA1-96 [RFC2404]或 NULL
MUST–	TripleDES-CBC [RFC2451]	
SHOULD+	128 位密钥的 AES-CBC [RFC3602]	AES-XCBC-MAC-96 [RFC3566]
SHOULD	AES-CTR [RFC3686]	
SHOULD NOT	DES-CBC [RFC2405]	
MAY		HMAC-MD5-96 [RFC2403]

11.3.5.4 ESP 数据包处理过程

对于发出去的包，首先需要查找相应的SA，接着，封装必要的数据，放到ESP的载荷数据域中，不同的模式，封装数据的范围不同，增加一些必要的填充数据，使用SA中商定的加密算法和密钥加密载荷数据和 ESP 尾部，如果还选择了完整性算法，应该在应用完整性算法之前进行加密，加密不包括 ICV 域。这种处理顺序可以让接收方快速检测和拒绝重放的数据包，也可以使接收方的解密和完整性检测并行处理。然后，产生序列号，针对加密后的数据计算 ICV。最后进行必要的分片处理。

对于接收到的包，首先进行分片装配，接着依据数据包的SPI、目标IP地址和ESP协议信息在 SAD 中查找匹配的 SA，然后使用一个滑动窗口检查序列号的重放，最后对 ICV 进行检查。如果都验证通过的话，根据SA中指定的算法和密钥、参数，解密加密数据，去掉填充数据，重构原始的 IP 包。

11.3.6 安全关联组合

因为一个SA只能对一个单向的通信流提供一个安全服务AH或ESP，当特定的通信量需要同时调用 AH 和 ESP 服务时，就要为相同的通信流提供多个 SA。SA 可以通过两种方式组合。一种是传输邻接(Transport Adjacency)，对同一数据分组应用多种协议，不形成隧道；另一种是隧道迭代(Iterated Tunneling)，通过多层隧道嵌套实现多层安全协议的应用。这两种方式也可以组合使用，例如在主机之间实现传输模式的 SA，而在网关之间实现隧道模式的 SA。

当希望为通信流同时提供加密和鉴别服务时，有以下三种组合方式：

(1) 使用同时提供加密和鉴别功能的ESP，这种方式又分为传输模式和隧道模式两种。传输模式的 ESP 不保护 IP 头。隧道模式的 ESP 保护整个 IP 数据包。这两种方式都是对密文进行鉴别而不是对明文。

(2) 传输邻接。使用两个捆绑的传输 SA，内部是没有鉴别选项的 ESP SA，外部是 AH SA，如图11.24 所示。此时，仅对 IP 载荷加密，但鉴别范围包含除可变域之外的原始 IP 头。这种处理顺序可以让接收者在解密之前完成对重放和伪造的数据包的快速检测和拒绝，因此能够减少拒绝服务攻击的可能性，它也允许接收者同时进行解密和鉴别。

(3) 传输隧道束(Transport-Tunnel Bundle)。在这种模式中，加密之前进行鉴别有更多的优点，一方面可以使鉴别数据得到保护，另一方面在加密之前进行报文鉴别更加方便。

可以采用内部 AH 的传输 SA 和外部 ESP 的隧道 SA 的方式，如图11.25 所示。此时，除 IP 头可变域之外的原始数据包都得到鉴别，除了新 IP 头之外的整个数据包都得到加密。

图 11.24　传输邻接方式的加密和鉴别

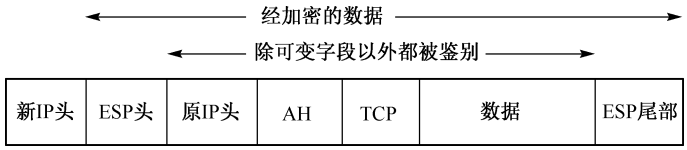


图 11.25　传输隧道束方式的加密和鉴别

11.3.7　密钥管理

IKE（Internet Key Exchange）是IPsec的一个组件，用于执行相互鉴别，建立和维护安全关联。IPsec 对 IP 数据报提供的安全服务需要双方共享一些信息，比如使用什么样的密码算法，密码算法使用的密钥等。以手工方式共享这些信息的扩展性不好，因此需要一个协议动态共享这些信息，IKE就是一个这样的协议。IKE执行双方的相互鉴别，建立一个包含共享秘密信息的IKE SA，用于建立 ESP SA 和 AH SA，以及一组 SA 用于保护其所传送的通信流的密码算法。我们把通过 IKE SA 建立的 ESP 和 AH SA 称为子 SA（CHILD_SA）。

所有的 IKE 通信由成对的消息组成：请求和响应，我们称其为交换（Exchange）。建立 IKE SA 的第一和第二对消息是 IKE_SA_初始化（IKE_SA_INIT）交换和 IKE_鉴别（IKE_AUTH）交换，接下来的IKE交换是建立_子SA（CREATE_CHILD_SA）交换或者信息（INFORMATIONAL）交换。在通常的情况下，有一个独立的IKE_SA_INIT交换和IKE_AUTH 交换（总共四个消息）来建立 IKE SA 和第一个子 SA。例外的情况下，每类交换都可以有多个。在所有情况下，所有的 IKE_SA_INIT 交换必须在其他类型的交换之前完成，然后是所有的 IKE_AUTH 交换，接着是任意数量和任意次序的 CREATE_CHILD_SA 和 INFORMATIONAL 交换。

第一个 IKE 会话（IKE_SA_INIT）协商 IKE_SA 的安全参数，如密码算法，发送非重复值，进行 Diffie-Hellman 密钥交换。第二个 IKE 会话（IKE_AUTH）鉴别以前的消息，交换标识和证书，建立第一个 AH/ESP 子 SA。这些信息的一部分是被基于 IKE_SA_INIT 交换建立的密钥加密和进行完整性保护的。IKE_SA_INIT 和 IKE_AUTH 交换在 IKEv1 中被称为阶段 1。接下来的交换类型是建立子 SA 的 CREATE_CHILD_SA 交换和信息交换，信息交换删除 SA，报告错误条件，或者完成别的内务处理。一个信息请求没有载荷，只用于检查活跃度。

11.4　SSL/TLS

在互联网上访问某些网站时也许你会注意到在浏览器窗口的下方会显示一个锁的小图标。这个小锁表示什么意思呢？它表示该网页被 SSL/TLS 保护着。SSL/TLS 是传输层的安全协议，它的优点是提供基于进程对进程的（而不是主机对主机的）安全服务。

1994 年 Netscape 开发了安全套接层协议（Secure Socket Layer，SSL），专门用于保护 Web

通信。该协议的第一个成熟的版本是 SSL2.0，该版本基本上解决了 Web 通信的安全问题。Microsoft 公司克服了 SSL2.0 的一些缺陷，发布了 PCT（Private Communication Technology），并在 IE 中支持。SSL3.0 版本增加了对除 RSA 算法之外其他算法的支持和一些新的安全特性，并且修改了前一个版本中存在的一些安全缺陷，于 1996 年发布。1997 年 IETF 发布了TLS1.0（Transport Layer Security 传输层安全协议，也被称为 SSL3.1）草案，同时，Microsoft 宣布放弃 PCT，与 Netscape 一起支持 TLS1.0。1999 年，正式发布了 RFC2246（The TLS Protocol v1.0）。2006 年，发布了 RFC4346（The TLS Protocol v1.1）和 RFC4366（TLS Extensions）以取代 RFC2246，2008 年 8 月发布了 RFC5246（The TLS Protocol v1.2）取代 RFC3268, RFC4346 和 RFC4366。

11.4.1 TLS 的体系结构

TLS 被设计用来使用 TCP 提供一个可靠的端到端的安全服务，为两个相互通信的应用之间提供保密性和数据完整性，协议由两层组成，如图11.26所示。

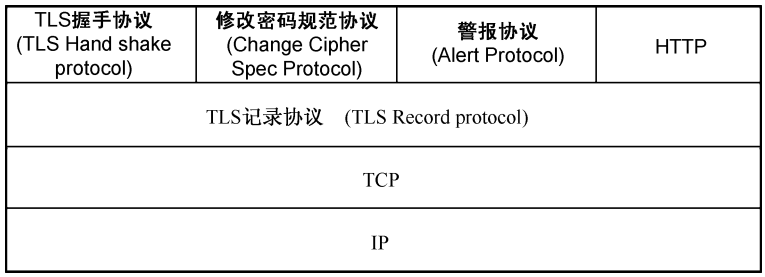


图 11.26 TLS 协议栈

TLS 记录协议（TLS Record Protocol）位于可靠的传输层协议（如 TCP）之上，它使用对称密码算法提供连接的机密性，对称算法的密钥对于每一个连接是唯一的，是由握手协议经过安全协商得到的；它使用带密钥的 MAC 算法保证连接的完整性。TLS 记录协议用于封装上层协议。

TLS 定义了三个高层协议：TLS 握手协议（TLS Handshake Protocol）、修改密码规范协议（Change Cipher Spec Protocol）和警报协议（Alert Protocol），用于对 TLS 交换进行管理。TLS 握手协议允许客户和服务器之间相互鉴别，协商加密算法和密钥，它提供的连接安全性具有三个特点：它使用非对称算法进行身份鉴别，至少对一方实现鉴别，也可以是双向鉴别；协商得到的共享密钥是安全的；协商过程是可靠的。

11.4.2 TLS 的记录协议

TLS 记录协议是一个分层协议。记录协议首先对要传输的消息进行分段处理，然后进行压缩，计算 MAC，最后加密并传输结果。对接收到的数据进行解密、验证、解压缩和重组。

11.4.2.1 TLS 连接状态

TLS 连接状态是 TLS 记录协议的操作环境，它指明了压缩算法、加密算法和 MAC 算法，以及这些算法的参数：MAC 和加密算法在读、写两个方向的密钥。逻辑上有四个连接状态，当前的读、写状态和挂起的读、写状态。所有的记录是在当前的读、写状态下处理的，挂起的读、写状态可以被 TLS 握手协议设置，修改密码规范协议可以选择把任何一个挂起的读、写状态变

成当前状态，当前读、写状态也可以变成挂起状态，挂起的读、写状态也可以被初始化为空状态，初始状态不规定任何加密、压缩和 MAC 算法。

一个 TLS 读、写状态有如下的安全参数：

- **连接端点**(Connection end)：一个连接中的客户端或者服务器端。
- **PRF 算法**：用于从主密钥推导密钥的算法。
- **总体加密算法**(Bulk encryption algorithm)：用于加密的算法，须说明算法的密钥长度、是分组还是流密码，分组密码的分组大小。
- **MAC 算法**：用于消息鉴别的算法，须说明 MAC 算法输出的消息鉴别码的长度。
- **压缩算法**(Compression algorithm)：用于数据压缩的算法。
- **主密钥**(Master secret)：连接双方共享的 48 字节秘密值。
- **客户端随机数**(Client random)：客户端提供的 32 字节数值。
- **服务器端随机数**(Server random)：服务器端提供的 32 字节数值。

记录层使用以上安全参数产生的下列 6 个参数：

- **客户端写 MAC 秘密值**(Client write MAC key)：一个密钥，用来对 client 发送的数据进行 MAC 操作。
- **服务器端写 MAC 秘密值**(Server write MAC key)：一个密钥，用来对server 发送的数据进行 MAC 操作。
- **客户端写密钥**(Client write encryption key)：用于client 进行数据加密，server 进行数据解密的对称秘密密钥。
- **服务器端写密钥**(Server write encryption key)：用于server 进行数据加密，client 进行数据解密的对称秘密密钥。
- **客户端写初始向量**(Client write IV)。
- **服务器端写初始向量**(Server write IV)。

一旦设定了安全参数并产生了相应的密钥，就可以把连接状态设置为当前状态，每一个当前的连接状态包含以下要素：

- **压缩状态**：当前的压缩算法。
- **密码状态**：当前的密码算法。
- **MAC 算法的密钥**。
- **序列号**：每一方为每一个连接的数据发送与接收维护单独的顺序号。当一个连接状态激活时，序号置为 0，最大为 $2^{64}-1$。

11.4.2.2　TLS 记录协议的操作

TLS 协议可以处理任意长度的数据，TLS 记录协议的操作过程如图11.27 所示，具体步骤如下：

第一步，分段，上层消息的数据被分片成 2^{14} (16 384) 字节大小的块或者更小。

第二步，压缩，是可选的，必须使用无损压缩算法对记录进行压缩，对于短的数据块，压缩算法可能导致数据增加，如果数据增加的话，则增加部分的长度不超过 1024 字节。

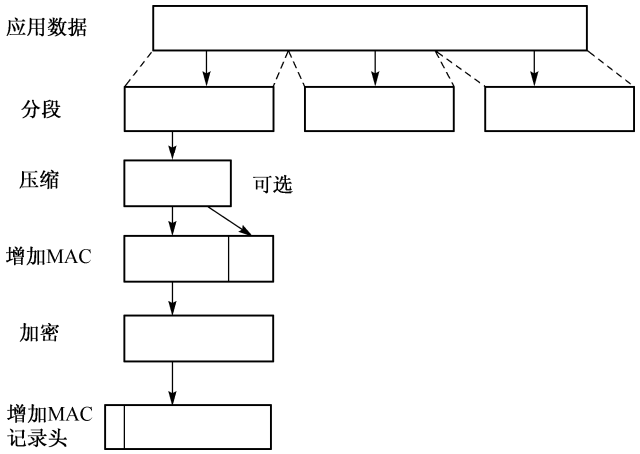


图 11.27 TLS 记录协议的操作过程

第三步，对压缩数据计算消息鉴别码。MAC 的计算公式如下：

MAC(MAC_write_key, seq_num + TLSCompressed.type + TLSCompressed.version + TLSCompressed.length + TLSCompressed.fragment)

这里“+”表示连接，seq_num 表示记录的序列号，TLSCompressed.type 表示此分段的上层协议类型。TLSCompressed.version 表示协议的版本号，该文档是 TLS1.2，使用版本值{3, 3}，是基于历史的原因。TLSCompressed.length 表示压缩分段的长度，不超过 2^{14} + 1024 字节。TLSCompressed.fragment 表示压缩的 TLS 明文分段。可供选择的 MAC 算法有 HMAC-MD5, HMAC-SHA1, HMAC-SHA-256, HMAC-SHA-384 和 HMAC-SHA-512。

第四步，加密，可供选择的加密算法有：128 位的流密码 RC4 与分组密码 3DES_EDE，128 位和 256 位密钥的 AES。分组密码采用 CBC 模式进行计算。

图 11.28 TLS 记录格式

最后一步是加上由如下域组成的 TLS 头。

- **内容类型(Content Type)**：一个 8 位字节，封装上层协议的类型。
- **协议主从版本号**：两个 8 位字节，如 TLS 的版本号为{3, 3}。
- **压缩长度**：16 位，明文段的长度。

TLS 记录格式如图11.28所示。

11.4.2.3 密钥的产生

为了产生记录协议所需要的密钥，使用下面的计算公式：

key_block = PRF(SecurityParameters.master_secret，“key expansion”，SecurityParameters.server_random + SecurityParameters.client_random)；

直到产生足够长的输出。然后从key_block中按以下顺序依次取得相应长度的密钥：客户端写 MAC 密钥、客户端读 MAC 密钥、客户端加密密钥、服务器端加密密钥、客户端写 IV 和服务器端写 IV。

下面给出 PRF(pseudorandom function)的描述，首先，我们定义一个数据扩展函数，P_hash(secret, data)，它使用一个散列函数把一个秘密值和种子值扩展为任意长度的输出。

$$P\_hash(secret, seed) = HMAC\_hash(secret, A(1) + seed) +$$
$$HMAC\_hash(secret, A(2) + seed) + HMAC\_hash(secret, A(3) + seed) + \cdots$$

A( )定义如下：

$$A(0) = seed$$
$$A(i) = HMAC\_hash(secret, A(i-1))$$

P_hash 可以被迭代执行多次，产生需要的足够数据，如使用P_SHA-256 产生 80 字节的数据，需要迭代三次，产生 96 字节的输出，取前 80 字节。

TLS 的 PRF 定义如下：

$$PRF(secret, label, seed) = P\_<hash>(secret, label + seed)$$

这里 label 是一个 ASCII 串。

11.4.3　修改密码规范协议

修改密码规范协议只有一个消息，被当前的连接状态所加密，这个消息只有一个字节，它由服务器或者客户端发送，通知下一个记录将使用新的密码规范和密钥进行保护。它使消息的接收方将把读挂起状态变成到读当前状态，发送方把写挂起状态变成写当前状态。

11.4.4　警报协议

警报消息传递消息的严重状态，警告或者致命的，以及关于警报的描述。致命的警报消息会导致连接立即中断。和其他消息一样，警报消息也是按照当前状态加密和压缩的。

致命的警报消息有：意外的消息、错误的MAC、解压缩失败、记录溢出、握手失败、非法参数、未知的 CA、访问拒绝、解码错误、解密错误、协议版本等。

警告类型的消息有：无证书、不支持的证书、吊销的证书、证书过期、未知证书等。

11.4.5　TLS 的握手协议

握手协议负责协商会话，一个会话由下列项组成：

- **会话标识符(Session Identifier)**：服务器选择的一个任意字节序列，用以标识一个活动或可激活的会话状态。
- **对端证书(Peer Certificate)**：标识服务器的 X.509v3 证书，可为空。
- **压缩算法(Compression Method)**：加密前进行数据压缩的算法。
- **密码规范(Cipher Spec)**：指明用于产生密钥的 PRF，数据加密算法(无或AES等)以及 MAC 算法(如 SHA-1)。还包括其他参数，如散列长度。
- **主密钥(Master Secret)**：客户端与服务器端共享的 48 字节秘密值。
- **可重用标志(Is Resumable)**：一个标志，指明该会话是否能用于产生一个新连接。

这些项用于创建记录层使用的安全参数，多个连接可以通过握手协议的重用特征重用一个会话。

TLS 握手协议包括以下步骤：

- 交换 Hello 消息，商定算法，交换随机数，对会话的重用进行检查。
- 交换必要的密码参数，允许客户和服务器商定预主密钥(Premaster secret)。

- 交换证书和密码信息，允许客户和服务器相互鉴别。
- 从预主密钥产生一个主密钥，交换随机数。
- 为记录层提供安全参数。
- 允许客户与服务器验证对方得到了同样的安全参数，握手协议没有遭到攻击者的攻击。

握手协议的消息处理过程由四个阶段组成，如图11.29所示。

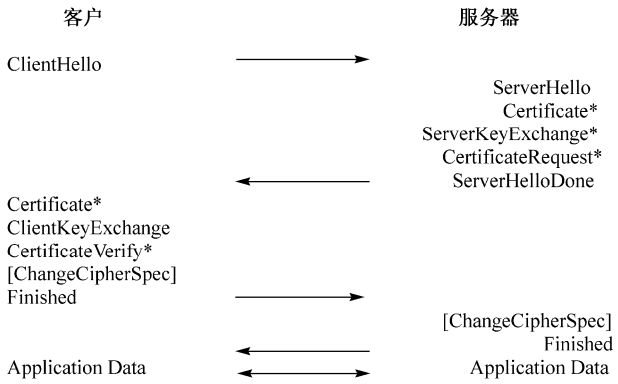


图 11.29　握手消息的处理过程

第一个阶段，建立客户与服务器的安全能力。客户端发送 ClientHello 消息，服务器端必须响应 ServerHello 消息，这组消息建立了如下属性：协议版本、会话 ID、密码算法组（Cipher Suite）、压缩算法，同时产生和交换了两个随机数： ClientHello.random 和 ServerHello.random。

第二个阶段是服务器鉴别和密钥交换阶段，由四个消息组成：Server Certificate, ServerKeyExchange, Certificate Request 和Server HelloDone。如果需要对服务器进行鉴别，服务器就要发送包含它的证书的Server Certificate 消息。接下来是 ServerKeyExchange 消息，如果服务器要求客户发送证书，它可以发送一个这样的请求 CertificateRequest，第二个阶段的最后一个消息是 ServerHelloDone。图中的*表示该消息是一个可选的或者基于条件的消息，不是总是发送的。

第三个阶段是客户端鉴别和密钥交换。如果服务器发送了CertificateRequest消息，客户端必须发送 Certificate 消息。接下来发送 ClientKeyExchange 消息，其内容取决于 ClientHello 和 ServerHello 消息选择的公钥算法。

第四个阶段是完成阶段，客户端发送 ChangeCipherSpec 消息，把商定的 Cipher Spec 从挂起状态转变为当前状态，然后客户用新的算法、密钥和秘密值发送完成（Finished）消息。作为响应，服务器发送自己的 ChangeCipherSpec 消息，然后在新的 Cipher Spec 下发送完成（Finished）消息。

至此，客户和服务器完成了握手，可以发送应用数据。

不管用什么样的公钥交换算法，从预主密钥产生主密钥的算法是相同的：

$$\text{master\_secret} = \text{PRF}(\text{pre\_master\_secret}, \text{“master secret”}, \text{ClientHello.random} + \text{ServerHello.random})\ [0..47];$$

预主密钥的长度可以随着密钥交换算法的不同而不同，但主密钥都是 48 字节。

11.4.6　TLS 的实现

目前，几乎所有操作平台上的Web浏览器（IE, Netscape）以及流行的Web服务器（IIS, Netscape Enterprise Server 等）都支持 TLS 协议。使用该协议便宜且开发成本小。TLS 的自由软件著名

的有 OpenSSL，其函数库是以 C 语言所写成，此软件是以 Eric Young 以及 Tim Hudson 两人所写的SSLeay 为基础发展的，SSLeay 随着两人前往 RSA 公司任职而停止开发。建立了 TLS 安全机制后，只有 TLS 允许的客户才能与 TLS 允许的 Web 站点进行通信，并且在使用 URL 资源定位器时，注意输入的是“https://”，而不是“http://”。

思考和练习题

(1) 在 TCP/IP 的不同层次实现安全分别有什么特点？
(2) TCP/IP 各层常用的安全协议有哪些？
(3) 简述 IPsec 的传输模式和隧道模式。
(4) IPsec 提供了哪些服务？
(5) IPsec 有几种方式同时提供机密性和鉴别？这些方式有什么区别？
(6) TLS 包含哪些协议？
(7) TLS 记录协议有哪些步骤？
(8) TLS 的连接和会话有什么区别和联系？
(9) TLS 提供了哪些服务？

实践/实验题

(1) 在 IIS 中实现 SSL/TLS
　(a) 实验目的：了解证书的内容和 CA，掌握 SSL/TLS 的原理及在 IIS 中的应用。
　(b) 实验环境：安装 Windows 2000 Server 操作系统的计算机以及与其联网的一台 Windows XP 计算机。
　(c) 实验内容和步骤：在 Windows 环境下安装和建立独立根 CA，生成 CA 证书，建立基于 SSL/TLS 与 Web 站点进行通信的安全连接。
(2) 在 Windows 中配置 IPsec
　(a) 实验目的：理解并掌握在 Windows 操作系统中利用 IPsec 配置 VPN 的方法。
　(b) 实验环境：多台 Windows 2000/XP Professinal 的计算机。
　(c) 实验内容和步骤：
　　任务一　配置 Windows 内置的 IPsec 安全策略。
　　任务二　配置专用的 IPsec 安全策略。

第 12 章　防火墙技术及应用

12.1　防火墙概述

12.1.1　防火墙的基本概念

防火墙的原义是指古代人们修建在房屋之间的一道墙，当某一房屋发生火灾的时候，它能防止火势蔓延到别的房屋。这里所谈的防火墙并不是真正用来防火的墙，而是目前最流行、使用最广泛的一种网络安全技术。防火墙是位于两个或多个网络之间，执行访问控制策略的一个或一组系统，是一类防范措施的总称。一个好的防火墙必须具备以下特点：它是不同网络或网络安全域之间信息通过的唯一出入口；只有被授权的合法数据，即防火墙系统中安全策略允许的数据，才可以通过；其自身能不受各种攻击的影响。防火墙的名称来源于这样一种事实，它将网络分隔为不同的物理子网，限制危害从一个子网扩散到另一子网，正如建筑上的防火墙防止火势蔓延一样。

12.1.2　防火墙的作用和局限性

防火墙确保一个单位内的网络与因特网的通信符合该单位的安全方针，为管理人员提供下列问题的答案：谁在使用网络，他们在网络上做什么，他们什么时间使用了网络，他们上网去了何处，谁要上网却没有成功？总体来说，防火墙为我们带来了如下好处：

⑴ 防火墙对企业内部网实现了集中的安全管理，可以强化网络安全策略，比分散的主机管理更经济易行。

⑵ 防火墙能防止非授权用户进入内部网络。

⑶ 防火墙可以方便地监视网络的安全性并报警。

⑷ 可以作为部署网络地址转换(Network Address Translation)的地点，利用 NAT 技术，可以缓解地址空间的短缺，隐藏内部网的结构。

⑸ 利用防火墙对内部网络的划分，可以实现重点网段的分离，从而限制安全问题的扩散。

⑹ 由于所有的访问都经过防火墙，防火墙是审计和记录网络的访问和使用的最佳地点。

防火墙可以提高内部网的安全性，但是防火墙也有它的一些缺陷和不足，主要有：

⑴ 为了提高安全性，限制或关闭了一些有用但存在安全缺陷的网络服务，给用户带来使用的不便。

⑵ 目前防火墙对于来自网络内部的攻击还无能为力。

⑶ 防火墙不能防范不经过防火墙的攻击，如内部网用户通过 SLIP 或 PPP 直接进入 Internet。

⑷ 防火墙对用户不完全透明，可能带来传输延迟、瓶颈及单点失效。

⑸ 防火墙不能有效地防范数据驱动式攻击。

⑹ 作为一种被动的防护手段，防火墙不能防范因特网上不断出现的新的威胁和攻击。

12.1.3　防火墙的安全策略

在构筑防火墙之前，需要制定一套完整有效的安全策略。安全策略包含网络服务的访问策略和设计策略。

防火墙包含着这样一对机制：一种机制是阻拦数据流的通过，另一种机制是允许数据流的通过。防火墙设计策略存在两种情形：一种是“一切未被允许的就是禁止的”，另一种是“一切未被禁止的都是允许的”。第一种的特点是安全性好，但是用户所能使用的服务范围受到严格限制。第二种的特点是可以为用户提供更多的服务，但是在日益增多的网络服务面前，很难为用户提供可靠的安全防护。一些防火墙在二者之间采取折中。

网络服务访问策略是一种高层次的具体到事件的策略，主要用于定义在网络中允许或禁止的服务。具体来说涉及用户账号策略、用户权限策略、信任关系策略、包过滤策略、鉴别策略、签名策略、数据加密策略、密钥分配策略和审计策略。

12.2　防火墙的体系结构

在防火墙与网络的配置上，有以下四种典型结构：包过滤防火墙、双宿/多宿主机模式、屏蔽主机模式、屏蔽子网模式。在介绍这几种结构前，先来了解一下几个概念。

(1) **堡垒主机(Bastion host)**：堡垒主机是一种配置了安全防范措施的网络上的计算机，堡垒主机为网络之间的通信提供了一个阻塞点，也就是说如果没有堡垒主机，网络之间将不能相互访问。

(2) **双宿主机(Dual-homed host)**：有两个网络接口的计算机系统，一个接口接内部网，一个接口接外部网。

12.2.1　包过滤型防火墙

包过滤(Package Filtering)型防火墙往往用一台路由器来实现，如图12.1所示。其基本思想很简单，对所接收的每个数据包进行检查，根据过滤规则，然后决定转发或者丢弃该包，因为一个服务的数据包有流入和流出的，往往配置成双向的。过滤规则基于IP包头信息进行设置，如IP源地址、目的地址、TCP/UDP端口、ICMP消息类型、TCP头中的ACK位。过滤器往往建立一组规则，根据IP包是否匹配规则中指定的条件来做出决定。如果匹配则按规则执行，没有匹配，则按默认策略。有两种基本的默认策略，一种是没有被拒绝的流量都可以通过，管理员必须针对每一种新出现的攻击，制定新的规则；另一种是没有被允许的流量都要拒绝，这种策略比较保守，根据需要，逐渐开放，安全性比较高。因为规则是按一定次序放在规则表中，还需要制定一个匹配原则：首次匹配或末次匹配，即对数据包按过滤规则表中第一条匹配的规则或者最后一条匹配的规则进行处理。

包过滤防火墙的优点是实现简单、效率高、费用低，并对用户透明，其缺点是维护困难，要求管理员熟悉每一种协议，因为其过滤是在网络层进行的，不支持用户鉴别。

图 12.1　包过滤防火墙

12.2.2 双宿/多宿主机模式

双宿/多宿主机防火墙(Dual-Homed/Multi-Homed Firewall)又称为双宿/多宿网关防火墙，它是一种拥有两个或多个连接到不同网络的网络接口的防火墙，通常用一台装有两块或多块网卡的堡垒主机做防火墙，两块或多块网卡各自与受保护网和外部网相连，其配置结构如图12.2 所示。这种防火墙的特点是主机的路由功能是被禁止的，两个网络之间的通信通过应用层代理服务来完成。如果一旦黑客侵入堡垒主机并使其具有路由功能，防火墙将变得无用。

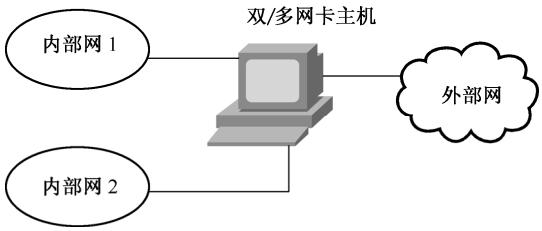


图 12.2 双宿/多宿主机模式防火墙配置

12.2.3 屏蔽主机模式

屏蔽主机防火墙(Screened Host Firewall)由包过滤路由器和堡垒主机组成，其配置如图 12.3 所示。在这种方式的防火墙中，堡垒主机安装在内部网络上，通常在路由器上设立过滤规则，并使这个堡垒主机成为从外部网络唯一可直接到达的主机，这确保了内部网络不受未被授权外部用户的攻击。屏蔽主机防火墙实现了网络层和应用层的安全，因而比单独的包过滤或应用网关代理更安全。在这一方式下，过滤路由器是否配置正确是这种防火墙安全与否的关键，如果路由表遭到破坏，堡垒主机就可能被越过，使内部网完全暴露。

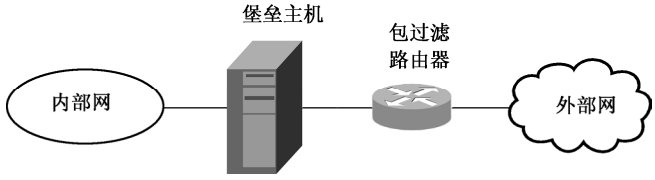


图 12.3 屏蔽主机模式防火墙配置

12.2.4 屏蔽子网模式

屏蔽子网防火墙(Screened Subnet mode)的配置如图12.4 所示，采用了两个包过滤路由器和一个堡垒主机，在内外网络之间建立了一个被隔离的子网，定义为“非军事区(Demilitarized Zone)或者停火区”网络，有时也称为周边网(Perimeter Network)。网络管理员将堡垒主机、Web 服务器、Mail 服务器等公用服务器放在非军事区网络中。内部网络和外部网络均可访问屏蔽子网，但禁止它们穿过屏蔽子网通信。在这一配置中，即使堡垒主机被入侵者控制，内部网仍受到内部包过滤路由器的保护。

实际应用中，防火墙可以根据具体的网络环境配置成其他结构，例如：

- 一个堡垒主机和一个非军事区。
- 合并 DMZ 的内部路由器和外部路由器结构。
- 两个堡垒主机和两个非军事区。

- 两个堡垒主机和一个非军事区。
- 使用多堡垒主机。
- 合并堡垒主机与外部路由器。
- 合并堡垒主机与内部路由器。
- 使用多台内部路由器。
- 使用多台外部路由器。
- 使用多个周边网络。
- 使用双宿主机与屏蔽子网。
- ……

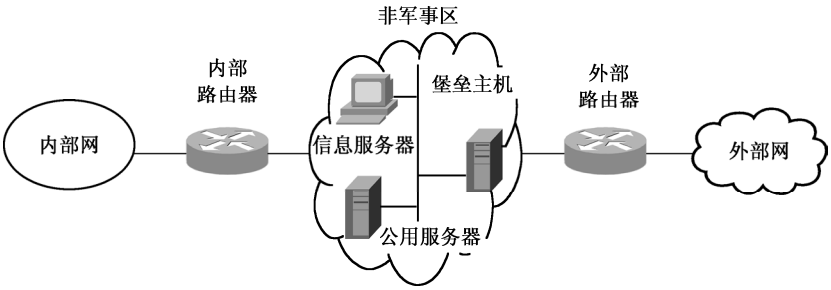


图 12.4　屏蔽子网模式防火墙配置

12.3　防火墙相关技术

通常将现在流行的防火墙划分为两大类型：包过滤型和代理型。包过滤型防火墙又可以分为：静态包过滤型(Static packet filtering)和状态监测型(Stateful inspection)防火墙。代理型防火墙又包括电路级网关(Circuit level gateway)防火墙和应用级网关(Application level gateway)防火墙。

12.3.1　静态包过滤防火墙

12.3.1.1　静态包过滤防火墙的基本原理

静态包过滤防火墙工作在 OSI 模型的网络层或 TCP/IP 协议的IP层，如图12.5 所示。依据系统事先设定好的过滤逻辑，即静态规则，检查数据流中的每个数据包，根据流经该设备的数据包头信息，决定是否允许该数据包通过。创建包过滤规则需要考虑以下这些问题：打算提供何种网络服务，并以何种方向提供这些服务？需要限制任何内部主机与因特网连接的能力吗？因特网上是否有可信任的主机，可以用某种形式访问内部网络吗？

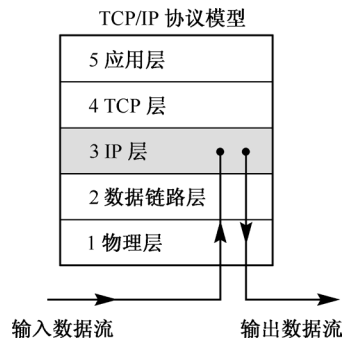


图 12.5　静态包过滤防火墙工作示意图

12.3.1.2　生成过滤规则的信息和约定

1．生成包过滤规则的信息和重要约定

生成包过滤规则的信息有数据包的源地址、目的地址、所用端口号、数据的对话协议及数据包头中的各种标志位等因素，如表 12.1 所示。包过滤根据其中的信息或它们的组合来确定是否允许该数据包通过。对于包过滤防火墙来说，下面的规则是一些必须遵守的重要约定：

- 访问规则要使用 IP 地址，而不使用主机名或域名。
- 不要回应所有经过外部网络接口来的ICMP包。因为任何返回的ICMP数据包都会给攻击者提示一些网络的信息。
- 要丢弃所有通过外部网络适配器流入，且其源地址是来自受保护网络的包。这些数据包显然是伪造的数据包。

表 12.1 生成包过滤规则的信息

信 息 类 型	示 例
数据包协议类型	TCP, UDP, ICMP, IGMP 等
源、目的 IP 地址	162.105.*.*
源、目的端口	FTP, HTTP, DNS 等
IP 选项	源路由、记录路由等
TCP 选项	SYN, ACK, FIN, RST 等
其他协议选项	ICMP Echo, ICMP Echo Reply 等
数据包流向	in 或 out
数据包流经网络接口	eth0、eth1

2. 基于 IP 头信息的过滤规则

IP 头中用于过滤的三种重要信息是 IP 地址、协议类型和 IP 选项。IP 选项的信息如表 12.2 所示。IP 选项用于指示一些特殊的功能。选项中包括一个记录路由的功能，让每个处理数据包的路由器都将自己的地址记录到该包中，时间戳功能让每个路由器在数据包中记录自己的地址和处理包的时间。源路由选项有39字节，3 字节是附加信息，36 字节是地址信息。源路由选项有两个，“**宽松源路由**”指明数据包在发往其目的地的过程中必须经过的一组路由器，而“**严格源路由**”则指定了该数据包只能由列出的路由器处理，并且所经过路由器的顺序不可更改。源路由的工作过程是：取出源路由清单中第一个地址，使它成为目的地址，如果是严格源路由，那么它必须是下一跳。在数据包达到目的地址后，从清单中取出下一个地址，使它变为新的目的地址。对于宽松源路由，在数据包到达清单上指出的地址以前，它经过多少跳是没有关系的。源路由原本是用来调试网络故障和进行其他维护操作的，指定数据包到达目的地应该采纳的路径，黑客会利用源路由选项迫使一个数据包按照特定路由回到自己的计算机，防火墙应丢掉所有打开源路由选项的包。

表 12.2 IP 选项的信息

任 选 类	任 选 号	使 用
0	0	指明是任选表的尾部
0	1	无任选项
0	2	军事使用的安全任选项
0	3	宽松源路由
0	7	激活路由记录
0	9	严格源路由
2	4	时间戳激活

3. 基于 TCP 头信息的过滤规则

IP头主要用于数据包在因特网上的路由。TCP头与连接的可靠性和报文的发送顺序有关。TCP 头中用于生成过滤规则的信息有：端口、SYN 和 ACK 标志位。

SYN 洪泛(SYN Flooding)是一种拒绝服务攻击，其特征是黑客机器向受害主机发送大量伪造源地址的TCP SYN报文，占用受害主机的资源。防火墙可以通过监视网络数据包和日志文件，让不断发送 SYN 置位包的主机不能通过防火墙。

如果允许内部网主机以 telnet 方式访问 Internet，过滤规则如表 12.3 所示。

表 12.3　允许内部网主机以 telnet 方式访问 Internet 的过滤规则

规　则	数据包方向	源　地　址	目 的 地 址	协　议	源　端　口	目 的 端 口	ACK 设置	动　作
A	出	192.168.4.1	任意	TCP	>1023	23	任意	允许
B	入	任意	192.168.4.1	TCP	23	>1023	1	允许

如果不检查 ACK 位，黑客用端口23连接到内部网主机一个大于1023的端口上将是被允许的。这是系统的一个安全缺口。设置 ACK 位后，由于黑客第一个请求包的 ACK 不被置位，该请求包被拒绝；若黑客把第一个请求包ACK置位，该数据可通过防火墙，但目标主机会把它当做以前连接的一个数据包，因为它不属于任何连接(序列号不匹配)，所以被丢弃。

4．基于 ICMP 包的过滤

ICMP 数据可被攻击者用于收集信息，因此需要阻止以下 ICMP 包：流入的 echo 请求和流出的 echo 响应、流入的重定向报文、流出的目的不可达报文和流出的服务不可用报文。

12.3.1.3　过滤规则的设置实例

1．针对 telnet 服务的防火墙规则

telnet 是 Internet 远程登录服务的标准协议和主要方式。它为用户提供了在本地计算机上完成远程主机工作的能力，是基于 TCP 协议之上的应用协议。如果允许内网用户以 telnet 方式访问外网服务器。如图12.6所示，流出的数据包具有如下特性：IP 源地址是内网地址，目标地址为外网服务器地址，TCP协议的目标端口是23，主动发起请求的数据包的源端口号大于1023，发送的第一个数据包的 ACK 位为 0，其余数据包的 ACK 位为 1。流入的数据包具有如下特征：IP源地址是外网服务器的IP地址，目标地址为内网地址，TCP协议的源端口是 23，目的端口与发起请求的数据包端口一致，是一个大于1023的端口号，所有往内的数据包都是响应数据包，其ACK标志为1。根据这些信息可以创建表12.4中的过滤规则 A 和 B。如果允许外网用户以telnet方式访问内网服务器，可以创建过滤规则 C 和D。根据默认的安全策略“一切未被允许的都是拒绝的”可以创建表中的最后一条规则E。这样可以基于首次匹配的原则对数据包进行处理，如果数据的特征符合某一条规则就按相应的规则进行处理，如果数据包和前面任何一条规则都没有匹配上，就根据最后一条规则处理。

