第 3 章　信息加密技术

信息保密技术是利用数学或物理手段，对信息的传输和存储进行保护以防止泄露的技术。信息保密技术主要包括信息加密技术和信息隐藏技术。本章介绍信息加密技术，第 4 章将介绍信息隐藏技术。信息加密是指使有用的信息变为看上去似为无用的乱码，使攻击者无法读懂信息的内容从而保护信息。信息加密技术是保障信息安全的最基本、最核心的技术措施，是现代密码学的主要组成部分。

3.1　密码学的发展历程

人类早在远古时期就有了相互隐瞒信息的想法，自从有了文字来表达人们的思想开始，人类就懂得了如何用文字与他人分享信息，以及用文字秘密传递信息的方法，这就催生了信息保密科学的诞生和发展。密码学的发展可以追溯到 4000 年前，其发展历史比较悠久。密码学的发展大致经历了手工加密、机械加密和计算机加密 3 个阶段。

1. 手工加密阶段

早在公元前 1900 年左右，一位埃及书吏就在碑文中使用了非标准的象形文字。据推测，这些“秘密书写”是为了给墓主的生活增加神秘气氛，从而提高他们的声望。这可能是最早有关密码的记载了。

公元前 1500 年左右，美索不达米亚人在一块板上记录了被加密的陶器上釉规则。

公元前 600～前 500 年，希伯来人设计了 3 种不同的加密方法，它们都以替换为基本原理，一个字母表的字母与另一个字母表的字母配对，通过用相配对的字母替换明文的每个字母，从而生成密文。

公元前 500 年左右，古希腊斯巴达出现了原始的密码器，其方法是用一条带子缠绕在一根木棍（手杖）上，沿手杖纵轴方向写上文字，解下来的带子上便是些杂乱无章的符号，称为斯巴达手杖，如图 3-1 所示。解密者只需找到相同的手杖，再把带子绑上去，沿手杖纵方向即可读出原文。

图 3-1　斯巴达手杖

根据《论要塞的防护》（希腊人 Aeneas Tacticus 著）一书记载，公元前 2 世纪，希腊人 Polybius 设计了一种表格，使用了将字母编码成符号的方法，人们将该表称为 Polybius 校验表，如表 3-1 所示。将每个字母表示成两位数，其中第一个数字表示字母所在的行数，第二个数字表示字母所在的列数，如字母 A 对应“11”，字母 B 对应“12”，字母 C 对应“13”等。明文“enemy”被表示成一串数字，即 1533153254。

表 3-1　Polybius 校验表

列	行				
	1	**2**	**3**	**4**	**5**
1	A	B	C	D	E
2	F	G	H	I/J	K
3	L	M	N	O	P
4	Q	R	S	T	U
5	V	W	X	Y	Z

公元前 100 年左右，著名的恺撒(Caesar)密码被应用于战争中，它是最简单的一种加密办法，即用单字母来代替明文中的字母。

公元 800 年左右，阿拉伯密码学家阿尔·金迪提出解密的频率分析方法，即通过分析计算密文中字母出现的频率来破译密码。

公元 16 世纪中期，意大利数学家卡尔达诺(Cardano)发明了卡尔达诺漏板，将其覆盖在密文上，可从漏板中读出明文，这是较早的一种分置式密码。

我国很早就出现了藏头诗、藏尾诗、漏格诗及绘画等，人们将要表达的真正意思隐藏在诗文或画卷中，一般人只注意诗或画自身表达的意境，而不会去注意或很难发现隐藏在其中的“诗外之音”。

古典密码的加密方法一般是采用文字置换，主要使用手工方式实现，因此称这一时期为密码学发展的手工加密阶段。

2. 机械加密阶段

20 世纪 20 年代，随着机械和机电技术的成熟，以及电报和无线电技术的出现，引起了密码设备的一场革命——转轮密码机的发明。转轮密码机的出现是密码学的重要标志之一。通过硬件卷绕可实现从转轮密码机的一边到另一边的单字母代替，将多个这样的转轮密码机连接起来，便可实现几乎任何复杂度的多个字母代替。随着转轮密码机的出现，传统密码学有了很大的进展，利用机械转轮密码机可以开发出极其复杂的加密系统。

1921 年以后的几十年里，Hebern 构造了一系列稳步改进的转轮密码机，并将其投入到美国海军的试用评估中，并申请了美国转轮密码机的专利。这种装置在随后的近 50 年中被指定为美军的主要密码设备。

在 Hebern 发明转轮密码机的同时，欧洲的工程师(如荷兰的 Hugo Koch、德国的 Arthur Scherbius)独立地提出了转轮密码机的概念。Arthur Scherbius 于 1919 年设计了历史上著名的转轮密码机——德国的 Enigma 机。在第二次世界大战期间，Enigma 机曾作为德国海、陆、空三军中最高级的密码机。英国军队从 1942 年 2～12 月都没能解出德国潜艇发出的信号。因此，随后英国发明并使用了德国 Enigma 机的改进型密码机，它在英国军队通信中被广泛使用，并帮助他们破译了德国军队的信号。转轮密码机的使用大大提高了密码加密速度，但由于密钥量有限，在第二次世界大战中后期，它引出了一场关于加密与破译的对抗。第二次世界大战期间，波兰人和英国人破译了 Enigma 密码，美国密码分析者破译了日本的 RED、ORANGE 和 PURPLE 密码，这对盟军获胜起到了关键的作用，是密码分析史上最伟大的成功。

3. 计算机加密阶段

计算机科学的发展刺激和推动了密码学进入计算机加密阶段。一方面,电子计算机成为破译密码的有力武器;另一方面,计算机和电子学给密码的设计带来了前所未有的自由,利用计算机可以轻易地摆脱原先用铅笔和纸进行手工设计时易犯的错误,也不用面对机械式转轮机实现方式的高额费用。利用计算机还可以设计出更为复杂的密码系统。

在1949年以前出现的密码技术还算不上真正的科学,那时的密码专家常常是凭借直觉进行密码设计和分析的。1949年,Shannon发表了《保密系统的通信理论》,为密码学的发展奠定了理论基础,使密码学成为一门真正的科学。1949—1975年,密码学主要研究单钥密码体制,且发展比较缓慢。1976年,Diffie和Hellman发表了《密码学的新方向》一文,提出了一种新的密码设计思想,从而开创了公钥密码学的新纪元。1977年,美国国家技术标准局(NIST)正式公布了数据加密标准(Data Encryption Standard,DES),将DES算法公开,揭开了密码学的神秘面纱,大大推动了密码学理论的发展和技术应用。

十多年来,由于现实生活的实际需要及计算技术的发展,密码学的每一个研究领域都出现了许多新的课题。例如,在分组密码领域,以往人们认为安全的DES算法,在新的分析法及计算技术面前已被证明不再安全了。于是,美国于1997年1月开始征集新一代数据加密标准,即高级数据加密标准(Advanced Encryption Standard,AES)。目前,AES征集活动已经选择了比利时密码学家设计的Rijndael算法作为新一代数据加密标准,且该征集活动在密码界又掀起了一次分组密码研究的高潮。同时,在公钥密码领域,椭圆曲线密码体制由于具有安全性高、计算速度快等优点而引起了人们的普遍关注,一些新的公钥密码体制(如基于格的公钥体制NTRU、基于身份的和无证书的公钥密码体制)相继被提出。在数字签名方面,各种有不同实际应用背景的签名方案(如盲签名、群签名、环签名、指定验证人签名、聚合签名等)不断出现。在应用方面,各种有实用价值的密码体制的快速实现受到了专家的高度重视,许多密码标准、应用软件和产品被开发和应用。一些国家(如美国、中国等)已经颁布了数字签名法,使数字签名在电子商务和电子政务等领域得到了法律的认可。随着其他技术的发展,一些具有潜在密码应用价值的技术也得到了密码学家的重视,出现了一些新的密码技术,如混沌密码、量子密码、DNA密码等。现在,密码学的研究和应用已大规模地扩展到了民用方面。

3.2 密码学中的基本术语

密码学的英文为Cryptography,该词来源于古希腊语的Kryptos和Graphein,希腊语的原意是密写术,即将易懂的信息(如文字)通过一些变换转换为难以理解的信息(如令人费解的符号)。密码学研究进行保密通信和如何实现信息保密的问题,具体指通信保密传输和信息存储加密等。它以认识密码变换的本质、研究密码保密与破译的基本规律为对象,主要以可靠的数学方法和理论为基础,对解决信息安全中的机密性、数据完整性、认证和身份识别,对信息的可控性及不可抵赖性等问题提供系统的理论、方法和技术。密码学包括两个分支:密码编码学和密码分析学。密码编码学研究对信息进行编码,实现对信息的隐藏;密码分析学研究加密消息的破译或消息的伪造。下面是密码学中一些常用的术语。

(1) 明文(plaintext/message)：指待加密的信息，用 P 或 M 表示。明文可以是文本文件、图形、数字化存储的语音流或数字化视频图像的比特流等。

(2) 密文(cipertext)：指明文经过加密处理后的形式，用 C 表示。

(3) 加密(encryption)：指用某种方法伪装消息以隐藏它的内容的过程。

(4) 加密算法(encryption algorithm)：指将明文变换为密文的变换函数，通常用 E 表示。

(5) 解密(decryption)：指把密文转换为明文的过程。

(6) 解密算法(decryption algorithm)：指将密文变换为明文的变换函数，通常用 D 表示。

(7) 密钥(key)：变换函数所用的一个控制参数。加密和解密算法的操作通常是在一组密钥控制下进行的，分别称为加密密钥和解密密钥，通常用 K 表示。

(8) 密码分析(cryptanalysis)：指截获密文者试图通过分析截获的密文从而推断出原来的明文或密钥的过程。

(9) 被动攻击(passive attack)：指对一个保密系统采取截获密文并对其进行分析和攻击。这种攻击对密文没有破坏作用。

(10) 主动攻击(active attack)：指攻击者非法侵入一个密码系统，采用伪造、修改、删除等手段向系统注入假消息进行欺骗。这种攻击对密文具有破坏作用。

(11) 密码系统(cryptosystem)：指用于加密和解密的系统。加密时，系统输入明文和加密密钥，加密变换后，输出密文；解密时，系统输入密文和解密密钥，解密变换后，输出明文。在基于密码的保密系统中，为了便于研究其一般规律，通常将密码系统抽象为一般模型，如图 3-2 所示。

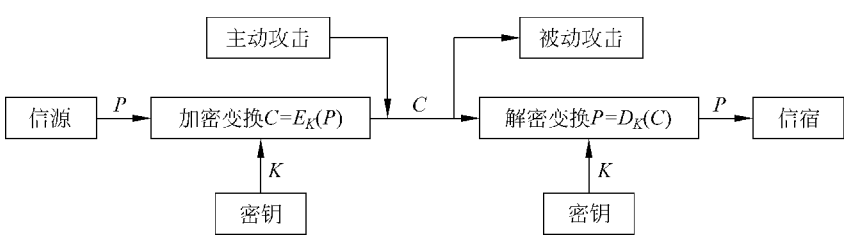

图 3-2　密码系统模型

(12) 密码体制：密码系统采用的基本工作方式称为密码体制。密码体制的要素是密码算法和密钥。根据密钥的使用方式和密码算法的加密方式可以对密码系统进行不同的分类。

柯克霍夫(Kerckhoffs)原则：密码系统的安全性取决于密钥，而不是密码算法，即密码算法要公开。柯克霍夫原则是荷兰密码学家 Kerckhoffs 于 1883 年在名著《军事密码学》中提出的基本假设。遵循这个假设的好处是，它是评估算法安全性唯一可用的方式。因为如果密码算法保密，密码算法的安全强度就无法进行评估；防止算法设计者在算法中隐藏后门。因为算法被公开后，密码学家可以研究、分析其是否存在漏洞，同时也接受攻击者的检验，有助于推广使用。当前网络应用十分普及，密码算法的应用不再局限于传统的军事领域，只有公开使用，密码算法才可能被大多数人接受并使用。同时，对用户而言，只需掌握密钥就可以使用了，非常方便。

3.3 古典密码体制

古典密码时期一般认为是从古代到19世纪末,这个时期生产力水平低,加密、解密方法主要以纸、笔或简单的器械来实现,在这个时期提出和使用的密码称为古典密码。古典密码是密码学发展的初级阶段。尽管古典密码大都较简单,但由于其安全性差,目前应用很少。研究古典密码的原理,有助于理解、构造和分析近代密码。替代(substitution)和置换(permutation)是古典密码中用到的两种基本处理技巧,它们在现代密码学中也得到了广泛使用。

3.3.1 替代密码

替代密码(substitution cipher)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。古典密码学中采用替代运算的典型密码算法有单表密码、多表密码等。

1. 单表密码

单表密码全称为单表替代密码。单表替代密码是对明文中的所有字母都使用同一个映射,即

$$\forall p \in P, \quad \text{有 } E_K: P \to C, \quad E_K(p) = c$$

为了保证加密的可逆性,一般要求映射 E 是一一映射。单表替代密码最典型的例子就是著名的恺撒密码,一般意义上的单表替代也称移位密码、乘法密码、仿射密码、使用密钥词(组)的单表替代和随机替代等。下面通过恺撒密码和使用密钥词(组)的单表替代为例进行介绍。

(1) 恺撒密码。恺撒密码是把字母表中的每个字母用该字母后面第3个字母进行替代,如表3-2所示。为便于区分,下面用小写字母表示明文,大写字母表示密文。

表3-2 恺撒密码

明文	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	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3-1】 明文:this is a book。

