第一章　密码学的基本概念

§1.1 密码学的应用

自从有了消息的传递，就有了对消息保密的需要。因此，密码学的历史可以说是源远流长。但由于在很长的时间内，密码仅限于军事、政治和外交的用途，密码学的知识和经验也仅掌握在与军事、政治和外交有关的密码机关手中，再加上通信手段比较落后，所以，不论密码理论还是密码技术，发展都很缓慢。

由于科学技术的进步，信息交换的手段越来越先进，信息交换的速度越来越快，信息交换的内容越来越广泛，信息交换的形式越来越丰富，信息交换的规模也越来越大。到了本世纪 70 年代，随着信息的激增，对信息保密的需求也从军事、政治和外交等领域迅速扩展到民用和商用领域，从而导致了密码学知识的广泛传播。计算机技术和微电子技术的发展为密码学理论的研究和实现提供了强有力的手段和工具。进入 80 年代以后，对密码理论和技术的研究更是呈爆炸性增长趋势。密码学在雷达、导航、遥控、遥测等领域占有重要地位，这已是众所周知的事实。除此之外，密码学正渗透到通信、电子邮政、计算机、金融系统、各种管理信息系统甚至家庭等各部门和领域。保密的作用也已不再仅仅是“保密”，还有认证、鉴别和数字签名等新的功能。在对信息密码的需求日益广泛和日益迫切的同时，人们对密码技术的要求也越来越高。

对普通的家庭来说，生活中许多地方需要保密。个人隐私需要保密；家庭收入需要保密；现金储蓄需要保密；私人通话常常牵涉到个人秘密，因此也需要保密……。

对于学校来说，管理部门越来越多地依赖于计算机。例如人事档案、工资分配、财务数据、科研项目和科研成果管理、课程和教室管理、资源和设备管理、学生档案和学生成绩等重要数据都存储于计算机信息媒体上。这不仅要求数据是保密的，而且要求数据是完整的，即未经授权的人不能增加、修改、删除或破坏任何信息。

对企业来说，产品配方或产品的核心技术需要保密；新产品的设计需要保密；产、销、利润情况需要保密；市场策略需要保密……

在计算机通信网络中，网络中心不希望非法用户接入系统；广大用户希望他自己输入、存储的数据不会被包括系统操作员在内的其他人随意利用；软件设计者不希望自己的辛勤劳动成果被他人无偿占有……。为了保证通信网络能正常、安全地运行，就要求系统能够对信息实施有效保护、对合法用户进行认证并能准确地鉴别出非法用户.这些都需要借助于密码学的力量。

随着我国金关、金卡、金桥、金库、金穗等“金”字工程的相继出台，我国信息高速公路的建设也进入飞速发展阶段。信息高速公路的飞速发展，一方面预示着以往由人工和书面文字完成的业务将越来越多地被“电子业务”所取代。如“电子邮政”将越来越多地代替“书面邮政”；银行系统各类报表“电子报纸”也会在某种程度上冲击“印刷报业”……。另一方面，意味着各类磁卡和 IC 智能卡将在许多领域如金融流通、通信、交通、事物管理、娱乐、防盗系统、个人学习以及服务行业得到广泛应用。如信用卡、高速公路卡、电话卡、身份卡、医疗卡、钥匙卡、会员卡等等。凡此种种，不是需要加密，就是需要认证、鉴别或者是数字签名。的确，高速通信网络安全可靠的运行，是信息高速公路成功的必要条件。可见，在社会越来越依赖于信息的同时，信息密码与安全的地位也越来越重要。

§1.2 密码学的基本概念

密码学是保密学的一部分，保密学是研究密码系统或通信安全的科学 ，它包含两个分支，既密码学（cryptology）和密码分析学（cryptanalytics）。密码学是对信息进行编码实现隐蔽信息的一门学问，密码分析学是研究分析破译密码的学问。两者相互对立，而又相互促进。

采用密码方法可以隐蔽和保护需要保密的消息，使未授权者不能提取信息，被隐蔽的消息称作明文（plaintext），密码可将明文变换成另一种隐蔽形式，称为密文（ciphertext）。这种变换称为加密（encryption）。其逆过程，从密文恢复出明文的过程称为解密（decryption）。对明文进行加密时采用的一组规则称为加密算法，对密文解密时采用的一组规则称为解密算法。加密算法和解密算法通常都是在一组密钥（key）控制下进行的，分别称为解密密钥和解密密钥。若采用的解密密钥和解密密钥相同，或者实际上等同，即从一个易于得出另一个，称为单钥或对称密码体制（one-key *or* symmetric cryptosystem）。若解密密钥和解密密钥不相同，从一个难以推出另一个，则称双钥或公开密码体制（two-key *or* public cryptosystem），这是 1976 年由 Diffie 和 Hellman 等人所开创的新体制，它是密码学理论的划时代突破。密钥是密码体制安全保密的关键，它的产生和管理是密码学中重要的研究课题。

在信息的传输和处理系统中，除了意定的接收者外，还有非授权接收者，他们通过各种办法（如搭线窃听、电磁窃听、声音窃听等）来窃取信息。他们虽然不知道系统所用的密钥，但通过分析可能从截获的密文中推断出原来的明文，这一过程称为密码分析。对一个保密系统采取截获密文进行分析的这类攻击称为被动攻击（passive attack）。密码系统还可能遭受的另一类攻击是主动攻击（active attack）。非法入侵者主动向系统干扰，采用删除、更改、增添、重放、伪造等方法向系统加入假消息，这是通信系统中棘手的问题。

一个密码通信系统可以用图 1—1—1 表示。它由以下几部分组成：

明文消息空间 M；密文消息空间 E；密钥空间 K_1 和 K_2，单钥体制下 $K_1 = K_2 = K$，此时密钥 k 需经过安全的密钥信道由发方传给收方；加密变换 E_{k1}，$M \to E$，其中 $k_1 \in K_1$，由加密器完成；解密变换 D_{k2}，$E \to M$，其中 $k_2 \in K_2$，由解密器实现。称总体 $(M, E, K_1, K_2, E_{k1}, D_{k2})$ 为一保密系统（secrecy system）。对于给定明文消息 $m \in M$，密钥 $k_1 \in K$，加密变换将明文 m 变换为密文 c

$$c = f(m, k_1) = E_{k1}(m) \qquad m \in M, k_1 \in K_1 \qquad (1—1—1)$$

接收端利用通过安全信道送来的密钥 k（单钥体制下）或利用本地密钥发生器产生的解密密钥 $k_2 \in K_2$（双钥体制下）控制解密操作 D，对受到密文进行变换得到恢复的明文消息

$$m = D_{k2}(c) \qquad m \in M, k_2 \in K_2 \qquad (1—1—2)$$

而密码分析者，则利用其选定的变换函数 h，对截获的密文 c 进行变换，得到的明文是明

文空间的某个元素

$$m' = h(c) \qquad m \in M, k_2 \in K_2 \qquad (1—1—3)$$

一般$m' \neq m$。

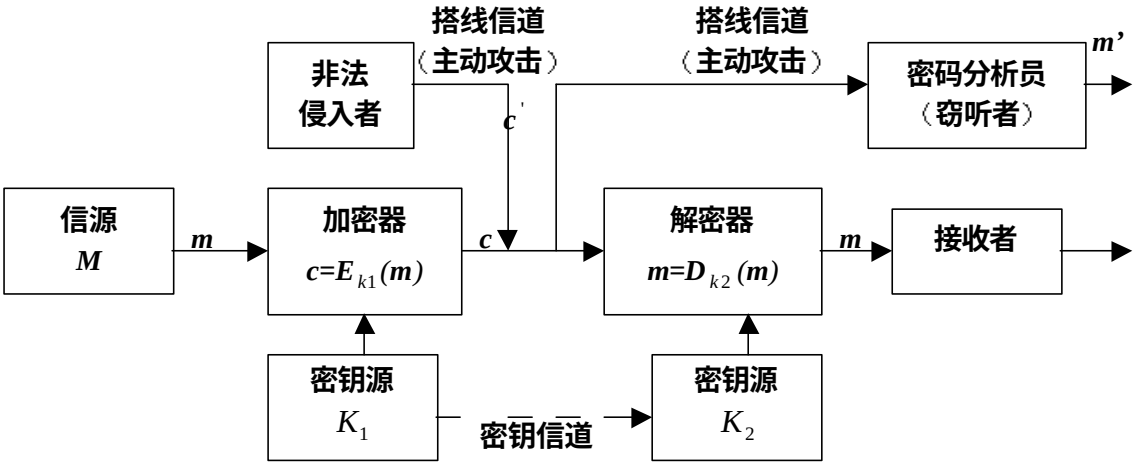


图 1—1—1　密码系统模型

为了保护信息的保密性，抗击密码分析，保密系统应当满足下述要求:

（1）系统即使达不到理论上是不可破的，即 $p_r(m' = m) = 0$，也应当是实际上不可破的。就是说。从截获的密文或某些已知明文密文对，要决定密钥或任意明文在计算上是不可行的。

（2）系统的保密性不依赖于对加密体制或算法的保密，而依赖于密钥。

（3）加密算法和解密算法适用于所有密钥空间中的元素。

（4）系统便于实现和使用方便。

为了防止消息被窜改、删除、重放和伪造的一种有效方法是是使发送的消息具有被验证的能力，使接收者或第三者能够识别和确认消息的真伪，实现这类功能的密码系统被称为认证系统（authentication system）。消息的认证性和消息的保密性不同，保密性是使截获者在不知道密钥的情况下不能解读密文的内容；而认证性是使任何不知道密钥的人不能构造出一个密报，使意定的接收者脱密成一个可理解的消息（合法的消息）。认证理论和技术是随着进 20 年来计算机通信的普遍应用而迅速发展起来的，它成为保密学研究的一个重要领域。入传统的手书签名正在被更迅速、更经济和更安全的数字签名（digital signature）所代替。

一个安全的认证系统应满足下述条件:

（1）意定的接收者能够检验和证实消息的合法性和真实性。

（2）消息的发送者能够对所发的消息不能抵赖。

（3）除了合法的消息发送者之外，其他人不能伪造合法的消息。而且在已知合法密文和相应消息下，要确定加密密钥和系统地伪造合法密文在计算上是不可能的。

（4）必要是可有第三者进行仲裁。

消息的安全性除了上述的保密性和认证性以外，还有一个重要的方面是它的完整性（integrity）。它表示在有自然和认为干扰条件下，系统保持恢复消息和原来发送消息一致

性的能力。实际中常借助于检错和纠错技术来保证消息的完整性。

信息系统的安全的中心内容是保证信息在系统中的保密性、认证性和完整性。

单钥体制的加密密钥和解密密钥相同。对数据进行加密的单钥系统如图 1—1—2 所示。系统的保密性主要取决密钥的安全性。必须通过安全可靠的途径将密钥送至受端。如何产生满足保密要求的密钥是这类体制设计和实现的主要课题。另一个重要的问题是如何将密钥安全可靠地分配给通信对方，在网通信条件下就更为复杂，包括密钥产生、分配、存储、销毁等多方面的问题，统称为密钥管理（key management）。这是影响系统安全的关键因素，即使密码算法再好，若密钥管理问题处理不好，就很难保证系统的安全保密。

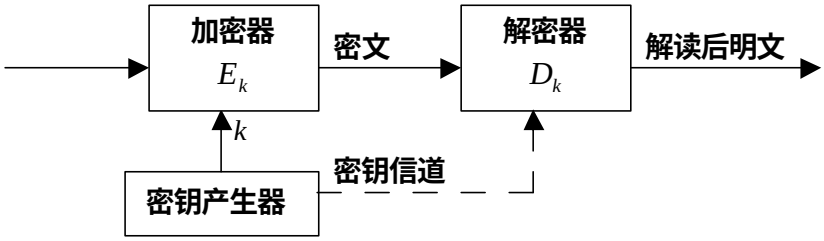


图 1—1—2　单钥保密系统

对明文消息的加密方式有两种方式：一是对明文消息按字符诸位地加密，称之为流密码（stream cipher）；另一种是将明文消息分组（含有多个字符），逐组地进行加密，称之为分组密码（block cipher）。

单钥加密的古典算法有简单代换、多表代换、同态代换等多种，我们将下一章作一简单介绍。单钥体制不仅能用于数据加密，也可用于消息认证，有关内容将在第五章介绍。

第二章　　古典密码体制与密码学信息理论基础

§2.1 古典密码体制

本节简单介绍几种古典密码体制，来说明构造密码的一般方法，这些密码现在已经很少使用了，但研究这些密码的原理，对于理解、构造和分析现代实用密码还是很有用的。

一 、单表代换密码

单表代换密码是对明文的所有字母都用一个固定的明文字母表到密文字母表的映射。即令明文 $m = m_0 m_1 \cdots$，**则相应密文为**

$$c = E(m) = c_0 c_1 \cdots = f(m_0) f(m_1) \cdots \qquad (2—1—1)$$

以下是几种常用的单表代换密码

1. **加同余密码， 也叫移位代换密码**（shift substitution cipher）**是最简单的一类代换密码，其加密变换为:**

$$E_k(i) = (i + k) \equiv j \bmod q \qquad 0 \leqslant i，j < q \qquad (2—1—2)$$

$$K = \{ k \mid 0 \leqslant k < q \}$$

式中， $\bmod q$ **表示以** q **除得的余数。密钥空间元素个数为** q，**其中有一恒等变换，即** $k=0$。
解密变换为：

$$D_k(j) = E_{q-k}(j) \equiv j + q - k \equiv i + k - k \equiv i \bmod q \qquad (2—1—3)$$

例：2—2—1 凯撒（Cacser）密码是对英文 26 个字母进行移位代换的密码。$q=26$，$k=3$，**则有下述代换:**

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

若明文

***m*=**Casear cipher is a shift substitution

则密文

C=E(***m***)=FDVHDU FLSHU LV D VKLIW VXEVWLWXWLRQ

解密运算为 $D_3 = E_{23}$，**即可用密钥** $k=23$ **的加密表进行加密运算就可以恢复出明文。**

2. **乘数密码**（multiplicative cipher）**的加密变换为:**

$$E_k(i) = ik \equiv j \bmod q \qquad 0 \leqslant j < q \qquad (2—1—4)$$

这种密码又称为采样密码（decimation cipher），**因为密文字母表是将明文字母表按下标每格** k **位取出一个字母排列而成**（**字母表首尾相接**）。**显然仅当**（k，q）=1 **时，即** k **与** q **互**

素时，明文字母于密文字母才是一一对应的

例：2—1—2　英文字母表 q=26，取 k=9，则有如下明文密文字母对应表

表 2—1—1

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

对明文

m= multiplicative cipher

有密文

C=EYVPUFVUSAPUHK　SUFLKX

3. 线性同余密码。将移位密码和乘数密码进行组合就可以得到更多的选择方式获密钥。按

$$E_k(i) = ik_1 + k_0 \equiv j \bmod q \qquad k_1, k_2 \in Z_q \qquad (2—1—5)$$

加密称作线性同余密码，也叫仿射密码（affine cipher）。其中，(k_1, q) =1，以$[k_1, k_0]$表示密钥。当$k_1 = 1$时得到移位密码，$k_0 = 0$ 时得到移位密码。

二、 多表代换密码

多表代换密码是以一系列(两个以上)代换表依次对明文消息的字母进行代换的加密方法 ，令明文字母表为Z_q，令$p = (p_1, p_2, \cdots)$为代换序列，明文字母序列为$m = m_1 m_2 \cdots$，则相应的密文字母序列为

$$c = E_k(m) = p(m) = p_1(m_1)p_2(m_2)\cdots \qquad (2—1—6)$$

若 π 为非周期无限序列，则相应的密码为非周期多表代换密码。这类密码，对每个明文字母都采用不同的代换表（或密钥）进行加密，称作是一次一密钥密码（one-time pad cipher）。这是一种在理论上唯一不可破的密码。这种密码对于明文的特征可实现完全隐蔽，担由于需要的密钥量和明文消息长度相同而难以广泛使用。

为了减少密钥量，实际中多采用周期多表代换密码，即代换表个数有限，重复使用，此时代换序列

$$p = p_1 p_2 \cdots p_d p_1 p_2 \cdots p_d \cdots \qquad (2—1—7)$$

相对应于明文字母序列 m 的密文为

$$c = E_k(m) = p(m) = p_1(m_1)p_2(m_2)\cdots p_d(m_d)p_1(m_{d+1})\cdots p_d(m_{2d})\cdots \qquad (2—1—8)$$

当 d=1 时就退化为单表代换。

比较有名的多表代换密码有维吉尼亚（Vigenere）、博福特（Beaufort）、滚动密钥（running-key）和弗纳姆（Vernam）等。

1．维吉尼亚密码

一种以移位代换为基础的周期代换密码，为 1858 年法国密码学家维吉尼亚提出。

首先构造一个维吉尼亚方阵。

表 2—1—2　维吉尼亚方阵

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

它的基本阵列是 26 **行、**26 **列的方阵。方阵的第一行是** a **到** z **按正常顺序排列的字母表，第二行是第一行左移循环一位得到的，第三行又是第二循环左移一位得到的，依此类推，得到其余各行。然后在基本方阵的最上方附加一行，最左侧附加一列，分别依序写上** a **到** z **共** 26 **个字母，表的第一行与与附加列上的的字母** a **相对应，表的第二行与附加列上的字母** b **相对应，…，最后一行与附加列上的字母** z **相对应。如果把上面的附加行看作是明文序列，则下面的** 26 **行就分别构成了左移** 0 **位，**1 **位，**2 **位…，**25 **位的** 26 **个单表代换加同余密码的密文序列。加密时，按照密钥字的指示，决定采用哪一个单表。例如，密钥字是** bupt，**加密时，明文的第一个字母用与附加列上字母** b **相对应的密码表进行加密，明文的第二个字母用与附加列的字母** u **相对应的密码表进行加密，依此类推。**

下面我们用一个实例来说明维吉尼亚密码加密和解密的原理。设密钥字是 encryption，**待加密的明文是** public key distribution 。**由于密钥字比明文短，所以要重复书写密钥字，以得与明文等长的密钥序列。**

表 2—1—3 **维吉尼亚密码加密解密举例**

密钥字	e	n	c	r	y	p	t	i	o	n	e	n	c	r	y	p	t	i	o	n	e	
明 文	p	u	b	l	i	c	k	e	y	d	i	s	t	r	i	b	u	t	i	o	n	
密 文	T	H	D	C	G	R	D	M	M	Q	M	F	V	R	G	Q	N	B	W	B	R	

现在按上表对明文进行加密代换。第一个密钥字母是 e，**对第一个明文字母** p **进行加密时，选用左边附加列上的字母** e **对应的那一行作为代替密码表，查出与** p **相对应的密文字母是** T，**第二个密钥字母是** n，**用附加列上字母** n **所对应的一行作为代替密码表，与明文** u **进行替换，对应的密文是** H。**同理，将所有的密文字母替换完毕就可以得到表** 2.2 **中所示的密文** THDCGRDMMQMFVRGQNBWBR。**由于在上述加密变换中，明文的每个字母是按照密**

钥字的指示，选用不同的加同余密码替换而成的。因此同一个字母在明文中的位置不同，对应的密文字母就不同。例如，上例明文中有四个 i，但分别对应密文字母 G、M、W。反之，密文中的三个相同字母 M 却分别对应明文中的三个不同字母 e、y、i。 在这种体制中，所用到的单表数目与所用的密钥字长有关，并按密钥字长对这些单表轮流使用，因而可呈现固定的周期性，周期长度就是密钥字长。显然，密钥字越长，破译时就越困难。

维吉尼亚密码加密和解密的数学模型可以表示为：

令英文字母 a，b，…，z 对应于从 0 到 25 的整数。设明文是 n 个字母组成的字符串即：

$$m = m_1 m_2 \cdots m_n$$

密钥字周期性地延伸就给出了明文加密所需的工作密钥

$$k = k_1 k_2 \cdots k_n \qquad E(m) = C = c_1 c_2 \cdots c_n$$

$$\text{加密：} \quad c_i = m_i + k_i \mod 26 \qquad (2\text{—}1\text{—}9)$$

$$\text{解密：} \quad m_i = c_i - k_i \mod 26 \quad i=1，2，3\cdots，n \qquad (2\text{—}1\text{—}10)$$

类似可构造博福特加密算法如下：

$$E(m) = C = c_1 c_2 \cdots c_n$$

$$c_i = k_i - m_i \mod 26，\ i=1，2，\cdots，n \qquad (2\text{—}1\text{—}11)$$

§2.2 密码分析初步

一、概述

密码分析是截收者在不知道解密密钥及通信者所采用的加密体制的细节条件下，对密文进行分析，试图获取机密信息。研究分析解密规律的科学称作密码分析学。密码分析在外交、军事、公安、商业等方面都具有重要作用，也是研究历史、考古、古语言学和古乐理论的重要手段之一。

密码设计和密码分析是共生的、又是互逆的，两者密切有关但追求的目标相反。两者解决问题的途径有很大差别。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、语言学等知识外，还要靠经验、统计、测试、眼力、直觉判断能力……，有时还靠点运气。密码分析过程通常包括：分析（统计截获报文材料）、假设、推断和证实等步骤。

破译或攻击（break 或 attack）密码的方法有穷举破译法（exhaustive attack method）和分析法两类。穷举法又称作强力法(brute force method），这是对截收的密报依次用各种可解的密钥试译，直到得到有意义的明文；或在不变密钥下，对所有可能的明文加密直到得到与截获密报一致为止，此法又称为完全试凑法(complete trial-and-error method)。只要有足够多的计算时间和存贮容量，原则上穷举法总是可以成功的。但实际中，任何一种能保障安全要求的实用密码都会设计得使这一方法在实际上是不可行的。最著名的用穷举法攻击密码的例子是对美国数据加密标准（DES）的破译研究，这将在第三章中提到。

为了减少搜索计算量，可以采用较有效的改进试凑法。它将密钥空间划分成几个（例如，q

个）等可能的子集，对密钥可能落入哪个子集进行判断，至多需进行 q 次试验。在确定了正确密钥所在的子集后，就对该子集再进行类似的划分并检验正确密钥所在的集。依此类推，就可最终判断出所用的正确密钥了。关键在于如何实现密钥空间的等概子集的划分。

分析破译法有确定性的和统计的两类。

确定性分析法是利用一个或几个已知量（比如，已知密文或明文—密文对）用数学关系式表示出所求未知量（如密钥等）。已知量和未知量的关系视加密和解密算法而定寻求这种关系是确定性分析法的关键步骤。例如，第六章中讨论的 n 级线性移存器序列作为流的流密码，就可在已知 2n 密文 bit 下，通过求解线性方程组破译。

统计分析法是利用明文的已知统计规律进行破译的方法。密码破译者对截收的密文进行统计分析，总结出其间的统计规律，并与明文的统计规律进行对照比较，从中提取出明文和密文之间的对应或变换信息。

密码分析之所以能够破译密码，最根本的是依赖于明文中的多余度，这是 Shannon1949 年用他所开创的信息论理论第一次透彻地阐明的密码分析的基本问题。有关保密学的信息理论将在本章第 3 节中介绍。

破译者通常是在下述三种条件下工作的:

（1）唯密文破译（ciphertext only attacks），分析者从仅知道的截获密文进行分析，得出明文或密钥。

（2）已知明文破译（know plaintext attacks），分析者除了有截获的密文外，还有一些已知的明文一密文对（通过各种手段得到的）可供利用。

（3）选择明文破译（chosen plaintext attacks），分析者可以用他所选择的任何明文，在同一未知密钥下加密得到相应的密文。也就是说，分析者可以选定任何明文—密文对来进行攻击，以确定未知的密钥。

密码分析的成功除了用上述的数学演绎和归纳法外，还要利用大胆的猜测和对一些特殊或异常情况的敏感性·例如，若幸运地在两份密报中发现了相同的码字或片断，就可假定这两份报的报头明文相同·又如，在战地条件下，根据战事情况可以猜测当时收到的报文中某些密文的含义，如“攻击”或“开炮”等等。依靠这种所谓“可能字法”，常常可以幸运地破译一份报文。

一个保密系统是否被“攻破”，并无严格的标准。如果不管采用什么密钥，敌手都能从密文迅速地确定出明文，则此系统当然已被攻破，这也就意味着敌手能迅速确定系统所用的密钥。如果对大部分密钥而言，敌手都能从密文迅速地确定出明文，该体制也可说已被攻破。但破译者有时也可能满足于能从密文偶然确定出一小部分明文，虽然此时保密系统实际上并未被攻破，但部分机密信息已被泄露。

密码史表明，密码分析者的成就似乎远比密码设计者的成就更令人赞叹，许多开始时被设计者吹为“百年或千年难破”的密码，没过多久就被密码分析者巧妙地攻破了。在第二次世界大战中，美军破译了日本的“紫密”，使得日本在中途岛战役中大败。一些专家们估计，同盟军在密码破译上的成功至少使第二次世界大战缩短了 8 年。

二、语言的统计特性

任何一种语言都有其内在的规律性。如果取一本非专门性书籍，统计足够长的课文就会发现，字母（或字符）出现的频度会反映出相应语言的统计持平。以英语为例，在英语的 26 个字母中，各字母出现的频率是大不相同的，对各类不同资料的大量统计分析表明，各字母出现的频率是稳定的，并且完全可以推测，表 2—2—1 给出了各字母出现的概率值。

表 2—2—1　英文字母出现的概率

a	0.0856	g	0.0199	m	0.0249	s	0.0607	y	0.1999
b	0.0139	h	0.0528	n	0.0707	t	0.1045	z	0.0008
c	0.0279	i	0.0627	o	0.0797	u	0.0249		
d	0.0378	j	0.0013	p	0.0199	v	0.0092		
e	0.1304	k	0.0420	q	0.0012	w	0.0149		
f	0.0289	l	0.0339	r	0.0677	x	0.0017		

由表可见，字母 e 出现的概率最高，而 z 出现的概率最低，依据各字母出现概率大小的不同，可以将 26 各英文字母划分为五组，如表 2—2—2 所示。

表 2—2—2　英文明文字母按出现概率大小分组表

1。	e（>0.1）
2。	t，a，o，i，n，s，h，r（0.05～0.1）
3。	d，l（0.03～0.05）
4。	c，u，m，w，f，g，y，p，b（0.01～0.03）
5。	v，k，j，x，q，z（<0.01）

由于字母 e 的出现概率最高，因此，在单表密码中，只要密文具有适当长度，那么出现概率最高的字母几乎肯定就是字母 e 的密文等价字母。

在英语中，英语中常出现的双字母和三字母组合，也是对单表密码进行密码分析的有力手段。

在英语中，下面 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

出现的概率要比其它的双字母高得多，有此，可从密文字母出现的概率来推测它们的等价双字母。此外，象 oo，ee 这种双字母，如在密文中出现相邻的相同字母，则更易推断它们等价的明文双字母。

在英语中，下面 12 个三字母组合：

the，ing，and，her，ere，ent，tha，nth，was，eth，for，dth

出现的概率很高，尤以 the 最为突出，这些三字母组合对密码分析也有重要的启示作用。

还有，英语单词以 e，s，t，d 结尾的超过一半，以 t，a，s，w 为起始字母的约为一半。此外，语言中的某些常用用法，有些也会给密码分析者提供有价值的线索，如信的开头写 Dear，信的结尾写 Sincerely Yours。第二次世界大战期间，德国人喜欢用一些具有“爱国精神”的词汇作为密码表的密钥，又喜欢用德国格言试拍新电，这为法国密码分析者最终破译德国的 ADFGX 密码表提供了帮助。

三、单表密码分析举例

单表密码的最大特点是明文和密文字母之间的一一代替关系。这就使得明文中的一些固有特性和规律不可避免地反映到密文中去。例如明文是语言，则语言的各种统计特性都会在密文中体现。

例：仅知下列密文是密钥词组密码.

