计算机网络信息安全研究

耿新宇◎著

天津出版传媒集团
天津科学技术出版社

计算机网络信息安全研究

耿新宇　著

天津出版传媒集团
天津科学技术出版社

内 容 简 介

本书是一本计算机网络信息安全专业著作，主要内容包括：信息加密技术的探究、局域网安全技术探究、病毒及其防范技术探究、数字认证与 VPN 技术探究等内容。本书在内容选取上，力求反映计算机网络安全的新问题、新技术和新应用，满足构造计算机网络安全的需要。全书理论性、知识性、技术性较强，向读者介绍计算机网络安全的理论知识和常用技术，具有一定的学术价值。

图书在版编目（CIP）数据

计算机网络信息安全研究 / 耿新宇著. —天津:
天津科学技术出版社, 2015.2
ISBN 978-7-5308-9580-1
Ⅰ. ①计… Ⅱ. ①耿… Ⅲ. ①计算机网络—信息安全
—研究 Ⅳ. ①TP393.08
中国版本图书馆 CIP 数据核字(2015)第 034245 号

责任编辑：刘丽燕
责任印制：兰 毅

天津出版传媒集团
天津科学技术出版社 出版

出版人：蔡 颢
天津市西康路 35 号 邮编 300051
电话（022）23332490
网址：www.tjkjcbs.com.cn
新华书店经销
天津印艺通制版印刷责任有限公司印刷

开本 787×1092 1/16 印张 9.75 字数 234 000
2015 年 2 月第 1 版第 1 次印刷
定价：58.00 元

前　言

网络安全是一门涉及计算机科学、网络技术、通信技术、密码技术、信息安全技术、应用数学、数论、信息论等多个学科的综合性学科。随着计算机网络的普及和发展，我们的生活和工作越来越依赖于网络，与此相关的网络安全问题也随之凸现出来，并逐渐成为网络应用所面临的主要问题。

网络发展的早期，人们更多地强调网络的方便性和可用性，忽略了网络的安全性。当网络仅仅用来传送一般性信息的时候，当网络的覆盖面积仅限于一幢大楼、一个校园的时候，安全问题并没有突出地表现出来。但是，当在网络上运行关键性的信息如银行业务等，当企业的主要业务运行在网络上，当政府部门的活动日益网络化时，计算机网络安全就成为一个不容忽视的问题。随着组织和部门对网络依赖性的增强，一个相对较小的网络也突显出一定的安全问题，尤其是组织部门的网络，要面对来自外部网络的各种安全威胁，即使是网络出于自身利益的考虑没有明确的安全要求，也可能由于被攻击者利用而带来不必要的法律纠纷。网络黑客的攻击、网络病毒的泛滥和各种网络业务的安全要求已经构成了对网络安全的迫切需求。

近几年来，有关计算机网络安全方面的著作不断涌现，这些著作各有特点，为各层次各类型读者提供了宝贵的资料，也指导和帮助着国内计算机网络安全技术的应用与研究。本书有以下两个方面的主要特点。

第一是通俗易懂。计算机网络安全的理论性、知识性、技术性较强，本书以清晰的思路、合理的体系、通俗的语言，向读者介绍计算机网络安全的理论知识和常用技术。

第二是注重实用。学习本书可使读者方便地掌握计算机网络安全的概念，掌握设计和维护网络及其应用系统安全的手段和方法，熟悉使用常见安全技术解决安全问题。在内容选取上，力求反映计算机网络安全的新问题、新技术和新应用，满足构造计算机网络安全的需要。

作者在向读者推荐本书的同时，也深感计算机网络安全技术的博大精深、日新月异，以编者的现有水平很难在本书中给予全面、准确和及时反映，书中难免会有疏漏甚至错误，在此恳请读者和专家批评指正。

作者

2014 年 1 月

目　　录

第一章　信息加密技术的探究

第一节　信息加密技术的发展历程

信息加密技术是一个既古老又新颖的领域。加密（Encryption），一般是指这样一个过程：将一组信息（或称明文，Plaintext）经过密钥（Key）及加密函数的转换，变成无阅读意义的密文（Ciphertext），而接收方则将此密文经过解密（Decryption）密钥和解密函数还原成明文。其基本模型如图 1.1 所示。事实上要想保密，最简单的做法就是不把它告诉别人，知道“秘密”的人越多，泄密的可能性越大，最后秘密也不称为秘密了。

