第八章 计算机密码学概要

从现在开始我们学习各类典型的网络安全协议，这是实施网络安全最重要和最广泛的技术途径。作为设计和应用各类网络安全协议的基础，计算机密码学起着重要的基础性作用，为此这一章对当代计算机密码学做一个概要性的讨论。本章内容分为两部分，第一部分(8.1节)概要讨论计算机密码学的数学基础，主要是一些在计算机密码学中最常用的数论知识；第二部分概要阐述几类最常用的密码方案及一些典型实例。这一章的目的是使本书的论述自封闭，不过于依赖其他著作，因此对计算机密码学及其数学基础知识的讨论在不失要点的前提下尽量简略，包括许多细节的处理都放在了习题之中(含有一定提示)。但另一方面，本章对很多论题的处理方式与目前大多数教科书不同，对基本概念做了详细解释并给出了许多普通教科书所没有包涵的、但非常有用也非常有趣的实例。最后一节给出了详细的参考书目以满足读者进一步深入学习计算机密码学的需要。

8.1 一些必要的数论知识

这一节概要阐述关于整数性质的一些重要概念和数学事实，它们构成当代各类密码方案安全性质的基础，也是实现各类安全方案的计算基础。

8.1.1 Euclid 定理及线性同余式

Z 表示全体整数的集合。对任何整数 a 和 b，符号(a,b)表示其最高公因子，$a|b$ 表示 b 能被 a 整除，$|a|$表示 a 的位数。发现于 2500 年前的 Euclid 定理是以下事实：

任给整数 a 和 b，必存在唯一的整数 q 和 r 满足 $a=bq+r$ 及 $0\le r<b$。

读者不难自己证明这一事实，实际上这就是小学算术中的带余数的除法。事实上，Euclid 定理确有一个等价的算法性阐述：

存在多项式算法 A，任给整数 a 和 b，A 计算出整数 q 和 r 满足 $a=bq+r$ 及 $0\le r<b$，且 A 的计算复杂度不超过 $\max(|a|^3,|b|^3)$。

尽管看似平凡，但实际上 Euclid 定理的几个等价形式有着广泛的应用。第一个等价形式是以下定理(证明见习题 8-2)：

任给整数 a 和 b，总存在整数 x 和 y(但不唯一)满足 $ax+by=(a,b)$。

注意以上命题的不平凡之处：最高公因子(a,b)系由乘法(即整数的素因子分解)刻画，然而以上定理清楚地表明，最高公因子(a,b)有一个等价的加法刻画。不仅如此，事实上还存在一个多项式算法，习惯上也称为 Euclid 算法，能从 a 和 b 出发计算出整数 x 和 y 满足 $ax+by=(a,b)$，从而计算出(a,b)。特别是，计算两个整数 a 和 b 的最高公因子(a,b)实际上并不需要先分解 a 和 b 的素因子[1]。

为表述 Euclid 定理的第二个等价形式，需要先引进同余的概念和同余符号。任给整数 N、a 和 b，如果存在整数 q 满足 $a=qN+b$，则说“a 和 b 模 N 同余”，记做 $a=b \bmod N$，N 称为模数或模。注意这里并为要求 $0\le b<N$。读者不难验证同余关系是一个等价关系。以下都是同余关系的例子：$11=3 \bmod 8$，$7=7 \bmod 8$，$7=-1 \bmod 8$。关于同余关系的常用性质见习题 8-3。

同余关系导至线性同余式的概念，这是形如 $ax=b \bmod N$ 的数论方程，其中 x 是待求解的未知量。解线性同余式 $ax=b \bmod N$ 就是求这样的整数 $0\le x<N$，使 $ax=b+yN$，其中 y 是某个整数。等价地，解线性同余式 $ax=b \bmod N$ 就是求这样的整数 $0\le x<N$ 使 $N|(ax-b)$。注意不是所有的线性性同余式都可解，例如 $3x=2 \bmod 6$ 就不存在解(为什么？)，即使有解也未必唯一，例如 $4x=2 \bmod 6$ 存在不止一个解 x=2 和 x=5。Euclid 定理的第二个等价形式正是关于线性同余式可解性的命题(证明见习题 8-4)：

任给整数 N、a 和 b，线性同余式 $ax=b \bmod N$ 存在整数解 x 当且仅当$(a,N)|b$，且这时恰有(a,N)个解 $0\le x<N$。特别地，若 a、N 互素则 $ax=b \bmod N$ 总存在且有唯一的整数解 $0\le x<N$。

以上定理还有一个更特殊但更常用的形式：若 a、N 互素则 $ax=1 \bmod N$ 总存在且有唯一的整数解 $0\le x<N$。这个解称为 a 的模 N 的逆，记做 $a^{-1} \bmod N$。另一种特殊但也常用的形式是 N 为素数 p，这时显然 $a=b \bmod p$ 对任何非零系数 a 总存在且有唯一的解 x。

Euclid 定理的第二个等价形式不仅断言解 x 存在，而且实际上可以给出一个多项式复杂度的算法实际计算出 x。从前面对求解线性同余式的解释不难看出，这一算法本质上可以从求 a 和 N 的最高公因子的 Euclid 算法变形而来。

8.1.2 中国剩余定理

考虑以下问题：任意给定两两互素的一组正整数 m_1、…、m_n 和整数 a_1、…、a_n，令 $M=m_1 \ldots m_n$，求整数 x 满足线性同余式组 $x=a_i \bmod m_i$, i=1,…,n。

[1] 实际上也做不到，因为求任意一个整数的素因子分解至今没有找到多项式算法。这就是所谓因子分解难解性假设，是当代一大类密码方案安全性的基础，例如著名的 RSA 加密方案的保密性就依赖于相关的命题。

这一问题在许多领域经常遇见，或作为将复杂问题约化为某些特殊问题的一种技术性步骤而出现。中国剩余定理[2]断言，以上问题的解存在而且在区间[0,M−1]上唯一，不仅如此，这一定理实际上还给出了对 x 的精确的求解公式：

$$x = \sum_{i=1}^{n} a_i M_i N_i \bmod M$$

其中 $M_i=m_1\ldots m_{i-1}m_{i+1}\ldots m_n$, $M_iN_i=1 \bmod m_i$。注意 M_i 和 m_i 互素，因此由关于线性同余方程的 Euclid 定理知 $M_iN_i=1 \bmod m_i$ 必有解 N_i 而且 N_i 唯一。

8.1.3 Fermat 定理和 Euler 定理

Fermat 定理是一个陈述简洁但非常深刻而有用的命题：

若 p 是素数且不整除 a，则 $a^{p-1}=1 \bmod p$。

为了概要解释以上命题为什么正确，我们按照以下步骤进行分析并请读者补足细节。首先注意(参见习题 8-3)只要对 $1\le a\le p-1$ 的 a 证明以上命题即可，为此考虑集合 $Z_p^*=\{1,2,\ldots,p-1\}$，当我们不关心具体元素时也简单地把它的元素记做 a_1、…、a_{p-1}，a 是其中之一。固定 a 而考虑映射 $a_i^*=aa_i \bmod p$，i=1,…,p−1，由于 p 是一个素数故这是一个一一对应(请验证!)，即当元素 a_i 跑遍集合 Z_p^* 时 a_i^* 也恰好跑遍集合 Z_p^*，因此 $a_1^*\ldots a_{p-1}^*= a_1\ldots a_{p-1} \bmod p$，但左面还等于 $a^{p-1}a_1\ldots a_{p-1} \bmod p$，从而 $a^{p-1}a_1\ldots a_{p-1}=a_1\ldots a_{p-1} \bmod p$，也就是(为什么？) $p\mid (a^{p-1}-1)\, a_1\ldots a_{p-1}$，故(为什么？) $p\mid (a^{p-1}-1)\, a_1\ldots a_{p-1}$，也就是 $a^{p-1}=1 \bmod p$。

Fermat 定理有一个非常有用的推广，这就是著名的 Euler 定理，为表达这一定理需要引入重要的 Euler 函数，记做 φ(N)，它的值等于在 1 和 N−1 之间与 N 互素的所有整数的个数，例如 φ(6)=2，因为在 1 和 5 之间与 6 互素的所有整数有 2 个，即 1 和 5；φ(15)=8，因为在 1 和 14 之间与 15 互素的所有整数有 8 个，即 1、2、4、7、8、11、13、14。

不难立刻看出，对任何素数 p 恒有 $\varphi(p)=p-1$。对任何正整数 N，Euler 函数有以下规律(这里不对此加以证明，感兴趣的读者可以参考章末列举的课本，但这些规律很容易记忆，因此希望读者熟练掌握)：

[2] 也称孙子定理，发现于 2500 年前，在《孙子算经》这部伟大的古代数学著作中以一个有趣的具体问题的形式表述并给出了一个非常一般性的求解算法，本质上就是这里所给出的用现代符号所表达的公式。值得指出的是，中国剩余定理属于数学中一类非常“好”的定理，它将一类一般性问题约化/分解为一组特殊问题，使得对一般性问题的理解完全归结为对几类特殊问题的理解，而且这里的约化/分解还可以算法性地实现，这就更为可贵。现代数学中还有一些这种类型的结果，其中读者最熟悉的当属线性代数中关于任意复系数矩阵的 Jordan 结构分解定理，其他重要的结果如有限生成交换群分解为所谓 1 秩自由群(整数群 Z)和循环群的结构定理等。实际上当代对所有这类结果都统称中国剩余定理或孙子定理，以表达世界数学界对中国这位伟大古代数学先哲的纪念和敬意。

$\varphi(N_1N_2)=\varphi(N_1)\varphi(N_2)$，其中 N_1、N_2 互素；

$\varphi(p^m)=p^m-p^{m-1}$，其中 p 是素数、m 是正整数。

注意如果 N_1、N_2 不互素则 $\varphi(N_1N_2)=\varphi(N_1)\varphi(N_2)$未必成立，例如 $4=\varphi(12)\neq\varphi(2)\varphi(6)=2$。具有这种性质的整数函数叫做严格积性函数。由以上公式不难看出，若已知 N 的素因子分解则总可以完全计算出 $\varphi(N)$。一个对计算机密码学影响深刻的性质是，直到目前为止除了先计算 N 的因子分解之外尚没有其它途径能有效计算出来 $\varphi(N)$，正是这一点构成了 RSA 加密方案的保密性的基础，这一点以后就会看到。

回到 Euler 定理，它陈述的事实是：

若整数 a、N 互素，则必有 $a^{\varphi(N)}=1 \bmod N$。

注意当 N 是素数时，Euler 定理就立刻回到了 Fermat 定理(为什么?)。虽然 Euler 定理成立的范围要广泛得多，但其证明与 Fermat 定理的证明本质相同，为此只要考虑集合 $Z^*_N=\{1\le a\le N-1：a \text{ 与 } N \text{ 互素}\}$，将前面对 Fermat 定理的论证结构转移到 Z^*_p 即可，具体细节请读者补足(这是一个熟练应用 Euclid 定理的很好的练习，留做习题 8-11)。

8.1.4 交换群的概念和密码学中常见的例子

读者在离散数学课程中学习过群的概念，这是一个抽象的数学对象，它统一刻画了许多数学对象的代数性质。特别地，正是在这一抽象、统一的基础上，我们才能最清晰地理解前面的 Euler 定理和 Fermat 定理这类看似非常意外的结论为什么确实是普遍正确的。此外，群对当代计算机密码学也有着实质性的应用。

目前密码学中几乎所有重要的群都是所谓有限交换群，为此我们仅局限于这一概念及其普遍性质就足够了，并且简称为群。有限交换群是具有二元算术运算的一个有限集合 G，将这一运算记为*，x,y,z 表示 G 的任意元素，运算*具有以下性质：

任何两个元素*运算的结果仍然是 G 的一个元素，这称为封闭性；

x*y=y*x，这称为交换性；

(x*y)*z=x*(y*z)，这称为结合性；

存在一个元素 e 使 $x*e=e*x=x$ 对任何 x 都成立，这称为单位性，元素 e 称做单位元素；

对任何 x 都存在一个元素，记做 x^{-1}，满足 $x*x^{-1}=x^{-1}*x=e$，这称为可逆性。

G 的元素个数称做群的阶，记号|G|。下面是密码学中常用的群：

(1) N 是正整数，Z^*_N≡{1≤a≤N−1：a 与 N 互素}，+和*分别表示 Z^*_N 上的模 N 加法和模 N 乘法，于是(Z^*_N, *)是群(为什么？)，单位元素是 1，群的阶是 Euler 函数 φ(N)，特别地，若 p 是素数则(Z^*_p,*)是 p−1 阶群。但注意(Z^*_N, +)并不是群(为什么？)。

(2) N 是正整数，QR_N≡{1≤a≤N−1：二次同余式 x^2= a mod N 有解}，*表示 QR_N 上模 N 乘法，(QR^*_N, *)是群，群的单位元素是 1。

(3) p 是素数，F_p≡{0,1,…,p−1}、F^*_p≡F_p\{0}，在 F_p 上定义模 p 的加法运算+，在 F^*_p 上定义模 p 的乘法运算*，于是(F^*_p,*)和(F_p,+)都是群，阶分别为 p−1 和 p(为什么？)。这个例子的特殊之处在于同一个集合 F_p 上有两种群运算，两种运算之间还满足通常我们所熟悉的加法和乘法之间的分配律(请验证！)，这使 F_p 具有比单纯的群更为丰富的性质。带+和*这两种运算的集合 F_p 称做特征为 *p* 的素域。素域是有限域(即仅有有限个元素的域)的一类特殊情形。

(4) 这是一个非常前沿的例子，读者可以跳过此例而不影响以后的学习。设 F_q 是有限域、阶为 q，A、B∈F_q 是给定的常数，F_q 上的椭圆曲线 E/F_q 是集合{(x,y): y^2=x^3+Ax+B}，在 E/F_q 上可以定义点的一种运算“+”使(x_1,y_1)+(x_2,y_2)=(x_3,y_3)，例如对 p≠2、3 的情形，x_3 和 y_3 按照以下公式计算：

$$x_3 = -x_1 - x_2 + (\frac{y_1 - y_2}{x_1 - x_2})^2, \qquad y_3 = -y_1 - (\frac{y_1 - y_2}{x_1 - x_2})(x_3 - x_1)$$

验证(E/F_q,+)是一个群是不平凡的事情，但这是千真万确的数学事实！因此，有限域上的椭圆曲线构成一个有限群

回到关于群的普遍性质。设 G 是一个有限群，N 是 G 的阶，*e* 是 G 的单位元素，Lagrange[3] 定理断言，对 G 的任何元素 a 必有 a^N=*e*。

注意将 Lagrange 定理应用于群 Z^*_N 便立刻得到 Euler 定理。要证明 Lagrange 定理为什么正确，只要将前面关于 Fermat 定理的证明中的特殊的群 Z_p*替换为抽象群 G，所有论证几乎可以逐字照搬(详细证明留给读者，为此需要关于抽象群的一个普遍性质，见习题 8-15)。