YKHLBA JCZ SVIJ JZB TZVHI JCZ VHJ DR IZXKHLBA VSS RDHEI DR YVJV LBXSKYLBA YLALJVS IFZZXC CVI LEFHDNZY EVBTRDSY JCZ FHLEVHT HZVIDB RDH JCLI VCI WZZB JCZ VYNZBJ DR ELXDZSZXJHDBLXI JCZ XDEFSZQLJT DR JCZ

RKBXJLDBI JCVJ XVB BDP WZ FZHRDHEZY WT JCZ EVXCLBZ CVI HLIZB YHVEVJKXVSST VI V HZIKSJ DR JCLI HZXZBJ YZNZSDFEZBJ LB JZXCBDSDAT E-VBT DR JCZ XLFCZH ITIJZEI JCVJ PZHZ DBXZ XDBILYZHZY IZXKHZ VHZ BDP WHZVMVWSZ

这段密文含有 338 个字母，按出现次数由多到少的顺序排列，依次为 Z，J，V，B，H，D，I，L，C，X，S，Y，E，R，T，F，K，A，W，N，P，M，Q，U，G，O。由于 Z 出现的次数(45 次)显著高于其它字母，推断 Z 对应明文字母 e。

排在第二位和第三位的字母依次是 J 和 V，所以这两个字母中的一个可能对应明文字母是 t。注意到 JCZ 这三个字母单词出现 8 次，这使我们断定 JCZ 对应明文 the，因此 J→t;C→h。

由于单字母字 V 的出现，我们可知它对应 a 或 i。由于 a 和 i 在英语中出现的频率差不多，所以我们还要借助其它信息来做进一步判断。注意到单词 JCVJ，如果 V 对应 i，则这个单词译成明文就是 thit，因为英文中没有这个单词，我们可以断定 V 代表 a。

再回到密文中进行观察，我们注意到 B 出现的频率比较高，这样，我们初步断定 B 对应表 2.2.2 中第二组 o，i，n，s，r 中的一个。研究单词 JZB，我们已知 J、Z 分别对应 t、e，对 B 进行各种可能性的试探，得到明文单词 teo，tei，ten，tes，ter。显然明文是 ten，即 B 对应 n。再看两字母单词 VI，由于 V 是 a，I 必定对应 s 或 n。但我们已知 B 对应 n，这就意味着 I 必为 s，即与密文单词 VI 对应的是 as。

密文单词字母中出现频率仅次于 B 的是 H、D、L，我们猜想这三个字母可能以某种顺序对应 o、i、r。 观察密文单词 VHJ，迫使 H 对应字母 r，即 VHJ→art。而 JCLI 迫使 L 对应 i，即 JCLI 的明文是 this。这样 D 可能对应 o。由密文单词 VSS，很容易猜想 S 对应 l，即 VSS 的明文是 all。在两字母单词 DR 中，由于 D 代表 o，故 DR 可能代表 or on of ok 中的一个。or 意味着明文字母 r 与自身相对应，显然不和理。由于 B 代表 n，所以也不是 on。那么 R 必代表 f 或 k。如果代表 k，则 RDH 译为 kor，不合理，故 R 代表 f，即 DR→of。

至此，我们可以构成下列明文。密文对照表：

表 2.2.3

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	V				Z	R						S		B	D			H	I	J						

从表中可以明显看出这个密钥词组密码所用的特定字母是 f。密文字母表中 V 与 Z 之间有三个间隔，而明文字母表中它们之间又仅有三个字母，故易推知 WXY 分别代表 bcd。继续发现 D 和 H 之间有两个间隔，并可判定 EFG 中的两个必为 p 和 q。再看两字母单词 WT，由于 W 是 b，故可推知 T 代表 e 或 y，由于 Z 代表 e，可以判定 T 对应 y，即 WT 对应 by。填入上表，立即可以断定 U 代表 z〈见表 2.2.4〉。

表 2.2.4

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	V	W	X	Y	Z	R		C	L			S		B	D			H	I	J					T	U

现在，密文单词 RDHEI 可译为 for?s。据此，我们猜想 E 代表 m。再看 EVBT，可译成 many，合理，故可断定 E 就是 m。那么 F 和 G 必然代表 p 和 q，再看 YLALJVS，可译成 dig?ital，由此猜想 A 代表 g，则 JZXCBDSDAT 译成 technology 完全合理，故确定 A 对应 g〈见表 2.2.5〉。

表 2.2.5

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	V	W	X	Y	Z	R		C	L			S	E	B	D	F	G	H	I	J					T	U

现在我们试图译出第一个单词 YKHLBA 得到 d?ring。显然?必定表示元音，因为我们已知

a，e，i，o，u 的密文。于是它必定为 u，即 K 代表 u。稍加思考，可以从 BDP 推知 P 对应 w，即 BDP→now，则 Q 必定对应 x。把 LEFHDNZY 译成 impro?ed 可猜想到 N 代表 v。再把最后一个单词 WHZVMVWSZ 译成 brea?able 可断定 M 代表 k，则剩下的最后一个密文字母 O 必然代表 j。

至此，我们已经得到了一个完整的明密文对照表（见表 2.2.6）。

表 2.2.6

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

根据上表，我们可以推知密钥词组为 RACAL COMSEC，而特定字母为 f。由此译出的消息如下：

during the last ten years the art of securing all forms of data including digital speech has improved manyfold primary reason for this has been the advent of microelectronics the complexity of the function that can now be performed by the machine has risen dramatically as a result of this recent development technology many of the cipher system that were once considered secure are now breakable

在破译上述密文时，我们仅用了很少一点统计学知识，更多地应用了英文知识，因为我们知道明文字母的顺序没有变，知道单词的长度，还知道哪几个顺序排列的字母构成一个单词。这些单词在破译上述密码时，给了我们很大帮助，也从相反方向提醒密码设计者，必须从内容到形式对明文进行全面伪装。

至于多表代换密码，原来明文中的这些特性通过多个表的平均作用隐蔽起来了，因此它的破译比单表代换要难。但是多表代换中的平均结果，会使密文的统计特性与明文的统计特性明显不同，随着多表代换周期的加大，这中差别也就更加明显，从此入手，就可以破译多表代换密码。对此不过多介绍，有兴趣的同学可以参考有关书籍。

§2.3 密码学信息理论基础

一、不确定性的度量——熵的概念

从密码学分析可以看出，明文毫无不确定性可言的。密文则不然，随着分析的进行，不确定性程度逐渐减小，最后完全确定。不同的密码，它们的强度也不一样。如何确定不确定性的程度呢？下面引进熵的概念，他是有商农（Shannon）提出的，他的信息论是密码学的一个理论基础。

设有概率为 p 的事件，p=1 时是确定事件，它没有任何不确定的因素，但随着 p 的减小，不确定性成分随之增大。从信息的角度来看，如果获得了确定性事件的信息，这个信息就没有价值，或者说它的信息量就等于零。比如说：“我出门看见太阳从东方升起”，没有人会对这句话感兴趣，因为他从中得不到任何新东西，但是如果说：“我看见不明飞行物”，则完全不一样。所以，不确定性越大，信息量也就越大，反之则越小，而确定性事件的信息量为零。

如何度量不确定性呢？给定一离散集合 $X=\{x_i, i=1, 2, \cdots, n\}$，令 x_i 出现的概率为 $p(x_i)$ $\geqslant 0$，且 $\sum_{i=1}^{n} p(x_i)=1$，事件 x_i 出现给出的信息量定义为

$$I(x_i) = -\log_a p(x_i) \qquad (2—3—1)$$

它表示了事件 x_i **出现的可能性大小，也是为事件** x_i **出现所必须付出的信息量，通常** a=2，**即采用以** 2 **为底的对数，相应的信息单位称作比特**（bit）。**将集合** X **中事件出现的信息量的统计平均值**

$$H(X) = -\sum_i p(x_i)\log_2 p(x_i) \geqslant 0 \qquad (2—3—2)$$

定义为集合 X **的熵**（entropy）。**它表示集合** X **中出现一个事件平均给出的信息量，或集合** X **中事件的平均不确定性**（average uncertainly），**或为确定集合** X **中一个事件必须提供的信息量。**

例 2—3—1 对某一地区进行多年的观察，5 **月份雨天的概率为** 0.4，**晴天的概率为** 0.6，**而** 11 **月份晴的概率为** 0.15，**雨天的概率为** 0.2。**设** A **为** 5 **月的天气，** B **为** 11 **月的天气，其熵分别为**

$$H(A) = -0.4\log 0.4 - 0.6\log 0.6 = 0.9694 \text{ bit}$$

$$H(B) = -0.65\log 0.65 - 0.15\log 0.15 - 0.2\log 0.2 = 1.2272 \text{ bit}$$

可以看出，11 **月的天气不确定性比较大一些。**

熵有如下的性质：对有 n **个事件的集合** X，**当** $p_1 = p_2 = \cdots p_n = \frac{1}{n}$ **时，**H（X）**的值达到最大。日常生活中的许多事情可以验证这个结论，举一个很简单的例子：如果一场足球比赛双方水平越接近，那么其最终结果的不确定性越大。**

二、暧昧度

设事件有两个事件集 $X = \{x_i, i = 1, \cdots, n\}$ **和** $Y = \{y_j, j = 1, \cdots, m\}$ **。则联合事件集** XY={ $x_i y_j$ i=1，…，n，　j=1，…，m}。**令联合事件** $x_i y_j$ **的概率为** $p(x_i y_j)$，**则有**

$$\sum_{i,j} p(x_i y_j) = \sum_i p(x_i)\sum_j p(y_j \mid x_i) = \sum_j p(y_j)\sum_i p(x_i y_j) = 1 \qquad (2—3—3)$$

类似于（2—3—2）**式有**

$$H(X) = -\sum_i p(x_i)\log_2 p(x_i) \qquad (2—3—4)$$

$$H(Y) = -\sum_j p(y_j)\log_2 p(y_j) \qquad (2—3—5)$$

$$H(XY) = -\sum_{ij} p(x_i y_j)\log_2 p(x_i y_j) \qquad (2—3—6)$$

称 H（XY）**为集** X **和** Y **的联合熵。可以证明**

$$H（XY）= H（X）+ H（Y|X）$$

$$=H（Y）＋H（Y|X）$$

其中， $H(X\mid Y)=-\sum_{ij}p(x_iy_j)\log_2 p(x_i\mid y_j)$

$$H(Y\mid X)=-\sum_{ij}p(x_iy_j)\log_2 p(y_j\mid x_i)$$

称为（给定了 Y 的条件下 X 的或者给定 X 的条件下 Y 的）条件熵，也叫暧昧度。
若 M 为明文空间，C 为密文空间，在截取密文 $c\in C$ 的条件下，明文 M 的条件熵为

$$H(M\mid C)=-\sum_{m\in M}p(m\mid c)\log_2 p(m\mid c) \qquad （2—3—7）$$

现取一段恺撒密码的密文：

htsxnijwfgqj

它的可能明文有如下 26 个

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

若取一个字母进行判断，无疑 e 的概率最大，这时它的条件熵用 $H_1(M\mid C)$ 表之，有：

$$H_1(M\mid C)=-\sum_{x=a}^{z}p(x)\log p(x)=4.125\ \text{bit}$$

若取两个字母进行分析，设 $m=xh$，但 $p(m)=p(x)\,p(h\mid x)$

其中 $p(h\mid x)$ 为字母 ξ 后缀 η 的概率。如 $p(a)$，等一样，有统计结果（从略）。计算结果见

表：

am	0.00241	0.07341
bn	0	0
co	0.00637	0.19441
dp	0	0
lq	0.00044	0.01353
fr	0.00351	0.10717
gs	0.00051	0.01555
ht	0.00123	0.03755
iu	0.00007	0.00211
jv	0	0
ku	0	0
lx	0	0
my	0.00048	0.01459
nz	0.00003	0.00086
oa	0.00068	0.02068
pb	0	0
qc	0	0
rd	0.00191	0.05827
se	0.01090	0.33255
tf	0.00007	0.00233
ug	0.00095	0.02903
vh	0	0
wi	0.00314	0.09568
xj	0	0
yk	0.00007	0.00207
zl	0.00001	0.00031

可算得

类似可得

即截获 5 **个字符时便可确定明文，现将** *n*=3， 4， 5 **的计算结果列为表，括号里的即为。**

m	n=3	n=4	n=5
amlqg	0.00002(0.00717)	0	0
bnmrh	0	0	0
consi	0.00139(0.49821)	0.00010(0.29412)	0.00001(1.00000)
dpotj	0	0	0
pqpuk	0	0	0
frqvl	0	0	0
gsrwm	0	0	0
htsxn	0.00005(0.01792)	0	0
iutyo	0.00001(0.00358)	0	0
jvuzp	0	0	0
kwvaq	0	0	0
lxwbr	0	0	0
myxcs	0	0	0
nzydt	0	0	0
oazeu	0	0	0
pbafv	0	0	0
qcbgw	0	0	0

rdchx	0.00001(0.00358)	0.00001(0.02941)	0
sediy	0.00130(0.46595)	0.00023(0.67847)	0
tfejz	0.00001(0.00358)	0	0
ugfka	0	0	0
vhglb	0	0	0
wihmc	0	0	0
xjind	0	0	0
ykjoe	0	0	0
zlkpf	0	0	0

从以上可知，随着 *n* 的增加不确定性随之减小。*n*=1 时 *e* 的概率最大。*n*=2 时，暧昧度降至 2.9497 比特，以 *se* 的出现概率最大。*n*=3 时 *con* 和 *sed* 的概率分别为 0.49821 和 0.46595。*n*=4 时情况仍不明朗。*n*=5 时，不确定性消失，即明文应为 considerable 。

三、商农(Shannon)理论

（1）随机密码是指假定:

（a） 密钥空间所有密钥均匀分布，即每个密钥概率相等。

（b） 长度为 *n* 的字符串有两部分组成，一部分是有意义的，另一部分是无意义的。

（c） 有意义的部分的字符串出现的概率也是相等的。

英文字母 26 个，由它们构成 *n* 位字符串的数目为 $26^n = 2^{rn}$，$r = \log_2 26 = 4.7004$，**其中有意义的一类其数目设为** $2^{a_n n}$，**则传输长度为 *n* 的字符串为有意义的明文的概率为:**

$$P = 2^{a_n n} / 2^{rn} = 2^{-(r-a_n)n} = 2^{-d_n n}$$

其中 $d_n = g - a_n$

称为语言的冗余度。

随机密码假设 *n* 位字符串中有意义的部分出现的概率是相等的。所以，任取密文 *c*，随机地选择一密钥对 *c* 解密，得到有意义明文的概率为 $2^{-d_n n}$，**因此，冗余度愈大获得有意义的明文的概率愈小。**

（2）唯一解码量

商农的贡献在于他提出了以冗余度作为密码分析的基础。从此，进一步对具有某种冗余度的明文定量给出破译密文所需的字母数目，他称之为唯一解码量。假定密钥空间 $K = \{k_1, k_2, \cdots kN_k\}$ **中每一个密钥都是等概率的，这样每一个密钥的概率为:**

$$P=1/N$$

$$H(K) = -N_k \frac{1}{N_k} \log_2 \frac{1}{N_k} = \log_2 N_k$$

所以，用所有的密钥进行脱密可能得到有意义的明文的期望值等于 $2^{H(K)} 2^{-d_n n} = 2^{H(K)-d_n n}$。**若 *H*(*K*)比 $d_n n$ 大，则获得有意义译文的数目也多。当 *H*(*K*) = $d_n n$ 时，有意义的明文正好只有一个，这个 *n* 就称之为唯一解码量，用 Ud（unicity distance）表示，即:**

$$Ud \bullet d_n = H(K)$$

Ud **给出了破译密码所需的最少密文字符数，也就是确定密钥所需的最小字符数。**
例如对于词组密钥密码，密钥的数目为 26!

$$H(K) = -(26!)\frac{1}{26!}\log_2\left(\frac{1}{26!}\right)$$

$$= \log_2(26!)$$

=88.382bit

$|A|=26$，$r=\log_2(26)$ =4.7004

至于有意义的明文数目，有多种不同的估计。其中最简单的一种是这样的，当明文长度 n **为充分大时，** 26 **个英文字母出现的数目设为** n_a，n_b，…，n_z。**若有意义的明文概率为** P，**则：**

$P = P_a^{n_a} P_b^{n_b} \cdots P_z^{n_z}$，**其中** P_a，P_b，$\cdots P_z$ **分别是英文字母** a，b，…z，**出现的概率，**
而且有：

$$n_a = nP_a,\ n_b = nP_b,\ \cdots n_z = nP_z$$

$$\therefore P = (P_a^{n_a} P_b^{n_b} \cdots P_z^{n_z})^n$$

令长度为 n **的有意义的明文数目为** S，**根据假定它们出现的概率相等，即：**

$$P = \frac{1}{S}, S = \frac{1}{P}$$

$$\log_2 S = -\log_2 P = -n(P_a \log_2 P_a + P_b \log_2 P_b + \cdots + P_z \log_2 P_z)$$

依据统计结果:

$$\sum_{x=a}^{z} P(x)\log_2 P(x) = -4.192$$

$$a_n = 4.192$$

但

$$g_n = 4.7004$$

$$d_n = g_n - a_n = 4.7004 - 4.192 = 0.5084$$

$$Ud = H(k)/d_n = 88.382/0.5084 = 174$$

也就是说对于词组密钥密码系统来说，唯一解码量为 174 **个字符。**
唯一解码量的估计归根结底和有意义明文数目的估计相关。上面的例子是假定:

$$P = P_a^{n_a} P_b^{n_b} \cdots P_z^{n_z} = (P_a^{n_a} P_b^{n_b} \cdots P_z^{n_z})^n$$

没有考虑到前后缀的关系。比如对于 2-字母组，

$$m = m_1 m_2 m_3 m_4 \cdots m_{2n-1} m_{2n}$$

出现概率为

$$P(m) = P(m_1 m_2) P(m_3 m_4) \cdots P(m_{2n-1} m_{2n})$$

唯一解码量只是理论地估计到至少需要多少个密文字母可以确定明文，但没有解决如何确定，需要多少个计算量。

（3）假定 $p(m)$表示明文 m 被发送的概率，同样 $p(c)$为收到密文 c 的概率，$p(c|m)$为发送明文 m，收到密文 c 的概率，依此同样理解 $p(m|c)$。根据概率乘法定理有：

$$p(m|c)\ p(c) = p(c|m)\ p(m)$$

所谓的完全保密指的是满足等式

$$p(m|c) = p(m)$$

的密码系统，直观上表示截获密文 c 对确定明文无帮助，不难推知：

$$p(m|c)\ p(c) = p(m)\ p(c) = p(c|m)\ p(m)$$

$$\therefore \quad p(c) = p(c|m)$$

实际上是完全保密的充要条件。

什么样的密码系统才是完全保密的？可以从 $p(m|c) = p(m)$和 $p(c) = p(c|m)$这两个条件来观察。

假如明文空间$M = \{m_1, m_2, \cdots m_p\}$，密文空间$C = \{c_1, c_2, \cdots c_q\}$，密钥空间$K = \{k_1, k_2, \cdots k_r\}$，给定$c \in C$，但

$p(m|c) = p(m)$

说明 C 空间里的任何一个密文，都可以在密钥空间 K 里找到一个密钥将它解密为明文 m，而且机会均等。同样从 $p(c|m) = p(c)$可以说明任何一个明文，都可以找到一密钥 k 将它加密成密文 c，而且机会也均等，如若不然，若$m \in M$，不存在$k \in K$，将 m 加密成 c。已知 c，便可以断定 $p(m|c) = 0$，和 $p(m|c) = p(m)$的假定矛盾。这样密文 c 对判定 m 不是没有帮助，恰恰相反，破译者便排除了 m 的可能性，提高了破译可能性。换一句话说，完全密码系统要求密钥量至少应该和明文一样多。

§2.4 现代密码中计算复杂性理论基础

一个密码系统的安全性可以通过破译该系统的最好算法的计算复杂度来度量，因而计算复杂性理论已成为现代密码学的基础。本节仅对密码学中经常遇到的计算复杂性理论作一简单介绍。

计算复杂度中的数学问题由下面两部分构成：（1）关于该问题的一个全面描述和有关的参量；（2）所求的解的一个明确的表达。一个问题的实例则是由一般问题的参量赋予特殊的值而得到。一个算法是用以解决某个问题的一串步骤或计算程序。任何算法处理的都是实例而不是问题本身，因而称一个算法解决了一个问题是指该算法对给定问题的任何实例 I，都给出答案。

对一个问题 π 和一个求解 π 的算法 α，问题 π 的实例 I 的输入长度是实例 I 的参数经编码后产生的输入序列的元素数目，有时称输入长度为问题 π 的输入规模。复杂性理论就是处理算法难度的分类。算法难度是指执行一个算法所耗费的资源量的一个测度，有四个基本

资源量：1）基本操作的数目；2）所耗费的时间；3）需要的存贮空间；4）需要的硬件数量。一个算法的复杂度就是用输入长度 n 描述的关于某种复杂度测度所耗费资源当 $n\to\infty$ 时的渐近行为。在复杂性理论中问题是依据所知的解决问题的最有效算法的复杂度来分类的。一个算法可以对某些实例很快得到答案而对另一些实例需更多努力才能得到答案，尽管输入长度一样。所以实际上有两个复杂度；一个是平均情况的行为，另一个是最坏情况的行为。一个算法的复杂度就采用它的最坏行为表示的渐近行为。

通常将运算时间可表示为输入长度的多项式函数的算法称为好算法，而将运行时间需输入长度的指数函数的算法称为坏算法。具有相同复杂度的算法归为等价类，两个更广的类是 P 类和 NP 类。属于 P 类的算法是具有最坏复杂度为输入长度的多项式时间阶的算法。若一个算法中每一步计算中，下一步是唯一确定的，则称该算法为确定性的，属于 P 类的算法都是确定性的，并且是 P-时间受限的，即执行时间是多项式时间。若一个算法，在它的某个计算步骤必须从一个有限的可选择中选一个作为下一步，则该算法称为非确定性的。一个非确定性图灵机（NDTM）是一个无限制的并行机器，在每一个判决步骤，它首先按可选择的数目来复制自己，并对各个选择并行独立地进行计算。而 NP 类则由所有能够由一个 NDTM 在 P-时间内计算的问题构成。此处在 P-时间内计算表示 NDTM 的一个处理器在 P(n)步运算后达到停机状态，其中 n 为输入长度，而步长为描述选择分支的判决树的深度。尽管确定性地判定一个 NP 一问题的复杂度通常是非多项式的，而验证一个解则只需多项式时间。人们通常把找不到好算发或一直未找到好算法的问题都称为难解问题。

在研究问题计算复杂度时，可通过比较将问题按复杂程度分类，P 类问题包括在 NP 类问题中，但 P=NP？是计算复杂性理论中迄今尚未解决的一个大问题。NP 类中存在一个子类，它具有下面性质：若子类中的某个问题具有确定性的 P-时间算法，则 NP 类中的所有问题都存在一个确寇性的 P-时间算法，即 P=NP，这一子类问题称为 NPC 问题，易见这是 NP 中最难的一类问题。如果 P≠NP，则称不属于 P 类又不是 NPC 的问题为 NPI 问题。在 NP 类中，所有问题的补问题，即求问题的互补答案的所有问题统称为 CO 一 NP 类。

在现代密码中，一个密码系统的破译常常可归结为求解某个数学问题，数学问题的算法求解的复杂性可通过计算复杂性理论来描述，因此计算复杂性理论为破译密码的计算复杂度提供了实际的度量方法。而计算复杂性理论中的一些典型的数学问题又给人们提供了设计实用安全的高强度密码系统的基础。例如基于 NPC 类中的背包问题而设计的背包公开钥密码系统，基于 NPI 类的大数因子分解问题的 RSA 公开钥密码系统等。因此算法与问题的复杂性理论已成为现代密码系统设计与分析的重要基础。

第三章　数据加密标准

§3.1 数据加密标准简介

数据加密标准（Data Encryption Standard，DES）是一种对计算机数据进行密码保护的一种数学算法。它的产生被认为是本世纪 70 年代信息加密技术发展史上的两大里程碑之一。

