软件加密与解密技术及其应用

实 例 专 辑

中国科学院成都计算机应用研究所情报室

软件加密与解密技术及其应用实例专辑

前　言

随着计算机的广泛应用，高技术不断地相互渗透，人们对软件及数据保护已提到议事日程，为了使计算机应用工作者尽快地掌握熟悉软件加密与解密技术知识，我们曾在1987年编了《微型、小型计算机操作系统应用技术与软件加密技术汇编》一书，受到了广大读者的欢迎，为使软件加密与解密技术资料内容更加丰富、系统、实用性更强。我们又大量搜集了近期最新资料，编制成《软件加密与解密技术及其应用实例专集》一书》该书是对前一套书下册的一个补充，以大量应用实例为主，具有较高的实用价值，是一本难得的内部参考性的情报资料。我们希望它能成为广大计算机应用工作者，获取成果捷径不可少的工具。

该书主要内容有，加密与解密基本原理与技术、方法与实践、文件保护与恢复等。

在资料的汇集中，因时间关系未一一征求作者的意见，有不妥的地方，敬请作者原谅，并对作者积极支持此工作表示衷心感谢。

由于水平有限及时间关系，搜集的资料还不很完全，错误难免，敬请读者指正，

该书在编审的过程中受到了高树清、吴浣尘、罗淳、孟晓玲、刘素琴徐敏如、张薇薇、梁军、郭俊茹等同志的支持表示感谢。

该书由杨明芳、周永培统编。

编辑

88.[illegible].20

《软件加密与解密技术及其应用实例专集》

目　　录

原 理 与 技 术

方 法 与 实 践

文件保护与恢复

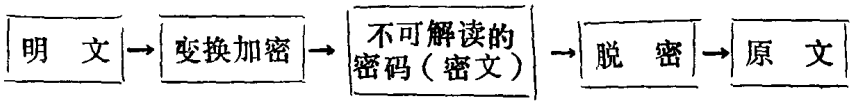
计 算 机 与 现 代 密 码 技 术

复旦大学　　李为鉴　鲍振东

摘要： 本文简要地介绍了加密的基本概念和加密的一般方法。在此基础上较具体地介绍了二进制的加密方法。文中着重介绍的DES体制、RSA体制和渐缩体制都是现在密码技术的研究成果，是在计算机上对信息进行加密的具有代表性的几个加密体制，很有应用价值。

在数据处理中，为了保密起见，可设法将数据进行适当的交换，使之成为“面目全非”的密码信息，这个过程通常称为数据加密，这是数据保密的一种重要手段，有关加密的各种方法不断的出现、讨论十分活跃。特别是计算机用于军事、经济、信贷、储蓄、管理等各个统域后，人们对数据信息的保密要求越来越重视。与之相应的就有一个如何将加密后的信息还原成原来数据的“本来面目”这个过程称为信息脱密。很明显，如果一个加密后的信息，很容易被别人找到脱密的办法，那么，这就是一个价值不大的加密系统，所以，研究加密系领，使得加密后的信息很难被别人找到脱密的办法，就形成一门新的学科——密码学。

我们可将加密到脱密的过程用下面的简图表示：



上述过程所研究的原理，手段和方法，就是密码学的基本内容。

密码学可以追溯到4000年前的象形文字时期。在数千年的密码历史中，真正用密码机进行实际使用，这还仅仅是进入二十世纪以后的事，A·S·Vernam于1920年所发明的电传打字机中的加密方式就可算是最早的“机械化密码”。

加 密 的 一 般 方 法

加密的方法大体上有以下两种：

1．代码法

设明文中所用到的所有词所组成的集合为A，密码字符或数字的集合B。我们确定一种一一对应的关系。

$$f:A\rightleftarrows B$$

对于每一个$a\in A$

$$f(a)=a \qquad b\in$$

且$f^{-1}(b)=a$

加密和脱密就是依靠着这种对应关系所编制的密码词典。显然，用这种方法进行加密，机动性很差、工作效率不高，如果利用电子计算机，那就要占用大量的存储空间以存放该密码词典，因此，这在实际上是被认为不适宜应用于计算机的加密方法。