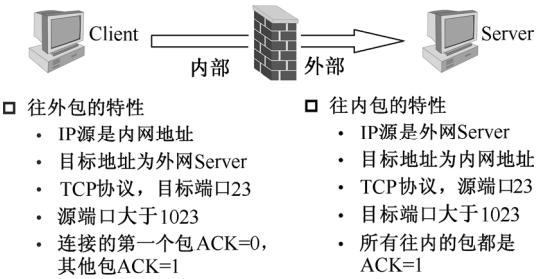


图 12.6　telnet 服务的数据包的特性

表 12.4 针对 telnet 服务的过滤规则

规 则	数据包方向	源 地 址	目 的 地 址	协 议	源 端 口	目 的 端 口	ACK	是 否 通 过
A	出	内部	外部	TCP	>1023	23	任意	允许
B	入	外部	内部	TCP	23	>1023	1	允许
C	入	外部	内部	TCP	>1023	23	任意	允许
D	出	内部	外部	TCP	23	>1023	1	允许
E	双向	任意	任意	任意	任意	任意	任意	拒绝

2．SMTP 数据包的过滤

同样可以给出双向允许的 SMTP 服务的过滤规则，如表 12.5 所示。规则 A 和 B 允许内网用户访问外网的 SMTP 服务，规则 C 和 D 允许外网用户访问内网的 SMTP 服务。

表 12.5 针对 SMTP 服务的过滤规则

规 则	数据包方向	源 地 址	目 的 地 址	协 议	源 端 口	目 的 端 口	ACK	是 否 通 过
A	出	内部	外部	TCP	>1023	25	任意	允许
B	入	外部	内部	TCP	25	>1023	1	允许
C	入	外部	内部	TCP	>1023	25	任意	允许
D	出	内部	外部	TCP	25	>1023	1	允许

3．HTTP 数据包的过滤

如果允许内部网用户访问外网的 WWW 服务，可以创建规则如表 12.6 所示。

表 12.6 针对 WWW 服务的过滤规则

规 则	数据包方向	源 地 址	目 的 地 址	协 议	源 端 口	目 的 端 口	ACK	是 否 通 过
A	出	内部	Internet	TCP	>1023	80	任意	允许
B	入	Internet	内部	TCP	80	>1023	1	允许

4．FTP 数据包的过滤

FTP 是建立在TCP之上的协议，用来把文件从一台计算机传递到另一台计算机。FTP 需要建立两个连接，一个是命令通道，另一个是数据通道。FTP 有两种连接模式：正常模式和 PASV 模式。正常模式要求服务器启动一个连接用于数据传输，服务器端的命令端口为 21，数据端口为 20。PASV 模式两条连接都由客户方启动，服务器端的命令端口为 21，数据端口为一个大于1023的数。若提供出站的 FTP 服务，表12.7给出了正常模式的过滤规则，规则 A 和 B 用于 FTP 命令通道，规则 C 和 D 用于 FTP 数据通道。表 12.8 给出了 PASV 模式的过滤规则，规则 A 和 B 用于 FTP 命令通道，规则 C 和 D 用于 FTP 数据通道。

表 12.7 正常模式的 FTP 过滤规则

规 则	数据包方向	源 地 址	目 的 地 址	协 议	源 端 口	目 的 端 口	ACK	是 否 通 过
A	出	内部	外部	TCP	>1023	21	任意	允许
B	入	外部	内部	TCP	21	>1023	1	允许
C	入	外部	内部	TCP	20	>1023	任意	允许
D	出	内部	外部	TCP	>1023	20	1	允许

表 12.8　PASV 模式的 FTP 过滤规则

规　则	数据包方向	源 地 址	目 的 地 址	协　议	源 端 口	目 的 端 口	ACK	是 否 通 过
A	出	内部	外部	TCP	>1023	21	任意	允许
B	入	外部	内部	TCP	21	>1023	1	允许
C	出	内部	外部	TCP	>1023	>1023	任意	允许
D	入	外部	内部	TCP	>1023	>1023	1	允许

正常模式的 FTP 服务要求服务器方发起主动的连接，用于数据传输，PASV 模式的数据传输要求开放所有大于1023的端口，都存在一定的安全隐患。一般情况下，若采用正常模式，需要经过代理。对于 PASV 模式，如果可以动态监视 FTP 命令通道发出的数据端口，而不是打开全部高编号端口，将会更安全，下面讲到的状态监测技术可以做到这一点。

5．对一些攻击的防范

一些针对静态包过滤防火墙的攻击及其防范措施如下：

- IP地址欺骗，例如，来自外部的攻击者假冒内部的IP地址。可以通过设置过滤规则禁止从外部网络适配器流入的具有内部地址的数据包，阻止此类攻击。
- 源路由攻击，即通过源指定路由的攻击。可以通过设置过滤规则禁止源路由选项打开的数据包阻止此类攻击。
- 小碎片攻击，利用IP 分片功能把 TCP 头部切分到不同的分片中。可以通过设置过滤规则丢弃分片太小的数据包。
- 利用复杂协议和管理员的配置失误进入防火墙，例如，利用FTP协议对内部进行探查。若采用正常模式的 FTP，可以设置为需要经过代理。

12.3.1.4　静态包过滤的特点

包过滤防火墙的优点是逻辑简单，价格便宜，对网络性能的影响较小，有较强的透明性。并且它的工作与应用层无关，无须改动客户机和主机上的任何应用程序，易于安装和使用。

静态包过滤的弱点是：配置基于包过滤方式的防火墙，需要对 IP, TCP, UDP, ICMP 等各种协议有深入的了解，否则容易出现因配置不当带来的问题；据以过滤判别的只有网络层和传输层的有限信息，因而各种安全要求不能得到充分满足；由于数据包的地址及端口号都在数据包的头部，不能彻底防止地址欺骗；允许外部客户和内部主机的直接连接；不提供用户鉴别机制。

12.3.2　状态监测防火墙

静态包过滤防火墙最明显的缺陷是为了实现期望的通信，它必须保持一些端口的永久开放，这就为潜在的攻击提供了机会。为了克服这一弱点，发展出了**动态包过滤技术**，它跟踪客户端口，而不是打开全部高编号端口给外部访问，当一组数据包通过打开的端口到达目的地，防火墙就关闭这些端口。状态监测技术就是在动态包过滤技术的基础上发展而来，是对动态包过滤技术的增强。状态监测防火墙创建通信状态表保存每一连接状态内容，记录外出的TCP 连接及相应的高编号客户端口，以验证任何进入的通信是否合法。规则可动态生成和删除。

著名的网络安全公司 Check Point 第一个研制出基于这种技术的防火墙——Check Point 防火墙-1。这种防火墙采用了一个在网关上执行网络安全策略的软件引擎，称为监测模块。

监测模块工作在链路层和网络层之间，如图 12.7 所示。对网络通信的各层实施监测分析，提取相关的通信和状态信息，并在动态连接表中进行状态及上下文信息的存储和更新，这些表被持续更新，为下一个通信检查提供累积的数据。数据包在网络链路层和网络层操作系统核心中被检查。

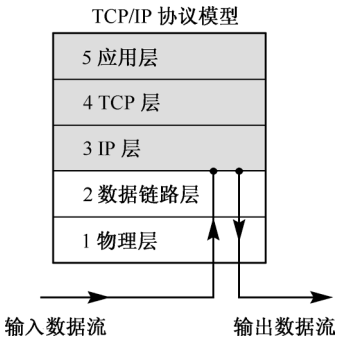


图 12.7 状态监测防火墙工作示意图

状态监测技术的另一个优点是：能够提供对基于无连接的协议(UDP)的应用(DNS, WAIS 等)及基于端口动态分配的协议(RPC)的应用(如 NFS, NIS)的安全支持，静态的包过滤和代理网关都不支持此类应用。总之，这类防火墙减少了端口的开放时间，提供了对几乎所有服务的支持，缺点是它也允许外部客户和内部主机的直接连接；不提供用户鉴别机制。

大多数商业版防火墙产品、个人防火墙和一些路由器都实现了包过滤技术，开放源码的包过滤防火墙软件有：FreeBSD 操作系统中的 Ipfilter 和 Ipfw，Linux 操作系统中的 Ipchains 和 Iptables，Iptables 是运行在 Linux 2.4(或此版本以上)上的防火墙 Ipchains 的升级。

12.3.3 应用级网关防火墙

应用级网关防火墙通常也称为应用代理服务器。它工作于 OSI 模型的应用层，如图 12.8 所示。用来提供应用层服务的控制，起到外部网络向内部网络或内部网络向外部网络申请服务时的转接作用。当外部网络向内部网络申请服务时，内部网络只接受代理提出的服务请求，拒绝外部网络其他节点的直接请求。所有的连接都通过防火墙，防火墙作为网关。代理服务器的工作过程为：首先，它对该用户的身份进行验证。若为合法用户，则把请求转发给真正的某个内部网络的主机，同时监控用户的操作，拒绝不合法的访问。当内部网络向外部网络申请服务时，代理服务器的工作过程刚好相反，如图 12.9 所示。

应用网关代理的优点是易于配置，界面友好；不允许内外网主机的直接连接；在应用层上实现，可以监视数据包的内容，实现基于用户的鉴别，提供比包过滤更详细的日志记录，例如在一个 HTTP 连接中，包过滤只能记录单个的数据包，无法记录文件名、URL 等信息；可以隐藏内部IP地址；可以给单个用户授权；可以为用户提供透明的加密机制；可以与鉴别、授权等安全手段方便地集成。

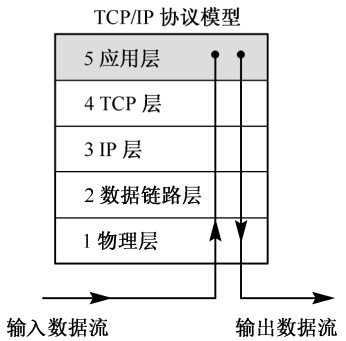


图 12.8 应用级网关防火墙工作示意图

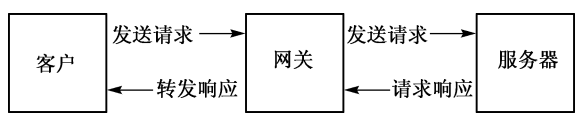


图 12.9 代理服务器的工作过程

代理技术的缺点是：代理速度比包过滤慢；代理对用户不透明，给用户的使用带来不便，而且这种代理技术需要针对每种协议设置一个不同的代理服务器，新的服务不能及时地被代理；客户软件需要修改，重新编译或者配置；有些服务要求建立直接连接，无法使用代理，比如聊天服务或者即时消息服务；代理服务不能避免协议本身的缺陷或者限制。

商业版的代理防火墙产品有 Microsoft Proxy Server，开放源码的代理防火墙软件有：TIS FWTK（Firewall toolkit），Apache 和 Squid。

12.3.4　电路级网关防火墙

本质上，电路级网关也是一种代理服务器，接收客户端连接请求，代理客户端完成网络连接，在客户和服务器间中转发数据，是一个通用代理服务器，它工作于OSI互联模型的会话层或是TCP/IP 协议的 TCP 层，如图12.10所示。它适用于多个协议，但它不能识别在同一个协议栈上运行的不同应用，当然也就不需要对不同的应用设置不同的代理模块，但这种代理需要对客户端做适当修改。它接受客户端的连接请求，代理客户端完成网络连接，建立起一个回路，对数据包起转发作用，数据包被提交给用户的应用层来处理。通过电路级网关传递的数据似乎起源于防火墙，隐藏了被保护网络的信息。

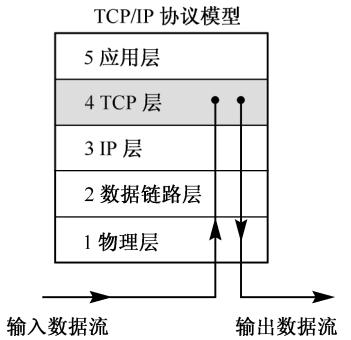


图 12.10　电路级网关防火墙工作示意图

电路级网关为一般的应用提供了一个框架。电路级网关的典型实现是Socks协议，专门设计用于防火墙，在应用层和传输层之间垫了一层。

混合型防火墙结合了包过滤防火墙和应用级代理的特点。如包过滤或状态监测防火墙实现基本的应用代理功能，以提供较好的网络通信审计与用户鉴别。应用代理网关防火墙实现基本的包过滤功能，更好地支持基于 UDP 的应用。

12.3.5　深度包检查技术

与只是检查数据包的包头部分的浅层包检查技术（通常称为状态监测技术）不同，**深度包检查**（Deep Packet Inspection，DPI）也称为**完全包检查**（Complete Packet Inspection）和**信息提取**（Information Extraction）。例如，传统包过滤通过端口号来识别应用类型，当监测到端口号为80 时，则认为该应用代表WWW 服务，而当前网络上的一些非法应用会采用隐藏或假冒端口号的方式躲避检测，此时采用传统检测方法已无能为力了，深度包检查技术的特征就是通过对应用层中的报文内容进行探测，从而确定报文的真正应用。非法应用虽然可以隐藏端口号，但较难以隐藏应用层的协议特征。深度包检查是一种网络包过滤技术，检查通过监测点的数据包的数据部分（也可能是包头），是否违反协议，是否是病毒、垃圾邮件和入侵；或者是一个预先定义的规则，判断是否可以让数据包通过或者它需要被路由到一个不同的目的地；或者用于收集统计信息。深度包检查使高级的安全功能成为可能，如Internet 数据挖掘、监听和审查。当前已经有不少厂商宣布其防火墙产品采用了深度包检查技术，如 Netscreen, Juniper 和 Cisco 等。

DPI 技术组合了入侵检测系统、入侵保护系统及传统状态监测防火墙的功能。这种组合使它能够检测单独的IDS、IPS 或者状态监测防火墙不能检测的攻击行为。状态监测防火墙可

以捕捉到一个数据流的起始和结束，但它不能捕捉到特定应用之外的事件，IDS可以检测到入侵，但却没有很大能力阻止攻击。DPI技术可以被用于阻止线速的病毒、蠕虫、缓冲区溢出攻击、拒绝服务攻击和复杂的入侵等。

DPI 可以检测 OSI 模型从第 2 层到第 7 层的内容，DPI 基于一个从数据包的数据部分提取的特征数据库识别和分类通信流，包头信息可以进一步改善对信息的控制和分类。分类的数据包可以被重定向、标记、阻止和限速，通过这种方式，可以识别和分类不同的 HTTP 错误。许多 DPI 设备支持对数据流的识别，允许基于累积流信息的控制，而不是对逐个数据包的分析。

12.3.6 分布式防火墙

传统防火墙技术存在如下问题或者局限性：传统防火墙依赖于防火墙一端可信，另一端是潜在的敌人；Internet的发展使从外部穿过防火墙访问内部网的需求增加了；一些内部主机需要更多的权限；只依赖于端-端加密并不能完全解决问题；传统防火墙过于依赖物理拓扑结构。此外，考虑到下面几个事实，个人防火墙已得到了广泛的应用，操作系统大多已提供了许多在传统意义上还属于防火墙的手段，IPv6以及IPsec技术的发展，因此，提出了分布式防火墙的概念和技术。

12.3.6.1 分布式防火墙的特点

分布式防火墙的主要特点如下：

(1) 它把防护工作放在主机端，驻留在被保护的主机上，所以也称为“主机防火墙”。该主机以外的网络都认为是不可信任的，因此可以针对该主机上运行的具体应用和对外提供的服务设定针对性很强的安全策略，它使安全策略不仅仅停留在网络与网络之间，而是把安全策略推广延伸到每个网络末端。

(2) 它打破了传统防火墙的物理拓扑结构，不单纯依靠物理位置来划分内外，而是由安全策略来划分内外网。个人防火墙是在分布式防火墙之前出现的一类防火墙产品，用于保护单一主机。分布式主机防火墙与个人防火墙有相似之处，如它们都对应个人系统，但又有本质差别。首先它们管理方式不同，个人防火墙的安全策略由系统使用者自己设置，目标是防止外部攻击，而针对桌面应用的主机防火墙的安全策略由整个系统的管理员统一安排和设置，除了对该桌面机起到保护作用外，也可以对该桌面机的对外访问加以控制，并且这种安全机制是桌面机的使用者不可见和不可改动的。其次，不同于个人防火墙面向个人用户，针对桌面应用的主机防火墙是面向企业级客户的，它与分布式防火墙其他产品共同构成一个企业级应用方案，形成一个安全策略中心统一管理，安全检查机制分散布置的分布式防火墙体系结构。

12.3.6.2 分布式防火墙的结构

典型的分布式防火墙系统结构由三部分组成：

(1) **策略描述语言**：用来说明哪些连接是允许的，哪些连接是禁止的。使用策略描述语言来制定策略，并编译成内部形式存储于策略数据库中。

(2) **一系列系统管理工具**：系统管理工具将策略分发给被防火墙保护的所有主机，各终端主机根据这些策略对数据包进行过滤。

(3) **IPsec 技术及其他高层安全协议**：基于IPsec等安全协议保护主机与主机之间、安全网关与安全网关之间、安全网关与主机之间的路径，保证策略的安全分发。

12.3.7　其他防火墙技术

在选择防火墙产品时，应以其支持的特征集而不是防火墙产品分类为依据。
除了上述主要的技术之外，一个好的防火墙还应该或者可能具有以下功能特点。

- **网络地址翻译**：网络地址翻译的目的一是解决IP地址空间不足问题，二是可以向外界隐藏内部网结构。网络地址翻译的方式可以是一对一的简单的地址翻译，被称为静态地址翻译；也可以是多个内部网地址翻译到一个IP地址的动态地址映射；还可以把多个内部网地址翻译到多个 IP 地址池。
- **热恢复策略或者称为双机热备**：采用某种机制监视主防火墙的响应，在主防火墙出故障时将全部通信转移到备份防火墙。
- **流量控制**：当受保护网络的某个主机的流量超过设定极限时，断开主机的连接。
- **IP 和 MAC 绑定**：把主机的 IP 地址与 MAC 地址绑定后，防止一个主机B假冒另一个主机A的 IP 地址上网。
- **身份鉴别**：在访问网络资源之前，鉴别用户身份，所有网络应用均可进行用户身份鉴别。
- **负载均衡**：逻辑上一个防火墙的功能可以由多个物理服务器分担。
- **内容安全**：在防火墙中可以对内网用户访问WWW页面内容进行访问规则限制，支持基于关键字的监视，实现阻断，保护内部网络。如禁止访问一些非法的、色情的站点。
- **防火墙的免疫设计**：防火墙面临的威胁有攻击者探测系统中装的是何种防火墙，并找出防火墙允许哪些服务通过，采取地址欺骗等手法绕过防火墙的鉴别机制以及寻找利用防火墙系统设计上的安全漏洞。防火墙的自免疫系统可以基于两种途径进行设计，一种是对防火墙的重要文件和数据进行完整性检查，另一种是对防火墙的重要文件和数据访问做详细记录和分析。

12.4　防火墙的实现和维护

防火墙的实现有两种方式，第一种是硬件防火墙，利用 ASIC 技术把算法固化在硬件中，Netscreen 是采用该技术的代表厂家。这种防火墙速度快，国外的高端防火墙大部分采用的是 ASIC 技术。第二种是基于 PC 构架、使用经过定制的通用操作系统的防火墙。这种防火墙可扩展性强，但易受操作系统本身脆弱性影响。为了保证防火墙的安全，应该去掉操作系统中不用的网络协议；去掉或关闭不用的网络服务或应用；去掉或关闭不用的用户或系统账号；及时给操作系统打补丁；去掉或关闭服务器中不用的物理网络接口(网卡)。

防火墙的维护可以有以下两种方式：第一种机制是命令行界面配置。技术熟练的管理员配置防火墙，当出现紧急情况时可以做出快速反应。第二种机制是通过图形用户界面(包括基于Web的配置界面)配置防火墙。图形界面简单、配置快，但配置粒度有限。对于任何一种，必须保证防火墙管理信息的传输安全，因为对防火墙最常用的攻击方法就是利用防火墙的远程管理资源。流行的控制方法有：加密、强用户鉴别及用IP地址限制访问。对于基于 Web 的界面，可使用 SSL 加密、用户 ID 和口令。

12.5 总结和展望

在实际应用中，我们可以根据对安全的需求选择合适的防火墙技术及相应的配置方式。防火墙最早只是提供了一种简单的数据流访问控制策略，随着各项信息技术的迅速发展，新的软件技术、信息安全思想和技术不断地被应用于防火墙的开发上，比如现代密码技术、一次口令系统、智能卡等。随着各种新的安全问题的出现，防火墙的概念和内涵也随着用户的需求而不断丰富，防火墙产品中集成了更多的功能特征，比如支持VPN构建、远程集中管理、对内容的过滤，阻止 ActiveX Java 等小程序，具有一定的病毒检测功能，与一些鉴别及访问控制技术的集成，表现出多功能化。

随着网络带宽的增加，防火墙向着高速的方向发展，应用ASIC, FPGA和网络处理器是实现高速防火墙的主要方法，高性能的算法也是关键。此外，防火墙自身的安全性和稳定性也需要不断发展和增强。

虽然现在个人防火墙已得到了广泛的应用，操作系统大多已提供了许多在传统意义上还属于防火墙的手段，作为构建安全网络环境的第一道屏障，防火墙这一概念还不能抛弃。

思考和练习题

(1) 简述防火墙的作用和局限性。
(2) 静态包过滤、状态监测和深度包检查防火墙有什么区别？
(3) 如何在防火墙上实现多对多的地址转换？
(4) 防火墙的基本体系结构有哪几种？各有什么优缺点？
(5) 分布式防火墙和个人防火墙有什么异同？

实践/实验题

(1) Windows XP 和 Linux 都自带包过滤防火墙，任选一个操作系统环境，利用操作系统提供的文档资料，熟悉其所带防火墙的配置命令，决定需要阻止哪些数据包，设置防火墙过滤这些数据包，并评估其优缺点。
(2) 下载一款个人防火墙(如天网个人防火墙试用版)，熟悉其功能和操作，决定需要阻止哪些数据包，设置防火墙过滤这些数据包，并评估其优缺点。

第 13 章　黑客攻击与防范技术

从信息安全技术体系的角度讲，攻击和评测的理论和实践是对信息系统安全性的考验。俗话说“知己知彼，百战不殆”，对黑客的攻击技术和方法进行深入、详细的了解，才能更有效地对系统提供保护。计算机病毒也属于一种攻击手段，本章不做详细介绍，在后面一章单独介绍。

13.1　认识黑客

人们通常把黑客(Hacker)看做是试图闯入计算机并造成破坏的人。而最早黑客的定义并不是太贬义。它是给予喜欢发现和解决技术挑战，攻击计算机网络系统，并且精通计算机技能的人的称号，与闯入计算机网络系统目的在于破坏和偷窃信息者不同。黑客，源于英语动词 hack，意为“劈，砍”，引申为“干了一件非常漂亮的工作”。在早期麻省理工学院的校园俚语中，“黑客”则有“恶作剧”之意，尤指手法巧妙、技术高明的恶作剧。日本《新黑客词典》把黑客定义为：“喜欢探索软件程序奥秘、并从中增长了其个人才干的人。他们不像绝大多数计算机使用者，只规规矩矩地了解别人指定了解的狭小部分知识”。

黑客不仅仅是在Internet上找麻烦的单独个体，事实上，他们是网上一个活跃的团体。每个团体的成员有不同的命名。飞客(Phreak)指的是早期攻击电话网的青少年，研究各种盗打电话而不用付费的技术。骇客(Cracker)是一个闯入计算机系统和网络试图破坏和偷窃个人信息的个体，与没有兴趣做破坏只是对技术上的挑战感兴趣的黑客相对应。快客(Whacker)是从事黑客活动但没有黑客技能的人，Whacker是穿透系统的人中，在技术和能力上最不复杂的一类。武士(Samurai)是被他人雇用的帮助他人提高网络安全的黑客，武士通常被公司付给薪金来攻击网络。幼虫(Lara)是一个崇拜真正黑客的初级黑客。欲望蜜蜂(Wannabee)是处于幼虫的初始阶段黑客的称呼，他们急于掌握入侵技术，但由于他们没有经验，因此即使没有恶意也可能造成很大危险。黑边黑客(Dark-Side)是指由于种种原因放弃黑客的道德信念而恶意攻击的黑客。半仙(Ddmigod)是一个具有多年经验在黑客团体具有世界级声誉的黑客。红客(Sneaker)指的是有道德的黑客，他们的行为是合法的，比如利用黑客攻击技术对系统的脆弱性进行探测和评估。

13.2　攻击的概念和分类

简单地说，所谓“攻击”，就是指一切针对计算机的非授权行为，攻击的全过程应该是由攻击者发起的，攻击者应用一定的攻击方法和攻击策略，利用一些攻击技术或工具，对目标信息系统进行非法访问，达到一定的攻击效果，并实现攻击者的预定攻击目标。因此，凡是试图绕过系统的安全策略或是对系统进行渗透，以获取信息、修改信息甚至破坏目标网络或系统功能的行为都可以称为攻击。

13.2.1 攻击方式的分类原则

分类法是在对事物特性进行分析的基础上，采用一定的科学方法，按照某种统一的标准，对目标样本进行分离或分组排序的一种理论。通过对样本的分类、排列，可以实现对样本本质特征的概括总结，增强人们对事物的理解。20 世纪 90 年代中后期，关于网络攻击分类方法的原则问题，人们进行了较多的探讨。其中，Amoroso 认为网络攻击分类体系应该具备的原则主要有：

(1) **互斥性**：各类别之间应该是互斥的，没有交叉和覆盖现象。

(2) **完备性(也称穷举性、无遗漏性)**：分类体系能够包含所有可能的攻击。

(3) **确定性(也称无二义性)**：对每一个分类的特点描述精确、清晰。

(4) **可重复性**：不同人根据同一原则对同一个样本进行重复分类的过程，得出的分类结果是一致的。

(5) **可接受性**：分类方法符合逻辑和惯例，易于被大多数人所接受。

(6) **可用性**：分类可用于该领域中的深入调查、研究，对不同领域的应用也具有实用价值。

另外，除了上述 6 条分类原则，还存在其他一些非主流的原则，如适应性、可理解性、稳定性、客观性、原子性等。此外，还存在其他一些更加复杂的原则，如与漏洞的分类方法相近似、所使用的技术术语应当具有准确的定义、对内部攻击和外部攻击应该加以区分等。不过，从已有的分类实践中，可以看出人们在研究分类体系时一般重点关注的原则还是由 Amoroso 所提出的 6 项基本原则。

13.2.2 攻击方式分类方法

关于攻击方式分类方法的研究，国内外的很多学者已经做了大量的工作，提出了众多的分类方式，大概可以分为下述 4 种：基于积累的经验术语、基于攻击行为的单一属性、基于多维攻击属性、基于攻击的具体应用环境。

13.2.2.1 基于经验术语的攻击分类方法

所谓经验术语，是指在网络安全、攻击、维护中所经常用到的技术术语，或者是对一些攻击的社会描述。一个流行且简单的计算机网络安全攻击分类就是有规则地列出一些单个的攻击术语定义。表 13.1 给出了 1995 年 Icove 根据经验给出的攻击方式分类表。1997 年，Cohen 提出了新的攻击方式，包含多达 94 种攻击。

表13.1 Icove 的经验分类列表

病毒	IP 欺骗	资料欺骗	特洛伊木马
蠕虫	口令窃取	拒绝服务	非授权资料复制
侵扰	越权访问	软件盗版	服务干扰
扫描	逻辑炸弹	隐蔽信道	数据扰乱
陷门	攻击隧道	搭线窃听	时间戳攻击
伪装	电磁泄漏	会话劫持	意大利腊肠片攻击

基于经验术语的分类方法虽然出现较早，但当时还很不成熟，在互斥性、完备性、确定性和可接受性等方面均无法得到满足。

13.2.2.2　基于单一属性的攻击分类方法

众所周知，每一种类型的攻击都有各自很多的属性，如攻击基于的操作系统平台、攻击的技术方法和漏洞的利用等。基于单一属性的攻击分类方法就是从其中一个特定的属性对攻击进行描述。Neumann 和 Parker 曾经基于攻击所采用的技术方法这一属性对攻击方式进行了分类，他们在分析了3000 余种攻击实例之后，将攻击分为外部滥用、硬件滥用、伪造、有害代码、旁路、主动滥用、被动滥用、恶意滥用、间接滥用9 类，并进一步将其细化为 26 种具体的滥用攻击，是基于单一属性的攻击分类方法的典型代表。

在信息技术安全评估标准(ITSEC)中提出的著名的 CIA 安全模型也是一种基于单一属性的攻击分类方法，它基于攻击对系统带来的影响这一属性，将影响分为三类：数据机密性、信息完整性和系统可用性。这种简单的分类很好地满足了互斥性要求，各个类型内涵和概念清晰，不容易交叉，但是将所有的攻击后果只分成三类，显得过于简单、单薄，即使一种攻击满足了上述三类中的一类或几类，也无法确定其攻击的本质到底是什么，因此，该分类方法的理论指导意义大于实际意义，可用性不能得到满足。另外，随着攻击技术的发展，现如今，很多攻击已经不能简单地归类到上述三种类型之中，因此，上述分类方法就显得有些过时了，不过这并不影响它的经典性。

Jayaram 和Morse 也曾基于攻击所采用的技术方法提出以下分类方法，将攻击分为如下几类：物理攻击、利用系统漏洞的攻击、恶意代码攻击、窃取权限攻击和基于网络通信的窃听或欺骗攻击等。

基于单一属性的攻击分类方法在确定性和可接受性方面有了很大的进步，已经基本能够保证其定义的清晰程度，可以满足一些场合的需要，但其只能在理论上反映攻击的单一属性和特点，对其他特点过于忽略，且实际应用比较困难，无法广泛采用，因此可用性还有待提高。

13.2.2.3　基于多维属性的攻击分类方法

基于多维属性的攻击分类方法已经比较成熟，在完备性、可理解性和可用性方面都取得了很大的进步，虽然有时候对某些攻击特别是混合攻击进行分类时，确定性和可重复性略显不足，但是对于理解攻击、描述攻击方面已经非常精确，适合用来对大量攻击进行分类整理，是当今主流的攻击分类方法。

Simon Hansman 提出了一种四维攻击分类法，对所采用的四个维度的解释如下：第一维是攻击向量(Attack vector)。攻击向量是指一种攻击为达到其目的所主要采用的攻击方式，其具体类别包括：病毒(Virus)、蠕虫(Worm)、木马(Trojan)、缓冲区溢出攻击(Buffer overflow)、拒绝服务攻击(Denial of service attack)、网络攻击(Network attack)、物理攻击(Physical attack)、口令攻击(Password attack)、信息搜集攻击(Information gathering attack)等。第二维是攻击目标(Attack target)。攻击的目标应该是具体的，明确的，主要可以分为三大类：硬件、软件和网络。为了更加清晰地描述攻击目标，这三大类还可以继续细分，硬件部分包括计算机、网络设施和外围设备，软件部分包括操作系统和应用程序，网络部分是包括网络内部的协议及其拓扑结构。第三维是攻击所利用的漏洞。第三个维度是入侵一般都会用到的是国际漏洞公布组织所公布过的通用漏洞(Common Vulnerabilities and Exposures，CVE)，这个通用漏洞的披露是为了给计算机和网络中一些已发现的漏洞做一个普遍的定义和解释，随着时间的增长，这个通用漏洞的集合也会不断增加。第四维涉及的是攻击自身之外的一些效果，如一个蠕虫可能也会有木马的特性，也就是说某种攻击还具有其他种类的攻击特征，此时，可以用第一

维的另一种攻击进行描述。第四维把攻击分为 5 类：第一维的攻击、信息损坏型、信息泄露型、服务窃取型和颠覆型。第四维的描述实际上是为了人们能够更好地分析混合攻击，现实中有很大一部分攻击都是属于这种类型。除了这四个维度以外，Simon Hansman 还提供了一些其他维度用来对这种分类方式提供补充，这些维度虽然看起来比上述的四个维度更加抽象，但是在精确描述一种攻击上同样有着很大的作用，它们分别是：攻击所造成的伤害(Damage)、攻击所造成的经济损失(Cost)、攻击的传播能力(Propagation)和攻击的防御难度(Defense)，这些维度对攻击的描述都是基于攻击的后果，也就是说只有在一种攻击完全实施之后，才能够精确地判断出它所能造成的伤害或损失，也才能在以上几个维度有精确的描述。

13.2.2.4 基于应用的攻击分类方法

基于多属性的攻击分类方法虽然具有很好的普适性，但是在实际应用中，往往只需要用到攻击的某一部分特性，或者只需要针对特定的网络环境对攻击进行分类，这就要求我们在进行攻击分类时要有不同的侧重点，以实际应用的要求为分类主要依据，因而就产生了基于应用的攻击分类方法。基于应用的攻击分类方法是针对特定的工程应用方式、在特定的系统和网络环境中所采用的对攻击进行分类描述的方法，对于特定的场合有着很大的利用价值，能够对于某些特殊类型的攻击进行更加细化的分析，有利于描述其固有的特点及其中的关键属性，能够很好地满足互斥性和确定性，但由于其往往只是针对特殊类型的攻击或者是特定情况下的攻击，所以并不具有普适性，因而完备性无法得到满足，难以适用于多种应用。

13.2.3 基于多维属性的攻击分类

本节根据攻击所采用的技术手段、攻击的来源、攻击的入口、攻击的目标、攻击的行为、攻击的意图、漏洞的利用、攻击的适用平台等属性对攻击进行分类描述。

13.2.3.1 攻击所采用的技术手段

攻击所采用的技术手段是攻击最为重要的一个内在属性，根据攻击所采用的技术手段可以将攻击分为以下几类：

(1) **信息探测攻击**：这一部分主要包括对目标的踩点和扫描。信息探测是黑客在实际攻击开始前所进行的必要的情报收集工作，攻击者通过这个过程尽可能地了解与攻击目标安全相关的方方面面信息，以便能够集中火力进行攻击，主要类型有地址扫描、漏洞扫描、端口扫描、网络拓扑结构扫描等。

(2) **信息窃听攻击**：信息窃听是指攻击者通过非法手段对系统和网络的活动进行监视，从而获得一些用户隐秘的或与安全相关的信息。目前，信息窃听攻击的主要类型包括电磁泄漏、键击记录、网络嗅探和网络拦截等。

(3) **信息欺骗攻击**：信息欺骗是指攻击者冒充正常的用户通过合法渠道获得系统和网络的访问权，并获取相应关键信息。信息欺骗攻击的主要类型有DNS 欺骗、路由欺骗、IP 欺骗和 ARP 欺骗等。

(4) **信息利用攻击**：信息利用攻击是指攻击者根据已经探测或截获的信息，或相应的网络及系统漏洞进行攻击。攻击者可以利用的漏洞主要分为网络传输和协议的漏洞、系统或应用软件的漏洞以及管理的漏洞。攻击者可以利用网络传输时对协议的信任以及网络传输的漏洞进入系统，也可以利用系统内部或服务进程的 BUG 和配置错误进行攻击，还可以利用各种方式从系统管理员和用户那里诱骗或套取可用于非法进入的系统

信息，包括口令、用户名等。

(5) **拒绝服务攻击**：拒绝服务攻击就是指攻击者利用某种手段，使目标机器停止提供服务或资源访问，是最简单的攻击，也是最彻底的攻击，其意图就是彻底的破坏，这也往往比取得访问权要容易得多。拒绝服务攻击根据其实施原理可以分为以下三种类型：导致异常型、欺骗型、资源耗尽型；根据其实施目标也可以分为三种类型：消耗主机资源、消耗网络资源、针对网络设备。现在流行的拒绝服务攻击方式主要有：SYN 洪泛(SYN Flooding)、UDP 洪泛(UDP Flooding)、泪滴(teardrop)攻击、Land 攻击、Smurf 攻击等。

(6) **数据驱动攻击**：数据驱动攻击是通过向某个程序发送数据，以产生非预期结果的攻击，通常为攻击者给出访问目录系统的权限。数据驱动攻击分为缓冲区溢出攻击、格式化字符串攻击、输入验证攻击、同步漏洞攻击和信任漏洞攻击等。

(7) **信息隐藏攻击**：信息隐藏攻击是指攻击者在完成攻击目标后，通过隐藏技术来消除攻击所留下的蛛丝马迹，避免被系统管理员发现，同时保留隐密的通道，使其以后能够轻易地进入目标系统。信息隐藏攻击包括后门安装、日志清理等。

13.2.3.2　攻击的来源

攻击的来源这一属性主要是为了帮助我们确定攻击者相对于被攻击目标在网络中所处的位置。处于不同位置的攻击者所使用的攻击方式一般是不同的，所达到的攻击效果也是不相同的，按照其来源可以将攻击分为三类：

(1) **来自外部网络的攻击**：主要是指来自于本地局域网以外的网络攻击，一般会经过路由器和网关防火墙，所以攻击成功的难度比较高，往往会依托于网络连接、数据访问或内部服务器所开放的各种应用服务来实施。

(2) **来自内部网络的攻击**：主要是指来源于本地局域网内部网络连接的攻击，这部分攻击由于来自于网络内部，不会被网关防火墙屏蔽，所以危险性更高，也更容易实施，各种网络窃听和嗅探就是属于这类攻击。

(3) **来自于本地主机的攻击**：主要是指在目标主机上直接实施的攻击，这种攻击直接作用于目标主机上，不会受到任何防火墙的拦截，因此危险系数最高，一般用来实施提升用户权限等目的。

13.2.3.3　攻击的入口

网络攻击的入口可以简单地理解为攻击者进入被攻击目标的通道，是进入目标系统的“门户”。攻击入口的分类主要是为了确定系统与外界、内网与外网进行信息交换的接口。从操作系统的角度出发，计算机一般有用户接口、网络协议接口、网络管理接口、设备接口四个接口，所有的攻击都是从这四个接口进入计算机系统的。对这四种接口的详细描述如下：

(1) **用户接口**：是指在操作系统上所安装的各种应用服务系统所提供的与外界进行交互的接口，如 Web 服务、FTP 服务和电子邮件服务等。

(2) **网络协议接口**：指操作系统能够与外界进行网络通信所提供的接口，以及为了保证内网能够与外网通信所提供的接口，主要是指 TCP/IP 协议族中 IP 层以下所提供的各种服务。

(3) **网络管理接口**：指操作系统在基本网络通信配置基础上，增加网络管理功能时，所涉及的网络管理、配置模块与外界的接口，例如 SNMP 管理模块等。

(4) **设备接口**：指操作系统与各外围设备之间进行通信所提供的接口，特别指外围设备驱动程序。

13.2.3.4 攻击的目标

攻击的目标应该是具体的，明确的，能够被攻击，且能够对目标所属的系统造成直接或间接的伤害。借鉴已有的攻击分类，并加以扩展，可以将攻击的目标分为四大类：

(1) **对硬件资源的攻击**：这部分攻击的目标主要是针对于系统的硬件资源，例如主机的硬盘和内存、网络设备以及一些外围设备等。所谓的攻击并不是指毁坏这些硬件，这对于网络攻击来说是几乎不可能完成的，属于物理攻击的范畴。所谓对硬件资源的攻击是指对目标硬件资源的盗用，即在不合法的情况下使用额外的内存、CPU供自己进行运算，使得合法用户在需要这些资源时无法获得使用权等。

(2) **对程序的攻击**：这部分攻击的目标主要是针对于系统上所运行的一些程序，不仅包括应用程序，还包括操作系统程序。

(3) **对网络的攻击**：对网络的攻击针对的主要是内部网络、局域网以及VPN等，其攻击目标主要是网络协议，针对的是这些网络协议的漏洞，如 TCP/IP 协议、ARP 协议等。

(4) **对数据信息的攻击**：所谓数据信息是指各网络节点中存放的、对外提供服务的或者是网络上流动交换的数据。对数据信息的攻击主要是指对这些信息进行监听、窃取和修改等。

13.2.3.5 攻击的行为

攻击的行为这一属性主要是用来描述攻击者在攻击的过程中所进行的操作，在复合攻击中往往会用到多种攻击行为，利用攻击行为可以更好地分析攻击的技术实现过程。一般情况下，攻击的行为可以分为以下这些类别：扫描、探测、迂回、欺骗、洪泛、鉴别、盗窃、读取、修改、覆盖、删除、溢出等。利用这些简单的攻击行为往往可以构成很复杂的攻击行为。