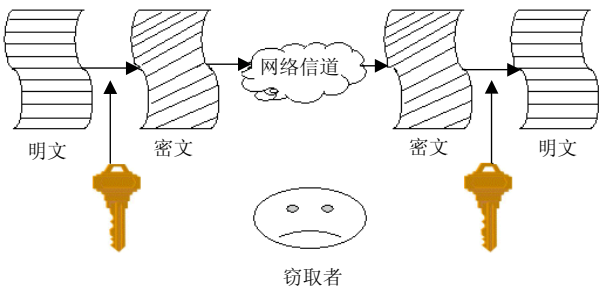

图 1.1　信息加密基本模型

在古代，保守一个秘密似乎要容易一些，因为只有少数人才有读书、写字的特权，如果一个秘密是书写下来的，那么只有数量极少的人才知道它是什么意思。随着越来越多的人掌握了读写文字的能力，越来越有必要在这些人中保守秘密。

早期的加密方法非常简单。据说恺撒大帝曾用一种初级的密码来加密消息，对那些他认为能够分享秘密的人，便告诉他们如何重新组合原来的消息。这种密码便是著名的“恺撒密码（The Caesar Cipher）”。它其实是一种简单的替换加密法：字母表中的每个字母依次都被其后的第三个字母取代。换言之，字母 A 变成 D、B 变成 E……X 变成 A、Y 变成 B、Z 变成 C，依此类推。这种加密技术的一个变种是 ROT-13 密码，每个字母均循环移动 13 个位置。

简单的替换加密存在重大的缺陷，因为重复出现的某个字母总是会用相同的字母替代。通过对某种语言的分析，便可知道字母被移位的大致距离。

在古代，人和人之间的身份验证也很重要。如果只有少数人能读会写，那么签名就足以证明一个人的身份。但随着掌握读写技能的人越来越多，印章逐渐成为“签署人”的一种独特的记号。利用这种记号，便可证明信件、文档和法令签署人的身份确实无误。但

随着技术的发展，人们可轻松仿制出各式各样的印章，所以它也失却了原先的“独特”性。

发展到近代，密码和与之对应的译码技术在历史上占据了重要的地位。第二次世界大战中，德国政府使用一种名为 Enigma 的加密设备，对自己的通信进行加密。这种设备使用了一系列转轮（Enigma 机器共准备了五个，但每次通信的时候，均只使用其中的三个）。这些转轮包含了字母表中的所有字母，每个都可以单独进行设置。对正常输入的文字来说，其中每个字母都被转换成“看似”随机的输出字符。之所以说它“看似”随40 机，是由于换位顺序的组合是一个天文数字。对 Enigma 机器的破解首先由波兰发起，最后由英国完成。

自恺撒大帝的年代开始，一直到当代，通信技术在稳步地发展。从信件到电报、电传、电话、传真以及 E-mail，人和人之间的通信变得越来越方便和普遍。与此同时，保障这些通信的安全也逐渐成为一项重要课题。

一种通信方法的安全取决于建立通信的那种媒体。媒体越开放，消息落入他人之手的可能性越大。现代通信方法一般都是开放和公用的。打一次电话，或者发一次传真，信号会穿越一个共享的、公共的“电路交换”网络。而发一次 E-mail 也会穿越一个共享的、公共的、包交换的网络。在网络中，位于通信双方两个端点之间的任何一个实体均可将消息（信号）轻易拦截下来。如果要通过现代的通信技术来进行数据的保密传输，便必须采用某种形式的加密技术，防范那些“偷窥者”窃取秘密。

现代的基本加密技术要依赖于消息的接收者已知的一项秘密。通常，解密方法（即“算法”）是任何人都知道的，就像所有人都知道怎样打开门一样。然而，真正用来解开这一秘密的“密钥”却并非人人皆知——就像钥匙一样，一扇门的钥匙并不是任何人都拿得到的。当然，还有某些加密系统建立在一种保密的算法基础上，通常把它称为“隐匿保密”。但大多数研究者都反对使用这种加密方法，因为它未向公众开放，人们无从得知它的加密能力到底有多强，是否存在缺陷等（目前针对“加密芯片”展开的辩论便是这样的一个典型例子）。

加密工具并非只有单独的一种。有多种技术都可用来加密信息、安全地交换密钥、维持信息完整以及确保一条消息的真实性。将所有技术组合在一起，才能在日益开放的环境中，提供保守一项“秘密”所需的各项服务。