由于本世纪 60 年代计算机得到了迅猛的发展，大量的数据资料被集中储存在大型计算机数据库中，并在计算机通信网中进行传输。其中有些通信具有高度的机密性，有些数据具有极为重要的价值，因此对计算机通信及计算机数据进行保护的需求日益增长。当时的美国虽然已经制订了数据保密措施，但是人们普遍认为这些保密措施只对业余人员有效。对于职业人员来说，要截收通信信号，要绕过报警系统并接通通信设备，要取出、复制、删除、插入或更改是不成问题的。针对这种情况，有人提出了两种对数据进行保护的方法。一种方法是对数据进行物理保护，即把重要的数据存放到安全的地方，如银行的地下室中；另一种方法是对数据进行密码保护。

由于普遍认为加密算法如果足够复杂的话，密码是一种有效的措施，所以美国国家标准局（NBS）于 1973 年 5 月发出通告，公开征求一种标准算法用于对计算机数据在传输和存储期间实现加密保护的密码算法。1975 年美国国家标准局接受了美国国际商业机器公司（IBM）推荐的一种密码算法，并向全国公布，征求对采用该算法作为美国信息加密标准的意见。经过两年的激烈争论，美国国家标准局于 1977 年 7 月正式采用该算法作为美国数据加密标准。1980 年 12 月，美国国家标准协会从（ANSI）正式采用这个算法作为美国的商用加密算法。

随着计算机和通信技术的进一步发展，跨国的大型计算机通信网的建立和扩大，人们对国际计算机通信网中数据的传输和存储进行保护的要求也越来越迫切，这就要求一个数据加密的国际标准。为此，国际标准化组织（ISO）信息处理系统技术委员会领导的工作组，经过几年的努力，于 1985 年形成了“数据加密算法 DEA一1”和“64 比特分组密码算法的工作方式”两个国际标准草案。实际上“数据加密算法 DEA一1 除了没有规定密钥分组中每个字节最后一个比特的作用外，其余和美国数据加密标准 DES 完全相同。

DES 是一种对称密码体制，它所使用的加密和解密密钥是相同的，是一种典型的按分组方式工作的密码。其基本思想是将二进制序列的明文分成每 64bit 一组，用长为 64bit 的密钥对其进行 16 轮代换和换位加密，最后形成密文。DES 的巧妙之处在于，除了密钥输入顺序之外，其加密和解密的步骤完全相同，这就使得在制作 DES 芯片时，易于做到标准化和通用化，这一点尤其适合现代通信的需要。在 DES 出现以后，经过许多专家学者的分析论证，证明它是一种性能良好的数据加密算法，不仅随机特性好，线性复杂度高，而且易于实现，加上能够标准化和通用化，因此，DES 在国际得到了广泛的应用。

§3.2 DES 加密、解密原理

DES 是典型的传统密码体制，它利用传统的换位和置换等加密方法，现介绍算法如下：

DES 主要包含三个部分。一个是密钥产生部分；一个是换位操作，即初始置换和逆初始置换部分，另一个是复杂的、与密钥有关的乘积变换部分。加密前，先将明文分成 64bit 的分组，然后将 64bit 二进制码输入到密码器中。密码器对输入的 64 位码首先进行初始置换，

然后在 64bit 主密钥产生的 16 个子密钥控制下进行 16 轮乘积变换，接着再进行末置换就得到 64 位已加密的密文。DES 算法的主要步骤如图 3—2—1 所示。

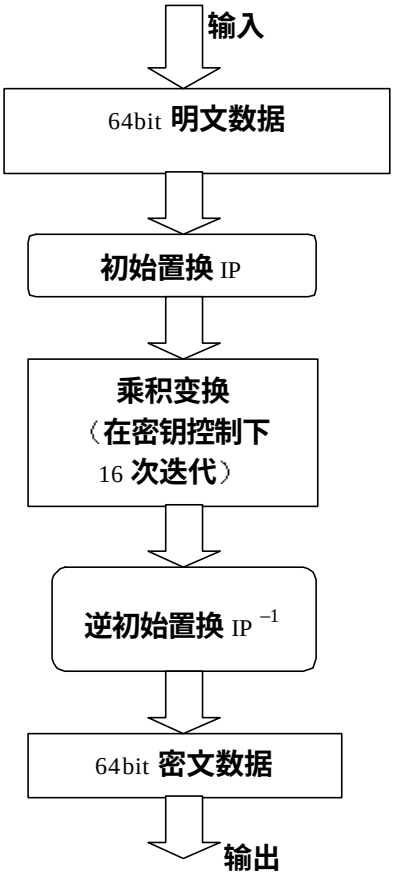


图 3—2—1 DES 算法框图

假定信息空间都是{0，1}组成的字符串，信息被分成 64bit 的块，密钥是 56bit。经过 DES 加密的密文也是 64bit 的块。设 m 是一个 64bit 的信息块，k 为 56bit 的密钥，即:

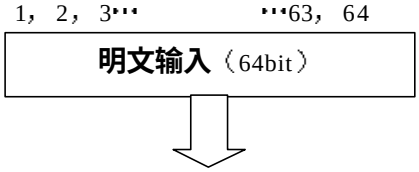
$m = m_1 m_2 \ldots m_{64}$　　$m_i = 1，2，\cdots，64$

$k = k_1 k_2 \ldots k_{64}$　　$k_i = 1，2，\cdots，64$

其中，k_8，k_{16}，k_{32}，k_{32}，k_{40}，k_{48}，k_{56}，k_{64} 是奇偶校验位。真正起作用的密钥仅 56 位。

（1）初始置换 IP

将 64 个明文比特的位置进行置换，得到一个乱序的 64bit 明文组，然后分成左右两段，每段为 32bit，以 L_0 和 R_0 表示，如图 3—2—2 所示。



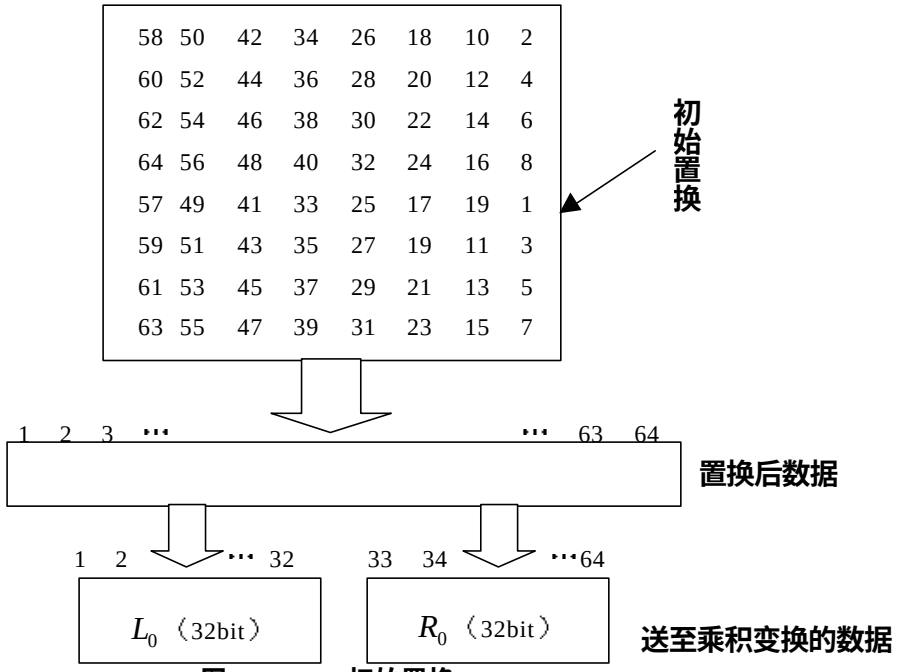


图 3—2—2 初始置换 IP

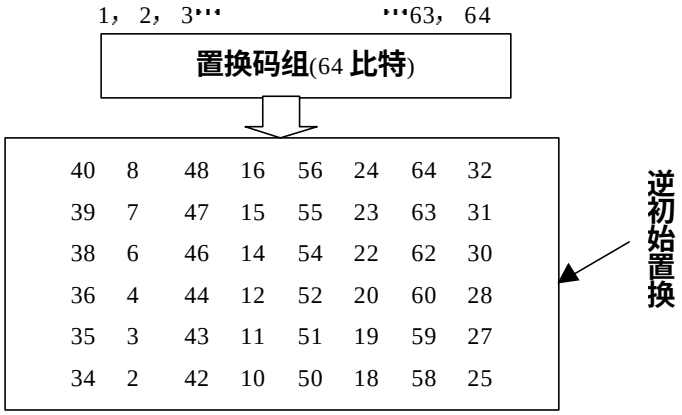
如图可知 IP 中各列元素位置号数相差为 8，相当于将原明文各字节按列写出，各列比特经过偶采样和奇采样置换后，再对各行进行逆序。将阵中元素按行读得的结果。

例如:输入 $m = m_1 m_2 \ldots m_{64}$

输出 $\mathrm{IP}(m) = m_{58} m_{50} m_{42} m_{34} \ldots m_{23} m_{15} m_7$

（2）逆初始置换 IP $^{-1}$

将 16 轮迭代后给出的 64bit 组进行置换，得到输出的密文组，如图 3—2—3 所示。输出为阵中元素按行读的结果。注意 IP $^{-1}$ **到中的第 58 位正好是 1，也就是说在 IP 的置换下第 58 位换为第一位，同样，在 IP** $^{-1}$ **的置换下，应将第 1 位换回第 58 位，依此类推。由此可见，输入组 m 和 IP** $^{-1}$ **（IP（m））是一样的。IP 和 IP** $^{-1}$ **在密码上的意义不大，它的作用在于打乱原来输入 *m* 的 ASCII 码字划分关系。**



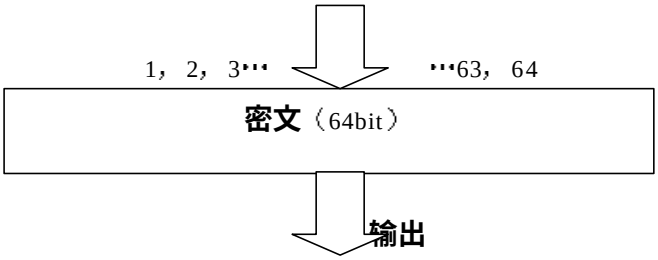


图 3—2—3 逆初始置换 IP $^{-1}$

(3) 乘积变换 *T*

它是 DES **算法的核心部分，如图** 3—2—4。**将经过** IP **置换后的的数据分成** 32bit **左右两组，在迭代过程中彼此左右交换位置。每次迭代只对右边的** 32bit **进行一系列的加密变换，在次轮迭代即将结束时，把左边的** 32bit **与右边的** 32bit **诸位模** 2 **相加，作为下一轮迭代时右边的段，并将原来右边的未经变换的段直接送到左边的寄存器中作为下一轮迭代时左边的段。在每一轮迭代时，右边段要经过选择扩展运算 *E*，密钥加密运算，选择压缩运算 *S*，置换运算 *P* 和左右混合运算。**

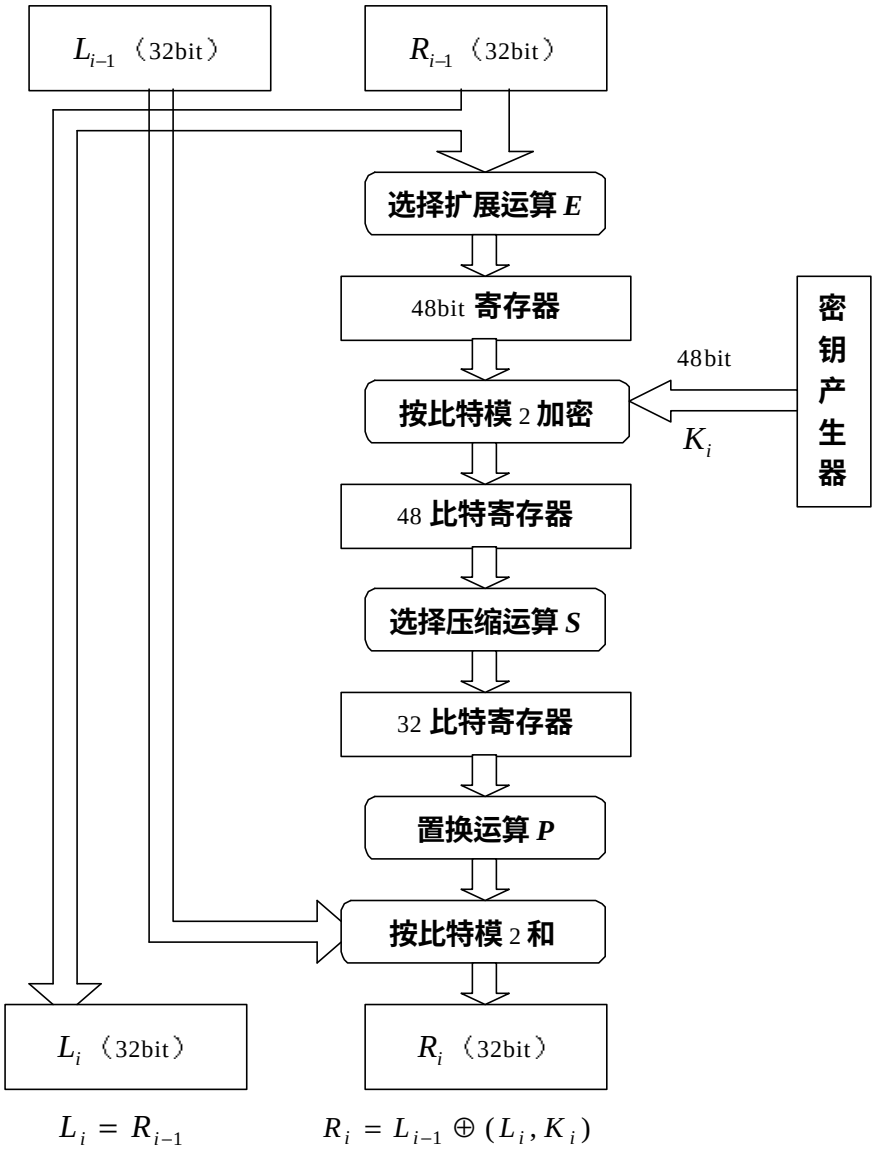


图 3—2—4 乘积变换框图

选择扩展运算 *E*　将输入的 32bit R_{i-1} 扩展成 48bit 输出，其变换表在图 3—2—5 中给出。

令 ***s*** 表示 ***E*** 的输入的下标，则 *E* 的输出将是对原下标 ***s*** ≡0 或 1（mod 4）的各比特重复一次得到的。即对原第 32，1，4，5，8，9，12，13，16，17，20，21，24，25，28，29 各位重复一次得到数据扩展。将表中数据按行读出即得到 48bit 输出。

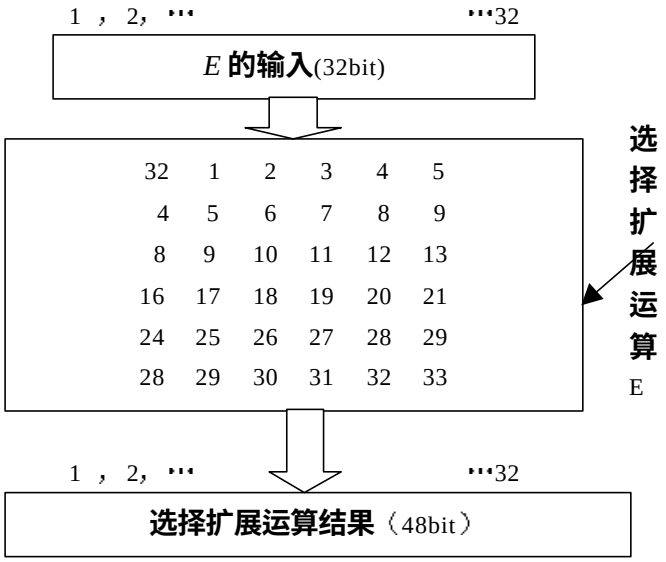


图 3—2—5　选择扩展运算 *E*

密钥加密运算　将子密钥产生器输出的 48bit 子密钥 K_i 与选择扩展运算 ***E*** 输出的 48bit 数据按位模 2 相加。

选择压缩运算 *S*　将前面送来的 48bit 数据自左至右分成 8 组，每组 6bit。然后并行送入 8 个 S—盒，每个 S—盒为一非线性代换网络，有 4 个输出。盒 S_1 至 S_8 的选择函数关系如表 3—2—1 所示。运算 S 的框图在 3—2—6 中给出。

表 3—2—1　DES 的选择压缩函数表

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	
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	S_1
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13	
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	S_2
3	13	8	10	1	3	15	4	2	11	6	7	42	0	5	14	9	
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	S_3
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	
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	S_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	
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	S_5
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	
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	S_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	
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	S_7
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	
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	S_8
3	2	1	14	7	1	10	7	13	15	12	9	0	3	5	6	11	

8 个 S 盒是将 6bit 的输入映射为 4bit 的输出。

以S_1盒为例说明它们的功能如下：

若输入为$b_1b_2b_3b_4b_5b_6$

其中b_1b_6两位二进制数表达了 0～3 之间的数，$b_2b_3b_4b_5$为四位二进制数表达 0～15 之间的某个数。在S_1表中的b_1b_6行$b_2b_3b_4b_5$列找到一数 m，$0 \leqslant m \leqslant 15$，若 m 用二进制表示为 $m_1m_2m_3m_4$ ，则$m_1m_2m_3m_4$ 便是它的 4bit 输出。

例如，输入为 001111，b_1b_6=01=1，$b_2b_3b_4b_5$=0111=7，即在S_1盒中的第 1 行第 7 列求得数 1，所以它的 4bit 输出为 0001。

又如对于S_2盒输入为 101011，则S_2盒中 3 行 5 列元素为 15，故输出为 1111

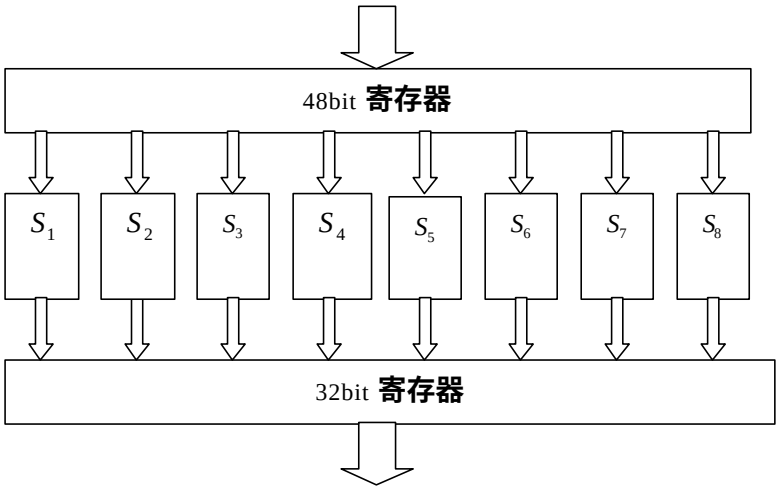


图 3—2—6　选择压缩运算 *S*

S 盒是 DES 的核心,也是 DES 算法最敏感的部分，其设计原理至今仍讳莫如深，显得非常神秘——所有的替换都是固定的，但是又没有明显的理由说明为什么要这样。有许多密码学家担心 D 美国国家安全局设计 *S* 盒时隐藏了某些“陷门”，使得只有他们才可以破译算法，但研究中并没有找到弱点。

美国国家安全局曾透露了 *S* 盒的几条设计准则:

1. 所有的 *S* 盒都不是它输入的线性仿射函数，换句话说，就是没有一个线性方程能将四个输出比特表示成六个比特输入的函数。
2. 改变 *S* 盒的 1 位输入，输出至少改变 2 位。这意味着 *S* 盒是经过精心设计的，它最大程度上增大了扩散量。
3. *S* 盒的任意一位输出保持不变时，0 和 1 个数之差极小。即如果保持一位不变而改变其它五位，那么其输出 0 和 1 的个数不应相差太多。

置换运算 ***P***　对S_1至S_8盒输出的 32bit 数据进行坐标变换。如图 3—2—7 所示。置换 ***P*** 输出的 32bit 数据与左边 32bit 即 R_{i-1} 诸位模 2 相加所得到的 32bit 作为下一轮迭代用的右边的数字段。并将R_{i-1}并行送到左边的寄存器作为下一轮迭代用的左边的数字段。

选择函数输出（32bit）

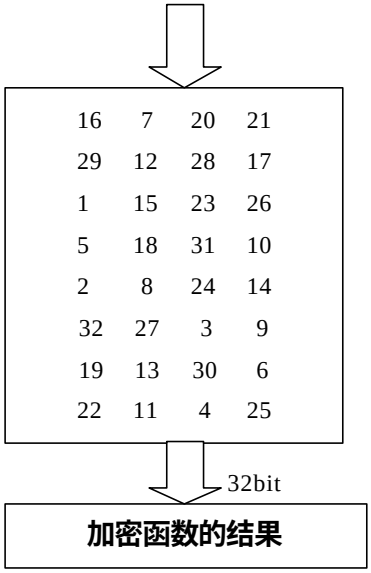


图 3—2—7　置换运算 *P*

子密钥产生器　　**将** 64bit **初始密钥经过置换选择** PC—1、**循环移位置换、置换选择** PC—2 **给出每次迭代加密用的子密钥** K_i，**参看图** 3—2—8。

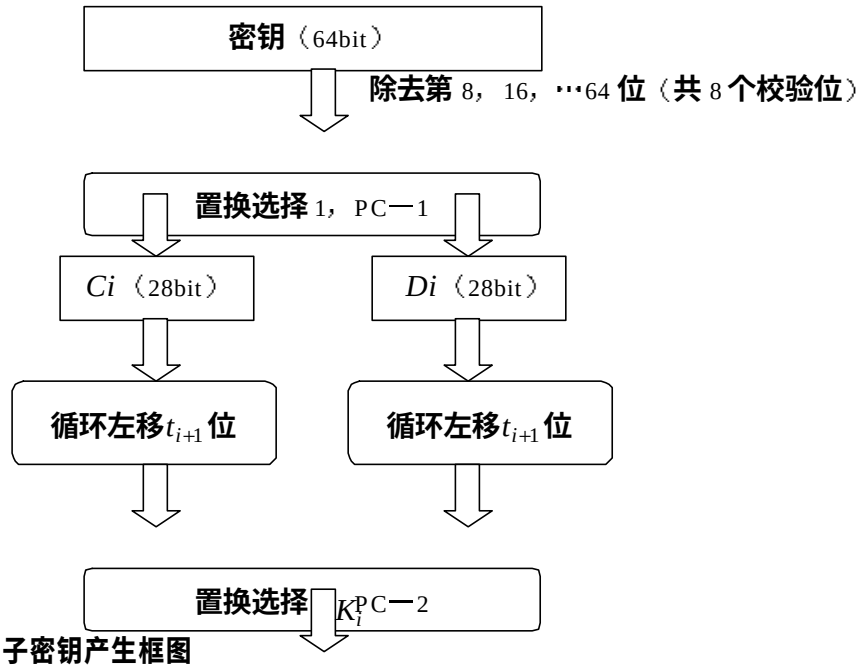


图 3—2—8　子密钥产生框图

在 64bit **初始密钥中有** 8 **位为校验位，其位置号为** 8，16，24，32，48，56，**和** 64。**其余** 56 **位为有效位，用于子密钥计算。将这** 56 **位送入置换选择** PC1，**参看图** 3—2—9。**经过坐标置换后分为两组，每组为** 28bit **分别送入** C **寄存器和** D **寄存器中。在各次迭代中，** C **和** D **寄存器分别将存数进行左循环移位置换，移位次数在表** 3—2—2 **中给出。每次移位后，将** C **和** D **寄存器的存数送给置换选择** PC2，**参看图** 3—2—10。**置换选择** PC2 **将** C **中第** 9，18，22，25 **位和** D **中第** 7，9，15，26 **位删去，并将其余数字置换位置后送出** 48bit **数字作为第**

i 次迭代时所用的子密钥 k_i 。

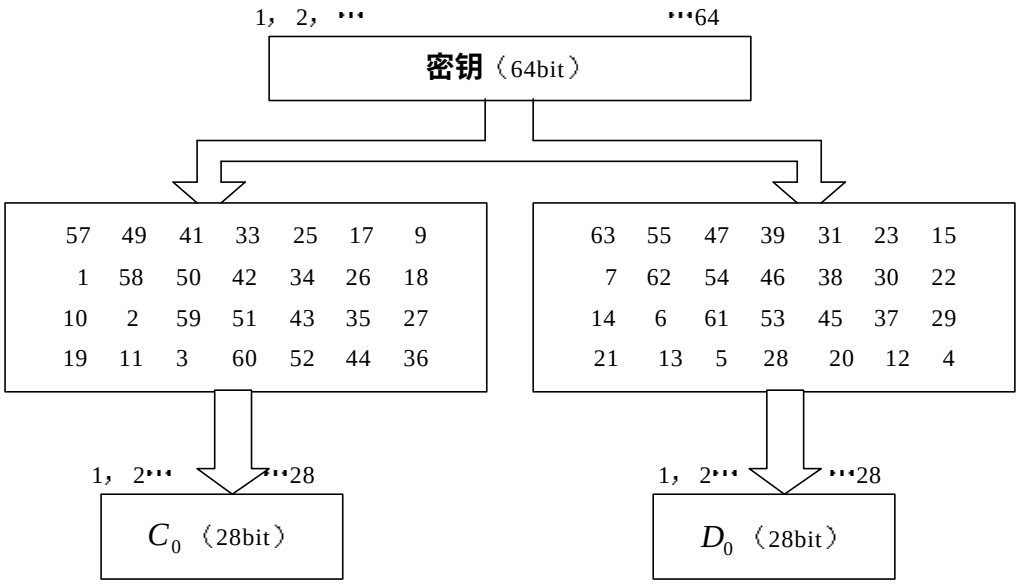


图 3—2—9　置换选择 PC -1

例如： $k = k_1 k_2 \ldots k_{64}$ **则**

$C_0 = k_{57} k_{49} \ldots k_{44} k_{36}$　　$D_0 = k_{63} k_{55} \ldots k_{12} k_4$