13.2.3.6 攻击的意图

攻击的意图也可以称为攻击的结果，它主要反映了攻击者之所以要对系统进行攻击所要达到的目的，是表述攻击对目标系统造成影响的关键指标。根据攻击对目标系统所要造成的实质性的后果，可以将攻击分为 5 类：

(1) **获取信息**：通过攻击行为使目标设备的系统信息、数据信息、用户信息被窃取或被复制，但数据和信息的完整性并没有遭到破坏的情况称为获取信息。如实际攻击发起前所进行的扫描系统、服务等用户信息或探测防火墙安全规则等操作。

(2) **篡改信息**：通过攻击行为使目标设备的系统信息、数据信息、用户信息被破坏、修改甚至删除，数据和信息丧失了完整性和可用性的情况称为篡改信息。例如很多木马对注册表信息的修改以及病毒对系统文件的删除或修改等。

(3) **获取权限**：通过攻击行为，使攻击者拥有了对目标设备的访问权限，并且能够部分利用系统的服务，作为其进一步攻击的手段，这种情况称为获取权限。大部分针对个人计算机的攻击都是以获取权限为主要目标的。

(4) **提升权限**：众所周知，操作系统对用户的权限分成了很多级别，不同的用户能够行使的权限是不同的。为了能够利用更多的系统资源，利用更多的服务，甚至能够增加自己需要的服务，就必须使原本所拥有的访问权限得到提升，这种情况称为提升权限。

操作系统中，很多的缓冲区溢出漏洞都能帮助我们获取权限和提升权限。

(5) **拒绝服务**：通过攻击行为，使目标设备的系统资源被滥用，无法对外界提供正常的服务或无法正常运行，最终导致系统功能丧失，这种情况称为拒绝服务。这种攻击是最为常见的攻击，虽然技术含量不高，但是如果出于某种特殊的目的或是针对特定的服务器，还是很有可能造成较大的经济损失的。

13.2.3.7　漏洞的利用

基于漏洞的利用所进行的攻击分类能够从原理上更清晰地说明攻击成功所采用的技术方式。在根据漏洞分类攻击时，首先应以CVE库中对漏洞攻击的描述作为基准。CVE的英文全称是 Common Vulnerabilities & Exposures，即公共漏洞和风险，CVE 为已经暴露出来的信息安全漏洞或风险给出一个确定的、唯一的、公共的名称，其功能就好像是一个字典表。同时，CVE 还给每一个已经发现的漏洞或风险一个标准化的描述，任何完全迥异的漏洞库都可以用同一个语言表述，这样可以帮助用户在各自独立的漏洞数据库中和漏洞评估工具中共享数据。长久以来，CVE已经成为了安全信息共享的“关键字”。所以，对于攻击工具分类时，要想研究其所利用的漏洞，就应该首先给出该漏洞的CVE漏洞号，然后就可以快速地在任何其他CVE兼容的数据库中找到相应的信息。

如果所利用的漏洞比较新颖，在CVE库中还没有出现，可以根据漏洞的来源进行分类描述，一般参照 Howard 所提出的三种类型进行分类：

(1) **设计上的漏洞**：即在程序的设计过程中已经存在的漏洞和不完善的风险。

(2) **实现上的漏洞**：即在程序的实现过程中，因为代码编写不严谨，没有严格依照设计要求，或者是没有足够强大的输入验证和访问验证，竞争条件错误等。

(3) **配置上的漏洞**：即在软件设备的配置过程中，进行了错误的配置，使用了不够完善的默认，或者是没有设置足够完善的安全策略，以及软件安装错误等。

13.2.3.8　攻击的适用平台

攻击的适用平台，顾名思义，是指攻击主要是针对哪些操作系统平台展开的，这些攻击工具在哪些操作系统平台才能够正常运行，并达到相应的目的，对哪些操作系统平台能够产生威胁等。因此，其分类标准就是不同的操作系统平台，当然了，每一大类的操作系统中还可以分为不同的型号，以达到精确描述的目的。同时，需要注意的是，对于某些攻击来说，特别是某些针对网络协议漏洞的攻击，可能会适用于多个平台，会跨越多个类别，不过，具体到攻击工具上来说，由于不同类型的操作系统运行可执行文件的方式不同，所以一般不会出现跨平台的工具。根据攻击的适用平台可以将攻击大致分为三类：Windows系列、Linux 系列和其他操作系统。

13.3　信息收集技术

一个完整的攻击过程，首先是信息收集攻击，在获得对系统的访问权力后，进一步获得系统超级用户的权力，接下来利用获得的权力和系统的漏洞，可以修改文件，运行任何程序，留下下次入侵的缺口，或破坏整个系统。之后，入侵者会消除自己的入侵痕迹。

黑客在攻击之前需要收集信息，才能实施有效的攻击，信息收集是黑客攻击的第一步，也是一把双刃剑，管理员也可以利用信息收集技术来发现系统的弱点。对于攻击者来说，他

想知道的信息有：域名、经过网络可以到达的 IP 地址、每个主机上运行的 TCP 和 UDP 服务、系统体系结构、访问控制机制、系统信息(用户名和用户组名、系统标识、路由表、SNMP 信息等)和其他信息，如模拟/数字电话号码、鉴别机制等。信息收集的步骤可以分为 6 步：找到初始信息、找到网络的地址范围、找到活动的机器、找到开放端口和入口点、弄清操作系统与针对特定应用和服务的漏洞扫描。

13.3.1 初始信息的收集

13.3.1.1 域名

互联网上的网站无穷多，要记住每个网站的IP 地址是不可能的，为了方便记忆，就产生了域名系统DNS(Domain Name System)，DNS 是一个全球分布式数据库系统，在基于 TCP/IP 的网络中，用于主机名和 IP 地址间的相互转换。对于每一个 DNS 节点，包含有该节点所在机器的信息、邮件服务器的信息、主机 CPU 和操作系统等信息。Bind 是一款开放源码的 DNS 服务器软件，Bind 由美国加州大学 Berkeley 分校开发和维护，全名为 Berkeley Internet Name Domain，它是目前世界上使用最为广泛的 DNS 服务器软件，支持各种 UNIX 平台和 Windows 平台。在 Windows 操作系统中，除了DNS，微软公司还沿用另一套名字系统：NetBIOS 名字系统。NetBIOS 是介于会话层与表示层之间的一个协议。NetBIOS 名是一种非层次的名字空间，要求在整个网络中必须唯一。网络邻居和net 命令使用这种命名空间。在 Windows 系统中，名字解析过程大致如下：对于明显的非 NetBIOS 名(长度大于 15 个字符或名字中包含“.”号)，使用域名解析方式，先查询 HOSTS 文件，再查询 DNS 服务器；如果长度不大于 15 个字符或名字中不包含“.”号，系统假定其为 NetBIOS 名，使用 NetBIOS 方式处理。

13.3.1.2 公开的信息

公开的信息可以从目标机构的网页入手，上面也许会泄露一些信息，例如，机构的位置、联系人电话号码、姓名和电子邮件地址、所用安全机制及策略、网络拓扑结构；从一些新闻报道和出版发行物中也可以获得信息，例如，A 公司采用 X 系统，B 公司发生安全事件(多次)；在新闻组或论坛上，也可能从管理人员的求助信息中发现想要的信息；通过搜索引擎可以获取到大量的信息。这样的信息可以合法地获取。

此外，还可以通过一些非网络技术的探查手段。如**社会工程攻击**，它是一种利用“社会工程学”来实施的网络攻击行为。**社会工程学**是一种利用人的弱点，如人的本能反应、好奇心、信任、贪便宜等弱点进行诸如欺骗、伤害等危害手段，获取自身利益的手法。如通过一些公开的信息，获取支持人员的信任，假冒网管人员，骗取员工的信任达到安装木马、修改口令等目的。

13.3.1.3 whois

简单来说，whois就是一个用来查询已经被注册域名的详细信息的数据库(如域名所有人、域名注册商、域名注册日期和过期日期等)。它是一种客户/服务器结构，客户端发出请求，接收结果，并按格式显示到客户屏幕上；服务器端建立数据库，接受注册请求，提供在线查询服务。UNIX 系统自带 whois 程序，Windows 也有一些 whois 的查询工具，也可以直接通过 Web 查询。

如 http://ewhois.cnnic.cn/是中国互联网中心提供的国内(cn)域名查询网址，http://www.nic.edu.cn/cgi-bin/reg/otherobj提供edu.cn域名的查询。通过这些查询可以得到的信息包括：注册机

构的相关信息、域名相关的信息、管理方面联系人的信息、记录创建和更新的时间、主 DNS 服务器和辅 DNS 服务器。

在 Windows 系统中，要执行 whois 命令就需要一个第三方的工具Sam Spade，Sam Spade 提供了一个友好的 GUI 界面，能方便地完成多种网络查询任务，它开发的本意是用于追查垃圾邮件制造者，但也能用于其他大量的网络探测、网络管理和与安全有关的任务，包括 ping, nslookup, whois, dig, traceroute, finger 等命令功能，是一个网络工具集。

13.3.1.4　一些查询工具

1．Nslookup

Nslookup 是一个功能强大的客户程序，使用 Nslookup 可查到域名服务器地址和 IP 地址。Nslookup 分两种运行模式，一种是通过命令行提交命令的非交互方式，如在系统提示符下输入命令 nslookup cs.pku.edu.cn，返回如下信息：

```
C:\>nslookup cs.pku.edu.cn
Server:  pkuns.pku.edu.cn
Address:  162.105.129.x

Non-authoritative answer:
Name:    cs.pku.edu.cn
Address:  162.105.203.x
```

也就是说 cs.pku.edu.cn 的域名服务器是 pkuns.pku.edu.cn，其 IP 地址为 162.105.129.*x*，cs.pku.edu.cn 的 IP 地址是 162.105.203.*x*。

Nslookup 的另一种运行模式是交互方式，可以访问DNS 数据库中所有开放的信息。在系统提示符号下输入命令就可以启动 Nslookup，在命令提示符号下输入 help 可以得到 Nslookup 命令的在线帮助。如下所示：

```
C:\>nslookup
Default Server:  pkuns.pku.edu.cn
Address:  162.105.129.x

> help
```

2．Ping

Ping（Packet InterNet Groper）命令发送 ICMP Echo Request 消息，等待 Echo Reply 消息，可以确定网络和外部主机的状态，可以用来调试网络的软件和硬件。对发送的每一个包，显示响应的输出，并计算网络来回的时间，最后显示统计结果——丢包率。

Ping 有许多命令行参数，可以改变默认的行为。可以输入命令 ping /?查看该命令的使用帮助。不能 ping 成功的原因有：主机没有处于活动状态，网卡没有配置正确，ICMP Echo Request 和 Echo Reply 包被防火墙阻止掉了。也可以尝试增大 TTL 值重新发送。得到 IP 地址的简单方法是 Ping 域名。“ping 127.0.0.1”命令在本机上做回路测试，用来验证本机的 TCP/IP 协议簇是否被正确安装。

3．Finger

如果目标机器的TCP 79 端口开放了Finger 服务，很容易得到系统中的用户信息，UNIX

系统下，直接输入“Finger”命令，会得到当前注册登录到本系统的用户信息，Windows 系统下，用 telnet 建立到目标服务器TCP 79端口的连接。Finger 命令格式为“finger[选项]用户名@主机名”，运行Finger命令后会显示系统中某个用户的用户名、主目录、停滞时间、登录时间、登录 shell 等信息。

13.3.2 网络地址范围的探查

13.3.2.1 网络地址范围的探查的命令

网络地址范围的探查可以使用 Ping 和 Traceroute 命令。UNIX 系统下的 Traceroute 可以用来发现实际的路由路径。它的基本原理是给目标的一个无效端口发送一系列 UDP 包(默认大小为38字节)，UDP 包的端口设置为一个不太可能用到的值，默认值为33 434，因此，目标会送回一个 ICMP Destination Unreachable 消息，指示端口不可达。发送的UDP包的 TTL(Time to live，生存周期)字段从 1 开始递增，中间路由器会返回一个 ICMP Time Exceeded 消息。Traceroute 有一些命令行参数，可以改变默认的行为。Traceroute 可以用来发现到一台主机的路径，为勾画出网络拓扑图提供最基本的依据。Traceroute 的最后一跳是目的地址。Windows 平台上对应的命令为 tracert，Traceroute 允许指定宽松的源路由选项，不过，许多防火墙是禁止带源路由的包的。

在 UNIX 和 Windows 系统中，分别使用-g 和-j 选项表示宽松源路由，命令格式分别为：

```
Traceroute -g 10.10.10.5 10.35.50.10
Tracert -j 10.10.10.5 10.35.50.10
```

Windows 系统中的 Tracert 默认情况下使用的是 ICMP Echo Request 包。例如：

```
C:\>tracert 202.108.x.x

Tracing route to 202.108.x.x over a maximum of 30 hops:

  1    25 ms    19 ms    24 ms  114.245.x.x
  2    18 ms    17 ms    17 ms  125.35.65.77
  3    18 ms    16 ms    17 ms  bt-227-117.bta.net.cn [202.106.227.117]
  4    16 ms    22 ms    17 ms  bt-228-069.bta.net.cn [202.106.228.69]
  5    19 ms    17 ms    17 ms  61.148.157.237
  6    18 ms    18 ms    17 ms  61.148.143.26
  7    22 ms    18 ms    17 ms  210.74.176.138
  8    17 ms    17 ms    18 ms  202.108.x.x

Trace complete.
```

Pathping 是Windows 2000 中的一个新工具，结合了Ping 和Tracert 的特点，可以统计分组丢失率、RTT(Round Trip Time)时间等。

13.3.2.2 对抗初始信息收集的对策

对抗初始信息收集的对策有：采用防火墙，设置相应的过滤规则，如禁止 ICMP 的 Echo Request 和 Echo Reply 包；使用网络入侵检测系统 NIDS(Network Intrusion Detection System)，发现并阻止对主机和网络的探测行为；使用其他工具，如 Rotoroutor，它可以记录外来的 Traceroute 请求，产生虚假的应答。

13.3.3　查找活动的机器

13.3.3.1　扫描技术概述

扫描技术基于TCP/IP 协议，对各种网络服务，无论是主机或者防火墙、路由器都适用。扫描技术是一把双刃剑，安全管理员可以用来确保自己系统的安全性，黑客用来探查系统的入侵点。端口扫描的技术已经非常成熟，目前有大量的商业、非商业的扫描器。扫描技术分为如下四类：主机扫描、端口扫描、操作系统辨识和漏洞扫描。**主机扫描**能够发现系统存活情况，确定在目标网络上的主机是否可达，同时尽可能多映射目标网络的拓扑结构，主要利用ICMP 数据包。**端口扫描**用于发现远程主机开放的端口，也就是发现哪些服务在运行。**操作系统指纹扫描**根据协议栈判别操作系统。**漏洞扫描**能够暴露网络上潜在的脆弱性，避免遭受不必要的攻击。

13.3.3.2　传统的主机扫描技术

主机扫描分为传统主机扫描技术和高级主机扫描技术。传统的主机扫描技术利用 ICMP 的请求/应答报文。主要有以下四种。

(1) 通过发送一个 ICMP Echo Request 数据包到目标主机，如果接收到 ICMP Echo Reply 数据包，说明主机是存活状态。如果没有收到 Echo Reply，就可以初步判断主机没有在线或者使用了某些过滤设备过滤了 ICMP 的 Echo Reply 消息。可以使用 Ping 命令。

(2) 使用ICMP Echo Request 轮询多个主机称为 Ping 扫描(ICMP Sweep 或者 Ping Sweep)，对于小的或者中等网络使用这种方法来探测主机是一种比较可接受的行为，但对于一些大的网络如 A, B 类子网，这种方法就显得比较慢，因为 Ping 在处理下一个命令之前将会等待正在探测主机的回应。UNIX 下有一些并行处理工具，处理效率较高，如 fping 和 Nmap，Windows 系统同样功能的工具有 pinger。

(3) Broadcast ICMP(ICMP 广播)，通过发送 ICMP Echo Request 到广播地址或者目标网络地址可以简单地反映目标网络中活动的主机，这样的请求会广播到目标网络中的所有主机，所有活动的主机都将会发送 ICMP Echo Reply 到攻击者的源 IP 地址。这种技术的主机探测只适用于目标网络的 UNIX 主机。

这三种方法的缺点是会在目标主机的 DNS 服务器中留下攻击者的 LOG 记录。因为从域名到 IP 地址的转化需要查询该 DNS 服务器。

(4) 其他 ICMP 消息类型(13 和 14, 15 和 16, 17 和 18)也可以用于对主机或网络设备，如路由器等的探测。例如，ICMP Time Stamp Request 和 Reply 允许一个节点查询另一个节点的当前时间，返回值是自午夜开始计算的毫秒数。发送者可以初始化标识符和序列号，请求端还填写发起时间戳，然后发送报文给应答系统。应答系统接受请求后，填写接受和传送的时间戳，把信息类型改变为 Reply 应答并送回给发送者。返回值是自午夜开始计算的毫秒数。在处理时间戳请求时应该最小化可变性延迟(Minimum Variability Delay)。接收主机必须“回答”每个它接收到的时间戳请求。请求是一个IP 广播地址或者 IP 多播地址的 ICMP 时间戳请求可以丢弃。在 ICMP 时间戳回应信息里的 IP 源地址必须与响应的时间戳请求信息的源地址相同。如果在时间戳请求中接收到源路由选项，返回路由必须保留并为时间戳返回选项(Timestamp Reply Option)设置源路由。

13.3.3.3 高级的主机扫描技术

利用被探测主机产生的 ICMP 错误报文可以进行复杂的主机探测，主要有以下几种方式。

(1) **异常的 IP 包头**：向目标主机发送包头错误的 IP 包，目标主机或过滤设备会反馈 ICMP Parameter Problem Error 信息。常见的伪造错误字段为 Header Length 和 IP Options。不同厂家的路由器和操作系统对这些错误的处理方式不同，返回的结果也不同。

(2) **IP 头中设置无效的字段值**：向目标主机发送的 IP 包中填充错误的字段值，比如协议项填一个没使用的超大值，目标主机或过滤设备会反馈 ICMP Destination Unreachable 信息。

(3) **错误的数据分片**：当目标主机接收到错误的数据分片(如某些分片丢失)，并且在规定的时间间隔内得不到更正时，将丢弃这些错误数据包，并向发送主机反馈 ICMP Fragment Reassembly Time Exceeded 错误报文。

(4) **通过超长包探测内部路由器**：若构造的数据包长度超过目标系统所在路由器的 PMTU 且设置禁止分片标志，该路由器会反馈 Fragmentation Needed and Don't Fragment Bit was Set 差错报文。

如果我们不能从目标得到Unreachable报文或者分片组装超时错误报文，可以做下面的判断：防火墙过滤了我们发送的协议类型；防火墙过滤了我们指定的端口；防火墙阻塞 ICMP 的 Destination Unreachable 或者 Protocol Unreachable 错误消息；防火墙对我们指定的主机进行了 ICMP 错误报文的阻塞。

(5) **用UDP扫描**：向目标主机特定端口发送一个 0 字节数据的 UDP 包，关闭端口会反馈 ICMP Port Unreachable 错误报文，而开放的端口则没有任何反馈。通过多个端口的扫描，可以探测到目标系统的存活情况。

(6) **反向映射探测**：用于探测被过滤设备或防火墙保护的网络和主机。构造可能的内部 IP 地址列表，并向这些地址发送数据包。当对方路由器接收到这些数据包时，会进行IP识别并路由，对不在其服务的范围的 IP 包发送 ICMP Host Unreachable 或 ICMP Time Exceeded 错误报文，没有接收到相应错误报文的 IP 地址可被认为在该网络中。

对主机探测的工具非常多，比如著名的nmap, netcat, superscan，以及国内的 X-Scanner，等等。

13.3.3.4 防范主机扫描的对策

主机扫描技术大多使用了 ICMP 数据包，基本的防范对策有：使用可以检测并记录 ICMP 扫描的工具，使用入侵检测系统，在防火墙或路由器中设置允许进出自己网络的ICMP分组类型。

13.3.4 查找开放端口和入口点

13.3.4.1 端口扫描基础

端口扫描的直接成果就是得到目标主机开放和关闭的端口列表，这些开放的端口往往与一定的服务相对应，通过这些开放的端口，黑客就能了解主机运行的服务，然后就可以进一步整理和分析这些服务可能存在的漏洞，随后采取针对性的攻击。

端口扫描建立在 TCP/IP 协议基础之上，在 TCP/IP 的实现中，一般遵循以下原则：

- 当一个 SYN 或者 FIN 数据包到达一个关闭的端口，TCP 丢弃数据包同时发送一个 RST 数据包。

- 当一个 SYN 数据包到达一个监听端口时，正常的三阶段握手继续，回答一个 SYN|ACK 数据包。
- 当一个 SYN|ACK 或者 FIN 数据包到达一个监听端口时，数据包被丢弃。
- 当一个 SYN|ACK 或者 FIN 数据包到达一个关闭端口时，数据包被丢弃，并返回一个 RST 数据包。
- 当一个包含 ACK 的数据包到达一个监听或者关闭的端口时，数据包被丢弃，同时发送一个 RST 数据包。
- 当一个 SYN 位关闭的数据包到达一个监听端口时，数据包被丢弃。

13.3.4.2　端口扫描的类型

已知的端口扫描的类型包括：开放扫描(Open Scanning)、半开扫描(Half-Open Scanning)和秘密扫描(Stealth Scanning)，具体分类如表 13.1 所示。

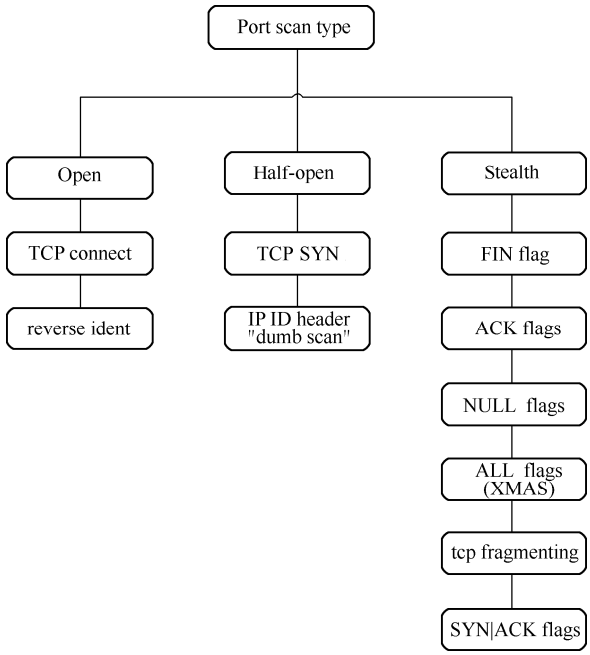


图 13.1　常见端口扫描技术分类

13.3.4.3　开放扫描

开放扫描需要扫描方通过三次握手过程与目标主机建立完整的 TCP 连接，可靠性高，但会产生大量审计数据，容易被发现。开放扫描的方法有：TCP Connect 扫描和 TCP 反向探测(Reverse ident)扫描。

1．TCP Connect 扫描

TCP Connect 扫描使用 Socket 函数 connect()来探测对方的端口是否是活动的。如果 connect()返回成功，则说明该端口是开放的，否则该端口就是关闭的。该扫描方法的优点在于非常易于实现，只需使用系统调用即可完成，而且不需要用户有特殊的权限。另外一个优点就是扫描结果准确。其缺点是无法实施源地址欺骗，因为成功连接的第三步还需要根据服务器发送的序列号来发送 ACK 标志的数据包。很容易被对方的入侵检测系统或防火墙检测

到。

2．TCP 反向探测扫描

该方法主要是基于鉴别协议(Identification Protocol)，参见 RFC1413，该协议提供了一种根据 TCP 连接的源、目的端口来得到所有者身份的方法。该应用基于TCP协议，服务器在 113 端口监听连接信息。当一个连接建立时，服务器从发送来的信息中读取一行，该行指明了客户端感兴趣的 TCP 连接。如果该连接确实存在，服务器会把该连接的所有者信息发送给客户端机器，然后服务器可以关闭该连接，也可以继续等待下一次查询。例如可以先连接到 80 端口，然后通过 ident 来发现服务器是否在root下运行。建议关闭 ident 服务，或者在防火墙上禁止，除非是为了审计的目的。这种方法只能在和目标端口建立了一个完整的 TCP 连接后才能看到。

13.3.4.4 半开扫描

与开放扫描相反，**半开扫描**方法并不使用完整的 TCP 三次握手来进行连接尝试。根据发送的初始 TCP 包的标志位的不同，存在多种半开扫描方式，如TCP SYN扫描和IP ID头扫描(IP ID Header Scanning)。

1．TCP SYN 扫描

扫描器向目标主机的选择端口发送 SYN 置 1 的数据包，如果应答是 RST 置 1 的数据包，那么说明端口是关闭的，如图13.2所示；如果应答是 SYN 和 ACK 置 1 的数据包，说明目标端口处于监听状态，再传送一个 RST 包给目标机停止建立连接，如图13.3所示。需要注意的是第三步客户端发送 RST 数据包给服务器是必需的。由于在 TCP SYN 扫描时，全连接尚未建立，所以这种技术通常被称为半开扫描。

TCP SYN 扫描的优点是隐蔽性比全连接扫描好，很少有系统会记录这样的行为，另外它的扫描结果也是相当准确的，能达到很快的速度。缺点是通常构造SYN 数据包需要超级用户或者授权用户访问专门的系统调用。SYN 洪泛是一种常见的拒绝服务的攻击方法，许多防火墙和入侵检测系统对 SYN 包都建立了报警和过滤机制，因此 SYN 扫描的隐蔽性逐渐下降。

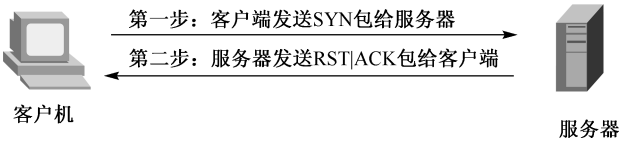


图 13.2 目标端口关闭时 TCP SYN 扫描的步骤

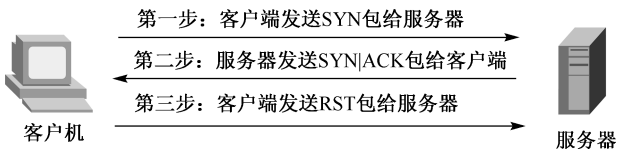


图 13.3 目标端口开放时 TCP SYN 扫描的步骤

2．IP ID 头扫描

IP ID 头扫描也称为**哑扫描**(Dump Scanning)，它主要利用了大多数操作系统 TCP/IP 协议栈的某些特殊实现。其实现主要依赖两个条件，TCP SYN 扫描和第三方主机，第三方主机是一台哑主机。**哑主机**是一台连接在 Internet 上的机器，但是很少甚至几乎没有同其他机器进行通信。该方法由 Antirez 首先使用，并在 Bugtraq 上公布。其原理是：扫描主机通过伪造第三方主机的 IP 地址向目标主机发起 SYN 扫描，并通过观察其 IP 序列号的增长规律获取端口的状态。

该扫描方式的最大优点就是被扫描的主机无法追查扫描者的源地址，另外该扫描方式也不是必须使用TCP SYN扫描，只要满足端口开放和关闭时，哑主机对扫描的主机有不同的响应。它的缺点就是必须依赖第三方的哑主机。

13.3.4.5　秘密扫描

秘密扫描是指一些较难被入侵检测系统和操作系统发现的扫描方法。这些方法的特征是：设置TCP头中的某个或某些标志位(如 ACK, FIN, RST)、不设置 TCP 头中的任何标志位、设置 TCP 头中的所有标志位；能穿透防火墙，伪装成正常的数据通信；该类扫描的另一个特点就是一般是反向确定结果的(False positives)，即当对方没有任何响应时认为目标端口是开放的，而如果返回数据包则认为目标端口是关闭的。有两种方法可以用来确定扫描的结果，一种是设定超时(Time out)时间，当超过一个设定时间没有返回数据包，就认为没有数据包返回，另一种是同时对多个端口进行扫描，对比扫描结果，如果所有端口都没有数据包返回，则认为没有返回数据包。

具体有：TCP SYN|ACK 扫描、TCP FIN 扫描、TCP ACK 扫描、TCP NULL(空)扫描、TCP XMAS Tree 扫描和 UDP 扫描等。

1．TCP SYN|ACK 扫描

通过发送带 SYN|ACK 标志的数据包，一个关闭的端口会返回 RST 标志，如图13.4所示；而一个开放的端口则会忽略该包，不返回任何信息。这种反向确定的方法的主要缺点就是扫描结果的可信度下降。因为存在这样一种可能性：目标端口是关闭的，但 Server 回送的 RST 包被防火墙过滤了，或者因为网络原因在传输过程中丢失了，这时接收不到RST 数据包，就会认定目标端口时开放的。这就造成了结果的误判。该扫描的优点是扫描速度快，能躲避常规的检测。

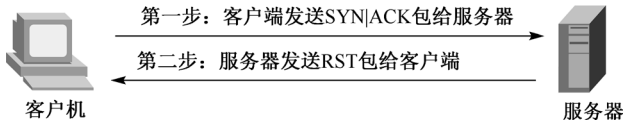


图 13.4　目标端口关闭时 SYN|ACK 扫描的步骤

2．TCP FIN 扫描

TCP FIN 扫描使用 FIN 数据包来探测端口，当一个 FIN 数据包到达一个关闭的端口时，数据包会被丢掉，并且返回一个RST数据包，如图13.5所示；当一个FIN数据包到达一个打开的端口时，数据包只是简单被丢掉，不返回RST数据包。FIN扫描只对UNIX/Linux系统有效。FIN扫描的优点是比TCP SYN扫描更为隐蔽，能够通过只检测 SYN 包的防火墙或者入侵检测系统。缺点是因为是反向确定结果，如果因为网络的传输收不到返回包就会导致错误的判断，扫描结果不是很可靠，而且对于 Windows 系列操作系统无效。

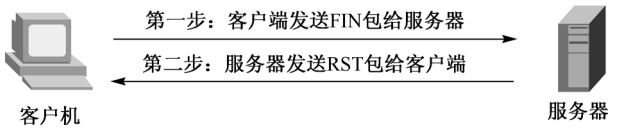


图 13.5　目标端口关闭时 FIN 扫描的步骤

3．TCP ACK 扫描

TCP ACK 扫描主要利用了某些操作系统的 IP 协议实现的Bug，这些操作系统主要指BSD系列。有两个方法可以鉴别端口的状态，一个方法是检查回送的 RST 包的 IP 头的 TTL 值，另一个方法是检查 TCP 头中的 WINDOW 值。

当 Client 发送一个 ACK 数据包给服务器时，服务器会回送一个 RST 数据包。开放端口发送的RST包的IP头中的TTL值一般比关闭的端口小，而且小于等于 64。如以下扫描结果表示端口 2 是开放的。

```
packet 1:  host xxx.xxx.xxx.xxx  port 1  F:RST-->TTL: 70  win: 0
packet 2:  host xxx.xxx.xxx.xxx  port 2  F:RST-->TTL: 40  win: 0
packet 3:  host xxx.xxx.xxx.xxx  port 3  F:RST-->TTL: 70  win: 0
packet 4:  host xxx.xxx.xxx.xxx  port 4  F:RST-->TTL: 70  win: 0
```

开放端口发送的 RST 包的 TCP 头的 WINDOW 值一般是非 0 的，而关闭的端口返回的包的 WINDOW 值为 0。这对于早期的一些操作系统，如BSD 系列的 FreeBSD, OpenBSD，UNIX 中的 AIX 是适用的。但新的一些版本和其他的操作系统已经不能使用该方法扫描了。如以下扫描结果表示端口 7 是开放的。

```
packet 5:  host xxx.xxx.xxx.xxx  port 1  F:RST-->TTL: 64  win: 0
packet 6:  host xxx.xxx.xxx.xxx  port 2  F:RST-->TTL: 64  win: 0
packet 7:  host xxx.xxx.xxx.xxx  port 3  F:RST-->TTL: 64  win: 512
packet 8:  host xxx.xxx.xxx.xxx  port 4  F:RST-->TTL: 64  win: 0
```

4．TCP XMAS Tree 扫描

TCP XMAS树扫描是向目标发送一个将所有标志位都设置为1的 TCP 包，这些标志位包括：ACK, FIN, RST, SYN, URG和PSH。如果对方端口是开放的，则会完全忽略这个包，而不返回任何数据包；如果端口是关闭的，则会回送一个 RST 数据包。

5．TCP NULL 扫描

与 XMAS 扫描相反，TCP空扫描将TCP包中的所以标志位都置 0。当这个数据包被发送到基于BSD 操作系统的主机时，如果目标端口是开放的，则不会返回任何数据包。如果目标端口是关闭的，被扫描主机将发回一个 RST 包。不同的操作系统会有不同的响应方式。

6．UDP 扫描

这种扫描利用UDP协议，向目标端口发送一个UDP分组，开放的 UDP 端口并不需要送回ACK包，而关闭的端口也不要求送回错误包，所以利用 UDP 包进行扫描非常困难。有些协议栈实现的时候，对于关闭的 UDP 端口，会送回一个 ICMP Port Unreachable 的响应消息。其缺点是速度慢，而且 UDP 包和 ICMP 包都不是可靠的，需要 root 权限，才能读取 ICMP Port Unreachable 消息。虽然非 root 用户不能直接读取ICMP Port Unreachable 消息，但是 Linux 提供间接通知的方法。第二次对一个关闭的UDP端口调用 write( )总是会失败。在 ICMP 错误到达之前，在UDP端口上调用 recvfrom( )会返回 EAGAIN(重试)，否则会返回 ECONNREFUSED(连接拒绝)。

13.3.4.6 端口扫描的隐蔽性技术

在进行端口扫描时，为了防止被入侵监测系统发现，黑客还利用了许多技术来使自己的扫描行为更隐蔽。这些扫描技术包括：包特征的随机选择、慢速扫描、分片扫描、源地址欺骗、分布式(合作)扫描等。

1．包特征的随机化

包特征包括 IP 头中的 TTL(Time to live)字段，TCP 头中的源端口、目的端口等字段。在正常的通信中，某主机收到的数据包一般是杂乱无章的，所以为了将扫描行为伪装成正常通信，黑客就会将这些包特征随机化。

这些信息如果是静止不变或按一定的规律变化，则一些入侵检测系统会进行判断和报警。例如许多商业化的入侵检测系统监测顺序的端口连接尝试，比如从 20 一直到80。一旦符合这个特征，NIDS 就会报警。将目标端口的扫描顺序打乱能增加扫描的隐蔽性。

同样，TTL 字段的随机化也是非常重要的。如果收到的 IP 包的 TTL 字段的值都是相同的，入侵检测系统就会认为这些包来自同一台主机，从而有可能对这些包进行阻塞、报警和记录。

2．慢速扫描

许多入侵检测系统会统计某个IP地址在一段时间内的连接次数。当其频率超出设定的范围时，NIDS 就会报警。为了躲避这类的检测，谨慎的黑客会用很慢的速度来扫描对方主机。

3．分片扫描

分片扫描并不直接发送TCP探测数据包，而是将数据包分成一些较小的IP分段。普通的 TCP 包中，源端口、目标端口和标志位一般都是在同一个包中的，而分片时可以将TCP 头分成两半。因为最小分片的长度是 64 bit，足以容纳 IP 头部及源端口和目标端口，所以可以把它们放在第一个包中，把 TCP 的标志位拆分到下一个包中，从而躲避防火墙或入侵检测系统的探测。因为许多入侵检测系统和防火墙是根据数据包的特征，包括各个协议的头部：IP 头、TCP 头、ICMP 头等，进行判断。对于分片的包，它就无法正常地进行分析判断。目标主机得到所有的分片后，会对其进行重组，再进行处理。攻击者就可以根据主机对该包的反应来对目标端口的状态进行判断，就跟没有分片时的扫描结果一样。不过有些操作系统对于分片的包不能正确地加以重组，有的会加以简单地忽略，严重的可能导致系统的死机。

4．源 IP 欺骗

黑客为了使自己计算机的真实IP 不被发现，在进行端口扫描时，伪造大量含有虚假源IP 地址的数据包同时发给扫描目标。因此，目标主机无法从这么多IP 地址中判断扫描真实的发起者。包特征随机化在此也很重要，如果目标主机的入侵检测系统得到的包的TTL 值都是相同的，它会判断这些包实际上来源于同一个 IP，即同一台机器。入侵检测系统的另一个办法是判断这些源 IP 的主机是否真的存在或活动。一个简单的方法就是ping 这些源 IP。如果这些 IP 是黑客随机生成的，则大多数的IP 对应的机器事实上是不存在的，因此入侵检测系统可以把黑客的真实地址锁定在小部分活动主机中。黑客也有相应的方法来对付。他先收集一些活动的主机 IP 做一个列表，把这些列表作为 IP 欺骗时使用的源地址。

5．分布式扫描

入侵检测系统会记录从某个IP 地址发起的连接请求的数量，如果超出一定的限度就会报警。躲避此类检测的一个方法是慢速扫描，另一个方法就是分布式扫描，也称为合作扫描，即一组黑客共同对一台目标主机或某个网络进行扫描，他们之间就可以进行扫描分工。例如每个人扫描某几个端口和某几台主机。这样每个黑客发起的扫描数较小，难以被发现。

13.3.4.7　对抗端口扫描的对策

对抗端口扫描的对策有以下几种：设置防火墙过滤规则，阻止对端口的扫描，例如可以设置检测SYN 扫描而忽略FIN 扫描；使用入侵检测系统，禁止所有不必要的服务，把自己的暴露程度降到最低，UNIX或Linux 中，在/etc/inetd.conf 中注释掉不必要的服务，并在系统启

动脚本中禁止其他不必要的服务，Windows 中通过 Services 禁止敏感服务，如 IIS。

13.3.5 操作系统辨识

黑客能够攻击成功的主要因素是利用了各种软件漏洞，而许多漏洞是与相应版本的操作系统相关的，这是进行操作系统辨识的主要原因。一方面，从操作系统或者应用系统的具体实现中发掘出来的攻击手段需要辨识系统，另一方面，操作系统的信息还可以与其他信息结合起来，比如漏洞库，实施攻击。辨识操作系统的典型技术是TCP/IP 栈指纹技术。此外还有一些获得操作系统信息的简单方法：如DNS 会泄露出操作系统的信息，从一些端口服务，例如 telnet, http, ftp 等服务的提示信息也可以获得操作系统信息。

13.3.5.1 DNS 泄露出的操作系统信息

输入 ping 10.1.0.9 得到如下输出：

```
C:\>ping 10.1.0.9

Pinging 10.1.0.9 with 32 bytes of data:

Reply from 10.1.0.9: bytes=32 time=17ms TTL=248
Reply from 10.1.0.9: bytes=32 time=18ms TTL=248
Reply from 10.1.0.9: bytes=32 time=17ms TTL=248
Reply from 10.1.0.9: bytes=32 time=16ms TTL=248

Ping statistics for 10.1.0.9:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 16ms, Maximum = 18ms, Average = 17ms
```