2．密法

密码法通常有代替法、置换法和乘积密码法。

代替法：将明文中的每个字母用别的字母或符号来代替，从而达到加密的一种方法。

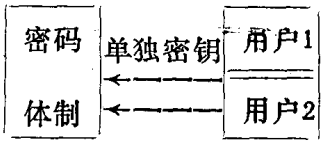
置换法：将明文中的字母的排列顺序进行改变，从而达到加密的一种方法 。

乘积密码法：通过混合采用代替法和置 换法，使明 文变换成 难于破译 的密文的一种方法。

事实表明，特别是在第二次世界大战以后，关于密码方法的研究表明，乘积密码法是一种较可靠的加密方法。

密 码 体 制 的 概 述

如上所述，乘积密码法是混合地使用代替法和置换法所得到的一种比较好的加密方法。用不同的混合法所得的加密方法就形成不同的加密体制。



一般地说，加密体制就是用各种加密方法组合起来所形成的一种算法，并且通过该体制的用户所选定的单独密钥的作用来决定算法的具体实现。

在这种情况下，如果整个算法也能保密的话，这对密码的用户来说，当然是再好也没有了，然而，密码体制是由电子线路、计算机程序来实现的。因此，就密码体制来说，在一般情况下是难以保密的。可见，密码体制的设计必须把注意力集中在这样一点上：使企图破译密码的人，在不知道密钥的情况下，难以实现对密文的破译。这样，保护整个数据的机密，改变成保护密钥的机密就可以了。

密码体制的好坏，应该有一些准则。按照C·E·Shannon的提法，应该具备以下五个准则：

（ 1 ）破译密码需要极大的工作量。

（ 2 ）密钥的长度很小。

（ 3 ）加密和脱密所进行的操作比较简单。

（ 4 ）即使产生错误，错误的扩散也很小。

（ 5 ）信息被加密后并不改变原信息的长度。

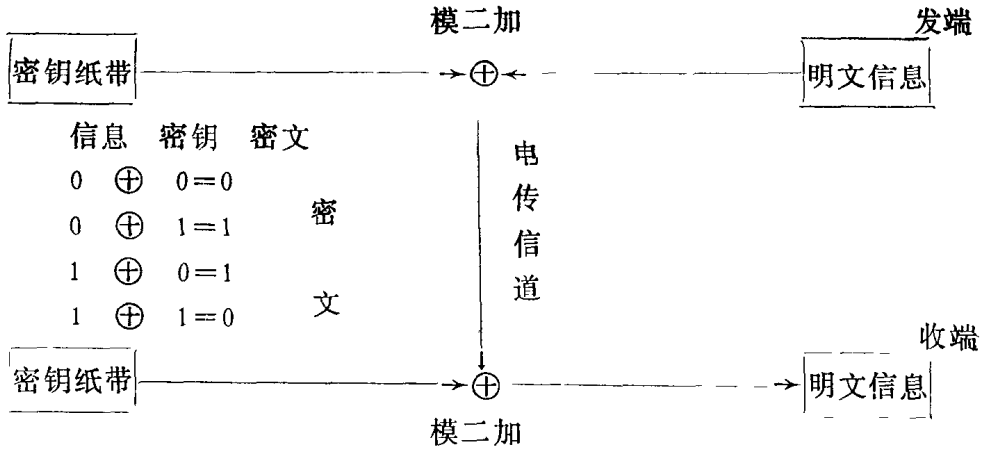
由于现代计算机的发展，算法可以组合到硬件中去，还可以用微编码和微程序，因此，在不降低容量的范围内，算法可以搞得复杂化些。

计算机都是二进制运算的，所以有关二进制信息的加密方法，很自然的引起人们的重视。

早在1920年弗纳姆就提出过一个加密方案，如下图所示：

这个方案的实现要求发端与收端有相同的二进制数码序列的密钥纸带，且双方要保持同步，对于发端来说，将明文信息与密钥序列模二相加后形成密文，并通过电传信道将密文发送出去，对于收端来说，与发端同步，并将收到的密文与密钥序列通过模二加法而得到原来的明文。

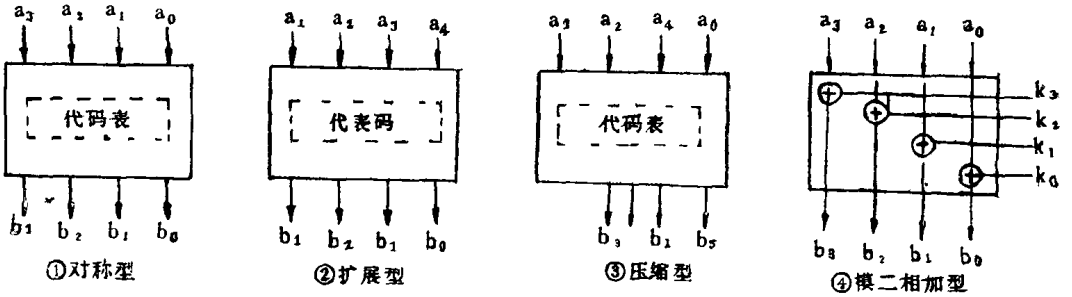
在这个方案中，特定的密钥序列只用一次，因此，即使密文被窃，也是难以破译的，密钥只用 一次，这就称为“一次一密体制”。显然，当明 文信息 量很大时，密钥 序列也就相