设 G 是一个有限群，N 是 G 的阶，*e* 是 G 的单位元素。对 G 的任何元素 a，a 的阶定义做使得 a^d=*e* 的最小正整数 d。不同的元素可以有不同的阶。对任何元素 a，从 Lagrange 定理可以推断 a 的阶 d 一定存在而且不超过群 G 的阶 N，但实际上还可以证明更精确的结论(习题 8-16)：

d 一定整除 N。

[3] Lagrange 是与 Euler 同时代的伟大的法国数学家,与 Euler 一样, Lagrange 也是十八世纪中后期的全能数学家, 其创造性工作不仅涵盖那个时代的全部数学, 还实质性地深入到那个时代最重要的应用数学,特别是力学和天文学, 其重要性影响至今。

群有许多具体类型，在密码学中至今应用最多的都是最简单的一类，即所谓循环群。这种群的结构特别简单，所有元素仅由一个元素生成，具体概念就是：G 定义做循环群，如果存在元素 g_0 使任何非单位元素 g∈G 都存在整数 m 使 $g=g_0^m$（g_0^m 表示 g_0 按照群 G 上的算术运算和自己运算 m 次），g_0 称为 G 的生成子。给定循环群 G，其生成子可能不止一个。注意按照定义，循环群一定是交换群。

在前面的例子中，$(F^*_p,*)$是循环群(但这一事实的证明并不简单，不在此详述)，其生成子称为素数 p 的原根。不难验证，这样的生成子一共有 φ(p-1)个(习题 8-17)。

关于群的另一重要概念是子群：H 是 G 的子集，如果 G 上的群运算应用于 H 的元素恰使 H 也是一个群，则 H 定义做 G 的子群。关于子群的一个普遍而惊人的规律也是 Lagrange 发现的，也称为 Lagrange 定理：*子群 H 的阶必整除 G 的阶*。

g∈G，所有形如 g^m 的元素(m 取遍整数集合)构成的子集记做<g>，不难验证<g>是群 G 的子群，阶恰是元素 g 的阶，于是前一个 Lagrange 定理就成为这里的 Lagrange 定理的特殊推论。关于这里的 Lagrange 定理的证明，可以参考任何一本近世代数的教科书。

8.1.5* 二次剩余及二次同余式 $x^2=a \bmod p$ 在特殊情况下的解

一次同余式 $ax=b \bmod N$ 是否可解以及如何求解有完整的理论，这就是 Euclid 定理和 Euclid 算法。接下来一个自然的问题建立二次同余式的类似理论，即二次剩余理论，但后者的情形比前者复杂，这里为方便读者参考给一个概要性的介绍，读者也可以跳过本小节而不影响今后的学习。

二次剩余理论中最深刻的结论之一是 *Gauss-Euler-Legend* 互反律。首先定义 *Legend* 符号：对素数 p 及与之互素的整数 a，(a/p)=+1 表示 a 是 p 的二次剩余，即存在 x 满足方程 $x^2=a \bmod p$；(a/p)= −1 表示 a 是 p 的非二次剩余，即方程 $x^2=a \bmod p$ 无解；对 p|a 的情形，约定(a/p)=0。*Legend* 符号有以下性质：

(1) (ab/p)=(a/p)(b/p)

(2) $(2/p)=(-1)^{(p^2-1)/8}$、a 奇则$(a/p)=a^{(p-1)/2} \bmod p$

(3) $(p/q)(q/p)=(-1)^{(p-1)(q-1)/4}$ (*Gauss-Euler-Legend* 互反律)

对整数 $m=\prod_{i=1}^{s} p_i^{e_i}$、$n=\prod_{j=1}^{t} q_j^{f_j}$，定义 *Jaccobi* 符号$(m/n)=\prod_{i=1}^{s}\prod_{j=1}^{t}(p_i/q_j)^{e_i f_j}$ (右面出现的(p/q)是 *Legend* 符号)。不难验证 *Jaccobi* 符号也有形式上完全相同的互反律

$(m/n)(n/m)=(-1)^{(m-1)(n-1)/4}$，但注意对合数 m，(a/m)=1 不再意味着 $x^2=a \bmod m$ 一定有解。

对素数 p，二次同余式 $x^2=a \bmod p$ 在一些特殊情况下有非常有用的显式解。

(1) 素数 p=3 mod 4, Legend 符号(a/p)=1, 则 $\pm a^{(p+1)/4}$ 是方程 $x^2=a \bmod p$ 的解。

事实上，直接计算有 $(a^{(p+1)/4})^2=a^{(p+1)/2}=a^{(p-1)/2}a=(a/p)a=a \bmod p$。

(2) 素数 p=5 mod 8, (a/p)=1, 首先注意这时 $a^{(p-1)/4}=\pm 1 \bmod p$ 且 $y^2=-1 \bmod p$ 总存在解 β，前者成立是因为 $0=a^{(p-1)/2}-1=(a^{(p-1)/4}-1)(a^{(p-1)/4}+1) \bmod p$，后者成立是因为 $(-1/p)=(-1)^{(p-1)/2}=1 \bmod p$ 因此−1 是 p 的二次剩余。根据这些性质不难验证：

若 $a^{(p-1)/4}=1 \bmod p$ 则 $\pm a^{(p+3)/8}$ 是方程 $x^2=a \bmod p$ 的解；

若 $a^{(p-1)/4}=-1 \bmod p$ 则 $\pm \beta a^{(p+3)/8}$ 是方程 $x^2=a \bmod p$ 的解。

在不对 p 的性质做任何假设的一般情况下，有 *Shanks* 算法求解方程 $x^2=a \bmod p$。

8.2 消息认证与数字签名方案

消息认证方案和数字签名方案都属于构造各类计算机密码方案和协议的最基本的工具，其安全性在本质上都是某种意义上的抗伪造性质。消息认证方案(*Message Authentication Scheme*)用于保证数据一致性，防止数据在从发送方到达接收方的过程中被篡改。为达到防篡改的目的，发送方和接收方事先（通过其它安全途径，例如后面章节将要讨论的密钥协商机制）获得一个共享密钥，发送方以该密钥为参数计算数据认证码，接收方则以该密钥为参数对数据和认证码进行验证。直观地说，消息认证方案的安全目标在于保证任何攻击者在未知密钥的情况下，不可能有效伪造出合法的消息认证码。从这一意义上讲，消息认证方案可以看做一种对称的数字签名方案。

8.2.1 消息认证方案

消息认证方案是构造其他高级密码方案的一个非常有用的工具，而且一般情况下计算效率都很高，例如通常都比数字签名方案计算效率高得多。精确地说，消息认证方案 Σ=(KG, Tag, Vf) 是一组概率多项式算法[4]（以后简称 P.P.T.算法）KG, Tag 和确定性算法 Vf。设 *k* 是复杂度参数，通常是方案的某个特征参数的位数（后面会有很多具体例子），KG 是钥生成算法，以

[4] 概率算法 A 无非是这样一类算法，其每一步都可能是不确定的，结果取决于运行期间的某中随机因素，因此即使对同一输入运行多次，其输出也各有不同。密码学领域所关心的算法多数是这类算法，特别是对攻击者的描述，这样做的直观动机是：不仅允许攻击者计算，而且允许其猜测。这当然比完全确定性的算法更符合现实。

k 为输入并输出密钥 K；Tag 是消息认证码生成算法，以密钥 K 和消息明文 M 为输入并输出认证码 t；Vf 是确定性的验证算法，以密钥 K、消息明文 M 和认证码 t 为输入并输出验证结果：1 表示接受(验证成功)、0 表示拒绝(验证失败)。KG, Tag 和 Vf 还必须满足一致性关系：对任何 *k* 和 M 恒有 P[K←KG(k); t←Tag(K, M): Vf(K,M,t)=1]=1，这里 P[Z]表示事件 Z 的概率。

在实际应用中，消息认证码(*Message Authentication Code*)方案是一类最常涌的特殊的消息认证方案，今后简称为 MAC 方案。精确的说，MAC 方案 MAC=(KG, Tag, Vf) 是一组算法 KG, Tag 和 Vf，其中 KG 是钥生成算法，与前面的描述相同；Tag 是确定性的认证码生成算法；算法 Vf(K,M,t)验证 Tag(K,M)=t 是否成立，若成立则输出 1 否则输出 0。注意由这一定义，一致性关系总是自然地成立。

以上概念并未涉及安全性质，只是刻画了消息认证方案的算法性质。为了刻画安全性质，需要从两方面考虑，一方面是攻击者的能力，另一方面是其拟达到的目标(这也是我们表达密码方案和安全协议的安全性质的本质方法，今后经常用到)。从这两方面考虑，定性地说，一个消息认证方案是安全的，是指任何攻击者—在这里被抽象为任何 P.P.T.算法 A—在未知密钥 K 的条件下，无论能得到多少个消息 M_i 的真实的消息认证码 t_i(=Tag(K,M_i))，也无法生成这样一个新消息 M*和这样一个数 t*，使得 Vf(K,M*,t*)=1(严格来讲，“无法生成”的涵义是指概率 P[Vf(K,M*,t*)=1]随复杂度参数 *k* 增大而下降而且下降的速度比 *k* 的任何多项式的倒数都快，例如象 2^{-k} 这样下降。但作为一本导论性教程，本书尽力避免这种精确而复杂的理论表述)。这就是消息认证方案在抗伪造性意义上的安全性质。

注意在刻画以上概念时，攻击者 A 的能力有这样几方面：A 是 P.P.T.算法，即 A 仅具有现实的计算能力(A 不具备超越 P.P.T.的计算能力，否则几乎一切密码方案都会在这种“超级”计算机面前失效，但这并不是现实世界所面临的情况)；A 能获得某些附加信息(我们不关心 A 如何获得这些信息，只关心 A 确实能够得到这些信息)，这就是一组消息 M_i 的真实的消息认证码 t_i=Tag(K, M_i), 这样考虑的目的是防止当消息之间存在某种关系的时候，其消息认证码也具有某种(可能相同也可能不同的)关系，从而使伪造者有机可乘。在现实世界中这是完全可能的，例如一个通讯线路的窃听者从其记录的消息中就能得到大量这类信息。在以上概念中，攻击者 A 的目标是：生成某个数 t*和某个消息 M*，使得 Vf(K,M*,t*)=1，即(欺骗！)使消息接收者相信数 t*是消息 M*的认证码，从而相信 M*不是伪造的而确实是来源于合法的发送方，也就是持有密钥 K 的发送方。在现实世界中，不难想象 M*应是一个对攻击者 A

有利的消息，但理论模型并不关心这一点，只要 A 能够以现实的计算能力达到这样一个目标就认为 A 成功、消息认证方案失效，因此这里的理论意义上的抗伪造性涵义是非常保守的，它涵盖了现实世界所关心的一切情况：如果一个消息认证方案在这一意义下抗伪造，在实际应用情形中一定满足现实的抗伪造性要求！

最后指出一点：对实际方案的构造而言，消息和消息认证码总是个有限长度的字，因此两者的对应关系只有有限个，从概念上讲，攻击者总可以通过纯粹的猜测来生成可能的 M* 和 t*，这也正是以 P.P.T.算法表达攻击行为的原因之一。但另一方面，纯粹的猜测所能成功达到攻击目标的概率很低，对一个安全的方案，这一概率随复杂度参数 k 增大而下降而且下降的速度比 k 的任何多项式的倒数都快，例如象 2^{-k} 这样下降。一般而论要设计一个密码方案抵抗纯粹的猜测性攻击是容易的，这也就是前一段扩号中的注释所表达的概念的由来。

具体的消息认证方案常基于著名的 MD5 或 SHA/SHA-1 散列来构造，但我们今后的讨论并不依赖任何一种这类消息认证方案的具体构造，因此不在此继续深入了，有兴趣的读者可以参考章末列举的密码学参考书，例如 Stinson 的教科书第 4 章和 Oorschot 等的著作中的第 9 章。

8.2.2 数字签名方案

数字签名方案是用以保证数据一致性或数据源一致性(也称完整性)的公钥体制密码方案。精确地说，数字签名方案 Ξ=(KG, Sig, Vf)是一组 P.P.T.算法 KG、Sig 和确定性算法 Vf。设 k 是复杂度参数，KG 是钥生成算法，输出公钥/私钥对(pk,sk)，其中公钥 pk 被公开，私钥 sk 则仅被签名者持有，而且不存在有效的算法从公钥 pk 计算出私钥 sk；Sig 是签名算法，以私钥 sk 和消息 M 为输入、输出签名 σ；Vf 是验证算法，以公钥 pk、消息 M 和字串 σ 为输入并输出验证结果：1 表示接受(验证成功)、0 表示拒绝(验证失败)。KG, Sig 和 Vf 还必须满足一致性关系：对任何 k 和 M 恒有 P[(pk,sk)←KG(k); σ←Sig(sk, M): Vf(pk,M, σ)=1]=1，其中 P[Z]表示事件 Z 的概率。

为了刻画数字签名方案的安全性质，这里也从两方面考虑，即攻击者的能力及其拟达到的目标(注意与上一小节的方法类似)。定性地说，一个数字签名方案是安全的，是指任何攻击者—在这里被抽象为任何 P.P.T.算法 A—在未知私钥 sk(但已知公钥 pk)的条件下，无论能得到多少个消息 M_i 的真实的消息签名 $\sigma_i(=Sig(sk,M_i))$，也无法生成这样一个新消息 M*和这样一个数 σ*，使得 Vf(pk,M*,σ*)=1(“无法生成”的定量的涵义与上一段的解释相同)。这就

是数字签名方案在抗伪造意义上的安全性质。

在刻画以上概念时，攻击者 A 的能力有这样几方面：A 是 P.P.T.算法，即 A 仅具有现实的计算能力；A 能获得某些附加信息，这就是一组消息 M_i 的真实的数字签名 σ_i, 这样考虑的目的是防止当消息之间存在某种关系的时候，其数字签名也具有某种关系，从而使伪造者有机可乘。攻击者 A 的目标是：生成某个数 σ*和某个消息 M*，使得 Vf(pk,M*,σ*)=1，即(欺骗！)使消息接收者相信数 σ*确是 sk 的持有者对消息 M*的数字签名，从而相信 M*不是伪造的而确实是来源于合法的发送方，也就是私钥 sk 的持有者。在现实世界中，不难想象 M*应是一条对攻击者 A 有利的消息，但理论模型并不关心这一点，只要 A 能够以现实的计算能力达到这样一个目标就认为 A 成功、数字签名方案失效，因此这里的理论意义上的抗伪造性涵义是保守的，涵盖现实世界所关心的一切情况：如果一个数字签名方案在这一意义下抗伪造，在实际应用情形中一定满足现实的抗伪造性要求。

与消息认证方案类似，对实际方案的构造而言，消息和数字签名都是有限长度的字，因此两者的对应关系只有有限个，攻击者(P.P.T.算法)总可以通过纯粹的猜测来生成可能的 M*和 σ*，但另一方面，纯粹的猜测所能成功达到攻击目标的概率很低，对一个安全的数字签名方案，这一概率随复杂度参数 *k* 增大而迅速下降，例如象 2^{-k} 这样指数式地下降，因此这类猜测并不构成有实际意义的威胁。