输入 ping -a 127.0.0.1 得到如下输出：

```
C:\>ping -a 127.0.0.1

Pinging localhost [127.0.0.1] with 32 bytes of data:

Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128

Ping statistics for 127.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

根据 ICMP 报文的 TTL 值，就可以大概知道主机的类型，如 TTL = 125 左右的主机应该是 Windows 系列的，TTL＝235 左右的主机应该是 UNIX 系列的，这是因为不同的操作系统对 ICMP 报文的处理与应答是有所不同的。

13.3.5.2　TCP/IP 栈指纹技术

TCP/IP栈指纹技术根据各个OS在TCP/IP协议栈实现上的不同特点，采用黑盒测试方法，研究其对各种探测的响应，形成识别指纹，进而识别目标主机运行的操作系统。根据采集指纹信息的方式，又可以分为主动扫描和被动扫描两种方式。被动扫描通过被动监测收集数据包，再对数据包的不同特征，如TCP WINDOW 大小，IP 头 TTL，TOS，DF 位等参数，进行分析，来识别操作系统。被动扫描基本不具备攻击特征，具有很好的隐蔽性，但其实现严格依赖扫描主机所处的网络拓扑结构，和主动探测相比较，具有速度慢、可靠性不高等缺点。主动扫描采用向目标系统发送构造的特殊包并监控其应答的方式，寻找不同操作系统之间在处理网络数据包上的差异，并且把足够多的差异组合起来，以便精确地识别出一个系统的 OS 版本。主动扫描具有速度快、可靠性高等优点，但同样严重依赖于目标系统网络拓扑结构和过滤规则。

13.3.5.3　对抗操作系统辨识的对策

对抗操作系统辨识的对策有以下几种：一些端口扫描监测工具可以监视操作系统检测活动；可以打一些让操作系统识别失效的补丁；修改OS 的源代码或改动某个OS 参数以达到改变单个独特的协议栈特征的目的；通过防火墙和路由器的规则配置阻止对操作系统的主动探测数据包；使用入侵检测系统发现对操作系统辨识的行为。

13.3.6　针对特定应用和服务的漏洞扫描

漏洞扫描器是一种自动检测远程或本地主机安全性弱点的程序。按常规标准，可以将漏洞扫描器分为两种类型：主机漏洞扫描器(Host Scanner)和网络漏洞扫描器(Network Scanner)。基于主机的扫描工具运行于单个主机，扫描目标为本地主机，分析本地文件内容，对用户账号文件、系统权限、系统配置文件、关键文件、日志文件、用户口令、网络接口状态、系统服务、软件脆弱性以及其他同安全规则抵触的对象进行检查。网络漏洞扫描器通过远程检测目标主机TCP/IP 不同端口的服务，记录目标给予的回答。通过这种方法，可以收集到很多目标主机的各种信息，例如：是否能匿名登录，是否有可写的 FTP 目录，是否能用 telnet 等，将这些相关信息与网络漏洞扫描系统提供的漏洞库进行匹配，查看是否有满足匹配条件的漏洞存在。此外，还可以通过模拟黑客的攻击手法，对目标主机系统进行攻击性的安全漏洞扫描，如测试弱口令等。若模拟攻击成功，则表明目标主机系统存在安全漏洞。

13.4　口令攻击

人们随时都在和各种口令打交道，如开机口令、在ATM机上取钱的口令、访问电子邮箱的口令、登录论坛的口令，口令是进入计算机系统的第一道防线，是提供保护的重要手段，也是黑客攻击的首要目标。针对口令的攻击表现为以下几种形式：弱口令扫描、暴力穷举、字典破解、密码监听、木马和键盘记录程序和社会工程攻击等。

弱口令扫描是用一些简单口令对用户进行探测。因为实际探测表明，有很大比例的用户采用这样的口令，如“123456”、“123”，或者采用用户名作为口令。

暴力穷举对每一个可能的口令进行试探，攻击时间完全取决于机器的运算速度，把用户常用的口令构造成字典的字典攻击，大大减少了运算的次数，提高了攻击成功率。

通过嗅探器监听网络中的数据包，也能获得它所连接的网段上传输的密码，嗅探成功基

于以下条件：共享信道，如广播型以太网，任何一个站点发送帧，其他所有站点都会收到该帧；协议不加密，口令是明文传输的，才能监听到明文形式的口令；网卡设置为混杂模式，处于这种模式的网卡能接受网络中所有数据包。

被评论称为“世界头号黑客”的凯文·米特尼克(Kevin David Mitnick)在其所著的《欺骗的艺术》一书中揭示了政府、企业和我们每一个人，在社会工程师的入侵面前是多么的脆弱和易受攻击。在这个重视信息安全的时代，我们在技术上投入大量的资金来保护计算机网络和数据，而该书告诉我们，骗取内部人员的信任和绕过所有技术上的保护是多么的轻而易举。

一些木马能够记录受害者的键盘敲击，并且在受害者不知道的情况下，把所找到的密码发送到指定的信箱。

针对口令破解攻击的防范措施有：安装入侵检测系统，检测口令破解的行为； 安装安全评估系统，先于入侵者进行模拟口令破解，以便及早发现弱口令并解决；提高安全意识，避免弱口令。

针对网络嗅探攻击的防范措施有：安装VPN网关，防止对网关间信道进行嗅探；对内部网络通信采取加密处理，防止对内部网段的密码监听；采用交换设备进行网络分段；采取技术手段发现处于混杂模式的主机。

13.5 欺骗攻击

技术性的欺骗攻击利用了 TCP/IP 协议的缺陷，不涉及软件的漏洞。欺骗攻击虽然是技术含量较低的攻击方式，历史上还是发生过一些欺骗攻击的著名事件，如 1994 年 12 月 25 日，凯文·米特尼克利用 IP 欺骗技术攻破了 San Diego 计算中心；1999 年，RSA Security 公司网站遭受 DNS 欺骗攻击；1998 年，中国台湾地区某电子商务网站遭受 Web 欺骗攻击，造成大量客户的信用卡密码泄露。欺骗攻击具体形式有：IP 欺骗、邮件欺骗、Web 欺骗、ARP 欺骗和非技术性欺骗。

13.5.1 IP 欺骗

13.5.1.1 IP 欺骗的原理

IP 欺骗是假冒他人的IP地址来获得信息或发送信息。IP欺骗的主要动机有三类：隐藏自己的IP地址，防止被跟踪；希望获得以IP地址作为授权依据的授权；为了穿越防火墙，躲过基于IP地址的过滤规则。IP 欺骗之所以能成功，是因为IP协议中，信任服务的基础仅仅是建立在网络地址的验证上。IP 欺骗的形式分为单向欺骗、双向 IP 欺骗和 TCP 会话劫持。单向欺骗不考虑回传数据包的要求，如图13.6所示，攻击者的 IP 地址为 10.20.20.20，他给 Bob 的主机发送一个数据包，声称来自地址 10.10.10.20，Bob 的主机就会把应答数据包发到地址 10.10.10.20，攻击者看不到返回的数据包。双向 IP 欺骗要求看到回传的数据包，如图13.7所示，第一步，攻击者假冒 IP 地址10.10.10.20 给 Bob 的主机发送一个数据包，Bob 的主机的应答数据包经过攻击者转发到地址 10.10.10.20，这样攻击者就可以看到应答数据包。在双向 IP 欺骗中，为了得到从目的机器返回被欺骗机器的流量，一个方法是攻击者插入到正常情况下流量经过的地方。

要实现 IP 地址欺骗，最简单的做法是盗用 IP 地址，在 Windows 系统中，可以使用网络配置工具改变机器的 IP 地址，在 UNIX 系统上，可以使用 ifconfig 命令来改变地址。但这还不算是真正的 IP 地址欺骗，因为从攻击者机器发出的数据包的 IP 地址和他本机盗用的 IP 地

址还是一致的。攻击者更希望实现不改变本机的 IP 地址，发送 IP 包，但 IP 包头填上假冒的源 IP 地址。在 UNIX/Linux 系统中，直接用socket就可以发送，但是需要 root 权限。此外，还有一些构造 IP 包的开发工具，在 Windows 系统中，可以使用 Winpcap 构造 IP 包，在 UNIX 系统中，可以用libnet构造IP包。在 UNIX 系统平台上的网络安全工具开发中，目前最为流行的 C 语言函数库有 libnet, libpcap, libnids 和libicmp 等。它们分别从不同层次和角度提供了不同的功能函数，使网络开发人员能够忽略网络底层细节的实现，从而专注于程序本身具体功能的设计与开发。其中，libnet提供的接口函数主要实现和封装了数据包的构造和发送过程，libpcap 提供的接口函数主要实现和封装了与数据包截获有关的过程，libnids 提供的接口函数主要实现了开发网络入侵检测系统所必需的一些结构框架，libicmp 等相对较为简单，它封装的是 ICMP 数据包的主要处理过程，包括构造、发送、接收等。Winpcap（Windows packet capture）是 Windows 32 平台下一个免费的包截获与网络分析系统。开发 Winpcap 这个项目的目的在于为 Win32 应用程序提供访问网络底层的能力。它提供了以下的各项功能：捕获原始数据包，包括在共享网络上各主机发送/接收的以及相互之间交换的数据包；在数据包发往应用程序之前，按照自定义的规则将某些特殊的数据包过滤掉；在网络上发送原始的数据包；收集网络通信过程中的统计信息。

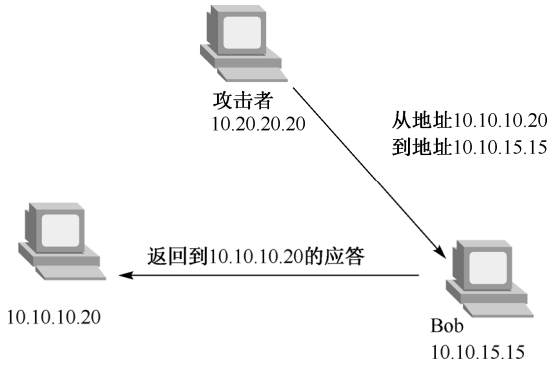


图 13.6　单向 IP 欺骗

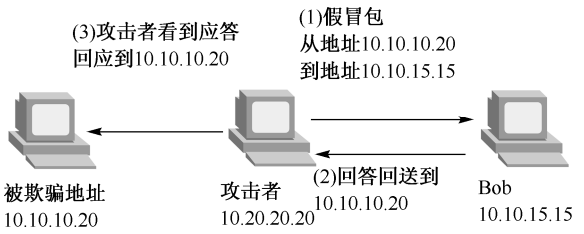


图 13.7　双向 IP 欺骗

13.5.1.2　如何避免 IP 欺骗

为了避免IP 欺骗，可以从主机保护、网络保护两方面采取措施。保护自己的机器不被用来实施IP 欺骗，可以采用物理防护措施，防止没有权限的用户接触本机；可以设置登录口令和权限控制，不允许没有权限的用户修改配置信息。但还没有办法保护自己的机器不会成为假冒的对象。在网络上可以采取的措施为在路由器上设置欺骗过滤器，如入口过滤，禁止外来的包带有内部IP地址；出口过滤，禁止内部的包带有外部 IP 地址；在路由器上禁止有源路由选项的数据包。

13.5.2 邮件欺骗

邮件欺骗简单地说就是假冒他人的 E-mail 地址发送信息，电子邮件欺骗的动机通常有：隐藏发信人的身份，冒充别人或给其他人找麻烦，骗取敏感信息等。欺骗的形式有使用相似的电子邮件地址、修改邮件客户软件的账号配置、直接连到 SMTP 服务器上发信。电子邮件欺骗成功的要诀是基本的电子邮件协议不包括签名机制。

13.5.2.1 使用相似的电子邮件地址

这种欺骗是发信人使用被假冒者的名字注册一个账号，然后给目标发送一封正常的信，如果你收到这样一封信，“我是××，请把×××发送给我。我在外面度假，请送到我的个人信箱”，你能识别吗？为了避免相似邮件攻击，应该建立公司的电子邮件，规定任何与工作有关的活动必须使用工作邮件。另一种可能的方法是使用数字签名。

13.5.2.2 修改邮件客户软件的账号配置

图 13.8 Outlook Express 的账号配置

如果攻击者有一个 Outlook Express 的邮件客户端，他就可以如图13.8所示配置邮件账户，姓名属性，会出现在“From”和“Reply-To”字段中，然后显示在“发件人”信息中；电子邮件地址，会出现在“From”字段中；回复地址，会出现在“Reply-To”字段中，可以不填。攻击者可以在相应的栏目填入虚假的信息。

为了免受修改邮件客户的攻击，首先需要制定一个电子邮件的安全政策，其次要在系统中保留邮件发送记录。对于收信人，只要查看完整的电子邮件头，就可以发现是否有人进行了邮件欺骗，还可以找到信息的来源，因为多数邮件系统允许用户查看邮件从源地址到目的地址经过的所有主机。通过查看如下收到的邮件头，我们可以知道该封邮件实际是从 162.105.30.*xxx* 发出的。

```
Return-Path: <test@test.pku.edu.cn>
X-Original-To: wangzhao@infosec.pku.edu.cn
Delivered-To: wangzhao@infosec.pku.edu.cn
Received: from LENOVO243AF5FB (unknown [162.105.30.xxx])
    by infosec.pku.edu.cn (Postfix) with ESMTP id BB559DE1B
    for <wangzhao@infosec.pku.edu.cn>; Wed, 27 May 2009 19:11:39 +0800 (CST)
Message-ID: <1AB313F970A048E4AD12FFFF9E619AA0@LENOVO243AF5FB>
From: "Alice" <test@test.pku.edu.cn>
To: "Wang Zhao" <wangzhao@infosec.pku.edu.cn>
Subject: test
Date: Wed, 27 May 2009 19:11:56 +0800
```

13.5.2.3 直接连接 SMTP 服务器

执行邮件欺骗第三种方法是直接连接SMTP服务器的 25 端口，然后发送命令，常见命令为 Helo, Mail from, Rcpt to, Data 和Quit。在 Mail from 命令后填上虚假的邮件地址就可以了。

为了防止这种方式的邮件欺骗，可以对邮件服务器进行如下验证，SMTP服务器验证发送者的身份，以及发送的邮件地址是否与邮件服务器属于相同的域；验证接收方的域名与邮件服务器的域名是否相同；通过反向DNS解析验证发送者的域名是否有效。此外，还要为邮件服务器安装上所有最新的补丁。但是这不能防止一个内部用户假冒另一个内部用户发送邮件，此外，攻击者也可以运行自己的 SMTP 邮件服务器。

13.5.3　TCP 会话劫持

TCP 会话劫持(Session hijacking)是一种高级的欺骗形式，前面所介绍的欺骗技术是攻击者伪装成合法用户，以获得一定的利益，而劫持是积极主动地使一个在线的用户下线，或者冒充这个用户发送消息，以便达到自己的目的。会话劫持分两种：被动劫持和主动劫持。被动劫持，实际上就是藏在后面监听所有的会话流量，常常用来发现密码或者其他敏感信息。主动劫持通过找到当前活动的会话，并且把会话接管过来。迫使一方下线，由劫持者取而代之，危害更大，因为攻击者接管了一个合法的会话之后，可以做许多危害性更大的事情。

猜测序列号是 TCP 会话劫持最关键的一步，如果猜测不当就会造成 ACK 风暴。当一个主机接收到一个不期望的数据包的时候，它会用自己的序列号发送ACK，而这个包本身也是不可被接受的。于是，两边不停地发送 ACK 包，形成 ACK 包的循环，被称为 ACK 风暴。如果被劫持用户不掉线，也会发生 ACK 风暴。如果有一个 ACK 包丢掉，则风暴停止。

可以进行会话劫持的一些工具有 Juggernaut, Hunt, TTY Watcher 和 IP Watcher。防止会话劫持的一些措施有：使用安全协议如 SSH, VPN 对 TCP 会话加密；在防火墙配置一些过滤规则，限制尽可能少量的外部许可连接的 IP 地址；使用一些检测机制，当发现 ACK 包的数量明显增加时，就提示可能发生了会话劫持；拥有完善的鉴别措施也是防止会话劫持的有效途径，如果用户长时间在线，可以要求用户定时进行身份鉴别。

此外还有 Web 欺骗、ARP 欺骗和 DNS 欺骗等形式的欺骗技术，从这些欺骗技术，我们可以看到，IP 协议的脆弱性，应用层上也缺乏有效的安全措施。在网络攻击技术中，欺骗技术是比较初级的，技术含量并不高，它是针对Internet中各种不完善的机制而发展起来的。非技术性的欺骗，比如，实施社会工程攻击，利用了人们的弱点，以顺从你的意愿、满足你的欲望的方式，让你上当。毕竟网络世界与现实世界是紧密相关的，避免被欺骗最好的办法是教育、教育、再教育，增强每一个Internet用户的安全意识，增强网络管理人员以及软件开发人员的安全意识更加重要。

13.6　拒绝服务攻击

13.6.1　拒绝服务攻击的类型

回顾信息安全的三个主要需求：保密性、完整性和可用性，拒绝服务攻击(Deny of Service，DoS)是针对可用性发起的攻击。DoS 攻击通过某些手段使得目标系统或者网络不能提供正常的服务，主要是利用了TCP/IP 协议中存在的设计缺陷和操作系统及网络设备的网络协议栈存在的实现缺陷。虽然DoS难以防范，有些 DoS 可以通过管理的手段防止。DoS 通常是黑客无法攻入目标系统的最后一招，进行恶意破坏或者报复。

一些商业及政府网站都曾经遭受拒绝服务攻击，在2000年2月，发生的一次对某些高利润

站点 Yahoo, eBay, Buy.com 等的拒绝服务攻击，持续了近两天，使这些公司遭受了很大的损失。事后这些攻击确定为分布式的拒绝服务攻击。

从攻击技术看，DoS 攻击表现为：带宽消耗、系统资源消耗、程序实现上的缺陷、系统策略的修改和物理部件的移除或破坏。带宽消耗是通过网络发送大量信息，用足够的传输信息消耗掉有限的带宽资源。系统资源消耗是向系统发送大量信息，针对操作系统中有限的资源，如进程数、磁盘、CPU、内存、文件句柄，等等。利用程序实现上的缺陷，对异常行为的不正确处理，通过发送一些非法数据包使系统死机或重启，比如 Ping of Death。修改或篡改系统策略，也可以使得它不能提供正常的服务。最直接的一种方式就是物理部件的直接移除或破坏。

从攻击目标来看，有通用类型的DoS 攻击和系统相关的攻击。通用类型的DoS 攻击往往是与具体系统无关的，比如针对协议设计缺陷的攻击；系统相关的攻击，这类攻击往往与具体的实现有关。最终，所有的攻击都是系统相关的，因为有些系统可以针对协议的缺陷提供一些补救措施，从而免受此类攻击。

一些典型的拒绝服务攻击有：Ping of Death、泪滴（Teardrop）、UDP 洪泛（UDP Flooding）、Land、SYN 洪泛（SYN Flooding）、Smurf 等。

13.6.2 Ping of Death

Ping of Death 直接利用 ping 包，即 ICMP Echo 包，发送异常的，长度超过 TCP/IP 允许数据包的最大长度 65 536 的数据包，有些系统在收到大量比最大包还要长的数据包，会挂起或者死机，很多操作系统都受到这个攻击的影响。

现在所有的标准TCP/IP 实现都已实现了对付超大尺寸包的功能，并且大多数防火墙能够自动过滤这些攻击，包括：从 Windows 98 之后的 Windows NT（Service Pack 3 之后），Linux, Solaris 和Mac OS 都具有抵抗一般 Ping of Death 攻击的能力。如当对一个实现了对付超大尺寸包的 Windows 系统发送超长包的显示如下：