其实，世上本不存在“绝对安全”的东西。对任何一个秘密来说，都存在泄密的可能。分析专家必须根据实际情况判断出泄密的后果有多严重，以及泄密的可能性有多大。通常，一种加密方法的“强壮程度”是由其计算的复杂程度来决定的。例如，假设某种特定的加密系统复杂程度是 2^{32}，我们便认为破解它需要进行 2^{32} 次独立的运算，这个数量从表面上看似乎非常大，但对一台高速计算机来说，它每秒钟也许能执行数千乃至上万次这样的解密运算，所以，对这种加密系统来说，其能力尚不足以保证秘密的安全。正是考虑到这样的情况，所以我们一般用“计算安全”来量度一个加密系统的安全程度。

第二节　信息加密的实现原理

密钥是为了有效控制加密、解密算法的实现而设置的，在这些算法的实现过程中，

需要有某些只被通信双方所掌握的专门的、关键的信息参与，这些信息就称为密钥。加密在许多场合集中表现为对密钥的应用，因此密钥往往是保密与窃密的主要对象。

在现代计算机网络中一般采取两种加密形式：对称密钥（又称单密钥、私钥）体系和非对称密钥（又称公开密钥、公钥）体系。采用何种加密算法要结合具体的环境和系统而定，而不能简单地根据加密强度来做出判断和选择。因为除了加密算法本身之外，密钥的合理分配、加密效率、与现有系统的结合性，以及投入产出分析等都应在实际应用环境中具体考虑。

就公开密钥加密体系而言，关键部分是建立在“单向函数和活门”的基础之上的。所谓“单向函数”是指一个函数很容易朝一个方向计算，但很难（甚至不可能）逆向回溯。所谓“活门”是指一种可供回溯的“小道”。

为使单向函数能有效地应用于加密系统，它必须有能力对任何输入都进行这样的单向计算。例如，在一个有限的范围内，很容易计算出数字的乘积，但却很难分解出生成那个乘积的各个乘数因子。另一个实际应用的例子是离散对数问题：一个大质数 p，以及一个底数 g，对于已知的一个特定值 y，求指数 x，如下所示。

$$g^x=y \bmod p$$

模指数很容易便可计算出来，但假若想通过一次离散对数运算恢复原来的指数，却是异常艰难的。对于每一类数字，如奇数、回文数字、可用 47 除尽的数字，其离散对数问题（Discrete Logarithm Problem）如何解决至今仍然非常困难。类似的还有 n 级多项 mod p 计算问题、乘法因子分解问题和背包问题（Knapsack Problem）等。至今还没有找出一个真正的单向函数，但某些函数拥有单向函数的一些属性，所以通常将它们也称为“单向函数”。

在现代加密技术中，一般将单向散列函数应用于身份验证及完整性校验。单向散列函数不同于单向函数。散列函数采用一条长度可变的消息作为输入，对其进行压缩，再产生一个长度固定的摘要信息，一致的输入会产生一致的输出。由于对任何长度的输入来说，输出信息的长度是固定的，所以显而易见，对一种散列算法 H 来说，可能存在两个不同的输入，如 X 和 Y，但它们的摘要信息 H(X) 和 H(Y) 却相同，这样便会产生冲突。单向散列函数的设计宗旨便是尽可能地降低这种冲突的发生。

当今流行的散列函数是 MD5（Message Digest 5，消息摘要 5）、SHA（Secure Hash Algorithm，安全散列算法）和 RIPE MD。尽管它们生成的摘要长度不同，运算速度不同，抗冲突特性也不同，但都是目前所广泛采用的。

另外一种经常用到的技术是简单的“异或”（XOR）函数。它既不是单向函数，也不是活门函数，但同样是构建加密系统一种有用的工具。有基本数学知识的人都知道，两个 0 进行 XOR 运算的结果是 0，两个 1 进行 XOR 运算还是 0，而一个 0 和一个 1 的 XOR 运算结果是 1。XOR 运算一个非常重要的特点就是它的交替性，取得任何数据后，用长度固定的一个 Key 值对其执行 XOR 运算，得到结果后，再用同样的 Key 值对结果执行 XOR 运算，便能恢复为原来的数据。这其实就是一种非常简化的“加密”算法，但要注意只要知道了一组输入或输出对数据，就马上能推断出密钥。