至此已经定性地解释了消息认证方案和数字签名方案在抗伪造意义上的安全性质。可以看到两者有类似之处，最大的区别在于前者的发送方和验证方都使用同一个密钥 K 作为秘密参数(如何生成 K 并非消息认证方案本身要解决的问题，方案的组成算法 KG 只是用来表达认证方案需要什么样的 K，以后在讨论安全协议时会看到一些生成 K 的具体过程的例子)，后者则只有发送方持有秘密参数 sk，验证方的一切计算都是基于公开的参数，例如公钥 pk，具有典型的信息不对称特点，这也是当代公钥密码方案的共同特点，下面还会继续看到。下面给出几个著名的数字签名方案实例，它们都被广泛应用于当代各类网络安全系统。

例 8-1(*Schnorr* 数字签名方案,1991) G 是 q 阶循环群，g 是 G 的生成子(回顾 8.1.4 节)，q 是 *k* 位素数(注：q 以 2 进制或 10 进制或其它进制表示都是无关紧要的，这些区别仅仅使 *k* 相差一个常数倍)；$H:\{0,1\}^+ \rightarrow Z_q$ 是一个抗冲突的散列函数[5]。*Schnorr* 方案的组成算法如下：

公钥/私钥生成算法 KG(*k*, G, g, q):

$x \leftarrow^{\$} Z_q^*$; $y \leftarrow g^{-x}$; vk←y; sk←x; return(vk,sk);

[5] 严格地说, H 应是一个所谓随机 oracle, 但这一概念所涉及的理论非常微妙, 读者在这里将其理解为类似于 MD5 或 SHA 那样的散列函数就可以了。

其中符号 $x\leftarrow^{\$}Z_q^*$表示在集合 Z_q^*上按照均匀分布随机生成 x，因此 KG 是一个 P.P.T.算法，公钥和私钥分别是 vk 和 sk。注意 KG 的输入中出现群 $\mathcal{G}$，意思是 KG 依赖于群运算的定义(用来计算 y)，而非存储所有的群元素。

签名算法 $Sig^H(sk, M)$，其中 sk=x:

$K\leftarrow^{\$}Z_q$; $r\leftarrow g^K$; $h\leftarrow H(M,r)$; $s\leftarrow(K+xh) \bmod q$; return(r,h,s);

注意这也是一个 P.P.T.算法，随机性来源于中间变量 K，M 的签名 σ 是个三元组(r,h,s)。

验证算法 $Vf^H(vk, M, (r,h,s))$，其中 vk=y:

return($h=H(M,r) \wedge r=g^s y^h$);

对没有发生伪造攻击的情况，sk 的持有者对消息 M 的签字(r,h,s)总能通过验证，即满足一致性条件 $Vf^H(vk, M, (r,h,s))=1$。事实上，这时第一个验证方程 h=H(M,r)当然成立，而另一个验证方程成立是因为 $g^s y^h=g^s g^{-xh}= g^{s-hx}= g^{(s-hx) \bmod q}$(为什么？)$= g^K= r$。

注意签名算法 $Sig^H(sk, M)$每次必须独立地随机生成 K，K 不能够取常数或使多个消息共享同一个 K，否则达不到安全目的：假如 Sig^H 对两个不同的消息 M_1、M_2 使用同一个 K，显然 r 在两个签名中也有相同的值，由此所输出的数字签名分别为(r,h_1,s_1)和(r,h_2,s_2)，并且因为消息 $M_1 \neq M_2$ 故 $h_1 \neq h_2 \bmod q$ (这一点源于 *H* 的抗冲突性质，在实践中应用 MD5 或 SHA 这类散列函数时就会具有这类性质)。攻击者(一个 P.P.T.算法)A 从所观测到的(r,h_1,s_1)、(r,h_2,s_2)、M_1、M_2(但这里不需要消息 M_1 和 M_2)和公钥信息 q 解以下方程组，其中 K 和 x 作为未知量：

$$s_1=(K+xh_1) \bmod q$$
$$s_2=(K+xh_2) \bmod q$$

因为 $h_1 \neq h_2 \bmod q$，即$(h_1-h_2, q)=1$(为什么?)，所以能完全解出私钥 $x=(h_1-h_2)^{-1}(s_1-s_2) \bmod q$(请验证！)。既然私钥 x 能被破译出来，A 当然能由此伪造 x 的持有者对任何消息的数字签名。注意以上算法 A 确实是多项式复杂度的现实算法，这就解释了为什么中间变量 K 必须每次随机独立地生成，特别是，签名算法 $Sig^H(sk, M)$本质上是随机而非确定性的。

已经证明：若群 $\mathcal{G}$ 上的离散对数问题难解，则 *Schnorr* 数字签名方案抗伪造攻击。所谓 $\mathcal{G}$ 上的离散对数问题(记做 DLP)是指：已知生成子 g，对任意给定的群元素 *y* 求整数 m 使 $y=g^m$。这一问题看似简单，但实际上惊人地困难(当然不是任何群上的 DLP 都难解)，例如对 8.1.4 节中所例举的那些群至今都没有找到 P.P.T.求解算法。有趣的是，恰是这类难解性问题构成当代具体密码方案安全性质的基础，这一点今后还会看到更多的例子。

例 8-2 (*ElGamal* 数字签名方案,1985) p 是所谓 *k* 位 α-难解型素数(即 p−1 有素因子 q 使 $(p-1)/q \leq |p|^{\alpha}$, |p|表示 p 的位数)，g 是 Z_p^*的生成子，$H:\{0,1\}^+ \rightarrow \{0,1\}^k$ 是抗冲突的散列函数(参

考例 8-1 的注释)，p、g、*H* 是公开的参数。方案的组成算法如下：

公钥/私钥生成算法 KG(k, g, p):

x←$ {1,2,…,p-1}; y←g^x mod p; vk←y; sk←x; return(vk,sk);

公钥和私钥分别是 vk 和 sk,其他符号同例 8-1。

签名算法 $Sig^H(x, M)$:

K←$ Z_p*; r←g^K mod p; h←H(M,r);

由方程 h=(xr+Ks)mod (p−1)解出 s;

return(r, h, s);

验证算法 $Vf^H(y, M, (r,h,s))$:

return(h=H(M, r)∧$g^h = y^r r^s$ mod p);

请读者仿例 8-1 自行验证，当不存在攻击时以上签字方案总能通过验证，即满足一致性条件，验证时要用到 *Fermat* 定理。同样，签名算法每次也必须独立地随机生成 K，K 不能够取常数或使多个消息共享同一个 K，否则达不到安全目的。

另一些著名的数字签名方案的例子在习题中给出(习题 8-20)。

8.2.3* 消息认证与数字签名方案的安全模型

为方便爱深入钻研的读者参考和查阅，这里给出关于消息认证方案和数字签名方案的抗伪造性的精确概念。这一节的内容可以略去而不影响今后的学习。在下面的表达中，若 A 和 B 都是算法，则符号 $A^{B(.)}$表示算法 A 在运行期间作为子程序调用算法 B。注意这是所谓黑箱调用，即 A 仅对 B 提供输入并从 B 获取输出，除此以外 A 从不改变 B 的内部运行结构。符号 $A^{B(x,.)}(a)$表示算法 A 在运行期间作为子程序调用算法 B，B 的运行以一个 A 未知的输入 x 为参数，A 则以 a 为其自身的输入。另一个需要的概念是 *k* 的速降函数 ε (*k*)，它是这样一类函数，对任何正数 n 都满足 $\lim_{k\to\infty} k^n \varepsilon(k)=0$。例如 $2^{-k/100000}$、$100000k^{-\log k}$、$100000(\log k)^{-k}$ 等都是速降函数，但 $k^{-1000000000000}$ 不是速降函数。

MAC=(KG, Mac, Vf)是一个消息认证码方案，KG, Mac, Vf 分别是密钥生成算法、MAC 算法和验证算法。MAC 定义做抗选择消息伪造攻击，若对任何 P.P.T.算法 A，以下对抗实验输出为“1”的概率 $Adv_{MAC, A}(k)$的上界是 *k* 的速降函数， *k* 是复杂性参数：

K←KG(*k*); /*随机生成公钥/私钥*/

$(M^*,t^*)\leftarrow A^{MAC(K, .)}(k)$;

/*A 可以任意方式得到消息-签字对，然后生成一个消息-签字对(M*,t*)*/

if Vf(K, M*, t*)=1 且 M*是新消息

/*即 A 从未以 M*为输入调用过 MAC(K,.)*/

then output(1) else output(0)

/*若 A 伪造成功则实验输出“1”，否则实验输出“0”.*/

S=(KG, Sig, Vf)是一个数字签名方案，KG, Sig, Vf 分别是签字私钥/公钥生成算法、签字算法和验证算法。S 定义做抗选择消息伪造攻击，若对任何 P.P.T.算法 A，以下对抗实验输出为“1”的概率 $Adv_{S, A}(k)$的上界是 k 的速降函数，k 是复杂性参数：

(vk, sk)←KG(k); /*随机生成公钥/私钥*/

$(M^*, \sigma^*)\leftarrow A^{Sig(sk, \cdot)}(vk)$;

/*A 可以任意方式得到消息-签字对，然后生成一个消息-签字对(M*, σ*)*/

if Vf(pk, M*, σ*)=1 且 M*是新消息

/*即 A 从未以 M*为输入调用过 Sig(sk,.)*/

then output(1) else output(0);

/*若 A 伪造成功则实验输出“1”，否则输出“0”.*/

8.3 公钥加密方案和典型实例

直观地讲，一个加密方案如果能够保住秘密，则破译者在仅仅依据密文本身和所有合法公开信息的情况下无论花费多少现实的计算资源也并不比他随机猜测能够获得更多有关明文的信息。这种保密性的直观解释适合于对称加密(在下一节阐述)和公钥加密两类方案，这一节首先阐述公钥加密方案。一个公钥加密方案 Π=(KG, E, D)是一组 P.P.T.算法 KG、E 和确定性算法 D。设 k 是复杂度参数，KG 是公钥/私钥的生成算法，接收输入 k 并输出公钥−私钥对(pk, sk), 其中公钥 pk 被公开，私钥 sk 则仅被合法解密者持有，而且不存在有效的算法从公钥 pk 计算出私钥 sk；E 是加密算法，以公钥 pk 和消息明文 M 为输入并输出密文 y；D 是解密算法，以私钥 sk 和密文 y 为输入并输出明文 M.。三个算法还必须满足一致性关系：对任何 k 和 M 恒有 P[(pk, sk)←KG(k); y←E(pk, M): D(sk, y)=M]=1,这里 P[Z]表示事件 Z 的概率。

和上一节一样，本节不深入剖析加密方案保密性的精确含义，而是列举一些被广泛应用的公钥加密方案的例子。这些方案都已经在理论上被证明是安全(即保密)的。

8.3.1 RSA 与 OAEP/RSA 方案

最著名也是最简单的例子是 RSA 方案[6]，这一方案的公钥是一个大整数 N 和一个奇整数 *e*，私钥是 N 的素因子 p,q(p≠q、N=pq)和另一个整数 d，d、*e* 和 Euler 函数 φ(N)满足这样的关系： (e,φ(N))=1 且 *e*d=1 mod φ(N)(回顾 8.1.1 节，第一个条件保证第二个线性同余方程确实有解 d)。加密时，每个消息 M 看做[1,N−1]上的一个整数，计算出的密文是 $y=M^e \bmod N$。注意加密计算仅仅用到公开的合法信息 *e* 和 N。解密时，出来公开的合法信息之外，还需要私钥 d(这保证了只有 d 的持有者才有能力解密)，计算出 $y^d \bmod N$。要解释为什么这一解密算法是正确的，只要注意当 M 与 N 互素时有

$$y^d \bmod N = M^{de} \bmod N = M^{1+j\varphi(N)} \bmod N = (\text{由 Euler 定理}) \ M \bmod N = M$$

j 是一个整数但我们并不关心其真正的值，因为 Euler 定理总使 $M^{j\varphi(N)} \bmod N = (M^{\varphi(N)})^j \bmod N = 1$。当 M 与 N 并非互素时，不妨设这时 M=hq 且 h 与 p 互素(否则 M=0 mod N)，即 M 是 N 的素因子 q 的倍数，于是

$$y^d \bmod p = M^{de} \bmod p = M^{1+j\varphi(N)} \bmod p = M^{1+j(p-1)(q-1)} \bmod p$$

$$= M(M^{(p-1)} \bmod p)^{j(q-1)} \bmod p = M \bmod p \ (\text{因为由 } \textit{Fermat} \text{ 定理有 } M^{(p-1)} \bmod p = 1)$$

而 $y^d \bmod q = 0 = M \bmod q$

再由中国剩余定理便知 $y^d \bmod N = M$。

RSA 方案的保密性依赖于对大数 N 做因子分解的难解性。目前被工业界所广泛应用的并非上面这个 RSA 方案，而是下面例 8-3 中的方案，这就是著名的 PKCS#1 标准，两者所依赖的难解性相同，至于为什么用下面这个方案而不用纯粹的 RSA，后面再行解释。

例 8-3(*OAEP/RSA* 公钥方案(PCKS#1), 1994)公钥 pk=(*e*,N,H,G)，其中 N 是两个大素数的乘积，*e* 是和 φ(N)互素的一个正整数，G、H 是两个随机散列函数(参考注 5)，输出分别为 $n+k_0$ 位和位 k_0 二进制数，N 为 $n+2k_0$ 位二进制数；私钥 sk=d，d 是满足方程 *e*d=1 mod ϕ(N)的一个正整数。

下面的符号 x||y 表示两个字串 x 和 y 的联结；注意一个对象有时被作为一个数，有时又被作为一个纯粹的字串，这在上下文中都是清楚的，请注意区别。

加密算法 E(pk, M)，其中消息 M 是 n 位二进制数：

随机选取一个 k_0 位的二进制数 r;

s←(M||r) ⊕ G(r); /*按位异或*/

[6] 发表于 1978 年，名称 RSA 源于其发明者 Rivest、Shamir 和 Adellman。

t←r⊕H(s); /*按位异或*/

y←$(s||t)^e$ mod N;

return(y);

解密算法 D(sk, y):

x←y^d mod N;

parse x as s||t, |s|= $n+k_0$, |t|=k_0;

r←t⊕H(s);

M*←s⊕G(r);

if M*=M||r /*注意 M*的最低 k_0 位恰是 r*/

then return(M); /*M 是 M*的最高 $n+k_0$ 位*/

else return(“错误”);

读者不难验证这一方案确实满足一致性条件，加密算法输出的密文总能被正确地解密出原来的明文。已经证明：若 RSA 函数 $y=x^e \bmod N$(其中 N=pq、p 和 q 是未知的奇素因子)是单向的，则 *OAEP/RSA* 方案保密。这里一个函数 F(x)的所谓单向性是指：给定自变量 x，计算 y=F(x)总是容易的，即存在多项式复杂度的算法完成这一计算；但从任何 y 计算一个原像 x 是困难的，不存在任何 P.P.T.算法能够完成这一计算任务。