应的很长。

对于二进制信息，也可进行代替和置换，简述如下：

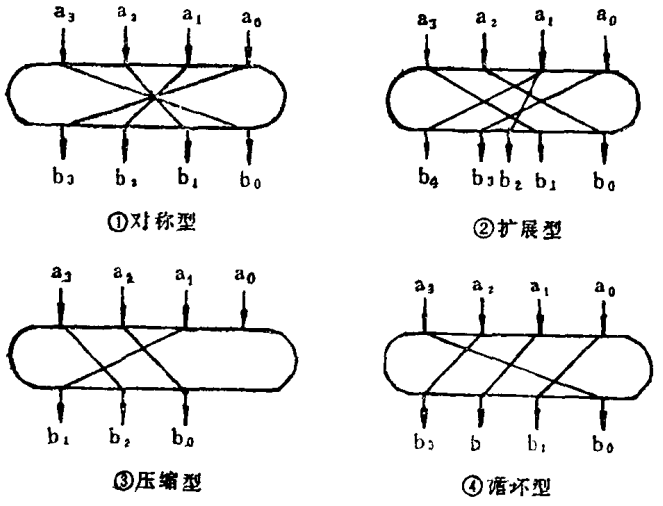
代替：设a为输入，b为输出。



	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
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

只对压缩型举例说明：对于输入（$a_5a_4a_3a_2a_1a_0$）$_2$按照以下的代码表中的（a_5a_0）$_2$行与（$a_4a_3a_2a_1$）$_2$列交义位置上的数的二进制数作为输出。

例如：以（110111）$_2$作为输入，则输出应为：



（11）$_2$＝3行与（1011）$_2$＝11列交义位置上的数14

故输出应为（1110）$_2$0

置换：同样，设a为输入，b为输出

只对扩展型举例说明：由下表可知，将输入的第32位作为输出的第一位；输入的第一位作为输出的第二位；……；输入的第四位即作为输出第五位，又作出输出的第七位；……；输入的第32位作为输出的第47位；输入的第一位作为输出的第48位。

①	^{1}②	^{2}③	^{3}④	^{4}⑤	^{5}⑥
^{4}⑦	^{5}⑧	^{6}⑨	^{7}⑩	^{8}⑪	^{9}⑫
^{8}⑬	^{9}⑭	^{10}⑮	^{11}⑯	^{12}⑰	^{13}⑱
^{12}⑲	^{13}⑳	^{14}㉑	^{15}㉒	^{16}㉓	^{17}㉔
^{16}㉕	^{17}㉖	^{18}㉗	^{19}㉘	^{20}㉙	^{21}㉚
^{20}㉛	^{21}㉜	^{22}㉝	^{23}㉞	^{24}㉟	^{25}㊱
^{24}㊲	^{25}㊳	^{26}㊴	^{27}㊵	^{28}㊶	^{29}㊷
㊸	^{29}㊹	^{30}㊺	^{31}㊻	^{32}㊼	㊽

例如，对于输入

1 0 1 1 1 0 0 1 0 1 0 1 0 0 1 0 0

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17

0 1 0 1 1 1 0 1 1 0 0 1 1 0

18 19 20 21 22 23 24 25 26 27 28 29 30 31

0

32

根据上表，经扩展后将输出

0 1 0 1 0 1 1 1 1 1 1 0 0 1 0 1 0 1 0 10 100

1 0 0 0 0 0 1 0 1 0 1 1 1 0 10 1 1 0 0 1

0 1 1 0 0 1

我们知道，密码体制的要害是具以难以被攻击的特征，因此，一个密码体制，总应该有某种关键所在。设计陷门函数，并在密码体制中应用陷门函数，这就是一个很重要的手段。

所谓陷门函数是指具有以下性质的函数F：

对于定义域X，值域Y，若

（1）计算函数F的算法存在，即对于xj∈X，$F(x_1x_2\cdots x_n)=y$ y∈Y，是容易实现的。

（2）对于几乎所有的y∈Y，要求出xi所需要化费的计算时间或占用的存储空间，实际上都是不可能实现的。

例如，一个大的合整数n，分解它为素因数的问题。在每次运算操作时间为1·—us的机器上，当n＝30位十进制时，约化费3.9小时；当n＝100位时，需74年；当n＝200位时，要用$3.8\times10^{\prime}$年

因此，若X是由100位以下的所有的素数对组成的集合，则对于$(x_1, x_2)\in X$，若定义函数$F(x_1, x_2)=x_1\cdot x_2$，由于一个大的合数分解成两个素数的乘积是一个要化费很多时间的问题，所以，这里所定义的F便可看作是一个陷门函数。