数据的机密性是由加密算法提供的。算法将一条正常的消息（明文）转换成乱码（密

文），再将乱码转换回正常的消息，实现加密（编码）和解密（译码）的过程。有些加密算法是对称的，即用来加密的可同样用来解密，而另一些算法是不对称的。不对称算法虽然有两个独立的函数（一个用于加密，另一个用于解密），但人们并不将其看作两个算法，而是当作单独的一种算法。因此，无论一种特定算法的“对称性”如何，加密算法都是可以交替（双向）使用的，即

明文=解密函数（加密函数（明文））

目前实用的加密技术使用若干种不同的算法。但基本的只有两种：一种是使用密钥，另一种是使用算法（本身不依赖于密钥）。

不使用密钥的加密技术十分简单，通过替换或编码以达到加密目的。例如，可以通过给每一个字母的 ASCII 值加上一个数来加密一组英文信息，如图 1.2 所示。这种算法实际上并不那么安全，它们很容易被破译，一旦知道了加密算法，就能够破译加密过的信息。

```
This is an encrypted message
Zmny ny fs jshxduzji rjyyflj
```

图 1.2　依赖变换算法的信息加密

一些更安全的加密算法是将数据与一种密钥配合使用。两种主要的加密算法是对称密钥加密和公开密钥加密。在以后的章节中将会具体讨论。

就对称密钥加密算法而言，算法中只存在一个密钥。同一个密钥被用于加密、解密过程。为了保证安全，必须保护好这个密钥而且确保只有一个人知道。私有密钥加密的另一个特点就是其使用的密钥长度一般都比较短，这使得它的算法实现比非对称加密要快，也要容易一些。对称密钥的一个主要缺陷是需要将密钥分配给每个需要的人，这样密钥分配和管理本身就是一个大问题。另外，如果暴露或损坏密钥，那么就等于暴露或损坏了用它加密过的信息，因此，有必要经常更改密钥。如果只有对称密钥方案，建议将其与数字签名一同使用，因为这样会更加有效也更加安全。

第三节　关于对称加密算法

一、对称加密的基本原理

对称加密算法一般以“块”或“流”的方式对输入信息进行处理。块加密算法的常用算法包括 DES、3DES、CAST 和 Blowfish 等，它们一般每次对一个数据块进行处理。至于块的大小，则取决于算法本身（目前多数使用系统均采用 64 位的块长度），对一个块的处理称为加密算法的“处理单位”。另一方面，流加密算法每次处理的是数据的一个位（或者一个字节），用一个键值适当地进行种子化处理，便能生成一个位（这里的“位”指二进制的位）流。

无论是块加密还是流加密，它们都适用于批量信息的加密处理。块加密算法可采用不同的模式工作，一种模式是每次都用同一个密钥；另一种模式是将上一次操作的结果“喂”给当前操作，从而将数据块连接到一起。综合运用这些模式，便可使一种加密算法变得更为“健壮”，对特定的攻击产生更强的免疫力。例如，块加密算法的基本应用就是

“电子密码本（Electronic Code Book，ECB）”模式。每个明文块都加密成一个密文块，由于使用相同的密钥，相同的明文块会加密成相同的密文块，所以对一段已知的明文来说，完全能构建出一个密码本，其中包含所有的密文组合。如果我们知道一个 IP 数据包已进行了加密处理，那么由于密文的头 20 个字节代表的是 IP 头，因此可利用一个密码本推断出真实的密钥。

在块加密算法的具体应用中，由于不能保证输入数据的长度正好为一个密码块长度的整数倍，所以根据具体的模式，需要对输入进行适当的填充。假如块的长度是 64 位，而最后一个输入块的大小仅为 48 位，那么就有必要增添 16 位的填充数据，然后才能执行加密（或解密）运算。

加密块链接（CBC）模式可取得前一个密文块，并在对下一个明文块进行加密之前，先对两者执行一次 XOR 运算，如图 1.3 所示。假如是第一个块，那么与它进行 XOR 运算的是一个初始化矢量（Initialization Vector，IV）。IV 必须具有“健壮”的伪随机特性，以确保完全一致的明文不会产生完全一致的密文。解密过程与加密相反：每个块都会进行解密，并在对前一个块进行解密之前，对两者进行一次 XOR 运算。解密到第一个块时，它同样会与 IV 进行 XOR 运算。目前使用的所有加密算法都属于块加密算法，采用 CBC 模式运行。