8.3.2 ElGamal 方案与 Cramer-Shoup 方案

上一小节的公钥加密方案的保密性依赖于大整数的因子分解问题的难解性，这一节中例子的保密性则与离散对数问题的难解性有关。具体来说，它们都依赖于所谓判定性 *Diffie-Hellman* 问题的难解性。给定一个循环群 $\mathcal{G}$ 和一个(公开的)生成子 g，任给三个元素 U、V 和 W，既然 G 是循环群，当然总有整数 a 和 b 使 $U=g^a$、$V=g^b$。$\mathcal{G}$ 上的所谓判定性 *Diffie-Hellman* 问题，记做 DDHP，是在已知(g,U,V,W)的条件下判定 W 是否等于 g^{ab}。对这一问题，取复杂性参数 k 为群 $\mathcal{G}$ 的阶数的位数。如果对阶数充分大的循环群 $\mathcal{G}$ 不存在任何 P.P.T.算法能以 k 的非速降函数的概率输出正确的结果，就说群 $\mathcal{G}$ 上的 DDHP 难解。例如，目前普遍相信(但至今未被证明)乘法群 F_p*上的 DDHP 难解。

不难看出，如果群 $\mathcal{G}$ 上的 DLP(离散对数问题，见例 8-1 的说明)可以被有效求解，则 DDHP 易解，因为 DLP 易解意味着可以有效计算出 U、V 对 g 的指数 a 和 b，从而总可以毫不含糊地判定 W 是否等于 g^{ab}。然而，如果 $\mathcal{G}$ 上的 DLP 难解，至今尚不知道如何有效求解 $\mathcal{G}$ 上的 DDHP，从这一观点看，对 DLP 难解的循环群 $\mathcal{G}$，其上的 DDHP 难解是一个充分

合理的假设。此外，这里也顺便指出一个相关的难解性问题，它也常被用到，这就是所谓计算性 *Diffie-Hellman* 问题的难解性。给定一个循环群 $\mathcal{G}$ 和一个(公开的)生成子 g，任给两个元素 U、V，既然 G 是循环群，当然总有整数 *a* 和 *b* 使 $U=g^a$、$V=g^b$。$\mathcal{G}$ 上的所谓计算性 *Diffie-Hellman* 问题，记做 CDHP，是在已知(g,U,V)的条件下计算出群元素 g^{ab}。对这一问题，取复杂性参数 *k* 为群 $\mathcal{G}$ 的阶数的位数。如果对阶数充分大的循环群 $\mathcal{G}$ 不存在任何 P.P.T.算法能以 *k* 的非速降函数的概率输出正确的结果，就说群 $\mathcal{G}$ 上的 CDHP 难解。读者不难看出，如果群上的 CDHP 易解，则其上的 DDHP 易解，或等价地表达为：如果群上的 DDHP 难解，则其上的 CDHP 难解解。总结起来，群 $\mathcal{G}$ 上的这三个难解性假设的关系是这样：

“DDHP 难解” 蕴涵 “CDHP 难解” 蕴涵 “DLP 难解”

例 8-4(*ElGamal* 方案,1985)$\mathcal{G}$ 是素 q 阶循环群，生成子为 g，各组成算法如下：

密钥生成算法 KG(q,g):

$x \leftarrow^{\$} Z_q$; $X \leftarrow g^x$; pk←(q,g,X); sk←(q,g,x); return(pk, sk);

加密算法 E(pk, M), $M \in \mathcal{G}$:

$r \leftarrow^{\$} Z_q$; $Y \leftarrow g^r$; $T \leftarrow X^r$; W←TM; return(Y, W);

解密算法 D(sk, (Y,W)):

$T \leftarrow Y^x$; $M \leftarrow WT^{-1}$; return(M);

很容易验证以上算法满足一致性条件：在解密算法中 $Y^x=g^{xr}$ 故 $W(Y^x)^{-1}=((g^x)^rM)g^{-xr}=M$。已经证明，若循环群 $\mathcal{G}$ 上的 DDHP 难解则 *ElGamal* 方案保密。

例 8-5(*Cramer-Shoup* 方案,1999) $\mathcal{G}$ 是素 q 阶循环群, 生成子为 g，各组成算法如下：

密钥生成算法 KG(q,g_1,g_2,K)：

$g_1 \leftarrow g$;

$x_1,x_2,y_1,y_2, z \leftarrow^{\$} Z_q$;

$c \leftarrow g_1^{x1} g_2^{x2}$;

$d \leftarrow g_1^{y1} g_2^{y2}$;

$h \leftarrow g_1^{z}$;

pk←(c,d,h);

sk←(x_1,x_2,y_1,y_2, z);

return(pk, sk);

加密算法 E(pk, M), $M \in \mathcal{G}$:

$r \leftarrow^{\$} Z_q$;

$u_1 \leftarrow g_1^r$; $u_2 \leftarrow g_2^r$;

$e \leftarrow Mh^r$;

$T \leftarrow H_K(u_1,u_2,e)$;

$v \leftarrow c^r d^{rT}$;

return(u_1,u_2,e,v);

解密算法 D(sk,Y), $Y=(u_1,u_2,e,v)$：

$T \leftarrow H_K(u_1,u_2,e)$;

if $v=u_1^{x1+Ty1}u_2^{x2+Ty2}$ then $M \leftarrow e/u_1^z$ else M←“错误”;

return(M);

很容易验证以上算法满足一致性条件，我们把这一计算留给读者。已经证明：若群 G 上的 DDHP 难解则 *Cramer-Shoup* 方案保密。

其他一些有趣的公钥加密方案见习题 8-22。在结束这一小节之前，我们给出一个重要概念，即密文非可塑性，以此解释以上几个公钥加密方案有什么不同。为解释密文非可塑性概念，我们先对 RSA 方案做一个有用的观察。设攻击者 A 截获到一个 RSA 方案的密文 y，不妨设 $y=M^e \bmod N$，当然这里 A 除了 y 和合法的公开信息 *e* 与 N 之外并不知道其它信息，特别是 A 并不知道 M 和 d，但 A 可以这样对 y 实施变形：A 选择一个常数 h 并计算 $y^*=h^e y \bmod N$，然后 A 将 y*而非 y 传递给合法接收者，即私钥 d 的持有者 B，这样一来 B 最终解密出来的明文是什么？注意到 $y^*=h^e y \bmod N = h^e M^e \bmod N = (hM)^e \bmod N$，因此毫无疑问 B 最终解密出来的是 hM mod N 而非 M！进一步，B 能知道这一解密并非输出原来的明文吗？不能，因为就 RSA 方案本身而论，B 没有任何根据在解密过程中判断被解密的密文本身是否被如上篡改过！

这个例子表明，对 RSA 方案，即使攻击者并不知道(甚至永远不可能知道)明文 M 本身，却有可能通过变形密文而决定合法解密者的解密结果。以上缺陷当然不是在任何情况下都对攻击者 A 有利，例如对一个有现实意义的消息 M，消息 hM mod N 却未必总有实际意义。但无论如何，这是一个可能被利用的缺陷，具有这一缺陷的加密方案就称为具有密文可塑性。RSA 方案甚至还具有另一种形式的密文可塑性：对密文 $y=M^e \bmod N$，若 A 选择一个常数 t 并计算 $y^*=y^t \bmod N$，然后将 y*而非 y 传递给合法接收者 B，这样一来 B 最终解密出来的明文将是 $M^t \bmod N$(为什么？)。除了 RSA 方案，*ElGamal* 方案也具有密文可塑性，具体分析

见习题 8-23。

最高保密程度的加密方案应该抗密文变形，也就是具有密文非可塑性：一旦密文被变形，解密算法在解密计算过程中总能以接近于 1 的概率识别出来并且输出一个错误。这就是 *RSA* 方案、*ElGamal* 方案与 *OAEP/RSA* 方案和 *Cramer-Shoup* 方案之间的本质差别：前者仅仅对其明文本身保密但不具备密文非可塑性，后者不仅保密密文所对应的明文本身，而且能够被严格证明具有密文非可塑性。更深入的理论分析不能在此深入了，感兴趣的读者可以阅读章末所列举的作者的专著。

8.3.3 公钥基础设施

公钥密码方案，无论是数字签名方案还是加密方案，都具有两个密钥，即一个(公开的)公钥 pk 和一个仅由合法签字者或解密者所持有的私钥 sk。任何人需要验证 σ 是否为合法签字者 B 对消息 M 的签名，只要已知 B 的签字私钥 sk 所对应的公钥 pk，通过计算 Vf(pk,M, σ)=1 即可判定这一点；任何人需要向某个对象 C 发送保密消息 M，只要已知 C 的解密私钥 sk 所对应的公钥 pk，计算密文 E(pk,M)即可达到目的。然而，这里需要一个前提，那就是：pk 确实是 B(或 C)的公钥 pk，或者等价地说，B(或 C)确实持有与 pk 所对应的那个私钥 sk。于是，一个自然的问题是：如何向需要验证签字或保密通讯的主体保证这一点？

这就是公钥基础设施(*Public-Key Infrastructure*, 简称 *PKI*)的目的，它为实际应用中的公钥归属提供可信任的保证，具体机制是所谓公钥证书。PKI 最主要的组成部分包括一个(或一组)可信任的服务器，称为证书权威机构(*Certificate Authority*, 简称 *CA*)，实际上是一个可信任和高度可靠的文件服务器，以及用以访问该服务器的安全协议。*CA* 的主要功能是发布公钥证书，公钥证书实际上是一个含 *CA* 服务器的数字签名的文件，文件正文由一系列的属性构成，包括证书发布主体(*CA* 服务器)的名字、证书的标识号、版本号、发布时间、失效时间(或有效期)、公钥本身、该公钥隶属的主体(更精确地说，就是持有与该公钥所对应的那个独一无二的私钥的主体)的标识或名称、该公钥的功能(用以数字签名或加密)、该公钥所具体应用的算法或密码方案，以及一些附加属性。所有这些属性的值以证书发布者的私钥签名，以表明该证书的合法来源。证书发布者用以对证书进行签字的私钥所对应的公钥是一个公开发布的信息，在该发布机构起作用的范围内是一个可信任的参数，因此任何公钥的使用者可以以此验证该证书是否合法，并根据合法证书所表达的属性来正确地使用公钥。这就是对 *PKI* 功能的直观解释。

在实际应用中，为 PKI 有效履行公钥发布与管理功能，还需要一些附加机制。例如，

为使某些公钥的私钥因意外泄露而造成的损失最小，*PKI* 都具有专门发布失效证书的机制；为在一个非常大的范围内高效验证公钥证书，PKI 还提供一个层次结构，这个层次结构具体由多个 *CA* 服务器而非单独一个 *CA* 服务器组成，每个服务器仅针对一个特定的信任域发布公钥证书，处于大范围的、不属于同一个信任域的验证者可以通过信任链验证公钥证书是否合法，用这种方式既保证 PKI 的基本功能又保证实施 *PKI* 的灵活性。

8.3.4* 公钥加密方案的安全模型

和 8.2.3 节一样，为深入钻研的读者查阅方便，我们在这里给出公钥加密方案Π=(KG,E,D)保密性的数学定义，其中 KG、E、D 分别是公钥/私钥生成算法、加密算法和解密算法。

Π定义做选择明文保密，若对任何概率多项式算法 $A=(A_1,A_2)$，以下对抗实验输出为“1”的概率减去 1/2 后的绝对值 $Adv_{\Pi A}(k)$是 *k* 的速降函数，*k* 是复杂性参数。

(pk, sk)←KG(k); /*随机生成公钥/私钥*/

$(M_0, M_1, St)\leftarrow A_1(pk)$；/*攻击的第一阶段: 根据公钥 pk 生成两个长度相同的消息 $M_0 \neq M_1$，这里 A 被输入公钥 pk(但 A 未知私钥 sk)*/

$b\leftarrow^{\$}\{0,1\}$;

$y^*\leftarrow E(pk, M_b)$; /*随机选择 M_0, M_1 中的一个消息来加密*/

$d\leftarrow A_2(y^*, St)$; /*攻击的第二阶段: 根据密文 *y**和第一阶段的全部信息 St 来推断被选择的究竟是哪一个消息*/

if d=b then output(1) else output(0);

/*若推断正确则实验输出“1”，否则实验输出“0”。*/

可以证明，在相应的难解性假设下例 8-3 到例 8-5 都在以上意义下保密。例 8-3 和例 8-5 还在更强的意义下保密，这就是密文非可塑性，但这里不再深入刻画密文非可塑性的安全模型了。

如果Π满足以上保密性的定义，那么不难得出以下推论：第一，在未知私钥 sk 的情况下，仅根据公钥 pk 和密文 y=E(pk, M)计算出明文 M 的概率一定是 *k* 的速降函数。第二，从公钥 pk 计算出私钥 sk 的概率一定是 *k* 的速降函数。这些都是安全加密方案所应该具备的直观特性，现在被一个统一的安全模型所蕴涵。第三，加密算法 E 一定是概率性算法(即非确定性算法)。注意在这一意义上，RSA 方案甚至是不安全的，因为它的加密方案是一个完全确定性的算法，因此不满足以上保密性的定义；但 *OAEP/RSA* 与 *RSA* 不同，已经证明它确实满足这里的安全性定义。

8.4 对称加密方案

这一节首先概要讨论对称加密方案，然后结合前面的公钥加密方案介绍几个非常有用的混合加密方案。

8.4.1 对称加密方案

对称加密方案 Π=(KG, E, D)是一组 P.P.T.算法 KG、E 和确定性算法 D。设 *k* 是复杂度参数，一般取密钥的位数，KG 是密钥生成算法，接收输入 *k* 并输出密钥 K，注意这里不象公钥方案那样有两个密钥 pk 和 sk 分别用于加密和解密，而是仅有一个密钥 K 既用于加密也用于解密，因此 K 必须被合法的通讯双方共同持有(而在公钥加密方案中公钥 pk 被公开，私钥 sk 则仅被合法解密者持有，而且不存在有效的算法从公钥 pk 计算出私钥 sk)；E 是加密算法，以密钥 K 和消息明文 M 为输入并输出密文 y；D 是解密算法，以密钥 K 和密文 y 为输入并输出明文 M.。三个算法还必须满足一致性关系：对任何 *k* 和 M 恒有 P[K←KG(k); y←E(K, M): D(K, y)=M]=1。

目前常用的对称加密方案有 DES 方案(56 位密钥)及其变体，例如 3DES 方案(等效于 112 位密钥)；AES 方案(密钥长度任意)；IDEA 方案(128 位密钥)和 Blowfish 方案(密钥长度任意，最长 448 位)等。关于这些加密方案的算法已有非常多的书籍描述，这里不打算再赘述这方面的内容，这些方案的细节对我们后面的应用也并不重要，但在此指出关于对称加密方案的两个与其应用有关的重要技术：密文模式和密文完整性。