```
C:\>ping -l 65570 162.105.30.200
  Bad value for option -l, valid range is from 0 to 65500.
```

13.6.3 IP 碎片

IP 碎片攻击方式是发送一系列高度碎片化的超大ICMP 数据包，它利用那些在TCP/IP 协议栈实现中，对IP 碎片包头所包含信息的信任来实现自己的攻击。IP 头中有一个分段偏移，指示该分段所包含的是原数据包哪一段的信息，某些 TCP/IP（包括 Service Pack 4 以前的 NT）在收到含有重叠偏移的伪造分段时将崩溃。常见的IP碎片程序有jolt2, teardrop, newtear, syndrop 和 boink 等。

泪滴是一种IP 碎片攻击，也是一种常见的拒绝攻击方式，它的意思是气得让人掉泪。它的攻击形式非常简单，发送一些IP 分片异常的数据包。它利用的是利用IP 包的分片装配过程中，由于分片重叠，计算过程中出现长度为负值，在执行memcpy的时候导致系统崩溃。当网络分组穿越不同的网络时，有时候需要根据网络最大传输单元MTU 来把它们分割成较小的片，早期的Linux 系统在处理IP 分片重组问题时，尽管对片段是否过长进行检查，但对过短的片段却没有进行验证，所以导致了泪滴形式的攻击。受影响的系统有 Linux/Windows NT/95。

如图13.9所示，在 Linux 2.0 内核中有以下处理：当发现有位置重合时（offset2<end1），将

offset 向后调到 end1（offset2=end1），然后更改 len2 的值：len2 = end2−offset2，此时len2变成了一个小于零的值，会导致以后处理时出现溢出。

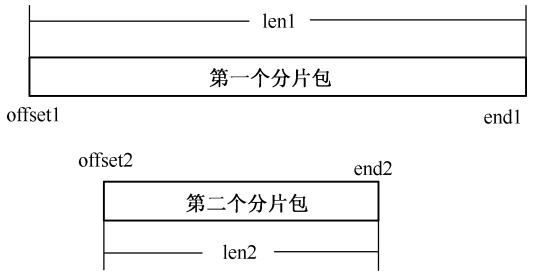


图 13.9　异常分片重组

防御IP 碎片攻击的措施有：在设置防火墙时对分段进行重组，而不是转发它们；加入条件判断，对这种异常的包特殊处理，如分片组装时检查 len2 的值便可，这样可以发现所有泪滴攻击的变种（如 newtear）。

13.6.4　UDP 洪泛

UDP 洪泛攻击的原理是：各种各样的假冒攻击利用简单的TCP/IP服务，如chargen和 Echo来传送毫无用处的占满带宽的数据。通过伪造与某一主机的 chargen 服务之间的一次 UDP 连接，回复地址指向开着Echo服务的一台主机，这样就在两台主机之间生成足够多的无用数据流，导致带宽耗尽的拒绝服务攻击。

关掉不必要的 TCP/IP 服务，或者对防火墙进行配置，阻断来自Internet 的对这些服务的 UDP 请求都可以防范 UDP 洪泛攻击。

13.6.5　SYN 洪泛

SYN 洪泛攻击利用 TCP 连接三次握手过程，打开大量的半开 TCP 连接，使得目标机器不能进一步接受TCP 连接。每个机器都需要为这种半开连接分配一定的资源，并且，这种半开连接的数量是有限制的，达到最大数量时，CPU 满负荷或内存不足，机器就不再接受进来的连接请求，如图13.10所示。在SYN 洪泛攻击中，连接请求是正常的，但是，源IP地址往往是伪造的，并且是一台不可达的机器的IP地址，否则，被伪造地址的机器会重置这些半开连接。一般，半开连接超时之后，会自动被清除，所以，攻击者的系统发出SYN包的速度要比目标机器清除半开连接的速度要快。任何连接到 Internet 上并提供基于 TCP 的网络服务，都有可能成为攻击的目标。这样的攻击很难跟踪，因为源地址往往不可信，而且不在线。

SYN洪泛攻击的攻击特征是：目标主机的网络上出现大量的SYN包，而没有相应的应答包；SYN包的源地址可能是伪造的，甚至无规律可循。

可以在主机和网络上采取措施防止SYN洪泛攻击。防火墙或者路由器可以在给定时间内只允许有限数量的半开连接，入侵检测可以发现这样的 DoS 攻击行为。主机上可以限制 SYN Timeout的时间。此外一些操作系统也实现了防止 SYN 洪泛攻击的功能，如 Linux 和 Solaris 使用了一种被称为 SYN cookie 的技术来解决 SYN 洪泛攻击：在半开连接队列之外另设置了一套机制，使得合法连接得以正常继续。

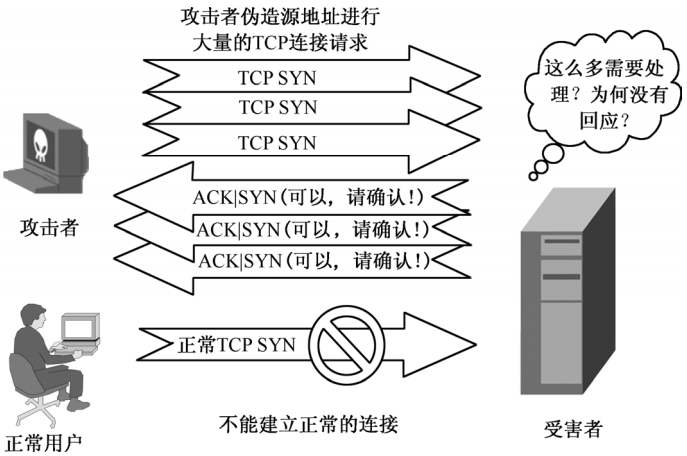


图 13.10　SYN 洪泛攻击示意图

13.6.6　Smurf

在 Smurf 攻击中，攻击者向一个广播地址发送 ICMP Echo 请求，并且用受害者的 IP 地址作为源地址，于是，广播地址网络上的每台机器响应这些Echo请求，同时向受害者主机发送 ICMP Echo Reply 应答。于是，受害者主机会被这些大量的应答包淹没，如图13.11所示。大多数操作系统和路由器都会受到此类攻击的影响。此类攻击还有一个变种叫做 fraggle，使用 UDP 包，或称为 udpsmurf。比如，攻击者向 7 号端口发送 ICMP Echo 请求，如果目标机器的端口开着，则发送 ICMP Echo Reply，否则，产生 ICMP 端口不可达消息。这个攻击的两个主要的特点是使用伪造的数据包和使用广播地址。不仅被伪造地址的机器受害，目标网络本身也是受害者，它们要发送大量的应答数据包。Smurf攻击涉及三方：攻击者、中间目标网络和受害者。它以较小的网络带宽资源，通过放大作用，吃掉较大带宽的受害者系统。作为Smurf放大器的网络不仅允许ICMP Echo请求发给网络的广播地址，并且允许ICMP Echo-Reply发送回去，这样的网络越多，对Internet的危害就越大。实施 Smurf 攻击需要长期的准备，首先找到足够多的中间网络，集中向这些中间网络发出 ICMP Echo 包。

防止Smurf攻击的措施关键在于防止攻击者利用中间网络，设置网关或者防火墙，禁止源地址非本网络IP地址的数据包出去，禁止外来的IP广播消息，但是，如果攻击者从内部机器发起攻击，仍然不能阻止Smurf攻击。设置路由器，使其禁止对定向广播的支持。针对最终受害者，还没有直接的方法可以阻止自己接收 ICMP Echo Reply 消息，但是可以在路由器上阻止这样的应答消息。

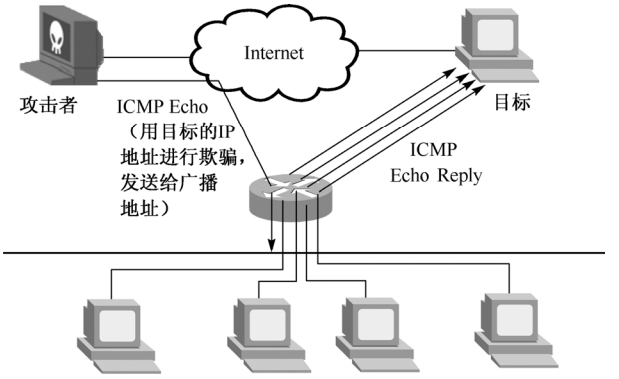


图 13.11　Smurf 攻击示意图

13.6.7　Land

这是一种比较老的攻击，目前大部分操作系统都能避免。在Land攻击中，向目标主机的某个开放端口发送一个特别的SYN 包，它的源地址和目标地址都被设置成同一个地址，源端口和目的端口相同，大多数主机都不知道怎么处理这样的数据包，导致瘫痪或挂起，不同的系统对 Land 攻击反应不同，许多 UNIX 实现将崩溃，NT 变得极其缓慢(大约持续五分钟)。

打最新的补丁，或者在防火墙进行配置，将那些在外部接口上入站的含有内部源地址的数据包过滤掉，包括 10 域、127 域、192.168 域、172.16 到 172.31 域，都可以比较有效地防范 Land 攻击。

13.6.8　分布式拒绝服务攻击

传统的拒绝服务是一台机器向受害者发起攻击，分布式拒绝服务攻击(Distributed Denial of Service attack，DDoS)不是仅仅一台机器而是多台主机合作，同时向一个目标发起攻击。DDoS 攻击模型如图13.12所示，包含攻击者、主控端和代理端三个层次。

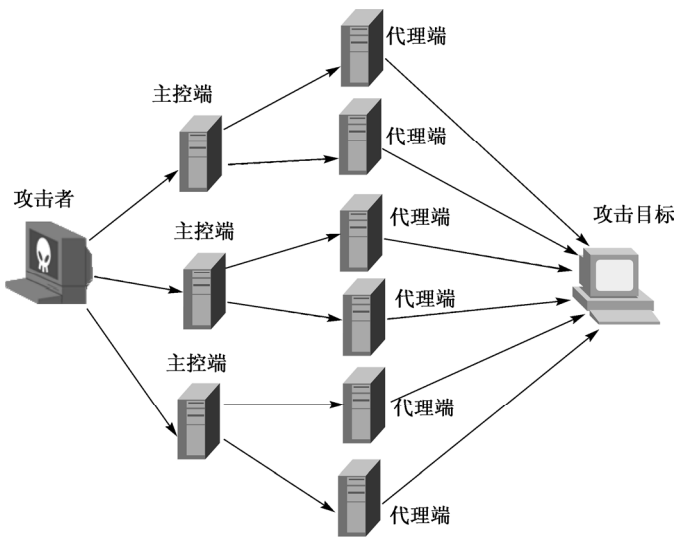


图 13.12　DDoS 攻击模型

攻击者所用的计算机是攻击主控台，可以是网络上的任何一台主机，甚至可以是一个活动的便携机。攻击者操纵整个攻击过程，它向主控端发送攻击命令。主控端是攻击者非法侵入并控制的一些主机，这些主机还分别控制大量的客户主机。主控端主机的上面安装了特定的程序，因此它们可以接受攻击者发来的特殊指令，并且可以把这些命令发送到代理主机上。代理端同样也是攻击者侵入并控制的一批主机，它们上面运行攻击程序，接收和运行主控端发来的命令。代理端主机是攻击的执行者，真正向受害者主机发送攻击。

DDoS 攻击的主要工具有 TFN(Tribe Flood Network), TFN2K, Trinoo 和 Stacheldraht。

DDoS攻击的防御策略有：及早发现系统存在的攻击漏洞，及时安装系统补丁程序。对一些重要的信息(例如系统配置信息)建立和完善备份机制。对一些特权账号(例如管理员账号)的密码设置要谨慎。网络管理方面，要经常检查系统的物理环境，禁止那些不必要的网络服务。建立边界安全界限，确保输出的包受到正确限制。经常检测系统配置信息，并注意查看

每天的安全日志。利用网络安全设备(例如防火墙)来加固网络的安全性，配置好它们的安全规则，过滤掉所有可能的伪造数据包。

13.7 缓冲区溢出攻击

1988 年的 Morris 蠕虫病毒，感染了6000多台机器：它利用 UNIX 服务 finger 中的缓冲区溢出漏洞来获得访问权限，得到一个 shell。1996 年前后，开始出现大量的缓冲区溢出(Buffer Overflows)攻击，因此引起人们的广泛关注，源码开放的操作系统首当其冲。随后，Windows 系统下的Buffer Overflows也相继被发掘出来，已经有一些非常经典细致的文章来介绍与 Buffer Overflows 有关的技术。

缓冲区是程序运行期间，在内存中分配的一个连续的区域，用于保存包括字符数组在内的各种数据类型。**溢出**是所填充的数据超出了原有缓冲区的边界，并非法占据了另一段内存区域。**缓冲区溢出**是指由于填充数据越界而导致程序原有流程的改变，黑客借此精心构造填充数据，让程序转而执行特殊的代码，最终获得系统的控制权。

一个运行的程序占用的内存从逻辑上可分为代码区和数据区。文本区或者说代码区存放的是程序的执行代码以及只读数据。数据区可分为静态数据区和动态数据区。静态数据区存储静态变量和全局变量。动态数据区可按多种方式组织，典型的组织是将这个存储区域分为栈(stack)区域和堆(heap)区域。堆是存储程序运行过程中动态分配的数据块，是不连续的内存区域，需要程序员自己申请，并指明大小，在 C 语言中用 malloc\calloc 函数进行分配。栈用于存储函数调用所传递的参数、返回地址、局部变量等内容，是一块连续的内存区域，由系统自动分配。图13.13简要描述了一种典型的进程内存分布，不同操作系统的实际具体情况各不相同，比图示中的更复杂。图中，堆和栈在使用时相向生长，栈向上生长，即向低地址方向扩展，而堆向下增长，即向高地址方向扩展，它们之间的剩余部分是自由空间。

图 13.13 进程内存的分布

每一次过程或函数调用，在堆栈中必须保存称为**栈帧**的数据结构，里面包含传递给函数的参数、函数返回后下一条指令的地址、函数中分配的局部变量、恢复前一个栈帧需要的数据(基地址寄存器的值)。在函数的栈帧中，很重要的一部分是就是留给局部变量的存储空间，如果向该区域填充的数据超过了预先分配的大小，很可能会覆盖掉下面的返回地址，黑客只要构造传递给程序的参数字串，使其溢出，用自己的代码地址覆盖原有的返回地址，就可以实现改变程序流程的目的。

在C语言中，指针和数组越界不保护是缓冲区溢出的根源，而且，在C语言标准库中就有许多能提供溢出的函数，如 strcat(), strcpy(), sprintf(), vsprintf(), bcopy(), gets()和scanf()。虽然编程的问题都可以在开发阶段防止，事实上，并没有这么简单。有些开发人员没有意识到问题的存在；有些开发人员不愿意使用边界检查，因为会影响效率和性能；另一方面，遗留下来的代码还很多。在开发过程中，尽量使用带有边界检查的函数版本，或者自己进行越界检查是防止缓冲区溢出的基本方法。

缓冲区溢出是代码中固有的漏洞，除了在开发阶段要注意编写正确的代码之外，对于用户而言，一般的防范措施为：关闭端口或服务，管理员应该知道自己的系统上安装了什么，

并且哪些服务正在运行，因为一些缓冲区溢出漏洞与特定的应用或服务相关；安装软件厂商的补丁、漏洞一公布，大的厂商就会及时提供补丁，而很多攻击都是在用户没有及时打补丁，基于已知的漏洞攻击成功的；在防火墙上过滤特殊的流量也是一个基本方法，但是无法阻止内部人员的溢出攻击；此外，还可以自己检查关键的服务程序，看看是否有可怕的漏洞；最后，为了限制黑客溢出成功获得的权限，以所需要的最小权限运行软件。

思考和练习题

(1) 攻击方式的分类原则有哪些？
(2) 攻击方式的分类方法有哪些？
(3) 基于攻击技术可以把攻击分为哪些类型？
(4) 基于多维属性的攻击分类方法中，可以利用攻击的哪些属性进行分类？
(5) 扫描的目的是什么？包含哪些类型？
(6) 主机扫描的技术有哪几种？
(7) 端口扫描的类型有哪些？
(8) 简述 IP 欺骗技术。
(9) 典型的拒绝服务攻击有哪几种？
(10) 简述 Smurf 攻击的原理和防范措施。
(11) 简述分布式拒绝服务攻击的原理和防范措施。

实践/实验题

(1) 使用某种扫描软件，查看一台主机，说明该主机运行的是什么操作系统，提供了哪些服务，有什么漏洞。
(2) 编写一个缓冲区溢出的程序，在攻击成功后显示“Buffer Overflow Success”。

第 14 章　计算机病毒及其防治

本章从计算机病毒的定义、命名、分类、特性、发展、基本原理等方面对计算机病毒进行了全面的分析介绍，在此基础上，归纳和总结了现有的反病毒技术及对策。

14.1　计算机病毒的定义

“计算机病毒”为什么叫做病毒呢？首先，与生物医学上的“病毒”不同，它不是天然存在的，是某些人利用计算机软硬件的脆弱性和计算机体系结构本身的缺陷，编制的具有特殊功能的程序。生物学中的病毒具有传染性、流行性以及针对性，而计算机病毒具备了很多与生物病毒相同的特征，因此，这一名词是由生物医学上的“病毒”一词借用而来的。

从广义上定义，凡能够引起计算机故障，破坏计算机数据的程序统称为计算机病毒。依据此定义，诸如逻辑炸弹、蠕虫、木马等均可称为计算机病毒。在国内，研究者从不同角度给出了计算机病毒的定义。一种较为广泛的定义是**计算机病毒**就是能够通过某种途径潜伏在计算机存储介质(或程序)里，当达到某种条件时即被激活的具有对计算机资源进行破坏作用的一组程序或指令集合。

美国计算机安全专家 Fred Cohen 1989 年把计算机病毒定义为：“病毒程序通过修改(操作)而传染其他程序，即修改其他程序使之含有病毒自身的精确版本或可能演化版本、变种或其他的病毒繁衍体。病毒可以看做是攻击者愿意使用的任何代码的携带者。病毒中的代码可经由系统或网络进行扩散，从而强行修改程序和数据。”这一定义具有一定的狭义性，但是比较实用。

直至1994年2月18日，我国正式颁布实施了《中华人民共和国计算机信息系统安全保护条例》，在《条例》第二十八条中明确指出：“计算机病毒，是指编制或者在计算机程序中插入的破坏计算机功能或者毁坏数据，影响计算机使用，并能自我复制的一组计算机指令或者程序代码。”此定义具有法律性、权威性。

14.2　计算机病毒的基本特征

计算机病毒种类繁多，各自具有不同的特征，从计算机病毒的定义可以看出传染性和破坏性是其最基本的特征，其次，计算机病毒还具有隐蔽性、可触发性、非授权性等一系列共性，具体如下所述。

(1) **寄生性(依附性)**：计算机病毒是一种特殊的计算机程序，它不是以独立的文件的形式存在的，它寄生在合法的程序中，这些合法的程序可以是系统引导程序、可执行程序、一般应用程序等。病毒所寄生的合法程序被称为病毒的载体，也称为病毒的宿主程序。病毒程序嵌入到宿主程序中，依赖于宿主程序的执行而生存，这就是计算机病毒的寄生性。

(2) **传染性**：计算机病毒的传染性是指计算机病毒会通过各种渠道从已被感染的计算机扩散到未被感染的计算机。正常的计算机程序一般是不会将自身的代码强行连接到其他

程序之上的。而病毒却能使自身的代码强行传染到一切符合其传染条件的未受到传染的程序之上。是否具有传染性是判别一个程序是否为计算机病毒的最重要条件。

(3) **隐蔽性**：计算机病毒在发作之前，必须能够将自身很好地隐蔽起来，不被用户发觉，这样才能实现进入计算机系统、进行广泛传播的目的。计算机病毒的隐蔽性表现为传染的隐蔽性与存在的隐蔽性。传染的隐蔽性指的是大多数病毒在进行传染时速度是极快的，一般不具有外部表现，使人非常不易察觉。病毒程序存在的隐蔽性指的是，计算机病毒一般是具有很高编程技巧、短小精悍的程序，通常附在正常程序中或磁盘较隐蔽的地方，也有个别的以隐含文件形式出现。

(4) **潜伏性**：计算机病毒的潜伏性具体有两种表现形式。第一，病毒程序不用专用检测程序是检查不出来的；第二，计算机病毒的内部往往有一种破坏触发机制，不满足触发条件时，计算机病毒除了传染外不进行任何破坏。病毒程序为了达到不断传播并破坏系统的目的，一般不会在传染某一程序后立即发作，否则就暴露了自身。通常，一个编制精巧的计算机病毒程序，可以在几周、几个月内甚至几年内隐藏在合法文件中，对其他系统进行传染，而不被人发现。潜伏性越好，其在系统中的存在时间就会越长，病毒的传染范围就会越大。

(5) **可触发性**：因某个特征或数值的出现，诱使病毒实施感染或进行攻击的特性称为可触发性。

(6) **破坏性**：所有的计算机病毒都是一种可执行程序，而这一可执行程序又必然要运行，所以对系统来讲，所有的计算机病毒都存在一个共同的危害，即降低计算机系统的工作效率，占用系统资源，其具体情况取决于入侵系统的病毒程序。同时计算机病毒的破坏性主要取决于计算机病毒设计者的目的，它可以彻底破坏系统的正常运行，也可以只是做一些无意义的炫耀。虽然并非所有的病毒都对系统产生极其恶劣的破坏作用，但有时几种本来没有多大破坏作用的病毒交叉感染，也会导致系统崩溃等重大恶果。

(7) **产生的必然性**：计算机病毒存在的理论依据来自于冯·诺依曼结构及信息共享，从理论上讲如果要彻底消灭病毒，只有摒弃冯·诺依曼结构及信息共享，显然，此二者都是无法摒弃的。

(8) **非授权性**：从本质上说，计算机病毒是非授权的对程序体字符型信息进行加工的过程。一般正常程序的目的对用户是可见的、透明的。而病毒具有正常程序的一切特性，它隐藏在正常程序中，当用户调用正常程序时，它窃取到系统的控制权，先于正常程序执行，病毒的动作、目的对用户是未知的，是未经用户允许的。

14.3　计算机病毒的分类

从第一个病毒出世以来，病毒的数量仍在不断增加。虽然病毒数量繁多，然而万变不离其宗，其还是有规律可循的。为了更好地了解它们，可按照计算机病毒的特点及特性进行分类。计算机病毒的分类方法有许多种。下面介绍几种常用的分类方法。

14.3.1　按照计算机病毒攻击的操作系统分类

传统方式下，特定的病毒只能在特定的操作系统下运行，例如小球病毒是针对 IBM PC 及其兼容机上的 DOS 操作系统的。CIH 病毒只能在 Windows 95/98 环境下运行，在 DOS,

Windows 3.x 和 Windows NT 环境下则不能运行。“宏病毒”的出现则改写了这一观念，它可跨越 Windows 3.x、Windows 95/98/NT 和 Windows 2000/XP 多种系统环境，感染 Office 文件。

14.3.2 按照计算机病毒的链接方式分类

计算机病毒必须在隐蔽自身的同时被系统“合法”地调用执行，这一点是通过计算机病毒与系统内可执行文件建立链接来实现的。计算机病毒程序链接的对象是系统内的可执行文件，根据计算机病毒对这些文件的链接形式不同，病毒可分为如下 5 类。

1．源码型病毒

该病毒攻击高级语言编写的程序，在高级语言所编写的程序编译前插入到原程序中，经编译成为合法程序的一部分。这种病毒与用户的合法程序结合紧密，使得清除工作十分困难。由于这种病毒要求设计者对所要侵入的高级语言源程序的掌握程度非常高，编写困难，并且，可被这种病毒感染的程序对象也比较有限，因而，这种病毒并不多见。它往往隐藏在大型程序中，一旦被侵入，破坏性也非常大。

2．嵌入型病毒

这种病毒将自身嵌入到攻击目标中，代替宿主程序中不常用到的堆栈区或功能模块，而不是链接在它的首部或尾部。该病毒的特点与源码型病毒类似，难以编写，一旦侵入程序体后也较难消除，往往只能用破坏宿主程序的方法来清除病毒程序，这种病毒能够传染的对象也受到一定的限制。

3．外壳型病毒

寄生在宿主程序的前面或后面，并修改程序的第一个执行指令，使病毒先于宿主程序执行，这样随着宿主程序的使用而传染扩散。这种病毒最为常见，易于编写，传染的对象不受限制，传染性很强，因为这种病毒引起宿主程序长度的变化，所以也容易被检测出来，一般测试文件大小就可知。

4．操作系统型病毒

这种病毒在运行时，用自己的逻辑模块取代操作系统的部分合法程序模块。根据病毒自身的特点，被取代的操作系统合法模块在整个操作系统中的地位和作用以及病毒对操作系统模块取代方式等诸因素的不同，这类病毒的结构、破坏性以及传染速度都可以有很大差异。它可以具有很强的破坏力，导致整个系统的瘫痪。

5．译码型病毒

隐藏在微软 Office 等文档中，如宏病毒、脚本病毒，一般是解释执行此类病毒。

14.3.3 按照寄生方式和传染途径分类

人们习惯将计算机病毒按寄生方式和传染途径来分类。计算机病毒按其寄生方式大致可分为两类，一是引导型病毒，二是文件型病毒；它们再按其传染途径又可分为驻留内存型和不驻留内存型，驻留内存型按其驻留内存方式又可细分。混合型病毒集引导型和文件型病毒特性于一体。

引导型病毒就是通过磁盘引导区传染的病毒，主要是用病毒的全部或部分逻辑取代正常的引导记录，而将正常的引导记录隐藏在磁盘的其他地方。由于引导区是磁盘能正常使用的先决条件，因此，这种病毒在运行的一开始(如系统启动)就能获得控制权，其传染性较大。引导型病毒会去改写磁盘上的引导扇区(BOOT SECTOR)的内容，软盘或硬盘都有可能感染

病毒。还可能改写硬盘上的分区表(FAT)。引导型病毒是一种在 ROM BIOS 之后，系统引导时出现的病毒。该类病毒在操作系统启动之前运行，依托的环境是BIOS中断服务程序。它利用操作系统的引导模块放在某个固定的位置，并且控制权的转交方式是以物理地址为依据，而不是以操作系统引导区的内容为依据，因而病毒占据该物理位置即可获得控制权，而将真正的引导区内容搬家转移或替换，待病毒程序被执行后，将控制权交给真正的引导区内容，使得系统看似正常运转，而实际上病毒已隐藏在系统中，伺机传染、发作。引导型病毒几乎清一色都会常驻在内存中，差别只在于内存中的位置。所谓“常驻”，是指应用程序把要执行的部分在内存中驻留一份。这样就可不必在每次要执行它的时候都到硬盘中搜寻，以提高效率。

文件型病毒主要以感染可执行程序为主，病毒通常寄生在可执行程序中，一旦程序被执行，病毒也就被激活，并将自身驻留内存，然后设置触发条件，进行感染。这里的可执行文件不仅包括后缀为.com 和.exe 文件,还包括这些文件中可能使用的数据文件,如 Word 和 Excel 等软件使用的文档文件，因为这些数据文件中可能包括一些指令集，使得病毒可以利用这些指令集藏身并进行传播，这就是宏病毒。大多数文件型病毒都是常驻在内存中的。文件型病毒分为源码型病毒、嵌入型病毒和外壳型病毒。

混合型病毒是指既传染磁盘引导扇区又传染可执行文件的病毒，综合了引导型和文件型病毒的特性。此种病毒透过这两种方式来感染，更增加了病毒的传染性以及存活率。要清除这类病毒，必须同时解除文件上的病毒和主引导区的病毒，所以这类病毒清除比较困难。

14.3.4　三类特殊的病毒

从广义的病毒定义，逻辑炸弹、特洛伊木马和蠕虫都属于计算机病毒，但是这三类病毒与通常所说的病毒相比，又具有一定的特殊性。

逻辑炸弹(Logic bombs)，是指修改计算机程序，使它在某种特殊条件下按某种不同的方式运行。逻辑炸弹一般是通过授权用户故意安装的恶意代码软件，但并不进行自我复制。时间炸弹的一个著名例子是，一个不满意的雇员，设置了一个自动加载的程序，它周期性地监测组织的员工薪水表或个人数据库，如果发现他的名字不在职员名单中，就于两个星期后，删除客户数据。设置两周的时间滞后，是希望由于他的免职导致调查人员不能把他和删除所有记录的事情联系起来。

特洛伊木马泛指那些内部包含有为完成特殊任务而编制的代码的程序，一种潜伏执行非授权功能的技术。木马本身不进行自我复制。木马程序是一个表面上看起来有用的程序或命令，但当被用户执行后，它会执行一些恶意操作，如修改数据库、删除文件等。除此之外，有很多木马程序隐藏在你机器中，使你的机器可以被他人远程控制。这种远程控制程序可以被称为后门工具，它一般都采用 Client/Server 的模式。

计算机蠕虫程序是一种通过某种网络媒介——电子邮件、TCP/IP 等自身从一台计算机复制到其他计算机的程序。与病毒在文件之间进行传播不同，它们是从一台计算机传播到另一台计算机，从而感染整个系统。蠕虫程序在计算机之间进行传播时很少依赖(或者完全不依赖)人的行为。蠕虫程序不进行直接的破坏，不像一般病毒那样感染文件，它只是在计算机内存中精确地自我复制，并向网络上尽可能多的计算机发送自身的副本。这种自我复制每时每刻都在高速进行，当内存中蠕虫运行的进程数量增多到一定程度后，可能会耗尽系统资源，使系统不堪重负而崩溃，造成严重危害。

14.4 计算机病毒的命名

14.4.1 常用的命名方法

对病毒命名的目的是使人们快速、准确地辨识出该病毒，以便防范和清除。一种计算机病毒往往有多个名字，如“PE_KRIZ.3740”病毒，又称圣诞 CIH 病毒，圣诞节病毒(Christmas Virus)、W32.KRIZ病毒。通常人们在没有见到对某种病毒的确切描述和公认的命名或国际标准命名时，会根据病毒发作症状、病毒的工作机理、表现形式、内含的特征串、发作日期或时间、该病毒的发现地、病毒宣布的编写者的名称、病毒宣布的编写时间、病毒感染文件时文件增加的字节长度或病毒自身代码长度等各种特征为其命名。

14.4.2 国际上对病毒命名的惯例

国际上还有如下对病毒命名的惯例，一般惯例为前缀 + 病毒名 + 后缀。前缀表示该病毒发作的操作平台或者病毒的类型，而DOS 下的病毒一般是没有前缀的；病毒名为该病毒的名称及其家族；后缀一般可以不要，只是以此区别在该病毒家族中各病毒的不同，可以为字母，或者为数字以说明此病毒的大小。例如：“W97M.Melissa.BG”病毒，BG 表示在 Melissa 病毒家族中的一个变种，W97M 表示该病毒是一个 Word 97 宏病毒。“PE_KRIZ.3740”病毒， PE 是 Portable Execute 的缩写(与 Native Execute 的NE 相对照)，它表明这个病毒类型是32 位寻址的线性增强模型保护模式文件；“KRIZ”是这个病毒的真实名字，“3740”，表征着病毒的属性，即这个病毒恶性代码大小为 3740 字节。由于这个病毒是一个 Win32 Exec 型的病毒，所以也有人称它为 W32.KRIZ 病毒。表 14.1 给出了一些前缀的含义。

表 14.1 国际病毒命名惯例的前缀含义表

WM	Word 宏病毒，在 Word 6.0 和 Word 95(Word 7.0)下制作完成并传播发作，同样 Word 97(Word 8.0)或以上的 Word 下传播发作
W97M	Word 97 宏病毒，在 Word 97 下制作完成，并只在 Word 97 或以上版本的或以上的 Word 传播发作的病毒
XM	Excel 宏病毒，在Excel 5.0 和 Excel 95下制作完成并传播发作，同样，此种病毒也可以在 Excel 97 或以上版本传播发作
X97M	在 Excel 97 下制作完成的 Excel 宏病毒，也可以在 Excel 5.0 和 Excel 97 下传播发作
XF	Excel 程序(Excel Formula)病毒，此类病毒是用 Excel 4.0 把程序片断植入新的 Excel 文档中的
AM	在 Access 95 下制作完成并传播发作的 Access 宏病毒
A97M	在 Access 97 下制作完成并传播发作的 Access 宏病毒
W95	Windows 95 病毒，运行在 Windows 95 操作系统下，当然也可以运行在 Windows 98 下
Win	Windows 3.x 病毒，感染 Windows 3.x 操作系统的文件
W32	32 位 Windows 病毒，感染所有的 32 位 Windows 平台
WNT	32 位 Windows 病毒，但只感染 Windows NT 操作系统
W2K	32 位 Windows 病毒，特别是在 Windows 2000 或 XP 下的有破坏性的病毒
UNIX	运行在 UNIX 系统中的病毒或外壳脚本
HLLC	高级语言同伴(High Level Language Companion)病毒，他们通常是 DOS 病毒，通过新建一个附加的文件来传播
HLLP	高级语言寄生(High Level Language Parasitic)病毒，通常也是 DOS 病毒，寄生在主文件中
HLLO	高级语言改写(High Level Language Overwriting)病毒，通常是 DOS 病毒，以病毒代码改写主文件

(续表)

Trojan/Troj	特洛伊木马
VBS	Visual Basic Script 程序语言编写的病毒
AOL	美国在线(AOL)环境下特殊的木马，其目的通常为窃取 AOL 的密码等信息
PWSTEAL	窃取密码等信息的木马
Java	用 Java 程序语言编写的病毒
Perl	用 Perl 程序语言编写的病毒
PHP	用 PHP 程序语言编写的病毒
bat	批处理文件病毒或木马程序
PE	PE 是 Portable Execute 的缩写，它表明这个病毒类型是 32 位寻址的线性增强模型保护模式文件

14.5　计算机病毒的发展历程

随着计算机技术的发展，操作系统不断更新换代，反病毒技术的不断推动，计算机病毒也在不断发展，有一种观点认为其发展历程可以粗略地划分为以下四个阶段。

14.5.1　第一阶段

这一阶段可以认为在1986—1989年之间，这一时期是计算机病毒的萌芽和滋生时期。这一阶段的计算机病毒具有如下的一些特点：

(1) 病毒攻击的目标比较单一，或者是传染磁盘引导扇区，或者是传染可执行文件。

(2) 病毒程序主要采取截获系统中断向量的方式监视系统的运行状态，并在一定的条件下对目标进行传染。

(3) 病毒传染目标以后的特征比较明显，如磁盘上出现坏扇区，可执行文件的长度增加、文件建立的日期和时间发生变化，等等。这些特征容易被人工或查毒软件所发现。

14.5.2　第二阶段

第二阶段是在1989—1991年之间，这一阶段的计算机病毒具有如下特点：

(1) 病毒攻击的目标趋于混合型，即一种病毒既可传染磁盘引导扇区，又可能传染可执行文件。

(2) 病毒传染目标后没有明显的特征。

(3) 病毒程序往往采取了自我保护措施，如加密技术、反跟踪技术、隐蔽技术，制造障碍，增加人们分析和解剖的难度，同时也增加了软件检测、杀毒的难度。加密技术是一种防止静态分析的技术，分析者在不执行病毒的情况下，不能阅读加密过的病毒程序。反跟踪技术使得分析者无法动态跟踪病毒程序的运行。隐蔽技术是与计算机病毒检测技术相对应的，一般来说，有什么样的检测技术，就有什么样的隐蔽技术。采用隐蔽技术的病毒进入内存后，计算机用户不能轻易发现它的存在。例如4096病毒感染文件时，文件长度增加4096字节，但是当用 DIR 命令看时，由于病毒获得了系统的控制权，用户看到的还是原来长度、日期的文件。

(4) 出现了许多病毒变种，这些变种比原病毒具有更强的隐蔽性和破坏性。

14.5.3　第三阶段

第三阶段从1992年开始至1995年，这一时期是病毒的成熟发展阶段，病毒开始向多维化方向发展，即传统病毒传染的过程与病毒自身运行的时间和空间无关，而新型的计算机病毒

则将与病毒自身运行的时间、空间和宿主程序紧密相关，无疑导致了计算机病毒检测和消除的困难。多态病毒、伴随病毒等都是这一时期出现的。

(1) 伴随型病毒，伴随病毒通过创建一个新的扩展文件把自己附着在一个可执行文件上。具有代表性的是“金蝉”病毒，它感染EXE文件时生成一个和EXE同名的扩展名为COM的伴随体；它感染COM文件时，改原来的COM文件为同名的EXE文件，再产生一个原名的伴随体，文件扩展名为COM。如果有两个同名但扩展名不同的文件，一个是EXE文件，一个是COM文件，操作系统总会调用COM文件。这样，在DOS加载文件时，病毒就取得控制权。这类病毒的特点是不改变原来的文件内容、日期及属性，解除病毒时只要将其伴随体删除即可。

(2) 多态病毒，又被称为多形病毒、千面人病毒，是指采用特殊加密技术编写的病毒，能够自身加密，加密的病毒经常隐藏它的特征，它每次感染可执行文件时，都产生一个解密程序，生成不同的病毒特征，所以在完全多态性病毒的主要不同样本中，甚至不存在连续两个字节是相同的，从而避开反病毒软件。多态病毒是一种混合性病毒，它既能感染引导区又能感染程序区。

(3) 出现了一些能生产病毒的软件工具，只要是具备一点计算机知识的人，利用病毒生成工具就可以轻易地制造出算法各异、功能各异的计算机病毒，而且可以设计出非常复杂的具有多形性特征的病毒。

14.5.4 第四阶段

20世纪90年代以后，随着Windows操作系统的流行、Internet、远程访问服务的开通，病毒表现出如下特征：

(1) 产生了许多以Windows平台为特定目标的计算机病毒，包括一些针对像Windows 95这种32位操作系统的计算机病毒，如CIH。1996年，随着Windows和Windows 95的日益普及，利用Windows进行工作的病毒开始发展，它们修改NE和PE文件，这类病毒的机制更为复杂，它们利用保护模式和API调用接口工作，清除方法也比较复杂。

(2) 1996年，随着Windows、Office软件的流行，宏病毒广泛传播。宏病毒是使用某个应用程序自带的宏编程语言编写的病毒，与以往的病毒不同，宏病毒冲破了以往病毒在单一平台上传播的局限，容易编写，容易传播。根据国内外的统计，宏病毒的感染率已高达90%以上。

(3) 病毒流行面更加广泛，突破了地域的限制。1996年下半年随着国内Internet的大量普及，E-mail的使用，病毒出现了通过E-mail扩散的增长趋势。出现了大量夹杂于E-mail内的Word宏病毒、蠕虫病毒，甚至邮件本身就是一个病毒。一项由国际计算机安全协会(International Computer Security Association，ICSA)所公布的《2000年度病毒传播趋势报告》显示，电子邮件已经跃升为计算机病毒最主要的传播介质，由它引起的病毒感染率由1998年的32%，1999年的56%，已经大幅增长至2000年的87%。经由磁盘、网络下载的病毒感染率则急剧下降。为此，国际计算机安全协会新定义了一种病毒类型——通过E-mail大量散播(Mass Mailers)型病毒。这一类型的病毒有Happy 99(Win32/Ska)病毒、梅莉莎病毒、I LOVE YOU病毒及NAVIDAD病毒等。

(4) 随着Internet网发展，产生了HTML和Java病毒。

(5) 病毒综合具有多种特性，如梅莉莎病毒就兼具宏病毒、蠕虫、电子邮件病毒的特性。

这一时期的病毒的最大特点是利用 Internet 作为其主要传播途径，Internet 的传播方式包括 WWW 浏览、IRC 聊天、FTP 下载、BBS 论坛等。因而，病毒传播快、隐蔽性强、破坏性大。

14.6　计算机病毒的基本原理

14.6.1　计算机病毒的逻辑结构

计算机病毒是人为编制的一种计算机程序，这种程序有其自身的结构特点。计算机病毒程序一般包括 3 个功能模块：病毒的引导模块、传染模块、破坏(或表现)模块，下面分别加以描述。

1．病毒的引导模块

引导模块的作用是当病毒的宿主程序开始工作时将病毒程序从外存引入内存，使其与宿主程序独立，并且使病毒的传染模块和破坏模块处于活动状态，以监视系统运行。当出现满足触发条件的情况时，病毒就会按设计者的意图向系统发动进攻。某些病毒程序的引导模块还负责将分别存储的病毒程序链接在一起，进行重新装配，构成完整的病毒体，使其投入运行。

2．病毒的传染模块

传染模块负责将病毒传染给其他计算机程序，使病毒向外扩散。病毒的传染模块由两部分组成：病毒传染的条件判断部分和病毒传染程序主体部分，其中，条件判断部分负责判定被传染对象是否具备被传染条件，传染程序主体部分负责将病毒的再生体与宿主程序链接，完成病毒传染工作。

3．病毒的破坏(表现)模块

该模块是病毒的核心部分，它体现了病毒制造者的意图。由于恶作剧型病毒没有明显的破坏意图，病毒带来的破坏性较小，所以这类病毒的破坏模块可被称为表现模块。病毒的破坏模块也是由两部分组成：病毒破坏的条件判断部分和破坏程序主体部分。条件判断部分时刻判断运行过程中是否出现了满足病毒触发条件的情况，如某一特定日期、某一特定的用户键组合，当条件满足时，病毒才调用破坏程序的主体部分。病毒破坏程序的主体部分负责实施病毒的表现或破坏工作，如删除数据、改写文件、发出异常声音或图像等。

总体来说，计算机病毒的引导模块是传染模块和破坏模块的基础，传染模块和破坏模块依赖引导模块进入计算机系统。计算机病毒的逻辑结构可用图14.1表示。

引导模块
传染模块：传染的条件判断部分 传染的程序主体部分
破坏模块：破坏的条件判断部分 破坏的程序主体部分

图 14.1　计算机病毒的逻辑结构

需要说明的是，并不是所有的计算机病毒都由这三大模块组成，有的可能会没有引导模块，有的可能没有破坏模块，有的病毒三个模块之间的界限并不明显。

绝大多数计算机病毒，都是通过截取操作系统的系统调用功能进行自身的繁殖、传播和破坏的。在DOS下，操作系统的功能是通过各种软中断来实现的，如大家都知道INT 21H是DOS中断，INT 13H 和 INT 10H是BIOS中的磁盘中断和视频中断。病毒程序常常通过修改一些标准的系统中断入口地址，使该中断指向病毒程序的传染模块、破坏模块的入口，为病毒传染模块和破坏模块的运行做准备。

在 Windows 环境下，PE病毒的基本机制是相同的，但是其程序结构更复杂，技巧性更强，Win32 环境下的系统功能调用，是通过调用动态链接库(如 kernel32.DLL)中的 API 函数实现的，它实际上是以一种新的方法代替了 DOS 中用软中断的方式。

14.6.2 计算机病毒的工作流程

当计算机病毒仅仅寄生在存储介质上时，被称为静态病毒。当病毒随宿主程序的运行而进入内存、正处于运行状态或能够立即获得运行权时，被称为动态病毒。绝大多数引导型病毒、文件型病毒以及宏病毒具有相似的工作流程，现分述如下。

14.6.2.1 引导型病毒的工作流程

引导型病毒寄生在磁盘引导区，系统启动后，将磁盘引导区的内容读入内存某处，病毒代码就随着进入了内存。接着系统执行引导区内容，这时，最先被执行的是病毒的引导模块，引导模块将全部的病毒代码驻留到内存某处，并对这一区域实行保护。然后，引导模块会修改系统的参数，为病毒的传染和破坏设置触发条件。最后，病毒会执行系统的正常引导过程，完成系统的引导工作，否则，它会立刻暴露。在用户看来，一切正常，实际上，病毒已驻留内存，时刻监视系统的运行，等待适当的触发条件将其激活，其流程如图 14.2 所示。

14.6.2.2 文件型病毒的工作流程

当运行带毒的可执行文件时，病毒的引导模块就会自动装入内存并获得执行权，将整个计算机病毒程序送入系统，完成病毒程序的安装，然后修改系统的中断向量，使之分别指向病毒的传染模块和破坏模块，于是，病毒的传染模块和破坏模块从静态变成动态，可以对系统进行监视，在一定条件下就可以进行传染和破坏。最后，病毒把执行权交给可执行文件，以使得用户觉察不到病毒的存在，其工作流程如图14.3 所示。

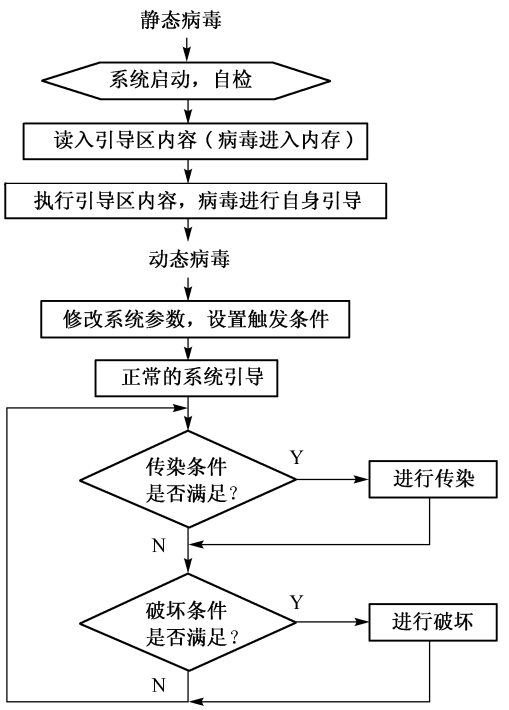


图 14.2 引导型病毒工作流程

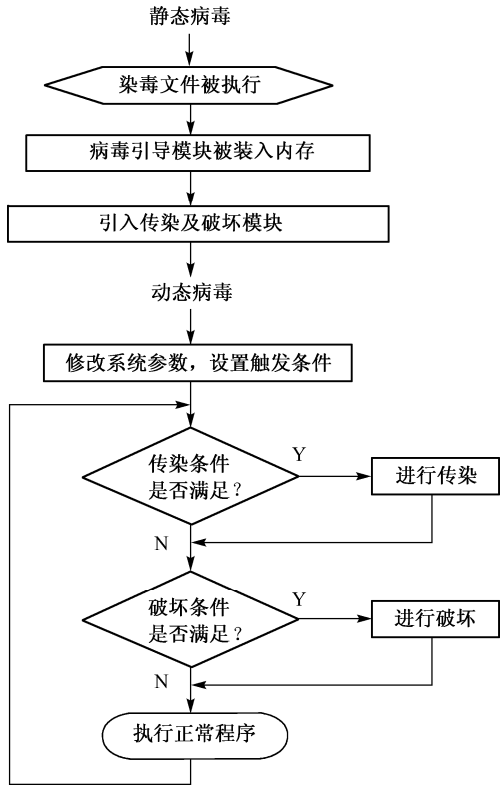


图 14.3 文件型病毒工作流程

14.6.2.3　宏病毒的工作流程

宏病毒是一类主要感染 Word 文档和文档模板文件的病毒。其传播和引导过程与寄生在.com 和.exe 文件中的病毒有很大不同。宏病毒寄生在 Office 文档所包含的宏中，它一般不修改这些文档的头部，而只修改这些文档中包含的宏，其工作流程比较简单。

当被病毒感染的Office文档被打开时，只要文档中的宏被运行，隐藏在其中的病毒就被激活了，它立即开始监视用户对Office软件的一切操作，进行触发条件判断，条件满足就会发作。软件关闭后，宏病毒随之退出，不会驻留在内存中，其流程如图14.4所示。

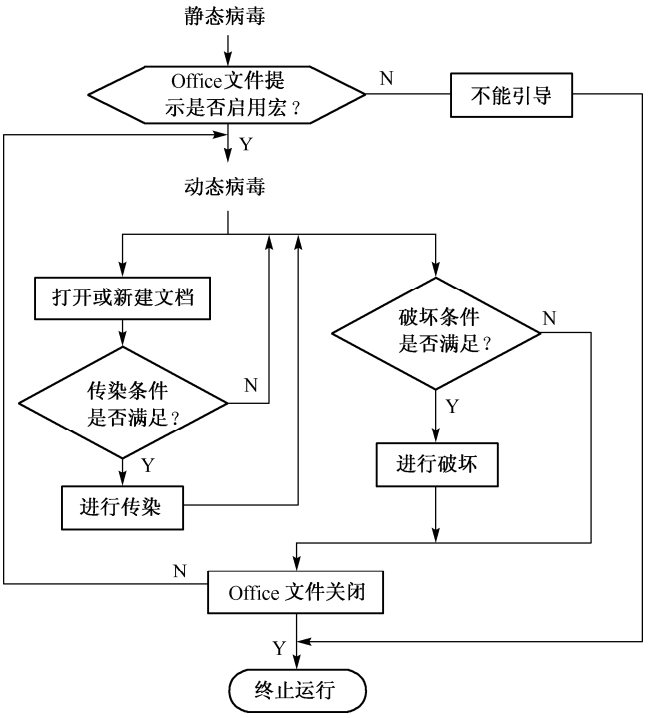


图 14.4　宏病毒工作流程

14.6.3　计算机病毒存在的理论基础

计算机病毒存在的理论依据来自于冯・诺依曼结构及信息共享，冯・诺依曼结构的计算机允许程序及系统自我复制，而信息共享为这种复制的传播提供了途径。早在1949 年，也就是世界上第一台计算机诞生后4年，计算机的创始人冯・诺依曼(John von Neumann)就发表了“复杂自动机器的理论和结构”的论文，指出计算机程序可以在内存中进行复制即“程序复制机理”的理论。在这以后，许多计算机人员发展和应用了程序自我复制理论。尤其是在 1983 年，弗雷德・科恩(Fred Cohen)博士研制出一种运行过程中可以复制自身的破坏性程序，莱恩・阿德勒曼(Len Adleman)将它命名为计算机病毒(Computer Virus)，并通过一系列实验演示，从而在实验上证实了计算机病毒的实际存在。

根据 Fred Cohen 的对计算机病毒的定义，计算机病毒是一种程序，它用修改其他程序的方法将自身的精确复制或可能演化的副本放入其他程序，从而感染其他程序，由于这种感染特性，病毒可以在信息流的途径中传播，从而破坏信息的完整性。Cohen 于 1988 年又撰文强

调：病毒不是利用操作系统运行的有错误和缺陷的程序，病毒是正常的用户程序，它们仅仅使用了那些每天都被使用的正常操作。

14.7 特洛伊木马

14.7.1 木马的定义

关于木马的定义在各类安全文件中都有涉及，其中最广为人知的定义是在RFC1244 中给出的：“特洛伊木马是这样一种程序，它提供了一些有用的，或者仅仅是有意思的功能。但是通常要做一些用户不希望的事，诸如在你不了解的情况下复制文件或窃取密码”。换言之，木马就是指那些内部包含为完成特殊任务而编制的代码的程序，这些特殊功能处于隐蔽状态，执行时不为人知。

14.7.2 木马的特性

据不完全统计，目前世界上有着上千种木马程序。虽然这些程序使用不同的程序语言进行编制，在不同的平台环境下运行，发挥着不同的作用，但是它们有着许多共同的特征。

伪装性：程序将自己的服务器端伪装成合法程序，并且诱惑被攻击者执行，使木马代码会在未经授权的情况下装载到系统开始运行。

隐藏性：木马程序同病毒程序一样，不会暴露在系统进程管理器内，也不会让使用者察觉到木马的存在。具体表现为隐蔽运行、隐蔽启动、隐蔽通信和隐蔽存在。通常情况下，木马在系统中运行时外观上是隐蔽的，不会打开程序窗口，不会在任务栏中显示，也不发出声音以及其他容易被用户发现的动作。隐蔽运行的主要方面是对运行木马在内存中的运行存在形式进行隐蔽和欺骗。木马植入目标系统后会在系统每次启动时隐蔽地启动，或者在系统运行时在一定的触发条件满足时启动运行。木马通常要与控制端通信。木马一般会对其通信加以隐蔽。木马植入目标系统后会在宿主磁盘空间中生成自己的木马文件。木马为了不被当做可疑文件，一般会在木马文件的命名上、文件类型的外观上、文件属性上欺骗目标系统的用户。

破坏性：通过远程控制，攻击者可以通过木马程序对系统中的文件进行删除、编辑等破坏操作。

窃密性：木马程序最大的特点是可以窥视被入侵的计算机上的所有资料，不仅包括硬盘上的文件，还包括显示器的画面、使用者在操作计算机过程中输入的所有命令。

14.7.3 木马的组成

对木马程序而言，它一般包括两个部分：客户端和服务器端。

服务器端安装在被控制的计算机中，它一般通过电子邮件或其他手段让用户在其计算机中运行，以达到控制该用户计算机的目的。

客户端程序是控制者所使用的，用于对受控的计算机进行控制。服务器端程序和客户端程序建立起连接就可以实现对远程计算机的控制了。

木马运行时，首先服务器端程序获得本地计算机的最高操作权限，当本地计算机连入网络后，客户端程序可以与服务器端程序直接建立起连接，并可以向服务器端程序发送各种基本的操作请求，并由服务器端程序完成这些请求，也就实现对本地计算机的控制了。

木马本身不具备繁殖性和自动感染的功能。

14.7.4　木马的类型

根据木马功能的差异，木马基本上可以分为以下几类：

远程访问型木马：是现在最广泛的特洛伊木马，它可以访问受害人的硬盘，并对其进行控制。这种木马用起来非常简单，只要某用户运行一下服务端程序，并获取该用户的IP地址，就可以访问该用户的计算机。这种木马可以使远程控制者在本地机器上做任意的事情，比如键盘记录、上传和下载功能、截取屏幕，等等。这种类型的木马有著名的BO(Back Office)和国产的冰河等。

信息收集型木马：此类木马被植入目标系统并运行后，木马会记录或收集系统各类重要的信息，如窃取用户名、系统口令、ICQ 密码，记录系统操作、键盘按键情况等，并把其通过一定的方式发送给特定的攻击者。如密码发送型木马的目的是找到所有的隐藏密码，并且在受害者不知道的情况下把它们发送到指定的信箱。键盘记录型木马非常简单，它们只做一种事情，就是记录受害者的键盘敲击，并且在 LOG 文件里做完整的记录。