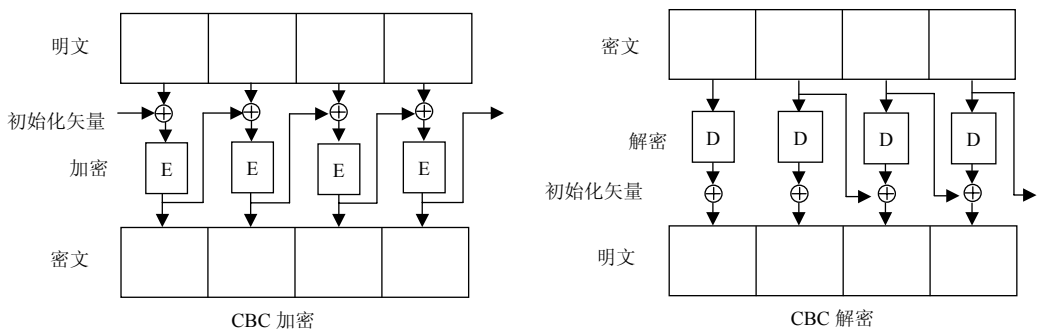

图 1.3　加密块链接方式

其他流行的模式包括加密回馈模式（Cipher Feedback Mode，CFB）和输出回馈模式（Output Feedback Mode，OFB），前者的前一个密文块会被加密，并与当前的明文块进行 XOR 运算（第一个明文块只与 IV 进行 XOR 运算）；后者会维持一种加密状态，不断地加密，并与明文块进行 XOR 运算，以生成密文（IV 代表初始的加密状态）。

二、DES 算法实现

数据加密标准（Data Encryption Standard，DES）是使用最为普遍的对称密钥算法。DES 算法于 1975 年由 IBM 发明并公开发表，并于 1976 年批准成为美国政府标准。DES 算法在 POS、ATM、磁卡及智能卡（IC 卡）、加油站、高速公路收费站等领域被广泛应用，以此来实现关键数据的保密，如信用卡持卡人的 PIN 的加密传输，IC 卡与 POS 间的双向认证、金融交易数据包的 MAC 校验等，均用到 DES 算法。

DES 算法的处理速度比较快。根据 RSA 实验室提供的数据，当 DES 完全由软件实现时，它至少比 RSA 算法快 100 倍。如果由硬件实现，DES 比 RSA 快 1000 甚至 10000 倍。因为 DES 使用 S 盒（或称选择盒，是一组高度非线性函数。在 DES 中 S 盒像一组表，是 DES 真正执行加密，解密运算的函数部分）运算，只使用简单的表查找功能，而 RSA 却则建立在非常大的整数运算上。

DES 使用相同的加密、解密算法，密钥是任意一个 64 位的自然数。算法的工作方式决定了只有 56 位有效（8 位用作校验）。NIST 授权 DES 成为美国政府的加密标准，但只适用于加密“绝密级以下信息”，尽管 DES 被认为十分安全，但确实存在方法可以攻破它。

通过穷尽搜索密钥空间，提供总共 2^{56}（大约 7.2×10^{16}）个可能的密钥。如果每秒能检测一百万个密钥，则需 2000 年。但有一组 Internet 用户，花费了 4 个多月时间分工合作解决了 RSA DES 挑战并最终攻破了这一算法。

该小组在检验了大约 18×10^{15} 个密钥后找到了正确的密钥，并恢复了如下明文。

strong cryptography makes the world a safer place.

该小组采用“强行攻击（Brute-Force）”的技术，即所有参加这一挑战的计算机搜索所有可能的密钥，一共有超过 72057594037927936 个密钥。当把这一正确密钥报告给 RSA Data Security 公司时，该小组已经搜索了大约所有可能密钥的 25%。强行攻击是破译 DES 密码的通用方法，通过不同的加密分析，可以将密钥数量降至 2^{47} 个，但这仍是一个很大的工程。如果 DES 使用长度超过 56 位的密钥，那么破译它的可能性几乎为零。