密文:WKLV LV D ERRN。

明文和密文空间是26个字母的循环,所以z后面的字母是a。如果为每个字母分配一个数值(a=0,b=1,…,z=25),则该算法能够表示为:

$$C = E_K(p) = (p + 3) \pmod{26}$$

其中 C 代表密文,p 代表明文。

(2) 使用密钥词(组)的单表替代。这种密码选用一个英文短语或单词串作为密钥,去掉其中重复的字母得到一个无重复字母的字母串,然后再将字母表中的其他字母依次写于此字母串之后,就构造出一个字母替代表。这种单表替代泄露给破译者的信息更少,而且密

钥可以随时更改，增加了灵活性。

【例 3-2】 设密钥为 time。密码表如表 3-3 所示。

表 3-3　密钥为 time 的密码表

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

因此，如果明文为“code”，则对应的密文为“MNEA”。

【例 3-3】 设密钥为 timeisup。密码表如表 3-4 所示。

表 3-4　密钥为 timeisup 的密码表

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

因此，如果明文为“code”，则对应的密文为“MHEI”。

单表替代密码的密钥量很小，不能抵抗穷尽搜索攻击，而且很容易受到统计分析的攻击。因为如果密码分析者知道明文的某些性质（如非压缩的英文），则分析者就能够利用该语言的规律性进行分析，从这一点意义上讲，汉语在加密方面的特性要优于英语，因为汉语常用字有 3000 多个，而英语只有 26 个字母。

2. 多表密码

单表替代密码的明文中单字母出现频率分布与密文中的相同，为了克服这个缺点，多表替代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，其中每个映射是简单替代密码中的一对一映射（即处理明文消息时使用不同的单字母替代）。多表替代密码将明文字符划分为长度相同的消息单元，称为明文组，对不同明文组进行不同的替代，即使用了多张单字母替代表，从而使同一个字符对应不同的密文，改变了单表代替中密文与明文字母的唯一对应性，使密码分析更加困难。多字母代替的优点是很容易将字母的自然频度隐蔽或均匀化，从而可以抗击统计概率分析。Playfair 密码、Vigenere 密码、Hill 密码都是这一类型的密码。

（1）Playfair 密码。Playfair 密码出现于 1854 年，它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双字母组合。Playfair 密码基于一个 5×5 字母矩阵，该矩阵使用一个关键词（密钥）来构造，其构造方法是：从左至右、从上至下依次填入关键词的字母（去除重复的字母），然后再以字母表顺序依次填入其他字母。字母 I 和 J 被算为一个字母（即 J 被当作 I 处理）。

对每一对明文字母 p_1、p_2 的加密方法如下。

① 若 p_1、p_2 在同一行，则对应的密文 C_1 和 C_2 分别是紧靠 p_1、p_2 右端的字母。其中第一列被看作是最后一列的右方（解密时反向）。

② 若 p_1、p_2 在同一列，则对应的密文 C_1 和 C_2 分别是紧靠 p_1、p_2 下方的字母。其中第一行被看作是最后一行的下方（解密时反向）。

③ 若 p_1、p_2 不在同一行，也不在同一列，则 C_1 和 C_2 是由 p_1 和 p_2 确定的矩形的其他两角的字母，并且 C_1 和 p_1、C_2 和 p_2 同行（解密时处理方法相同）。

④ 若 $p_1=p_2$,则在重复字母之间插入一个字母(如 Q,需要事先约定),并用前述方法处理。

⑤ 若明文字母数为奇数,则在明文的末端添加某个事先约定的字母作为填充。

【例 3-4】 密钥是 monarchy。

解:构造的字母矩阵如表 3-5 所示。

表 3-5 字母矩阵表

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

如果明文是 P=armuhsea,先将明文分成两个字母一组:

ar mu hs ea

根据表 3-5 中的对应密文为:

RM CM BP IM(JM)

Playfair 密码与简单的单一字母替代密码相比有了很大的进步。首先,虽然仅有 26 个字母,但有 676(26×26)种双字母组合,因此识别各种双字母组合要比简单的单一字母替代密码困难得多;其次,各个字母组的频率要比单字母范围大,这使得频率分析更加困难。尽管如此,Playfair 密码还是相对容易被攻破的,因为它仍然使许多明文语言的结构保存完好。几百字的密文通常就足以用统计分析破译了。

区别 Playfair 密码和单表密码的有效方法是:计算在文本中每个字母出现的频率,并与字母 e(最为常用的字母)出现的频率相除,设 e 的相对频率为 1,则其他字母的相对频率可以得出,如 t 的相对频率为 0.67,然后画一个图线,水平轴上的点对应于以递减频率顺序排列的字母。为了归一化该图线,在密文中出现的每个字母的数量再次被 e 在明文中出现的次数相除。因此结果图线显示了由加密屏蔽的字母的频率分布程度,这使得分解替代密码十分容易。如果该频率分布信息全部隐藏在该加密过程中,频率的明文图线将是平坦的,使用单字母统计分析方法将很难破译该密码。

(2) Vigenere 密码。Vigenere 密码是 16 世纪法国著名密码学家 Blaise de Vigenere 于 1568 年发明的,它是最著名的多表替代密码的例子。Vigenere 密码使用一个词组作为密钥,密钥中每一个字母用来确定一个替代表,每一个密钥字母被用来加密一个明文字母,第一个密钥字母加密明文的第一个字母,第二个密钥字母加密明文的第二个字母,等所有密钥字母使用完后,密钥又再循环使用。

为了帮助理解该算法,需要构建一个表,如图 3-3 所示,26 个密文都是水平排列的,最左边一列为密钥字母,最上面一行为明文字母。其加解密过程如下。

加密过程:给定一个密钥字母 k 和一个明文字母 p,密文字母就是位于 k 所在行与 p 所在列交叉点上的那个字母。

解密过程:由密钥字母决定行,在该行中找到密文字母,密文字母所在列的列首对应的明文字母就是相应的明文。

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

图 3-3　Vigenere 表

假设数字 0～25 分别表示 26 个英文字母 a～z，则 Vigenere 密码亦可用下列公式表示：

加密算法：

$$c_i = p_i + k_i \pmod{26}$$

解密算法：

$$p_i = c_i - k_i \pmod{26}$$

其中，p_i、c_i、k_i 分别表示第 i 个明文、密文和密钥字母编码，密钥字母编码有 L 个。

【例 3-5】 假设英文字母表（$n=26$），密钥 k＝college，当明文 m＝a man liberal in his views 时，使用 Vigenere 密码技术后得到的密文是什么？

解：

① $p_1 = a \rightarrow 0$，$k_1 = c \rightarrow 2$

$c_1 = 0+2 \pmod{26} = 2 \rightarrow c$

② $p_2 = m \rightarrow 12$，$k_2 = o \rightarrow 14$

$c_2 = 12+14 \pmod{26} = 0 \rightarrow a$

⋮

㉑ $p_{21} = s \rightarrow 18$，$k_{21} = e \rightarrow 4$

$c_{21} = 18+4 \pmod{26} = 22 \rightarrow w$

即密文为 $c = c_1 c_2 \cdots c_{21}$＝C ALZ POFGFLW MT LKG GTICW。

(3) Hill 密码。Hill 密码是由数学家 Lester Hill 于 1929 年研制的,它也是一种多表密码,实际上它是仿射密码技术的特例。其基本加密思想将 n 个明文字母通过线性变换,将它们转换为 n 个密文字母。解密只需做一次逆变换即可。

算法的密钥 $K=\{Z_{26}$ 上的 $n\times n$ 可逆矩阵$\}$,明文 $\boldsymbol{M}$ 与密文 $\boldsymbol{C}$ 均为 n 维向量,记为:

$$\boldsymbol{M}=\begin{bmatrix}m_1\\m_2\\\vdots\\m_n\end{bmatrix},\quad \boldsymbol{C}=\begin{bmatrix}c_1\\c_2\\\vdots\\c_n\end{bmatrix},\quad \boldsymbol{K}=[k_{ij}]_{n\times n}=\begin{bmatrix}k_{11}&k_{12}&\cdots&k_{1n}\\k_{21}&k_{22}&\cdots&k_{2n}\\\vdots&\vdots&\ddots&\vdots\\k_{n1}&k_{n2}&\cdots&k_{nn}\end{bmatrix}$$

其中,

$$\begin{cases}c_1=k_{11}m_1+k_{12}m_2+\cdots+k_{1n}m_n \bmod 26\\c_2=k_{21}m_1+k_{22}m_2+\cdots+k_{2n}m_n \bmod 26\\\vdots\\c_n=k_{n1}m_1+k_{n2}m_2+\cdots+k_{nn}m_n \bmod 26\end{cases}$$

或写成 $\boldsymbol{C}=\boldsymbol{K}\cdot\boldsymbol{M}(\bmod 26)$。

解密变换则为 $\boldsymbol{M}=\boldsymbol{K}^{-1}\cdot\boldsymbol{C}(\bmod 26)$。

其中,$\boldsymbol{K}^{-1}$ 为 $\boldsymbol{K}$ 在模 26 上的逆矩阵,满足 $\boldsymbol{K}\boldsymbol{K}^{-1}=\boldsymbol{K}^{-1}\boldsymbol{K}=\boldsymbol{I}$ (mod 26),这里 $\boldsymbol{I}$ 为单位矩阵。

【例 3-6】 设明文消息为 good,试用 $n=2$,密钥 $\boldsymbol{K}=\begin{bmatrix}11&8\\3&7\end{bmatrix}$ 的 Hill 密码对其进行加密,然后再进行解密。

解:将明文划分为两组,即(g,o)和(o,d),即(6,14)和(14,3)。加密过程为:

$$\begin{bmatrix}c_1\\c_2\end{bmatrix}=K\begin{bmatrix}m_1\\m_2\end{bmatrix}=\begin{bmatrix}11&8\\3&7\end{bmatrix}\begin{bmatrix}6\\14\end{bmatrix}=\begin{bmatrix}178\\116\end{bmatrix}\equiv\begin{bmatrix}22\\12\end{bmatrix}(\bmod 26)\Rightarrow\begin{bmatrix}w\\m\end{bmatrix}$$

$$\begin{bmatrix}c_3\\c_4\end{bmatrix}=K\begin{bmatrix}m_3\\m_4\end{bmatrix}=\begin{bmatrix}11&8\\3&7\end{bmatrix}\begin{bmatrix}14\\3\end{bmatrix}=\begin{bmatrix}178\\63\end{bmatrix}\equiv\begin{bmatrix}22\\11\end{bmatrix}(\bmod 26)\Rightarrow\begin{bmatrix}w\\l\end{bmatrix}$$

因此,good 的加密结果为 WMWL。显然,明文不同位置的字母“o”加密成的密文字母不同。为了解密,由前面计算得 $\boldsymbol{K}^{-1}=\begin{bmatrix}7&18\\23&11\end{bmatrix}$,可由密文解密计算出明文:

$$\begin{bmatrix}m_1\\m_2\end{bmatrix}=\boldsymbol{K}^{-1}\begin{bmatrix}c_1\\c_2\end{bmatrix}=\begin{bmatrix}7&18\\23&11\end{bmatrix}\begin{bmatrix}22\\12\end{bmatrix}=\begin{bmatrix}370\\638\end{bmatrix}\equiv\begin{bmatrix}6\\14\end{bmatrix}(\bmod 26)\Rightarrow\begin{bmatrix}g\\o\end{bmatrix}$$

$$\begin{bmatrix}m_3\\m_4\end{bmatrix}=\boldsymbol{K}^{-1}\begin{bmatrix}c_3\\c_4\end{bmatrix}=\begin{bmatrix}7&18\\23&11\end{bmatrix}\begin{bmatrix}22\\11\end{bmatrix}=\begin{bmatrix}352\\627\end{bmatrix}\equiv\begin{bmatrix}14\\3\end{bmatrix}(\bmod 26)\Rightarrow\begin{bmatrix}o\\d\end{bmatrix}$$

因此,解密得到正确的明文为“good”。

Hill 密码特点如下。

(1) 可以较好地抑制自然语言的统计特性,不再有单字母替换的一一对应关系,对抗“唯密文攻击”有较高安全强度。

(2) 密钥空间较大,在忽略密钥矩阵 $\boldsymbol{K}$ 可逆限制条件下,$|K|=26^{n\times n}$。

(3) 易受已知明文攻击及选择明文攻击。

3.3.2　置换密码

置换密码(permutation cipher)加密过程中明文的字母保持相同,但顺序被打乱了,又被称为换位密码。在这里介绍一种较常见的置换处理方法:将明文按行写在一张格纸上,然后再按列的方式读出结果,即为密文;为了增加变换的复杂性,可以设定读出列的不同次序(该次序即为算法的密钥)。

【例 3-7】 明文为 cryptography is an applied science,假设密钥为 creny,用换位加密方法确定其密文。

解:根据密钥 creny 中各个字母在英文字母表中的出现次序可确定其排序为 14235(即 c 第 1 个出现,r 第 4 个出现,…,y 第 5 个出现)。将明文按照密钥的长度(5 个字符)逐行列出,如表 3-6 所示。

表 3-6　置换表

1	4	2	3	5
c	r	y	p	t
o	g	r	a	p
h	y	i	s	a
n	a	p	p	l
i	e	d	s	c
i	e	n	c	e

然后依照密钥决定的次序按列依次读出,因此,密文为 COHNII YRIPDN PASPSC RGYAEE TPALCE。

在置换密码中,明文的字母相同,但出现的顺序被打乱了,经过多步置换会进一步打乱字母顺序。但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辨别。如果将置换密码与其他密码技术结合,则可以得出十分有效的密码编码方案。

3.4　对称密码体制

对称密码体制(symmetric encryption)也称为秘密密钥密码体制、单密钥密码体制或常规密码体制,其模型如图 3-4 所示。如果一个密码算法的加密密钥和解密密钥相同,或者由其中一个很容易推导出另一个,该算法就是对称密码算法,满足关系 $M = D_K(C) = D_K(E_K(M))$。

