第 3 章　密码与身份认证技术

3.1　密码学基本概念

3.1.1　密码学的定义和作用

密码学是主要研究通信安全和保密的学科，它包括两个分支：密码编码学和密码分析学。密码编码学主要研究对信息进行变换，以保护信息在传递过程中不被敌方窃取、解读和利用的方法；而密码分析学则与密码编码学相反，它主要研究如何分析和破译密码。这两者之间既相互对立又相互促进。

密码学的基本思想是对机密信息进行伪装。一个密码系统完成如下伪装：加密者对需要进行伪装的机密信息（明文）进行变换（加密变换），得到另外一种看起来似乎与原有信息不相关的表示（密文）。如果合法用户（接收者）获得了伪装的信息，可以通过事先约定的密钥，从伪装的信息中分析得到原有的机密信息（解密变换）；而如果不合法的用户（密码分析者）试图从这种伪装的信息中分析得到原有的机密信息，这种分析过程要么是根本不可能的，要么代价过于巨大，以致无法进行。

图 3.1 给出了在互联网环境下使用加密技术的加密和解密过程。

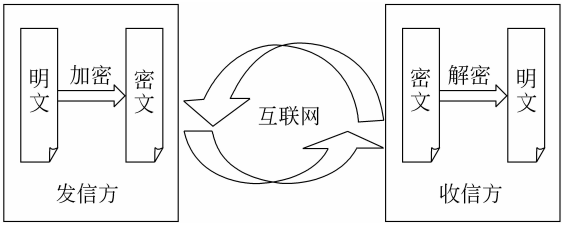

图 3.1　加密和解密过程

使用密码学可以达到以下目的：

（1）保密性。防止用户的标识或数据被读取。

（2）数据完整性。防止数据被更改。

（3）身份验证。确保数据发自特定的一方。

3.1.2　密码学的发展历程

人类有记载的通信密码始于公元前 400 年。密码学的起源可以追溯到人类刚刚出现并且尝试去学习如何通信的时候，为了确保通信的机密性，最先是有意识地使用一些简单的方法来加密信息，通过一些（密码）象形文字传达信息。随后，由于文字的出现和使用，确保通

信的机密性就成为一种艺术，古代发明了不少加密信息和传达信息的方法。例如，我国古代的烽火是一种传递军情的方法，古代的兵符是用来传达信息的密令，闯荡江湖的侠士都有秘密的黑道行话，起义军在起义前约定地下联络的暗语，这都促进了密码学的发展。

而密码学真正成为科学是在19世纪末和20世纪初期，由于军事、数学和通信等相关技术的发展，特别是两次世界大战中对军事信息保密传递和破获敌方信息的需求，使密码学得到了空前的发展，并广泛地用于军事情报部门的决策。

太平洋战争中，美军破译了日本海军的密码，读懂了日本舰队司令官山本五十六发给各指挥官的命令，在中途岛彻底击溃了日本海军，导致了太平洋战争的决定性转折，而且不久后还击毙了山本五十六。德国在第二次世界大战的初期在密码破译方面占据着优势地位，德国于战争期间使用的密码机Enigma如图3.2所示。因此，可以说，密码学在战争中起着非常重要的作用。

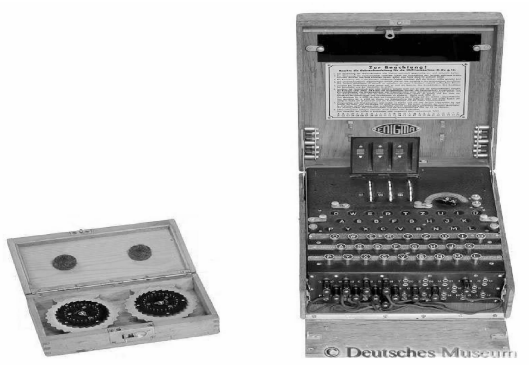

图3.2　德国密码机Enigma

1883年，Kerchoffs第一次明确提出了编码的原则：加密算法应在算法公开时不影响明文和密钥的安全。这一原则已得到普遍承认，成为判定密码强度的标准，实际上也成为传统密码和现代密码的分界线。