下面我们详细分析一下 DES 的处理过程。DES 数据加密算法的基本流程如图 1.4 所示。该算法输入的是 64 位的明文，在 64 位的密钥控制下，通过初始换位 IP 变成 T_0=IP(T)，再对 T_0 经过 16 层的加密变换，最后再通过逆初始变换得到 64 位的密文。密文的每一位都由明文的每一位和密钥的每一位联合确定。DES 的加密过程可分为加密处理、加密变换和子密钥生成几个部分。下面分别进行分析。

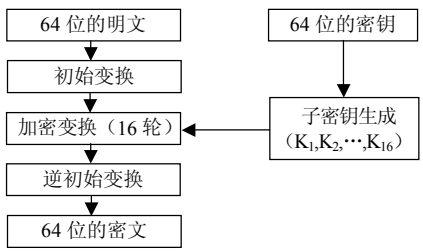

图 1.4　DES 数据加密基本流程

1. 加密处理过程

（1）初始变换

加密处理首先要对 64 位的明文按表 1.1 所示的初始换位表 IP 进行变换。表中的数值表示输入位被置换后的新位的位置。例如，输入的第 58 位，在输出时被置换到第 1 位；输入的第 7 位，在输出时被置换到第 64 位。

表 1.1　初始换位表 IP

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8

57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

（2）加密处理

上述换位处理的输出，中间要经过 16 层复杂的加密变换。初始换位的 64 位的输出成为下一步的输入，此 64 位分成左、右两个 32 位，分别记为 L_0 和 R_0，从 L_0、R_0 到 L_{16}、R_{16} 共进行 16 轮加密变换。换完之后，若经过第 n 轮处理后的左右 32 位分别为 L_n 和 R_n，则 L_n 和 R_n 可做如下的定义。

$$L_n=R_n-1$$

$$R_n=L_{n-1}\oplus f（R_{n-1},K_n）$$

这里，K_n 是向第 n 轮输入的 48 位的子密钥；L_{n-1} 和 R_{n-1} 分别是第 n−1 轮加密的输出；f 是 Mangler 函数。过程如图 1.5 和图 1.6 所示。

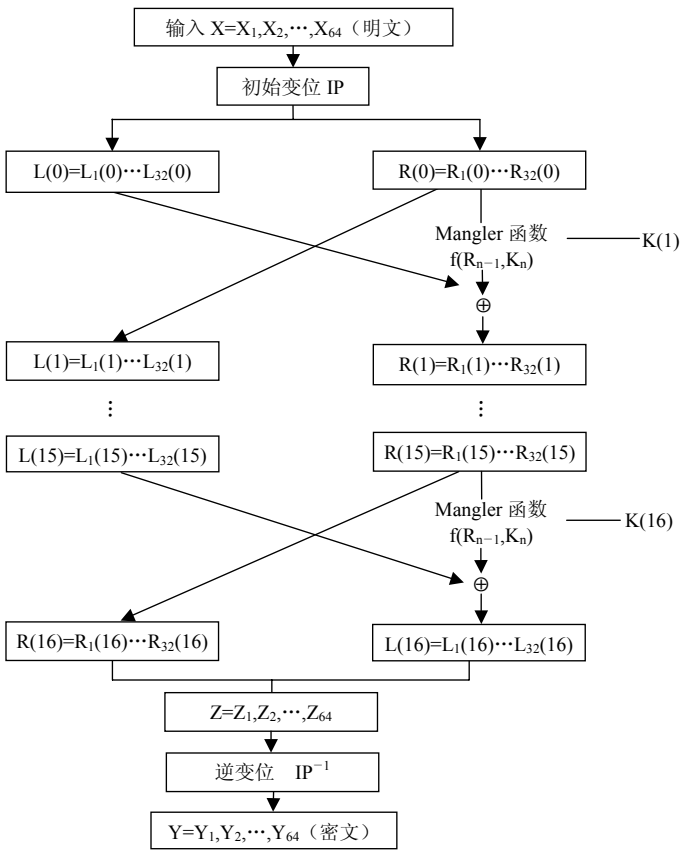

图 1.5　RES 算法框图（数据部分）

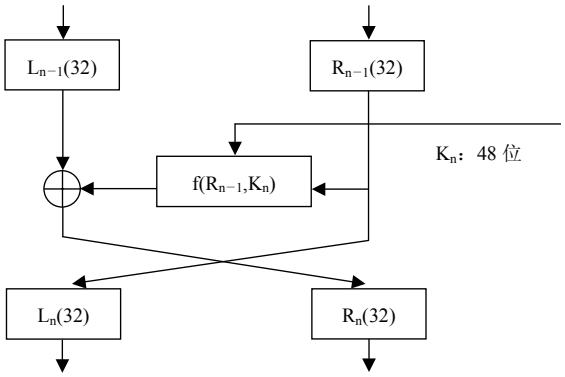

图 1.6　第 n 轮的加密变换

（3）最后换位

进行 16 轮的加密变换之后，将 L_{16} 和 R_{16} 合成 64 位的数据，再按表 1.2 所示的最后换位表进行 IP^{-1} 的换位，得到 64 位的密文，这就是 DES 加密的结果。