一个攻击者(密码分析者)能基于不安全的公开信道观察密文 C,但不能接触到明文 M 或密钥 K,他可以试图恢复明文 M 或密钥 K。假定他知道加密算法 E 和解密算法 D,只对当前这个特定的消息感兴趣,则努力的焦点是通过产生一个明文的估计值 M' 来恢复明文 M。如果他也对读取未来的消息感兴趣,就需要通过产生一个密钥的估计值 K' 来恢复密钥 K,这是一个密码分析的过程。

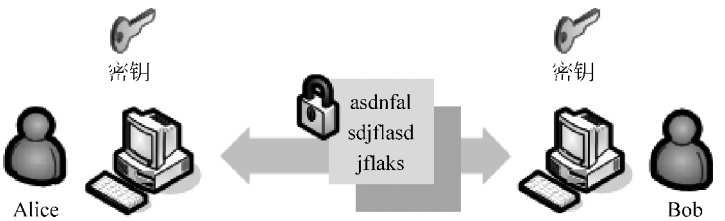

图 3-4 对称密码模型

对称密码体制的安全性主要取决于两个因素：一是加密算法必须足够安全，使得不必为算法保密，仅根据密文就能破译出消息是计算上不可行的；二是密钥的安全性，即密钥必须保密并保证有足够大的密钥空间。对称密码体制要求基于密文和加密/解密算法的知识能破译出消息的做法在计算上是不可行的。

对称密码算法的优缺点如下。

(1) 优点：加密、解密处理速度快，保密度高等。

(2) 缺点：①密钥是保密通信安全的关键，发信方必须安全、妥善地把密钥护送到收信方，不能泄露其内容。如何才能把密钥安全地送到收信方，是对称密码算法的突出问题。对称密码算法的密钥分发过程复杂，所花代价高；②多人通信时密钥组合的数量会出现爆炸性膨胀，使密钥分发更加复杂化，若有 N 个用户进行两两通信，总共需要的密钥数为 $N(N-1)/2$ 个；③通信双方必须统一密钥，才能发送保密的信息。如果发信人与收信人素不相识，这就无法向对方发送秘密信息了；④除了密钥管理与分发问题外，对称密码算法还存在数字签名困难问题(通信双方拥有同样的消息，接收方可以伪造签名，发送方也可以否认发送过某消息)。

对称密码体制分为两类：一类是对明文的单个位(或字节)进行运算的算法，称为序列密码算法，也称为流密码算法(stream cipher)；另一类是把明文信息划分成不同的块(或小组)结构，分别对每个块(或小组)进行加密和解密，称为分组密码算法(Block cipher)。

3.4.1 序列密码

序列密码是将明文划分成单个位(如数字 0 或 1)作为加密单位产生明文序列，然后将其与密钥流序列逐位进行模 2 加运算，用符号表示为⊕，其结果作为密文的方法。加密过程如图 3-5 所示。

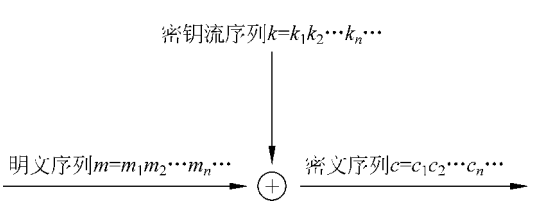

图 3-5 序列密码加密过程

加密算法：$c_i=m_i+k_i \pmod 2$。

解密算法：$m_i=c_i+k_i \pmod 2$。

【例 3-8】 设明文序列 M 是一串二进制数据 $M=(101011001111000011111111)_2$，密钥 $K=(111100001111000011110000)_2$，则

加密过程：$C=M+K \pmod 2=(010111000000000000001111)_2$。

解密过程：$M=C+K \pmod 2=(101011001111000011111111)_2$。

序列密码分为同步序列密码和自同步序列密码两种。同步序列密码要求发送方和接收方必须是同步的，在同样的位置用同样的密钥才能保证正确的解密。如果在传输过程中密

文序列有被篡改、删除、插入等错误导致同步失效，则不可能成功解密，只能通过重新同步来实现解密、恢复密文。在传输期间，一个密文位的改变只影响该位的恢复，不会对后继位产生影响。自同步序列密码密钥的产生与密钥和已产生的固定数量的密文位有关，因此，密文中产生的一个错误会影响到后面有限位的正确解密。所以，自同步密码的密码分析比同步密码的密码分析更加困难。

序列密码具有实现简单、便于硬件计算、加密与解密处理速度快、低错误(没有或只有有限位的错误)传播等优点，但同时也暴露出对错误的产生不敏感的缺点。序列密码涉及大量的理论知识，许多研究成果并没有完全公开，这也许是因为序列密码目前主要用于军事和外交等机要部门的缘故。目前，公开的序列密码主要有 RC4、SEAL 等。

序列密码的安全强度依赖于密钥流产生器所产生的密钥流序列的特性，关键是密钥生成器的设计及收发两端密钥流产生的同步技术。

1. 伪随机序列

在序列密码中，一个好的密钥流序列应该满足：具有良好的伪随机性，如极大的周期、极大的线性复杂度、序列中 0 和 1 的分布均匀；产生的算法简单；硬件实现方便。

产生密钥流序列的一种简单方法是使用自然现象随机生成，如半导体电阻器的热噪声、公共场所的噪声源等。还有一种方法是使用软件以简单的数学函数来实现，如标准 C 语言库函数中的 rand()函数，它可以产生介于 0～65 535 的任何一个整数，以此作为“种子”输入，随后再产生比特流。rand()建立在一个线性同余生成器的基础上，如 $k_n = ak_{n-1} + b(\mathrm{mod}\ m)$，$k_0$ 作为初始值，a、b 和 m 都是整数。但这只能作为以实验为目的的例子，不能满足密码学意义上的要求。

产生伪随机数的一个不错的选择是使用数论中的难题。最常用的是 BBS 伪随机序列生成器。首先产生两个大素数 p 和 q，且 $p=q=3(\mathrm{mod}\ 4)$，设 $n=pq$，并选择一个随机整数 x，x 与 n 是互素的，且设初始输入 $x_0 = x^2(\mathrm{mod}\ n)$，BBS 通过如下过程产生一个随机序列 $b_1, b_2, \cdots$。

(1) $x_j = x_{j-1}(\mathrm{mod}\ n)$。

(2) b_j 是 x_j 的最低有效比特。

例如，设 $p=24672462467892469787$ 和 $q=396736894567834589803$，则

$$n=9788476140853110794168855217413715781961$$

令 $x=873245647888478349013$，则初始输入

$$x_0 = x^2 \bmod n = 884529871047878009708991774601012286317 2$$

$x_1, x_2, \cdots, x_8$ 的值分别为：

$$x_1=711889428113132952274596245549812382240 8$$

$$x_2=314517460888889316415138015206070451822 7$$

$$x_3=489800778230715623327223318557489943035 5$$

$$x_4=393545781893511292234709354618967231038 9$$

$$x_5=675099511510097048901761303198740246040$$

$$x_6=428991482877174013354619065826651517132 6$$

$$x_7=443106671145437826089038638559381752166 8$$

$$x_8=733687612419504639741423533367500537243 6$$

取上述任意一个比特串,当 x 的值为奇数时,b 的值取1;当 x 的值为偶数时,b 的值取0,故产生的随机序列 $b_1,b_2,\cdots,b_8=0,1,1,1,0,0,0,0$。可见,产生密钥流序列的方法很多,常见的方法有线性同余法、线性反馈移位寄存器、非线性反馈移位寄存器、有限自动机和混沌密码等。

2. 线性反馈移位寄存器

通常,产生密钥流序列的硬件是反馈移位寄存器。一个反馈移位寄存器由两部分组成:移位寄存器和反馈函数,如图3-6所示。

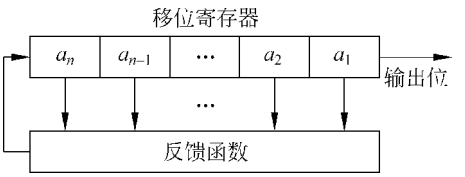

图3-6 反馈移位寄存器

移位寄存器由 n 个寄存器组成,每个寄存器只能存储一个位,在一个控制时钟周期内,根据寄存器当前的状态计算反馈函数 $f(a_1,a_2,\cdots,a_n)$作为下一时钟周期的内容,每次输出最右端一位 a_1,同时,寄存器中所有位都右移一位,最左端的位由反馈函数计算得到。$a_i(t)$表示 t 时刻第 i 个寄存器的内容,用 $a_i(t+1)$表示 $a_i(t)$下一时刻的内容,则有

移位:

$$a_i(t+1)=a_{i+1}(t),\quad i=1,2,\cdots,n-1$$

反馈:

$$a_n(t+1)=f(a_1(t),a_2(t),\cdots,a_n(t))$$

如果反馈函数 $f(a_1,a_2,\cdots,a_n)=k_1a_n\oplus k_2a_{n-1}\oplus\cdots\oplus k_na_1$,其中 $k\in\{0,1\}$,则该反馈函数是 $a_1,a_2,\cdots,a_n$ 的线性函数,对应的反馈移位寄存器称为线性反馈移位寄存器(Linear Feedback Shift Register,LFSR)。

【例3-9】 设线性反馈移位寄存器为

$$a_i(t+1)=a_{i+1}(t),\quad i=1,2,3,4$$

$$a_4(t+1)=a_1(t)\oplus a_3(t)$$

对应$(k_1,k_2,k_3,k_4)=(0,1,0,1)$,设初始状态为$(a_1,a_2,a_3,a_4)=(0,1,1,1)$,各个时刻的状态如表3-7所示。

表3-7 LFSR在不同时刻的状态

t	a_4	a_3	a_2	a_1
0	1	1	1	0
1	1	1	1	1
2	0	1	1	1
3	0	0	1	1
4	1	0	0	1
5	1	1	0	0
6	1	1	1	0

由表 3-7 可知，$t=6$ 时的状态恢复到 $t=0$ 时的状态，且往后循环。因此，该反馈移位寄存器的周期是 6，输出序列为 0111100…，表中对应 a_1 的状态。本例中，若反馈函数为 $a_4(t+1)=a_1(t)\oplus a_4(t)$，则周期达到 15，输出序列为 0110010001111010…。对于 4 级线性反馈移位寄存器而言，所有可能状态为 $2^4=16$ 种，除去全 0 状态，最大可能周期为 15。对于 n 级线性反馈移位寄存器，不可能产生全 0 状态，因此，最大可能周期为 2^n-1，而能够产生最大周期的 LFSR 是必需的，这就要求线性反馈函数符合一定的条件。关于随机序列的周期及线性复杂度的有关知识，需要读者具备一定的数学基础，本书不再展开讨论。

选择线性反馈移位寄存器作为密钥流生成器的主要原因有：适合硬件实现；能产生大的周期序列；能产生具有良好的统计特性的序列；它的结构能够应用代数方法进行很好的分析。实际应用中，通常将多个 LFSR 组合起来构造非线性反馈移位寄存器，n 级非线性反馈移位寄存器产生伪随机序列的周期最大可达 2^n，因此，研究产生最大周期序列的方法具有重要意义。

3. RC4

RC4 是由麻省理工学院的 Ron Rivest 教授在 1987 年为 RSA 公司设计的一种可变密钥长度、面向字节流的序列密码。RC4 是目前使用最广泛的序列密码之一，已应用于 Microsoft Windows、Lotus Notes 和其他应用软件中，特别是应用到 SSL 协议和无线通信方面。

RC4 算法很简单，它以一个数据表为基础，对表进行非线性变换，从而产生密码流序列。RC4 包含两个主要算法：密钥调度算法(Key-Scheduling Algorithm，KSA)和伪随机生成算法(Pseudo Random Generation Algorithm，PRGA)。

KSA 的作用是将一个随机密钥(大小为 40～256 位)变换成一个初始置换表 S。过程如下。

(1) S 表中包含 256 个元素 $S[0]$～$S[255]$，对其初始化，令 $S[i]=i$，$0\leqslant i\leqslant 255$。

(2) 用主密钥填充字符表 K，如果密钥的长度小于 256B，则依次重复填充，直至将 K 填满，$K=\{K[i],0\leqslant i\leqslant 255\}$。

(3) 令 $j=0$。

(4) 对于 i 从 0 到 255 循环：

① $j=j+S[i]+K[i]\pmod{256}$。

② 交换 $S[i]$和 $S[j]$。

PRGA 的作用是从 S 表中随机选取元素，并产生密钥流。过程如下。

(1) $i=0$，$j=0$。

(2) $i=i+1\pmod{256}$。

(3) $j=j+S[i]\pmod{256}$。

(4) 交换 $S[i]$和 $S[j]$。

(5) $t=S[i]+S[j]\pmod{256}$。

(6) 输出密钥字 $k=S[t]$。

虽然 RC4 要求主密钥 K 至少为 40 位，但为了保证安全强度，目前至少要达到 128 位。

3.4.2 分组密码

设明文消息被划分成若干固定长度的组 $m=(m_1,m_2,\cdots,m_n)$,其中 $m_i=0$ 或 $1,i=1,2,\cdots,n$,每一组的长度为 n,各组分别在密钥 $k=(k_1,k_2,\cdots,k_t)$的作用下变换成长度为 r 的密文分组 $c=(c_1,c_2,\cdots,c_r)$。分组密码的模型如图 3-7 所示。

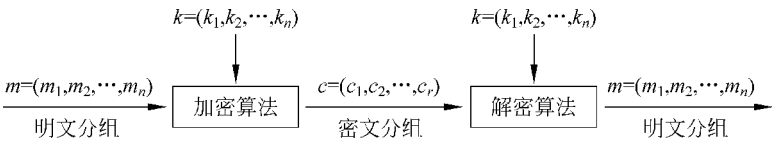

图 3-7 分组密码模型

分组密码的本质就是由密钥 $k=(k_1,k_2,\cdots,k_n)$控制的从明文空间 M(长为 n 的比特串的集合)到密文空间 C(长为 r 的比特串的集合)的一个一对一映射。为了保证密码算法的安全强度,加密变换的构造应遵循下列几个原则。