几个具体的加密体制

加密体制有很多，这里就几个具有代表性的加密体制作出介绍。

一、DES体制

DES是Data EucryPtion Standard的缩写，是美国国家标准局的数据加密标准。这个体制于1971—1972初由IBM公司的沃尔特·塔奇曼（W·Tuchman）和卡尔耶迈斯（C·Meyers）研究成功的。在1973—1974年，美国国家标准局曾两次公开征求能适用电子计算机的保密方案，经过挑选采用了DES系统。经过几年的研究和讨论于1977年1月批准，1977年7月15日生效。后来，花了将近17人年的讨论有关共有简捷的方法来攻击。如果要用穷举法来攻击的话，即使一个微秒穷举一个密钥，没有2^{55}个密钥，就要花费约2283年的时间。

DES体制是将加密的信息按每64比特分成组，然后用密钥ki（i=1，2，…，16）经过16次迭代加密运算，从而得到64比特的密文作出输出。

DES体制的加密算法，如图1所示：

这里，作三点说明：

（1）对于f（R，K），采用如下的体制（图2）

（2）这个体制经过如此复杂的16轮操作，其目的就是为的使明文尽可能增大其混乱性和扩散性，使得输出不残存统计规律，以达到破译者不能从反向推算出密钥。

（3）在算法中所使用的密钥K_1—K_{16}。是用户从给定的64bit的主密钥经过一定的运算而产生的，并不是对用户直接给出16种密钥。

由于在主密钥的64bit中，有8bit是用作奇偶检验码，剩下56bit才作为有效密钥使用，因此，用户可选用密钥可有2^{56}种。

因此，使即在给定输入（明文）和输出（密文）的情况下，要想从中推出密钥K_1，K_2，…，K_{16}，必须用穷举法去搜索p_{16}^{56}种情况，这在实际上是难以完成的，尽管对这个系统有不同看法，但是塔奇曼和迈耶斯认为：DES系统在可以预见的将来（约5年到10年时间内）是保密的，用计算机来破译DES密码，将花费极大的费用和力量，因而也将是不现实的。

二、RSA体制（即由Riverst Shamir和Adleman三人提出的体制）

1976年由W.Diffie和M.Hellman[1]在“密码学新方向”中提出了公开密钥密码的设想。所谓“公开”就是加密密钥可以公开，而解密密密钥于以保密。这样，关于密钥分发问题就很方便了，公开密钥就可以像电话号码本那样的公开。譬如，A发给→B，那么A只要查一下密钥本上有关B的加密密钥，然后，A就可按这个公开密钥对明文进行加密后发送给B，最后，B就根据自己的解密密钥而将密文变换成原来的明文。

对于任何一个明文P，如果明文很长的话，可以将它分段，每一段信息N可以看成0到$n-1$的一个数。这里n是一个很大的合数，取数r，将N加密为M，使得$N^r \equiv M \pmod n$。称（r，n）为加密密钥，它可以公开。可以选取一个适当的s，使得$M^s \equiv N \pmod n$，称（s，n）为脱密密钥，将它进行保密。