下面介绍如何从 C_i、D_i **求** C_{i+1}、D_{i+1}，i=0，1，2，…15，

设 $C_i = c_1 c_2 \ldots c_{28}$，$D_i = d_1 d_2 \ldots d_{28}$

首先要作左移（LS）**运算，左移的位数见表** 3—2—2

表 3—2—2　移位次数表

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

例如，设

$C_1 = c_1 c_2 \ldots c_{28}$，$D_1 = d_1 d_2 \ldots d_{28}$

则　　$C_2 = c_2 c_3 \ldots c_{28} c_1$，$D_2 = d_2 d_3 \ldots d_{28} d_1$

C_4 和 D_4 是由 C_3、D_3 左移 2 位而得到的，而 C_3、D_3 由 C_2 和 D_2 左移一位而得，故

$C_4 = c_5 c_6 \ldots c_{28} c_1 c_2 c_3 c_4$，$D_4 = d_5 d_6 \ldots d_1 d_2 d_3 d_4$

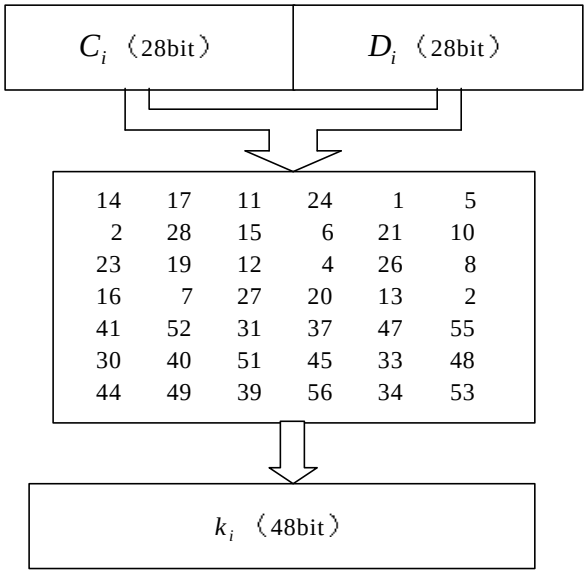


图 3—2—10　置换选择 PC—2

例如， $C_iD_i = b_1b_2 \ldots b_{56}$，**则** $k_i = b_{14}b_{17}b_{11}b_{24} \ldots b_{36}b_{29}b_{32}$

至此，我们已经将 DES **算法的基本构成作了介绍，加密过程可以归纳如下:**

令 IP **表示初始置换，** i **为迭代次数变量，** f **为加密函数，** k_i **为密钥，** $\oplus$ **表示模** 2 **加。**

加密过程

$L_0\ R_0 \leftarrow$IP（<64bit **输入码组**>）

$L_i \leftarrow R_{i-1}$

i=1，2，…，16　　（3—2—1）

$R_i \leftarrow L_{i-1} \oplus f(R_{i-1}, k_i)$

i=1，2，…，16　　（3—2—2）

<64bit **密文**>$\leftarrow$IP^{-1}($R_{16}\ L_{16}$)

DES **的解密过程和加密过程相似，只不过将密钥的顺序倒过来**

解密过程

$R_{16}\ L_{16} \leftarrow$IP(<64bit **密文**>)

$R_{i-1} \leftarrow L_i$

i=16，15 …，1　　（3—2—3）

$L_{i-1} \leftarrow R_i \oplus f(L_{i-1}, k_i)$

i=16，15 …，1　　（3—2—1）

<64bit **明文**>$\leftarrow$IP^{-1}($L_0\ R_0$)

§3.3 DES 的安全性

DES 的出现是密码学史上的一个创举。以前任何设计者对于密码体制及其设计细节都是严加保密的，而 DES 公开发表，任人测、研究和分析，无须经过许可就可以制作 DES 的芯片和以 DES 为基础的保密设备。DES 的安全性完全依赖与所用的密钥。

自从 DES 问世至今已经有 20 多个年头，尽管一开始人们就对它有颇多的担心和争议，这些年来许许多多的人对它进行各种各样的研究攻击，而且它逐渐地走向老化，但是至今为止并没有人真正地破译 DES。从目前的成果来看，除了穷举搜索攻击之外，就没有更好的方法破译 DES 了。

20 年来对 DES 进行的大量研究，主要成果集中在以下几个方面。

弱密钥

DES 算法在每次迭代时都有一个子密钥供加密用，如果一个外部密钥所产生的所有子密钥都是一样的，则这个密钥就称为弱密钥（weak key）。

若 k 为弱密钥，则有

$$DES_k(DES_k(x))=x \qquad (3-3-1)$$

$$DES_k^{-1}(DES_k^{-1}(x))=x \qquad (3-3-1)$$

即以 k 对 x 加密或解密两次都可以恢复出明文，其加密运算和解密运算没有区别。而对一般密钥只满足

（3—3—3）

弱密钥使 DES 在选择明文攻击下的搜索量减半。经分析可知，符合上述条件的外部密钥 k 共 4 个，它们是（16 进制表示）：

01 01 01 01 01 01 01 01

1F 1F 1F 1F 0E 0E 0E 0E

E0 E0 E0 E0 F1 F1 F1 F1

FE FE FE FE FE FE FE FE

若给定密钥 k，相应的 16 个子密钥只有两种，且每种都出现 8 次，就称它为半弱密钥（semi-weak key）。半弱密钥的特点是成对出现，具有下述性质：若 k_1 和 k_2 互一对半弱密钥，x 为明文组，则有

$$DES_{k_2}(DES_{k_1}(x))=DES_{k_1}(DES_{k_2}(x))=x \qquad (3-3-4)$$

半弱密钥共有 12 个，组成 6 对：

01	FE	01	FE	01	FE	01	FE	互逆对
FE	01	FE	01	FE	01	FE	01	
1F	E0	1F	E0	1F	E0	1F	E0	互逆对
E0	1F	E0	1F	E0	1F	E0	1F	
01	E0	01	E0	01	E0	01	E0	互逆对
E0	01	E0	01	E0	01	E0	01	
1F	FE	1F	FE	1F	FE	1F	FE	互逆对
FE	1F	FE	1F	FE	1F	FE	1F	
01	1F	01	1F	01	1F	01	1F	互逆对
1F	01	1F	01	1F	01	1F	01	
E0	FE	E0	FE	E0	FE	E0	FE	互逆对

FE	E0	FE	E0	FE	E0	FE	E0

如果随机选取密钥，在总数2^{56}个密钥中，弱密钥所占比例很小，加以注意就可以避开，对 DES 的安全性影响不大。

密文与明文、密文与密钥的相关性

有人详细研究了 DES 的输入，表明每个密文比特都是所有明文比特和所有密钥比特的复合函数，并且指出达到这一要求所需的迭代次数最少为 5，迭代 8 次以后输出和输入就可认为是不相关的了。

密钥搜索机

对 DES 的安全性的意见中，较为一致的看法是 DES 的密钥短了些。密钥长度是 56bit，密钥量为 2~56＝10~17 个。选择长密钥时会使成本提高，运行速度降低。若要对 DES 进行密钥搜索破译，分析者在得到一组明文-密文的情况下，可对明文进行不同的密钥加密，直到得到的密文与已知的密文-明文对中的相符，就可确定所用的密钥了。

1979 年，Differ 和 Hellman 认为利用 100 万个超大规模集成电路块所组成的一台专门用于破译 DES 的并行计算机能在一天中穷举搜索所有＃个密钥。每个集成块每微秒检查一个密钥，则每天可以检查＃个密钥。如果用一个集成块检查全部密钥，则几乎要花费＃天，约 2285 年。但是如果用＃个集成块，则在一天时间内可以检查密钥空间，这样一台机器在 1977 年将耗资约 2000 万美元。Differ 和 Hellman 据此指出，除了象美国国家安全局那样的机构外，任何人不可能破译 DES。但他们预测，到 1990 年制造和破译 DES 专用机的成本将要大幅度下降，到 1990 年 DES 将完全是不安全的。

事实证明，他们的预测是有道理的。在过去相当长的一段时间里，人们找不道比穷举搜索更有效的方法攻击 DES，也没有能力对 56 比特的密钥进行穷举搜索，因而在过去 DES 是安全的，但目前的事实证明这个历史已成为过去，DES 不能经受住穷举攻击。请看下面一段摘自报纸文章的文章:

Rocker Verser 是科罗拉多州 Loveland 公司的一名程序员，在 RSA 公司宣布进行名为“秘密密钥挑战”的竞赛后，Rocker Verser 开发了一个破译软件并组织了一个名为“DESCHELL”的小组，他们经过 96 天的协作努力，终于赢得了 RSA 公司发起的这场挑战赛。在这次 DES 挑战竞赛中，高校参与 积极性非常高，远远出乎意料。麻省理工学院的学生和教职员工都争相参与破译 56bit DES。该学院的计算机与遍及全美国的其它院校和单位的数万台设备一道，用穷举搜索的方法破译。参与破译的方法就是从 DESCHELL 分发主管处申请一个密钥区间，并通过 Internet 下载程序，这样就可以开始破译了。一个非正式的竞争迅速形成，因为每个人都想成为胜利者，许多学生们宿舍里的机器彻夜运转。

DESCHELL 小组吸引了 Internet 上数万名志愿者加入。Verser 说：“数万台计算机协同工作是在政府组织之外所取得的最大成果。”Verser 认为这项计划有两个方面值得一提:

第一，这是历史上第一次有人公开声明能破译 DES 加密的消息，而且这项计划是数万名从未见面的志愿者使用普通的 PC 机，利用“空闲”机时完成的。Verser 说：“对于一个坚定的攻击者，来说，DES 已经不安全了。”

第二，这项计划证明了 Internet 的超级计算能力，因为他们并没有花费任何额外代价，而使用空闲计算机机时。想象使用数百万台连接了 Internet 的计算机是什么情景。

按照 DESCHELL 最后搜索速度估算，穷举不同密钥空间所需的时间如下统计:

密钥长度（bit）	穷举时间
40	78 秒
48	5 小时

56	59 天
64	41 年
72	10,696 年
80	2,738,198 年
88	700,978,948 年
96	179,450,610,898
112	11,760,475,235,863,837 年
128	770,734,505,057,572,442,069 年

§4.5 基于编码的公钥密码

在密码学的研究中有一种努力，企图将纠错编码用于密码方面。由于纠错编码有着许多独特的功能，所以基于编码的公钥密码具有一些潜在的优势，可以依次编码完成对信息的加密和纠错两种功能，但是一般来讲，两者之间是有矛盾的。
为了讨论方便起见，有必要介绍一些编码理论的基本知识。

4.5.1编码理论简介:

一、
通常以（n，l）组码表示由 l 位 0，1 字符串 n 位 0，1 字符串的编码过程，即编码过程为将明文

$$m = m_1 m_2 \cdots m_i \qquad m_i \in \{0,1\} \quad i=0,1,\cdots,l$$

变换为码文

$$E(m) = w = w_1 w_2 \cdots w_n \qquad w_i \in \{0,1\} \quad i=0,1\cdots,n$$

码文在信道传输过程中难免受到干扰出错，编码的目的在于发现错误（检错），进而将错误纠正过来（纠错），所以检错和纠错是编码理论的主要内容。
例 4.5.1 (n+1，n)检错码，即在 n 位明文后面附加一位奇偶校验位，即

$$E(m) = m_1 m_2 \cdots m_n m_{n+1}$$

如果传输过程中出现了一个错，即将某一位的 0 错为 1，或将 1 错为 0，则可立即被发现。当然若 n 位中出两个错，反而发现不了，不过在 n 位中出两个错的可能性比较小。这样的码并不能判定哪位出了差错，只能是检错码。
例 4.5.2 (3n，n)码，即

$$E(m) = m_1 m_2 \cdots m_n m_1 m_2 \cdots m_n m_1 m_2 \cdots m_n$$

即对每组位的码连续传输了 3 遍，若某一位出现不同，则必两组是一样的，便以出现两次一样为准，达到纠错的目的。当然同时在同一位上出现两次或三次错的可能性虽存在，但终究不多见。设一次传输出错概率为 P_e =0.001，则(3n，n)码传输正确的概率为

$$\overline{P_e}^3 + 3P_e\overline{P_e}^2 = 0.997003 + 0.002994 = 0.999997$$

所以传输正确的概率从单次传输的 0.999，提高到 0.999997，但是码文比明文扩大了 3 倍，这很难被人接受。
纠错编码有许多种，可以对其进行不同的分类。

二、码间距离与码重

令 $m = m_1 m_2 \cdots m_i$， $m' = m_1' m_2' \cdots m_n'$ 都是长度为 n 的 0，1 符号串，定义

$$w(m)=\sum_{i=1}^{n}m_i$$

为 m 的 Hamming 重量或权，也就是 m 的 n 位中 1 的数目。

$$d(m,m')=w(m\oplus m')$$

称为 m 和 m' 的 Hamming 距离。

引理 4.5.1　若 $\boldsymbol{a}$，$\boldsymbol{b}$，$\boldsymbol{c}$ 都是长度为的 0，1 符号串，即

$$\boldsymbol{a}=a_1a_2\cdots a_n$$

$$\boldsymbol{b}=b_1b_2\cdots b_n$$

$$\boldsymbol{c}=c_1c_2\cdots c_n$$

$a_i,b_i,c_i\in\{0,1\}$，i=1，2，…n，则

（a）d（$\boldsymbol{a}$，$\boldsymbol{b}$）= d（$\boldsymbol{b}$，$\boldsymbol{a}$）

（b）d（$\boldsymbol{a}$，$\boldsymbol{c}$）≤ d（$\boldsymbol{a}$，$\boldsymbol{b}$）+ d（$\boldsymbol{b}$，$\boldsymbol{c}$）

定理 4.5.1　一组码可以检出 k 个错误的充要条件是码间最短距离至少为 k+1。

证明：　设 $\boldsymbol{w}=w_1w_2\cdots w_n$ 是码字，$\boldsymbol{w}$ 传输中有误差 $\boldsymbol{e}$，指收到的是 $\boldsymbol{r}=\boldsymbol{w}+\boldsymbol{e}$，$\boldsymbol{e}$ 能被检出的充要条件是 $\boldsymbol{w}+\boldsymbol{e}$ 不是码字。因此能检出 k 个错误的充要条件是码间最短距离至少为 k+1。

定理 4.5.2　若两个码字组之间的最短距离为 2 k +1，则可以纠正不超过 k 的误差。

证明：　设码字 $\boldsymbol{a}$ 传输过程中发生误差，得到的是 $\boldsymbol{r}$，且 d（$\boldsymbol{a}$，$\boldsymbol{r}$）≤ k，则不存在别的码字与 $\boldsymbol{r}$ 的距离不超过 k。如若不然，设为 $\boldsymbol{b}$，满足 d（$\boldsymbol{r}$，$\boldsymbol{b}$）≤ k，将有

$$d(\boldsymbol{a},\boldsymbol{b})\leqslant d(\boldsymbol{a},\boldsymbol{r})+d(\boldsymbol{r},\boldsymbol{b})$$
$$\leqslant k+k<2k+1$$

与假设矛盾，定理得证。

定理 4.5.2 **是极大似然译码方法的依据，例如有码字**

100010　01001　10101　01110

两两作模 2 **加运算⊕如下：**

	10010	01001	10101	01110
10010	-----	11011	00111	11100
01001	11011	-----	11100	00111
10101	00111	11100	-----	11011
01110	11100	00111	11011	-----

$\min\limits_{i\neq j}\{d(\boldsymbol{a}_i,\boldsymbol{a}_j)=3\}$，故能纠正一个错误。

$\boldsymbol{w}$	10010	01001	10101	01110
$\boldsymbol{r}$	00010	11001	00101	11110
	11010	00001	11101	00110
	10110	01101	10001	01010
	10000	01011	10111	01100
	10011	01000	10100	01111

上表给出距离为 1 的译码表，比如 $\boldsymbol{r}$=01011，它与 01001 距离为 1，故译为 01001。认为第 4 位出错。码字太大时靠译码表译码工作量和存储量都太大。

三、生成矩阵

令

$$G=(g_{ij})_{l\times n}=\begin{pmatrix}G_0\\G_1\\\cdots\\\cdots\\G_l\end{pmatrix}=\begin{pmatrix}g_{11}&g_{12}&\cdots&g_{1n}\\g_{21}&g_{21}&\cdots&g_{2n}\\\cdots&\cdots&&\cdots\\g_{l1}&g_{l2}&\cdots&g_{\ln}\end{pmatrix}$$

其中G_i为矩阵G的第i行向量，即

$$G_i=(g_{i1}g_{i2}\cdots g_{in})$$

$$g_{ij}\in\{0,1\}\text{，}i=1\text{，}\cdots\text{，}l\text{；}j=0\text{，}1\text{，}\cdots,n$$

设$m=m_1m_2\cdots m_l$为长度l的信息块或明文块，编码过程为:

$$E(m)=(m_1m_2\cdots m_{l-1})\begin{pmatrix}g_{11}&g_{12}&\cdots&g_{1n}\\g_{21}&g_{21}&\cdots&g_{2n}\\\cdots&\cdots&\cdots&\cdots\\g_{l1}&g_{l2}&\cdots&g_{\ln}\end{pmatrix}=mG$$

$$=\sum_{i=1}^{l}m_iG_i=W$$

例如:

$G=\begin{pmatrix}1&0&0&1&1&0\\0&1&0&0&1&1\\0&0&1&1&0&1\end{pmatrix}$对应于信息 101，对应的码字为

$$W=(101)\begin{pmatrix}1&0&0&1&1&0\\0&1&0&0&1&1\\0&0&1&1&0&1\end{pmatrix}$$

=（100110）+（001101）

=（101011）

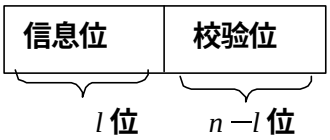
现将长度为 3 的信息和对应的码字列表如下：

信息			码字					
0	0	0	0	0	0	0	0	0
0	0	1	0	0	1	1	0	1
0	1	0	0	1	0	0	1	1
0	1	1	0	1	1	1	1	0
1	0	0	1	0	0	1	1	0
1	0	1	1	0	1	0	1	1
1	1	0	1	1	0	1	0	1
1	1	1	1	1	1	0	0	0

码字的（6 位）前 3 位为信息位，余下的 3 位是校验位。可将这种情况推广到（n，l）码上，即

$$G=\begin{pmatrix}1&0&0&\cdots&0&g_{11}&g_{12}&\cdots&g_{1n-l}\\0&1&0&\cdots&0&g_{21}&g_{22}&\cdots&g_{2n-l}\\\cdots&\cdots&\cdots&&\cdots&\cdots&\cdots&\cdots&\cdots\\0&0&0&\cdots&1&g_{l1}&g_{l2}&\cdots&g_{ln-l}\end{pmatrix}_{l\times n}$$

信息$m=m_1m_2\cdots m_l$经编码过程得到 n 位码字



为了方便起见，假定$m=m_1m_2\cdots m_l$明文对应的码文是

$$W=(m_1m_2\cdots m_l)=\begin{pmatrix}1&0&0&\cdots&0&g_{11}&g_{12}&\cdots&g_{1n-l}\\0&1&0&\cdots&0&g_{21}&g_{22}&\cdots&g_{2n-l}\\&&&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots\\0&0&0&\cdots&1&g_{l1}&g_{l2}&\cdots&g_{ln-l}\end{pmatrix}_{l\times n}$$

$=(m_1m_2m_3\cdots m_lm_{l+1}\cdots m_n)=\boldsymbol{r}$

显然有

$$m_{l+j}=\sum_{i=1}^{l}g_{ij}m_i\ ,\ j=1,\ 2,\ \cdots,\ n-l$$

或

$$g_{1j}m_1+g_{2j}m_2+\cdots+g_{ij}m_l+m_{l+j}=0$$

$$j=1,\ 2,\ \cdots,\ n-l$$

也就是说m_{l+1}，m_{l+2}，…，m_n即附加的多余部分是用来检验传输过程是否出错及哪些出错？所以叫校验位。$n-l$个方程可以用矩阵形式表示为：

$$\begin{pmatrix}g_{11}&g_{21}&\cdots&g_{l1}&1&0&\cdots&0\\g_{12}&g_{22}&\cdots&g_{l2}&0&1&\cdots&0\\\cdots&\cdots&\cdots&\cdots&&&&\\g_{1,n-l}&g_{2,n-l}&\cdots&g_{l,n-l}&0&0&\cdots&1\end{pmatrix}\begin{pmatrix}m_1\\m_2\\\cdots\\m_n\end{pmatrix}=\begin{pmatrix}0\\0\\\\\cdots\\0\end{pmatrix}$$

或写作

$$Hr^T=O \qquad (****-3-2)$$

称之为校验方程，$H=(h_{ij})_{n-l\,x_n}$称为校验矩阵。显然生成矩阵和校验矩阵有如下关系：

若

$G=(I_{(l)}|A)_{l\times n}$

其中 $A=(a_{ij})_{l\times(n-l)}$ **，** $I_{(l)}$ **为** l **阶单位矩阵，则有**

$H=(A^T|I_{(n-l)})_{(n-l)\times n}$

校验矩阵可以用来纠正错误。*式说明正确传输应该满足的等式。如若 $r=W+e$**，** e **是误差。**

$$Hr^T=H(W+e)^T=HW^T+He^T=He^T$$

若 $He^T\neq 0$**，可由** He^T **看出究竟第几位出类了错给予纠正。以**

$$H=\begin{pmatrix}1&0&1&1&0&0\\1&1&0&0&1&0\\0&1&1&0&0&1\end{pmatrix}\qquad r=1\ 0\ 0\ 1\ 0\ 1$$

为例

$$Hr^T=\begin{pmatrix}1&0&1&1&0&0\\1&1&0&0&1&0\\0&1&1&0&0&1\end{pmatrix}\begin{pmatrix}1\\0\\0\\1\\0\\1\end{pmatrix}=\begin{pmatrix}0\\1\\1\end{pmatrix}$$

$(011)^T$ **正好是** H **矩阵的第** 2 **列，故可知第** 2 **位出错。将** $r=100101$ **的第** 2 **位的** 0 **改为** 1**，故得**

$$W=110101$$

信息位为 110**，译码便结束。**

当然，若出现两个以上错误，这种方法便失败，也就是说不能获得正确的纠错，现将译码步骤总结如下:

设收到的信息为 $r=r_1r_2\cdots r_n$

S1: 计算 $S=Hr^T$

S2: 若 $S=O$**，则可认为传输过程是正确的，则明文** $m=r_1r_2\cdots r_l$ **，若** $S\neq O$ **转 S3。**

S3: 若 S **是矩阵** H **的第** i **列，则认为** r_i **有错误，予以改正。然后取前面的** l **位作为明文；**

若 S **不是** H **的列向量（且不为零），则认为传输过程中出现两个以上的错误，无法正确纠错。**

定理 4.5.3 $(n-l)\times n$ **的校验矩阵能正确纠正** 1 **个错误的充要条件是** H **的各列为不相同的非零列向量。**

（4）依据 $k\times n$ 阶矩阵 H 能纠一个错的充要条件是 H 矩阵的各列为不相同的非零列向量，当 k 确定以后，n 最大可取 2^k-1。又因 $k=n-l$，故 $l=n-k$。

例如 $k=3$ 时，$n=2^3-1=7$，$l=7-3=4$

例如

$$H=\begin{pmatrix}0&1&1&1&0&0\\1&0&1&0&1&0\\1&1&0&0&0&1\end{pmatrix}$$

H 的各列包含了除 $\begin{pmatrix}0\\0\\0\end{pmatrix}$ 以外的所有状态，对应生产矩阵为

$$G=\begin{pmatrix}1&0&0&0&0&1&1\\0&1&0&0&1&0&1\\0&0&1&0&1&1&0\\0&0&0&1&1&1&1\end{pmatrix}$$

用这种办法构成的可纠一个错的码称为（7，4）Hamming 码。

又如 $k=4$ 时，$n=2^4-1=15$，$l=15-4=11$

也可以构造 Hamming 码，0，1 字符长串度为 11 的明文经编码得长度为 15 的码字。
下面便是（15，11）的码的校验矩阵。

$$H=\begin{pmatrix}1&1&1&0&0&0&1&1&1&0&1&1&0&0&0\\1&0&0&1&1&0&1&1&0&1&1&0&1&0&0\\0&1&0&1&0&1&1&0&1&1&1&0&0&1&0\\0&0&1&0&1&1&0&1&1&1&1&0&0&0&1\end{pmatrix}$$

从校验矩阵不难求得对应的生成矩阵 G，这里从略。

4.5.2BCH 码和 Goppa 码

一、数学基本概念

二、BCH 码

BCH 是 Bose，Chaudhari，Hocquenghem 的缩写。BCH 码是他们三人于 1959 年同时发明的。

假定 α 是域 $GF(2^4)$ 的本原元素，满足 $\alpha^4+\alpha+1=0$。即 $\mathrm{mod}(\alpha^4+\alpha+1=0)$ 构成一个域。令

$$H=\begin{pmatrix}1&\alpha&\alpha^2&\alpha^3&\cdots&\alpha^{14}\\1&\alpha^3&(\alpha^2)^3&(\alpha^3)^3&\cdots&(\alpha^{14})^3\end{pmatrix}\qquad（4-5-1）$$

由于 $\alpha^0=1$，$\alpha^1=\alpha$，α^2，α^3，$\alpha^4=1+\alpha$，…，$\alpha^{14}=\alpha+\alpha^3+\alpha^4+=1+\alpha^3$，

$$\alpha^{15}=\alpha+\alpha^4=1$$

$$\therefore H=\begin{pmatrix}1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \cdots & \alpha^{14}\\ 1 & \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} & 1 & \cdots & \alpha^{12}\end{pmatrix} \qquad (4—5—2)$$

用下列矩阵来表示

$$H=\begin{pmatrix}
1&0&0&0&1&0&0&1&1&0&1&0&1&1&1\\
0&1&0&0&1&1&0&1&0&1&1&1&1&0&0\\
0&0&1&0&0&1&1&0&1&0&1&1&1&1&0\\
0&0&0&1&0&0&1&1&0&1&0&1&1&1&1\\
&&&&&&\cdots&&\cdots&&&&&&\\
1&0&0&0&1&1&0&0&0&1&1&0&0&0&1\\
0&0&0&1&1&0&0&0&1&1&0&0&0&1&1\\
0&0&1&0&1&0&0&1&0&1&0&0&1&0&1\\
0&1&1&1&1&0&1&1&1&1&0&1&1&1&1
\end{pmatrix}$$