(1) 分组长度足够大。当分组长度 n 较小时,容易受到暴力穷举攻击,因此要有足够大的分组长度 n 来保证足够大的明文空间,避免给攻击者提供太多的明文统计特征信息。

(2) 密钥量空间足够大,以抵抗攻击者通过穷举密钥破译密文或获得密钥信息。

(3) 加密变换足够复杂,以加强分组密码算法自身的安全性,使攻击者无法利用简单的数学关系找到破译缺口。

(4) 加密和解密运算简单,易于实现。分组加密算法将信息分成固定长度的二进制位串进行变换。为便于软、硬件的实现,一般应选取加法、乘法、异或和移位等简单的运算,以避免使用逐比特的转换。

(5) 加密和解密的逻辑结构最好一致。如果加密、解密过程的算法逻辑部件一致,那么加密、解密可以由同一部件实现,区别在于所使用的密钥不同,以简化密码系统整体结构的复杂性。

古典密码中最基本的变换是替代和置换,其目的是产生尽可能混乱的密文。分组密码同样离不开这两种最基本的变换,替代变换就是经过复杂的变换关系将输入位进行转换,记为 S,称为 S 盒;移位变换就是将输入位的排列位置进行变换,记为 P,称为 P 盒,如图 3-8 所示。

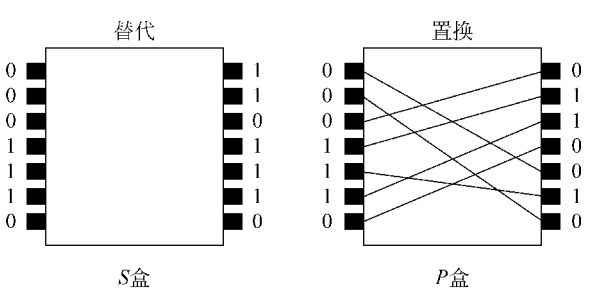

图 3-8 两种基本变换

分组密码由多重 S 盒和 P 盒组合而成。S 盒的直接作用是将输入位进行某种变换,以起到混乱的作用;P 盒的直接作用就是移动输入位的排列位置关系,以起到扩散的作用。

分组密码算法就是采用“混乱与扩散”两个主要思想进行设计的，这是 Shannon 为了有效抵抗攻击者对密码体制的统计分析提出的基本设计思想，也可以认为是分组密码算法设计的基本原理。实现分组密码算法设计的具体操作包括以下 3 个方面。

(1) 替代。替代是指将明文位用某种变换关系变换成新的位，以使所产生的密文是一堆杂乱无章的乱码，这种变换与明文和密钥密切相关，要求尽可能地使密文与明文和密钥之间的关系十分复杂，使破译者很难从中发现规律和依赖关系，从而加强隐蔽性。在分组密码算法中采用复杂的非线性替代变换就可达到比较好的混乱效果。

(2) 置换。置换是指让明文中的每一位(包括密钥的每一位)直接或间接影响输出密文中的许多位，即将每一比特明文(或密钥)的影响尽可能迅速地作用到较多的输出密文位中，以便达到隐蔽明文的统计特性。这种效果也称为“雪崩效应”，也就是说，输入即使只有很小的变化，也会导致输出位发生巨大变化。分组密码算法设计中的置换操作就是为了达到扩散的目的。

(3) 乘积变换。在分组密码算法设计中，为了增强算法的复杂度，常用的方法是采用乘积变换的思想，即加密算法不仅是简单的一次或两次基本的 S 盒和 P 盒变换，而是通过两次或两次以上 S 盒和 P 盒的反复应用，也就是迭代的思想，克服单一密码变换的弱点，构成更强的加密结果，以强化其复杂程度。后面介绍的一些分组密码算法，无一例外地都采用了这种乘积密码的思想。

3.4.3　数据加密标准(DES)

20 世纪 60 年代末，IBM 公司开始研制计算机密码算法，在 1971 年结束时提出了一种称为 Luciffer 的密码算法，它是当时最好的算法，也是最初的数据加密算法。1973 年美国国家标准局(NBS，现在的美国国家标准技术研究所，NIST)征求国家密码标准方案，IBM 就提交了这个算法。1977 年 7 月 15 日，该算法被正式采纳作为美国联邦信息处理标准生效，成为事实上的国际商用数据加密标准被使用，即数据加密标准(Data Encryption Standard，DES)。当时规定其有效期为 5 年，后经几次授权续用，真正有效期限长达 20 年。在这 20 年中，DES 算法在数据加密领域发挥了不可替代的作用。进入 20 世纪 90 年代以后，由于 DES 密钥长度偏短等缺陷，不断受到诸如差分密码分析(由以色列密码专家 Shamir 提出)和线性密码分析(由日本密码学家 Matsui 等提出)等各种攻击威胁，使其安全性受到严重的挑战，而且不断传出被破解的消息。鉴于此，美国国家保密局经多年授权评估后认为，DES 算法已没有安全性可言。于是 NIST 决定在 1998 年 12 月以后不再使用 DES 来保护官方机密，只推荐作为一般商业使用。1999 年又颁布新标准，并规定 DES 只能用于遗留密码系统，但可以使用加密的 3DES 加密算法。但不管怎样，DES 的出现推动了分组密码理论的研究，起到了促进分组密码发展的重要作用，而且它的设计思想对掌握分组密码的基本理论和工程应用有着重要的参考价值。

1. DES 算法加密过程

DES 对 64 位的明文分组进行操作。通过一个初始置换，将明文分组分成左半部分和右半部分，各 32 位长。然后进行 16 轮完全相同的运算，这些运算被称为函数 f，在运算过程中数据与密钥结合。经过 16 轮后，左、右半部分合在一起，经过一个末置换(初始置换的逆置换)，这样该算法就完成了。DES 算法的加密过程如图 3-9 所示。

DES算法的特点：①分组加密算法，以64位为分组，64位一组的明文从算法一端输入，64位密文从另一端输出；②对称算法，加密和解密用同一密钥；③有效密钥长度为56位，密钥通常表示为64位数，但每个第8位用作奇偶校验，可以忽略；④替代和置换，DES算法是两种加密技术的组合——先替代后置换；⑤易于实现，DES算法只是使用了标准的算术和逻辑运算，其作用的数最多也只有64位，因此用20世纪70年代末期的硬件技术很容易实现。

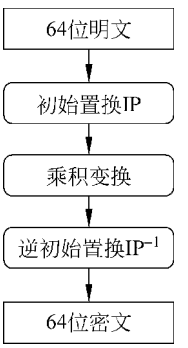

图3-9 DES算法的加密过程

DES算法的具体加密过程如下。

(1) 初始置换IP。初始置换方法是将64位明文的位置顺序打乱，表中的数字代表64位明文的输入顺序号，表中的位置代表置换后的输出顺序，表中的位置顺序是先按行后按列进行排序。例如，表中第一行第一列的数字为58，表示将原来排在第58位的比特位排在第1位；第一行第二列的数字为50，表示将原来排在第50位的比特位排在第2位，依次类推。不妨设输入位序为 $m_1 m_2 \cdots m_{64}$，初始置换后变为 $m'_1 m'_2 \cdots m'_n = m_{58} m_{50} \cdots m_7$。初始置换表中的位序特征：64位输入按8行8列进行排列，最右边一列按2、4、6、8和1、3、5、7的次序进行排列，往左边各列的位序号依次紧邻其右边一列各序号加8，如图3-10所示。

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

图3-10 初始变换IP

(2) 乘积变换(16轮迭代)。乘积变换部分要进行16轮迭代，如图3-11所示。将初始置换得到的64位结果分为两半，记为 L_0 和 R_0，各32位。设初始密钥为64位，经密钥扩展算法产生16个48位的子密钥，记为 $K_1, K_2, \cdots, K_{16}$，每轮迭代的逻辑关系为

$$\begin{cases} L_i = R_{i-1} \\ R_i = L_{i-1} \oplus f(R_{i-1}, K_i) \end{cases}$$

其中 $1 \leqslant i \leqslant 16$，函数是每轮变换的核心变换。

(3) 逆初始置换 IP^{-1}。逆初始置换 IP^{-1} 与初始置换正好相反，如图3-12所示。例如，处在第1位的比特位置换后排在第58位，处在第2位的比特位置换后排在第50位。逆初始置换后变为 $m'_1 m'_2 \cdots m'_{64} = m_{40} m_8 \cdots m_{25}$。逆初始置换表中的位序特征：64位输入依然按8行8列进行排列，1～8按列从下往上进行排列，然后是9～16排在右边一列，依次进行排4列，然后从33开始排在第一列的左边，从41开始排在第二列的左边，交叉进行。

2. 乘积变换中的 f 变换

乘积变换的核心是 f 变换，它是非线性的，是每轮实现混乱的最关键的模块，输入32位，经过扩展置换变成48位，与子密钥进行异或运算，选择 S 盒替换，将48位压缩还原成

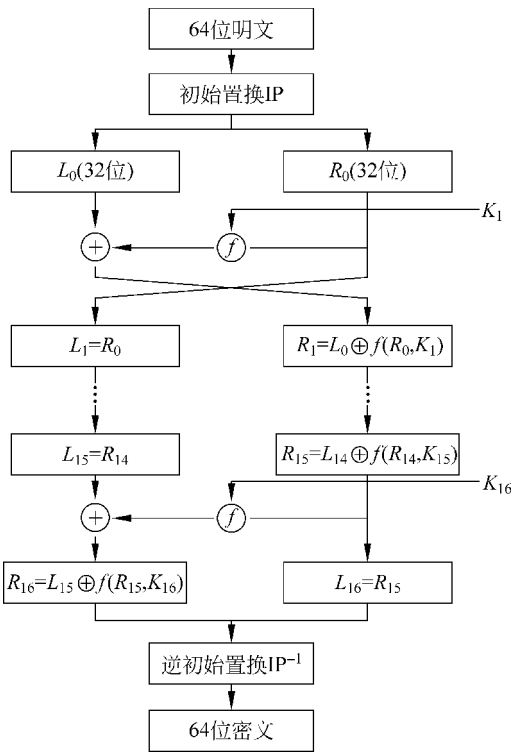

图 3-11　DES 算法的乘积变换

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

图 3-12　逆初始变换 IP^{-1}

32 位，再进行 P 盒替换，输出 32 位。如图 3-13 所示，虚线部分为 f 变换。详细的变化过程如图 3-14 所示。

(1) 扩展置换。扩展置换将 32 位扩展为 48 位，按图 3-15 所示的排列方式进行重新排列。

(2) S 盒替换。将 48 位按 6 位分为 1 组，共 8 组，也称为 8 个 S 盒，记为 $S_1S_2\cdots S_8$。每个 S 盒产生 4 位输出。8 个 S 盒的替换表如表 3-8 所示。

每个 S 盒都由 4 行×16 列组成，每行是 0～15 的一个全排列，每个数字用对应的 4 位二进制比特串表示。例如，9 用 1001 表示，7 用 0111 表示。设 6 位输入为 $a_1a_2a_3a_4a_5a_6$，将 a_1a_6 组成一个 2 位二进制数，对应 S 盒表中的行号；将 $a_2a_3a_4a_5$ 组成一个 4 位二进制数，对应 S 盒表中的列号。这样，映射到交叉点的数据就是该 S 盒的输出。

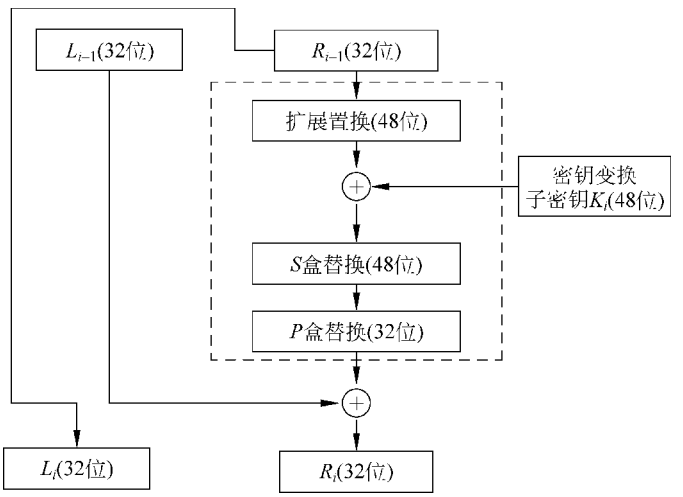

图 3-13 一轮迭代过程

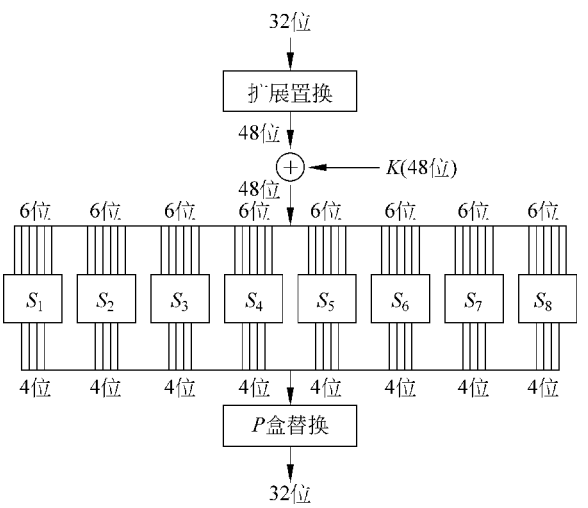

图 3-14 f 变换的计算过程

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	21	32	1

图 3-15 扩展置换

例如,已知第 2 个 S 盒的输入为 111101,则 $a_1=1$,$a_6=1$,$a_1a_6=(11)_2=3$,表明对应的行号为 3,$(a_2a_3a_4a_5)=(1110)_2=14$,表明对应列号为 14。查第 2 个 S 盒替换表,S_2 中行号为 3、列号为 14 的数据为 14,转化成二进制得到的输出为 1110。