表 1.2　最后换位表 IP^{-1}

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

2. 加密变换

计算 f(R, K) 的方式如图 1.7 所示。在 DES 算法中，其他部分都是线性的，而 f(R, K) 变换是非线性的，因此可以产生强度很高的密码。

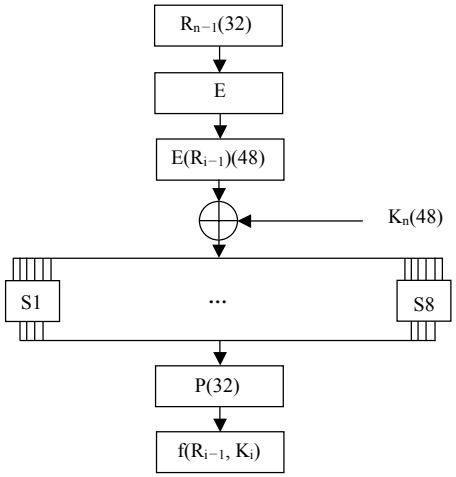

图 1.7　f(R, K)函数的计算

32 位的 R 先按表 1.3 所示的扩展换位表 E 进行扩展换位处理，得到 48 位的 R'。将这 48 位的 R'和 48 位的密钥 K 进行异或运算，并分成 6 位的 8 个分组，输入 S1～S8 的 8 个 S 盒中，S1～S8 称为选择函数，这些 S 盒输入 6 位，输出 4 位。S 盒如表 1.4 所示。

表 1.3　扩展换位表 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

表 1.4　S 盒替换表

	行＼列	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S1	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
S2	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	3	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
S3	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
S4	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
S5	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
S6	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	11	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
S7	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
S8	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	7	11	4	1	9	12	14	2	0	6	10	13	15	3	6	8
	3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

一个 S 盒中具有四种替换表（用行号 0、1、2、3 表示），究竟采用哪一行，要通过输入的六位的开头和末尾两位选定，然后按选定的替换表将输入的六位的中间四位进行代替，下面举例说明。当向 S1 输入“011011”时，因开头和结尾的组合是“01”，所以选中编号为“1”的替代表；又根据中间四位“1101”，选定第 13 列，查表第 1 行第 13 列所指的值为 5，即输出为“0101”，这四位就是经过替代后的值。按此进行，输出 32 位。再用表 1.5 所示的单纯换位表 P 进行变换，这样就完成了 f(R, K)的变换。

表 1.5　单纯换位表 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

3. 子密钥的生成

下面说明子密钥 K_1～K_{16} 的 16 个子密钥的生成（Mangler 函数）过程，在 64 位的密钥中包含了 8 位的奇偶校验位，所以密钥的实际长度为 56 位，而每轮要生成 48 位的子密钥。

输入的 64 位密钥，首先通过压缩换位（PC-1）去掉校验位，输出 56 位的密钥，每层分成两部分，上部分 28 位为 C_0，下部分为 D_0。C_0 和 D_0 依次进行循环左移操作生成了 C_1 和 D_1，将 C_1 和 D_1 合成为 56 位，再通过压缩换位（PC-2）输出 48 位的子密钥 K_1，再将 C_1 和 D_1 进行循环左移操作和 PC-2 压缩换位，得到子密钥 K_2……，以此类推，就可以得到 16 个子密钥。密钥压缩换位如表 1.6 所示。要注意的是，在产生子密钥的过程中，L_1、L_2、L_9、L_{16} 是循环左移 1 位，其余都左移 2 位，左移次数如表 1.7 所示。

表 1.6　密钥压缩换位

压缩换位 PC-1							压缩换位 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	42	35	27	23	19	12	4	26	8
19	11	3	60	52	44	36	16	7	27	20	13	2
63	55	46	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