上一段所列举的那些对称加密方案的例子都是所谓分组加密方案，它们在实施之前需要将任意长度的明文消息划分为特定长度的分组，然后将加密算法逐个作用于固定长度的分组之上，例如 DES 和 IDEA 方案的分组长度是 64 位，Blowfish 方案是 32 位，等等。

如很多书籍描述的那样，需要使用特定的密文模式使分组加密后的序列连接起来成为一个拟随机序列。仅就那些常用的密文模式如 CBC、CFB、OFB 等的算法而论实际上非常简单，这里也不打算赘述(与理解本书后面的应用也没有直接关系)，但初学者经常疑惑为什么一定要这样做？不这样做就不安全吗？的确如此！看下面这个例子：假设 A 拟向 B 发送明文 M=“X 欠 Y509 万”，M(假定)被划分为分组“X”“欠”“Y”“5”“0”“9”“万”，如果仅仅对每个分组实施对称加密(哪怕假定该加密算法本身绝对不可破译)，得到的密文将是

y=E(K, “X”)||E(K,“欠”)||E(K,“Y”)||E(K,“5”)||E(K,“0”)||E(K,“9”)||E(K,“万”)

=y_1|| y_2|| y_3|| y_4|| y_5|| y_6|| y_7

攻击者截获密文 y，将各个分量 y_1、y_2、y_3、y_4、y_5、y_6、y_7 重新组合为 y*=y_3|| y_2|| y_1|| y_4|| y_5|| y_6|| y_7，将这个重新处理后的密文 y*转发到 B，读者不难看出 B 解密 y*所得到的明文将是什么？如果组合为 y_1|| y_2|| y_3|| y_4|| y_5|| y_6|| y_7||y_6(即去掉最后一个密文分组)、组合为 y_1|| y_2|| y_3|| y_4|| y_5|| y_6|| y_6||y_6||y_6|| y_6|| y_7(即有意重复某个数字的密文分组)，B 解密后所得到的明文又将是什么？问题的关键在于：即使加密算法 E 本身永远不会被破译，以上加密过程就因此而真正安全吗？

这个例子所引出的是密文完整性问题，即安全的对称加密方案不仅需要保密，而且必须防止合法的密文被篡改。密文模式正是这样一种机制，将一个消息中的各个分组的密文前后关联起来，以此抵御对密文的可能篡改。采用密文模式从以上明文所生成的密文是 y= y_0||y_1|| y_2|| y_3|| y_4|| y_5|| y_6|| y_7，其中 y_0 是个纯粹的随机数，y_1=*f*(K,E(K,“X”),y_0)，y_2=*f*(K,E(K,“欠”),y_1)，y_3=*f*(K,E(K,“Y”),y_2)，等等，如此一来，(通过适当设计模式函数 *f*)在未知密钥 K 的情况下要象前面的例子中那样操纵密文就不再可行，这也就是必须采用密文模式的原因。注意当采用特定模式后，同一个明文分组的真正的密文是与其上下文有关的，在不同的上下文中其密文各不相同，这些都增加了攻击者篡改密文的难度。

保证密文完整性还有另一类可证明的通用方法。设 MAC=(KG^m, MAC, Vf)是第 8.2.1 节描述的任何消息认证方案、Π^s=(KG^s, E^s, D^s)是任何对称加密方案，两个方案的密钥分别记为 K^m 和 K^s。对明文 M，合成的加密算法 E(K^s||K^m, M)做以下计算：

y←E^s(K^s, M); z←y||Mac(K^m, y); return(z); /*z*/

其中 E^s(K^s, M)理解为对 M 实施带特定模式的对称加密计算。不难写出针对以上加密过程的解密算法 D(K^s||K^m, z)：

parse z as y||t;

if Mac(K^m, y)=t *then* return(D^s(K^s, y)) else return(“错误”);

并且容易验证 E 和 D 满足一致性关系。最后指出，以上方案正是虚拟专用网络的基本协议 IPSec/ESP(RFC#2405)所采用的对称加密方案。

8.4.2* 混合加密方案

前面已经讨论对称加密方案和公钥加密方案，包括公钥方案的各种实例。较之对称加密方案，公钥方案的优点是不言而喻的。然而在实际应用中，一个普遍存在的问题是当加密长

消息时公钥方案的效率明显不如对称加密方案，要达到相同的安全强度其计算速度可能比 DES 或 AES 这样的对称方案慢 2-3 个数量级，这在很大程度上限制了公钥方案的实际应用。为克服这一缺陷，一个常用的、也是自然的方法是联合应用公钥加密和对称加密方案，使用后者加密真正的(任意长度的)明文，而仅用前者加密对称方案的(固定长度的)密钥。如此一来，两者的优点都得到发扬，特别是明文数据的长度可以不受限制。这就是所谓混合方案的思想。读者很容易想到的一个混合加密方案是 $E^a(pk,K)||E^s(K,M)$，这里 E^a、E^s 分别是公钥方案和对称方案的加密算法，K 是随机生成的、用于加密明文 M 的密钥，被 E^a 所加密。注意合法解密方所需要的仅仅是与公钥 pk 对应的私钥 sk，因此这一混合方案本质上仍然属于公钥方案。这一节给出一些常用的混合方案的具体实例，它们都有很高的计算效率而且都被严格证明具有最高程度的保密性，特别地，它们都是密文非可塑的。

例 8-6(*Fujisaki-Okamoto* 混合方案,1999) $\Pi=(KG, E, D, G, H)$ 由一个公钥加密方案 $\Pi^a=(KG^a, E^a, D^a)$和一个对称加密方案 $\Pi^s=(KG^s, E^s, D^s)$ 复合而成，其中 *G* 和 *H* 是随机散列函数；. $KG=KG^a$，即 Π 的公钥和私钥分别就是 Π^a 的公钥和私钥；用 u||v 表示两个字符 u 和 v 联结，$E^a(pk,U;r)$表示加密算法 E^a 对消息 U 的加密，并且计算过程中以 r 为随机数(回顾上一节所有安全的公钥加密算法都是 P.P.T.算法)，*Fujisaki-Okamoto* 混合方案的加密算法 $E(pk,M)=E^a(pk, \sigma ; H(\sigma ||M))||E^s(G(\sigma),M)$，其中 σ 从 Π^a 的明文空间中随机生成，$H(\sigma ||M)$ 用做随机算法 E^a 中的随机数。解密算法 D(sk,y)如下：

parse y as $y_1||y_2$;

$\sigma \leftarrow D^a(sk,y_1)$;

$M \leftarrow D^s(G(\sigma),y_2)$;

if $y_1=E^a(pk, \sigma ; H(\sigma ||M))$ then output(M) else output(“错误”);

容易验证以上方案满足一致性条件。已经证明，只要 Π^a 和 Π^s 具有很弱的保密性质，作为整体的混合加密方案 Π^a 就能具备最高程度的保密性。这是混合方案的另一种优点：它能将弱保密的方案“提升”为强保密的方案，而计算效率比直接构造的这类强保密方案要高得多。下面的例子也都具有这种特点。

例 8-7(*REACT* 混合加密方案,1999) $\Pi=(KG, E, D, G, H)$ 由公钥加密方案 $\Pi^a=(KG^a, E^a, D^a)$和对称加密方案 $\Pi^s=(KG^s, E^s, D^s)$ 按以下方式复合而成：G、H 是随机散列函数，$KG=KG^a$；加密算法 $E(pk,M)=E^a(pk,R;u)||E^s(G(R),M)||H(R,m,y_1,y_2)$，其中 u 是加密算法 E^a 中的随机数，$y_1=E^a(pk,R;u)$，$y_2=E^s(G(R),M)$。解密算法 D(sk,y)如下：

parse y as $y_1||y_2||h$;

$R \leftarrow D^a(sk,y_1)$;

$M \leftarrow D^s(G(R),y_2)$;

if h=H(R,M, y_1,y_2) then output(M) else output("错误");

例 8-8(*GEM* 混合加密方案,2002)F、G 和 H 是随机散列函数，GEM 方案 Π=(KG, E, D, F, G, H)由公钥加密方案 Π^a=(KG^a, E^a, D^a)和对称加密方案 Π^s=(KG^s, E^s, D^s)按以下方式复合而成：KG=KG^a；加密算法 E(pk,M; r||u)=$y_1||y_2$，其中 r 是随机数、u 是加密算法 E^a 的随机数，s=F(M||r)、W=s||(r⊕H(s))、K=G(W||y_1)、$y_1=E^a(pk,W;u)$、$y_2=E^s(K,M)$；解密算法 D(sk,y)如下：

parse y as $y_1||y_2$;

$W \leftarrow D^a(sk,y_1)$;

$K \leftarrow G(W||y_1)$;

$M \leftarrow D^s(K,y_2)$;

Parse W as s||t;

r←t⊕H(s);

if s=F(M||r) then output(M) else output("错误");

以上所有这些混合方案实际上都是通用的结构框架，当以任何公钥方案和对称方案代入是，就能得到各种不同的混合方案的实例化。最计算效率方面，以上三个例子中 *GEM* 方案的效率最高。

8.5* 基于身份的加密方案

前面阐述的所有公钥加密方案在实现时都具有一个共同特点，这就是其公钥必须具有某种特殊的数学结构或性质，例如 RSA 方案的指数 *e* 和模数 N 必须满足条件：N 有两个大素因子 p 和 q、(p−1)/2 和(q−1)/2 也必须是大素数且 *e* 必须与 N 的 Euler 函数值互素，总之公钥与公钥/私钥持有者的固有特征毫不相干，是一个外加到持有者身上的事物，因此在实际应用中不得不借助于公钥基础设施特别是公钥证书将公钥与其持有者的身份联系起来(更精确地说，是将公钥与持有与该公钥所配偶的那个私钥的对象的身份联系起来)。除了这一类普通公钥加密方案，最近还实现了另一类所谓基于身份的公钥加密方案(*identity-based encryption scheme*，简称 IBE 方案)，这类方案能够以任何符号为公钥，例如图像、指纹、email 地址等等。因为可以取用户的某种固有信息为公钥，所以基于身份的密码体制不需要公钥证书，使得这类方案在实际应用中具有特殊价值。与普通公钥加密方案不同，在结构上这类密

码方案都包括一个用户私钥生成算法，它以全局私钥(参见下面的定义和例子)和用户身份标识(公钥)为输入来生成用户私钥，因此这类方案天然地具有所谓密钥托管(*key escrow*)的特点。

精确地说，一个 IBE 方案 Π=(Setup, UKG, E, D)是一组 P.P.T.算法，其中 Setup 是全局密钥(亦称主密钥)生成算法，以复杂性参数 k 为输入并输出全局公钥-私钥偶(mpk, msk)；UKG 是用户私钥生成算法，以全局私钥 msk、用户身份标识 a 为输入并输出 a 的私钥 usk(a)；E 是加密算法，以全局公钥 mpk、用户身份标识 a 和消息 M 为输入并输出密文 y；D 是解密算法，以全局公钥 mpk、用户私钥 usk(a)和密文 y 为输入并输出明文 M。所有这些算法还满足一致性关系：对任何 k、a 和 M 恒有

P[(mpk,msk)←Setup(k); usk(a)←UKG(msk,a); y←E(mpk, a, M): D(mpk, usk(a), y)=M]=1

由于 IBE 方案的特殊结构，在刻画其保密性质时需要考虑所谓合谋攻击，这时攻击者可能(通过非法入侵或合谋)持有某些合法用户 $a_1,\ldots,a_n$ 的私钥 usk(a_1),…usk(a_n)，IBE 方案的保密性要求：只要攻击者不持有私钥 usk(a)，无论事先能获得多少 usk(a_1),…usk(a_n)($a_1,\ldots,a_n\neq a$)都无法从密文 E(mpk, a, M)有效获取关于明文 M 的信息。安全的 IBE 方案必须能抵抗这类攻击。

下面是第一个实用的 IBE 方案的实例。

例 8-9(*Boneh-Franklin* 方案,2000) 实用性 IBE 方案的数学构造及其安全性证明需要用到所谓双线性群偶及双线性 *Diffie-Hellman* 问题的难解性假设，这类数学对象最典型的实例就是有限域上的(超奇异)椭圆曲线上的 Weil 配偶或 Tate 配偶(它们的数学性质类似于读者在线性代数中学过的二次型)，这里对此给出一个概要描述。

一个双线性群偶 χ 是一个多元组(q, P, G, G_T, e:G×G→G_T)，其中 G 和 G_T 是 q 阶群，q 是素数(因此 G 和 G_T 都是循环群)，P 是 G 的一个生成子。映射 e 有以下性质：对任何整数 m、n 和 U、V∈G 有 $e(mU,nV)=e(U,V)^{mn}$；设 Q_T 是 G_T 的一个生成子，U∈G 是一个任意给定的元素，若对任何 V∈G 都有 $e(U,V)=Q_T$ 则 U 必定是 G 的一个生成子；最后，e 是可以有效计算的。以上三条性质分别称为双线性、正则性和实效性。

双线性群偶 χ=(q, P, G, G_T, e:G×G→G_T)上的计算性双线性 *Diffie-Hellman* 问题(简称 *CBDHP*)是这样一类问题：任给 G 上的元素 U=aP、V=bP、W=cP(但 a、b、c 未知)，求 $e(P,P)^{abc}$。设 k 表示 q 的位数，χ 上的 *CBDH* 问题难解性假设是指对任何 P.P.T.算法 A，概率 P[$a,b,c\leftarrow^{\$}Z_q$;U←$a$P; V←$b$P;W←$c$P;z←A(χ,U,V,W): z=$e(P,P)^{abc}$]至多是复杂性参数 k 的速降

函数。

还有另一类所谓双线性群偶上的判定性 *Diffie-Hellman* 问题难解性假设(简称 *DBDHP* 难解性假设)也经常用到，其涵义是指对任何 P.P.T.算法 A，以下概率的差

$$Adv^{DBDH}_{\chi,A}(k,\chi)\equiv|\,P[a,b,c\leftarrow^{\$}Z_q;\ U\leftarrow aP;\ V\leftarrow bP;\ W\leftarrow cP;\ Z\leftarrow abcP;A(\chi,U,V,W,Z)=1]$$

$$-\ P[a,b,c\leftarrow^{\$}Z_q;U\leftarrow aP;\ V\leftarrow bP;W\leftarrow cP;\ Z\leftarrow^{\$}G;\ A(\chi,U,V,W,Z)=1]|$$

至多是 k 的速降函数，即 A 不能有效区分输入(χ,U,V,W,Z_0)和输入(χ,U,V,W,Z_1)，其中 $U\leftarrow aP$; $V=bP$、$W=cP$、$Z_0=abcP$、Z_1 随机。显然，*DBDH* 问题难解蕴涵 *CBDH* 问题难解。