对于n，r，s的选取，可以如下进行：

1．先取两个非常大的随机素数P，q（p，位）均为几百位且这两个数之间相差近一

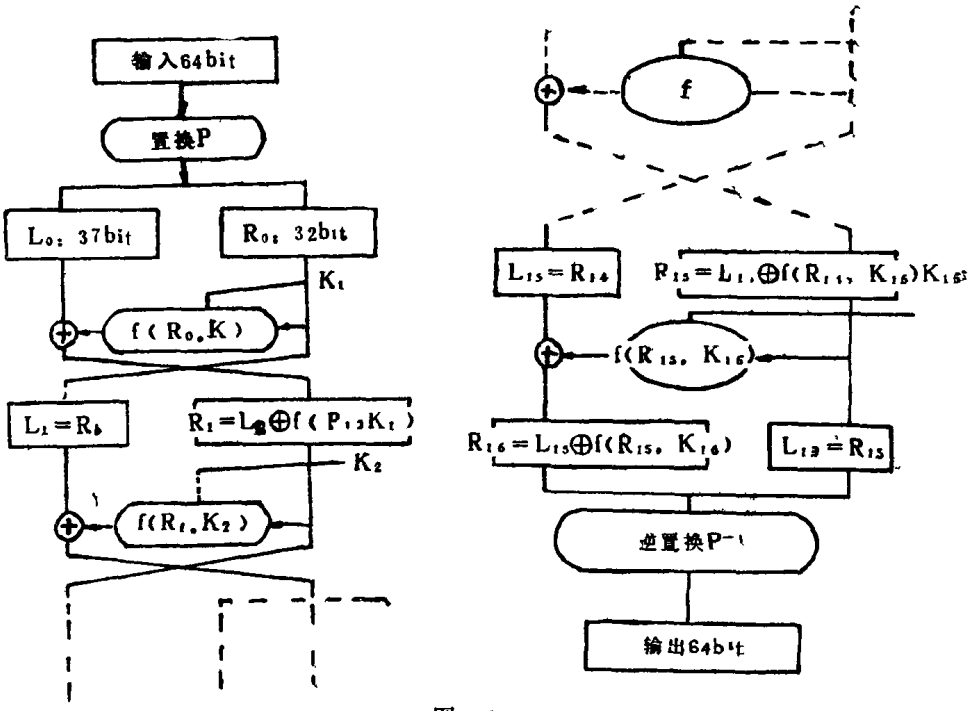


图 1

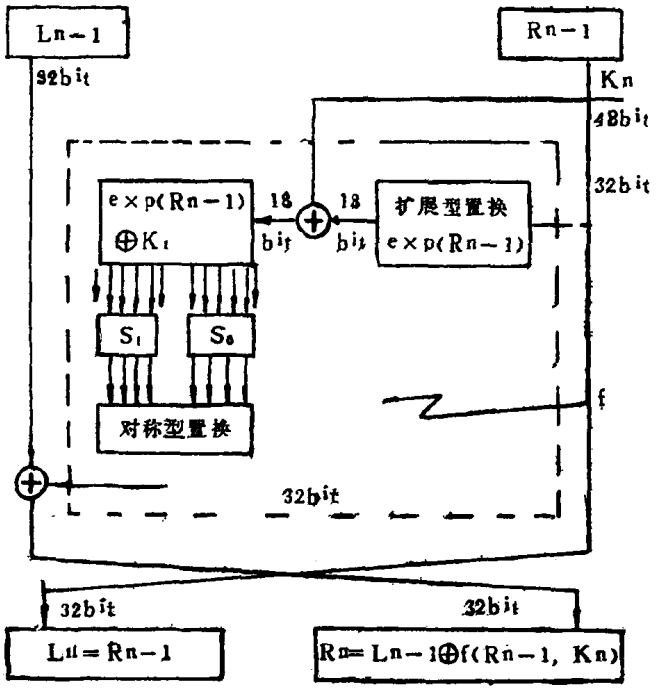


图 2

百，令n=P·q。

2. 随机地取一个大整数r，使它 带 足，GCD(r, P−1)=1，GCD(r,q−1)=1。

3. 由s·r≡1mod(p−1)·(q−1)，求出整数S。

上述办法选取的n,r,s,由于n很大，在不知 道P,q的 情 况下，要从n出 发 进行分解是很难的，因此，在公开(r,n)的情况下，要 求得s是 困难的。这就达到了保密的目的。

现将用（r，n）作为加密密钥，（s，n）作为脱密密钥的算法及其正确性证明如下，

$N^r \equiv M \pmod n$　　加密

$M^s = (N^r) = N^{r \cdot s}$

由于$r \cdot s \equiv 1 \pmod{(p-1)\cdot(q-1)}$

所以$r \cdot s = k \cdot (p-1) \cdot (q=1) + 1$

故　$N^{r \cdot s} = Nk(p-1)\cdot(q-1)+1$

由　Fermat定理可知，$aq^{-1} \equiv 1 \pmod p$

所以$(N^r)^s \equiv N \pmod p$

$(N^r)^s \equiv \pmod q$

最后即得$(N^r)^s \equiv N \pmod n$

1976年，由Bob Blakley和G.R.Blakley发表的长篇文章“Security of Numbe Theoretic Public key Cryptosystems Against Random Attack”防止随机攻击的数论公开密钥密码体制的“保密性”。详细地论述了该体制的理论依据和具体实现。整个体制实质上是RSA体制的推广和发展，关于这个体制的详细论证，可参阅上面提供的文章。现将该体制的具体算法叙述如下：