这里用

$$\begin{pmatrix}0\\1\\0\\1\end{pmatrix},\ \begin{pmatrix}1\\0\\1\\1\end{pmatrix}$$

分别表示$\alpha+\alpha^3=\alpha^9$，$1+\alpha^2+\alpha^3=\alpha^{13}$。**其它依此类推。**

若在第 h，k **两位发生错误，则校验子**

$$S=\begin{pmatrix}\alpha^h+\alpha^k\\ \alpha^{3h}+\alpha^{3k}\end{pmatrix}=\begin{Bmatrix}z_1\\ z_2\end{Bmatrix}$$

$$z_1=\alpha^h+\alpha^k,\quad z_2=\alpha^{3h}+\alpha^{3k}$$

$$\therefore z_2=(\alpha^h+\alpha^k)(\alpha^{2h}+\alpha^{h+k}+\alpha^{2k})$$

$$=z_1(z_1^2+\alpha^{h+k})$$

$$\alpha^{h+k}=\frac{z_2}{z_1}+z_1^2$$

故α^h **和**α^k **是方程**

$$z^2+z_1z+(\frac{z_2}{z_1}+z_1^2)=0 \qquad (4—5—3)$$

的两个根。

若 $z_1 = z_2 = 0$，则认为无差错，若 $z_2 = z_1^3 = \alpha^{3h}$，则认为有一个错误发生在第 h 位。

例4.5.1　若在第 7、9 位发生错误，则

$$z_1 = \begin{pmatrix}0\\0\\1\\1\end{pmatrix} + \begin{pmatrix}1\\0\\1\\0\end{pmatrix} = \begin{pmatrix}1\\0\\0\\1\end{pmatrix} = \alpha^{14}$$

$$z_2 = \begin{pmatrix}0\\0\\0\\1\end{pmatrix} + \begin{pmatrix}0\\1\\0\\1\end{pmatrix} = \begin{pmatrix}0\\1\\0\\0\end{pmatrix} = \alpha = \alpha^{16}$$

$$\frac{z_2}{z_1} + z_1^2 = \alpha^2 + \alpha^{13} = \alpha^2 + 1 + \alpha^2 + \alpha^3 = 1 + \alpha^3 = \alpha^{14}$$

$$\therefore z_2 + \alpha^{14} z + \alpha^{14} = (z + \alpha^6)(z + \alpha^8) = 0$$

这就验证了 BCH 纠错是正确的。

三、 Goppa 码

长度为 n 的 Goppa 码要用到两个概念，一是系数在 GF

第四章　公钥密码体制

§ 4.1　公钥密码体制的产生

密码学在相当长的时间内，仅在军事、外交等与国家有关的机要部门有所应用。信息加密技术也仅为这些机要部门所掌握。然而，随着社会的经济发展和科学技术的进步，这种情况发生了变化。

微电子技术、计算机技术和通信技术的飞速发展，使信息业迅速膨胀。信息交流的规模、内容、形式、手段和频繁程度都呈激增的态势。四通八达的信息网络更是给人们的工作和生活带来了很大的变化。人们在享受高科技成果的同时，也日益感受到其背后潜伏着的脆弱性、不安全性和危险性。个人通信中的私人秘密不希望他人知道，生意往来中的商业秘密不愿意暴露给他人，计算机网络中的共享数据不想让非法用户随意享用，各部门和系统中的重要资料和档案更不能让无关的人随意插手……。总之，随着信息交换量的激增，人们对通信保密和安全的需求也日益增长、日益迫切。于是，密码学就从机要部门走向了非机要部门，密码技术也迅速为民间人士所掌握和运用。

但是，随着微电子技术、计算机技术和现代通信技术的迅猛发展，通信保密所遭受到的威胁也越来越大。

首先，微波通信、卫星通信、无线移动通信的发展和电磁辐射技术的应用，使窃听者能方便地利用固定的或移动的天线，截获空间的通信电波或捕捉到微弱的电磁辐射信号，而毋需再冒搭线窃听的危险。

其次，近代计算机技术的惊人发展，使其计算速度越来越快、存储容量越来越大、功能越来越强。因而密码分析者可利用的计算机的性能越来越好。加上电子通信越来越多地使用数字形式，这就使得密码分析者能方便地利用计算机，对所截获的电子情报进行高强度的分析。

再者，高速信息网络以及高科技的发展也给密码分析者提供了越来越多的获取信息的途径、越来越先进的获取信息的手段和越来越好的获取信息的方法。

还有，随着对保密通信威胁的增长，人们对保密通信的要求已不再局限于“不泄露”，而是更进一步要求保密通信不被“破坏”或者遭到破坏后能很快地识别出来。这就要求通信系统不仅有保密的功能，还要有认证和鉴别的功能。换句话说，通信系统不仅要做到“保密”，而且要做到“安全”。

由上可见，当代保密通信所面临的工作环境对密码体制提出了越来越多、越来越高的要求。要满足这些要求，早期的古典密码体制已显得无能为力。那么，什么样的密码体制能满足现代密码通信的要求呢？

当代人们根据加密密钥能否公开，将密码体制划分为两大类，一类叫传统密码体制，另一类叫公钥密码体制。

所谓传统密码体制，指的是几千年来密码通信一直沿用的一类密码体制。这类密码体制在加密和解密时，使用的是同一个密钥，或者虽然使用不同的密钥，但是能通过加密密钥方便地导出解密密钥。这类密码体制也叫做对称密码体制，或单密钥密码体制。显然，在这类密码体制中，加密密钥是整个密码通信系统的核心机密，一旦加密密钥被暴露，整个密码体制也就失去了保密作用。

随着信息加密技术应用领域的扩大，尤其是从单纯的军事、外交、情报领域的使用，扩大

到民用领域（如商业、金融、计算机通信网络中的信息保密等），传统密码体制在应用中暴露出越来越多的缺陷，如:

（1）密钥管理的麻烦

密钥管理包括密钥保存和分配。密钥分配，是传统密码体制遇到的最大困难之一。因为在以传统密码体制为基础的保密通信中，通信双方所使用的都是同一个密钥，而密钥又是传统密码体制的核心，它是绝对不能餐露给第三者的。于是，在一次密码通信开始之前，信息的发送方必须提前把所用的保密密钥，经过特殊的秘密渠道，如信使、挂号信等，或者经一条特殊的保密通信线路（即密钥信道）送到信息的接收方。在计算机通信网中，则要由主机把该次所用的密钥分送给交换信息的两个用户。

经特殊的密钥信道分配保密密钥是相当困难的。随着系统用户的增加，这种困难程度变得越来越严重。而近代商用和民用密码通信的特征之一，就是用户很多。如在一个民用密码通信网中，用户数为 n，则每个用户都要保存 n 个密钥，包括他自己的密钥。系统可能的用户对数，亦即系统拥有的密钥总数，将增加到 $n(n-1)/2$。一个拥有 10 万用户的民用密码通信网，就要拥有 5×10^9 个密钥。显然，要十分妥善地保存这些密钥，本身就是一个难题，而要经特殊的保密信道分配这么多密钥，更是难以想象!试想要做到真正保密，那么每两个用户之间都需要一条秘密信道，这在经济上也是绝对不允许的。不仅如此，按传统方法分配密钥，必然带来实际通信时间的推迟。然而在商业上，时间的贻误往往意味着经济的损失，这是任何一个商业用户所不希望的。

（2）不能提供法律证据

在现今国际和国内的商业往来中，合同和协议的真实性是由书面签字来保证的。一张签字的合同可以作为法律上的证据。但是现代电子通信最易于剪接、窜改和伪造。要用“电子信件”代替“书面信件”，不仅要解决“保密问题”，而且要解决“证实问题”。

所谓信息的保密，是指信息从发送方流通到接收方的过程中，内容不会泄露给任何非法截收的第三者。而信息的证实，是指两个方面:一方面是指对发送方的证实；另一方面是指对接收方的证实。也就是能够确认接收方所收到和保存的信息，确实是由发送方发出的，既不是伪造的，也没有经过包括接收方在内的其他人所窜改。在传统密码体制中，接收方利用保密密钥，对密文信息进行解密变换，能成功地解决信息的保密问题。因为只有掌握解密密钥的合法接收者，才能从密文恢复出明文。这能够解决接收方对发送方的证实问题，使接收方能够确认该信息确实是由发送方送出的，因为只有合法的发送方才拥有该次通信的加密变换密钥。但是却无法解决对接收方的证实问题。因为接收方所掌握的解密密钥和发送方的加密密钥相同，完全有能力窜改他接收到的文件或伪造文件。基于同样的理由，发送方完全有借口抵赖他曾发出的文件。例如，用户甲利用商用密码通信向用户乙订购了一式批货物，用户乙如数寄出，后来由于该商品价格猛跌，用户甲拒绝付款。这样，甲乙双方就发生争端，由于上述原因，用户乙不能提供有说服力的法律证据，则法院无法受理此案。基于这种原因，传统密码体制难以在商业上获得更广泛的应用。

（3）缺乏自动检测保密密钥泄密的能力

由于传统密码体制在分配密钥上的实际困难，因此，通信密钥一旦指定，总要使用一定时间，难以随时更换，更难一次一换。所以，窃听者一旦破译出通信密钥，则在该密钥有效使用期内，就能顺利地破译出该密钥加密的所有信息。而合法的密码通信双方却无法察觉，显然这将会使通信双方蒙受巨大的损失。

为了对付密钥失窃所带来的损失和危害，最好采用一次一密的密码体制，即每次通信都采用不同的密钥。这要求密钥的分配迅速、保密和经济，而传统密码体制是做不到这一点的。

为了解决传统密码体制所不能解决的问题，就必须寻求新的密码体制，于是公钥密码体制便应运而生。

§4.2 公钥密码体制的基本原理

本世纪 70 年代，美国学者 Diffie 和 Hellman,以及以色列学者 Merkle 分别独立地提出了一种全新的密码体制的概念。Diffie 和 Hellman 首先将这个概念公布在 1976 年美国国家计算机会议上，几个月后,他们这篇开创性的论文《密码学的新方向》（“New Directions in Cryptography"）被发表在 IEEE 杂志信息理论卷上，由于印刷原因，Merk1e 对这一领域的首次贡献直到 1978 年才出版。他们所创造的新的密码学理论，突破了传统密码体制对称密钥的概念，树起了近代密码学的又一大里程碑。

不同于以前采用相同加密和解密密钥的对称密码体制，Diffie 和 Hellman 提出，采用双钥体制的每个用户都有一对选定的密钥：一个可以是公开的，以k_1表示；另一个则是秘密的，以k_2表示。公开的密钥可以象电话号码一样公布，因此称为公钥密码体制（public key cryptosystem，PKC）或双钥体制（tow-key cryptosystem）。

公双钥体制的主要特点是将加密和解密的能力分开，因而可以实现多个用户的的信息只能由一个用户解读；或只能由一个用户加密消息而由多个用户解读。前者可用于公共网络中实现保密通信，而后者可以用于认证系统中对消息进行数字签字。

公钥体制用于保密通信可用图 4—2—1 表示。图中，假定用户 A 向用户 B 发送机密消息 m。用户 A 在公钥本上查到用户 B 的公开钥k_{B1}，以它对消息 m 进行加密得到密文 $c=E_{B1}(m)$，送给用户 B。用户 B 以自己的秘密钥k_{B2}对进行解密变换得到原来的消息

$$m = D_{kB2}(c) = D_{kB2}(E_{kB1}(m)) \qquad (4—2—1)$$

系统的安全保障在于从公开钥k_{B1}和密文 c 要推出明文 m 在计算上是不可行的。

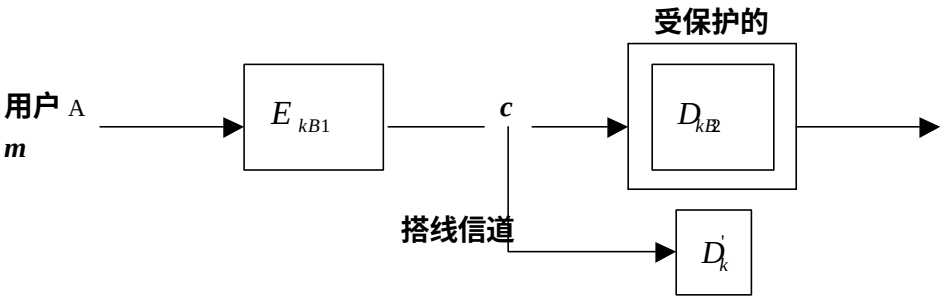


图 4—2—1 双钥保密系统

由于任一用户都可以用用户 B 的公开钥k_{B1}向他发送机密消息，因而密文 c 不具有确证性。双钥体制也可以用于认证系统。为了使用户 A 发给用户 B 的消息具有确证性，可以将公开钥和秘密钥反过来用，如图 4—2—2 所示。用户 A 以自己的秘密钥k_{A2}对消息 m 进行解密

变换，得到密文 $c = D_{kA2}(m)$ 送给用户 B。用户 B 收到 c 后可用 A 的公开钥 k_{A1} 对 c 进行加密变换得到恢复的消息

$$m = E_{kA1}(c) = E_{kA1}(D_{kA2}(m)) \qquad \text{〈4—2—2〉}$$

由于 k_{A2} 是保密的，其他人都不可能伪造密文 c，使用 A 的公开钥得到有意义的消息 m。因而可以验证消息 m 来自 A 而不是来自于其他人，实现了对 A 所发消息的认证。

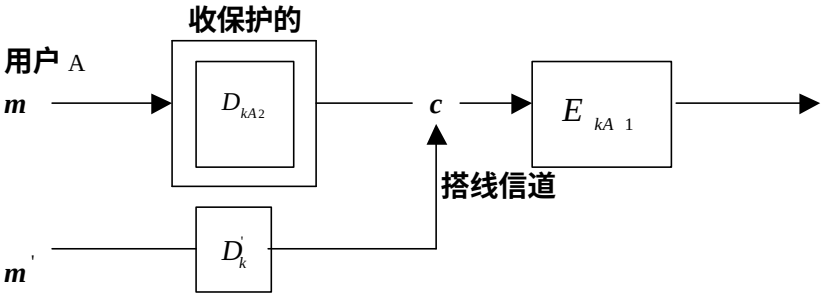


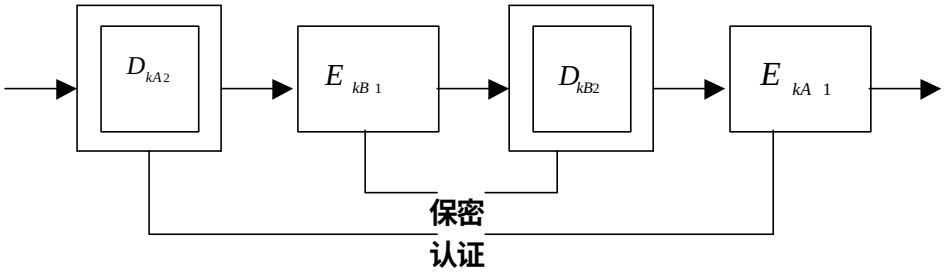
图 4—2—2 双钥认证系统

为了要实现保密性和确证性，要采用双重加密、解密，如图 4—2—3 所示。在明文消息空间和密文消息空间等价，且加密、解密运算次序可换，即 $E_{k1}(D_{k1}) = D_{k2}(E_{k1}(m)) = m$ 的条件下，双钥体制就不难实现。例如，用户 A 要向用户 B 传送具有认证性的机密消息 m，可将 B 的一队密钥作为加密和解密用，而将 A 的一对密钥作为认证之用。可按图 x-1-3 的顺序进行变换。A 发送给 B 的密文为

$$= E_{kA1}(D_{kB2}(c))$$

$$= E_{kA1}(D_{kB2}(E_{kB1}((m))))$$

$$= E_{kA1}((m))$$



图—1—3 双钥保密和认证系统

B 恢复明文的过程为

$$m = E_{kA1}(D_{kA2}(c))$$

$$= E_{kA1}(D_{kB2}(E_{kB1}(D_{kA2}(m))))$$

$$= E_{kA1}\ (\ D_{kA2}(\boldsymbol{m}))$$

双钥体制特别适用于多用户通信，它大大减少了多用户之间通信所需的密钥量，便于密钥管理。这个体制的出现是密码学研究中的一相重大突破。它标志了现代密码学的诞生。

公钥密码一问世就受到了众多密码学家的青睐。许多优点使它特别适合于现代密码通信。近 20 年来，出现了许许多多建立在各种数学基础之上的公钥密码体制，不过遗憾的是大多数公钥密码体制未能经受住密码分析的考验，只有少数几种公钥密密码体制被认为是安全的。另外，公钥密码体制的工作基础是构造单向函数，这使得它的加密和解密要经过比较复杂的计算过程。因此，一般来说，公钥密码体制的工作速率还远低于传统密码体制，这是它的最大缺点。

公钥密码体制一般分为两大类，一类叫公钥密码体制，另一类叫公钥密钥分配体制（public key distribution system）。公钥密码体制直接对待发送的明文进行加密，它对信息加密和解密使用完全不同的密钥，加密密钥可以公开。而公钥密钥分配体制并不直接对待加密的明文进行加密，而是设法在不保密的信道上传送秘密密钥，一旦通信双方获得了该次通信的加密、解密密钥，就可按传统密码体制的工作方式进行加密和解密。

§4.3 RSA 公开密钥体制

Diffie 和 Hellman 提出公钥设想后两年，先后提出了 MH-背包公钥体制，RSA 公开密钥体制。MH-背包公钥体制将在随后介绍，现在先讨论 RSA 公开密钥体制。RSA 虽稍后于 MH-背包公钥体制，但它到目前为止仍不失为有希望的一种，它是美国 MIT 的三位学者 Rivest、Shamir 和 Adleman 1978 年在他们的著名论文“A Method for Obtaining Digital Signature and Public–Key Cryptosystem”中提出的。该体制的名称正是由三位作者名字的第一个字母拼合而成。 RSA 的基础是数论中的下述论断：求两个大素数（比如大于十进制 100 位）的乘积，在计算机上很容易实现，但是要将这样一个大合数分解成两个大素数之积，在计算机上很难实现。RSA 公钥密码体制的核心是大数幂剩余计算。至今为止，RSA 公钥密码体制被认为是少数几个比较理想的公钥密码体制之一。

一、 数学基本概念

为了介绍这种公钥密码体制，我们引进集合

$$Z_N = \{0,1,2,3\ldots N-1\}$$

及其子集

$$Z_N^* = \{a \mid a \in Z_N, \gcd(a,N) = 1\}$$ 其中，gcd（a，N)表示 a 和 N 的最大公因数

可以证明Z_N^* 对于模 N 乘法运算是一个交换群，故称Z_N^* 为模 N 乘群。首先介绍两个定理并给出证明，证明仅供参考。

定理 4.3.1（欧拉定理）$\forall a \in Z_N^*$ 有

$$a^{j(N)} \equiv 1 \mod N \qquad (4-3-1)$$

其中 $j(N)$ 是欧拉函数，表示比 N 小但与 N 互素的正整数的个数。

证明　　记

$$Z_N^* = \{a_1,\ a_2,\ldots a_{j(N)}\}$$

对于 $\forall a \in Z_N^*$，**由于存在** a **的逆元** $a^{-1} \in Z_N^*$，**所以容易证明以下集合仍然是** Z_N^*：

$$\{aa_1 \bmod N, aa_2 \bmod N, \ldots, aa_{j(N)} \bmod N\}$$

于是有

$$\prod_{i=1}^{j(N)} (aa_i) \equiv \prod_{i=1}^{j(N)} a_i \mod N$$

即：

$$a^{j(N)} \prod_{i=1}^{j(N)} a_i \equiv \prod_{i=1}^{j(N)} a_i \mod N$$

所以

$$a^{j(N)} \equiv 1 \mod N$$

定理 4.3.2 设 p, q **是两个不同的素数** $N = p * q$，$j(N) = (p-1)(q-1)$，$m \in Z_N$，**则对于任意非负整数** k **有：**

$$m^{kj(N)+1} \equiv m \mod N \qquad (4-3-2)$$

证明：若 gcd（m，N）=1，**则由欧拉定理可知：**

$$m^{j(N)} \equiv 1 \bmod N$$

若 gcd（m，N）>1，**由于** $N = p * q$，**故** gcd（m，N）**必含** p、q **之一，**

设 gcd（m，N）=p，**或** $m=cq$，$1 \leqslant c<q$，**由欧拉定理:**

$$m^{j(q)} \equiv 1 \bmod q$$

因此对任何非负整数 k，**总有**

$$m^{k(q-1)} \equiv 1 \bmod q$$

$$m^{k(p-1)(q-1)} \equiv (1)^{k(p-1)} \equiv 1 \bmod q$$

即

$$m^{kj(N)} \equiv 1 \bmod q$$

或

$$1 = m^{kj(N)+1} + hq$$

其中 h 是某个整数。由假定 $m=cp$，

$$\therefore m = m^{kj(N)+1} + hcpq$$

这就证明了

$$m \equiv m^{kj(N)+1} \bmod N$$

定理 4.3.3 辗转相除法，又称欧几里得算法

给定整数 b 和 c，且设 $c>0$，重复地使用带余除法，即用每次的余数为除数去除上一次的余数，直至余数为 0，这样可得下面一组方程

$$b = cq_1 + r_1, 0 < r_1 < c ,$$

$$c = r_1 q_2 + r_2, 0 < r_2 < r_1 ,$$

$$r_1 = r_2 q_3 + r_3, 0 < r_3 < r_2 ,$$

$$\cdots\cdots \quad \cdots\cdots$$

$$r_{j-2} = r_{j-1} q_j + r_j, 0 < r_j < r_{j-1} ,$$

$$r_{j-1} = r_j q_{j+1} ,$$

那么在上面带余除法过程中，最后一个不为 0 的余数就是和的最大公因数 gcd（b，c）。

欧几里得算法经常要在公钥密码体制中用到，我们举一个例子，以便清楚地理解上述过程。

令 b=963，c=657

则有下述方程组

963=657×1+306

657=306×2+45

45=36×1+9

36=9×4

因此 gcd（963，657）=9。通过消除上述余数 36，45，和 306，可以把最大公因数 9 表示为 963 和 657 的一种线性组合

9=45—36

=45—（306—45×6）

=—306＋7×45

=—306＋7（657—306×2）

=7×657—15（963—657）

=22×657—15×963

二、RSA 公钥密钥体制的加密和解密

现在，我们来描述 RSA 体制的任一用户，比如用户 B，为自己构造密钥（key generation）的步骤。

KG 1 随机选取两个大素数 p 与 q，$p \neq p$，计算出 N=$p*q$，$j(N) = (p-1)(q-1)$ 。

KG 2 在模 $j(N)$ 乘群 Z_N^* 求出一对逆元 $e, d, e \neq d$，使

$$e * d \equiv 1 \bmod j(N) \qquad (4-3-3)$$

KG 3　公布自己的公开密钥（N，e），密藏自己的秘密密钥（p，q，d）。

假设用户 A 要向 B 传送消息。A 首先需要将消息按比特串分组，分组长度 l 保证 $2^l \leqslant N$。

用 m 表示某一组消息的十进制记法：$0 \leqslant m \leqslant N$。由用户 A 与用户 B 分别实施的加密与解密运算如下：

加密：

$$c \equiv m^e \bmod N \qquad (4-3-4)$$

解密：

$$m \equiv c^d \bmod N \qquad (4-3-5)$$

式（4—3—5）的正确性由定理 4.3.2 及式（4—3—3）保证。

为了保证 RSA 体制的安全性，目前要求选取不小于十进制 100 位的素数作为 p 与 q 以防止攻击者简单分解 N 得到 p 与 q。

例 4.3.1　设 $p = 43, q = 59, N = pq = 43 \times 59 = 2537, j(N) = 42 \times 58 = 2436$，取 e=13，求 e 的逆元 d：

解方程 $de = 1 \bmod 2436$

∵　2436＝13×187＋5，13＝2×5＋3

　　5＝3＋2，3＝2＋1

∴　1＝3—2，2＝5—3，3＝13—2×5

　　5＝2436—13×187

∴　1＝3—2＝3—（5—3）＝2×3—5

　　＝2（13—2×5）—5＝2×13—5×5

　　＝2×13—5（2436—13×187）

　　＝937×13—5×2346

即： 937×13≡1 mod 2436

取 e＝13 时，d＝937

若有明文： public key encryptions

先将明文分块为：

pu　bl　ic　ke　nc　ry　pt　io　ns

将明文数字化（令 a，b，…z **分别为** 00，01，…25**）得：**

1520　0111　0802　1004　2404

1302　1724　1519　0814　1418

利用加密可得密文：

0095　1648　1410　1299　1365

1379　2333　2132　1751　1289

RSA 体制不仅能实现保密通信，还能实现数字签名，因而特别适用于现代密码通信的要求。在众多的公钥密码体制中，RSA 公钥密码体制被认为是比较完善的一个，自从它问世至今的 20 余年中，尽管有许多密码学家对它进行了长期而深入的分析，但是一直没有发现它有明显的脆弱性。

所以，RSA 公钥密码体制至少有以下一些优点：

（1）数学表达式简单，在公钥密码体制中是最容易理解和实现的一个。这个体制也是目前国际上比较流行的公钥密码体制之一。
（2）RSA 的安全性基于大数分解的困难性。到目前为止，除了大数分解之外人们还没有发现一种其它的方法能够对 RSA 进行有效的密码分析。虽然 RSA 也有一些弱点，但是只要设计密码参数时仔细一点，这些弱点是可以避免的。RSA 公钥密码体制的安全性比较高。
（3）RSA 公钥密码体制具有一些传统密码体制不能实现的一些功能，如认证、鉴别和数字签名等，特别适合于现代密码通信。
在实际应用中，RSA 公钥密码体制的公开和秘密密钥参数是一对 100~200 位的大素数的函数，其本身的数值也很大。因此，这中体制的加、解密速度很慢，这是它广泛应用的最大障碍。虽然有很多人致力于 RSA 运算速度的研究，并且在这方面不断取得进展，利用一些 RSA 快速算法，可以减少 30%左右的求模运算量。但即使如此，到目前为止，RSA 芯片的速度仍比 DES 芯片的速度慢至少两个数量级。