设(q,P,G_1,G_2,e)是双线性群偶，k 是复杂性参数，H_1: $\{0,1\}^*\rightarrow G_1$、H_2: $G_2\rightarrow\{0,1\}^n$ 是两个随机散列函数，n 是明文消息的字长。*Boneh-Franklin* 方案的组成算法如下。

全局密钥生成算法 Setup(k):

$s\leftarrow^{\$}Z^*_q$; $mpk\leftarrow sP$; $msk\leftarrow s$; return(mpk, msk);

用户私钥生成算法 UKG(msk,a), $a\in\{0,1\}^+$是身份标识, msk=s:

$usk\leftarrow sH_1(a)$; return(usk);

加密算法 E(mpk,a,M):

$r\leftarrow^{\$}Z^*_q$; $T\leftarrow M\oplus H_2(e(H_1(a), mpk)^r)$; $y\leftarrow rP\|T$; return(y);

解密算法 D(mpk,usk,$y_0\|T$):

$M\leftarrow T\oplus H_2(e(usk, y_0))$; return(M);

不难验证以上方案满足一致性条件，事实上有 e 的双线性性质有

$$e(usk, y_0)=e(sH_1(a), rP)= e(H_1(a), P)^{sr}= e(H_1(a), sP)^r=e(H_1(a), mpk)^r$$

故 $T\oplus H_2(e(usk, y_0))=M$ 确实成立。已经证明，若双线性群偶上的 CBDH 问题难解，则 *Boneh-Franklin* 方案保密。注意 *Boneh-Franklin* 方案是密文可塑的，但通过上一节的混合方案，例如 *Fujisaki-Okamoto* 变换，可以非常容易地构成密文非可塑的 IBE 加密方案。习题中还给出了更多的 IBE 方案的例子。

IBE 方案是近来椭圆曲线密码学的一项成就，以往的普通方法很难构造出实用的 IBE 方案。IBE 方案有很多奇妙的应用，例如利用它可以构成这样一类密码方案，用以在一个数据库中检索是否出现特定的关键字，然而一方面数据库中的记录是被全部加密的，另一方面数

据库管理系统本身也不知道被检索的关键字是什么!利用这种技术可以用来实现在公共数据库或文件服务器上的私有数据的托管存储业务，其它方面还有更复杂而新奇的应用，但需要比较复杂的数学准备，这里就不继续深入了。

8.6 特殊密码方案

计算机密码学的内容远远超越加密/解密。除了前面常用的基本密码方案，为开阔读者的视野，这一节再介绍几类特殊的密码方案，它们服务于特殊的目的，但都具有十分现实的意义。实际构造这类方案并精确描述其安全性质是比较复杂的事情，远远超越了本教程的目的，所以这里的介绍仅限于定性的概念性解释。

组群签字方案

组群签名是这样一类数字签名方案，合法签字方是一个集体中的许多成员，他们各自持有不同的签字私钥，但用以验证签字的公钥只有一个，任何合法私钥所生成的签字都能通过验证。组群签字方案也有一个对偶的密码协议形式，这就是所谓身份托管(*identity escrow*)协议，用以鉴别一个对象是否属于一个合法的组群但却不暴露该对象的具体身份。组群签名方案能有效应用与集体授权签字的验证，例如一个企业的领导层中如果不止一名成员对某类事务有签字权，则这类方案是非常实用的，它完全不对签字的验证方暴露领导层中的具体签字者。此外，这类方案还广泛应用于防伪设备生产商的授权验证机制。

时变数字签名方案

时变数字签名方案是这样一种数字签名方案，其公钥固定不变但私钥随时间做周期性更新，并且所有组成算法在每个时间区间(即周期)上都满足一致性条件。时变数字签名方案的安全目标是所谓抗前向伪造性质：即使某一时间区间上的私钥被泄露，据此仍然不能伪造出以往任何时间区间上的数字签名(但可能伪造出未来时间区间上的数字签名)。抗前向伪造性质使得私钥的泄露不会诋毁过去已经生成的数字签名的合法性，在这个意义上，私钥的周期性更新提高了数字签名方案的安全性。在实际应用中，私钥泄露机率比较大的应用环境特别适合于采用这类签名方案。

更精确地说，时变数字签名方案 Ξ=(KG, Upd, Sig, Vf, N)是一组 P.P.T.算法 KG、Upd、Sig 和 Vf。*k* 是复杂度参数，N(k)是时间区间总数，KG 是钥生成算法，输出公钥-初始私钥对(pk,sk(0))；Upd 是私钥更新算法，以当前时间区间序号 i、当前私钥 sk(i)及公钥 pk 为输入并输出下一时间区间上的私钥 sk(i+1)；Sig 是签名算法，以当前时间区间序号 i、当前私

钥 sk(i)和明文 M 为输入并输出签名 σ；Vf 是验证算法，以公钥 pk、明文 M 和字串 σ 为输入并输出验证结果：1 表示接受(验证成功)、0 表示拒绝(验证失败)。KG、Upd、Sig 和 Vf 还满足一致性关系：对所有的 k、i≤N 和 M 恒有 P[(pk,sk(0))←KG(k,N); sk(i)←Upd(pk,sk(i-1),i-1); σ←Sig(sk(i), i, M): Vf(pk, M, σ)=1]=1。不失一般性，总可以假设在时间区间 i 上生成的数字签名 σ 含有分量 i，因此 i 不再显式出现于 Vf(pk, M, σ)的表达式中。

一个著名的时变签名方案见习题 8-29。

时变加密方案

时变公钥加密方案是这样一类公钥加密方案，其公钥不变但对应的私钥随时间变化，从而形成私钥的时间序列 sk(0),sk(1),sk(2),…。私钥在特定的时间区间起始点上被更新，更新计算完全由私钥持有方控制，一旦生成新的私钥 sk(i)，旧的私钥 sk(i-1)即被永久删除。在特定时间区间上，加密计算除了需要公钥参数 pk（不随时间变化）之外，还需要显式输入当前时间区间的编号 i，以使解密方正确解密。因为公钥不变而私钥随时间变化，所以时间区间编号 i 对这类方案的加密计算有实质性的意义。时变公钥加密方案的前向保密性的直观涵义是：即使某一时间区间 i 上的私钥 sk(i)被泄露，凭借 sk(i)也无法破译以前时间区间上生成的任何密文。

很明显，普通(非时变)公钥加密方案是时变公钥加密方案的特例，前者相当于只有一个无限长度的时间区间，而后者的前向安全性提供了比前者的普通安全性更强的能力，保证即使当前的私钥被泄露也能保护历史数据。时变公钥加密方案及前向安全性也可以推广到时变对称加密方案。具有前向安全能力的密码方案特别适合于高威胁环境中的安全计算，这些计算设备被入侵的机率较高，因此私钥被泄露的概率高，前向安全能力保证在入侵发生后受破坏的仅仅是入侵当时和从此以后的安全计算[7]，但对所有以前完成的计算，其信息安全不受破坏。

更精确地说，一个时变公钥加密方案 Π=(KG, Upd, E, D)是一组概率多项式算法 KG, Upd, E 和 D，其中 KG 是钥生成算法，以复杂性参数 *k* 和区间总数 N 为输入并输出公钥及初始私钥偶(pk, sk(0))=KG(k,N)；Upd 是私钥更新算法，以公钥 pk、当前时间区间的编号 i、当前私钥 sk(i)为输入并输出下一区间上的私钥 sk(i+1)=Upd(pk,i,sk(i))；E 是加密算法，以公钥 pk、当前时间区间的编号 i 和明文 M 为输入并输出密文 y=E(pk,i,M)；D 是解密算法，以公钥 pk、当前私钥 sk(i) 、当前时间区间的编号 i 和密文 y 为输入并输出明文 M；所有算法

[7] 在实际应用中, 这一点可以通过其他机制消除, 例如具有入侵检测和自毁能力的设备通过硬机制来抑制入侵的后续影响。

满足一致性关系：对任何 k、i 和 M 恒有

P[(pk, sk(0))←KG(k,N); sk(i)←Upd(pk,i-1,Upd(pk,i-2,…,Upd(pk,0,sk(0))…)); y←E(pk, i, M): D(pk, sk(i), i, y)=M]=1。

时变公钥加密方案的构造比较复杂，直到 2003 年才由著名的密码学家 Canneti、Katz 和 Halevi 借助于层次化的基于身份公钥的加密方案(简称 HIBE，一种比第 8.5 节介绍的 IBE 方案更复杂的加密方案)实现出一类实用的构造，这里不再具体深入了，感兴趣的读者可以参考章末列举的作者的专著。

广播加密方案

广播加密方案针对这样一类用途，其中明文 M 的合法接收者不是单独一个特定的个体，而是一个组群中的任何个体，例如一个软件产品或音像产品的内容 M，凡是合法购买者都应该能够访问 M，而那些非购买者则不应该有能力访问之。这种授权访问的要求很自然地导致对 M 实行加密的想法，而且表面上看来似乎对所有合法访问者给予同一个私钥 sk 不就可以解决问题吗？然而实际问题是：如果这些合法访问者中出现盗版转卖 M 的情况，如何发现这些“背叛者”？更复杂但更现实的情况是，如果多个合法访问者合谋盗版转卖 M，如何识别出其中至少一个“背叛者”？再进一步，所谓对 M 具有合法访问权的组群在现实之中是一个动态的实体，每时每刻都有新成员加入(例如新购买者)，同时也有成员被剔除(例如过期用户或识别出的背叛者)，这些更进一步是问题复杂化。毫无疑问，这正是知识产权保护的密码学模型，也是广播加密最直接的应用领域。

概要地说，广播加密方案具有一个公钥 pk 和任意多个私钥，用 S 表示当前合法成员的组群，对每个合法成员 i∈ S 都分配一个私钥 sk(i)。对明文内容 M 的加密是一种混合方案，即随机生成一个对称密钥 K，用公钥 pk 和与 S 有关的一个参数 Ω(S)加密 K，然后再用 K 加密 M，完整的密文是 $E(pk,\Omega(S),K)\|E^s(K,M)$，其中 E^s 可以是任何一种保密的对称加密算法，真正属于广播加密方案的算法是 E。E 具有这样的特点：存在一个解密算法 D，只有当 i∈S 时才有 D(sk(i), E(pk, Ω(S),K))=K，否则 D 输出一个与 K 无关的随机数。不难看出这一特点保证只有当前 S 的合法成员才能访问到 M。广播加密方案还具有其它附加性质，目的是捕获盗版或合谋盗版成员并加以剔除，这些技术比较深奥，不在此继续深入了。

秘密分割方案

设想对一个明文 M 的加密有这样的要求：M 的密文有 5 个合法解密者，其中(任意)3 人同意才能正确解密。一般地，如果对一个秘密消息 M 的访问策略要求 N 个合法解密者中

至少有 k 人合作才能正确解密，这就需要秘密分割类型的密码方案。这类方案还有很多其他用途，并且需要用到某些特殊的数学构造，感兴趣的读者可以参考章末 Schnerier 和 Ooschort 等的著作。

8.7 小结与进一步学习的指南

Diffie 和 *Hellman* 发表于 1976 年的开创性工作使计算机密码学进入到公钥密码学这一新时代，大量新颖的密码方案随后被创造出来，这些创造体现了公钥密码思想内涵的丰富与深刻。公钥密码学的能力远远不限于加密/解密和数字签名，可以毫不夸张地说当代密码学的方方面面都渗透着公钥密码的思想，从最基本的加密/解密方案到最复杂的网络安全协议，各种形式的公钥密码技术都起着决定性的作用。

本章是对计算机密码学的一个概要性的导引，但内容涵盖了比较前沿的结果，许多内容并没有被其他密码学教科书包括进去。8.1 节所有内容都是大学本科水平的初等数论和近世代数教科书的标准内容，我们只是在此针对密码学的需要而重新组织了材料。从密码学的角度对这些内容的一个非常好的阐述是下面名著，它的前身是加拿大 Waterloo 大学三年级本科生的讲义，值得作为本章内容的主要参考书之一：

[加]D.R.Stinson 著《密码学原理与实践》(第 2 版)，冯登国 译，北京：电子工业出版社，2004

以上课本也概要介绍了椭圆曲线密码学。椭圆曲线上的点群对密码学的基本价值在于，它上面的一般性离散对数问题比通常有限域上的一般性离散对数问题“还要难解”[8]，因此能以更短的密钥(从而更高的计算与带宽效率)达到同样的安全强度，或(等价地)以同样长度的密钥达到高得多的安全强度。椭圆曲线对现代密码学的另一方面的价值在于，它上面有更丰富的结构可以利用，例如 *Weil* 配偶和 *Tate* 配偶这样的双线性结构，可以提供构造公钥方案秘密陷门的全新途径。

密码学领域的一部百科全书式的巨著是[加]S.Ooschort A.Menezes D.Vanstone 著 《应用密码学手册》，胡磊 等译，北京：电子工业出版社，2005。原著出版于 1997 年，虽然作为研究工作者的参考书来讲已略显过时，但作为有志于密码学领域的学生，这仍然是一部值得非常认真学习的著作。包含更近期内容的教科书是：

[8] 对椭圆曲线上的一般性离散对数问题，已知的算法都是指数复杂度的，目前还没有发现所谓亚指数算法，但对有限域，例如 F_p，多年以前就发现了亚指数算法。

[美]W.Mao 著《现代密码学理论与实践》(影印版)，北京：电子工业出版社，2005

该书作者指出在这部教科书中将有意识地摈弃所谓“教科书版”密码方案而更致力于解释“工业版”的方案和协议，对此作者深为赞同。这部著作还讨论了基于身份的加密方案和安全证明理论(最后几章)，这在其他教科书中是很少有的，值得一读。对密码学领域的研究生而言，更高深的专著可以参考：

田园 著《计算机密码学：通用方案构造及安全证明》北京：电子工业出版社，2008

书中系统阐述了密码方案及协议的安全证明技术。关于相关数论算法的精彩讨论可以参考前述 Stinson 的著作第 5、6 两章和 Ooschort 等著作的第 14 章，以及 E.Knuth 的名著 *The Art of Computer Programming* 第 2 卷第 4 章。

关于因特网的公钥基础设施和公钥证书的详细的技术规范是 RFC 2509 和 X.509 协议。S.Ooschort 等的著作也对此有详细阐述。关于各类对称加密方案的算法性描述及评论，可以参考下面这部名著[9]，该书的附录中包括许多密码方案的 C 源程序：

B.Schnerier《应用密码学：算法、协议和 C 源程序》，吴世忠 等译，北京：机械工业出版社，2000。

习 题

8-1 证明：任给整数 a 和 b，如果存在整数 x_0 和 y_0 满足 $ax_0+by_0=(a,b)$，则任何满足方程 $ax+by=(a,b)$的整数 x 和 y 必有形式 $x=x_0+kb, y=y_0-ka$，其中 k 是任何整数(因此方程 $ax+by=(a,b)$ 的整数解 x 和 y 不唯一)。

8-2 按以下步骤证明“任给整数 a 和 b，必存在整数 x 和 y 满足 $ax+by=(a,b)$”：考虑 a 和 b 的整系数线性组合所构成的集合 $D=\{ ax+by: x,y\in Z\}$，令 d^*是 D 中最小的正数。(1)对 a 和 b 的任何公因子 d，证明 $d|d^*$和 $d^*|b$ 必成立；(2)用原始的 Euclid 定理证明 $d^*|a$，$d^*|b$。综合这

[9] Schnerier 的书第一版出版于 1996 年, 中译本依据的就是这一版本。Schnerier 的书对密码知识的传播起了非常积极的作用, 美国著名的密码学者曾有评论这是“NSA(美国国家安全局)永远都不愿意看到出版的书”, 但这部书包含很多细节上的错误或不确切之处, 在密码学界也是众所周知的。这里作者可以提一个从一位很著名的英国同行那里听说的笑话: 几年前某位学者声称发现了破译著名的散列函数 MD5 的方法(对 MD5 的详细描述见 Schnerier 的书第一版第 18 章), 著名的密码学家 E.Biham 听说后不相信, 即使发现者当面用其软件向 Biham 演示生成 MD5 的一对冲突后 Biham 仍很怀疑。当 Biham 得知这位学者是根据 Schnerier 的书获得关于 MD5 的知识时顿时恍然大悟, 为自己的怀疑找到了证据, 因为这本书所描述的关于 MD5 的某些微妙的细节并不确切。关于 MD5 的权威设计文档应该是 R.Rivest(他也是 RSA 方案的发明者之一) 1992 年所撰写的 RFC#1321“The MD5 Message-Digest Algorithm”(读者可以从 www.ietf.org 获得), 在仔细查阅了 RFC#1321 后这一错误得到了纠正。当然这位学者最终仍然得到了正确的结果, MD5 确实存在一个较弱意义上的安全缺陷。这里作者毫不怀疑 Schnerier 的治学态度与能力, 只是借这个故事再次告诉读者科学创造的曲折与不易, 同时在密码学研究领域凡事谨慎!