对于给定的g，w，u

1．随机地选择奇正整数a，使得

$$g-1 < \log_2 a < g + \frac{w}{2} - 1$$

2．对于每个质数$r<u$作

GCD（r，a），GCD（r，2a+1），GCD（a，（u−1）/2）

如果这三者中有一个不是1，则再重做1。

3．判定a与2a+1是否同时为质数，若不是则重复做1，

4．随机地选择奇正整数b，使得

$$g+w-1 < \log_2 b < g + \frac{3}{2}w - 1$$

5．对于每个质数$r<u$，作

GCD（r，b），GCD（r，2b+1），GCD（b，（u−1）/2）

如果这三者中有一个不是1，则返回到4。

6．判定b与2b+1是否同时为质数，若不是则再重做4

7．作GCD（a，b），GCD（a，2b+1）

GCD（2a+1，b），GCD（2a+1，2b+1），若其中一个不等于1，则返回到1。

8．解以下六对同余式

（i）$A \equiv 0 \pmod{2a+1}$

$A \equiv 1 \pmod{2b+1}$

（ii）$B \equiv 1 \pmod{2a+1}$

$B \equiv 0 \pmod{2b+1}$

（iii）$C \equiv 0 \pmod{2a+1}$

$C \equiv -1 \pmod{2b+1}$

(iv) $D\equiv -1 \pmod{2a+1}$

$D\equiv 0 \pmod{2b+1}$

(v) $E\equiv 1 \pmod{2a+1}$

$E\equiv -1 \pmod{2b+1}$

(vi) $F\equiv -1 \pmod{2a+1}$

$F\equiv 1 \pmod{2b+1}$

当这六对同余方程的解都不是发文信息时，就转向9，否则就转向1。

9. 计算 $p=2a+1$，，$q=2b+1$，$m=p\cdot q$

$v=2ab$

10. 由 $u.d\equiv 1 \pmod{v}$ 求出其最小正整数d。

然后，以（u，m）为加密密钥，（d，m）为脱密密钥，那么，很明显有$(x^n)^d - x^{n\cdot d} = x^{k\cdot 2ab+1}$，故有$(x^n)^d \equiv X \pmod{m}$，

为了使得所取的m较大，且p，q相差也较大，所以，就要求使m满足以下的关系式

$2g < \log_2 m < 2g+2w$

如果要求选取的p、q的范围为：

$2^g < p, q < 10\times 2^g$

则有

$2g < \log_2(p\cdot q) < 2\log_2 10 + 2g$

即

$2g < \log_2 m < 2\log_2 10 + 2g$

按上述范围取p和q，那么$w=\log_2 10=3.321\cdots$。称w为宽度。

如果取p、q为两个十进制的100位的质数，那么，以上述范围，可求得$g=328.870\cdots$，称g为规格。

为了保证加密密钥中的u能使下面的不等式成立

$x^n > m$（对于每一个正整数$x\geqq$）

只须取$u>2g+2^w$即可，这是因为

$x^n > x^{2g+2w} > x^{\log_2 m} \geqslant 2^{\log_2 m} = m$

因此，如果取$g=350$，$w=5$，那么，u应取为$u>2g+2w=2\times 350+2\times 5=710$

随着计算机网络的发展，公开密钥密码体制较好地解决了通信用户数量极其庞大时密钥的分发问题，可以与数量不确定的大量用户进行保密通信，这显然是它的一大优点。然而，在具体的实施中，对于公开密钥表的使用和管理能否有可信赖的公正第三者这也是一个实际问题，保密通信总是在互相认识的双方为了保守信息的机密而采用的通信方式，在这种情况下，通信双方之间或几方之间当然是经过周密而又慎重的考虑才确定加密和脱密的办法，因此，有关密钥的分发数量也是相当有限并且受到严格限制的。完全的公开究竟又有多大意义呢？有关这些问题，只能在实际使用的过程中获得满意的解答。

三、渐缩体制

对于给定的正整数a_1，a_2，$\cdots a_n$和S，求使得

$S=a_1x_1+a_2x_2+\cdots+a_nx_n$

成立的{0，1}解（即xiE{0，1/）。这就称为渐缩问题。这个问题当n相当大时，解的

选择方式用穷举法去做的话，共有2^n种不同的方式。因此，对于一般的ai(i=1，2，…，n)来说，渐缩问题是一个NP问题。

当a_1，a_2…，a_n满足以下的关系式时

$$\begin{aligned} a_2 &> a_1 \\ a_3 &> a_1 + a_2 \\ a_4 &> a_1 + a_2 + a_3 \\ &\vdots \\ a_k &> a_1 + a_2 + \cdots + a_{k-1} \end{aligned} \qquad (3.1)$$