三、RSA 公钥密码体制的安全性*

对密码分析者来说，攻击 RSA 体制的最显而易见的方法就是分解模数 N，因为一旦求出 N 的两个素因子 p 和 q，N 的欧拉数 $j(N)=(p-1)(q-1)$ 立即可得，再求出以为 $j(N)$ 模的公钥 e 的乘逆 d，从而破译 RSA 的秘密密钥。
但是分解大合数是众所周知的数学难题。近 300 多年来，许多著名的数学加对这个问题进行了大量研究。虽然近代科学家借助电子计算机和先进的科学技术在这方面取得了很大的进展，数学家已经成功分解了 150 位（十进制）的大合数，但是至今为止，要分解一个 200 位的大合数，还是无能为力的。

如果有一种方法能够不分解 N 而方便地计算出 $j(N)$，则 RSA 体制可轻而易举地被攻破。

不过目前还没有什么人能够证明直接计算 $j(N)$ 比分解 N 更容易。从另一个方面也说明为什么 RSA 密码体制的模数 N 必须选择合数而不能选择素数。因为如果模 N 选择素数那么公布了 N 也就公布了 $j(N)$，从 $j(N)$ 和公开钥 e 可以很容易地推导出秘密密钥 d。

既然秘密密钥 d 不能推导出来，那么它能不能直接计算出来呢？最显而易见的方法当然是穷举搜索法。但是只要密码设计者仔细选取 d，并使它本身足够大，就可以挫败密码分析者的穷举搜索攻击。
如果密码分析者知道了 d，可以方便地按下述方法实现对的分解 N。首先计算出（$ed-1$），由 RSA 体制的协议可知，（$ed-1$）必然是 $j(N)$ 的一个倍数。密码学家 Miller 指出，利用 $j(N)$ 的任何倍数都能成功实现对 N 的分解。但是到目前为止，还没有人能够提出这类对 N 进行因数分解的实际方法。因此可以断言，密码分析者不可能用比分解 N 更容易的方法直接计算秘密密钥，否则大合数分解的难题就不存在了。
但是这并不意味着密码分析者不能通过其它途径来破译 RSA 体制。
美国学者 Simmons 指出 RSA 公钥密码体制存在着一种潜在的弱点，这就是幂剩余变换的周期性。利用这种特殊的周期性，可以不必破译秘密密钥而直接得到明文。其方法如下：
设是待 m 加密的明文，（N，e）是 RSA 密码体制的公开密钥，则密文

$$c = m^e \bmod N$$

密码分析者从不保密信道上得到密文 c 后，并不设法获取秘密密钥 d，而是利用公开密钥对密文 c 依次进行如下变换:

$$c^e = c_1 \bmod N$$

$$c_1^e = c_2 \bmod N$$

… …

$$c_{k-1}^e = c_k \bmod N$$

$$c_k^e = c_{k+1} = c \bmod N$$

上述一组变换经过（k+1）步迭代之后，幂剩余变换的结果恰好等于密文 c。对照式 $c = m^e \bmod N$ 可知第 k 步的变换结果 c_k 就是明文 m。这样，利用众所周知的公开密钥（N，e）和截获的密文成功地破译了 RSA 公开密钥体制。

设 RSA 公开密钥体制的一组参数为 p=383，q=563，m=pq=215629，e=49，d=56957，加密和解密变换式分别为:

$$m^{49} = c \bmod 215629$$
$$c^{56957} = m \bmod 215629$$

如果明文信息 m=123456，则相应的密文为

$$123456^{49} = 1603 \bmod 215629$$

密码分析者利用公钥（215629，49）和密文 1603 按所示的一组变换进行运算，得到

$c_1 = 180661$，$c_2 = 109265$，$c_3 = 131172$，$c_4 = 98178$，$c_5 = 56372$，$c_8 = 85978$，

$c_9 = 123456$，$c_{10} = 1603 = c \bmod 215629$。由此可见仅仅经过 10 步变换，就得到 $c_{10} = c$，于是立即推知 $c_9 = m = 123456$。

这是由于幂剩余函数满足下述周期性定理。即若为 N 素数或不同素数之积，且 $\gcd(e, \varphi(N)) = 1$，则

$$a^{ek} = a \bmod N \qquad （4—3—6）$$

式中 k 为某一正整数。当上述例子中模数的欧拉数为 $\varphi(N) = 214684$ 时，49 的指数是 10。也就是说周期 k 仅为 10，这固然是加密指数 e 的特定选择所决定的。这种结果也很容易理解。因为 $\gcd(e, \varphi(N)) = 1$，所以每次变换均为一一映射，由于信息空间是有限的，连

续映射，最终一定会恢复到初始状态。
要避免密码分析者利用幂剩余的周期性破译 RSA 体制，就要使 k 足够大以至于密码分析者在合理的时间内不能够破译 RSA 密文。事实上，幂剩余函数的周期 k 与模数 N 的两个素因子的欧拉数 $j(p)$ 和 $j(q)$ 的最小公倍数 l 有关，l 越大，幂剩余函数的周期就越长，反之 l 越小，周期就越小。显然，为了使 RSA 公钥密码体制实际上是不可能破译的，理当选择 l 较大的素因子 p 和 q。
还有一个值得注意的问题是，在以 RSA 公钥密码体制作为加密标准的通信网中，每个用户的密钥不能有相同的模值。这里最显而易见的问题是同一明文用相同的模数但用不同的指数加密，而且这两个指数是互素的。一般情况下这很有可能，那么无需任何一个解密密钥就可恢复出明文。

假设 m 是明文信息，有两个不同用户的加密密钥分别是 e_1 和 e_2，他们有共同的模数 N，两个密文信息分别为：

$$c_1 = m^{e_1} \bmod N$$
$$c_2 = m^{e_2} \bmod N$$

密码分析者知道 m，e_1，e_2，c_1 和 c_2，他们用下述方法恢复出明文 m。

由于 e_1 和 e_2 和互素，由欧几里德算法能够找出 r 和 s 使之满足

$$re_1 + se_2 = 1$$

在 r 和 s 中，有一个是负数。假定 r 是负数，那么再用欧几里德算法计算出 c_1^{-1}，然后有

$$(c_1^{-1})^{-r} c_2^s = m^{e_1 r + e_2 s} \equiv m \bmod N$$

这样，无需破译秘密密钥 d，就可以得到明文 m。这种方法叫做 RSA 的共模攻击。避免这种攻击的措施是不在一组用户之间共享 N。
综上所述，在使用 RSA 公钥密码体制时，必须注意以下问题:

（1）选择素数 p 和 q 时，应使这两个素数的欧拉数 $j(p)$ 和 $j(q)$ 的最小公倍数 l 尽可能大，l 越大，幂剩余函数的周期就越长。这样就可以避免密码分析者利用剩余函数的周期性破译该体制。
（2）密钥中的各项参数应选得足够大，以避免密码分析者利用穷举搜索来破译该体制。
（4）在同一个通信网络中，不同的用户不应该使用共同的模数。

四、Rabin 体制

如果密码分析者能够有效地分解一般的大合数，比如 $N=p*q$，那么他就能破译 RSA 体制。然而对其逆命题的真实性尚未有结论。为此，Rabin 提出了一种变形 RSA 体制，并证明了破译改体制等价于解决整数分解问题。
（1） 密钥生成
每一用户各自互异的素数 p，q，要求

$$p \equiv 3 \bmod 4,\quad q \equiv 3 \bmod 4 \qquad (4-3-7)$$

计算 $N=p*q$，**公布** N，**密藏** p，q。**此外需确定明文空间** $\Omega_N \subset Z_N$，**公布将电文（原始明文，以便与式** 4—3—9 **中的** m **相区别）映射人** Ω_N **的可逆映射** $\boldsymbol{r}$。

（2） 加密

1．**取** $\log_2|\Omega_N|$ **比特长的电文** m，**计算**

$$x = \boldsymbol{r}(m) \in \Omega_N \qquad (4-3-8)$$

2．**计算并发送**

$$c = x^2 \bmod N \qquad (4-3-9)$$

（3） 解密

1．**求出整数** r，s（**常备**）：

$$r \equiv 1 \bmod p \qquad r \equiv 0 \bmod q$$

$$s \equiv 0 \bmod p \qquad s \equiv 1 \bmod q \qquad (4-3-10)$$

2．**求出同余方程**

$$u^2 \equiv c \bmod p$$

$$v^2 \equiv c \bmod q \qquad (4-3-11)$$

的解 $\pm u$，$\pm v$：

$$u = c^{(p+1)/4} \bmod p \qquad v = c^{(q+1)/4} \bmod q \qquad (4-3-12)$$

由此得出方程式的四个解：

$$y = (\pm u) \bullet r + (\pm v) \bullet s \bmod N \qquad (4-3-13)$$

3．**“四择一”：选出唯一属于** Ω_N **的解** y，：**令**

$$x = y \qquad y \in \Omega_N \qquad (4-3-14)$$

4．**恢复原电文** m

$$m = \boldsymbol{r}^{-1}(x)$$

这里，当电文 m **表示某种语言的文章字段时，可以利用语言的冗余度来完成“四择一”的工作。如果** m **是随机比特串，则需要利用明文空间** Ω_N **的某些易于检测的特征来确定唯一解。理论上，可以取明文空间** $\Omega_N = QR_N \subset Z_N^*$，**表示模** N **平方剩余集，即** $QR_N = \{a \mid \exists x \in Z, x^2 \equiv a \bmod N\}$，**也就是说，若存在整数** x，**满足** $x^2 \equiv a \bmod N$，**则**

称 a 模 N 的平方剩余。

§4.4 Merkle－Hellman 背包体制

自 Diffie 和 Hellman 的“密码学的新方向”发表之后，提出了许多种公钥密码体制，刚刚介绍过的 RSA 体制是最为成功的一种。虽然有的体制被认为是不安全的，但其破译方法的研究给人认识公钥密码以诸多启示。Merkle－Hellman 背包体制是最早提出的一种公钥密码体制，设计者选择组合数学中背包问题（knapsack problem）作为设计该体制的理论基础。

（1）背包问题

所谓背包问题是这样的：已知一可装最大重量为 b 的背包，及重量分别为 $a_1, a_2, \cdots a_n$ 的 n 个物品，要求从这个物品中选取若干个正好装满这背包。这问题导致求 $x_i=0$ 或 1，$i=1, 2, \cdots, n$ 使满足

$$\sum_{i=1}^{n} a_i x_i = b, \qquad (4-3-1)$$

其中 $a_1, a_2, \cdots a_n$ 和 b 都是正整数。

背包问题是著名的难题，至今还没有好的求解方法。若对 2^n 种所有可能性进行穷举搜索，当 n 较大，比如 n=100，$2^{100} \approx 1.27 \times 10^{30}$，对其穷举实际是不可能的。算法的复杂性理论已经证明，背包问题是属于 NPC 类问题。必须指出的是并非所有背包问题都没有有效算法。若序列 $a_1, a_2, \cdots a_n$ 满足条件：

$$a_{k+1} > \sum_{i=1}^{k} a_i, \quad 1 \leqslant k < n \qquad (4-3-2)$$

时，有多项式解法，这样的序列成为超递增序列（superincreasing sequence）。

例如：$x_1 + 2x_2 + 4x_3 + 8x_4 + 16x_5 + 32x_6 = 43$

因 43>32，故 $x_6 = 1$，以之代入得：

$x_1 + 2x_2 + 4x_3 + 8x_4 + 16x_5 = 11$，类似容易求得：$x_1 = x_2 = x_4 = x_6 = 1, x_3 = x_5 = 0$

相应的背包问题成为超递增背包问题。

（2）MH 背包体制

求解背包问题的难易程度强烈地依赖于序列 $a_1, a_2, \cdots a_n$ 的选择，除了某些特殊的背包之外，一般的背包问题都认为是难解的，而易解的背包问题，往往可以修改成为难解的背包问题。MH 背包体制的基本思路是：从易解的背包问题出发，经过某种数学变换，将简单

背包问题转换成极难求解的复杂背包问题。对于用于秘密密钥的合法接收者来说，是面对求解简单背包问题，而对不掌握秘密密钥的非法用户来说，则是面对复杂的背包问题。

一　密钥生成

（每一用户 U 各自）选取超递增序列$\{a_i\}_1^n$，模数$M > \sum_{i=1}^{n} a_i$，Z_M^*中一对互逆元W, W^{-1}，$W<M$且 gcd（W，M）=1，计算

$$b_i = Wa_i \bmod M \qquad （4—3—3）$$

将$\{b_i\}_1^n$公布作为公开密钥，密藏$_{\text{fLine}}\{a_i\}_1^n$、$M$。上述用模乘法“搅乱”$a_i$，以隐藏$\{a_i\}_1^n$成为超递增性的做法称为 Merkle—Hellman 变换，可以反复进行多次。

二　加密

用$m = (m_1, m_2, \cdots, m_n)$记 n 比特明文，发信者利用收信者的公开密钥$\{b_i\}_1^n$将其加密成

$$c = \sum_{i=1}^{n} m_i b_i$$

三　解密

收信者首先计算

$$S = W^{-1} c \bmod M$$

然后得

$$\sum_{i=1}^{n} m_i a_i = S$$

这是一个超递增背包问题，可以简单解出$\{m_i\}_1^n$。

例 4.4.1　设 n=6，超递增序列$\{a_i\}_1^6 = \{24,64,128,255,500,999\}$。选取 M=2000，W=69。

注意到 gcd（M，W）=1，可求出$W^{-1} = 29 \bmod M$。按式（4—3—3）求得本用户的公开密钥（背包矢量）

$$\{b_i\}_1^6 = \{Wa_i \bmod M\}_1^6 = \{1312,416,832,1595,500,931\}$$

明文（101010）对应的密文是

$$c=1312+832+500=2644$$

用户解密时，先计算出

$$S = W^{-1} c \bmod M \ =676$$

再解超递增背包问题：

$$48m_1 + 64m_2 + 128m_3 + 255m_4 + 500m_5 + 999m_6 = 676$$

易求得

$$(m_1, m_2, m_3, m_4, m_5, m_6) = (1,0,1,0,1,0)$$

∴明文 m=（101010）

Merkle 和 Hellman 发表背包公钥密码体制后，声称：谁能攻破它奖励 50 美元。虽然它们利用 Merkle—Hellman 变换将超递增的特性隐藏起来，但得到的序列毕竟不是“正宗”的，终究露出了“尾巴”，两年后被 Shamir 抓住并将它破译。Merkle 和 Hellman 又企图通过多次的 Merkle—Hellman 变换试图将它变得彻底一些，他们再次悬赏给破译者。再过两年，新的方法彻底解决了低密度的背包公钥密码问题，MH 背包公钥密码系统受到致命打击。虽然近年来提出了许多修改的和推广的背包公钥体制，但几乎所有的背包系统都被破译了。尽管背包问题是个 NPC 问题，但是构造背包类公钥密码体制时，都只是利用背包问题的一个实例，另外在设置陷门时，其结构和公开密钥都提供了充分多供破译者利用的信息。背包系统及其破译推动了算法研究领域的发展，也为破译公开密钥体制“等价解密密钥存在”的重要启示和“不理会解密密钥直接从密文恢复明文的直接破译”的实例。

§4.5 ElGamal 体制

ElGamal 体制也是当前国际上流行的公钥密码体制。它是 Diffie—Hellman 体制的变形，不仅可以用于加密过程，而且可以用于数字签名，其安全性基于计算离散对数的难度。

产生密钥时，首先选择一个素数 p，两个随机数 g 和 d，g 和 d 都小于 q。然后计算

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

公开密钥是 y、g 和 p。g 和 p 都可以由一组用户共享，秘密密钥是 d。

一、ElGamal 签名

假设 x 是明文信息，为了给它签名首先选择一个随机数 k，k 与（p—1）互素。然后计算

$$a = g^k \bmod p \qquad （4—5—1）$$

再利用欧几里德算法从下式求出 b：

$$x = da + kb \bmod(p-1) \qquad （4—5—2）$$

签名是一队数 a 和 b。随机变量 k 保持秘密。

要验证签名，只要验证

$$y^a a^b \bmod p = g^x \bmod p \qquad （4—5—3）$$

表 4.5.1 总结了这种方法。

例如，选择 p=11，g=2，秘密密钥 d=8，计算

$$y = g^d \bmod p = 2^8 \bmod 11 = 3$$

则公开密钥是 y=3，g=2，和 p=11。
为了给 x=5 签名，首先选择随机数 k=9，验证 k 和（p—1）的公因数 gcd（9，10），计算

$$a = g^k \bmod p = 2^9 \bmod 11 = 6$$

利用欧几里德算法求：

$$x = ad + kb \bmod (p-1)$$

$$= 8 \times 6 + 9 \times b \bmod 10$$

解得 b=3，那么签名就是 a=6 和 b=3。
为验证签名，计算

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

$$3^6 6^3 \bmod 11 = 2^5 \bmod 11$$

由于满足式（4—5—3），故认可签名（参见表 4.5.1）。

表 4.5.1 ElGammal 签名

公开密钥	
p	素数，可由一组用户共享
g	整数，<p，可由一组用户共享
y	$g^d \bmod p$
秘密密钥	
d	整数，<p
签名	
k	秘密，随机选取，与（p—1）互素
a（签名）$= g^x \bmod p$	
b（签名），满足 x=da＋kb mod（p—1）	
验证	
如果 $y^a a^b \bmod p = g^x \bmod p$，则可以认为签名有效。	

二、ElGamal 加密

ElGamal 方案的的一种修改形式可对明文信息进行加密。要加密明文 x，首先选择随机数 k，k 与（p—1）互素。接着计算

$$a = g^k \bmod p \qquad (4—5—4)$$

$$b = y^k x \bmod p \qquad (4—5—5)$$

a 和 b 是密文。注意密文的长度是明文的两倍。
密文和经不保密信道传送到接受方后，接收方计算

$$b / a^d = \bmod p \qquad (4—5—5)$$

将式（4—5—4）代入（4—5—5）得：

$$b / a^d \equiv y^k x / a^d \equiv g^{dk} x / g^{dk} \equiv x \bmod p \qquad (4—5—6)$$

从而恢复出明文 x。表 4.5.2 对 ElGamal 加密方案进行了总结。

表 4.5.2 ElGammal 加密

公开密钥	
p	**素数，可由一组用户共享**
g	**整数，**<p**，可由一组用户共享**
y	$g^d \bmod p$
秘密密钥	
d	**整数，**<p
签名	
k **整数，随机选取，与**（p—1）**互素**	
a（**密文**）$= g^x \bmod p$	
b（**密文**）$= y^k x \bmod p$	
解密	
x（**明文**）$= b / a^d = \bmod p$	

除了上述公钥密码体制之外，还有一种专门用于数字签名的算法 DSA。DSA 是 1991 年美国国家标准与技术研究所提出的用于新的数字签名标准中的数字签名算法（）。这种算法是 ElGamal 算法的变形。DSA 签名过程中最大的计算量是对一个与待签名文件内容无关的参数的计算，因此可以事先计算出这个参数。这样，在允许预处理的情况下，DSA 签名计算就变成一件非常容易的事。这一特点特别适和用于智能卡的应用。

第五章　认证系统

信息系统安全的另一重要方面是防止对手对系统进行主动攻击，如伪造、窜改信息等。认证（authentication）则是防止主动攻击的重要技术，它对于开放环境中的各种信息系统的安全有重要作用。认证的主要目的有二：第一，验证信息的发送者是真正的，而不是冒充的，从为信源识别；第二，验证信息的完整性，在传送过程中未被窜改、重放或延迟等。

目前有关认证的实用技术，主要有：

（1）信息认证（message authentication）。如在通信网中，用户 A 将消息送给用户 B。这里的用户可能是个人、机关团体、处理机等等。用户 B 需要确定收到的消息是否来自 A，而且还要确定来自 A 的消息有没有被别人修改过，有时 A 也需要知道送出的消息是否正确地到达目的地。

（2）身份认证（identity authentication）。通信和数据系统的安全性常取决于能否正确地验证通信或终端用户的个人身份。如机要部门或地区的进入，自动出纳机提款，以及各种计算机资源系统的介入都需要对用户的个人身份进行识别认可。

（3）数字签名（digital signature）。消息认证和身份认证解决了通信双方利害一致条件下防止第三者伪装和破坏问题。当通信双方是竞争对象时，如政治、外交和商业条约或契约的签订，传统上仍采用手书签名，以便在法律上生效。随着计算机通信网络的发展，人们希望通过电子设备实现快速。远距离的交易，数字签名发应运而生，并开始应用于商业通信系统，如电子邮件，电子商务和办公自动化等系统中。

§5.1　消息认证

消息认证是是意定的接收者能够检验收到的消息是否真实的方法。检验内容应包括：①证实报文的信源和宿；②报文内容是否曾收到偶然或有意地窜改；③报文的序号和时间性。总之，消息认证使接收者能识别信息的源、内容的真伪、时间性和意定的信宿。这种认证只在相互通信的双方之间进行，而不允许第三者进行上述认证。认证不一定是实时的，如存储系统或电子邮件系统。

一、防止偶然错误

为了防止人工操作和传输过程中的偶然错误，可采用多次输入和多次传输比较法进行校验，也可采用冗余校验法。

例如，当输入数据时，根据统计有 80%为单个字符错误，相邻换位错误为 6%，此时可以采用素数 11 为模的校验方法。如对数字序列 x=530128，先按各为之和在模 11 下为 1 来增加一位校验。在模 11 的计数中以罗马字 X 表示数字 10。然后对 x 的各位数字自左至右做如下运算。令 z 的最左位为 0，与 x 的相应位数字按模 11 相加后再按模 11 乘以 2，所得的结果作为 x 的下一位数字，再与 x 对应位数字作类似运算，依此类推，直到求出 z 的对应校验数字。将 x 与 z 的对应数字按模 11 相加求出最后的数字数字序列 y。运算过程如下：

								校验位
给定数字序列	x=	5	3	0	1	2	8	4
	z=	0	X	4	8	7	7	8
最终数字序列	y=	5	2	4	9	9	4	1

其中

$$z_i \equiv 2(x_{i+1} + z_{i+1}) \equiv 2y_{i+1} \bmod 11 \qquad (5—1—1)$$

$$y_i \equiv x_i + z_i \bmod 11 \qquad (5—1—2)$$

y 的最低位数字 y_0 实际满足校验关系式

$$5\times2^6 + 2\times2^5 + 4\times2^4 + 9\times2^3 + 9\times2^2 + 4\times2^1 + 1\times2^0 \equiv 0 \bmod 11$$

类似地，y 的最低位数字 y_1 也满足类似关系式，即

$$5\times2^5 + 2\times2^4 + 4\times2^3 + 9\times2^2 + 9\times2^1 + 4\times2^0 \equiv 0 \bmod 11$$

依此类推，y 的各位数字都满足类似关系。y 已知时，可以用类似的方法构造数字序列 z'，而后恢复出原来的数字序列 x，如

x =	5	2	4	9	9	4	1
z'=	0	X	4	8	7	7	8
y =	5	3	0	1	2	8	4

其中各位数字应满足下式：

$$z_i' \equiv 2y_{i+1} \bmod 11 \qquad (5—1—3)$$

$$x_i \equiv y_i - z_i \bmod 11 \qquad (5—1—4)$$

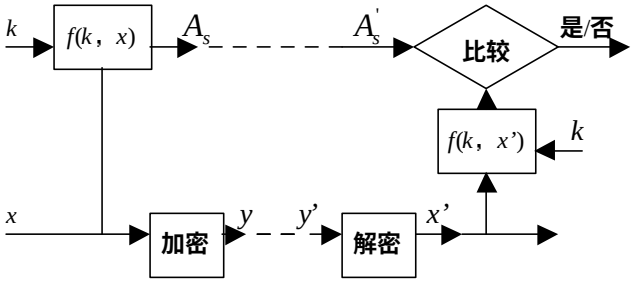
这种校验方法当然不能防止对手伪造合法消息，但可以有效地防止偶然性错误。

采用一致校验检错码是检测传输中由于干扰造成的错误的有效方法。例如对每个长为 7bit 的字符增加一个校验比特，组成长为 8bit 的码字。这是广泛采用的最简单的检错码（ASCII 代码）。CCITT 的 V.41 建议规定通信线路中采用 16bit 的循环冗余校验（cyclic redundancy checks）码，用于线路协议如 HDLC（高级数据链路控制协议）中。这类码可发现所有单个或两比特错误，所有奇数个比特错误，所有长度不超过 16bit 的突发错误，99.997%的长为 17bit 的突发错误和 99.998%的长为 18bit 以上的突发错误。此外，还可用前向纠错技术保证数据传输的可靠性。但这种检错和纠错技术都不能对付伪造数据一类的主动攻击。为此，必须采用密码技术，它所引入的冗余数字不再是公开的简单校验关系，而是含有秘密参量的复杂变换了。

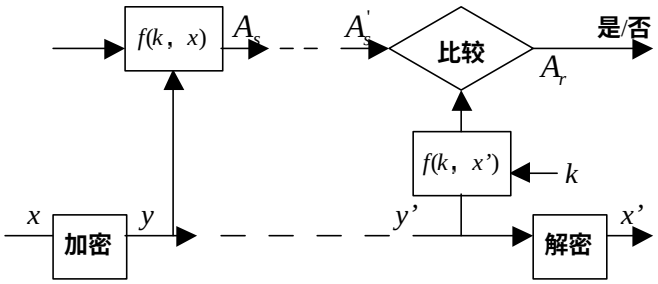
二、消息内容认证的基本方法

消息内容认证的基本方法有二：一是采用消息认证码（Message Authentication Code）简记为 MAC；另一是利用窜改检测码（Manipulation Detection Code），简记作 MDC。MAC 法利用函数 f 和密钥 k 将要发送的明文 x 或密文 y 变换成 r bit 的消息认证码 $f(k, x)$或称其为认证符附加在 x 或 y 之后送出，以 $x \| A_s$ 或 $y \| A_s$ 表示，其中“‖”符号表示表示数字的链接。MDC 法利用一个函数将要发送的明文数据变换成 r bit 的窜改检测码 D_s 附加在明文之后，再一起加密实现保密认证。