Schnerier 的书对初学者是一个很全面的导引。他后来创立著名的安全服务公司 counterpane(网址 bt.counterpane.com)并出任第一届总裁。目前公认的高效对称加密方案之一的 Blowfish 方案就是 Schnerier 的发明, 而且 Schnerier 主动放弃了对 Blowfish 的发明专利。

两点结论，便知 d*=(*a*,*b*)，并且显然存在 *x*=*x**和 *y*=y*使(*a*,*b*)=*ax*+*by*。

8-3 设 *a*=*b* mod N, *m*∈Z，验证以下关系成立：

(1) (*a*±*m*) mod N = (*a* mod N ± *m* mod N) mod N，因此 *a*±*m* mod N 即有明确的意义；

(2) (*am*) mod N = ((*a* mod N)(*m* mod N)) mod N，因此 *am* mod N 即有明确的意义；

(3) *a*+*m* = (*b*+*m*) mod N；

(4) *am* = *bm* mod N；

(5) *f*(*a*)= *f*(*b*) mod N，*f*(x)是任意给定的整系数多项式。

8-4 对线性同余式 *ax*=*b* mod N，(*a*,N)|*b*，按以下步骤分析方程的可解性和解的数目：

(1)先假设(*a*,N)=1 这一特殊情形，由第一种等价形式的 Euclid 定理知存在整数 *u* 和 *v* 使 *au*+*Nv*=1，两边乘以 b 便导出同余关系 *a*(*ub*)=*b* mod N，即 *ax*=*b* mod N 存在解 *ub* mod N，然后再证明这时解唯一。

(2) 对(*a*,N)|*b* 的情形，令 *d*=(*a*,N)，则可以写 *a*=*a***d*，*b*=*b***d*，N=N**d* 且(*a**,N*)=1，于是方程 *ax*=*b* mod N 可以变形为 *a***x**=*b** mod N*，由(1)知这时解 *x**存在，两边乘以 *d* 便导出同余关系 *ax**=*b* mod N，即 *ax*=*b* mod N 存在至少一个解 *x**。至于这时解的数目，可以令 x 是任何一个解：*ax*=*b* mod N，与 *ax**=*b* mod N 相减得 N|*a*(*x*−*x**)(为什么?)，由此得 N*|*a**(*x*−*x**)，即 *x*=*x**+*k*N*(为什么?)，由此导出 0≤*x*<N 的不同的解恰有 *d* 个(为什么?)。

8-5 (用手算)求解以下同余式，或判定该方程无解：

(1) 7x= 1 mod 11 (2) 9x=7 mod 10 (3) 5x=3 mod 12 (4) 2x=3 mod 16 (5) 6x=3 mod 19

8-6 设算法 A(*a*,N)求解两个整数的最高公因子(*a*,N)及系数 *x*,*y*: (*a*,N)=*ax*+*by*，算法 B(*a*,*b*,N)求解线性同余方程 *ax*=*b* mod N，两个算法的时间复杂度分别记做 T_A 和 T_B。

(1) 以 A 为子程序构造 B，并以 T_A 表达 T_B 的上界；

(2) 以 B 为子程序构造 A，并以 T_B 表达 T_A 的上界。

8-7 用中国余数定理求解以下方程：

(1) x=1 mod 7, x=5 mod 9, x=3 mod 11

(2) x=4 mod 7, x=1 mod 9, x=2 mod 11

(3) x=3 mod 5, x=5 mod 7, x=3 mod 12

(4) x=2 mod 5, x=3 mod 7, x=1 mod 12

(5) x=1 mod 5, x=1 mod 7, x=1 mod 19

8-8 m_1、…、m_n 是两两互素的一组正整数，*f*(x)是一个整系数多项式，整数 a_1、…、a_n 使 $f(a_i)=0 \mod m_i$, i=1,…,n。令 $M=m_1 \ldots m_n$，证明：若整数 *x* 满足线性同余式组 $x=a_i \mod m_i$,

i=1,…,n，则 $f(x)=0 \bmod M$。

8-9 (用手算)计算 φ(256)、φ(49)、φ(118)。

8-10 用 Fermat 定理或 Euler 定理计算 $3^{201} \bmod 11$、$3^{1025} \bmod 15$。

提示：201 mod φ(11)=？1025 mod φ(15)=？

8-11 完整给出 Euler 定理的证明。

8-12 记号 F_p 的涵义如 8.1.4 节的定义，系数属于 F_p 的多项式的集合记做 $F_p[x]$。

(1) 证明：按照多项式通常的加法运算和乘法运算，F_p 都构成群，注意这是无限群的例子。

(2) n 阶多项式 $f(x)\in F_p[x]$称为可约的，如果 $f(x)$能分解为 $F_p[x]$中阶数全部严格小于 n 的两个多项式的乘积。不可约的多项式称为素多项式，任何 $f(x)\in F_p[x]$在 $F_p[x]$中都有唯一的素因子分解[10]，因此对任何两个多项式 $g(x)$、$h(x)\in F_p[x]$都可以明确定义最高公因子，用记号(g(x),h(x))表示。证明对多项式也有相应的 Euclid 定理及等价形式：

任给 $g(x)$、$h(x)\in F_p[x]$，必存在唯一的 $q(x)$、$r(x)\in F_p[x]$满足 $f(x)=g(x)q(x)+r(x)$及 $0\le \deg(r(x))<\deg(g(x))$，deg 表示多项式的阶。

任给 $g(x)$、$h(x)\in F_p[x]$，总存在 $u(x)$和 $v(x)\in F_p[x]$(但不唯一)满足 $g(x)u(x)x+h(x)v(x)=(g(x),h(x))$。

任给 $f(x)$、$g(x)$、$h(x)\in F_p[x]$，则存在 $u(x)\in F_p[x]$满足线性同余式 $g(x)u(x)=h(x) \bmod f(x)$当且仅当$(g(x), f(x))|h(x)$，且这时 u 唯一。特别地，若 g、f 互素则 $g(x)u(x)=1 \bmod f(x)$ 必存在解 $u\in F_p[x]$。

8-13 从 $F_p[x]$取一个 n 阶不可约多项式 $f(x)$(见上一节的定义)，记 $F_p[x]/(f)\equiv\{h(x) \bmod f(x): h(x)\in F_p[x]\}$、$F^*_p[x]/(f)\equiv F_p[x]/(f)\backslash\{0\}$，并且在 $F_p[x]/(f(x))$上定义模 $f(x)$的多项式加法+和多项式乘法*。验证$(F_p[x]/(f),+)$和$(F^*_p[x]/(f),*)$都是群，阶分别为 p^n 和 p^n-1。

注：$F_p[x]/(f)$称做特征为 p 的 n 次扩域。素域 F_p 和扩域 $F_p[x]/(f)$统称为有限域，其元素的个数称为有限域的阶，它总是某个素数 p 的幂，p 称做有限域的特征。

8-14 对特征为 p 的素域或扩域(见上一题)中的任意元素 u、v，证明以下恒等式：

(1) pu=0

(2) $(u\pm v)^p=u^p\pm v^p$

(3) 若扩域的次数为 n，则 $(u\pm v)^{p^n}=u^{p^n}\pm v^{p^n}$。

8-15 G 是抽象群，a,b,c∈G：

[10] 这一性质的证明并不简单，直到 19 世纪初才由 Gauss 所证明，感兴趣的读者可以参考任何一本近世代数的课本。

(1) 证明：若 ab=ac 则 b=c；提示：两边乘以 a^{-1}.

(2) 仿 Fermat 定理的证明，对抽象群 G 给出 Lagrange 定理的完整证明。

8-16 群 G 的元素 $a\neq e$ 的阶是使得 $a^d=e$ 的最小正整数 d，证明：d 一定整除 G 的阶 N。

提示：从 Lagrange 定理有 $a^N=e$，又由 Euclid 定理总可以设 N=qd+r，其中 $0\leq r<d$，于是 $a^r=e$(为什么?)从而 r=0(为什么?)。

8-17 证明：$(F^*_p,*)$作为循环群(这里承认这一事实而不予证明)，其生成子一共恰有 φ(p-1)个。

提示：设 g 是生成子，则其他生成子总有形式 g^i，其中 i 是 1 和 p−1 之间的某个整数(为什么？)，这些 i 具有什么特殊性质？

8-18 这一习题的目的是证明因子分解问题与求平方根问题难度等价。N=pq，p 和 q 是(很大的)素数，a 是与 N 互素的整数且使 $x^2=a \bmod N$ 存在解。

(1) 若有算法 A 使 A(N)输出素因子 p 和 q，证明 $x^2=a \bmod N$ 的解可以由以下过程得到：

第一步：分别求解 $u^2=a \bmod p$ 和 $v^2=a \bmod q$，得解 u_1,u_2,v_1,v_2；

第二步：对每一个组合(u_i,v_j)由中国剩余定理计算 x：$x=u_i \bmod p$, $x=v_j \bmod q$。

由以上算法可以看到，$x^2=a \bmod N$ 一般地有四个解。

(2) 若存在一个算法 B，任给与 N 互素的整数 a 且 $x^2=a \bmod N$ 存在解(这由其他方法判定)，B(a,n)输出以上方程的四个解，证明以下算法可以得到 N 的素因子：

第一步：任取与 N 互素的 y，计算 $a=y^2 \bmod N$；

第二步：调用算法 B(a,N)输出 x_1, x_2, x_3, x_4，设其中 $x_3=+y \bmod N$, $x_4=-y \bmod N$；

第三步：由 Euclid 算法计算(n, x_1+x_3)，则输出必是 p 或 q 之一。

8-19 这一习题的目的是建立一个通用算法，由已知 N 的素因子分解求解任意的多项式方程 $f(x)=0 \bmod N$(但不考虑对 N 求其因子分解这一问题本身)。

第一步：由素因子分解式 $N=\Pi p^n$ 及中国剩余定理，解多项式方程 $f(x)=0 \bmod N$ 归结为对 N 的每个素因子 p 解多项式方程 $f(x)=0 \bmod p^n$ (参见习题 8-8)。

第二步：由下面的 Hensel 提升算法[11]将求解 $f(x)=0 \bmod p^n$ 归结为求解 $f(x)=0 \bmod p$。

Hensel 提升算法从解方程 $f(x)=0 \bmod p$ 开始，然后依次求解方程 $f(x)=0 \bmod p^2$、$f(x)=0 \bmod p^3$、…、$f(x)=0 \bmod p^n$, 从而将 $f(x)=0 \bmod p^n$ 的求解归结为对 $f(x)=0 \bmod p$ 的求解。具体迭代方法如下。设 x(i)是 $f(x)=0 \bmod p^i$ 的解且 $f'(x(i))\neq 0 \bmod p$，$f'(x)$ 是 $f(x)$ 的导数。不妨设

[11] Hensel 是一位德国犹太裔中学教师兼业余数学家，这一算法发现于 20 世纪 20 年代前后，属于所谓 p-adic 数的应用的一部分。Hensel 关于 p-adic 数理论的发现为近代数论开创了一个崭新局面。

$f(x(i))=A(i)p^i$(为什么?)，从而线性同余式 $A(i)+f'(x(i))y=0 \bmod p$ 对 y 可解(为什么?)，由欧氏算法解出 y, $1\le y\le p-1$。注意 $x(i+1)=(x(i)+yp^i) \bmod p^{i+1}$ 是 $f(x)=0 \bmod p^{i+1}$ 的解(提示：因为若令 $x(i+1)=(x(i)+yp^i) \bmod p^{i+1}$(为什么?)则 $f(x(i+1))=f(x(i))+f'(x(i))yp^i \bmod p^{i+1}=(A(i)+f'(x(i))y)p^i \bmod p^{i+1}$，因此 y 所满足的方程 $A(i)+f'(x(i))y=0 \bmod p$ 恰好能使 $f(x(i+1))=0 \bmod p^{i+1}$)。

特别地，((a/p)=1 时)已知对素数 p 存在有效算法求解 $x^2=a \bmod p$, 由此可以导出求解 $x^2=a \bmod p^n$ 的有效算法。

8-20 (*Fegei-Fiat-Shamir* 数字签名方案) p、q 是 *k* 位秘密素数，N=pq，$H:\{0,1\}^+\to\{0,1\}^k$ 是抗冲突的散列函数，N、*H* 公开, p、q 保密。方案的组成算法如下：

公钥/私钥生成算法 KG(k, *G*, g, q):

$x\leftarrow^{\$}\{1,2,\ldots,N-1\}$; $y\leftarrow^{\$}x^2 \bmod N$; vk←y; sk←x; return(vk,sk);

公钥和私钥分别为 vk 和 sk。

签名算法 Sig^H(sk, M), 其中 sk=x:

$r_i\leftarrow^{\$}Z_N$, $v_i\leftarrow r_i^2 \bmod N$, i=1,…,k;

$h\leftarrow H(M, v_1,\ldots,v_k)$;

$z_i\leftarrow r_i x^{e_i} \bmod N$, e_i 是 h 的第 i 位, i=1,…,k;