系统配置修改型木马：此类木马在目标系统上运行后修改系统的配置。例如共享(Share all)木马，其运行修改注册表使磁盘共享随后关闭。

后门型木马(Backdoor)：此类木马植入目标系统运行后打开一特定的后门以便攻击者进入此系统。例如 Win-ftp 木马，此木马运行后打开 TCP 端口 21，控制端可以由此端口进入系统。主要的后门类型有：Ftp Server, Proxy Server, HTTP Server, Telnet Server 等。

毁坏型木马的唯一功能是毁坏并且删除文件。这使它们非常简单，并且很容易被使用。它们可以自动地删除用户计算机上的所有的.DLL, INI 或 EXE 文件。

14.8　计算机病毒防治对策

“知己知彼，百战不殆”，了解计算机病毒的目的就是为了对计算机病毒进行积极地防治。在对计算机病毒特性、原理初步介绍的基础上，本节将介绍反病毒的原理、技术和策略。

14.8.1　怎样发现计算机病毒

计算机病毒发作时，通常会出现以下几种情况，这样我们就能尽早地发现和清除它们。

- 计算机运行比平常迟钝。
- 程序载入时间比平常久。
- 对一个简单的工作，磁盘似乎花了比预期长的时间。
- 不寻常的错误信息出现。
- 由于病毒程序的异常活动，造成对磁盘的异常访问。当你没有存取磁盘，但磁盘指示灯却亮了，表示计算机这时已经受到病毒感染了。
- 系统内存容量忽然大量减少。
- 磁盘可利用的空间突然减少。
- 可执行程序的大小改变了。
- 由于病毒可能通过将磁盘扇区标记为坏簇的方式把自己隐藏起来，磁盘坏簇会莫名其妙地增多。
- 程序同时存取多部磁盘。

- 内存内增加来路不明的常驻程序。
- 文件、数据奇怪的消失。
- 文件的内容被加上一些奇怪的资料。
- 文件名称、扩展名、日期、属性被更改过。
- 打印机出现异常。
- 死机现象增多。
- 出现一些异常的画面或声音。

异常现象的出现并不表明系统内肯定有病毒，仍需进一步的检查。

14.8.2 计算机病毒防治技术

病毒技术与反病毒技术存在着相互对立、相互依存的关系，它们都在彼此的较量中不断发展，当然，从总体上，反病毒技术要滞后于病毒技术。计算机病毒的防治技术可以分成四个方面，即检测、清除、免疫和预防。

14.8.2.1 计算机病毒检测技术

计算机感染病毒后，会引起一系列变化，检测正是以此为依据的。在病毒与反病毒的长期斗争中，病毒与反病毒技术都在不断发展提高，但病毒对反病毒技术永远都是超前的，从而使反病毒软件不可能检测到所有未知病毒。病毒检测常用的方法有：比较法、搜索法、分析法、行为检测法、感染实验法、软件模拟法等。

1．比较法

比较法是进行原始的或正常的特征与被检测对象的特征比较。由于病毒的感染，会引起文件长度和内容、内存、中断向量以及系统调用的变化，从这些特征的比较中可以发现异常，从而判断病毒的有无。比较法的好处是简单、方便，不需专用软件。缺点是无法确认计算机病毒的种类名称。另外，可能会存在以下问题：变化可能是合法的或是偶然因素造成的，有些病毒感染文件并不带来文件长度的改变，这就需要进一步地分析或采用其他的检测手段。

2．病毒校验和法

计算出正常文件的程序代码的校验和，并保存起来，可供被检测对象对照比较，以判断是否感染了计算机病毒。这种技术可侦测到各式的计算机病毒，包括未知病毒，但误判率高，无法确认病毒种类，无法侦测隐形计算机病毒。

3．搜索法

搜索法是用每一种计算机病毒体含有的特定字符串对被检测的对象进行扫描。如果在被检测对象内部发现了某一种特定字节串，就表明发现了该字节串所代表的计算机病毒。特征串选择的好坏，对于病毒的发现具有决定作用。当特征串选择得很好时，发现病毒的概率很高，即使对计算机病毒了解不多的人也能用它来发现计算机病毒。但是如何提取特征串，则需要足够的有关知识。搜索法也有自身的一些缺点，如被扫描的文件很长时，扫描所花时间也越多；不容易选出合适的特征串；计算机病毒代码库未及时更新时，无法识别出新的计算机病毒；不易识别变形计算机病毒等。不管怎样，搜索法仍被病毒检测软件广泛采用，是今天使用最为普遍的计算机病毒检测方法。

4．分析法

分析法的使用人员主要是反计算机病毒的技术专业人员。专业人员借助自己掌握的关于计算机及病毒的广泛知识及专用工具，分析计算机是否染毒，确认计算机病毒的类型，搞清楚病毒体的大致结构，从中提取特征识别用的字节串或特征字，用于增添到计算机病毒代码库或详细分析计算机病毒代码，制定相应的防杀计算机病毒方案。由于很多计算机病毒采用了自加密、反跟踪等一些隐蔽技术，使得计算机病毒的分析工作经常是冗长和枯燥的。对计算机病毒的分析同时是计算机病毒检测工作中不可缺少的重要环节。

5．行为监测法

由于病毒在感染及破坏时都表现出一些共同行为，而且比较特殊，这些行为在正常程序中比较罕见，因此可通过监测这些行为来检测病毒的存在与否。通常这些行为包括占用 INT 13H，对Boot扇区进行攻击，对 COM 及 EXE 文件做写入动作、病毒程序与宿主程序切换。该方法不仅可检测已知病毒，而且可预报未知病毒，但是有可能误报，而且不能确定病毒名称，使用有一定难度。

6．病毒行为软件模拟法

软件模拟法专门用来对付多态病毒，多态病毒在每次传染时都通过加密变化其特征码，使得搜索法失效。该方法监视病毒运行，待病毒自身密码破译后，再进行代码的分析。

7．感染实验法

该法利用病毒最重要的基本特性——感染特性。检测时，先运行可疑系统中的程序，再运行一些确切知道不带毒的正常程序，然后观察这些正常程序的长度和检验和，如果发现有变化，可断言系统中有病毒。

14.8.2.2　计算机病毒的清除

病毒的清除是指将染毒文件的病毒代码摘除，使之恢复为可正常运行的健全文件。病毒消除可手工进行，也可用专用软件杀毒。无论哪种方式，都是一种危险的操作。因为完全将病毒代码从染毒程序中摘除而不破坏原来的程序是困难的，弄不好会使原来的程序遭到彻底破坏无法恢复。而且，如果染毒过程中已将原程序覆盖或破坏，则是无法恢复的。

对于引导型病毒，由于其攻击部位主要在磁盘主引导区、Boot 区或FAT 表。因此要针对不同情况重写相应的区，对于文件型病毒，则必须仔细识别病毒特征代码，将特征代码从染毒程序中去掉。有时，文件型病毒可能被多次交叉感染，这样在去除病毒时还需辨明其感染的次序，并根据感染次序一层层剥除。有些病毒还要经过多次重复实验，方可找到感染次序。

14.8.2.3　计算机病毒的预防

当计算机染毒后，虽然可以采取一些技术措施进行检测和清除，但都是一些事后补救工作，给我们增添了一些不必要的麻烦，而且病毒可能已经造成一些不可挽回的损失。采取一些防御措施，防患于未然，是一条积极的途径。可以从技术和管理上采用如下预防措施：

(1) 经常进行数据备份，特别是一些非常重要的数据及文件，以免被病毒侵入后无法恢复。

(2) 对新购置的计算机、硬盘和软件，先用查毒软件检测后方可使用。

(3) 尽量避免在无防毒软件的机器上，使用可移动磁盘，以免感染病毒。

(4) 对计算机的使用权限进行严格控制，禁止来历不明的人和软件进入系统。

(5) 采用一套公认最好的驻留式防病毒软件，以便在对文件和磁盘操作时，进行实时监控，及时控制病毒的入侵。

(6) 选择一套公认最好的防毒软件，并及时、可靠地升级反病毒产品。

14.8.2.4 计算机病毒的免疫

病毒免疫原理是根据病毒签名来实现的，由于有些病毒在感染其他程序时要先判断是否已被感染过，即欲攻击的宿主程序是否已有相应病毒签名，如有则不再感染。因此，可人为地在健康程序中进行病毒签名，起到免疫效果。

总之，由于计算机病毒从理论上无法根除，反病毒技术与病毒技术的较量是一场长期的斗争。虽然，彻底防治计算机病毒是不可能的，但我们可以通过采取各种积极有效的措施预防、检测和清除病毒，把危害降低到尽可能低的程度。

思考和练习题

(1) 计算机病毒有哪些特征？
(2) 按照计算机病毒的链接方式可以把计算机病毒分为几类？
(3) 木马与狭义计算机病毒的联系和区别是什么？
(4) 木马与远程控制软件的联系和区别是什么？
(5) 蠕虫与狭义计算机病毒的联系和区别是什么？
(6) 计算机病毒的逻辑结构是怎样的？
(7) 计算机病毒存在的理论基础是什么？

实践/实验题

(1) 到互联网上下载一个防病毒软件，安装并运行，研究该软件，对该软件的功能做出评价，说明喜欢哪些功能，不喜欢哪些功能。
(2) 到互联网上查找最近 90 天的病毒资料，写一篇短文描述这些病毒是如何传播的，有哪些危害，应该采取哪些防御措施。
(3) 到互联网上查找一个木马的资料，写一篇短文描述该木马的原理，应该采取哪些措施来防御。

第 15 章　入侵检测技术

在前面的各章中我们介绍了加密、消息摘要、数字签名、身份鉴别、访问控制、安全协议、防火墙、防病毒等网络安全产品与技术，这些产品和技术都属于预防(Prevention)和防护(Protection)的措施。预防性安全措施通常采用严格的访问控制和数据加密策略来防护，这些措施都是以减慢交易为代价的，而且在复杂系统中，这些策略是不充分的。比如身份鉴别可以拒绝未授权用户的访问，但黑客可以通过口令攻击、密码监听等手段获取系统中的未授权信息。防火墙也是入侵者的目标，它本身并不是牢不可破的，防火墙必须配置正确，否则，攻击者就可能穿透防火墙；其次，防火墙对来自内部的攻击无能为力。1999年CSI/FBI(Computer Security Institute/Federal Bureau of Investigation)指出，82%的损失是内部威胁造成的。正是在这样的背景下，入侵检测(Intrusion Detection)技术开始出现并逐渐成为研究的热点。

15.1　入侵检测概述

15.1.1　入侵检测的概念

P^2DR 模型最早由ISS 公司提出，后来又出现了很多在此模型上的变种，但实质内容并没有发生变化。入侵检测就是P^2DR模型中的检测。信息系统只有两种状态：正常状态和异常状态，如图 15.1 所示，对应的安全工作也只有两个结果：出事和不出事。信息系统经过建设和启动，进入正常运转状态，检测机制对信息系统进行监测，当发现状态异常时，对系统及时进行状态调整，使系统恢复正常状态。检测是静态防护转化为动态防护的关键，是动态响应的依据，是落实和强制执行安全策略的有力工具。

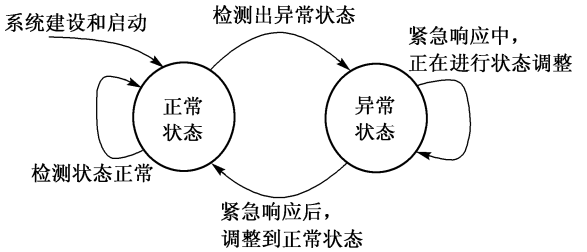


图 15.1　信息安全的两态论

NSTAC(National Security Telecommunications Advisory Board，美国国家安全通信委员会)的 IDSG(Intrusion Detection Sub-Group，入侵检测小组)是一个由美国总统特许的保护国家关键基础设施的小组。IDSG 于1997年给出了如下定义：入侵是对信息系统的非授权访问及(或)未经许可在信息系统中进行操作。入侵检测是对(网络)系统的运行状态进行监视，对企图入侵、正在进行的入侵或已经发生的入侵进行识别的过程。

进行入侵检测的软硬件组合便是**入侵检测系统**(Intrusion Detection System，IDS)。IDS 依照一定的安全策略，对网络、系统的运行状况进行监视，尽可能发现各种攻击企图、攻击行

为或者攻击结果，以保证网络系统资源的机密性、完整性和可用性。我们可以做一个形象的比喻：假如防火墙是一幢大楼的门卫，那么 IDS 就是这幢大楼里的监视系统。一旦小偷爬窗进入大楼，或内部人员有越界行为，只有实时监视系统才能发现情况并发出警告。

入侵检测系统提供对内部攻击、外部攻击和误操作的实时保护，这些主要通过以下任务来实现：

- 监视、分析用户计算机和网络的运行状况，查找非法用户和合法用户的越权操作。
- 检测系统配置的正确性和安全漏洞，并提示系统管理员修补漏洞。
- 识别已知活动模式的攻击行为并报警。
- 异常行为模式的统计分析。
- 评估重要系统和数据文件的完整性。
- 对操作系统进行审计跟踪管理，并识别用户违反安全策略的行为。

15.1.2 入侵检测的起源和发展

美国国防部在20世纪70年代支持“可信信息系统”的研究，可信系统研究成果的一系列文档被称为“彩虹系列”，因为它们的封皮颜色不同。最终把审计机制纳入《可信计算机系统评估准则》(TCSEC，橘皮书)C2级以上系统的要求之一。可信系统的定义是能够提供足够的硬件和软件，以确保系统同时处理一定范围内的敏感或分级信息。其中的一份文档《理解可信系统中的审计指南》被称为“褐皮书”，是说明可信系统的审计的。

1980年4月，James P. Anderson写了一份研究报告“计算机安全威胁监控与监视”(Computer Security Threat Monitoring and Surveillance)，这份报告被公认为入侵检测的开创性工作。他在报告中提出了精简审计的概念、风险和威胁分类方法、利用审计跟踪数据监视入侵活动的思想。

从 1984 年到 1986 年，乔治敦大学的 Dorothy Denning 和 SRI/CSL 的 Peter Neumann，研究和发展了一个实时入侵检测系统模型，命名为 IDES(入侵检测专家系统)，在 IDES 中实现了异常检测和误用检测。该项研究导致了随后几年内一系列系统原型的研究，如 Discovery, Haystack, MIDS, NADIR, NSM, Wisdom and Sense 等。

1990 年，加州大学戴维斯分校的 L. T. Heberlein 等人开发出了 NSM(Network Security Monitor)，该系统第一次直接将网络流作为审计数据来源，因而可以在不将审计数据转换成统一格式的情况下监控异种主机。入侵检测系统从此分为两大阵营：基于网络的 IDS 和基于主机的 IDS。

15.2 入侵检测系统的功能组成

简单地说，入侵检测系统包括三个功能部件：信息收集、信息分析和结果处理。

15.2.1 信息收集

入侵检测的第一步是信息收集，收集内容包括系统、网络、数据及用户活动的状态和行为。入侵检测在很大程度上依赖于收集信息的可靠性和正确性。因此，对于信息收集有以下原则：

- 需要在计算机网络系统中的若干不同关键点，不同网段和不同主机，收集信息，并且尽可能扩大检测范围，因为从一个源收集来的信息有可能看不出疑点。

- 要保证用于检测系统的软件的完整性，也就是说入侵检测系统软件本身应具有相当强的坚固性，防止被篡改而收集到错误的信息。
- 在一个环境中，审计信息必须与它要保护的系统分开来存储和处理，防止入侵者通过删除审计记录来使入侵检测系统失效，或者通过修改入侵检测器的结果来隐藏入侵的存在，同时要减轻操作系统执行入侵检测任务带来的操作负载。

按照数据来源可以把入侵检测系统分为基于主机的、基于网络的和混合型的三类。

基于主机的入侵检测系统获取数据的依据是系统运行所在的主机，保护的目标也是系统运行所在的主机。基于主机的监测收集在操作系统层的来自计算机内部的数据，包括操作系统审计跟踪信息和系统日志；来自主机上运行着的应用程序的数据，包括应用程序事件日志和其他存储在应用程序内部的数据；以及对系统对象的修改。

基于网络的入侵检测系统获取的数据是网络传输的数据包，保护的是网络的运行。

混合型的入侵检测系统则同时具有基于主机和基于网络的入侵检测系统的功能。

15.2.2　信息分析

我们从网络和主机日志获得信息后，如何来分析和判断呢？入侵检测的分析方法，主要可以分为异常检测(Anomaly Detection)和误用检测(Misuse Detection)。

异常检测首先总结正常操作应该具有的特征，被称为用户轮廓，当用户活动与正常行为有重大偏离时即被认为是入侵。

误用检测收集非正常操作的行为特征，也被称为特征检测(Signature Detection)，建立相关的特征库，当监测的用户或系统行为与库中的记录相匹配时，系统就认为这种行为是入侵。

第三种 IDS 被称为**基于规范的检测**(Specification-based Detection)，基于规范的检测与异常检测类似，都是检测与正常行为的偏离，但是异常检测系统把机器学习得到的参数作为正常行为的标准，而基于规范的检测系统人工制定期望的合法系统行为标准。它的优点是可以避免异常检测因为缺乏对以前正常行为的学习而造成的虚警，缺点是人工制定规范非常耗时。

完整性分析主要关注某个文件或对象是否被更改，这经常包括文件和目录的内容及属性，它在发现被更改的、被安装木马的应用程序方面特别有效，可以作为入侵检测的辅助性手段。

许多商业 IDS 实际上包含了以上所有分析方法，可以被称为混合 IDS(Hybrid IDS)。

15.2.3　结果处理

入侵检测可以达到两方面的目标，一方面是它可以提供可说明性，指从给定的活动或事件中，找到相关责任方的能力。建立可说明性的目标是获得补偿或针对责任方追究相关法律责任。另一方面，它可以采取一些积极的反应措施，如生成报告、发出警报，提示系统管理员修改目标机系统或入侵检测系统。

15.3　基于主机及基于网络的入侵检测系统

15.3.1　基于主机的入侵检测系统

15.3.1.1　基于主机的入侵检测系统的数据源

基于主机的入侵检测系统正在由于内部人员的威胁变得更重要，基于主机的入侵检测系统主要收集的信息是操作系统事件日志和关系数据库、Web 服务器等应用程序的日志，从这些信息中会发现入侵的行为。

1．系统或网络的日志文件

黑客经常在系统日志文件中留下他们的踪迹，因此，充分利用系统和网络日志文件信息是检测入侵的必要条件。日志文件中记录了各种行为类型，每种类型又包含不同的信息，例如记录“用户活动”类型的日志，就包含登录、用户ID改变、用户对文件的访问、授权和认证信息等内容。显然，对用户活动来讲，不正常的或不期望的行为就是重复登录失败、登录到不期望的位置以及非授权的企图访问重要文件，等等。

2．非正常的程序执行

网络系统上的程序执行一般包括操作系统、网络服务、用户启动的程序和特定目的的应用。每个在系统上执行的程序由一到多个进程来实现。一个进程的执行行为由它运行时执行的操作来表现，操作执行的方式不同，它利用的系统资源也就不同。操作包括计算、文件传输以及与网络上其他进程的通信。一个进程出现了不期望的行为可能表明黑客正在入侵你的系统。

3．系统目录和文件的异常变化

网络环境中的文件系统包含很多软件和数据文件，包含重要信息的文件和私有数据文件经常是黑客修改或破坏的目标。入侵者经常替换、修改和破坏他们获得访问权的系统上的文件，同时为了隐藏系统中他们的表现及活动痕迹，都会尽力去替换系统程序或修改系统日志文件。目录和文件中的不期望的改变(包括修改、创建和删除)，特别是那些正常情况下限制访问的，很可能就是一种入侵产生的指示和信号。

15.3.1.2 基于主机的入侵检测系统的系统结构

基于主机的入侵检测系统通常是基于代理的，代理是运行在目标系统上的可执行程序，与中央控制计算机(命令控制台)通信。本节给出两种典型的基于主机的入侵检测代理结构，分别称为集中式结构和分布式结构。这两者的差别是，在集中式的结构中，原始数据在分析之前要先发送到中央位置，如图 15.2 所示。在分布式结构中，原始数据在目标系统上实时分析，只有告警命令被发送给控制台，如图 15.3 所示。每种方式都各有优缺点，最好是能提供这两种类型的处理。

在集中式结构中，目标代理把审计子系统产生的审计数据集中送到命令控制台，检测引擎对数据进行处理，一方面创建日志，保留原始数据，供诉讼使用，另一方面对数据进行分析。如果发现某种入侵行为，则产生告警信息，通知安全人员。响应子系统根据收到的告警或者安全人员的指令完成相应的响应。最后把告警信息存储在数据库中，供用户或企业进行数据辨析。集中式结构的优点是对目标机的性能影响很小，因为分析是在目标机外进行的；有大量的用于支持起诉的原始数据，并且可用于行为的统计分析。它的缺点是不能进行实时检测，不能实时响应，而且将大量的原始数据集中起来会影响网络通信量。

在分布式结构中，由于目标代理和检测引擎都在目标机上，可以产生实时告警和进行实时响应。缺点是数据分析占用了目标机的资源，降低了目标机的性能；命令控制台只有各个检测引擎产生的告警信息，没有用于支持起诉的原始数据，无法进行行为的统计分析，降低了数据的辨析能力，系统离线时不能分析数据。

15.3.1.3 基于主机的入侵检测系统的优缺点

基于主机的入侵检测系统可以发现以下的入侵行为：如特权滥用、关键数据的访问及修改、安全配置的变化等。它具有基于网络的入侵检测系统无可比拟的优点，这些优点包括：

对网络流量不敏感、适用于加密和交换的环境、能更准确地确定攻击是否成功、监控粒度更细、配置灵活、不需要额外的硬件等。

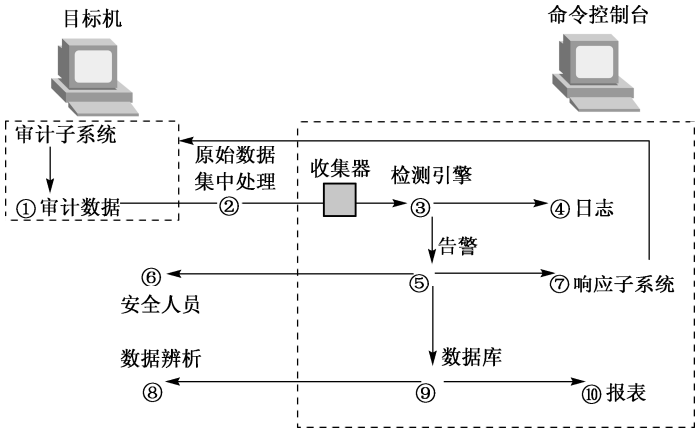


图 15.2　集中式基于主机的入侵检测系统结构

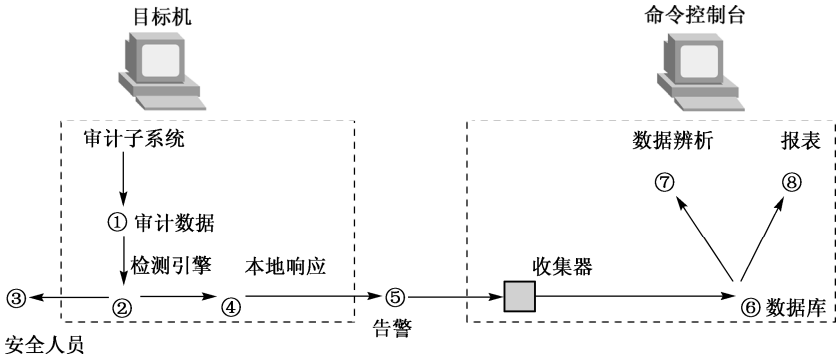


图 15.3　分布式基于主机的入侵检测系统结构

此外，基于主机的入侵检测系统面临的问题是：它安装在我们需要保护的设备上，这不可避免会降低应用系统的效率。全面部署和维护主机入侵检测系统的代价较大，很难将所有主机使用入侵检测系统保护，只能选择部分主机保护，那些未安装主机入侵检测系统的机器将成为保护的盲点，入侵者可利用这些机器达到攻击目标。主机入侵检测系统除了监测自身的主机以外，根本不监测网络上的情况。此外，主机入侵检测系统还可能收集到一些虚假的数据，UNIX 系统的二进制内核日志及 Windows NT/XP 安全事件日志是两个不错的审计源，但 UNIX 系统的 Syslog 及 Windows NT/XP 的应用程序事件日志就不是一个很好的审计源。

15.3.2　基于网络的入侵检测系统

15.3.2.1　基于网络的入侵检测系统的数据源

基于网络的入侵检测系统通过在共享网段上对通信数据进行侦听采集数据，分析可疑现象。在一个共享式网络中，我们可以监听所有的流量。能够监听网络流量，是网络管理人员了解网络工作状况，开发人员开发网络应用程序的需要。但是与此同时，黑客们也可以利用这种技术刺探网络情报。以太网工作基于一种叫做**载波侦听/冲突检测**(CSMA/CD)的技术。载波侦听是指网络中的每个站点都具有同等权利，在传输自己的数据时，首先监听信道是否

空闲，如果空闲，就传输自己的数据；如果信道被占用，就等待信道空闲。而冲突检测则是为了防止发生两个站点同时监测到网络没有被使用时而产生冲突的一种规则。由于使用了CSMA/CD技术，所有与网络连接的工作站都可以看到网络上传递的数据。网卡收、发数据包有两种工作模式：一种是混杂模式，在这种模式下，不管数据帧中的目的地址是否与自己的地址匹配，都接收下来；另一种是非混杂模式，在这种模式下，就只接收目的地址相匹配的数据帧，以及广播数据包和组播数据包。为了监视网络，网卡必须设置为混杂模式，只有这样，才能够把本网段上的所有数据包都收上来，才能够分析网络上正在发生的事情。

很多局域网，都是用 HUB 连在一起的，这时，通过网络的所有数据包发往每一个主机，也就是说，每一个主机都有机会看到网上的数据包，这是共享式网络。而交换式网络是指用交换机连接的局域网。在交换机中，有一个“MAC 地址——端口”对照表，数据进入交换机后，直接根据对照表，发送给了相应的主机，其他主机是没有可能收到发给别人的数据包的。但是，交换机往往带有一个监视端口，如果把监视主机连在这个端口上，那么所有流经网段的数据包都会从这个端口经过。

15.3.2.2 基于网络的入侵检测系统的系统结构

基于网络的入侵检测系统由遍及网络的传感器(Sensor)组成，传感器通常是独立的检测引擎，能获得网络分组、找寻误用模式，然后告警。典型的结构类型有两种：传统的基于传感器的结构和分布式网络节点结构(Network node)。

传统的基于传感器的结构又被称为混杂模式网络入侵检测系统，或网络分接器(Network tap)。传感器(通常是将以太网卡设置为混杂模式)用于“嗅探”网络上的数据分组，并将分组送往检测引擎。检测引擎安装在传感器计算机本身，分布在关键任务网段上，每个网段一个。中央控制台用于将来自多个传感器的告警相互关联起来。

分布式网络节点结构是为解决高速网络上的丢包问题，1999年6月，出现的一种新的结构，它将传感器分布到网络上的每台计算机上。每个传感器只检查流经它所驻留的计算机的网络分组，然后传感器相互通信，主控制台将所有的告警聚集、关联起来。

15.3.2.3 检测器的位置

基于网络的入侵检测系统需要有检测器才能工作。如果检测器放的位置不正确，入侵检测系统也无法工作在最佳状态。一般来说，检测器放在防火墙附近比较好。

1．放在防火墙之外

通常放在DMZ区，虽然无法检测到某些攻击，但可以看到自己的站点和防火墙暴露在多少种攻击之下。

2．检测器在防火墙内

检测器放在防火墙内的理由有：如果攻击者能够发现检测器，就可能采取措施躲避检测，从而减小攻击者的行动被审计的机会；防火墙内的系统会比外面的系统脆弱性少一些，如果检测器在防火墙内就会少一些干扰，从而有可能减少误报警；如果本应该被防火墙封锁的攻击渗透进来，检测器在防火墙内检测到或就能发现防火墙的设置失误；设置良好的防火墙能够阻止大部分的简单攻击，使检测器不用将大部分的注意力分散在这类攻击上。

3．防火墙内外都有检测器

如果机构有足够的经费，可以在防火墙内外都放置检测器，肯定可以获得比一个检测器

更多的好处，这时，无须猜测是否有攻击渗透过防火墙；可以更好地辨别攻击是否来自内部；也为怎样和何时触发警告提供了更多的灵活性。

4．检测器的其他位置

检测器最通常的位置在防火墙外，但这当然不是唯一对机构有利的摆放位置。许多入侵检测系统都可以在不同位置支持机构，这些检测器可能在：

- 高价值的地方，比如研究或会计网络。
- 有大量不稳定雇员(例如顾问或临时职员)的地方。
- 已被外部人员当做目标的子网或是已有迹象显示有入侵和其他非法活动的子网。

15.3.2.4　基于网络的入侵检测系统的优缺点

基于网络的入侵检测系统能检测非授权登录(Login)、进行其他攻击的起始点、口令下载、带宽窃取、拒绝服务、分布式拒绝服务等攻击。它具有监测速度快、隐蔽性好、视野更宽、较少的探测器、攻击者不易转移证据、操作系统无关性、可以配置在专门的机器上，不会占用被保护设备上的任何资源等优点。

基于网络的入侵检测系统面临的技术问题有：如果消息被分为多个分组，就能躲避检测；当网络速度增加时，分组会丢失；嗅探器检测程序，可以用来确定计算机上是否安装了网络分接器；在某些采用交换技术的网络环境中，交换机制使得网络报文不能在子网内任意广播，只能在设定的虚拟局域网(VLAN)内广播，这就使得进行网络监听的主机只能提取到本虚拟局域网内的数据，监视范围大为减小，监视的能力也受到削弱；有效载荷加密后，就不能检测出任何模式。

15.4　异常检测和误用检测

15.4.1　异常检测

如果系统错误地将异常活动定义为入侵，则称为**误报**(False positive)；如果系统未能检测出真正的入侵行为则称为**漏报**(False negative)。

什么叫异常呢？当然是不同于正常行为的行为。基于异常的检测技术就是基于这样的思想，即在正常情况下，每个人的行为都是有一定规律的，但黑客入侵系统时，其行为与正常情况是有差异的。如果我们可以把握正常情况下，用户行为的特征，就有可能发现异常行为，进而发现入侵行为。从理论上来讲，基于异常的入侵检测技术可以发现新的攻击手段发起的攻击行为，因为无论已知的，还是未知的，只要是攻击行为，就必然有别于正常使用系统的行为，我们就有可能发现它。

用户轮廓(Profile)通常定义为各种行为参数及其阈值的集合，用于描述正常行为范围。异常检测系统的效率取决于用户轮廓的完备性和监控的频率。这里需要考虑如下问题：首先，选择哪些数据来表现用户的行为？数据应能充分反映用户行为特征的全貌，同时应使需要的数据量最小，数据提取难度不应太大。其次，通过以上数据如何有效地表示用户的行为，主要在于对正常行为的学习和异常行为的检测方法的不同，同时还要考虑学习过程的时间长短、用户行为的时效性等问题。这些问题的解决需要的不仅仅是计算机知识，而是多学科知识的交叉，目前在理论和实践上，人们采用了多种技术来实现基于异常的入侵检测。

15.4.1.1 统计学的方法

统计分析方法首先给系统对象(如用户、文件、目录和设备等)创建一个统计描述，统计正常使用时的一些测量属性(如访问次数、操作失败次数和延时等)。测量属性的平均值将被用来与网络、系统的行为进行比较，任何观察值在正常值范围之外时，就认为有入侵发生。

这里通过SRI International 的 NIDES(Next-generation Intrusion Detection Expert System)来说明。NIDES通过将一个使用者的历史行为或长期行为与他的短期行为来比较检测入侵行为。它关心这样的两个方面：短期行为中没有出现的长期行为(没有按习惯方式进行操作)；非典型长期行为模式的短期行为(有异常行为发生)。短期行为与长期行为必然是有差异的。短期行为主要指某个特定行为，长期行为通常包括多个行为。NIDES 处理方式是这样的，它记录二者之间的差异，同时设定一个门限。当差异到大于门限，就会产生警告。

Dennying 提出可用于入侵检测 5 种统计模型，如下所示。

(1) **试验模型(Operational Model)**：该模型基于这样的假设，若已观测到变量x出现的次数超过某个预定的值，则就可能出现异常的情况。预定的值可以根据经验或者一段时间内的统计平均得到，如短时间内口令登录失败的次数超过某个给定值。

(2) **平均值和标准差模型(Mean and Standard Deviation Model)**：这个模型根据已观测到的随机变量x的样值$x_i(i=1, 2, \cdots, n)$，计算出这些样值的平均值 mean 和标准方差std，若新的样值x_{n+1}不在置信区间内时，则表明有可能是异常。该模型的优点在于不需要为了设定限制值而掌握正常活动的知识。相反，这个模型从观测中学习获取知识，置信区间的变动就反映出知识的增长过程。此模型可加上权重的计算，如最近取样的值的权重大些，就会更准确反映出系统的状态。

(3) **多变量模型(Multivariate Model)**：根据多个随机变量的综合结果来识别入侵行为，而不仅仅是单个变量。例如利用一个程序的CPU 使用时间、I/O、用户注册频度、通信会话时间等多个变量来检测入侵行为。

(4) **马尔可夫过程模型(Markov Process Model)**：该模型将离散的事件(审计记录)看做一个状态变量，然后用状态转移矩阵刻画状态的变化。若观察到一个新事件，而根据先前的状态和状态转移矩阵判断新事件的出现概率太低，则表明出现异常情况。

(5) **时序模型(Time Series Model)**：该模型根据事件$x_1, x_2, \cdots, x_n$之间的相隔时间和它们的值来判断入侵，若在某个时间内x出现的概率太低，则该事件可能是入侵。

15.4.1.2 基于神经网络的异常检测

之所以选择神经网络，是因为神经网络具有学习能力。这种能力可以使系统紧密跟踪用户行为并且根据其一段时间(可以根据情况进行调整)的变化进行调整。利用神经网络检测入侵包括两个阶段。首先是学习阶段，这个阶段使用代表用户行为的历史数据进行训练，完成神经网络的构建和组装；接着便进入入侵分析阶段，网络接收输入的事件数据，与参考的历史行为比较，判断出两者的相似度或偏离度。

15.4.1.3 基于数据挖掘的异常检测

数据挖掘也是这几年大家非常看重的一种技术，信息社会的信息膨胀使数据挖掘成为人们从信息海洋中提取有价值信息的好帮手。系统中通常记录了大量的日志信息，采用数据挖掘技术，我们有可能从这样大量的数据中提取出有价值信息，或是靠人工很难发现的“行为

模式”，从而实现异常检测。

其他常见的异常检测方法包括：基于特征选择异常检测、基于贝叶斯推理异常检测、基于贝叶斯网络异常检测、基于贝叶斯聚类异常检测、基于机器学习异常检测和基于模式预测异常检测等。

15.4.2　误用检测

误用检测的前提是所有的入侵行为都有可被检测到的特征。基于误用的入侵检测需要从各种网络攻击中提取出一定的攻击特征(模式)，作为监视网络攻击的依据。只要模式概括得合理，这种方式可以检测到许多攻击。对于新的攻击，如果没有增加模式，当然不能发现。这就好像防毒软件一样，只能查出杀死已知病毒，对于新出现的病毒，除非很快分析并增加病毒库，否则就无法检测。攻击特征库只有处于不断的动态更新过程之中，系统才能有效地运行。基于误用的入侵检测系统涉及的重要技术问题有：如何全面地描述攻击的特征，并用一种相对统一的表示方式加以表示；尽量排除干扰因素，减小误报率；发现攻击后，如何采取有效手段避免损失。如果入侵特征与正常的用户行为能匹配，则系统会发生误报；如果没有特征能与某种新的攻击行为匹配，则系统会发生漏报。采用特征匹配，误用检测能明显降低错报率，但漏报率随之增高。攻击特征的细微变化，会使得误用检测无能为力。误用检测中常用的方法有：模式匹配、专家系统、基于键盘监控的误用入侵检测等。

15.4.2.1　模式匹配

模式匹配是最为通用的误用检测技术，它拥有一个攻击特征数据库，当监测的用户或系统行为与库中的某个模式(特征)相匹配时，系统就认为这种行为是入侵。这种方法的特点是原理简单、扩展性好、检测效率高、可以实时监测，但只适用于检测比较简单的攻击。由于其实现、配置和维护都非常方便，因此得到了广泛的应用。Snort 系统就采用了这种检测手段。

例如，Land 攻击的特征是 IP 包的源地址和目标地址相同。只要检查 IP 报文中的源地址和目标地址，如果两者相同，就可以认定该数据包有问题。

SYN 洪泛的特征是检测到同一源地址对目标主机同一端口的大量连接企图(TCP SYN)。什么叫“大量”呢？阈值设置得不合理，就存在误报的可能。

15.4.2.2　专家系统

专家系统的应用方式是：首先使用类似于if-then的规则格式输入已有的知识(攻击模式)，然后输入检测数据(审计事件记录)，系统根据知识库中的内容对检测数据进行评估，判断是否存在入侵行为模式。专家系统的优点在于把系统的推理控制过程和问题的最终解答相分离，即用户不需要理解或干预专家系统内部的推理过程，而只需把专家系统看成是一个黑盒子。CLIPS(C Language Integrated Production System)是使用最广泛的专家系统工具，因为其高效、便宜，它的原始作者是 Gary Riley。

15.5　入侵检测的响应

一次完整的入侵检测系统包括数据收集、数据分析和结果处理三个部分，结果处理，也就是响应是其必要的组成。在设计响应机制时，必须综合考虑以下几个方面的因素：

- **系统用户**：入侵检测系统用户可以分为网络安全专家或管理员、系统管理员、安全调

查员。这三类人员对系统的使用目的、方式和熟悉程度不同，必须区别对待。

- **操作运行环境**：入侵检测系统提供的信息形式依赖其运行环境。
- **系统目标**：为用户提供关键数据和业务的系统，需要部分地提供主动响应机制。
- **规则需求**：在某些军事环境里，允许采取主动防御甚至攻击技术来对付入侵行为。

响应措施分为被动响应和主动响应两种类型。被动响应的方式一般是：警报显示和采用移动电话、电子邮件进行远程通报。主动响应的措施主要有三类，包括针对入侵者的措施，对系统的修正和收集攻击者更详细信息的措施。

15.5.1 针对入侵者的措施

自动响应是一种针对入侵者的措施，具体的形式有压制调速(Throttling)、TCP RESET、防火墙或网关的联动等。自动响应是最便宜、最容易的响应方式，是最具侵略性的形式，追踪入侵者实施攻击的发起地，并采取措施以禁用入侵者的机器或网络连接。但这个方式存在一些反击的风险。如果攻击者通过网络跳转、IP 地址欺骗的方式实施攻击，断开连接就会伤及无辜。

(1) **压制调速**：对于端口扫描、SYN洪泛等攻击形式，这是一种巧妙的响应方式。其思想是在检测到端口扫描或SYN 洪泛等行为时就开始增加延时，如果该行为继续，就继续增加延时。

(2) **撤销连接**：当说到连接时，主要是指TCP连接。当攻击者对一个激活的端口进行连接，他向该端口发送一个或数个包，包含有攻击字符串。入侵检测系统检测到攻击字符串后，命令防火墙撤销连接。

(3) **回避(Shun)**：通常是针对UDP协议。随着攻击的进行，会有一个新的进程以超级用户级别运行，撤销连接不起作用(因为它已经做好开始另一个连接的准备)，回避可以发挥一定作用。在正确实施回避技术时，系统不传递任何来自攻击者或发向攻击者的包。

(4) **隔离**：自动响应的最后一着。其思想是：如果在某一时间段发生了足够多次的攻击，入侵检测系统就发出命令，将路由器的电源断掉。

(5) **TCP Reset**：TCP Reset可能会断开其他人的TCP连接。这种响应的思想是如果发现一个TCP 连接被建立，而它连接的是你要保护的某种东西，就伪造一个Reset 并将其发送给发起连接的主机，使连接断开。

15.5.2 对系统的修正

针对受保护系统，可以采取的响应措施有：弥补引起攻击的缺陷，隔离导致问题的部分。针对入侵检测系统，可以采取的响应措施有：改变监控范围或收集数据的粒度，改变分析引擎的操作方式和参数，添加、修改检测规则。

15.5.3 收集攻击者的信息

检测到入侵后，把攻击者引导到经过特殊装备的诱骗服务器上，这些服务器可以模拟关键系统的文件系统和其他系统特征，引诱攻击者进入，记录下攻击者的行为，从而获得关于攻击者的详细信息。典型的诱骗服务器有蜜罐(Honey Pot)和沙箱(Manhunt, Mantrap)。

15.6　入侵检测的标准化工作

标准化是一个市场成熟的关键之一。随着网络规模的扩大，网络入侵的方式、类型、特征各不相同，入侵的活动变得复杂而又难以捉摸。某些入侵的活动靠单一 IDS 不能检测出来，如分布式攻击。网络管理员常因缺少证据而无法追踪入侵者，入侵者仍然可以进行非法活动。不同的 IDS 之间没有协作，结果造成缺少某种入侵模式而导致 IDS 不能发现新的入侵活动。目前网络的安全也要求 IDS 能够与访问控制、应急、入侵追踪等系统交换信息，相互协作，形成一个整体有效的安全保障系统。与 IDS 相关的标准有：通用入侵检测框架 CIDF（The Common Intrusion Detection Framework），IETF 入侵检测工作组（IDWG）的入侵检测信息交换格式 IDMEF（Intrusion Detection Message Exchange Format）以及漏洞和风险的标准 CVE（Comm- on Vulnerabilities and Exposures）。

15.6.1　通用入侵检测框架 CIDF

CIDF 早期由美国国防部高级研究计划局赞助研究，现在由 CIDF 工作组负责，这是一个开放组织。实际上 CIDF 已经成为一个开放的共享的资源。CIDF 是一套规范，它定义了 IDS 表达检测信息的标准语言以及 IDS 组件之间的通信协议。符合 CIDF 规范的 IDS 可以共享检测信息，相互通信，协同工作，还可以与其他系统配合实施统一的配置响应和恢复策略。CIDF 的主要作用在于集成各种 IDS 使之协同工作，实现各IDS之间的组件重用，所以 CIDF 也是构建分布式 IDS 的基础。

CIDF 的规格文档由四部分组成，分别为：

- 体系结构（The Common Intrusion Detection Framework Architecture）。
- 规范语言（Common Intrusion Specification Language）。
- 内部通信（Communication in the Common Intrusion Detection Framework）。
- 程序接口（Common Intrusion Detection Framework API）。

其中体系结构阐述了一个标准的IDS通用模型；规范语言定义了一个用来描述各种检测信息的标准语言；内部通信定义了IDS组件之间进行通信的标准协议；程序接口提供了一整套标准的应用程序接口（API 函数）。

在 CIDF 的体系结构文档中，它将一个 IDS 分为以下四个组件：事件产生器（Event generator）、事件分析器（Event analyzer）、事件数据库（Event databases）和响应单元（Response unit）。

CIDF 将IDS 需要分析的数据统称为事件（Event），它可以是基于网络的IDS 从网络中提取的数据包，也可以是基于主机的 IDS 从系统日志等其他途径得到的数据信息。

CIDF 组件之间是以通用入侵检测对象（Generalized Intrusion Detection Object，GIDO）的形式交换数据的，一个GIDO 可以表示在一些特定时刻发生的一些特定事件，也可以表示从一系列事件中得出的一些结论，还可以表示执行某个行动的指令。

CIDF 中的事件产生器负责从整个计算环境中获取事件，但它并不处理这些事件，而是将事件转化为 GIDO 标准格式提交给其他组件使用，显然事件产生器是所有 IDS 所需要的，同时也是可以重用的。CIDF 中的事件分析器接收GIDO，分析它们，然后以一个新的GIDO 形式返回分析结果。CIDF 中的事件数据库负责GIDO 的存储，它可以是复杂的数据库，也可以是

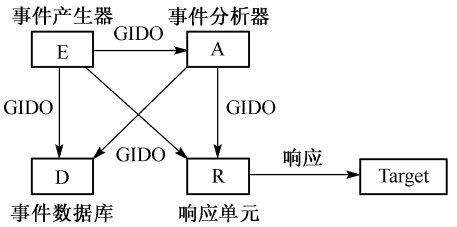


图 15.4 CIDF 构件间的通信路径

简单的文本文件。CIDF中的响应单元根据GIDO 做出反应，它可以是终止进程、切断连接、改变文件属性，也可以只是简单的报警。CIDF构件间的通信路径如图15.4 所示。

15.6.2 入侵检测交换格式

IDWG 从 CIDF 发展而来，IDWG 的最终目的是创建一种包括数据格式及交换协议的IETF标准，以便异种入侵检测系统能互相通信。IDWG 从 CIDF 草案开始，新的名称为入侵检测信息交换格式IDMEF。它与 CIDF 在三个方面不同：IETF RFC 过程被设计为产生商业标准，所有厂商的支持、关注都集中于 IETF，IDWG 的文档对业界具有可读性。

15.7 入侵防御系统

入侵检测系统的目的是提供监视、审计、取证和对网络活动的报告。入侵防御系统(Intrusion Prevention System，IPS)的目的是为资产、资源、数据和网络提供保护。IPS是一种能够检测已知和未知攻击并成功阻止攻击的软硬件系统。IDS 和 IPS 的差别在于确定性。IDS使用非确定性的方法从现在和历史的通信流中查找威胁或者潜在的威胁，包括执行通信流、通信模式和异常活动的统计分析。IPS必须是确定性的，它所执行的所有丢弃通信包的行为必须是正确的。IPS 被认为是一直在网络上处于工作状态，执行访问控制的决定。

防火墙提供了确定性的网络访问控制，是IPS的基本能力。IDS和防火墙的组合通常被看做是一种入侵防御系统。但这还不是真正意义上的入侵防御系统。从功能上来看，IDS是一种并联在网络上的设备，它只能被动地检测网络遭到了何种攻击，它的阻断攻击能力非常有限，一般只能通过发送TCP Reset包或联动防火墙来阻止攻击。包过滤防火墙工作在网络层和传输层，代理防火墙工作在网络层、传输层和应用层，状态监测防火墙利用了网络层以上的信息，IPS相对防火墙来说，功能比较单一，但 IPS 串联在网络上，利用了OSI参考模型的所有 7 层信息，对攻击进行过滤，提供了一种主动的、积极的入侵防范。

思考和练习题

(1) 什么是入侵检测？它和基于加密和访问控制的安全技术有什么不同？
(2) 入侵检测系统由哪几个功能模块组成？简述各模块的作用。
(3) 按收集的信息来源，入侵检测系统分为哪两类？各有什么特点？
(4) 入侵检测的分析方法有几种？各有什么特点？
(5) 入侵检测系统与入侵防御系统有什么联系和区别？

实践/实验题

下载一个Sniffer软件(例如 Snort 或者 Sniffer Pro)并安装和配置，练习使用 Sniffer捕获数据包。

第 16 章　信息安全评估标准

不同的应用环境、应用领域以及处理信息的敏感性的不同，对信息安全的需求也不一样。生产者、使用者、评估者从不同的角度出发，得出的结论也不尽相同。因此，必须有一个对信息产品和系统安全性进行定性和定量评估的规范和统一标准，使各种独立的安全评估结果具有可比性。这样的评估标准可作为系统安全评估的依据，也可作为生产厂家衡量其安全产品是否符合安全要求的依据，同时也是用户选购安全产品和系统的参考。

16.1　评估标准的发展历程

信息安全评估标准的制定是信息安全的制高点，世界各国都十分重视系统安全标准的研究，美国无疑又是这一方面的先驱。1967 年美国国防科学委员会提出计算机安全保护问题后，1970年美国国防部(DoD)在国家安全局(NSA)建立一个计算机安全评估中心(NCSC)，开始从事计算机安全评估的研究。1983 年年底，美国国防部发布了《可信计算机系统评估标准》(Trusted Computer System Evaluation Criteria，TCSEC)，该标准于1985 年修订后重新公布。从美国发布 TCSEC 开始，国际上，安全评估标准的发展走过了三个阶段，如图16.1所示。