表 3-8　*S* 盒替换表

列	行																
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7	S_1
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	7	7	3	10	5	0	
3	15	12	8	2	4	9	1	7	5	11	14	14	10	0	6	13	
0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10	S_2
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	
0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8	S_3
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	
0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15	S_4
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	
0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9	S_5
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	
0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11	S_6
1	19	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	
0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1	S_7
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	
0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7	S_8
1	1	5	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	5	8	
3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11	

（3）*P* 盒替换。*P* 盒替换就是将 *S* 盒替换后的 32 位作为输入，按图 3-16 所示的顺序重新排列，得到的 32 位结果即为 f 函数的输出 $f(R_{i-1},K_i)$。

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

图 3-16　*P* 盒替换

3. 子密钥的生成

初始密钥长度为 64 位，但每个第 8 位是奇偶校验位，分布在第 8、16、24、32、40、48、56 和 64 位的位置上，目的是用来检错，实际的初始密钥长度为 56 位。在 DES 算法中，每一轮迭代需要使用一个子密钥，子密钥是从用户输

入的初始密钥中产生的。图3-17所示为各轮子密钥的产生流程。

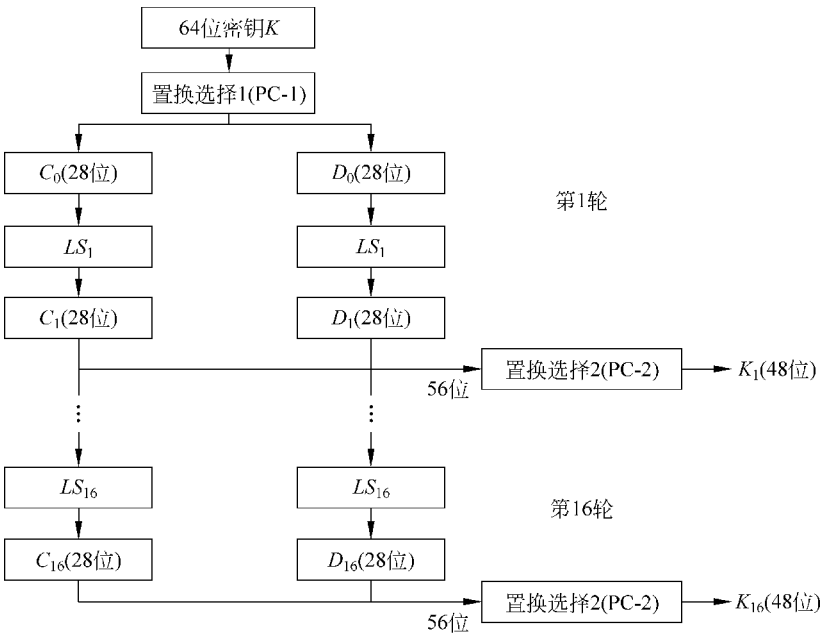

图3-17 子密钥的产生流程

子密钥的生成过程包括置换选择1(PC-1)、循环左移、置换选择2(PC-2)等变换,分别产生16个子密钥。

(1) 置换选择1(PC-1)。对于64位初始密钥K,按表3-9所示的置换选择表PC-1进行重新排列。不难算出,丢掉了其中8的整数倍位置上的比特位,置换选择1后的变换结果是56位。将前28位记为C_0,后28位记为D_0。

表3-9 "置换选择1(PC-1)"的置换表

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

(2) 循环左移。在不同轮次,循环左移LS_i($1 \leqslant i \leqslant 16$)的位数不同,如表3-10所示。第1轮循环左移$LS_1=1$,第2轮循环左移$LS_2=1$,第3轮循环左移$LS_3=2$,依次类推。

表3-10 循环左移的位数

迭代次数 i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
循环左移 LS_i	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

(3) 置换选择 2(PC-2)。与置换选择 1 相同,对输入的 32 位比特串按表 3-11 所示的置换选择表 PC-2 进行重新排列,输出即为子密钥。

表 3-11　“置换选择 2(PC-2)”的置换表

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

4. DES 解密过程

DES 解密过程的逻辑结构与加密过程一致,但必须注意以下两点。

(1) 第 16 轮迭代结束后须将左右两个分组交换位置,即将 L_{16} 与 R_{16} 交换顺序。

(2) 解密过程中使用的子密钥的顺序与加密时的顺序正好相反,依次为 K_{16},K_{15},…,K_1,即当把 64 位密文作为明文输入时,解密过程的第 1 轮迭代使用子密钥 K_{16},第 2 轮迭代使用子密钥 K_{15},…,第 16 轮迭代使用子密钥 K_1,同理,第 16 轮迭代后须交换顺序,最终输出得到 64 位明文。

5. DES 算法的安全隐患

DES 算法具有以下 3 种安全隐患。

(1) 密钥太短。DES 的初始密钥实际长度只有 56 位,批评者担心这个密钥长度不足以抵抗穷举搜索攻击,穷举搜索攻击破解密钥最多尝试的次数为 2^{56} 次,不太可能提供足够的安全性。1998 年前只有 DES 破译机的理论设计,1998 年后出现实用化的 DES 破译机。

(2) DES 的半公开性。DES 算法中的 8 个 S 盒替换表的设计标准(指详细准则)自 DES 公布以来仍未公开,替换表中的数据是否存在某种依存关系,用户无法确认。

(3) DES 迭代次数偏少。DES 算法的 16 轮迭代次数被认为偏少,在以后的 DES 改进算法中,都不同程度地进行了提高。

6. 三重 DES 应用

针对 DES 密钥位数和迭代次数偏少等问题,有人提出了多重 DES 来克服这些缺陷,比较典型的是 2DES、3DES 和 4DES 等几种形式,实用中一般广泛采用 3DES 方案,即三重 DES。它有以下 4 种使用模式。

(1) DES-EEE3 模式:使用 3 个不同密钥(K_1,K_2,K_3),采用 3 次加密算法。

(2) DES-EDE3 模式:使用 3 个不同密钥(K_1,K_2,K_3),采用加密—解密—加密算法。

(3) DES-EEE2 模式:使用两个不同密钥($K_1=K_3$,K_2),采用 3 次加密算法。

(4) DES-EDE2 模式:使用两个不同密钥($K_1=K_3$,K_2),采用加密—解密—加密算法。

3DES 的优点:密钥长度增加到 112 位或 168 位,抗穷举攻击的能力大大增强;DES 基本算法仍然可以继续使用。

3DES的缺点：处理速度相对较慢，因为3DES中共需迭代48次，同时密钥长度也增加了，计算时间明显增大；3DES算法的明文分组大小不变，仍为64位，加密的效率不高。

3.5 非对称密码体制

1976年，Diffie与Hellman在IEEE期刊上提出了划时代的公开密钥密码系统的概念，这个观念为密码学的研究开辟了一个新的方向，有效地解决了秘密密钥密码系统通信双方密钥共享困难的缺点，并引进了创新的数字签名的观念。非对称密码系统(asymmetric encryption)可为加解密或数字签名系统。由于加密或签名验证密钥是公开的，故称为公钥(public key)，而解密或签名产生密钥是秘密的，故称为私钥(private key)。因为公钥与私钥不同，且公钥与私钥必须存在成对(key pair)与唯一对应的数学关系，使得由公钥去推导私钥在计算上不可行，因此非对称密码系统又称为公开密钥系统或双钥系统，其模型如图3-18所示。公钥密码体制的公钥密码算法是基于数学问题求解的困难性而提出的算法，它不再是基于替代和置换的方法。

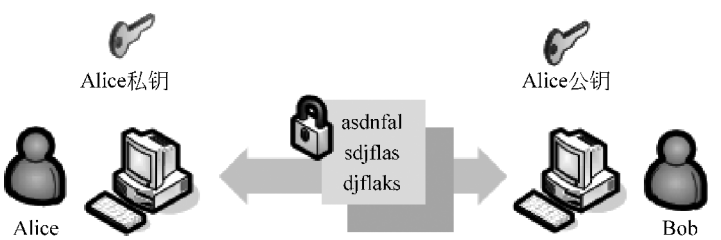

图3-18 非对称密码模型

公钥密码体制的产生主要基于两个原因：一是为了解决常规密钥密码体制的密钥管理与分配的问题；二是为了满足对数字签名的需求。因此，公钥密码体制在消息的保密性、密钥分配和认证领域有着重要的意义。

在公钥密码体制中，公钥是可以公开的信息，而私钥是需要保密的。加密算法E和解密算法D也都是公开的。用公钥对明文加密后，仅能用与之对应的私钥解密，才能恢复出明文，反之亦然。

公钥密码体制的优缺点如下。

优点：网络中的每一个用户只需要保存自己的私钥，N个用户仅需产生N对密钥。密钥少，便于管理；密钥分配简单，不需要秘密的通道和复杂的协议来传送密钥。公钥可基于公开的渠道(如密钥分发中心)分发给其他用户，而私钥则由用户自己保管；可以实现数字签名。

缺点：与对称密码体制相比，公钥密码体制的加密、解密处理速度较慢，同等安全强度下公钥密码体制的密钥位数要求多一些。

公钥密码体制比较流行的主要有两类：一类是基于因子分解难题的，其中最典型的是RSA密码算法；另一类是基于离散对数难题的，如EIGamal公钥密码体制和椭圆曲线公钥密码体制。

3.5.1　RSA 密码算法

RSA 密码算法是美国麻省理工学院的 Rivest、Shamir 和 Adleman 3 位学者于 1978 年提出的。RSA 密码算法方案是唯一被广泛接受并实现的通用公开密码算法，目前已经成为公钥密码的国际标准。它是第一个既能用于数据加密，也能用于数字签名的公开密钥密码算法。在 Internet 中，电子邮件收、发的加密和数字签名软件 PGP 就采用了 RSA 密码算法。

1. 算法描述

RSA 密码算法描述如下。

(1) 密钥对的产生。

① 选取两个大素数 p 和 q。

② 计算 $n=p\cdot q$ 及 n 的欧拉函数值 $\varphi(n)=(p-1)(q-1)$。

③ 然后随机选取整数 $e(1<e<\varphi(n))$，且满足 $\mathrm{GCD}(e,\varphi(n))=1$(GCD 表示求最大公约数运算)，即 $\varphi(n)$ 和 e 互素。

④ 由扩展的欧几里得算法求出 d，使得 $e\cdot d=1 \bmod \varphi(n)$。

⑤ 形成密钥对，其中公钥为 $\{e,n\}$，私钥为 $\{d,n\}$。p,q 是秘密参数，需要保密，如不需要保存，可销毁。

(2) 加密过程。加密时要使用接收方的公钥，不妨设接收方的公钥为 e，明文 m 满足 $0\leqslant m<n$(否则需要进行分组)，计算 $c=m^e(\bmod n)$，c 为密文。

(3) 解密过程。计算 $m=c^d(\bmod n)$。

【例 3-10】 选取 $p=11,q=13$，则

$$n=p\cdot q=11\times 13=143$$

$$\varphi(n)=(p-1)(q-1)=(11-1)(13-1)=120$$

然后，选择 $e=17$，满足 $\mathrm{GCD}(e,\varphi(n))=\mathrm{GCD}(17,120)=1$，计算 $d=e^{-1}(\bmod 120)$。因为 $1=120-7\times 17$，所以 $d=-7=113(\bmod 120)$，则公钥为 $(e,n)=(17,143)$，私钥为 $d=113$。

假设对明文信息 $m=24$ 进行加密，则密文为

$$c=m^e=24^{17}=7(\bmod 143)$$

密文 c 经公开信道发送到接收方后，接收方用私钥 d 对密文进行解密：

$$m=c^d=7^{113}=24(\bmod 143)$$

从而正确地恢复出明文。

2. 安全性分析

(1) RSA 的安全性依赖于著名的大整数因子分解的困难性问题。如果要求 n 很大，则攻击者将其成功地分解为 $p\cdot q$ 是困难的。反之，若 $n=p\cdot q$，则 RSA 便被攻破。因为一旦求得 n 的两个素因子 p 和 q，那么立即可得 n 的欧拉函数值为 $\varphi(n)=(p-1)(q-1)$，再利用欧几里得扩展算法求出 RSA 的私钥 $d=e^{-1}(\bmod \varphi(n))$。

虽然大整数的因子分解是十分困难的，但是随着科学技术的发展，人们对大整数因子分解的能力在不断提高，而且分解所需的成本在不断下降。1994 年，一个通过 Internet 上

1600余台计算机进行合作的小组仅仅在工作了8个月后就成功分解了129位的十进制数,1996年4月又破译了RSA-130,1999年2月又成功地分解了140位的十进制数。1999年8月,阿姆斯特丹的国家数学与计算机科学研究所一个国际密码研究小组通过一台Cray900-16超级计算机和300台个人计算机进行分布式处理,运用二次筛选法花费7个多月的时间成功地分解了155位的十进制数(相当于512位的二进制数)。这些工作结果使人们认识到,要安全地使用RSA,应当采用足够大的整数 n,建议选择 p 和 q 大约是100位的十进制素数,此时模长 n 大约是200位十进制数(实际要求 n 的长度至少是512比特),e 和 d 选择100位左右,密钥$\{e,n\}$或$\{d,n\}$的长度大约是300位十进制数,相当于1024位二进制数(因为 $\mathrm{lb}\ 10^{308}=308\times\mathrm{lb}\ 10\approx1024$)。不同应用可视具体情况而定,如安全电子交易(Secure Electronic Transaction,SET)协议中要求认证中心采用2048比特的密钥,其他实体则采用1024比特的密钥。