$\sigma_1\leftarrow v_1\ldots. v_k$; $\sigma_2\leftarrow z_1\ldots. z_k$;

return(σ_1,h,σ_2); /*对 M 的数字签名*/

验证算法 Vf^H(vk, M, (σ_1,h,σ_2)), 其中 vk=y:

parse σ_1 as $v_1\ldots. v_k$;

parse σ_2 as $z_1\ldots. z_k$; parse h as $e_1\ldots. e_k$;

return(h=H(M, σ_1) $\wedge_{i=1,\ldots,k}$ $z_i^2=v_i y^{e_i} \bmod N$);

验证该方案满足一致性条件，并解释为什么算法 Sig^H(sk, M)必须随机独立地生成各个 r_i。

8-21 S=(KG, Sig, Vf)是一个抗伪造的数字签名方案，KG, Sig, Vf 分别是签字私钥/公钥生成算法、签字算法和验证算法。*H* 是一个散列算法，将任意的字符串 M 映射到 S 的消息域。做以下签名方案 S^H，其私钥/公钥生成算法就是以上的 KG，签字算法 Sig^H(sk, M)= Sig(sk, H(M))。注意 S^H 比 S 的优越之处在于计算效率：H(M)通常比 M 短得多而且长度固定，例如取 H 为 MD5 或 SHA，则无论 M 多长 H(M)总是固定的 128 位或 160 位，所以计算效率更高。

(1) 请给出方案 S^H 的验证算法 Vf^H(vk, M, σ)并证明你的算法满足一致性条件；

(2) 如果存在一个有效算法 A 可以算出 H 的一个冲突，即 A 能有效计算出一对不同的消息 $M_1 \neq M_2$ 使 $H(M_1)=H(M_2)$，则 A 可以用来伪造签名方案 S^H 的数字签名，即 S^H 不能抵抗伪造攻击(虽然 S 抗伪造攻击)，为什么？这表明要使 S^H 抗伪造攻击，H 必须抗冲突。

8-22(*Paillier-Damgard-Catalano* 公钥加密方案,1999,2000,2002) 设 $N=p_1p_2$ 是有两个大素因子的 RSA 模数，正整数 $s<p_1,p_2$，首先我们接受一条普遍性质：在模 N^{s+1} 的乘法群 $Z^*_{N^{s+1}}$ 中 1+N 的阶为 N^S。以下讨论涉及 s=1 的情形。公钥 pk=RSA 公钥(e,N)；私钥 $sk=d$，其中 $ed=1 \bmod \varphi(N)$；明文 $m \in Z_N$；加密算法 $E(pk, m)$输出密文 $y \leftarrow (1+mN)r^e \bmod N^2$，其中 $r \leftarrow^{\$} Z^*_N$；对密文 y 的解密运算如下：$r \leftarrow y^d \bmod N$; $m \leftarrow ((r^{-e}y-1) \bmod N^2) / N$。已经证明：若 N 上的所谓判定性 e-次剩余问题难解，则以上方案保密。

(1) 验证以上加密算法是正确的，即满足一致性条件；

(2) 验证以上方案具有同态性质：$E(pk, m_1)E(pk, m_2)= E(pk, (m_1+m_2) \bmod N) \bmod N^2$。

注：同态性质使这一方案密文可塑，因此作为纯粹的加密方案并不可取。*Paillier-Damgard-Catalano* 公钥加密方案的真正用途在于作为构造许多复杂安全协议的工具，在这方面其同态性质具有重要应用。

8-23 对明文 M，*ElGamal* 方案(例 8-4)的合法密文 $y=(y_1,y_2)$, $y_1=g^r$, $y_2=g^{xr}M$。用直接的计算验证：

(1) 对任何常数 $a \in G$，$y^*=(y_1,ay_2)$是一个合法的密文且解密出的明文将是 aM；

(2) 对任何整数常数 b，$y^*=(y_1^b,y_2^b)$是一个合法的密文且解密出的明文将是 M^b；

(3) 对任何常数 $a \in G$ 和整数常数 b，$y^*=(y_1^b,ay_2^b)$是一个合法的密文且解密出的明文将是 aM^b。

8-24 在应用 RSA 方案时，不同的合法解密者应该被赋予不同的公钥模数 N 且另一个公钥分量即指数 e 不应该彼此互素(实际上一般取 e 都相同且等于 3，取 3 是为了使加密计算的效率最高)。原因如下：如果 RSA 模数相同且都等于 N，并且各个公钥指数 e 都彼此互素，若某个发送者需要将同一个消息 M 分别发送给 B 和 C，于是该发送者生成两个密文 $y_B=M^{e_B} \bmod N$ 和 $y_C=M^{e_C} \bmod N$。攻击者 A 截获到 y_B 和 y_C 后，将能够很容易地计算出来 M，为什么？注意对 *OAEP/RSA* 方案也有同样的问题。

提示：既然 e_B 和 e_c 公开而且互素，攻击者 A 可以由 Euclid 算法(8.1.1 节)计算出整数 u 和 v，使之满足 $e_Bu+e_Cv=(e_B, e_C)=1$。现在，$y_B^u y_C^v \bmod N$ 等于什么？

8-25 在几乎所有密码方案中都需要计算大数的幂，其中的指数一般来说也是很大的整数，例如本章的各种例子，因此这一习题的目的是建立一个非常通用的快速幂算法。给定 g 和指数 x，要计算 $y=g^x$(或 $g^x \bmod N$。因为模 N 运算属于群运算的定义的一部分，在下面省去符号 mod N，但读者应该记住所有乘法实际上都是模 N 的剩余乘法)。记 x 的 2-进制表达式为

$x(0)+2x(1)+2^2x(2)+\ldots+2^{n-1}x(n-1)$, 其中 $x(i)=0$ 或 1，再记

$$y(i) = g^{2^{n-1-i}x(n-1)+2^{n-2-i}x(n-2)+\ldots+2x(i+1)+x(i)},\ i=0,1,\cdots,n-1$$

(1)证明以下递归公式：

$$y(0)=y;\quad y(n-1)=g^{x(n-1)}=\begin{cases}1: x(n-1)=0\\ g: x(n-1)=1\end{cases}$$

$$y(i)=g^{x(i)}y(i+1)^2=\begin{cases}y(i+1)^2: x(i)=0\\ gy(i+1)^2: x(i)=1\end{cases},\ i=0,1,\cdots,n-2$$

(2) 根据以上公式写出一个计算 $y=g^x$ 的算法，其中主要的子程序是平方运算，并估计该算法的复杂度，即乘法运算的次数。

(3) 你能给出一个计算 $g_1^{x_1}g_2^{x_2}$ 的快速算法吗？

8-26 (1) 验证 REACT 混合方案和 GEM 混合方案的加密/解密算法满足一致性条件；

(2) 用 *ElGamal* 方案和 *Boneh-Franklin* 方案为基本公钥加密方案，代入 8.4.2 小节的混合加密结构，给出具体的实例化混合加密方案的算法。

8-27 (*Waters* IBE 方案,2005) 设(p, P, G_1, G_2, e)是双线性群偶，p 是 *k* 位素数，*Waters* IBE 方案组成算法如下：

全局公钥/私钥生成算法 Setup(k):

$Q\leftarrow^{\$}G_1$; $\alpha\leftarrow^{\$}Z_p$; $P_1\leftarrow\alpha P$; $Q_1\leftarrow\alpha Q$;

$U[0..n]\leftarrow^{\$}G_1^{n+1}$; $E\leftarrow e(P, Q)$;

$mpk\leftarrow(G_1,G_2,p,e,P, P_1,U,E)$;

$msk\leftarrow(mpk,Q_1)$;

return(mpk, msk);

用户私钥生成算法 UKG(msk,*a*), $a=a(1)\ldots a(n)\in\{0,1\}^n$, $msk=((G_1,G_2,p,e,P, P_1,U,E), Q_1)$:

$r\leftarrow^{\$}Z_p$; $V\leftarrow U[0]+\sum_{i=1}^{n}a(i)U[i]$;

$usk(a)\leftarrow(Q_1+rV, rP)$;

return(usk(*a*));

加密算法 E(mpk,*a*,M), $mpk=((G_1,G_2,p,e,P, P_1,U,E), Q_1)$, $M\in G_2$:

$$V\leftarrow U[0]+\sum_{i=1}^{n}a(i)U[i];$$

$t \leftarrow^{\$} Z_p$; $T \leftarrow E^t$;

$y \leftarrow (TM, tP, tV)$;

return(y);

解密算法 D(mpk,usk(*a*),y), usk(*a*)=(s_1, s_2), y=(y_1, y_2, y_3):

$T \leftarrow e(s_1, y_2)e(s_2, y_3)^{-1}$;

return($y_1 T^{-1}$);

(1) 验证这一方案满足一致性条件；

(2) 解释这一方案为什么密文可塑，并给出几种密文变形的可能方式；

(3) 利用第 8.4.2 节介绍的混合加密结构，具体给出以该 IBE 方案为其中公钥加密方案的混合加密方案实例。

注：已经证明若双线性群偶上的 *DBDH* 问题难解，则 *Waters* 方案保密。

8-28(*Boyen-Waters* IBE 方案,2006) 给定双线性群偶 $\mathcal{J}=\{(p,G_1,G_2,e)$，其中$|G_1|=|G_2|=p$、p 是 *k* 位素数、$P \in G_1$、$e: G_1 \times G_1 \to G_2$ 是非蜕化的双线性映射，*Boyen-Waters* 方案的组成算法如下：

全局公钥/私钥生成算法 Setup(k):

$g, g_0, g_1 \leftarrow^{\$} G_1$; $\omega, t_1, t_2, t_3, t_4 \leftarrow^{\$} Z_p$; $\Omega \leftarrow e(g,g)^{t_1 t_2 \omega}$;

$v_1 \leftarrow g^{t1}$; $v_2 \leftarrow g^{t2}$; $v_3 \leftarrow g^{t3}$; $v_4 \leftarrow g^{t4}$;

mpk$\leftarrow (G_1, G_2, p, e, \Omega, g, g_0, g_1, v_1, v_2, v_3, v_4)$;

msk$\leftarrow (\omega, t_1, t_2, t_3, t_4)$;

return(*mpk*, *msk*);

用户私钥生成算法 UKG(msk, *a*), $a \in Z_p$:

$r_1, r_2 \leftarrow^{\$} Z_p$;

usk(*a*)$\leftarrow (g^{r_1 t_1 t_2 + r_2 t_3 t_4}, g^{-\omega t_2}(g_0 g_1^{\ a})^{-r_1 t_2}, g^{-\omega t_1}(g_0 g_1^{\ a})^{-r_1 t_1}, (g_0 g_1^{\ a})^{-r_2 t_4}, (g_0 g_1^{\ a})^{-r_2 t_3})$;

return(usk(*a*));

加密算法 E(*mpk*, *a*, *M*), $M \in G_2$:

$s, s_1, s_2 \leftarrow^{\$} Z_p$;

$\xi \leftarrow (\Omega^s M, (g_0 g_1^{\ a})^s, v_1^{s-s1}, v_2^{s1}, v_3^{s-s2}, v_4^{s2})$;

return(ξ);

解密算法 D(*mpk*, usk(*a*), $(\xi_{00}, \xi_0, \xi_1, \xi_2, \xi_3, \xi_4)$), usk(*a*)$\equiv (d_0, d_1, d_2, d_3, d_4)$:

$T \leftarrow e(d_0, \xi_0)e(d_1, \xi_1)e(d_2, \xi_2)e(d_3, \xi_3)e(d_4, \xi_4)$;

return($\xi_{00}T$);

(1) 验证这一方案满足一致性条件；

(2) 解释这一方案为什么密文可塑，并给出几种对密文变形的可能方式；

(3) 利用第 8.4.2 节介绍的混合加密结构，具体给出以该 IBE 方案为其中公钥加密方案的混合加密方案实例。

注：已经证明若双线性群偶 $\mathcal{J}$ 上的 DBDHP 难解则该 IBE 方案保密。

8-29(时变 *GQ* 签字方案,1999) 设 k 是复杂度参数，L(k)是密钥中分量的个数，T(k)是时间区间总数，H:$\{0,1\}^{+}\rightarrow\{0,1\}^{L}$ 是随机散列函数，时变签字方案 Ξ_{GQ}=(KG, Upd, Sig, Vf, T)的 P.P.T. 算法 KG、Upd、Sig 和确定性算法 Vf 描述如下。

钥生成算法 KG(k,L,T):

随机生成 k 位素数 p 和 q、p=q=3 mod 4;

N←pq;

for j=1,…,L do{ $S_j\leftarrow^{\$}Z^*_N$; $U_j\leftarrow S_j^{2^{T+1}} \bmod N$; }

pk←(N,T,U_1,…,U_L);

sk←(N,T,0,$S_1(0)$,…,$S_L(0)$); /*对每个 j 记 $S_j(0)=S_j$*/

return(pk,sk(0))；

私钥更新算法 Upd(pk,sk(i),i): /*sk(i)=(N,T,i,$S_1(i)$,…,$S_L(i)$)*/

for j=1,…,L do $S_j(i+1)\leftarrow S_j(i)^2 \bmod N$;

sk(i+1)←(N,T,i+1,$S_1(i+1)$,…,$S_L(i+1)$);

delete(sk(i));

return(sk(i+1)); /*下一时间区间上的私钥*/

签名算法 Sig^H(sk(i), i, M):

$R\leftarrow^{\$}Z^*_N$; $Y\leftarrow R^{2^{T+1-i}} \bmod N$;

$h_1\ldots h_L\leftarrow H(Y||M||i)$; /*$h_j\in\{0,1\}$*/

$Z\leftarrow R\prod_{j=1}^{L} S_j^{h_j}(i) \bmod N$;

return($Y||Z||i$);

验证算法 Vf^H(pk, M, σ): /*pk(i)=(N,T,U_1,…,U_L), σ=$Y||Z||i$.*/

$h_1\ldots h_L\leftarrow H(Y||M||i)$; /*$h_j\in\{0,1\}$*/

$$if \ \ Z^{2^{T+1-i}} = Y\prod_{j=1}^{L} U_j^{h_j} \bmod N \ then \ \text{return(1)} \ else \ \text{return(0)};$$

请验证以上方案满足一致性条件。

注：*Bellare-Milner* 已经证明，若钥生成算法 KG 所生成的整数 N 的素因子分解问题难解则以上签名方案必有前向抗伪造性质。

8-30(*Lagrange* 插值公式) p 是素数，$f(x)$是一个系数属于 F_P*的 n 次多项式(因此这些系数有模 p 乘法的逆，或等价地，这些数之间可以做除法)，则 $f(x)$由 n+1 个不同的点 $x(i) \in F_P$*的上的值 $f(x(i))$(i=0,1,2,…,n)完全确定，具体公式是

$$f(x)=\sum_{i=0}^{n} f(x(i)) \prod_{j=0: j\neq i}^{n} (x-x(j))(x(i)-x(j))^{-1} \bmod p$$

验证以上公式的正确性。这个公式是很多秘密分割密码方案的基础。