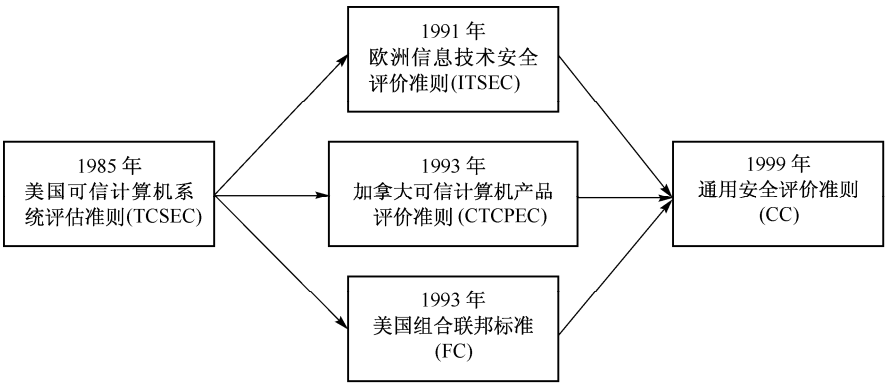


图 16.1　安全评估标准的发展历程

TCSEC 的第一版发布于 1983 年，1985 年最终修订。由于使用了橘色书皮，通常人们称其为“橘皮书”，后来在 NCSC 的主持下制定了一系列相关准则，称为彩虹系列，其中，1987 年，NCSC 为 TCSEC 提出的可信网络解释(TNI 1987)通常被称为“红皮书”。1991年，为TCSEC提出的可信数据库解释(TDI 1991)通常称为“紫皮书”。TCSEC 将计算机安全从低到高顺序分为四类八级：最低保护等级(D)、自主保护等级(C1, C2)、强制保护等级(B1, B2, B3)和验证保护等级(A1，超 A1)，为信息安全产品的测评提供准则和方法，指导信息安全产品的制造和应用。TCSEC 是针对孤立计算机系统提出的，特别是小型机和主机系统，假设有一定的物理屏障，该标准适合军队和政府，不适合企业，是一个静态模型。TNI 是把 TCSEC 的思想用到网络上，缺乏成功实践的支持。

在借鉴TCSEC成功经验的基础上，20 世纪 90 年代初，西欧四国(英、法、荷、德)联合提出了信息技术安全评价准则(ITSEC)。ITSEC作为多国安全评估标准的综合产物，适用于军队、政府和商业部门。它以超越 TCSEC 为目的，将安全概念分为功能与功能评估两部分。ITSEC定义了7个安全级别：从不能充分满足保证的 E0 级到形式化验证的 E6 级。

1989 年，加拿大发布了“加拿大可信计算机产品评价准则”(CTCPEC)1.0 版，1993 年，公布了3.0版。将安全分为功能性要求和保证性要求两部分。同年，美国对可信计算机系统评估准则(TCSEC)做了补充和修改，美国国家标准局和国家安全局合作制定了“组合的联邦标准”(简称 FC)，在此标准中引入了“保护轮廓”(PP)这一重要概念，每个轮廓都包括功能部分、开发保证部分和评测部分。主要供美国政府、民间和商业使用，但其有很多缺陷，只是一个过渡准则。这一阶段的安全性评估不仅全面包含了信息网络系统的安全性，而且内容也有很多的扩充，不单是功能要求，还包括了开发保证要求和评估要求。但是，各国标准分散，度量标准不尽相同。

由于信息产业发展的全球化，需要评估结果之间的相互认可。为了能集中世界各国评估准则的优点，集成单一的、能被广泛接受的信息技术评估准则，1990 年国际标准化组织 ISO 开始着手这一工作，但进展缓慢。1993 年 6 月，六国七方，他们是美国的国家标准及技术研究所(NIST)和国家安全局(NSA)，欧州的荷、法、德、英，北美的加拿大，开始进行制定通用安全评价准则(Common Criteria for IT security Evaluation，CC)的工作，并将他们的成果作为对国际的贡献提交给 ISO。1996 年 1 月 CC 出版了第一版，1998 年颁布了第二版，并于 1999 年被国际标准化组织采纳为国际标准(ISO/IEC 15408：1999)。

我国是国际标准化组织的成员国，我国的信息安全标准化工作在各方面的努力下，正在积极开展之中。从 20 世纪 80 年代中期开始，自主制定和视同采用了一批相应的信息安全标准。到2000年年底，已颁布的信息技术安全标准有22项，国家军用安全标准6项，涉及信息技术设备的安全、信息处理系统开放系统互联安全体系结构、数据加密、数字签名、实体鉴别、抗抵赖和防火墙安全技术等。在充分借鉴国际标准的前提下，我国制定了自己的安全评估标准。1999 年9月13日发布了《计算机信息系统安全保护等级划分准则》(GB 17859—1999)，并于2001 年 1 月 1 日起实施。此后又发布了一系列的相关配套标准：《信息系统安全等级保护基本要求》、《信息安全技术 信息系统通用安全技术要求》(GB/T 20271—2006)、《信息安全技术网络基础安全技术要求》(GB/T 20270—2006)、《信息安全技术 操作系统安全技术要求》(GB/T 20272—2006)、《信息安全技术 数据库管理系统安全技术要求》(GB/T 20273—2006)、《信息安全技术 服务器技术要求》、《信息安全技术 终端计算机系统安全等级技术要求》(GA/T 671—2006)等。2001 年 3 月，国家质量技术监督局发布了推荐性标准《信息技术 安全技术 信息技术安全性评估准则》(GB/T 18336—2001)，该标准等同采用国际标准 ISO/IEC 15408。

16.2 TCSEC

TCSEC 的提出，对计算机和网络系统具有以下 3 个作用：

(1) 向制造商提供一个检查和评估安全产品的依据，为用户提供一个选择参考。
(2) 为国防机关各部门提供一个度量标准，以评估计算机系统或其他敏感信息的可信程度。
(3) 在分析、研究规范时，为安全需求制定和安全系统设计提供了一个基础。

TCSEC采用等级评估的方法，将计算机安全分为 A, B, C, D 四等八级，各级具体安全要求如表 16.1 所示。下面，对每个安全等级的内容和要求做简要说明。

表 16.1　TCSEC 各等级安全要求

类　别	评　估	D	C1	C2	B1	B2	B3	A1	超 A1
安全策略	自主访问控制	*	*	—	—	*	—	—	
	客体重用			*	—	—	—	—	—
	标记				*	*	—	—	—
	标记的完整性				*	—	—	—	—
	标记的信息输出				*	—	—	—	—
	多级设备输出				*	—	—	—	—
	单级设备输出				*	—	—	—	—
	标记的硬拷贝输出				*	—	—	—	—
	强制访问控制				*	*	—	—	—
	主体安全标记					*	—	—	—
	设备标记					*	—	—	—
责任	标识与鉴别		*	*	*	—	—	—	—
	安全审计			*	*	*	*	—	—
	可信路径					*	*	—	—
保证	系统体系结构		*	*	*	—	—	—	—
	系统的完整性		*	—	—	—	—	—	—
	安全性测试		*	*	*	*	*	*	*
	设计规范与验证				*	*	*	*	*
	隐蔽信道分析					*	*	*	—
	可信设施管理					*	*	—	—
	配置管理					*	—	*	—
	可信恢复						*	—	—
	可信分发							*	—
文件	安全特性用户指南		*	—	—	—	—	—	—
	可信设施手册		*	*	*	*	*	—	—
	测试文档		*	—	—	*	—	—	—
	设计文件	*	—	*	*	*	*	*	

注：表中空白表示无要求；*表示增加或有修改；—表示与前一等级相同。

16.2.1　无保护级

D 等是最低保护等级，即无保护级。它是为那些经过评估，但不满足较高评估等级要求的系统设计的，只具有一个级别。该等是指不符合要求的那些系统。因此，这种系统不能在多用户环境下处理敏感信息。

16.2.2　自主保护级

C 等为自主保护级，它具有一定的保护能力，采用的措施则是自主访问控制和审计跟踪。它一般只适用于具有一定等级要求的多用户环境。这一等级分为 C1 和 C2 两个级别。

TCSEC使用了**可信计算基**(Trusted Computing Base，TCB)这一概念，计算机系统内保护装置的总体，包括硬件、固件、软件和负责执行安全策略的组合体。它建立了一个基本的保护环境并提供一个可信计算系统所要求的附加用户服务。

(1) **自主安全保护级(C1 级)**：C1 级TCB 通过隔离用户与数据，使用户具备自主安全保护的能力。在这一级，TCB应在命名用户和命名客体之间定义和进行访问控制。它允许客体拥有者自主定义和控制客体的使用权限。该级需要在进行任何活动之前，TCB去确认用户身份，并保护确认数据，以免未经授权对确认数据的访问和修改。同时TCB定期检查系统运行的正确性。系统的完整性要求硬件和软件能提供保证TCB连续有效操作的特性。目前生产的大多数计算机系统都能达到这一等级。

(2) **可控访问保护级(C2级)**：在C2 级，计算机系统比C1 级具有更细粒度的自主访问控制，通过注册过程控制、审计安全相关事件以及资源隔离，使单个用户为其行为负责。C2 级与C1 级的差别在于访问控制和审计跟踪。C1 级计算机没有审计跟踪功能。C2 级计算机必须识别单个用户。对每个审计事件，审计记录应包括：用户名、事件发生时间、事件类型、事件的成功或失败等。同时还提供了客体重用功能，即对一个未使用的存储客体，TCB 应该能够保证该客体不包含未授权主体的数据。

16.2.3 强制保护级

B 等为强制保护级。这一等级比 C 等的安全功能有很大增强。它要求对客体实施强制访问控制，并要求客体必须带有敏感标记，可信计算基利用它去施加强制访问控制。这一部分可分为 B1, B2 和B3 三级。

(1) **标记安全保护级(B1 级)**：该级具有C2 级的全部功能，并增加了标记、强制访问控制、责任、审计和保证功能。

(2) **结构化保护级(B2 级)**：该级着重强调实际中的评价手段。为此，B2 级增加了以下功能：

- **安全策略**：加强了强制访问功能。将强制访问控制对象，从主体和客体扩展到 I/O 设备等所有资源。并要求每种系统资源必须与安全标记相联系。
- **责任**：在责任方面，提高了连续保护和防渗透能力。保证了与用户之间开始注册和确认时的通路是可信的，提高了系统连续保护和防渗入的能力。使要求审计功能得到加强，能审计使用隐蔽存储信道的标记事件。隐蔽通道是指用违反系统安全策略的方法传输信息的通道，如一个进程直接或间接地对一个存储单元写，而另一个进程直接或间接地对该存储单元读，就形成一个隐蔽信道。
- **结构**：在结构上应支持操作人员和管理人员的分离，并执行最小特权原则。即每个主体进行授权任务时，应被授予完成任务所必需的最小存取权；还应划分与保护有关和无关部分，并把它的执行维持在一个固定的区域，以防止外部干预和篡改，其设计和实现要利用检测和评估，应支持操作人员和管理人员的分离。

(3) **安全区域保护级(B3 级)**：B3 级除满足 B2 级的所有要求外，它监督所有主体对客体的访问，防篡改，并提供分析和测试；并将审计机理扩展到能报知与安全有关的事件。系统要有恢复能力，为此，B3 级增加了一个安全策略。

- **安全策略**：采用访问控制表进行控制，允许用户指定和控制对客体的共享，也可以指定命名用户对客体访问方式。
- **责任**：它能监视安全审计事件的发生，当超过一定阈值时，能立即报知安全管理人员，进行处理。
- **保证**：只能完成与安全有关的管理功能，对其他完成非安全功能的操作要严加限制。

在系统出现故障和灾难性事件后，要提供过程和机理，以保证在不损害保护的条件下，使系统得到恢复。

16.2.4　验证保护级

A 等是验证保护级。它的显著特征是高度地保证正确实现 TCB。它使用形式化验证方法，以保证系统的自主访问和强制访问，能有效地使该系统存储和处理秘密信息或其他敏感信息。为证明TCB 满足设计、开发及实现等各个方面的安全要求，系统应提供丰富的文档信息。该等级分为 A1 和超 A1 两个级。

(1) **验证设计级(A1 级)**：A1 级系统在功能上和 B3 级系统是相同的，本级的主要特点是，要求用形式化设计规范和验证方法来对系统进行分析，确保 TCB 按设计需要实现。Honeywell 公司的 Scomp 系统被确定为 A1 级。

(2) **超 A1 级**：本级在A1 级基础上增加的许多安全措施超出了目前的技术发展，这里的讨论只是为了对今后的工作提供一些指导，随着更多、更好的分析技术的出现，本级系统的要求才会变得更加明确。

16.3　信息技术安全评估通用准则(CC)

16.3.1　CC 的范围

CC 重点考虑人为的信息威胁，无论其是有意的或无意的，不过，CC 也可用于非人为因素导致的威胁。

CC适用于硬件、固件和软件实现的信息技术安全措施，而某些内容因涉及特殊专业技术或仅是信息技术安全的外围技术，因此不在 CC 的范围内，例如：

(1) CC 不包括那些与信息技术安全措施没有直接关联的属于行政性管理安全措施的安全评估准则。这类管理安全措施在 TOE 的运行环境中被认为是 TOE 安全使用的前提条件。

(2) CC 不专门针对信息技术安全性的物理方面(诸如电磁辐射控制)的评估。

(3) CC 不涉及评估方法学，也不涉及评估机构使用 CC 的管理模式或法律框架。

(4) 评估结果用于产品和系统认可的过程不在 CC 的范围之内。

(5) CC 不包括密码算法固有质量评价准则。

16.3.2　CC 的组成

CC 由三个部分组成。

(1) **介绍和一般模型**：定义了安全评估的通用概念和原理，提出了评估的通用模型。建立了一些概念，这些概念可用于表达安全目的、选择和定义安全需求。

(2) **安全功能需求**：提出了一系列安全功能组件作为表示评估对象(TOE)功能要求的标准方法。该部分共列出了11个类、66 个子类和135个功能组件。11个功能类是：安全审计(FAU)、密码支持(FCS)、通信(FCO)、用户数据保护(FDP)、标识与鉴别(FIA)、安全管理(FMT)、隐私(FPR)、TOE 安全函数的保护(FPT)、资源利用(FUR)、TOE 访问(FTA)、可信路径/通道(FTP)。每一类都包含一些不同安全目标要求的族。例如，FCS类包含两个处理不同密码功能的族：密钥管理和密码运算。

(3) **安全保证需求(assurance requirement)**：提出了一系列安全保证组件，作为表示评估对象保证要求的标准方法。该部分包含有 8 个保证类，分别是：配置管理(ACM)、分发和操作(ADO)、开发(ADV)、指导性文档(AGD)、生命周期支持(ALC)、测试(ATE)、脆弱性评价(AVA)、保证维护(AMA)。每一类中都包含一些不同安全目标要求的族。例如，指导性文档类包含两个族：管理员指南和用户指南。CC 基于不断增加的保证范围提供了 7 个递增的评估保证等级，EAL1 到 EAL7。

- EAL1：功能测试。
- EAL2：结构测试。
- EAL3：系统测试和检查。
- EAL4：系统设计、测试和复查。
- EAL5：半形式化设计和测试。
- EAL6：半形式化验证的设计和测试。
- EAL7：形式化验证的设计和测试。

CC 的制定考虑了对前期标准的兼容，因而它们之间可建立如表 16.2 所示的粗略对应关系。

表 16.2 不同评估标准的评估级别的粗略对应关系

TCSEC	D	—	C1	C2	B1	B2	B3	A1
ITSEC	E0	—	E1	E2	E3	E4	E5	E6
CC	—	EAL1	EAL2	EAL3	EAL4	EAL5	EAL6	EAL7

16.4 GB 17859—1999

为了提高我国计算机信息系统安全保护水平，公安部提出并组织制定了强制性国家标准《计算机信息安全保护等级划分准则》(GB 17859—1999)，该标准是建立安全等级保护制度，实施安全等级管理的重要基础性标准。《计算机信息系统安全保护等级划分准则》规定了计算机信息系统安全保护能力的五个等级：即用户自主保护级、系统审计保护级、安全标记保护级、结构化保护级和访问验证保护级。详细阐述了自主访问控制、身份鉴别、数据完整性、审计、客体重用、强制访问控制、标记、可信路径、隐蔽通道分析及可信恢复等安全要素的内容，从安全策略(Policy)、责任(Accountability)及保证(Assurance)三个方面明确规定了各安全保护等级的要求。自主访问控制、客体重用、标记及强制访问控制属于策略范畴，身份鉴别、可信路径和审计属于责任范畴，数据完整性、隐蔽通道分析及可信恢复属于保证的范畴。等级和安全要素的对应关系如表 16.3 所示。但是在每一安全等级，相同名字的安全要素含义并不完全相同。

(1) **第一级**：用户自主保护级。本级的计算机信息系统可信计算基通过隔离用户与数据，使用户具备自主安全保护的能力。它具有多种形式的控制能力，对用户实施访问控制，即为用户提供可行的手段，保护用户和用户组信息，避免其他用户对数据的非法读、写与破坏。

(2) **第二级**：系统审计保护级。与用户自主保护级相比，本级的计算机信息系统可信计算基实施了粒度更细的自主访问控制，它通过登录规程、审计安全性相关事件和隔离资源，使用户对自己的行为负责。

表 16.3　安全保护等级与安全要素的对应关系

安 全 要 素	用户自主保护级	系统审计保护级	安全标记保护级	结构化保护级	访问验证保护级
自主访问控制	√	√	√	√	√
身份鉴别	√	√	√	√	√
数据完整性	√	√	√	√	√
审计		√	√	√	√
客体重用		√	√	√	√
标记			√	√	√
强制访问控制			√	√	√
隐蔽信道分析				√	√
可信路径				√	√
可信恢复					√

(3) **第三级**：安全标记保护级。本级的计算机信息系统可信计算基具有系统审计保护级的所有功能。此外，还提供有关安全策略模型、数据标记以及主体对客体强制访问控制的非形式化描述；具有准确地标记输出信息的能力；消除通过测试发现的任何错误。

(4) **第四级**：结构化保护级。本级的计算机信息系统可信计算基建立于一个明确定义的形式化安全策略模型之上，它要求将第三级系统中的自主和强制访问控制扩展到所有主体与客体。此外，还要考虑隐蔽通道。本级的计算机信息系统可信计算基必须结构化为关键保护元素和非关键保护元素。计算机信息系统可信计算基的接口也必须明确定义，使其设计与实现能经受更充分的测试和更完整的复审。加强了鉴别机制；支持系统管理员和操作员的职能；提供可信设施管理；增强了配置管理控制。系统具有相当的抗渗透能力。

(5) **第五级**：访问验证保护级。本级的计算机信息系统可信计算基满足访问监控器需求。访问监控器仲裁主体对客体的全部访问。访问监控器本身是抗篡改的；必须足够小，能够分析和测试。为了满足访问监控器需求，计算机信息系统可信计算基在其构造时，排除那些对实施安全策略来说并非必要的代码；在设计和实现时，从系统工程角度将其复杂性降低到最小程度。支持安全管理员职能；扩充审计机制，当发生与安全相关的事件时发出信号；提供系统恢复机制。系统具有很高的抗渗透能力。

16.5　GB/T 22239—2008

16.5.1　GB/T 22239—2008 简介

《信息系统安全等级保护基本要求》(GB/T 22239—2008)在 GB 17859—1999，GB/T 20269—2006，GB/T 20270—2006，GB/T 20271—2006等技术类标准的基础上，根据现有技术的发展水平，提出和规定了不同安全保护等级信息系统的最低保护要求，即基本安全要求，基本安全要求包括基本技术要求和基本管理要求，适用于指导不同安全保护等级信息系统的安全建设和监督管理。

《基本要求》的技术部分吸收和借鉴了GB 17859—1999 标准，采纳其中的身份鉴别、数据完整性、自主访问控制、强制访问控制、审计、客体重用(改为剩余信息保护)标记、可信

路径等 8 个安全机制的部分或全部内容，并将这些机制扩展到网络层、主机系统层、应用层和数据层。

《基本要求》的技术部分弱化了在信息系统中实现安全机制结构化设计及安全机制可信性方面的要求，例如没有提出信息系统的可信恢复，但在 4 级系统提出了灾难备份与恢复的要求，保证业务连续运行。《基本要求》没有对隐蔽通道分析的安全机制提出要求。

此外，《基本要求》的管理部分充分借鉴了ISO/IEC 17799—2005等国际上流行的信息安全管理方面的标准，尽量做到全方位的安全管理。

16.5.2 《基本要求》的框架结构

《基本要求》在整体框架结构上以三种分类为支撑点，自上而下分别为：类、控制点和项。其中，类表示《基本要求》在整体上大的分类，其中技术部分分为：物理安全、网络安全、主机安全、应用安全和数据安全及备份恢复 5 大类，管理部分分为：安全管理制度、安全管理机构、人员安全管理、系统建设管理和系统运维管理 5 大类，一共分为10个类别。控制点表示每个大类下的关键控制点，如物理安全大类中的“物理访问控制”作为一个控制点。而项则是控制点下的具体要求项，如“机房出入应安排专人负责，控制、鉴别和记录进入的人员”。

《基本要求》的具体框架结构如图16.2所示。

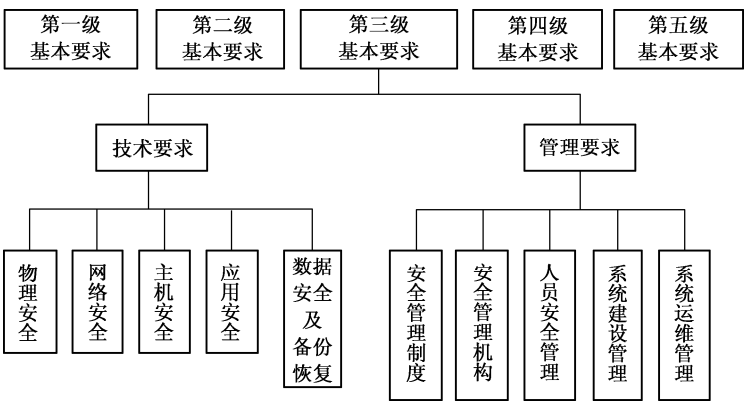


图 16.2 《基本要求》的框架结构

16.5.3 《基本要求》的技术要求

16.5.3.1 物理安全

物理安全保护的目的主要是使存放计算机、网络设备的机房以及信息系统的设备和存储数据的介质等免受物理环境、自然灾难以及人为操作失误和恶意操作等各种威胁所产生的攻击。物理安全是防护信息系统安全的最底层，缺乏物理安全，其他任何安全措施都是毫无意义的。

物理安全主要涉及的方面包括环境安全(防火、防水、防雷击等)设备和介质的防盗窃防破坏等方面。具体包括：物理位置的选择、物理访问控制、防盗窃和防破坏、防雷击、防火、防水和防潮、防静电、温湿度控制、电力供应和电磁防护 10 个控制点。

16.5.3.2　网络安全

网络安全为信息系统在网络环境的安全运行提供支持。一方面，确保网络设备的安全运行，提供有效的网络服务，另一方面，确保在网上传输数据的保密性、完整性和可用性等。由于网络环境是抵御外部攻击的第一道防线，因此必须进行各方面的防护。对网络安全的保护，主要关注两个方面：共享和安全。开放的网络环境便利了各种资源之间的流动、共享，但同时也打开了“罪恶”的大门。因此，必须在二者之间寻找恰当的平衡点，使得在尽可能安全的情况下实现最大程度的资源共享，这是我们实现网络安全的理想目标。

网络安全主要关注的方面包括：网络结构、网络边界以及网络设备自身安全等，具体的控制点包括：结构安全、访问控制、安全审计、边界完整性检查、入侵防范、恶意代码防范、网络设备防护 7 个控制点。

16.5.3.3　主机安全

主机系统安全是包括服务器、终端/工作站等在内的计算机设备在操作系统及数据库系统层面的安全。终端/工作站是带外设的台式机与笔记本计算机，服务器则包括应用程序、网络、Web、文件与通信等服务器。主机系统是构成信息系统的主要部分，其上承载着各种应用。因此，主机系统安全是保护信息系统安全的中坚力量。

主机系统安全涉及的控制点包括：身份鉴别、安全标记、访问控制、可信路径、安全审计、剩余信息保护、入侵防范、恶意代码防范和资源控制 9 个控制点。

16.5.3.4　应用安全

通过网络、主机系统的安全防护，最终应用安全成为信息系统整体防御的最后一道防线。在应用层面运行着信息系统的基于网络的应用以及特定业务应用。基于网络的应用是形成其他应用的基础，包括消息发送、Web 浏览等，可以说是基本的应用。业务应用采纳基本应用的功能以满足特定业务的要求，如电子商务、电子政务等。由于各种基本应用最终是为业务应用服务的，因此对应用系统的安全保护最终就是如何保护系统的各种业务应用程序安全运行。

应用安全主要涉及的安全控制点包括：身份鉴别、安全标记、访问控制、可信路径、安全审计、剩余信息保护、通信完整性、通信保密性、抗抵赖、软件容错、资源控制 11 个控制点。

16.5.3.5　数据安全及备份恢复

信息系统处理的各种数据(用户数据、系统数据、业务数据等)在维持系统正常运行上起着至关重要的作用。一旦数据遭到破坏(泄露、修改、毁坏)，都会在不同程度上造成影响，从而危害到系统的正常运行。由于信息系统的各个层面(网络、主机、应用等)都对各类数据进行传输、存储和处理等，因此，对数据的保护需要物理环境、网络、数据库和操作系统、应用程序等提供支持。各个“关口”把好了，数据本身再具有一些防御和修复手段，必然将对数据造成的损害降至最小。

另外，数据备份也是防止数据被破坏后无法恢复的重要手段，而硬件备份等更是保证系统可用的重要内容，在高级别的信息系统中采用异地适时备份会有效地防止灾难发生时可能造成的系统危害。

保证数据安全和备份恢复主要从：数据完整性、数据保密性、备份和恢复 3 个控制点考虑。

16.5.4 《基本要求》的管理要求

16.5.4.1 安全管理制度

在信息安全中，最活跃的因素是人，对人的管理包括法律、法规与政策的约束、安全指南的帮助、安全意识的提高、安全技能的培训、人力资源管理措施以及企业文化的熏陶，这些功能的实现都是以完备的安全管理政策和制度为前提。这里所说的安全管理制度包括信息安全工作的总体方针、策略、规范各种安全管理活动的管理制度以及管理人员或操作人员日常操作的操作规程。

安全管理制度主要包括：管理制度、制定和发布、评审和修订 3 个控制点。

16.5.4.2 安全管理机构

安全管理，首先要建立一个健全、务实、有效、统一指挥、统一步调的完善的安全管理机构，明确机构成员的安全职责，这是信息安全管理得以实施、推广的基础。在单位的内部结构上必须建立一整套从单位最高管理层(董事会)到执行管理层以及业务运营层的管理结构来约束和保证各项安全管理措施的执行。其主要工作内容包括对机构内重要的信息安全工作进行授权和审批、内部相关业务部门和安全管理部门之间的沟通协调以及与机构外部各类单位的合作、定期对系统的安全措施落实情况进行检查，以发现问题进行改进。

安全管理机构主要包括：岗位设置、人员配备、授权和审批、沟通和合作以及审核和检查 5 个控制点。

16.5.4.3 人员安全管理

人，是信息安全中最关键的因素，同时也是信息安全中最薄弱的环节。很多重要的信息系统安全问题都涉及用户、设计人员、实施人员以及管理人员。如果这些与人员有关的安全问题没有得到很好的解决，任何一个信息系统都不可能达到真正的安全。只有对人员进行了正确完善的管理，才有可能降低人为错误、盗窃、诈骗和误用设备的风险，从而减小了信息系统遭受人员错误造成损失的概率。

对人员安全的管理，主要涉及两方面：对内部人员的安全管理和对外部人员的安全管理。具体包括：人员录用、人员离岗、人员考核、安全意识教育与培训和外部人员访问管理 5 个控制点。

16.5.4.4 系统建设管理

信息系统的安全管理贯穿系统的整个生命周期，系统建设管理主要关注的是生命周期中的前 3 个阶段(即初始、采购和实施)中各项安全管理活动。

系统建设管理分别从工程实施建设前、建设过程以及建设完毕交付三方面考虑，具体包括系统定级、安全方案设计、产品采购和使用、自行软件开发、外包软件开发、工程实施、测试验收、系统交付、系统备案、等级测评和安全服务商选择 11 个控制点。

16.5.4.5 系统运维管理

信息系统建设完成投入运行之后，接下来就是如何维护和管理信息系统了。系统运行涉及很多管理方面，例如对环境的管理、介质的管理、资产的管理等。同时，还要监控系统由于一些原因发生的重大变化，安全措施也要进行相应的修改，以维护系统始终处于相应安全保护等级的安全状态中。

系统运维管理主要包括：环境管理、资产管理、介质管理、设备管理、监控管理和安全管理中心、网络安全管理、系统安全管理、恶意代码防范管理、密码管理、变更管理、备份与恢复管理、安全事件处置、应急预案管理 13 个控制点。

思考和练习题

(1) 简述国际安全评估标准的发展历程。
(2) 简述信息系统安全评估的目的和作用。
(3) TCSEC 的安全等级是怎样划分的？
(4)《计算机信息系统安全保护等级划分准则》(GB 17859—1999) 安全等级是怎样划分的？
(5) 信息系统安全等级保护基本要求》(GB/T 22239—2008) 规定了不同安全保护等级信息系统的基本保护要求，具体包含哪些类？

第 17 章　数据库系统的安全

今天，人们处于互联网和“无所不在的计算”的时代，各种数据库应用在互联网上被数据库服务提供商以在线的形式提供，或用于电子商务，或用于海量数据查询等各种应用。在这些应用中，数据库本身变得日益复杂，同时，为用户提供的服务响应也越来越丰富，于是，基于网络应用的数据库安全威胁也日益严重，情况也变得日益复杂。

17.1　数据库安全基本条件和安全威胁

安全的数据库系统应具备机密性、完整性与可用性三项基本条件。机密性是为了保护敏感性的数据避免被非法用户窃知；完整性是要防范数据库被有意或无意的破坏，以维持数据的正确性；而可用性是一旦数据库遭受不当的修改时，应能迅速恢复正常运作的能力，如备份或还原。三项基本条件中，机密性约束数据访问者，只有合法用户能够访问数据库中机密信息，完整性和可用性约束数据修改者，只有合法用户能够对数据库中机密信息进行合法修改。

数据库的安全威胁来自于许多不同的途径。如果我们跟踪一个分布式网络数据库应用的数据流过程，两种主要的安全问题就可以被归纳出：安全数据传输和安全数据存储及访问。如果把数据库的安全威胁分为物理的和逻辑的两个方面，那么在逻辑方面的威胁可以包括：信息的暴露，非法访问和篡改数据，以及服务拒绝等；在物理方面的威胁可以包括：访问密码的强行获取，窃取或破坏存储设备，电源破坏等。逻辑威胁和物理威胁都可以是故意的和意外的，如果安全威胁按照其所受攻击的来源来分类，可以分为外部侵入，内部管理漏洞和系统管理员。

17.2　数据库安全层次

数据库的安全层次如图17.1所示。

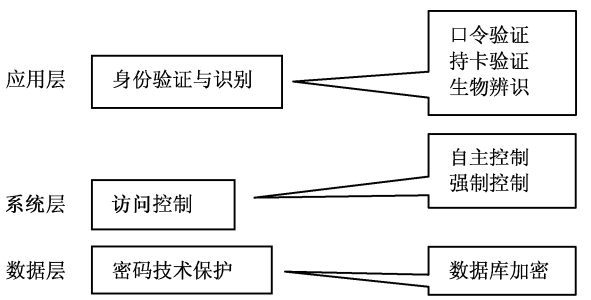


图 17.1　数据库的安全层次

17.2.1　应用层

这一层主要的安全控制手段是身份验证与识别，从身份验证与识别角度出发的几种方法，包括：

(1) **口令验证法**：口令验证法是最常用的身份验证方法，用户通过提供自己的用户名和口令来表明身份，系统一般将用户名和口令或口令的 Hash 码存入稳定存储介质中供将来用户登录时比较。然而口令容易泄露、被猜测或丢失，因此，在实际使用时有较大的风险。

(2) **持卡验证法**：将个人的身份凭证储存在智能卡上，提供用户的身份验证，但仍有被复制伪造以及遗失的风险。

(3) **生物辨识验证法**：这是目前正在蓬勃发展的一种新兴的身份验证与识别方法，采用用户生物特征如指纹、声音、瞳孔与容貌，等等，事先存在计算机内作为判定身份的依据。用户无须记忆或携带额外的身份标识，并且也不存在被猜测或丢失的风险，但是需要配合专用的仪器，因此目前尚未进入普及阶段。

17.2.2　系统层

这一层主要的安全控制手段是访问控制。因为单就身份验证与识别并不足以防范非法的使用，在用户通过身份验证与识别后进入数据库系统存取资源的同时，必须防范蓄意破坏或泄露数据的行为，所以数据库系统均以提供访问控制的数据安全方法将用户依不同身份做合理的授权。一般可分为自主型及强制型访问控制两种。

(1) **自主型访问控制**：自主型的方式可依据企业组织的策略(谁该存取什么数据，对数据能拥有什么权限)来制定访问规则。包括：指定权限与授权能力、虚表(View)机制，以及基于角色的访问控制等。

(2) **强制型访问控制**：在要求保密性高的数据库系统，通常将存取的数据库对象与用户强制分为不同的安全等级(Security Class)，将数据分为绝密(Top Secret，TS)、机密(Confidential，C)、秘密(Secret，S)与无密级(Unclassified，U)四种安全等级，等级关系为 TS>C>S>U，以安全等级为基础执行访问控制的数据库系统，又称多层次数据库系统(Multilevel Database System)。

另外，对于用户也需区分安全等级，并遵循 Bell-LaPadula 模式两个安全策略，即“禁止往上读”(No Read-up)和“禁止往下写”(No Write-down)。

由于前两个层次上的数据库安全研究工作的充分开展，目前多数数据库应用系统都建立了基于前两个层次的安全系统，例如设置防火墙和分布式入侵检测系统、采用口令、访问权控制等。这的确可以在一定程度上遏制黑客的入侵，但每周都有新的系统漏洞发现，手段高明的入侵者和新的攻击手法仍然能得逞。据美国安全杂志“SECURE CYBERSPACE”调查，89%的用户安装了防火墙，60%的用户安装了入侵检测系统，但其中仍有90%的用户的系统安全受到破坏。所以，网络安全措施不是信息安全的全部，来自外部的黑客和来自内部的攻击行为不可避免，而且这些非法攻击的真正目标是数据库。

17.2.3　数据层

在数据库安全领域的研究中，“数据库管理者”对数据库安全带来的安全隐患越来越受到研究者的关注，尤其是基于互联网的网络应用提供商(Internet service provider)模式的数据库系统中，这种安全威胁尤其严重。传统的数据库访问控制方法，对此安全隐患不能提供有效的安全防范。密码学的安全数据库技术，因为其基于数学难解问题的计算复杂性，成为解决数据库安全问题的日渐重要的方法。

基于密码学的数据层安全技术尤为重要，这一层主要的安全手段是密码技术保护。密码技术保护法通过将原本明文存储的数据库文件变为密文存储来保护数据库存储安全，即使攻击者攻破了前两层安全机制的防护，还需要破解数据的密文才能够看到数据明文，而目前广泛使用的加密技术都具有相当高的安全性，想要破解密文十分困难。密码技术保护法包括了数据库加密和数据库验证。

17.3 安全数据库技术及进展

20 世纪 60 年代，多用户资源共享计算机的安全控制问题是安全数据库技术形成的基础，也受到了高度重视。美国国防部组建了受美国国防科学委员会管辖的计算机安全特别行动小组，科研工作者提出了大量的计算机安全威胁，这些安全威胁后来被总结为三个方面：非授权的信息泄露、非授权的信息修改和拒绝服务。该时期的工作主要体现在访问控制抽象、计算机安全基本原理、安全模型和安全操作系统的设计开发几个方面。

1969年，B. W. Lampson第一次使用主体、客体和访问矩阵的思想形式化地对访问控制进行了抽象。同年，作为美国国防科学委员会计算机安全特别行动小组的工作成果，推出了题为《计算机系统安全控制》的报告。该报告结合实际国防信息安全等级划分体制，分析了多用户资源共享计算机系统中机密信息可能受到的安全威胁，提出了解决安全问题的建议方法，其主要目标是多级安全系统在计算机中的实现。报告认为计算机系统的安全控制是一个系统设计问题，必须从硬件、软件、人员和管理等方面综合考虑，并且给出了访问控制问题的形式化描述。此外，还指出了进行计算机安全技术评估的困难。

1972 年，美国空军的一个研究小组在 P. J. Anderson 的带领下进行了计算机安全的研究，并于当年完成了著名的 Anderson 报告。该报告提出了引用监控器、引用验证机制、安全内核和安全建模等重要思想。这些思想是在研究系统资源受控共享问题的背景下产生的。

授权机制与能够对程序运行进行控制的系统环境结合起来可以对受控共享提供支持，授权机制负责确定用户对系统资源的引用访问许可，程序运行控制负责把用户程序对资源的引用访问控制在授权范围之内。引用监控器是一个抽象的概念，它使得人们可以只考虑系统安全相关的方面。它位于主客体之间，根据当前的授权决定是否允许主体访问客体。报告中将引用监控器的实现称为引用验证机制，它是实现引用监控机的软硬件的结合。

1973年，B. Lampson在对程序进行限界问题的研究过程中提出隐通道的概念。他将隐通道定义为常规上不会用于传送信息但是被用来传递信息的渠道。同年，D. E. Bell 和 L. J. Lapadula 提出了第一个可证明的安全系统数学模型，被称为BLP 模型。该模型根据军方的安全策略设计，解决的根本问题是对具有密级划分的信息的访问进行控制。

Harrison等人在1976年提出了一个形式化的访问矩阵模型，该模型最初由 Lampson 提出。该模型的一大贡献就是阐明了保护系统的安全问题。

Anderson 报告之后，陆续启动了一系列的安全操作系统项目，其中早期影响较大的包括安全 Multics 和 Mitre 安全内核。安全 Multics 将 BLP 模型应用到系统中，成为第一个支持模型的系统。

1975 年 Hinke 和 Schaefer 在报告中给出了一个安全数据库研究的内容。该项目的研究目标是设计一个基于操作系统的可信数据库管理系统，该数据库管理系统的访问控制完全由操作系统进行并且试图达到高安全保证。紧接着 I. P. Sharp Associate 开发了一个多级关系数据库管理系统的模型，该模型基于数据库管理系统内部层次化体系结构。

1976 年，IBM 的 P. P. Griffiths 等发表题为“An Authorization Mechanism for a Relational Database System”的论文。文章基于 B. Lampson 的访问控制列表，主要探讨了授权模型，给出了分析关系数据库访问控制模型和机制的理论基础。其中采用的方法是其后若干商业数据库管理系统访问控制机制的基础。

在探索和开发计算机安全系统的同时，人们也在研究如何去衡量计算机系统的安全性。美国国防部从1977 年开始进行计算机安全的初始研究，其间取得了大量的研究和开发成果。经过与美国国家标准局的合作，美国国防部计算机安全中心在1983 年发表了可信计算机评估准则(TCSEC)，TCSEC 是历史上第一个计算机安全评估准则，这之后计算机系统安全研究进入一个新的阶段，数据库安全研究也进入了标准化时期。

TCSEC 给出了不同安全级别计算机系统的安全功能要求和安全保证要求，主要侧重于安全操作系统的评估。这一阶段的主要研究工作是如何开发高安全级别的数据库管理系统。其中比较有代表性的研究项目包括 Seaview, ASD 和 LDV。

Sea View(Secure data view)是美国空军资助，SRI 和 Gemini 公司共同进行的一个研究项目。它的研究目标是实现一个达到 TCSEC A1 级的安全数据库，访问控制粒度达到字段级。Sea View 采用了核心式的体系结构，由操作系统提供强制访问控制。该项目的研究取得了丰富的成果，包括Sea View 安全模型和验证技术、多级关系模型、多实例和安全数据库开发技术等。ASD(Advanced Secure DBMS)实际上是 TRW 国防系统小组的一个项目，该项目的研究目标是实现一个A1 级的安全数据库系统。该项目的目标就是进行级安全数据库技术的研究和开发。LDV(Lock Data View)也是美国空军资助进行的项目，该项目主要研究在LOCK 安全操作系统基础上设计和开发安全数据库系统。

数据库软件开发商也积极响应 TCSEC，开发出一些安全数据库产品。Oracle 的 Trusted Oracle 7 经评估达到了 B1 级，Sybase 的 SQL Server 达到 C2 级，SQL Secure Server 达到B1 级，并且是最早通过B1 级评估的安全数据库系统。Informix 的 INRORMIX-Online/ Secure 5.0 达到了 B1 级。

随着分布式技术的发展，分布数据库技术开始萌发和发展。其次，面向对象技术进入数据库领域形成了面向对象数据库。安全数据库本身的研究也从安全保密转移到完整性相关的工作，包括完整性模型、并发控制、多级事务处理以及安全数据库的备份和恢复等。研究重心的转移，一方面与安全保密数据库技术的成熟有关，另一方面也与数据库在商业领域全面应用导致对数据库完整性的关注有关。

基于角色的访问控制(RBAC)模型是比较有代表性的工作，自1994 年提出经过将近十多年的发展，RBAC 模型目前已经处于比较成熟的阶段。模型采用角色作为主体和客体间的桥梁，易于对企业安全策略建模；另一方面它使用约束对完整性有很好的支持，包括分权、最小特权等。此外，还可以配置模型支持多种安全策略，包括强制和自主安全策略。

支持多种安全策略的模型，比较有代表性的工作包括 Elisa Bertino 和 Sushil Jajodia 等人的研究成果。Elisa Bertino 于 1996 年提出了一个通用的授权模型，可以在关系数据库中支持多种自主访问控制策略。Sushil Jajodia 提出了一个称为多策略访问控制的框架，在该框架下可以在单一系统中使用多种访问控制策略。

面向对象数据库的起步比较早，在 1986 年，第一个商品化的面向对象数据库系统 Genmstone 就诞生了。由于对象数据拥有丰富的语义，其访问控制就必然涉及继承层次结构、复杂对象构造、多版本、控制粒度等问题，此外并发控制和安全恢复也有其特点。针对这些特点研究人员进行了大量工作。

XML 旨在通过各种各样的数据内容交换而使计算机系统协同作业。由于 XML 的广泛应用，使用数据库保存数据是发展的必然结果。如今数据库已经走入实用，已有的商业产品包括 Tamino, Oracle 9i 等。XML 数据库是一个正在展开的研究方向，目前已有的研究工作主要集中于数据安全，包括公司的安全套件、大学的文档细粒度访问控制研究项目等。

17.4 密码学安全数据库

传统的访问控制算法一直以来是主要的安全防范机制，但它已经不能满足今天日益复杂的安全威胁和日益丰富的私密性保护需求。例如，在访问控制模式下，数据库管理员拥有最高的特权，这就有可能产生安全漏洞。为了克服这些包括数据库管理员特权带来的安全漏洞，密码学技术被采用并深入到安全数据库技术的各个方面，可以统称为基于密码学的安全数据库技术。

根据统计，大多数网络数据库的数据被盗事件是源自数据库系统的内部人员，包括数据库操作者和管理者。所以，限制数据库管理员特权滥用，以保护数据的安全隐私就成为当今重要的研究课题。最初的方法是将密码学和访问控制结合起来，如安全管理方法 SA（Security Administer）和安全字典方法SD（Security Dictionary）。两种方法都是试图减少数据库管理员对安全相关的数据操作的干涉。

为了更有效地消除DBA安全漏洞，加密数据库的方法被提出，并且在安全数据库技术中崭露头角。然而，这项技术又带来了一个问题，就是对于加密的数据库内容，很难直接完成数据库的关系逻辑操作和运算。为克服这个障碍，许多加密数据技术被提出，如利用私有同态的原理，该原理是指对加密的数据进行一定的运算；利用次序保持加密数据库算法，该算法的优势在于加密数据库的 B 树索引可以继续保持有效，以用于加密数据库的查询操作。

需要指出的是，对安全数据库足够安全的需求和对加密数据库方便的查询能力的需求，二者不可避免地存在着矛盾。这就要求我们在设计一个密码学安全数据库的时候，必须对二者进行仔细的考察和折中以选择合适的密码学方法。也就是说，必须认真衡量数据库安全特性带来的利益和查询方便性的损失。

17.4.1 数据库加密粒度的选择

按照数据库的结构层次，数据库的加密粒度可以分为相应的数据库级、表级、记录级、字段级和数据项级。根据不同的应用需要，选择合适的加密粒度。

1．数据库级

加密的对象是整个数据库，这意味着对所有的用户数据表、系统数据表、索引、视图和存储过程等都进行加密处理。这种加密方法简单，只需要对存储在磁盘中的相应数据库文件进行加密处理即可，密钥的数量少，一个数据库只对应一个密钥，管理方便。但是，数据库一个重要特征是数据共享性高，被多个用户和应用共享使用，需要接受大量的随机访问。一般来说，用户访问数据库时，是为了将符合条件的记录检索出来。如果采用数据库级加密方式，即使只需要查询少量的记录，也需要对整个数据库进行解密，对系统性能会产生极大的影响。但是，对于辅存中备份的数据库，可以采取这种加密粒度。

2．表级

加密的对象是数据库中的表。通常来说，数据库包含多个表，只需要对其中一些包含敏感信息的表进行加密，以保护它们的安全性。与数据库级加密比较，采用表级加密粒度，

系统的查询性能会有所改善，因为对于未加密表的查询，与传统查询方法一样，系统性能不会受到影响，对于加密表的查询，只需要解密对应的加密表，而不要解密整个数据库。在实行表级加密时，可以采用对存储数据的磁盘块(页面)进行加密。但是，这种方法与 DBMS 集成时，需要对 DBMS 内部一些核心模块进行修改，包括对词法分析器、解释器和查询执行器的修改，而目前一些主流的商用 DBMS 都不开放源代码，很难把这种方法与它们集成起来。

3．记录级(行级)

加密的对象是数据表中的记录，记录中各字段值连接一起进行加密处理，加密后输出一列字符串。在实现记录级加密时，通过调用专门的加密函数，对页面中记录进行加密。例如在IBM DB2中，提供了editproc接口，可以实现记录级数据加密处理。与数据库和表级加密相比，这种加密的粒度更细，可选择的灵活性更好。比如说，一个公司的人事资料，要求对部门经理职位以上的人员采取加密措施进行保密，那么可以只选择这些记录加密，而不必要对所有记录进行加密。但是，和表级加密一样，这种方法也需要对 DBMS 内核进行修改。

4．字段级(列级)

加密的对象是关系中的某个字段。字段级加密是一个很好的选择，因为在实际生活中，一些重要和敏感的信息往往出现在关系中的某些列，如信用卡号、身份证号、银行账号等，只需要对这些重要数据进行加密保护，而没有必要对普通数据也进行加密。在实现字段级加密时，可以采取多种方式，既可以在DBMS外部(比如，应用程序)完成，也可以在DBMS内部(如内模式)完成。

5．数据项级(字段值级)

加密的对象是记录中的某个字段值，它是数据库加密的最小粒度。数据项级加密的方法更为灵活，它的实现方式与字段级加密相似，但其密钥管理将会更加复杂。

17.4.2　基于数据加密的访问控制

在传统的数据库访问控制算法中存在一些固有的缺陷。首先，它是基于服务器方式下的安全管理，DBA拥有最高的特权，从而产生内部安全漏洞；其次，访问控制是基于引用监督策略和访问控制矩阵，这样，在分布式网络数据库系统中，来自于网络的攻击有可能采用绕过访问控制环节，故意提升用户等级或者修改相关系统文件等方式进行攻击。针对这些安全漏洞，研究者们设计出基于密码学的安全访问控制算法。