(2) RSA的加密函数是一个单向函数,在已知明文 m 和公钥$\{e,n\}$的情况下,计算密文是很容易的;但反过来,在已知密文和公钥的情况下,恢复明文是不可行的。从分析(1)中得知,在 n 很大的情况下,不可能从$\{e,n\}$中求得 d,也不可能在已知 c 和$\{e,n\}$的情况下求得 d 或 m。

3.5.2 Diffie-Hellman密钥交换算法

Diffie和Hellman在1976年发表的论文中提出了公钥密码思想,但没有给出具体的方案,原因在于没有找到单向函数,但在该文中给出了通信双方通过信息交换协商密钥的算法,即Diffie-Hellman密钥交换算法,这是第一个密钥协商算法,只能用于密钥分配,而不能用于加密或解密信息。

1. 算法描述

Diffie-Hellman的安全性是基于 Z_p 上的离散对数问题。设 p 是一个满足要求的大素数,并且 $g(0<g<p)$是循环群 Z_p 的生成元,g 和 p 公开,所有用户都可以得到 g 和 p。在两个用户A与B通信时,它们可以通过如下步骤协商通信所使用的密钥。

(1) 用户A选取一个大的随机数 $\alpha(2\leqslant\alpha\leqslant p-2)$,计算 $S_A=g^{\alpha}(\mathrm{mod}\ p)$,并且把 S_A 发送给用户B。

(2) 用户B选取一个大随机数 $\beta(2\leqslant\beta\leqslant p-2)$,计算 $S_B=g^{\beta}(\mathrm{mod}\ p)$,并且把 S_B 发送给用户A。

(3) 用户A收到 S_B 后,计算 $k_{AB}=S_B^{\alpha}(\mathrm{mod}\ p)$;用户B收到 S_A 后,计算 $k_{BA}=S_A^{\beta}(\mathrm{mod}\ p)$。

由于有 $k_{AB}=S_B^{\alpha}(\mathrm{mod}\ p)=(g^{\beta}(\mathrm{mod}\ p))^{\alpha}(\mathrm{mod}\ p)=g^{\alpha\beta}(\mathrm{mod}\ p)=S_A^{\beta}(\mathrm{mod}\ p)=k_{BA}$,令 $k=k_{AB}=k_{BA}$,这样用户A和B就拥有了一个共享密钥 k,就能以 k 作为会话密钥进行保密通信了。

2. 安全性分析

当模 p 较小时,很容易求出离散对数。依目前的计算能力,当模 p 达到至少150位十进制数时,求离散对数成为一个数学难题。因此,Diffie-Hellman密钥交换算法要求模 p 至少达到150位十进制数,其安全性才能得到保证。但是,该算法容易遭受中间人攻击。造成中间人攻击的原因在于通信双方交换信息时不认证对方,攻击者很容易冒充其中一方获得

成功。

3.5.3　EIGamal 加密算法

EIGamal 公钥密码体制是由 EIGamal 在 1985 年提出的，是一种基于离散对数问题的公钥密码体制。该密码体制既可用于加密，又可用于数字签名，是除 RSA 密码算法之外最有代表性的公钥密码体制之一。由于 EIGamal 体制有较好的安全性，因此得到了广泛的应用。著名的美国数字签名标准 DSS 就是采用了 EIGamal 签名方案的一种变形。其算法描述如下。

(1) 密钥生成。首先随机选择一个大素数 p，且要求 $p-1$ 有大素数因子。$g \in Z_p^*$（Z_p 是一个有 p 个元素的有限域，Z_p^* 是由 Z_p 中的非零元构成的乘法群）是一个生成元。然后再选一个随机数 $x(1 \leqslant x \leqslant p-1)$，计算 $y = g^x \pmod p$，则公钥为 (y, g, p)，私钥为 x。

(2) 加密过程。不妨设信息接收方的公私钥对为 $\{x, y\}$，对于待加密的消息 $m \in Z_p$，发送方选择一个随机数 $k \in Z_{p-1}^*$，然后计算 $c_1 = g^k \pmod p$，$c_2 = my^k \pmod p$，则密文为 (c_1, c_2)。

(3) 解密过程。接收方收到密文 (c_1, c_2) 后，由私钥 x 计算 $c_2 (c_1^x)^{-1} \pmod p$，因

$$c_2 (c_1^x)^{-1} \pmod p = (my^k \pmod p)((g^k \pmod p)^x)^{-1} = m(y(g^x)^{-1})^k \pmod p = m$$

故消息 m 被恢复。

实际上，EIGamal 加密算法最大的特点在于它的“非确定性”。由于密文依赖于执行加密过程的发送方所选取的随机数 k，因此加密相同的明文可能会产生不同的密文。EIGamal 还具有消息扩展因子，即对于每个明文，其密文由两个 Z_p 上的元素组成。EIGamal 通过乘以 y^k 来掩盖明文 m，同样 g^k 也作为密文的一部分进行传送。因为正确的接收方知道解密密钥 x，他可以从 g^k 中计算得到 $(g^k)^x = (g^x)^k = y^k$，从而能够从 c_2 中“去除掩盖”而得到明文 m。

3.6　密码学的应用

密码学的作用不仅仅在于对明文的加密和对密文的解密，更重要的是它可以很好地解决网络通信中广泛存在的许多安全问题，如身份鉴别、数字签名、秘密共享和抗否认等。本节介绍密码应用模式、加密方式。

3.6.1　密码应用模式

DES、IDEA 及 AES 等分组加密算法的基本设计是针对一个分组的加密和解密的操作。然而，在实际的使用中被加密的数据不可能只有一个分组，需要分成多个分组进行操作。根据加密分组间的关联方式，分组密码主要分为以下 4 种模式。

1. 电子密码本模式

电子密码本(Electronic Code Book，ECB)是最基本的一种加密模式，分组长度为 64 位。每次加密均独立，且产生独立的密文分组，每一组的加密结果都不会影响其他分组，如图 3-19 所示。

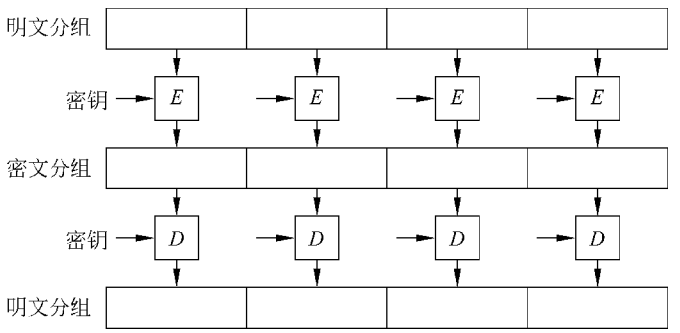

图 3-19 电子密码本模式

电子密码本模式的优点：可以利用平行处理来加速加密、解密运算，且在网络传输时，即使任一分组发生错误，也不会影响到其他分组。

电子密码本模式的缺点：对于多次出现的相同的明文，当该部分明文恰好是加密分组的大小时，可能发生相同的密文，如果密文内容遭到剪贴、替换等攻击，也不容易被发现。

在 ECB 模式中，加密函数 E 与解密函数 D 满足以下关系：

$$D_K(E_K(m)) = m$$

2. 密文链接模式

密文链接(Cipher Block Chaining，CBC)模式的执行方式如图 3-20 所示。第一个明文分组先与初始向量(Initialization Vector，IV)做异或(XOR)运算，再进行加密。其他每个明文分组加密之前，必须与前一个密文分组做一次异或运算，再进行加密。

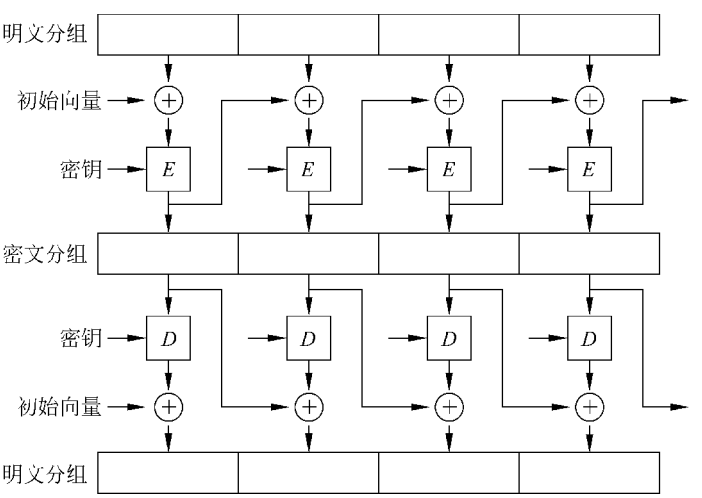

图 3-20 密文链接模式

密文链接模式的优点：每一个分组的加密结果均会受其前面所有分组内容的影响，所以即使在明文中多次出现相同的明文，也不会产生相同的密文；另外，密文内容若遭剪贴、替换，或者在网络传输的过程中发生错误，则其后续的密文将被破坏，无法顺利解密还原，因此，这一模式很难伪造成功。

密文链接模式的缺点：如果加密过程中出现错误，则这种错误会被无限放大，从而导致

加密失败；这种加密模式很容易受到攻击，遭到破坏。

在 CBC 模式中，加密函数 E 与解密函数 D 满足以下关系：

$$D_K(E_K(m)) = m$$

3. 密文反馈模式

密文反馈(Cipher Feed Back，CFB)模式如图 3-21 所示。CFB 需要一个初始向量 IV，加密后与第一个分组进行异或运算产生第一组密文；然后，对第一组密文加密再与第二个分组进行异或运算取得第二组密文，依次类推，直至加密完毕。

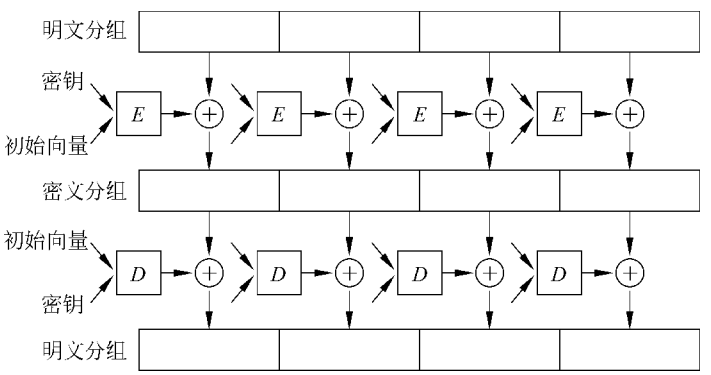

图 3-21　密文反馈模式

密文反馈模式的优点：每一个分组的加密结果受其前面所有分组内容的影响，即使出现多次相同的明文，均产生不相同的密文；这一模式可以作为密钥流生成器，产生密钥流。

密文反馈模式的缺点：与 CBC 模式的缺点类似。

在 CFB 模式中，加密函数 E 和解密函数 D 相同，满足以下关系：

$$D_K(\cdot) = E_K(\cdot)$$

4. 输出反馈模式

输出反馈(Output Feed Back，OFB)模式如图 3-22 所示。该模式产生与明文异或运算的密钥流，从而产生密文，这一点与 CFB 大致相同，唯一的差异点是与明文分组进行异或运算的输入部分是反复加密后得到的。

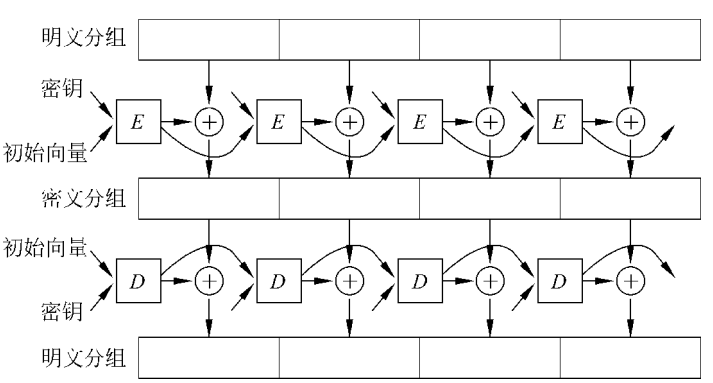

图 3-22　输出反馈模式

在OFB模式中,加密函数 E 和解密函数 D 相同,满足以下关系:

$$D_K(\cdot) = E_K(\cdot)$$

3.6.2 加密方式

在计算机网络中,既要保护网络传输过程中的数据,又要保护存储在计算机系统中的数据。对传输过程中的数据进行加密,称为"通信加密";对计算机系统中存储的数据进行加密,称为"文件加密"。如果以加密实现的通信层次来区分,加密可以在通信的3个不同层次来实现,即节点加密、链路加密和端到端加密3种。

1. 节点加密

节点加密是指对源节点到目的节点之间传输的数据进行加密。它工作在OSI参考模型的第一层和第二层;从实施对象来讲,它仅对报文加密,而不对报头加密,以便于传输路由根据其报头的标识进行选择。一般的节点加密使用特殊的加密硬件进行解密和重加密,因此,要保证节点在物理上是安全的,以避免信息泄露。

2. 链路加密

链路加密是对相邻节点之间的链路上所传输的数据进行加密。它工作在OSI参考模型的第二层,即在数据链路层进行。链路加密侧重于在通信链路上而不考虑信源和信宿,对通过各链路的数据采用不同的加密密钥提供安全保护,它不仅对数据加密,而且还对高层的协议信息(地址、检错、帧头帧尾)加密,在不同节点对之间使用不同的加密密钥。但在节点处,要先对接收到的数据进行解密,获得路由信息,然后再使用下一个链路的密钥对消息进行加密,再进行传输。在节点处传输数据以明文方式存在。因此,所有节点在物理上必须是安全的。