渐缩问题的解是容易得到的。

例如：

a_1	a_2	a_3	a_4
1	2	4	6	1

S 19=

因为：19>12　故应取$x_4=1$

19−12= 7

7>6　故应取x_3=

7−6= 1

1<4　故应取x_2= 0

1 = 1　故应取x_1= 1

1−1= 0 正好结束

所以，就解得$x_1=1$，$x_2=0$，$x_3=1$，$x_4=1$

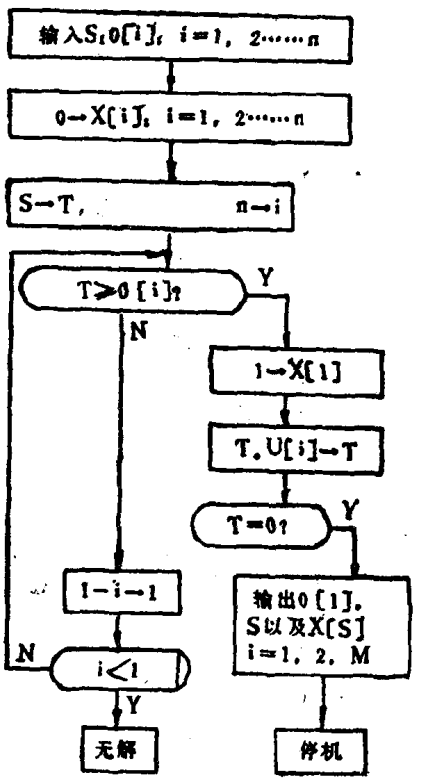


图 3

对于满足（3.1）的渐缩问题，可以有以下的算法。（图3）

在介绍过渐缩问题后，我们再来建立渐缩体制。

设已知ai（i=1，2，…n）满足条件（3.1），任取正整数m满足。

$$m > \sum_{i=1}^{n} ai$$

取得w，使得（w，m）=1，即w与m互质

由$w' \cdot w \equiv 1$（modm）解出w'

作$a_i' \equiv aiwi$　（modm）

对于已知的{0，1}信息x_1，x_2，…x_n，利用a_i'来进行加密，得到密文S

$$S \equiv \sum_{i=1}^{n} a_i' xi \ (\mathrm{mod}m)$$

接收者收到S后，可用以下的方法来脱

$Sw' \equiv S'$　（modm）

因为$S' \equiv Sw' \equiv \sum_{i=1}^{n} a_i w' xi$

$$\equiv \sum_{i=1}^{n} aiww'xi$$

$$\equiv \sum_{i=1}^{n} aixi\,(\mathrm{mod}m)$$

所以，就可以解出x_1，x_2，…，x_n。

在这个体制中，（a′，m）可以分开，w′作为解密密钥。

参考文献（略）

文献出处：《电脑应用时代》1958年 3 期1—8页

多功能磁盘管理软件COPY］［PLU S 4.3

徐 凯 泉

在APPLE机上进行磁盘作业时，为了处理类似拷贝文件，修复DOS,改写扇区，编辑扇区内容，恢复被删除的文件；检查文件错误，查询起始地址，了解磁盘文件空间分布等多种复杂问题，需要自编多个小程序，使用起来很不方便。在这里有必要推广多功能磁盘管理软件COPY］［PLUS 4.3。它能使你在APPLE磁盘作业中得心应手，事半功倍。

该软件共有16项功能，下面将其经常使用的几个部分重点介绍如下：

将含有COPY］［PLUS 4.3文件的磁盘放入驱动器D中，打入BRUN COPY］［PLUS4.3↙，优片刻便出现以下菜单（此时可将该磁盘取出）：

1. CATALOG	9. TRACK/SECTER
2. COPY	10. VIEW FILES
3. BEYCOPY	11. FIX FILE SINES
4. DELETE	12. CHANGE BOOT PROGRAM
5. LOCK UNLOCK	13. UNDELETE FILES
6. RENAME FILES	14. SECTOR EDITOR
7. FORMET	15. NEW DISK INFO
8. VERIFY	16. BOOT DISK

1）CATALOG列出磁盘目录，可列出四种目录供选择：

①NORMAL普通标准目录，相当于原CATALOG命令

②W/FILE LENGTHS以10进制和16进制两种方式列出当前磁 盘上所有文件长度。其中B型文件，同时列出起始地址。

③W/DELETE FILES列出当前磁盘中曾经建立过而又被删去的文件目录

④W/HIDDEN CHERS列出包含有隐藏字符（如∧A、∧D等）的文件名。

2）COPY用来拷贝文件，可有三种形式：

①FILES拷贝单个文件。一次可选择多个文件，拷贝到另一磁盘上。

②DISK将一个磁盘上全部文件拷贝到另一磁盘上。

③DOS将一个磁盘上的DOS系统拷入另一磁盘，而不删去其他文件。这对于有些磁盘DOS损坏不能引导时具有修复功能。

4)5)6)7)部分因使用较为容易，在此不再重复。

8）VERIFY检查磁盘

①DISK按磁道检查全盘。若遇到错误，便显示磁道和扇区号。

②FILES按文件逐个检查。当遇到文件不能调入内存时将显示文件名。

③D-SPEED用来检查磁盘速度，并指出与标准速度之差别。

9）TRACK/SECTOR MAP磁道/扇区图。将显示每个文件在磁盘中所在的位置，可供修改时查找文件位置使用。

10）VIEW FILES显示磁盘中文件所在扇区中的内容。其中①按16进制机器码表示②以文本形式显示。

11）FIX FILESIZES压缩文件在磁盘中所占多余扇区。有些文件在反复修改中，在磁盘上所占实际空间发生了变化，有时文件长度变短，而原有空间较长，因此利用功能可使多余空间得到释放，增加磁盘应有的容量。