接收者收到发送的消息序列后，按照发端同样的方法对接收的数据或解密后数据的前面部分进行计算，得到相应的 r 位数字 A_r 或 D_r，而后与接收恢复的 A_s' 或 D_s' 逐位进行比较，若全相同，就可认为收到的消息是合法的，否则就检出消息中有错或被窜改过。当主动攻击者在不知道密钥的条件下，随机选择 r bit 碰运气，其成功地伪造消息的概率为 2^{-r}。两种消息认证的原理框图在图（5—1—1）和（5—1—2）中给出。

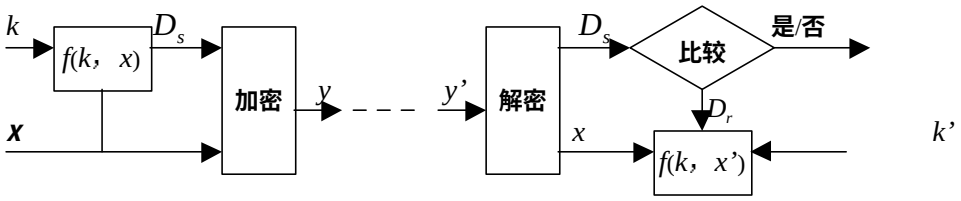


(*a*)



(*b*)

图—1—1 消息认证原理框图 I



图—1—2 消息认证的原理框图 II

图中，函数 f 的作用是对整个消息进行交换，产生一个长度固定但较短的数据组，这一过程可看作是一种压缩编码（compressed encoding）。函数 f 又称为单项杂凑函数（one-way hash function），一般不要求 f 为可逆变换。对函数 f 的选择有如下要求：①f 应对整个消息中比特计算，消息或密钥的任一比特数改变时，所得认证码希望有近一半的比特发生变化，其输出数据长度最好和消息自然分组一致；②对于给定消息 x 和相应的变换 $f(x)$，寻求伪消息 x' 使 $f(x')=f(x)$ 的难度足够大；③对于任意的 x, x_1 和 x_2，有 $f(x_1 \bullet x_2) \neq f(x_1) f(x_2)$，

$f(x) \neq cx, f(x) \neq x^a$，其中 c 为任意常数，a 为任意整数；④易于实现高速计算，硬件实现成本低，或便于软件实现。

对认证算法的要求与加密算法相近，但由于不需要可逆变换而比较容易满足。认证算法需要保密，而且要能经受住已知明文、选择明文攻击。

三、几种 MAC 算法

下面介绍几种实用的 MAC 算法。

十进制移位算法

十进制移位算法（Decimal Shift and Add Algorithm，DSA）特别适用于金融支付中的数值消息交换业务。

消息按十位十进制数字分段处理，不足十位时在右边以 0 补齐，下面举例说明，令 x_1=1583492637 是要认证的第一组消息，令 b_1=5236179902 和 b_2=4893524771 为认证用的密钥。DSA 算法是以 b_1 和 b_2 并行对 x_1 进行计算。先计算 x_1+b_1，x_1+b_2（$\bmod 10^{10}$），而后根据 b_2 的第一位数值控制对 x_1+b_1 循环右移 4 位，记作 $R(4)(x_1+b_1)$ 再与（x_1+b_1）相加得

$$R(4)(x_1+b_1)+(x_1+b_1) \equiv p_1 \bmod 10^{10}$$

类似地，右路在 b_1 的第一位数值控制下运算结果为

$$R(5)(x_1+b_2)+(x_1+b_2) \equiv q_1 \bmod 10^{10}$$

将第二组消息 x_2=5283586900 分别与 p_1 和 q_1 相加，其结果又分别以 b_2 和 b_1 的第二位控制循环移位后进行相加后得到 p_2 和 q_2，依此类推，直至进行十轮后得到 p_{10} 和 q_{10}。计算 $p_{10}+q_{10} \pmod{10^{10}}$，并将结果的后 6 位数与前 4 位数在 $\pmod{10^{10}}$ 下相加取其结果的后 6 位数字作为最后结果，参看表 5—1—1。若被认证的消息长度不足 10 组，则以 0 填充。

由计算过程可知，b_1 和 b_2 既是初始值，又是控制每轮移位次数的密钥。每完成一个认证码的计算之后，b_1 和 b_2 又会到原始次序，重新被用来计算下面十组消息的认证码。

方法的安全性。运算中按模 10^{10} 相加，但循环移位按 $(10^{10}-1)$ 计算。移位 d 次相当于以 $(10^{10}-1)d$ 来改变原来的值，且移位受密钥控制，可以随机地选择密钥。因此，这种算法很难用算术方法分析。而且最后的结果是由两组并行运算结果组合而成，即使知道了最后结果也不容易推断出 b_1 和 b_2。

表 3—1—1

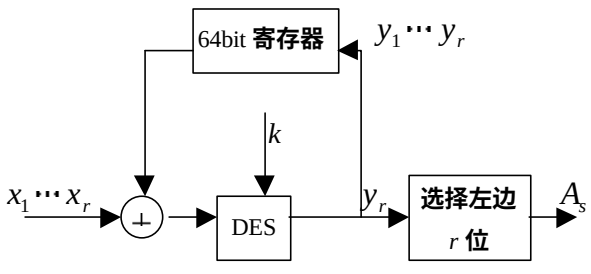
	左路	右路
第一轮	b_1 =5236179902	b_2 =4893524771
	+ x_1 =1583492637	+ x_1 =1583492637
	$b_1 + x_1$ =6819672539	$b_2 + x_1$ =6477017408
	$R(4)(b_1 + x_1)$ =2539681967	$R(5)(b_2 + x_1)$ =1740864770
第二轮	p_1 =9359354506	q_1 =8217882178
	+ x_2 =5283586900	+ x_2 =5283586900
	$p_1 + x_2$ =4642941406	$q_1 + x_2$ =3501469078
	$R(8)(p_1 + x_2)$ =4294140646	$R(2)(q_1 + x_2)$ =7835014690
第十轮	p_2 =8937082052	q_2 =1336483768
	…	…
	p_{10} =7869031447	q_{10} =3408472104
	$p_{10} + q_{10}$ =1277403551	$\bmod 10^{10}$
	403551	
	1277	
认证符	404828	$\bmod 10^{10}$

DSA **算法适用于编程的十进制计算，特别适用于银行业务中的认证系统。**

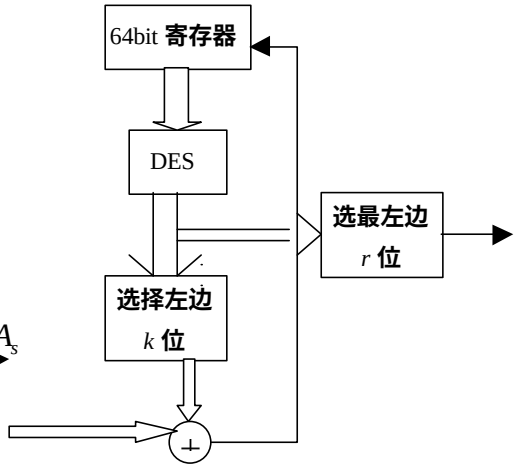
采用 DES 的认证算法

有两种基于 DES **认证算法，一种按** CFB **模式，一种按** CBC **模式进行。在** CBC **模式下，消息按** 64bit **分组，不足时以** 0 **补齐，送入** DES **系统加密，但不输出密文，只取加密结果最左边的** r **位作为认证符，如图—**1**—**3 **所示。图中** $x_1 \cdots x_r$ **和** $y_1 \cdots y_r$ **都是** 64bit **数组。**

在 CFB **模式下，当被认证消息的所有比特送入** DES **加密之后，将最后一组密文反馈到** DES **的输入寄存器运行一次，从所得的结果中选出最右边的** r **位作为认证符，如图—**1**—**3 **所示。** r **取的大小可由通信双方约定，如美国联邦电信建议采用** 24bit**，而美国金融系统采用** 32bit**。**



图—1**—**3 **利用工作于** CBC **模式下** DES **的认证法**



图—1**—**4 **利用工作于** CFB **模式下** DES **的认证法**

这两种模式所得到的认证符都由整个消息决定，因此在适用中必须确定消息始点、终点、长度和认证符的格式。而且认证符还与初始矢量 IV 有关，IV 不公开，它在防止密文第一组进行码本攻击中有用，而对应认证码安全性关系不太大。

四、加密消息认证的MDC 算法

很多情况下希望对消息既加密又认证。在 MAC 下可用两个不同的密钥，一个用于加密，一个用于计算认证符。这种方法可能是最安全的消息认证法。其缺点是要求两次加密，有两个初始矢量，使密钥管理困难些，对于用在 ISO 这类多层次结构中有许多不便之处。在 MDC 下，可先对消息增加某些人为的多余度，即 MDC，而后进行整体加密。其中 MDC 为检错消息在传输过程中是否被窜改提供了保证。下面介绍几种 MAC 的算法。

利用明文的校验和认证

美国联邦标准 1026 草案建议在数据链路层采用明文的校验和作为 MDC。它将消息划分成为 16bit 的字 $x_1 \cdots x_n$，最后一个字不足 16bit 时以 0 补齐。将这些字的相应位分别按模 2 相加后得到一个 16bit 的字段，附在明文之后作为 MDC。即 MDC= $x_1 \oplus x_2 \oplus \cdots \oplus x_n$。然后一起加密成密文。传送过程中若密文被改变，将破坏检验和而能检出。

这种简单方法无论是在 CBC 或 CFB 模式下加密都容易受到攻击。通过同时改变密文组中相应位的偶数个数字、或置换各组之位置、或嵌入满足检验和为零的数组都不会破坏 MDC，而可达到改变明文的目的。

利用模 2 的线性加法构造 MDC

这种线性加法实际上是最高位没有进位的普通加法。它可以防止上述置换和大多数嵌入伪数组的攻击，但不能检错出所有可能的嵌入窜改。例如，在密文前将初始矢量 IV 重复 2^i 次或在两组密文 c_i 和 c_{j+1} 之间嵌入 2^i 组 c_j 时，被检错出来的概率将不是 2^{-r}，而是 $2^{-(r-i)}$。这是由于最高位无进位造成的。因为如果 c_j 的最低位为 1，经过 2^r 次重复加之后，这一比特的影响将被移出此数组之外。克服的方法是选择 r 不大于原始消息的分组长度。如在 CBC 模式的 DES 情况，可选 r =128，实际上如果选 r=80，当消息长度不超过 500k bytes 时，就可满足要求。

五、纯明文消息认证

有时消息本身不需要加密，但对消息的真伪需要认证，如广播系统中，向多个用户传送明文消息，其中可有一个用户备有密钥和相应的设备，可对广播的明文消息进行认证。如果认证失败，就可以发出警报信号通知其它用户。而其它用户无需具备认证能力，这可使设备简化。另外在点－点通信情况下，收端有繁重的处理任务时，不可能对所有消息进行解密，此时可随机选择一些受到的消息进行认证。还有，当以计算机文本作为明文时，如果每次都要解密，将浪费计算机时间。因此不可能将全部程序都加密，但可设置认证字段对程序实现认证检验，以此来保证程序的安全性。

六、消息时间性的认证

消息加密可防止内容泄露，认证符可防止消息的窜改和伪造，但对手可以将截获的消息在原密钥使用期内重新注入到通信线路中进行捣乱。用以前消息代替存储器中的数据，或改变消息序列中消息的前后次序。称这类主动攻击为消息重放，它使消息认证的任务加大。若能将所有接收过的消息都存储起来，并于每一新收到的消息进行比较，就可检出新收到的是否是原来的消息的重放，但这是很不实际的。防止重放的基本原来是对采用同一密钥加密过的每一消息均赋以某个时变参数，使接收这能够检验收到的消息是新的，或将前后发出的消息按时间所形成的消息序列以某种方式联系在一起，使接收者可以检验它们的时间顺序。

在业务交换中的认证总是双向的，各方都要彼此确认。因此，在一般交换消息时，每个消息都是前一个从对方收到消息的一种相应。在短的业务中可能包含有一个、二个或三个消息。因而在双向认证中主要是三个消息的协议。下面介绍几种防重放的方法。

消息流水作业号

经常进行业务活动的两个机构，双方可对要交换的每个消息都加上流水作业号。若前次收到消息的作业号为 $n-1$，则当前收到的消息的作业号应为 n。当检验收到的作业号为 n 时就认可。这种方法可以检验出消息的重放和消息时序的改变。

链接认证符

设第次业务联系时甲向乙送出消息 x_1、业务流水号 n 以及 n 和 x_1 的认证符 A_1，以 $x_1 \| n \| A_1$ 表示。乙收到后检验 n 及 A_1 是否成立，若成立，乙就向甲送 $x_2 \| A_1 \| A_2$，其中 A_2 是由 x_2 和 A_1 计算的认证符。甲检验收到的 A_1 和 A_2 是否成立，若成立，则向乙送 $x_3 \| A_2 \| A_3$，其中 A_3 是由 x_3 和 A_2 计算的认证符，依此类推，直到此次交换业务完成。计算得到的认证符应接近于随机分布且长度足够大，如 16 到 32bit，以使对手难于成功地伪造。

在用 DES 的 CBC 或 CFB 模式下对消息加密时，可用类似的加密方法将消息的最后 64bit 作为下一个消息加密时用的初始矢量。这种消息链接方式下可不占用空间。上述方法中第一消息之后的流水号 n 表示整个这次交易的识别符。如果甲乙双方都可先向对方发消息时，可用一个公用业务流水号或分别用自己的流水号。若采用后者时，在第一次发送的数据中必须包含有作业号，以防止对手以截获的甲的以前业务消息重放给甲伪装作是乙先开始向甲发送业务消息。

通常当更换密钥时，先将流水号置 0，从新计数。若由于干扰或处理出错使双方不能正常认证时，就需要重新同步，为防止对手捣乱，可选的值大于甲、乙双方以前记录的流水号。

随机数认证法

在通信网的多用户的条件下，可以用密钥分配中心送来的会话密钥实现认证。在多用户通信情况下，要求每个用户都保留相互通信中用户之间各个业务流水号是不现实的。为了区分各个消息可采用下述方法。如网中甲、乙双方要进行通信时，在获得会话密钥后，甲可将随机选取的数 R 和消息 x_1 链接后计算出认证符 A_1，并链接成 $x_1 \| R \| A_1$ 送给乙。乙随机选取数 S，将要送给甲的消息 x_2 与 R 和 S 链接后计算认证符 A_2，将 $x_2 \| R \| S \| A_2$ 送给甲，甲检验 R 和 A_2 合格后，由消息 $x_3 \| S \|$ 算出 A_3，将 $x_3 \| S \| A_3$ 送给乙，乙检验 A_3 和

S 后才决定是否认可 x_1。这是三个消息的传送过程。对 R 和 S 的检验保证了消息的时间性和顺序。根据“生日问题”理论，当业务量为 N 次时，随机数应选择大于 N^2。这种方法的缺点是使传输字段长度增大。

时戳

当业务通信为单个消息时，其安全性较低。此时发端无法对收信者认证，而收信者必须能确定收到的消息不是重放以前的消息。业务流水号虽然能完成此任务，但在多用户通信下不实际。采用时戳，即在发送数据中附上消息送出的日期和时刻，也可起作业流水号的作用。线路交换网和现代分组交换网的时延都不超过一秒钟。因此，除了在数据速率很高的情况外，选择时戳精度在一秒左右就足够了。

时戳所需的比特数可按下述考虑来定。系统的运行时间大都不超过一百年，而年、月、日、时、分、秒、各用二位十进制数字来表示，因此时戳共需要 48bit。当系统提供标准时间时最为方便，否则需要设置晶体时钟来保持收发时间同步。

§5.2 身份认证

通信和数据系统的安全性常取决于能否正确验证通信用户或终端的个人身份，例如银行的自动柜员机（ATM）。过去依靠人工识别的工作逐渐有机器代替。随着信息业务的扩大，要求验证的对象的集合也迅速扩大，因而大大增加了身份认证的复杂性和实现的困难性。例如银行转帐系统可能有上百万用户，若用个人识别号（PIN）需要至少六位十进制数字。若要用个人签名来代替 PIN，须能区分数以百万计的签名。为保证系统有适当的安全性，合法的用户遭拒绝的概率要足够小，而伪造身份成功的概率也要足够小。

设计中除了安全性以外，还要考虑经济性和用户的方便性。

身份验证大致可以分为四类：①个人知道的某种事物，如口令(password)，帐号等；②个人持证(token)，如图章、标志、钥匙、护照等；③个人的特征，如指纹、声纹、手形、视网膜、血型、基因等；④个人的无意动作，如手书签名等。

一、根据已知事物验证身份

口令（密码，护字符）是最被广泛研究和使用的一种身份验证方法。一个大系统的口令的产生、管理和标准化是重要的研究课题。

口令可以分为数字、字母、特殊字符、控制字符等组成的长为 5～8 的字符串。其选择原则为：①易记；②难于被猜中或发现；③抗分析能力强。在实际系统中需要考虑和规定选择方法、使用期限、字符长度、分配和管理以及在计算机系统内的保护等。根据系统对安全水平的要求可有不同的选取。

在一般非保密的联机系统中，多个用户可共用一个口令，这当然容易被泄露。要求的安全性高时，每个用户须分配有专用的口令，系统可以知道哪个用户在联机。用户有可能将其有意泄露给熟人，也可能在使用中无意泄露。为了安全最好是将它记住，不要写在纸上。

当用户少时，每个用户可分有不同的口令，因而识别出口令就实现了个人身份的认证。当用户众多时，如银行系统，就不可能使每个用户得到不相同的口令。此时一个口令可能代表多个用户，识别出口令后还须根据其它附加信息才能完成对身份的认证。为了方便记忆，

银行系统常采用十进制数字作为口令。在发放口令时采用随机选取方式，使用户难以发现号码之间的联系，系统中心则采用存储口令和个人身份的其它有关信息以进行身份验证。
在要求较高的安全性时，可采用随时间而变化的口令，每次接入系统时都采用一个新口令，因而可以防止对手以截获到的口令进行诈骗。这要求用户要很好地保存其备用口令，且系统中心也要安全地存放各用户的口令表。
防止泄露是系统设计和运行的关键问题。一般，口令及其相应在传送过程中均须加密，而且常常要附上业务流水号及时戳等，以抗击重放攻击。为了避免被系统操作员及程序员利用，个人身份和口令都不能以明文形式在系统中心存放。可以用软件进行加密处理。现代 UNIX 系统对用户口令就采用加密方式，以用户个人口令的前 8 个字符作为 DES 体制的密钥，对一个常数进行加密，经过 25 此迭代后，将所得的 64bit 字段变换成 11 个打印出来的字符，存放在系统的字符表中（即使是经过加密的口令，在许多系统中也不允许被普通用户获得）。
分发口令的安全性是是极为重要的一环，必须采用可靠的手段。
为了安全常常限定口令的试验次数。例如在一个 ATM 终端上，一般允许重复送卡和打入 PIN 三次，超过三次，ATM 可自动将卡没收，将卡在中心注册表暂时注销，直到受权用户和系统中心取得联系后才恢复。
二、由个人持证身份认证
持证（token）为个人持有物，左右类似钥匙，用以开启电子设备。用得较多的是磁卡，一种嵌有磁条的塑料卡，磁条上记录有个人信息用于机器读入识别，在各类 ATM 上广泛使用。这类卡常和个人识别号（PIN）一起使用。当然，PIN 最好记忆而不要写出来。
这类卡容易制造，且磁条上记录的数据也不难被转录，因此要设法防止仿制。已发明了许多“安全特征”来改进卡的安全性。如采用水印花纹磁条，制造过程中在磁条上加了永久的不可擦掉的记录，难于仿制，用以区分真伪。
有源卡（active card），又称灵巧卡（smart card）或智能卡（intelligent card）。它将微处理器芯片嵌在塑卡上代替无源存储磁条。存储信息远远大于磁条的 250byte，且有处理功能。卡上的处理器有 4k byte 的小容量 EPROM，有的甚至有液晶显示和对话功能。
以智能卡代替无源卡使其安全性大大提高，因为对手难以改变或读出卡中的存数。它还克服了普通信用卡的一些严重缺点，如透支问题、转录信息及泄露口令等。
在智能卡上有一存储用户永久性信息的 ROM，在断电下信息不会消失。每次使用卡进行的交易额和支出总额都被记录下来，因而可确保不能超支。卡上的中央处理器对输入、输出数据进行处理。卡中存储器的某些部分信息只由发卡公司掌握和控制。通过中央处理器，智能卡本身就可检检测用卡人所提供任何暗号，并将它同存储于秘密区的正确暗号进行比较，并将结果输入到读卡机上显示出来。持卡人可以自行设计暗号，不必让发卡公司知道它。将暗号输入到卡的秘密区中，秘密区还存有持卡人的收支帐目、由公司选定的卡的编号及一组字母或数字，用以确定其合法性。存储器的公开区存有持卡人姓名、住址、电话号码和帐号，任何读卡机都可读出这些数据，但不能改变它。系统的中央处理机也不会改变公开区内的任何信息。
智能卡的存储容量和处理功能的进一步加强，会使它成为身份认证的一种更为变通的工具，可进一步扩大其应用范围，如作护照、电话电视等计费卡、个人履历记录、电子锁的开锁钥匙等。不久，个人签字、指纹、视网膜图样等信息都会存入智能卡，成为身份认证的更有效手段。
智能卡的管理系统的安全设计，如卡丢失后的安全保障及补发、防伪造等是实用中要解决的重要研究课题。

三、按个人特征验证身份

在安全性要求较高的系统，由口令和持证等所提供的安全保障不够完善。口令可能被泄露，证件可能丢失或伪造。更高级的身份验证是根据被授权用户的个人特征来进行的确证，它是一种可信度高而又难于伪造的验证方法。这种方法在刑事案件侦破中早就采用了。自 1870 年开始沿用了 40 年的法国 Bertillon 体制对人的前臂、手指长度、身高、足长等进行测试，根据人体测量学(anthropometry)进行身份验证。这比指纹还精确。使用中还未发现两个人的数值是完全相同的。

个人的特征包括很多，有静态的和动态的，如容貌、肤色、发长、身材、姿式、手印、指纹、脚印、唇印、颅相、口音、脚步声、体味、视网膜、血型、遗传因子 DNA、笔迹、习惯性签字，以及在外界刺激下的反应等。当然采用哪种方式还要为被验证者所接受。有些检验项目如唇印、足印等虽然鉴别率很高，但难于为人们接受而不能广泛使用。有些可由人工鉴别，有些则须借助仪器，当然不是所有场合都能采用。这类物理鉴别还可与告警装置配合使用，可作为一种“诱陷模式”（entrapment module）在重要入口进行接入控制，使敌手的风险加大。

个人的特征会随时间变化。验证设备须有一定的容差。容差太小可能使系统经常不能正确认出合法用户，造成虚警概率过大；若容差太太又难于区分用户，使非法用户成功入侵的机会加大，使得漏报概率加大。实际系统设计中要在这两者之间作最佳折衷选择。

下面介绍几种研究较多而又有实用价值的身份验证体制。

手书签字认证　　传统的协议、契约等都以手书签字生效。发生争执时则由法庭判决，一般都要经过专家鉴定。由于签字动作和字迹具有强烈的个性而可作为身份验证的可靠依据。由于形势发展的需要，机器自动识别手书签字的研究得到了广泛的重视，成为模式识别中的重要研究课题之一。机器识别的任务有二，一是签字的文字含义，二是手书的字迹风格。后者对于身份验证尤为重要。识别可从已有的手迹和签字过程的个人动作特征出发来实现。前者为静态识别，后者为动态识别。静态验证根据字迹的比例、斜的角度、整个签字布局及字母形态等。动态验证是根据实时签字过程进行证实。这要测量和分析书写时的节奏、笔划顺序、轻重、断点次数、环、拐点、斜率、速度、加速度等特征。

可能的伪造签字类型有二，一是不知真迹时按得到的信息（如银行支票上印的名字）随手签的字，另一是已知其真迹时的模仿签字或映描签字。前者比较容易识别，而后者的识别就困难得多。

签字系统作为接人控制设备的组成部分时，应先让用户书写几个签名进行分析，适当的参数存档备用。对于个别签字一致性极差的人要采用特殊方法对待，如采用容错值较大的准则处理。

指纹验证　　指纹验证早就用于契约签证和侦察破案。由于没有两个人的皮肤纹路图样完全相同，而且它不随时间而变化其形状，提取指纹作为永久记录存档又极为方便，这使它成为进行身份验证的可靠手段。每个指头的纹路可分为两大类，即环状和涡状，每类又根据其细节和分叉等分成 50~200 叫个不同的图样。通常由专家来进行指纹鉴别。近来，许多国家都在研究计算机自动识别指纹图样。将指纹验证作为接人控制手段会大大提高其安全性和可靠性。但由于指纹验证常和犯罪联系在一起，人们从心理上不愿接受按指纹，此外，这种机器识别指纹的成本目前还很高，所以还未能广泛地用在一般系统中。

语音验证　　每个人的说话声音都各有其特点，人对于语音的识别能力是很强的，即使在强干扰下，也能分辨出某个熟人的话音。在军事和商业通信中常常靠听对方的语音实现个人身份验证。长期以来，人们一直在研究用机器实现识别说话人的想法。这种技术有广泛的应用，其一就是用于个人身份验证。例如，可将由每个人讲的一个短语所分析出来的全

部特征参数存贮起来，如果每个人的参数都不完全相同就可实现身份验证。这种存贮的语音称作为语声纹（voice print）。

视网膜图样验证　人的视网膜血管的图样，即视网膜脉络具有良好的个人特征。这种识别系统已在研制中。其基本方法是利用光学和电子仪器将视网膜血管图样记录下来，根据对图样的节点和分支的检测结果进行分类识别。被识别人必须合作允许采样。研究表明，识别验证的效果相当好。如果注册人数小于 200 万时，其 I 型和 E 型错误率都为 0，所需时间为秒级，在要求可靠性高的场合可以发挥作用。其成本比较高。