3. 端到端加密

端到端加密是指为用户传送数据提供从发送端到接收端的加密服务。它工作在OSI参考模型的第六层或第七层,由发送端自动加密信息,并进入TCP/IP数据包回封,以密文的形式穿过互联网,当这些信息到达目的地时,将自动重组、解密,成为明文。端到端加密是面向用户的,它不对下层协议进行信息加密,协议信息以明文形式传输,用户数据在传输节点不需要解密。由于网络本身并不会知道正在传送的数据是加密数据,因此这对防止复制网络软件和软件泄露很有效。在网络上的每个用户可以拥有不同的加密密钥,而且网络本身不需要增添任何专门的加密、解密设备。

实训1:密码学实验

密码学是一门古老的科学,它是研究密码系统或通信安全的一门科学,分为密码编码学和密码分析学。密码编码学的目的是研究如何书写好的密码方法,保护信息不被侦察,即伪装消息。对给定的有意义的数据进行可逆的数学变换,将其变为表面上杂乱无章的数据,只有合法的接收者才能恢复数据。密码分析学是研究攻破一个系统的途径,恢复被隐蔽起来的消息的本来面目,即研究如何破译加密的消息。

实训目的

1. 掌握常用密码算法的设计思想及其安全性原理。
2. 能通过 CAP 进行常用密码算法的演示和分析。

实训环境

1. 设备：计算机。
2. 软件：CAP 加密分析软件。

实训内容

1. 经典加密法演示及分析。
2. 公钥加密法演示及分析。

实训步骤

1. 经典加密法演示及分析

(1) 原理及演示。首先，选择一个关键词，若关键词中有重复字母，则去除第一次出现之外的所有相同字母。例如，选定关键词“good”，则使用“god”。

然后，将关键词写在字母表下方，并用字母表的其他字母按标准的顺序填写余下的空间，如表 3-12 所示。

表 3-12　原表

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
g	o	d	a	b	c	e	f	h	i	j	k	l	m	n	p

显然，从字母 p 开始，所有的字母都不再替换。为了消除这种情况，可以允许关键词从字母表的任意位置开始，如让“god”从 h 开始，如表 3-13 所示。

表 3-13　新的替代表

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
t	u	v	w	x	y	z	g	o	d	a	b	c	e	f	h

最后，用 CAP 软件演示关键词加密法。

① 在 CAP 界面中单击 Cipher 菜单项，选择 Keyword 选项。在打开的界面中输入关键词和起始位置后，用 SetKey 设定，可在下方看到对应的字母表，如图 3-23 所示。

② 单击 Encipher 标签加密明文 this is a test for cipher experiment，即可在 Ciphertext 文本框中显示密文，如图 3-24 所示。

(2) 破解。在进行破解之前，需要了解一些关键词加密法的重要内容：明文是标准英语；每个明文字母已被唯一的密文替代。

于是，在分析破解的过程中，就可以利用字母的一些特性帮助破解，如字母出现的频率、与其他字母的联系、在单词中的位置等。(在标准英语中，e 是出现频率最高的字母，而 e 和 z 很少成对出现；th、he 和 er 成对出现很普遍。)

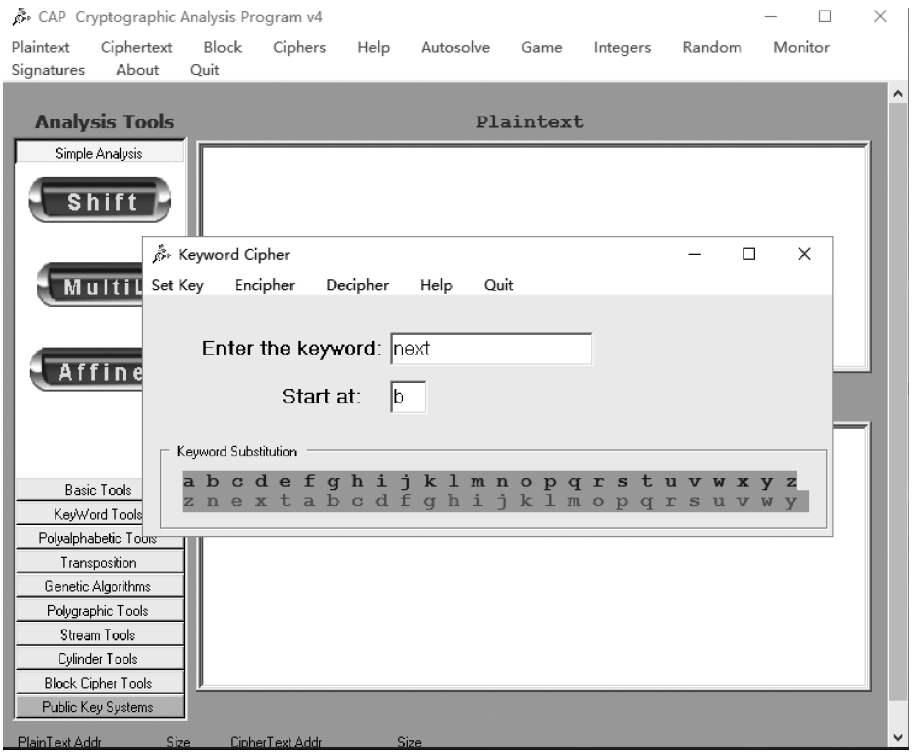

图 3-23 关键词加密法 Keyword 设定

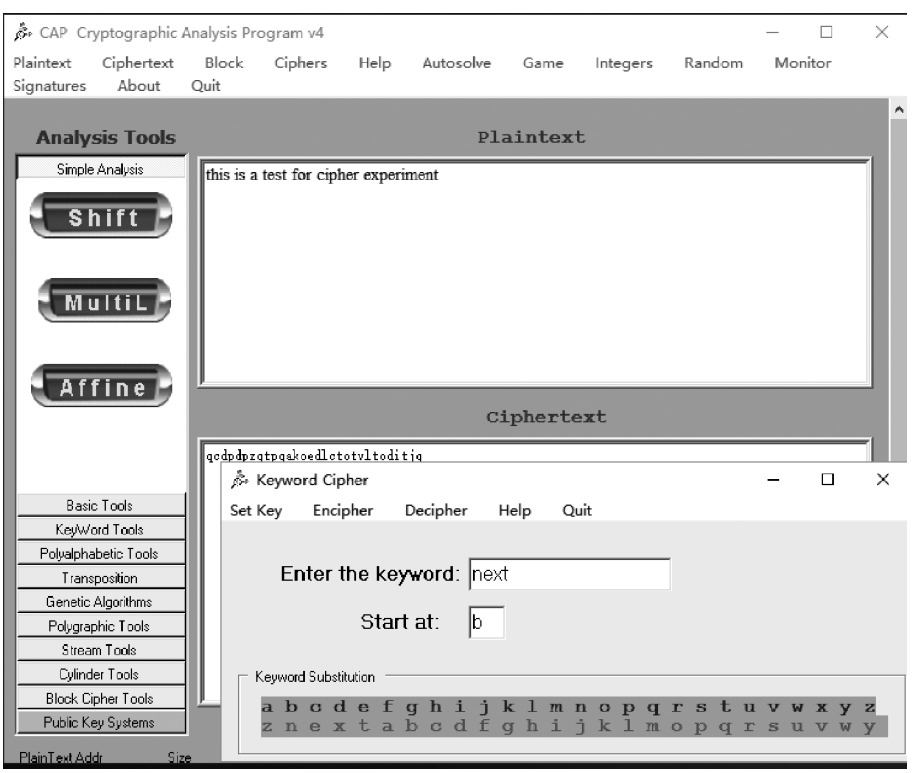

图 3-24 对明文加密

2. 公钥加密法演示及分析

(1) RSA 的实施。实施 RSA 公开密钥密码体制需要以下步骤。

① 设计密钥。仔细选取两个互异的大素数 p 和 q；令 $n=p\cdot q$ 及 n 的欧拉函数值 $\varphi(n)=(p-1)(q-1)$，接着寻找两个正整数 e 和 d，使其满足 $\mathrm{GCD}(e,\varphi(n))=1$，$e\cdot d=1(\mathrm{mod}\ \varphi(n))$。这里的$\{e,n\}$就是公开的加密密钥。

② 设计密文。把要发送的明文信息数字化和分块，其加密过程为 $c=m^e(\mathrm{mod}\ n)$。

③ 恢复明文。对 c 解密，$m=c^d(\mathrm{mod}\ n)$即可得到明文。

(2) CAP 软件的 RSA 实现。

① 选择 Integers 菜单中的 Prime Number Generation 选项，找到素数 p 和 q，如图 3-25 所示。

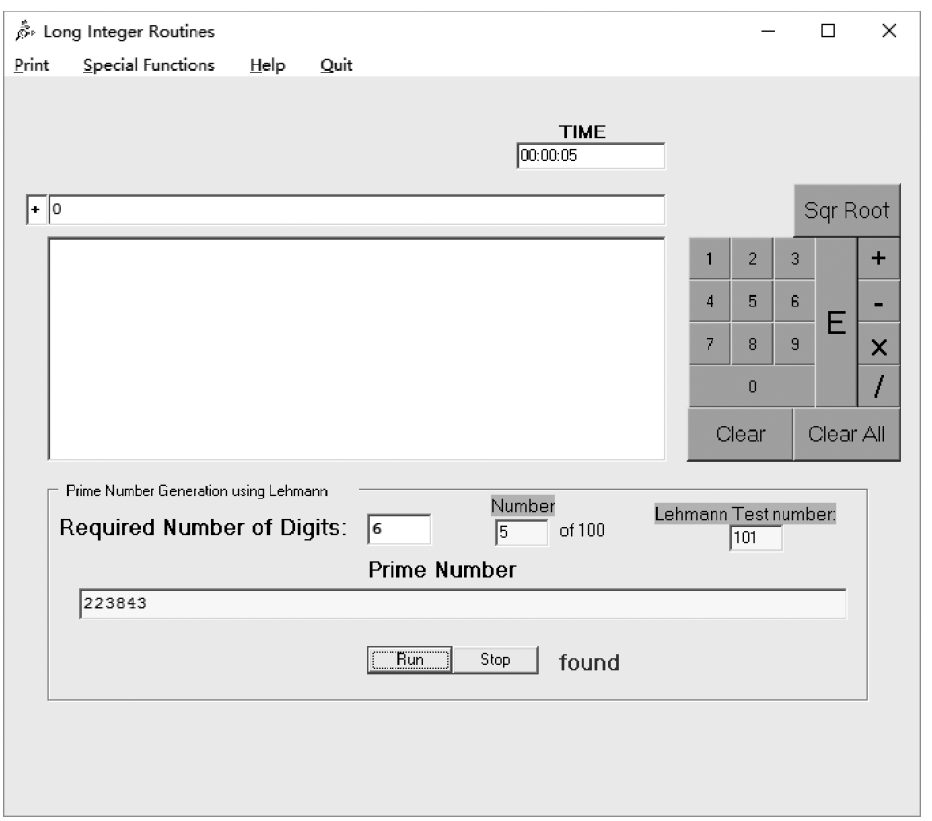

图 3-25　Long Integer Routines 运行界面

② 计算 n 和 $\varphi(n)$。

③ 选取素数 e，该值比 p 值大，且与$(p-1)(q-1)$互质；并在 Special Function 选项卡中的 Inverse Mod 栏，用反模运算得到私钥 d，如图 3-26 所示。