12）CHANGE BOOT PROGRAM此功能可用来改变引导程序名。可在整个磁盘中，随意命名任何一个文件名（可以是A型和B型）为引导程序名（只能选一个）。以后当引导该磁盘时，便立即运行新命名的程序。

13）UNDELETE FILES恢复曾被删除的文件。选择此功能。先列出全部被删文件的名称。你可用回车任意指定哪几个文件需要恢复。再按G即可恢复你失去的文件。若原删去的文件所在空间已被新文件占用，系统将告诉你该文件不可恢复。

14）SECTOR EDITOR扇区编辑。选择此功能后，系统显示：

```
] [J] [K] [M] MOVE CURSOR
[B] EGINNING, [E] ND, [R] EAD, [W] RITE
[H] EX OR [T] EXTENTER, [ESC] TO EXIT
[P] ATCH DISK READ WRI TE
```

当我们要修改扇区内容时，可先选[R]，将某一扇区内容读入内存，然后用[I] [J] [K] [M]键来移动光标到需要修改的地方去。再选择[H]按16进制机器码修改（显示于屏幕左边）；若选[T]则用文本形式修改（显示于屏幕右边）。[B]表示将光标移至文首。[E]表示将光标移至文尾。修改结束后按[W]可将已修改的内容存入原扇区。按ESC键即退出该功能。

15）NEW DISIKILFO重新设定初值。通过此功能还可设PRINTER ON状态，对必要的内容进行打印。

16）BOOT DISK退出此软件。在D1驱动器中，插好待起动的软盘，按回车后即可引导。

文献出处：《电子与电脑》1988年 4 期4—5页

数据加密的体制及应用

哈尔滨工业大学　朱志莹　曹珍富

一、引　言

计算机科学和其他科学的交叉渗透，产生了许多离散结构的问题，解决这些离散结构问题所创造的方法，往往产生新的科学分支，突破传统方法的应用范围。

我们熟知，使用计算机的一个基本手段是将所述问题数字化，例如数字信号处理、图象处理等都可归结为数据处理。而数据处理除了数据率庞大需要“压缩”以外，还有数据的加密处理。例如，随着计算机的遍及，“电子信件”将会逐步取代传统的书面信件，人们希望“电子信件”能象传统的书面信件一样，只有合法接收者才能知道信件内容，而任何第三者都只能看到„信封外皮”无法知道信的内容。这就要求对信的内容（可以数字化）提供加密。再加，计算机中重要数据的加密、重要软件的加密等等，都是数据加密的例子。

因为传统的加密方法是严格保密的，不便在更多的场合下使用（往往只能是政府的军事、外交等部门所专用）。1976年11月，美国的Ciffie和Hellman[1]突破了传统的密码技术，提出了数据加密的一种新的体制，该体制要求每一用户都有自己的两种密钥——加密密匙和解密密钥，其中加密密钥公开，便于通信者用来加密明码，而解密密钥严格保密起来。非法接收者想从公开的加密密钥解出解密密钥几乎是不可能的。这是轰动全球的公开密钥体制（public key cryptosystem），寻找这种体制可以归结为寻找所谓的陷门单向函数，本文将依次介绍陷门单向函数以及目前人们已经找到的一些实现方案。

二、陷门单向函数

首先，我们将Diffie和Hellman提出的公开密钥体制用数学语言来描述：

设用户x的加密密钥是E_x，解密密钥是D_x，则要求E_x和D_x实现互逆变换，即

$$D_x \cdot E_x = E_x \cdot D_x = I,$$

其中I是恒等变换，而E_x公开，这里要求两条：①D_x和E_x都容易计算；②仅从公开的E_x寻找D_x是困难的。

现在我们一般地看一下两个用户a和b是怎样实现秘密通信的。设用户Q开送信息P（称为明码）给用户b，则有

（1）a用b公开的加密密钥Eb对P进行变换，得到密码：

$C=Eb(P)$

（2）b收到a发送的C后，用只有他自己知道的解密密钥D_b变换C恢复明码P：

$D_b(D)=D_b(E_b(P))=I(P)=P$

由此可见，寻找适合下面三个条件的函数f（n）是构造公开密钥体制的一个重要途径：

（a）对f（n）的定义域中的每一个n，均存在函数$f^{-1}(m)$，使$f^{-1}(f(n))=n$；

（b）f（n）与$f^{-1}(m)$都容易计算；