脸型验证 有人设计出了从照片识别人脸轮廓的验证系统。对 100 个“好”的对象识别结果正确率达百分之百。但对差的对象的识别要困难得多，要求更细细地实验。对于不加选择的对象集合的身份验证几乎可达到的完全正确。这一研究还可扩展到对人耳形状的识别，而且结果令人鼓舞，可作为司法部门的有力辅助工具。

四、身份证实系统的设计

选择和设计实用身份证实系统是不容易的。分析比较语音、手书签字和指纹三种身份证实系统的性能表明选择评价这类系统的复杂性，需要从很多方面进行研究。美国 NBS 提出了下述 12 个需要考虑的问题：（1）抗欺诈能力；（2）伪造容易程度；（3）对于设陷的敏感性；（4）完成识别的时间；（5）方便用户；（6）识别设备；（7）设备使用目的所需的接口；（8）更新所需时间和工作量；（9）为支持验证过程系统的处理工作；（10）可靠性和可维护性；（11）防护器材费用；（12）分配和后勤支援。

总之，要考虑三个方面问题，一是作为安全设备的系统强度；二是对用户的接受性；三是系统的成本。

§5.3 数字签名

政治、军事和外交签署文件、命令和条约，商业上签订契约和合同，过去都采用手书签名或印鉴。签名起到认证、核准和生效作用。随着信息时代的来临，人们希望通过数字通信网进行迅速、远距离的贸易合同签名，数字（电子）签名法应运而生，并开始应用于商业通信系统。

类似于手书签名，数字签名也应满足以下要求：①收方能够确认对方的签名，但不能伪造。简记为 R1一条件。②发方发出签名的消息给收方后，就不能再否认他所签发的消息。简记为 S一条件。③收方对已收到的签名消息不能否认，即有收报认证。简记为 R2一条件。④第三者可以确认收发双方之间的消息传送，但不能伪造这一过程。简记为 T一条件。

数字签名与手书签名的区别在于，手书签名是模拟的，且因人而异。数字签名是 0 和 1 的数字串，因消息而异，数字签名与消息认证的区别在于，消息认证使收方能验证消息发送者及所发消息是否被窜改过。当收发者之间有利害冲突时，单纯用消息认证技术就无法解决他们之间的纠纷，此时必须借助于满足前述要求的数字签名技术。

为了实现签名目的，发方必须向收方提供足够的非保密信息，一般使其能验证消息的签名。但又不能泄露用于产生签名的机密信息，以防止他人伪造签名。因此，签名者和证实者可公用的信息不能太多。任何一种产生签名的算法或函数都应当提供这两种信息，而且从公开的信息很难推测出用于签名的机密信息。再有，任何一种数字签名的实现都有赖于仔细设计的通信协议。

数字签名的格式有两种，一种是经过密码变换的被签消息整体，一种是附加在被签名消息

之后或某一特定位置上的一段签名图样。

一、利用双钥体制的数字签名

双钥体制有两个密钥，一个是加密密钥，是公开的，以k_1表示，给定明文 x，有加密变换 E 易于求出相应的密文

$$y = E_{k_1}(x) \qquad (一3一1)$$

另一个密钥为解密密钥，是秘密的，只有选择它的用户知道，以k_2表示。在已知k_2下，易于有密文恢复出明文x，即

$$x = D_{k2}(E_{k_1}(x)) = D_{k1}(y) \qquad (一3一2)$$

但当不知道k_2时，要想从y找出明文是极为困难的。

双钥体制含有收发公用的信息，即公钥k_1和加密函数E_{k1}及明文取值范围，它又含有仅为某一用户所拥有的秘密信息k_2，而且从公开信息要推出秘密信息在计算上是不可行的。所以他它正好满足签名系统的要求。

假定用户 A 要向用户 B 发送签名消息。A 选择k_{A1}作为公开密钥在公钥本上公布，选择相应的秘密钥k_{A2}。A 要发送消息 x，就用秘密钥k_{A2}对 x 进行解密变换得

$$y = D_{kA2}(x) \qquad (一3一3)$$

当 B 收到y，可用 A 的公开密钥k_{A1}进行计算得

$$x = E_{kA1}(y) = E_{kA1}(D_{kA2}(x)) \qquad (一3一4)$$

由于k_{A2}是保密的，所以任何其它人不能伪装 A 发送假消息。又由于k_{A1}是公开的，所以任何其它用户均可用它来鉴别收到的y是否用户 A 发出的。这种签名法所得到的结果与所有明文 x 有关，即签名信息含于整个密文之中，而且不可能在保持签名 y 不变条件下改变用来计算它的明文 x，所以这种签名也是对消息 x 本身的验证。

要使收方能够检验发方送来的消息$x = E_{kA1}(y)$，除了有逆变换D_{kA1}外，还应当满足当敌人以假的k'_{A2}对假消息x'进行签名所得到的$y' = D_{kA2}(x')$在E_{kA1}变换下得到的结果，$E_{kA1}(y')$是有意义的明文 Q 的概率极小。这要求明文消息要有足够的冗余度。这是能够检验签名和消息认证是否成立的基本依据。如果原始消息的冗余度很小，就需要认为地增加些冗余度。实际通信中在原始消息前都要加上对方的识别符<IDS>、收方的识别符<IDR>及报文的流水号<i>等。令 Q=<IDS><IDR><i>的总长为 r bit，如果双钥体制是一个良好的伪随机数产生器，当敌人伪造签名消息时，收方以E_{kA1}检验一个正确的 Q 值的概率约为

$1/2^r$。选择足够大的 r 值就可以挫败这类主动攻击。

这种签名法可以实现 R1—，S—及 T—条件。因为第三者可以按收方的方法进行检验。为了实现 R2—条件需要收方向发方签署已收到报文的回执。这种签名法尤其适用于广播通信软件和软件认证系统中对于广播的消息和发行的软件进行签名，以防止别人窜改和伪造。这种签名法对所签的消息不能保密，如果需要保密时，须先对消息进行加密，然后对加密的消息进行签名。公钥签名法的密钥可以多次使用，一般无需第三者参与公证。但若发方想抵赖时，他可能故意将秘密钥公开，或宣布密钥已丢失或被盗，以诬赖别人发了该消息。为了防止这点，被签消息中应加上时戳、路由等信息，而且在丢失秘密钥时必须向权威机构报告注销。

下面简单介绍几种公钥签名技术。

RSA 签名法

这是利用 Rivest，Shamir 和 Adelman1978 年提出的一种双钥密码体制的数字签名法。它的公钥为大整数 m 的一个公钥加密指数 e，其秘密解密钥指数为 d。若对消息 x 签名，则计算：

$$y = x^d \bmod m \qquad (—3—5)$$

式中，$m=pq$，p 和 q 是两个大素数。则 m 的欧拉函数 $\varphi(m) = (p-1)(q-1)$。由用户随机选择的整数 e 满足 $(\varphi(m), e) = 1$，而秘密指数 d 为

$$d = e^{-1} \bmod \varphi(m) \qquad (—3—6)$$

可通过下式检验签名

$$(y)^d = x \bmod m \qquad (—3—7)$$

这种体制的安全性依赖于 $m = p_1 p_2$ 分解的困难性。对于给定的大整数 m，分解它所须的计算量为

$$\sqrt{\ln(m)\ln\ln(m)} \qquad (—3—8)$$

RSA 体制是双射的变换，因而没有消息扩展。其缺点是计算复杂。每个用户需选不同的模，因而造成各用户的数据包长度不一，须要重新分组。例如，若用户 A 和用户 B 相互加密签名通信，A 选用模 m_A 为公开 k_{eA}，保护 k_{dA}，B 以 m_B 为模，公开 k_{eB}，保护 k_{dB}，**则 A 的签名报文将在 1，2，…，$m_A - 1$ 范围内，而 B 的签名报文在 1，2，…，m_{B-1}** 范围内。若 $m_b > m_A$，采用先对消息签名而后加密，当 A→B 通信时不会发生困难；但当 B→A 通信时，由 k_{dB} 进行签署的较长码组须先重新划分成较短码组后，再用 k_{eA} 加密。解决这一困难的办法是，在某一团体内部的通信系统中，所有密钥都用同一个模。但这会是

安全性降低。一些用户可以凑在一起，用各自的（$de-1$）数组的最小公倍数 λ 来攻击另一些用户的密码。

OSS 签名法

1983 年 Schnorr 提出了一种基于二次型的签名法，Ong 和 Shamir 对它进行了改进，故称之为 OSS 签名法。

另模 $m=pq$，$p\neq q$ 且均为大素数。随机地选择 u 作为秘密钥，其中（u，m）=1，$1<u<m$。若 $(u,m)\neq 1$，则 $u=p$ 或 q，从而暴露了 m 的分解信息。选择

$$k\equiv\frac{-1}{u^2}\bmod m \qquad （一3一9）$$

作为公钥，对每个待签名消息 x 都随机选择一整数 r，（r，m）=1，并计算

$$s\equiv\frac{1}{2}(\frac{x}{r}+r)\bmod m \qquad （一3一10）$$

$$t\equiv\frac{u}{2}\bullet(\frac{x}{r}-r)\bmod m \qquad （一3一11）$$

作为对消息 x 的签名。由式（一3一9）和（一3一11）不难验证签名的公式

$$x=s^2+kt^2\bmod m \qquad （一3一12）$$

显然，消息 x 和 m 应互素，即 x 不能取 p 为或 q，当 m 很大时，这一事件出现的概率极低，故可以忽略。

这一算法的安全性取决于 m 的分解或在模 m 下求解平方根的困难性。至于构造一个成立的伪签名是否和大整数因子分解一样地难还为未证明。这一算法可以在普通微机上几秒钟实现，因而可和 RSA 法竞争，但它不适用于作加密用，其安全性也有待于进一步验证。

其它双钥签名法

大多数双钥体制都可象 RSA 体制那样用于数字签名。但 MH 型背包体制有数据扩展，由明文空间到密文空间的映射是不连续的，所以若先对消息空间施行解密变换时不一定有解，因而不能直接用于数字签名。其它公钥签名法有 R一签名法[Rabin 1978]，GMY 一签名法[Goldwasser 1983]，W一签名法[ElGamal 1984]等等。

二、利用单钥体制的数字签名

人们对双钥体制安全性的依赖程度尚不如对广泛使用的单钥体制。所以尽管已有多种双钥体制的签名法，人们还是想用单钥密码体制实现数字签名。一般单钥体制的运算简单，可以高速实现，且硬件实现也较方便，但其严重缺点是数据扩展太大。

一般单钥密码体制，在已知明文一密文或选择明文攻击下要推出密钥是相当难的，因此，利用这种体制可以构造出适用于签名用的单向函数。例如，令明文为 m，密文为 c，密钥为 k，加密运算为 E_k，即 $c=E_k(m)$，若由 c 和 m 难于求得密钥 k，则选择一个常数 a 用密钥 x 加密不难得到 $y=E_k(a)$，由此可定义一单向函数 $f(x)=y$。在已知 y 时要推出 x 就等价于已知明文 a 和条件 y 下确定密钥了。

Lamport-Diffie 签名法[Diffie 1976]

此法由 Lamport 和 Diffie 提出，是利用单钥密码体制构造单向函数实现签名的，今以 DES 密码为说明。对任一消息比特 x_1，发端 A 选择两个 56bit 密钥 k 和 K 作为加密钥，并随机

选择 64bit 数据数组u_i和U_i，计算

$$v_i = E_k(u_i) \qquad (3-13)$$

$$V_i = E_K(U_i) \qquad (3-14)$$

其E_k和E_K是以k和K作为密钥用 DES 算法的加密运算，v_i和V_i是相应的 64bit 密文。

发端将（u_i，v_i）和（U_i，V_i）作为验证参数。将v_i和V_i存于公钥本中备考，而将u_i和U_i保密，当$x_i=0$，A 就将u_i发给收端 B，若$x_i=1$就将U_i发给 B。由于加密函数E_k和E_K及v_i和V_i是公开的，B 可根据收到$x_i=0$或 1 对接收的量以k或K进行加密运算得到v_i'或V_i'，与原公钥本上的值进行比较，就可验证 A 对一比特消息x_1的签名。任何第三者也可用同样的办法进行核实。由于事先只有 A 知道u_i和U_i，如果 B 知道了u_i，就证明 A 曾将u_i发给 B，也就意味 A 签署了$x_i=0$。

若 A 想将$x=(x_1,x_2,x\cdots x_L)$发给 B，则 A 需预先准备好 2L 对（u_i，U_i）2L 对（v_i，V_i）值，将 2L 个（v_i，V_i）对公布，将 2L 个（u_i，U_i）对保密，待发送消息 x=（1，0，1，1，……1）时，A 就将 x 及（$u_1,U_2,u_3,U_4\cdots\cdots U_L$对 x 的签名发给 B，B 收到后根据 x 的分量值分别以 k 或 K 对签名分量加密并与公钥本上的v_i和V_i值进行比较即可验证签名。这一方法对 L bit 消息进行签名验证需要 2L 对（u_i，U_i）和 2L 对（v_i，V_i）值，在用 DES 密码时一个比特的签名将需 64×4=256bit 的数据扩展。

Rabin 签名法

Rabin[1978]曾提出一种用单钥密码体制，例如 DES，来实现经过压缩变换后的消息的签名法，如图 X 所示，发方随机选择偶数个 56bit 的 DES 密钥，如k_1，k_2，…k_{36}，消息 m 经过 CE 被压缩成 36bit 数数字$(h_1\cdots h_{36})$，分别用 36 个密钥加密后得s_i（64bit），$s_i=E_{ki}$（h_i），（i=1，2，…，36），将$s=s_1s_2\cdots s_{36}$作为发送消息 m 的签名。

为了使收方能够签名，发方将从 36 个密钥中随机选取一半，即 18 个公布给收方，剩下的一半保密，用于防止收方伪造，收方利用得到的 18 个密钥，可对经由 CE 变换后得到的 36bit 数据中的 18 个进行加密得到 S 的 18 个分量，与收到的签名 S 中相应分量进行比较就可以验证，但他不能产生其余 18 个分量，因而不能伪造发方的签名。第三者也可以判断签名的

真伪。由于发方只公布 18 个密钥，所以别人伪造的签名S'至多可以正确地产生 S 的 18 个分量。另一方面，发方也不能否认自己的签名，为此发方要事先向第三者提供他全部的密码。当发生争执时，若第三者验证签名可有 19 个以上的分量符合，则判收方赢。若只有 18 个以下的分量符合，则判发方胜。

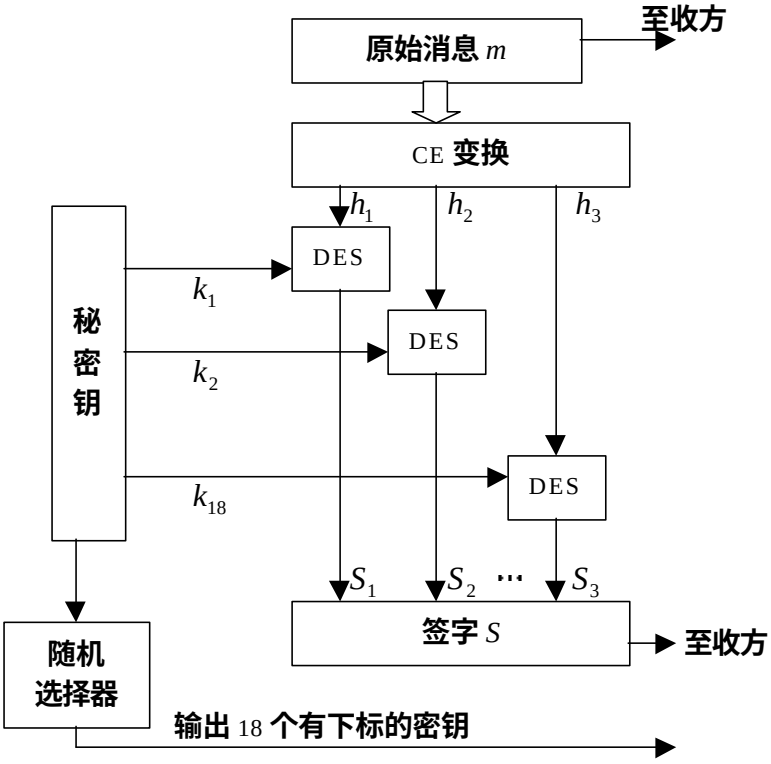


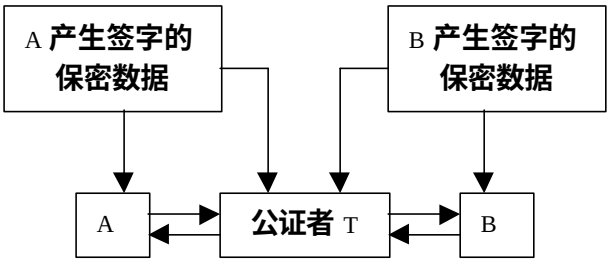
图 X—3—1 Rabin 签名法框图

选择偶数个密钥，并只公布一半的做法使得这一体制对于发方和收方较公平。发方否认签名成功的概率仅为$1/\binom{36}{18}\approx 10^{-10}$。若采用奇数个密钥而公布的密钥数稍过半数时，在发生争执后将对收方稍有利些。

这一方案对每一消息都需要传送 36×34=2304bit 数据，而收方要检验 1008bit 数据。而且 1008bit 的证实还有赖于所有消息数字的正确传送，整个消息 m 和 16 个密钥构成了系统的公钥。密钥都只能使用一次。因此，单钥体制的签名法所需密钥量太大，难于广泛使用，这是单钥体制签名法的致命弱点。

三、公证系统

通常重要协议和合同的签名都有证人参加，或者有证人联名签署文件用以防止伪造和反悔，并以此来提供文件的合法性和认证性。类似地，在数字通信系统中的公证系统（notarzation system）是有仲裁（arbitrator）和公证人（notary）参与的数字签名系统。由于这类系统是基于对第三者即证人的绝对信赖，故可以减少发方和收方共享的有关信息。它既能防止收方伪造签名或否认收到过发给它的签名消息，也能戳穿发方队所签发的签名消息的抵赖。公证人在此系统中起关键作用，因此要认真地选择以保证系统的安全性和可信赖性。建立和保证一个可信赖的公证系统可能要付出较高的代价。图一3一2 给出系统的框图。用户 A 和用户 B 分别与公证者共享他们自己的秘密签名信息。这类系统特别适用于公共事物管理机构下属的有众多用户的计算机网，也有人将它称作认证服务系统（authentication service system），而将公证人 T 称作认证服务员（server）。公证系统可用双钥或单钥秘密体制实现。

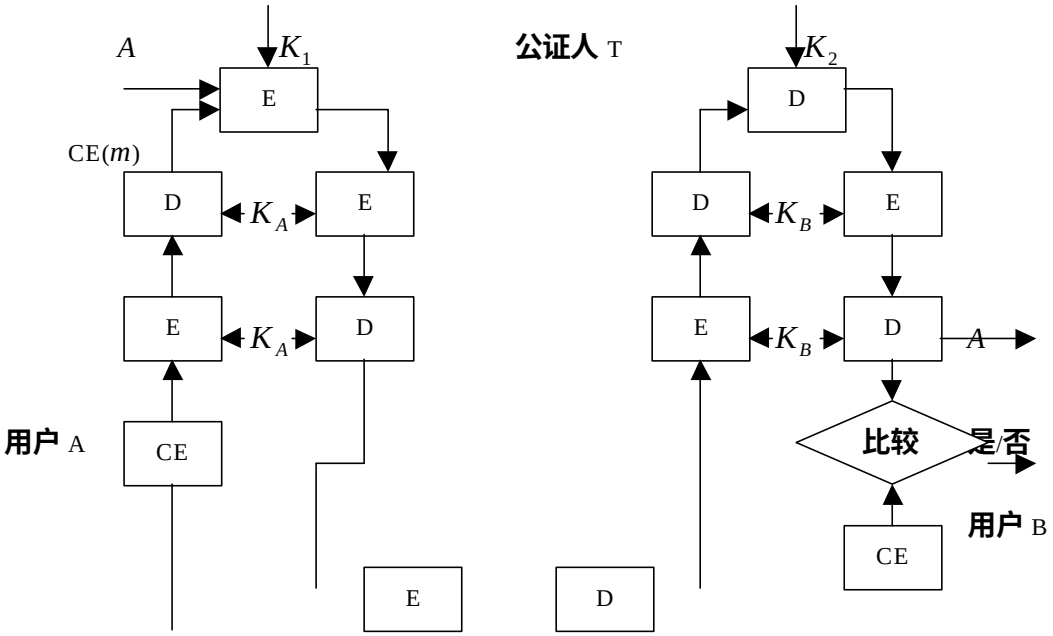


图一3一2 公证系统框图

利用单钥体制的公证系统

在公证系统中收方和发方共享的签名信息很少，从而克服了单钥签名系统要求密钥量大，数据扩展大和必须一次一密钥等缺点。

图一3一3 给出一种用单钥秘密体制实现公证系统的框图。用户 A 和 B 分别和公证人 T 共享的密钥 K_A 和 K_B 用来保证签名和仲裁，而用户 A 和 B 共享的密钥 K_{AB} 用于加密用户 A 给用户 B 的消息。这是为了防止证人阅读所传的信息。公证人 T 用其秘密钥 K_t 实现仲裁。



图—3—3 一种用单钥密码体制实现公证系统的框图

用户 A **向用户** B **送签名消息的过程如下：**

（1）A **先计算** CE(*m*)，**并以** K_A **加密后将** $E_{kA}(CE(m))$ **送给** T。

（2）T **收到** E_{kA}(CE(*m*))，**以** K_A **解密得到** CE(*m*)，**将** A **的识别符** IDA **链接于后，并用** K_t **加密后作为** A **的签名** *S*，**以** K_A **对** *S* **加密后送给** A。

（3）A **以** K_A **将** T **送来的数据解密得** *S*，**并与消息** *m* **链接后，用** K_{AB} **加密后送给** B。

用户 B **检验和接收消息的过程如下：**

（1）**将来自** A **的数据用** K_{AB} **解密得到签名** *S* **和消息** *m*，**以** K_B **对** *S* **加密后送给** T。

（2）T **收到** E_{kB}(*S*)**后以** K_B **解密得到** *S*，**再以** K_t **解密得到** CE(*m*)**和** IDA，**从而证实** A **给** B **的签名。**T **将解密的结果以** K_B **加密后传给** B。

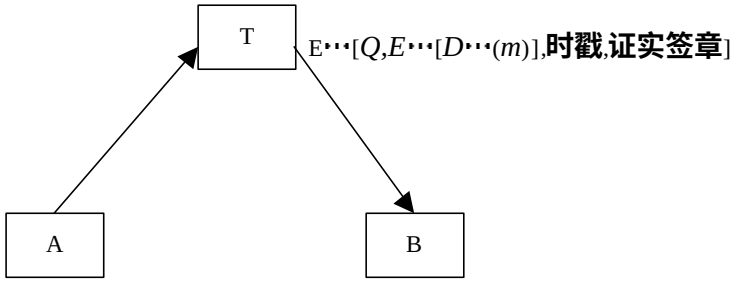
（3）B **将收到的** T **送来的数据以** K_B **解密得** CE(*m*)**和** IDA，**并将** CE(*m*)**与用接收消息计算的制进行比较实现对所收到的消息的认证。**

在这类系统中，如 T **是公证无私的就可以实现对多用户的公证。但不能防止** T **接收贿赂与** A **或** B **联合来欺骗另一用户的问题。这可以采用多个认证服务器的方法来解决。消息可以在多个认证人之间以公用密钥传送。发生争执时用户** A **和用户** B **可以请求不同的公证人出面，或用公证人多数表决方式来解决。**

上述公证体制要完成 A **到** B **的签名消息传送需要经过** 5 **个单向消息传送，任何一路传送收到干扰破坏都会造成认证失。一般发方不希望所传送的签名消息被认为是假的，而收方也不希望拒收一个有合法签名的消息。所以系统还需要预防主动攻击，如增加随机数或作业号等。**

利用双钥体制实现公证系统

双钥体制适用于数字签名，也可用来构成公证签名系统。图—3—4 给出以 RSA **体制实现公证系统的框图。**



令用户 A、**用户** B **和公证人** T **的密钥对分别为** (K_{Ae}, K_{Ad})、(K_{Be}, K_{Bd}) **和** (K_{Te}, K_{Td})。

各密钥对中前一个公开，后一个保密。若用户 A 想将消息 m 送给用户 B，其签名传送过程如下：

（1）A 计算 $E_{kBe}[D_{kAd}(m)]$ 和 $E_{kTe}[Q,E_{kBe},[D_{kAd}(m)]]$，并将后者送给 T。其中 Q 是收发方的认证符和流水作业号等信息。D_{kAd} 保证 A 的签名，而 E_{kBe} 保证对消息 m 的保密，第三者不能读出，E_{kTe} 保证 T 可以实施公证。

（2）T 计算 $E_{kTd}[E_{kTe}[Q,E_{kBe}[D_{kBe}(m)]]]=Q$，$E_{kBe}[D_{kAd}(m)]$，证实由 A 送给 B 的消息 m；而后计算 $E_{kTa}[Q,E_{kBe}[D_{kAd}(m)]]$、时戳，证实签名，并将结果送给 B。

（3）B 以 E_{kTe} 对收到的数据解密得 Q、$E_{kBe}[D_{kAd}(m)]$、时戳及 T 对 A 的签名证实信息，而后以 D_{kBd} 对 $E_{kBe}[D_{kAd}(m)]$ 解密得到 $D_{kAd}(m)$，再以 E_{kAe} 运算就得到消息 m。

这个方案有以下机构特点：①通信之前 A、B、T 之间没有共享信息；②不可能发送时间上不正确的消息，如重放等；③T 不能获取 A 给 B 的消息内容；④T 难于和 A 或 B 中的一方勾结欺骗另一方。