2003 年，丹麦科技大学的研究组提出了一个针对分布式文件系统密码学安全访问控制算法。该算法将加密算法和访问控制算法相结合来提高传统访问控制算法的机密性，同时提供完整性保护。算法主要基于一个开放的网络体系，传统的访问控制算法不能提供可靠的安全保障。在算法中，客户被分别赋予针对系统中存储的数据“读”和“写”两个权利。在读的状态下，客户被授予“对称密钥”，从而对从服务器获取的数据进行解密运算，还被授予“解密密钥”来验证加密数据的签名信息从而进行认证和完整性检验。在写的状态下，用户被分配“对称密钥”，对即将写入并发送到服务器的数据进行加密运算，同时被分配“加密密钥”对数据进行消息摘要的完整性计算及产生签名。服务器端被赋予“解密密钥”来检验数据的完整性。该算法虽然能限制服务器端的部分安全漏洞，但是算法的计算复杂度比较高，加密密钥和解密密钥一般为公钥体系结构，而且系统的存储负荷也比较大。

1998 年，IBM Watson 实验室的 Jingmin He 和 Min Wang 提出了一种基于密码学的关系

数据库管理系统。该方法引入了一种安全字典的概念，它可以为数据库提供许多安全服务功能，其中包括安全用户管理机制来适应各种关系数据库管理系统的安全需要。在该方法中，需要一个安全环境来存储安全字典。安全字典包括许多索引表和视图，这个安全字典由数据库服务器来维护，并只能通过系统命令的方式进行更新。对它的访问是由一个严格的授权和认证策略来控制的。该系统尚未提出有效的方案来实现在数据库服务器中建立和维护一个安全操作环境(SOE)，尤其是当安全操作环境中的数据需要被服务器频繁更新的情况下。

17.4.3 秘密同态加密算法

数据库加密方法可以不同的环境和应用，采取不同的加密方式，如以文件、记录、字段作为加密基本单位进行加密。但它们存在一个共同的问题是：对于密文数据库，若要对某些字段进行统计、平均、求和等数学运算时，必须先对这些字段进行解密运算，然后对明文进行数学运算，之后再加密。这样增大了时空开销；在实际应用中，对于某些重要或敏感数据，需要满足用户对其进行操作但又不让用户了解其中的信息(例如，在每个雇员的工薪信息保密的情况下给雇员的工薪增加15%)的需要。如果对数据库的密文进行常规的数据库操作，显然就可以避免上面存在的问题，并可以大大削减加、解密所需要的时空开销，大大提高数据库的运行效率。秘密同态(Privacy homomorphism)技术就是一个解决上述问题的有效方法。

秘密同态是由 Rivest 等人于 1978 年中提出的，是允许直接对密文进行操作的加密变换。但是由于其对已知明文攻击是不安全的，后来由 Domingo 做了进一步的改进。秘密同态技术最早是用于对统计数据进行加密的，由算法的同态性，保证了用户可以对敏感数据进行操作但又不泄露数据信息。秘密同态技术是建立在代数理论之上的，其基本思想如下：

假设 E_{k1} 和 D_{k1} 分别代表加密、解密函数，明文数据空间中的元素是有限集合$\{M_1, M_2, \cdots, M_n\}$，a 和 b 代表运算，若 $a(E_{k1}(M_1), E_{k1}(M_2), \cdots, E_{k1}(M_n)) = E_{k1}(b(M_1, M_2, \cdots, M_n))$成立，则称($E_{k1}$，$D_{k1}$，a，b)为一个秘密同态。

17.4.4 在加密数据上实现查询

在实现加密数据库时，仍然会存在一些具体的问题。一是如何加密数据内部的关系数据，也就是如何确定加密的层次，如磁盘的页，整个关系表，单个记录或者是某几个属性字段。采取任何一种方法都会有利有弊；二是如何将 SQL 应用于加密数据的逻辑运算或操作，尤其是实现加密数据的聚合，区间查找和格式匹配等逻辑操作。

1981 年，Davida 等人提出了一种数据库加密系统，这种方法是面向记录加密，对整个记录采用分组密码的方式进行加密，这样就可以防范针对某个属性字段的安全攻击。整个加密后数据块在解密时，每个属性字段可以被单独解密获得，而不会也不必解密其他的属性字段。这个方法可以很好地解决那些数据记录需要整体加密，而各个字段可以分别被不同权利的访问者解密使用的数据库应用场景。中国剩余定理是该方法的技术原理，问题是，因为中国剩余定理是一种基于素数的运算，如何选择素数，以及如何构建一种共用的方法，都是非常复杂的计算过程，所以该方法具有较困难的扩展性。

2002 年，Hacigumus 等人提出了一种安全数据库服务提供模型，它通过改进的 SQL 对加密的数据库进行关系运算。在该方法中，关系表中的记录采用常规的分组加密方法进行加密，对于每个属性字段，需要增加一个附属字段来标示该属性字段(未加密前)所属的数值区间。该附加字段的值用于加密数据库的查询操作，这样，会出现多个加密记录的相同字段对应相

同的附加字段值，在进行某字段的数值匹配查询时，就会产生多个匹配记录，然后，一个“后处理”的过程来解密这些记录，从而最终获得准确的查询结果。“后处理”的计算复杂度取决于属性字段的分段机制。该方法考虑了数据库技术的许多相关方面来适应加密数据库查询的特殊要求，例如语法规则，代数架构和查询优化等。 它的问题在于，如果字段值的分段过于粗糙，“后处理”的计算复杂度就越高，如果字段值的分段越细，加密数据的安全性就越低。

2000年，Song等人提出了一种实际的技术手段来解决加密数据库的查询问题，该方法针对的环境是，在一个“不可信”的服务器中存储了一些加密的文档，用户需要通过查询包含某个字(关键词)的文档。该算法把流密码和分组密码结合起来，以加速和减轻算法的处理开销。具体方法是，文档被分割成字的序列，每个字被一段伪随机序列以特殊的结构进行加密，当一个用户需要在加密的文档中查询一个字时，他给不可信服务器发送一段有关该字的“最小信息”，然后服务器进行查询并返回查出的结果。共有四种方案被提出，以逐步增加查询的安全等级。方法中，服务器可以获得用户需要查询的关键词的明文，同时服务器还可以获取关于这个词的加密值的一些信息，所以算法的安全性是有待提高的。

17.4.5　次序保留的加密数据库

一般来说，加密的数据库在执行查询操作时，索引文件是无法支持的，由于索引文件的失效，还会引申带来其他一些不利因素，即数据库在非加密时的具有的查询便利特性，在加密数据库中就很难支持了。为了解决这一问题，一种次序保留的加密数据库技术被提出了。

然而，需要指出的是，对于加密数据库的高安全性需求，是和针对加密数据库中加密数据查询操作的便利性相矛盾的。 一般来说，如果数据库因为加密带来的性能降低的程度是可控的，用户更愿意使用加密的数据库而不是“非加密”的数据库。

2003年，Ozsoyoglu等人提出了一种可查询的加密数据方法。该方法采用一组严格单调增加的，而且是可逆的多项式函数，作为加密运算，以一种多层嵌套的方式来构成次序保持的加密数据库，这样，数据库原有的B+树等类型的索引文件可以仍然保持有效，从而获得高效的加密数据库的查询。该算法的主要贡献在于其无损的多层加密函数，把对数运算和多项式运算结合起来，以防止整数的溢出错误和实数的精度错误。该算法对浮点数的精度(有效值)部分和指数部分分开进行运算。该方法的主要问题在于没有考虑数据库字段的输入数据的分布特性，也就是说加密后的数据和原数据的概率分布特性很相似。该方法需要一个独立的安全第三方来进行加解密运算。

另一种称为 OPES(Order Preserving Encryption Scheme)次序保持加密方法，是基于如下的应用环境条件：

(a) 数据库软件的存储系统是安全性脆弱的。

(b) 数据库软件是安全可信的。

(c) 所有的磁盘数据是加密的，包括数据库关系表，属性字段名和相应的值。

该方法采用数值分布变换的方式来实现次序保留，主要包括 3 个步骤：

(1) **建模**：对输入和输出的目标分布函数进行建模，分别变换成分段线性函数。

(2) **均匀化**：明文数据库被变换成一个平滑数据分布的数据库，数据库中的值是均匀分布的。

(3) **变换**：平滑数据库被变换成一个密文数据库，使数据库的值成为目标分布的特性。

该方法的安全特性是基于一个平滑函数，作为加解密函数，函数中的两个参数作为密钥。

该方法的缺陷在于：只能防范密文攻击；算法成立的前提条件是攻击者不掌握原始数据的任何相关信息(数值分布特点)，这是个很难满足的条件。另外，此方法不能很好适应数据库在服务提供模式下的应用安全，因为方法没能对数据库的管理者进行有效的安全限制。

17.5 主要商用安全数据库

Oracle公司在其产品——Oracle 8的第二版(8.1.6)第一次允许数据和应用程序管理者可以在数据库里直接加密字符(string)数据、二进制(binary)数据和大对象(LOB)数据，通过使用内置的 DBMS OBFUSCATION TOOLKIT 包完成加密任务。该包包含加密函数 DES Encrypt( )和解密函数 DES Decrypt( )，在数据传入DBMS之前，对字段值进行加密。当用户访问加密数据时，工具包对用户提供透明的访问方式，首先检查用户的合法性，然后产生密钥，选择加密数据并解密，最后返回结果集给用户。该包支持DES和3DES的加密算法，也支持数字摘要 MD5 算法，在 Oracle 9i 中还将支持 AES 加密算法。但是，美国对安全出口产品管理很严，在低于 Oracle 9i 的版本里，只能支持 56 位长度的密钥。在 Oracle 的所有产品中，不支持对索引字段进行加密处理，这在很大程度上限制了加密功能在实际中的使用。

IBM 公司在其产品 DB2 UDB(Universal Database)7.1 版第一次引入对字符型数进行加/解密的功能，并在 DB2 UDB 7.2 版和更高版本中继续使用。具体来说，DB2 UDB 支持 CHAR 和 VARCHAR 数据的字段级加密，它采用加密算法是分组算法 RC2，密钥的长度为 128 位，通过数字摘要算法MD2从加密口令获得，与用户认证口令是相互独立。存储加密数据时，要么在数据操作语句(DML)通过显式调用加密函数(Encrypt)，要么通过触发器监视插入(insert)和更新(update)语句，自动完成数据加密过程。读取加密数据时，不会自动完成解密，而是要求在选择语句(select)显式调用解密函数。如果不提供口令，数据以加密形式从表中读出；如果口令是错误的，将返回错误的结果。加密口令的改变是通过更新语句(update)完成，首先使用旧的口令解密数据，然后使用新的口令重新加密数据。密钥管理的工作由用户或应用程序负责。这种方法支持加密字段的索引，但是，由于索引是建立在密文而不是明文的基础上，功能受到许多限制。比如说它不能支持加密数据的范围查询。

IBM Cloudscape 数据库也包含加密功能，首先在 Informix Cloudscape 3.5 版引入了加密功能，并在随后的更高版本中继续使用。我们以5.0.9版为例，对其加密功能进行说明。Cloudscape DBMS 支持数据库级的加密，即数据库中所有数据都以加密形式存储在磁盘上，包括用户数据、索引数据、元数据和系统日志，等等。它使用加密算法为DES, 3DES-EDE 和 Blowfish，其中，DES 为默认的加密算法。存储加密数据时，在页面写入磁盘时进行加密：读取加密数据时，从磁盘读取时进行解密，只需要在第一次连接数据库时提供加密口令。

Microsoft公司在SQL Server 2000 中并没有提供内置的数据库加密功能。但是，在 Windows 2000 Server 中包含加密文件系统(Encrypted FileSystem，EFS)，可以用来对数据库中的各种数据(如触发器、存储过程、自定义函数和视图)进行文件级加密，Windows XP也同样支持 EFS 功能。EFS 是对 NTFS 文件系统的一种扩充，可以把 NTFS 文件以加密的形式存储在磁盘上。在 EFS 中，通过 WindowsCryptoAPI 组件完成加密和解密功能以及密钥管理。EFS 既包括对称加密算法，也包括公开密钥加密算法。在对文件内容加密时，使用了对称的分组算法 DESX，对应的密钥称为文件加密密钥(File Encryption Key，FEK)，在安全存放 FEK 时，使用了公开密钥加密算法RSA，公钥用于加密存放FEK 的文件，私钥由用户保管，用于解密存放FEK 的

文件。实际上，这种加密保护是在操作系统一级对数据库中的数据进行保护，加/解密过程对DBMS、用户和应用程序都是透明的。

Sybase 公司在其产品Adaptive Server Anywhere 8.0 版引入了数据库加密功能。它支持数据库级加密，加密功能是建立在文件基础上，数据库的所有文件都以加密形式存储，而且对于同一数据库，所有文件的加密密钥是相同的。加密使用的算法为 AES 和 MDSR，其中，AES 为默认的，密钥由用户在创建数据库时指定，密钥长度为 128 位。访问加密数据时，必须解密加密数据库的所有文件。它不支持对已经存在的数据库进行加密处理，如果要加密，只有新建一个加密数据库，把原来数据库中的所有表重新装载到新的数据库中。对于密钥的变更，也只能采取同样的方法。

思考和练习题

(1) 简述安全数据库安全处理层次。
(2) 简述密码同态加密数据库的基本原理。
(3) 简述次序保留加密数据库的基本原理。

实践/实验题

设计密码同态和次序保留的加密数据库模型。

参 考 文 献

[1] 赵战生．信息安全保障技术发展．第五届“全国计算机高级人才培训班”讲义，2001.

[2] 沈昌祥．信息安全工程技术．第五届“全国计算机高级人才培训班”讲义，2001.

[3] 赵战生．信息安全保障的政策、法规和标准．第五届“全国计算机高级人才培训班”讲义，2001.

[4] 赵战生．冯登国等．信息安全技术浅谈．北京：科学出版社．1999.

[5] [TCSEC1985] Department of Defense of USA，Trusted Computer System Evaluation Criteria.（Orange book），1985.

[6] William Stallings，孟庆树等译．密码编码学与网络安全——原理与实践（第四版）．北京：电子工业出版社．2007.

[7] B. Schneier.“Applied Cryptography: Protocols, Algorithms, and Source Code in C, 2nd ed.”, Published by John Wiley & Sons, Inc. 1996.

[8] 李克洪主编．实用密码学与计算机安全．沈阳：东北大学出版社．1997.

[9] 王育民，刘建伟编著．通信网的安全——理论与技术．西安：西安电子科技大学出版社．2000.

[10] http://williamstallings.com/Crypto3e.html

[11] 冯登国，裴定一．密码学导引．北京：科学出版社．1999.

[12] 冯登国等译．密码学原理与实践（第二版）．北京：电子工业出版社． 2003.

[13] 周玉洁，冯登国编著．公开密钥密码算法及其快速实现．北京：国防工业出版社．2002.

[14] 南湘浩，陈钟编著．网络安全技术概论．北京：国防工业出版社．2003.

[15] 段云所等．信息安全概论．北京：高等教育出版社．2003.

[16] http://www.pgp.com

[17] Alfred J. Menezes，Handbook of applied cryptography，CRC Press，1997.

[18] MIT Kerberos Web Site，http://web.mit.edu/kerberos/www

[19] 冯登国．计算机通信网络安全．北京：清华大学出版社．2001.

[20] David F. Ferraiolo etc，Proposed NIST Standard for role-based access control，ACM Transaction on information and system security，August 2001，Vol.4，No.3，p 224～274.

[21] Ravi S Sandhu and Pieraqngela Samarati，Access control：Principle and Practice，IEEE Communication magazine，Sept. 1994，p 40～48.

[22] GB/T 9387.2—1995，中华人民共和国国家标准．信息处理系统开放系统互连基本参考模型第2部分：安全体系结构．中国国家质量技术监督局．1995.

[23] Carl Ellison and Bruce Schneier，Ten risks of PKI：What You're not being Told about Public Key Infrastructure，Comptuer security journal，2000，Vol XVI，No.1，p1～8.

[24] 南湘浩等译．为人所不知的PKI十大风险．网络安全技术与应用．2003.

[25] http://ca.pku.edu.cn

[26] 北京启明星辰有限公司．防火墙原理与实用技术．北京：电子工业出版社．2002.

[27] William R Cheswick．戴宗坤译．防火墙与因特网安全．北京：机械工业出版社．2000.

[28] SOCKS Protocol Version 5，RFC1928.

[29] Terry William Ogletree．防火墙原理与实践．北京：电子工业出版社．2001.

[30] Doraswamy N.，Harkins D．京京工作室译．IPSec：新一代因特网安全标准．北京：机械工业出版社．2000.

[31] SSL v3 spec，http://home.netscape.com/eng/ssl3/

[32] http://www.ietf.org/rfc/rfc2246.txt

[33] Openssl，http://www.openssl.org

[34] Hackers Beware．黑客——攻击透析与防范．北京：电子工业出版社．2002.

[35] 黑客大曝光(第二版)．北京：清华出版社．2002.

[36] Remote OS detection via TCP/IP Stack FingerPrinting，http://www.insecure.org/nmap/nmap-fingerprinting-article.html

[37] The Art of Port Scanning, http://www.insecure.org/nmap/nmap_doc.html

[38] Michael Howard, David LeBlanc 著．编写安全的代码．微软公司核心技术书库．北京：机械工业出版社.

[39] 陈立新编著．计算机病毒防治百事通．北京：清华大学出版社．2000.

[40] 毛明，王贵和，何建波等编著．计算机病毒原理与反病毒工具．北京：科学技术文献出版社．1995.

[41] 精英工作室编著．计算机病毒防治完全手册．北京：中国电力出版社．2000.

[42] 卢开澄，郭宝安等编著．计算机系统安全．重庆：重庆出版社．1999.

[43] 牛少彰主编．信息安全概论．北京：北京邮电大学出版社．2004.

[44] 李剑编著．信息安全导论．北京：北京邮电大学出版社．2007.

[45] 赵树升，赵韶平编著．Windows 信息安全原理与实践．北京：清华大学出版社．2004.

[46] Anonymous 著，詹文军等译．Windows 安全黑客谈．北京：电子工业出版社．2002.

[47] Mandy Andress 著，杨涛等译．计算机安全原理．北京：机械工业出版社．2002.

[48] 陈波，于冷等编著．计算机系统安全原理与技术．北京：机械工业出版社．2006.

[49] Charles P. Pfleeger．Shari Lawrence Pfleeger 著．李毅超等译．信息安全原理与应用(第四版)．北京：电子工业出版社．2007.

[50] Matt Bishop 著，王立斌．黄征等译．计算机安全学导论．北京：电子工业出版社．2005.

[51] 石志国等编著．信息安全概论．北京：清华大学出版社．北京交通大学出版社．2007.

[52] Chuck Easttom 著，贺民等译．计算机安全基础．北京：清华大学出版社．2008.

[53] Rebecca Gurley Bace 著，陈明奇等译．入侵检测．北京：人民邮电出版社．2001.

[54] 韩东海，王超等．入侵检测系统及实例剖析．北京：清华大学出版社．2002.

[55] Paul E Proctor 著，邓琦皓等译．入侵检测实用手册．北京：中国电力出版社．2002.

[56] Stephen Northcutt，余青霓等译．网络入侵检测分析员手册．北京：人民邮电出版社．2000.

[57] 潘柱廷．入侵检测．第五届“全国计算机高级人才培训班”讲义．2001.

[58] Christos Douligeris，Dimitrios N. Serpanos，Network security current status and future directions，Published by John Wiley & Sons, Inc. 2007.

[59] Greg Holden 著，王斌，孔璐译．防火墙与网络安全——入侵检测和 VPNs．北京：清华大学出版社．2004.

[60] Atul Kahate 著，金名等译．密码学与网络安全(第二版)．北京：清华大学出版社．2009.

[61] 武新华主编．黑客攻防技术 24 小时轻松掌握．北京：中国铁道出版社．2006.

[62] 南湘浩著．CPK 密码体制与网际安全．北京：国防工业出版社．2008.

[63] 张仁斌等编著．计算机病毒与反病毒技术．北京：清华大学出版社．2006.

[64] David Salomon 著，蔡建，梁志敏译．数据保密与安全．北京：清华大学出版社．2005.

[65] F. L. Bauer 著，吴世忠等译．密码编码和密码分析原理与方法．北京：机械工业出版社．2001.

[66] 冯登国．密码分析学．北京：清华大学出版社．2000.

[67] 龙冬阳编著．应用编码与计算机密码学．北京：清华大学出版社．2005.

[68] 樊宸丰，林东编著．网络信息安全&PGP 加密．北京：清华大学出版社．1998.

[69] Steve Burnett，Stephen Paine 著．冯登国等译．密码工程实践指南．北京：清华大学出版社．2001.

[70] Michael E. Whitman，Herbert J. Mattord 译．信息安全原理．北京：清华大学出版社．2006.

[71] Schultz E E. Intrusion Detection Revisited. Network Security. 2000，14(2):6～9.

[72] Frincke D A，Huang M Y. Recent Advances in Intrusion Detection Systems. Computers Networks. 2000，34(4):541～545.

[73] Zambomi D，Spaford E H. A Framework and Protocol for a Distributed Intrusion Detection System. Technical Report. Purdue University.1998.

[74] Mukherjee B， Heberlein L T. Network Intrusion Detection. IEEE Network. 1994，5:26～41.

[75] 张然，钱德沛，过晓兵．防火墙与入侵检测技术．计算机应用研究．2000，22(1)．

[76] http://www.venustech.com.cn/NewsInfo/222/452.Html

[77] http://www.ranum.com/security/computer_security/editorials/deepinspect

[78] http://www.dvr100.com/jishuzhongxin/tongxunjishu/2007/06-23/32724.html

[79] http://www.cert.com/articles/tabloid/common/200308720958.shtml.状态检测工作机制．

[80] http://en.wikipedia.org/wiki/Deep_packet_inspection

[81] http://ec.icxo.com/htmlnews/2007/07/10/1157872.htm

[82] Schiller, J.,“Cryptographic Algorithms for use in the Internet Key Exchange Version 2（IKEv2）”, RFC 4307, December 2005.

[83] 3rd Eastlake, D.,“Cryptographic Algorithm Implementation Requirements For Encapsulating Security Payload（ESP）and Authentication Header（AH）”, RFC4305, December 2005.

[84] Harkins, D. and D. Carrel,“The Internet Key Exchange（IKE）”, RFC2409, November 1998.

[85] Kaufman, C., Ed.,“The Internet Key Exchange（IKEv2）Protocol”, RFC4306, December 2005.

[86] Kent, S.,“IP Encapsulating Security Payload（ESP）”, RFC4303, December 2005.

[87] Kent, S.,“IP Authentication Header”, RFC4302, December 2005.

[88] Frankel, S. and H. Herbert,“The AES-XCBC-MAC-96 Algorithm and Its Use With Ipsec”, RFC3566, September 2003.

[89] Frankel, S., Glenn, R., and S. Kelly,“The AES-CBC Cipher Algorithm and Its Use with Ipsec”, RFC 3602, September 2003.

[90] Housley, R.,“Using Advanced Encryption Standard（AES）Counter Mode With IPsec Encapsulating Security Payload（ESP）”, RFC 3686, January 2004.

[91] Kent, S. and K. Seo,“Security Architecture for the Internet Protocol”, RFC4301, December 2005.

[92] V. Manral，Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload（ESP）and Authentication Header（AH），RFC4835，April 2007.

[93] T. Dierks, E. Rescorla,The Transport Layer Security（TLS）Protocol Version 1.2,RFC5246, August 2008.

[94] National Institute of Standards and Technology. Specification for the Advanced Encryption Standard (AES). FIPS 197. November 26，2001.

[95] GB/T 18336—2001．中华人民共和国推荐标准．信息技术 信息技术安全性评估准则．中国国家质量技术监督局．2001.

[96] GB 17859—1999，中华人民共和国国家标准．信息安全技术计算机信息系统安全保护等级划分准则．中国国家质量技术监督局．1999.

[97] GB/T 22239—2008，中华人民共和国国家标准．信息安全技术 信息系统安全等级保护基本要求．中国国家质量技术监督局．2008.

[98] C E Shannon. Communication Theory of Secrecy Systems. Bell System Technical Journal. Oct 1949, 28:656～715.

[99] C E Shannon. Prediction and Entropy of printed English. Bell System Technical Journal. Jan 1951, 30: 50～64.

[100] DiffieW. Hellman,M. New Directions in Cryptography. IEEE Trans on Inform. Theory. 1976, Vol.IT-22(6):644～654.

[101] J. N. E. Bos and D. Chaum. Provably unforgeable signatures. Lecture Notes in Computer Science, 740(1993), 1～14（Advances in Cryptology-CRYPTO'92）

[102] D. Chaum, R. L. Rivest and A. T. Sherman. Advances in Cryptology: Proceedings of CRYPTO'82. Plenum Press, 1983.

[103] D. Chaum and Van Heijst E. Group Signatures. Advances in Cryptology -Eurocrypt'91. Springer-Verlag, 1991, p.257～265.

[104] Stadler M., Piveteau J.M. and Camenisch J. Fair Blind Signatures. Advances in Cryptology-Eurocrypt'95. Springer-Verlga. p.209～219.

[105] IBC 和 PKI 的组合应用研究．解放军信息工程大学博士论文．2007.

[106] 谢冬青，冷健编著．PKI 原理与技术．北京：清华大学出版社．2004.

[107] 余胜生，龙春，周敬利，胡熠峰．一种 PKI 混合多级信任模型的分析与实现．计算机应用．2003，23(10):18～20.

[108] 张涛，董占球．网络攻击行为分类技术的研究．计算机应用．April 2004，24(4)．

[109] NCSC-TG-021，Trusted Database Management System Interpretation of TCSEC[S]. US: National Computer Security Center, 1991.

[110] NCSC-TG-005, Trusted Network Interpretation of TCSEC [S]. US: National Computer Security Center, 1987.

[111] Canadian Trusted Computer Product Evaluation Criteria（CTCPEC), Version 3.0[S], Canadian System Security Centre, Communications Security Establishment, Government of Canada, 1993.

[112] Federal Criteria for Information Technology Security（FC). Draft Version 1.0（Volumes I and II） US: National Institute of Standards and Technology and the National Security Agency, 1993.

[113] Information Technology Security Evaluation Criteria（ITSEC), Version 1.2 [S]. Office for Official Publications of the European Communities, 1991.

[114] ISO/IEC 15408, 1999(E), Common Criteria for Information Technology Security Evaluation [S]. The International Organization for Standardization,1999.

[115] Elisa Bertino. Data security, Data&Knowledge Engineering 25（1998）199～216.

[116] D. E. Denning. Commutative filters for reducing inference threats in multilevel database systems, Proc.1985 IEEE Symposium on Security and Privacy, Oakland,CA, IEEE Computer Society, April 1. 985, p. 134～146.

[117] Ramez Elmasri and Shamkant Navathe. Fundamentals of Database Systems Second Edition.

[118] Silvans Castano, M. Fugini, G. Martella, and P. Samarati. 1995, Addison-Wesley Database Security.

[119] D. E. Denning and P.J. Denning. Data Security, ACM Computing Surverys, Vol. l l, No. 3, 1979, p.227～249.

[120] Simson Gafinkel with Gene Spafford. Web Security&Commerce, 'REILLY.

[121] Gail-Joon Ahn. Role-based access control in DCOM, Journal of System Architecture 46（2000）1175～1184.

[122] A. Lin, R. Brown. The application of security policy to role-based access control and the common data security architecture 1584～1593 Communications, 23（2000）.

[123] Duen-Ren Liu, Mei-Yu Wu, Jing-Jang Hwang. Designing Authorization Rules for Separation of Duty in Task-based Access Control, Journal of Information Management, Vol 8, No.l.

[124] D.E. Bell, L.J. LaPadula. Secure Computer Systems: Unified Exposition and Multics Interpretation（The Mitre Corp, MA）.

[125] Committee on Multilevel Management Security. Multilevel Data Management Security, Tech. report. Air Force Studies Board, National Research Council, 1982.

[126] 余祥宣．华中科技大学．http: //www.andin-info.com.cn/html/news-luntan.ppt 2004.3. S.

[127] Lampson B W. Protection. Proc. of The 5th Princeton Symposium on Information Science and Systems, March, 1971. p437～443.

[128] Denning P J. Protection-principles and practice. Proc. of the Spring Joint Computer Conference, volume 40, Montvale, New York, 1972.

[129] Harrison M A et al. Protection in operation system. Communications of the ACM, 19（8）, 1976.14～24.

[130] Denning D E. Database Security[M]. Annual Review Inc., 1988.

[131] Costich O, et al. A multilevel Transaction Problem for Multilevel secure database systems and its solution for replicated architecture. Proc. of IEEE computer society symposium on research in security and privacy. Oakland, CA, 1992. p192～203.

[132] Denning D E, et al. A multilevel relational data model In proceedings of the first international workshop on object-oriented database systems. Pacific Grove,1986.

[133] Dwyer P A, et al. Multilevel Security in database management systems, Computers and security, 6（3）, 1987. p252～260.

[134] Harrington A, et al. Cryptographic access control in a distributed file system. Proceedings of the eighth ACM symposium on Access control models and technologies. Como, Italy, 2003. p158～165.

[135] He J, et al. Cryptography and relational database management system Proc. of Database, Engineering and Application Symposium. Grenoble,France,2001. p273～284.

[136] Mattsson U, et al. Secure Data Functional Overview[M]. Protegity Technical Paper TWP-0011,2000.

[137] Davida G L, et al. A database encryption system with subkeys. ACM Transactions on Database Systems. Volume 6, Issue 2, June, 1981.p312～328.

[138] Hacigumus H, et al. Executing SQL over encrypted data in the database-service-provider model. ACM

SIGMOD Conference. Madison, Wisconsin, USA,2002. p216～227.

[139] Song D X, et al. Practical techniques for searches on encrypted data. IEEE Symposium on Security and Privacy. Los Alamitos: IEEE Computer Society Press,2000. p44～55.

[140] Gultekin S C, et al. Anti-tamper databases: Querying encrypted databases. Proc. of the 17th Annual IFIP WG 11.3 Working Conference on Database and Applications Security. Estes Park, Colorado, August. 2003. p133～146.

[141] Agrawal R, et al. Order preserving encryption for numeric data. Proceedings of the 2004 ACM SIGMOD international conference on Management of data, Paris, France. 2004. p563～574.

[142] Sinkov,A. Elementary cryptanalysis: A mathematical Approach. Washington, DC: The mathematical association of America, 1966.

[143] K.W.Campbell and M.J.Wiener. Proof that DES is not a group.In Advances in Cryptology: Proceedings of CRYPTO '92，p518～526，Santa Barbara，CA，1993. Springer-Verlag.

[144] 陆浪如．信息安全评估标准的研究与信息安全系统的设计．解放军信息工程大学博士论文．2001.

[145] Yuan Chun，Wen Zhen-Kun，Zhang Ji-Hong，Zhong Yu-Zhuo. 密码学访问控制和安全数据库技术现状. 电子学报. Journal of Tien Tzu Hsueh Pao. 2006，34(11): 2043～2046.

[146] 王昭，段云所，陈钟．信息安全的政策法规和标准．网络安全技术与应用．2001, 11: 61～64.

[147] 王昭，段云所，陈钟．数据加密算法的原理与应用．网络安全技术与应用．2001, 2:58～64.

[148] 王昭，段云所，陈钟．防火墙技术简介．网络安全技术与应用．2001, 8:61～64.

[149] ANSI X9.9, American National Standard for Financial Institution Message Authentication（Wholesale）, American Bankers Association, 1981.

[150] FIPS PUB 113, Computer Data Authentication, National Institute of Standards and Technology, 1985.

[151] ISO/IEC 9797, Information Technology—Security Techniques—Data Integrity Mechanism Using a Cryptographic Check Function Employing a Block Cipher Algorithm，ISO/IEC, 1994.