④ 在 Cipher 菜单中选择 RSA 选项，在打开的界面中输入之前得到的公钥和私钥，如图 3-27 所示。

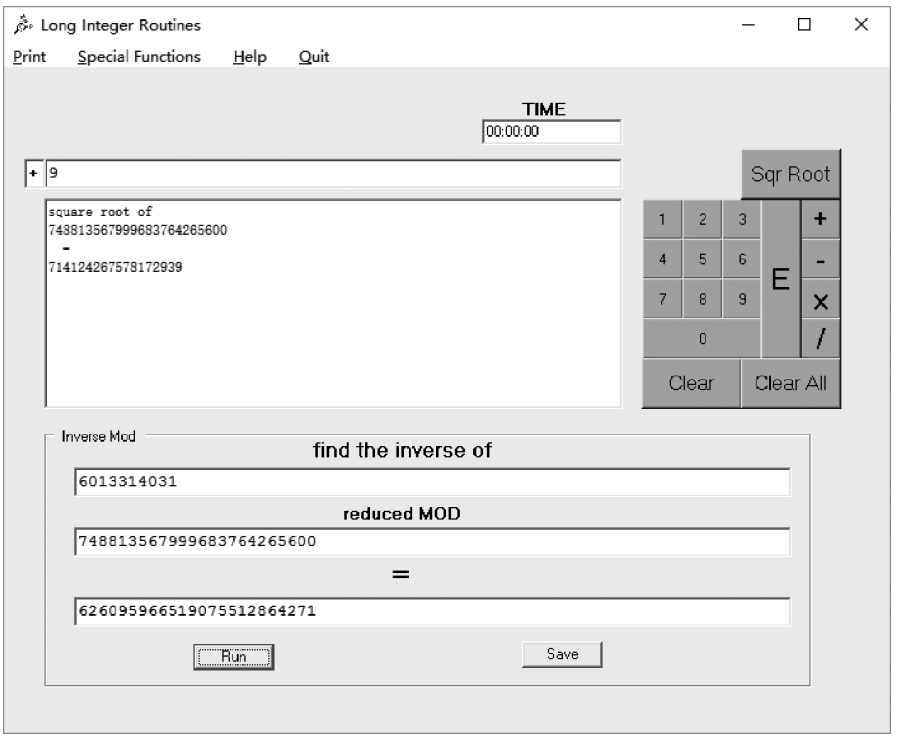

图 3-26 反模运算界面

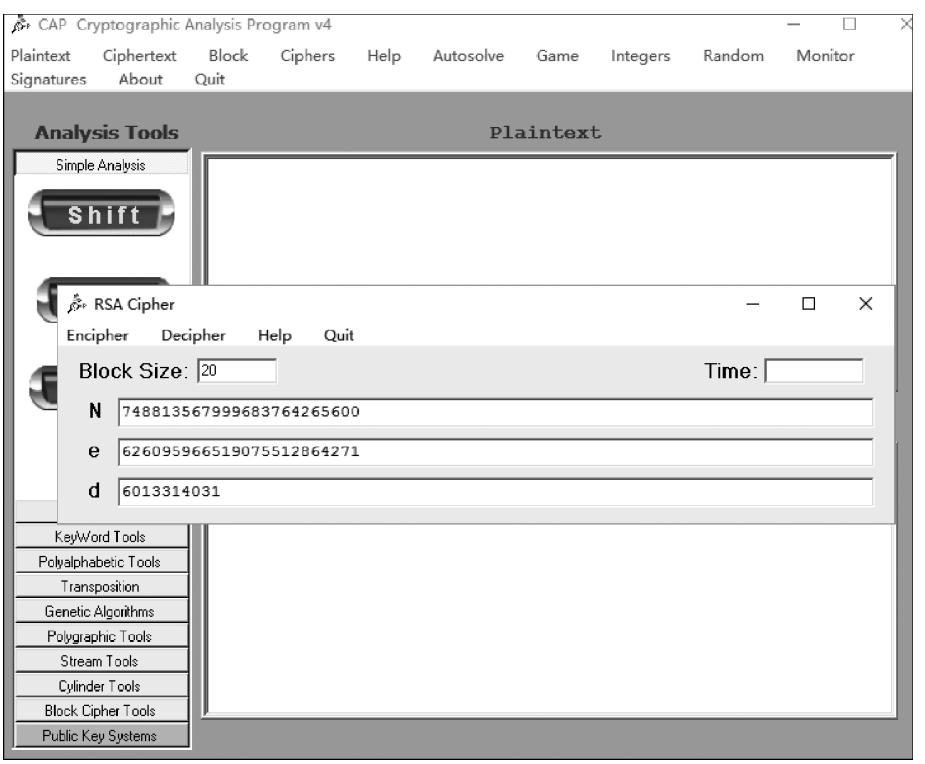

图 3-27 CAP 软件中的 RSA 实现

实训 2：PGP 加密软件的使用

PGP(Pretty Good Privacy)是一个基于 RSA 公钥加密体系的邮件加密软件，使用混合加密体制加密。PGP 最初的设计主要是用于电子邮件加密，如今已经发展到了可以加密整个硬盘、分区、文件、文件夹、邮件软件等。

实训目的

1. 了解 PGP 加密软件的基本功能。
2. 加深对公钥加密机制的理解。
3. 掌握 PGP 软件的加解密文件、签名等基本操作。

实训环境

网络环境，PGP 加密软件。

实训内容

1. 安装 PGP 软件。
2. 练习生成密钥对。
3. 练习一般文件的加密和解密。

实训步骤

1. 安装 PGP 软件

实验使用的是 PGP Desktop 10，分别有 32 位和 64 位版本，支持 Windows 2003、Windows 7、Windows 10，PGP 的安装要求使用管理员账号登录。

当提问是否已经有密钥对时，本实验选择“新用户”选项(如果已经有了自己的密钥对，就不要选择)。继续按照提示一步步安装，在安装完成之后，需要重新启动计算机，重启后，屏幕右下角的任务栏上会出现一个金黄色的“小锁”，这就是 PGP 图标，如图 3-28 所示。然后安装汉化包，重启计算机。

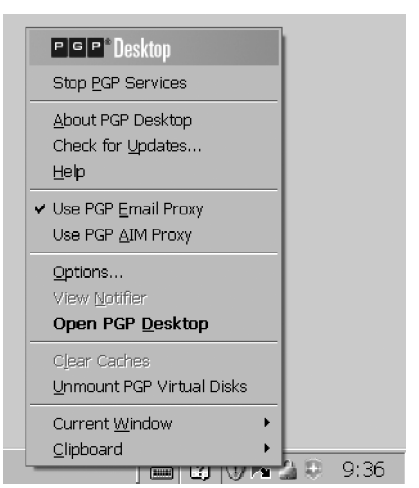

图 3-28　PGP 菜单

2. 创建和管理 PGP 的密钥

要使用 PGP 进行加密、解密和数字签名，首先必须生成一对属于自己的密钥对，公钥发送给其他人，让其进行加密；私钥留给自己用来解密及签名。PGP 的密钥经过加密后保存在文件中。

创建密钥对的步骤如下。

(1) 如果在安装 PGP 时选择“新用户”选项，安装程序将自动打开密钥对生成向导。也可以通过“开始”→“程序”→PGP→PGPkeys 命令，启动 PGPkeys 主界面，如图 3-29 所示。选择“密钥”→“新建密钥”命令，打开“PGP 密钥生成向导”界面，如图 3-30 所示。

图 3-29 PGPkeys 主界面

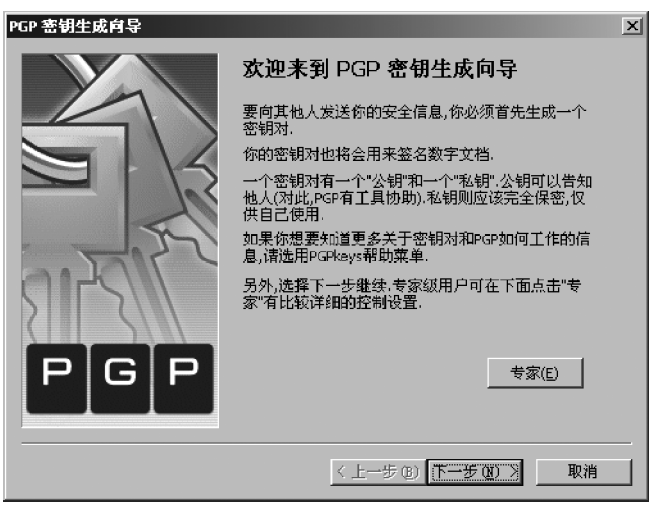

图 3-30 PGP 密钥生成向导

(2) 在“分配姓名和电子邮箱”中填写自己的姓名、邮件地址。

(3) 在“分配密码”文本框中输入自己的私钥保护密码。在需要使用私钥时输入此密码(注意,PGP 并不直接使用这个密码加密数据,并且可以更改这个密码,只要没有生成新的密钥对,仍然可以使用原来的密钥对进行文件的加密和解密)。

(4) 一直单击“下一步”按钮直到完成。这时将在密钥管理器中出现所生成的密钥对,如图 3-31 所示。

图 3-31 显示新建的密钥

导出公钥:在 PGPKey 中,选择自己的密钥对。选择“密钥”→“导出”命令可以导出自己的公钥或整个密钥对,如图 3-32 所示。可将自己的公钥导出到一个文件中。将这个密钥对文件上传到实验室服务器中的“PGP 公钥”中。

图 3-32　导出密钥对

导出自己的密钥对：当需要将密钥对备份起来，或者需要把密钥对转移到另外的计算机上时，可以利用导出密钥对的功能。可将自己的密钥对导出到另一个文件中，将这个密钥对文件保存到自己的邮箱或 U 盘上。

删除密钥对：按 Delete 键将选择的密钥对删除。

导入密钥对：选择“import”命令将刚才导出的密钥对重新导入到密钥管理器中。

导入其他人的公钥：从实验服务器上下载同组实验同学的公钥文件，将该公钥导入密钥管理器中。

3. 使用 PGP 进行文件的加密和解密

(1) 创建一个文本文件 test.txt，要包含以下内容。注意，用学生自己的学号、姓名、邮件地址替换相应的内容。

```
姓名：张三
学号：2974329923
邮件地址：zhang3@sise.com.cn

PGP is based on a widely accepted encryption technology known as public key cryptography in which two complementary keys — a keypair — are used to maintain secure communications.
```

(2) 选择 PGP 菜单中的 PGPmail 选项，打开 PGPmail 页面，如图 3-33 所示。

图 3-33　PGPmail

(3) 选择创建的文本文件 test.txt，如图 3-34 所示。

(4) 用刚才导入的公钥进行加密，将加密后的文件重命名为 test1.txt.pgp，如图 3-35 所示。

图 3-34 PGPmail 选择待加密的文件

图 3-35 用导入的公钥进行加密

将用来加密的公钥或密钥对拖入 Recipients 栏,其他不需要的拖出去。

(5) 解密文件,先双击生成的加密文件“test1. txt. pgp”,要求输入密钥的密码,如图 3-36 所示。

图 3-36 解密文件

(6) 输入正确的密码后,就可以解密原来的文件了。

本 章 小 结

(1) 密码学的发展大致经历了手工加密阶段、机械加密阶段和计算机加密阶段。密码技术是现代信息安全的基础和核心技术，它不仅能够对信息加密，还能完成信息的完整性验证、数字签名和身份认证等功能。按加密密钥和解密密钥是否相同，密码体制可分为对称密码体制和非对称密码体制。对称密码体制又可分为序列密码和分组密码。

(2) 替代密码和置换密码等是常用的古典密码案例，虽然在现代科技环境下已经过时，但它们包含的最基本的变换移位和代替在现代分组密码设计中仍然是最基本的变换。

(3) 对称密码体制要求加密、解密双方拥有相同的密钥，其特点是加密速度快、软/硬件容易实现，通常用于传输数据的加密。常用的加密算法有 DES 算法等。

(4) 非对称密码体制的加密密钥和解密密钥是不相同的。非对称密码被用作加密时，使用接收者的公开密钥，接收方用自己的私有密钥解密；用作数字签名时，使用发送方(签名人)的私有密钥加密(或称为签名)，接收方(或验证方)收到签名时使用发送方的公开密钥验证。常用的算法有 RSA 密码算法、Diffie Hellman 密钥交换算法、EIGamal 加密算法等。

(5) 加密可以用通信的 3 个不同层次来实现，即节点加密、链路加密和端到端加密。节点加密是指对源节点到目的节点之间传输的数据进行加密，不对报头加密；链路加密在数据链路层进行，是对相邻节点之间的链路上所传输的数据进行加密，在节点处，传输数据以明文形式存在，侧重于在通信链路上而不考虑信源和信宿；端到端加密是对源端用户到目的端用户的数据提供保护，传输数据在传输过程中始终以密文形式存在。

思　考　题

1. 简述密码学的发展历程。

2. 简述对称密码体制和非对称密码体制的优缺点。

3. 简述柯克霍夫原则。

4. 描述 DES 数据加密算法的流程。

5. 设明文为“visit beijing tomorrow”，密钥为“enjoy”，试用 Vigenere 算法对其加密。

6. 在 Alice 和 Bob 的保密通信中，传送的密文是“rjjy rj ts ymj xfggfym bj bnqq inxhzxxymj uqfs”，如果他们使用的是移位密码算法，试解密其通信内容。

7. 在一个使用 RSA 的公开密钥系统中，若截获了发给一个其公开密钥 $e=5$、$n=35$ 的用户的密文 $C=10$，则明文 M 是什么？

8. 简述 EIGamal 加密算法。

第4章　信息隐藏技术

信息隐藏(Information Hiding)作为一门新兴的交叉学科,伴随着信息和网络技术的飞速发展,在隐蔽通信、数字版权保护等方面起着越来越重要的作用。信息隐藏是将秘密信息隐藏在另一非机密的载体信息中,通过公共信道进行传递。秘密信息被隐藏后,攻击者无法判断载体信息中是否隐藏了秘密信息,也无法从载体信息中提取或去除所隐藏的秘密信息。信息隐藏技术研究的内容包括信息隐藏算法、数字水印、隐通道技术和匿名通信技术等。

4.1　信息隐藏的发展历史

4.1.1　传统的信息隐藏技术

古代信息隐藏的方法可以分为两种:一种是将机密信息进行各种变换,使非授权者无法理解,这就是密码术;另一种是将机密信息隐藏起来,使非授权者无法获取,如隐写术等。可以称它们为古代密码术和古代隐写术。可以把它们的发展看成两条线:一条是从古代密码术到现代密码学;另一条是从古代隐写术到信息隐藏技术。

古代隐写术包括技术性的隐写术、语言学中的隐写术和用于版权保护的隐写术。

1. 技术性的隐写术

技术性的隐写术由来已久。大约在公元前440年,隐写术就已经被应用了。据古希腊历史学家希罗多德记载,一位希腊贵族希斯泰乌斯(Histiaus)为了安全地把机密信息传送给米利都的阿里斯塔格鲁斯,怂恿他起兵反叛波斯人,想出一个绝妙的主意:剃光送信奴隶的头,在头顶上写下密信,等他的头发重新长出来,就将他派往米利都送信。类似的方法在20世纪初期仍然被德国间谍所使用。实际上,隐写术自古以来就一直被人们广泛地使用。隐写术的经典手法很多,下面仅列举一些例子。

(1) 使用不可见墨水给报纸上的某些字母作上标记来向间谍发送消息。

(2) 在一个录音带的某些位置上加一些不易察觉的回声。

(3) 将消息写在木板上,然后用石灰水把它刷白。

(4) 将信函隐藏在信使的鞋底里或妇女的耳饰中。

(5) 由信鸽携带便条传送消息。

(6) 通过改变字母笔画的高度或在掩蔽文体的字母上面或下面挖出非常小的小孔(或用无形的墨水印制作非常小的斑点)来隐藏正文。

(7) 在纸上打印各种小像素点组成的块来对诸如日期、打印机进行标识,将用户标识符等信息进行编码。

(8) 将秘密消息隐藏在大小不超过一个句号或小墨水点的空间里。

(9) 将消息隐藏在微缩胶片中。