随着信息化和数字化社会的发展，人们对信息安全和保密的重要性的认识不断提高。网络银行、电子购物和电子邮件等正在悄悄地融入人们的日常生活中，人们自然要关注其安全性。1977年，美国国家标准局公布实施了美国数据加密标准(DES)，军事部门垄断密码的局面被打破，民间力量开始全面介入密码学的研究和应用中。民用的加密产品在市场上大量出现，采用的加密算法有DES、IDEA和RSA等。

现有的密码体制类型繁多，各不相同。但是它们都可以分为私钥密码和公钥密码两类。前者的加密过程和解密过程相同，而且所用的密钥也相同；后者，每个用户都持有一对密钥。数据加密的模型如图3.3所示。

密码编码学主要致力于信息加密、信息认证、数字签名和密钥管理方面的研究。信息加密的目的在于将可读信息转变为无法识别的内容，使得截获这些信息的人无法阅读，同时信息的接收人能够验证接收到的信息是否被敌方篡改或替换过。数字签名就是信息的接收人能够确定接收到的信息是否确实是由所希望的发信人发出的。密钥管理是信息加密中最难的部分，因为信息加密的安全性取决于密钥。历史上，各国军事情报机构在猎取别国的密钥管理方法上要比破译加密算法成功得多。

密码分析学与密码编码学的方法不同，它不依赖于数学逻辑，必须凭经验，依赖客观世

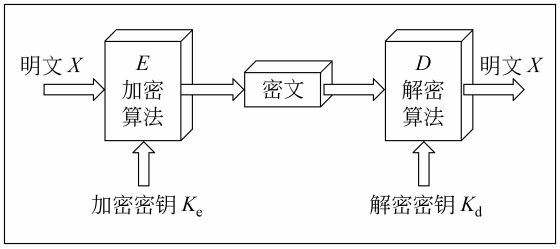

图 3.3　数据加密的模型

界觉察得到的事实。因而，密码分析更需要发挥人的聪明才智，更具有挑战性。

现代密码学是一门迅速发展的应用科学。随着互联网的迅速普及，人们依靠它传送大量的信息，但是这些信息在网络上的传输都是公开的。因此，对于关系到个人利益的信息必须经过加密之后才可以在网上传输，这离不开现代密码技术。

3.1.3　古典密码学

从密码学发展历程来看，可分为古典密码（以字符为基本加密单元的密码）和现代密码（以信息块为基本加密单元的密码）两个阶段。古典密码有着悠久的历史，从古代一直延续到计算机出现以前。古典密码学主要有两大基本方法：

（1）代替密码。将明文的字符替换为密文中的另一种字符，接收者只要对密文做反向替换就可以恢复明文。

（2）置换密码（又称易位密码）。明文的字母保持相同，但顺序被打乱了。

1. 代替密码

代替密码也称替换密码，是使用替换法进行加密所产生的密码。替换密码就是明文中每一个字符被替换成密文中的另一个字符，替换后的各字母保持原来的位置。接收者对密文进行逆替换就恢复了明文。

在代替密码加密体制中使用了密钥字母表。它可以由明文字母表构成，也可以由多个字母表构成。

在古典密码学中，有 4 种类型的代替密码：

（1）单表（简单）代替密码。明文的一个字符用相应的一个密文字符替换。加密过程是从明文字母表到密文字母表的一一映射，例如恺撒（Caesar）密码。

（2）同音（多名码）代替密码。与单表代替密码相似，唯一的不同是明文的一个字符可以映射成密文的几个字符之一，同音代替的密文并不唯一。

（3）多字母组代替密码。字符块被成组加密，例如，ABA 可能对应 RTQ，ABB 可能对应 SLL，等等。Playfair 密码就是这类密码的实例。

（4）多表代替密码。由多个单表代替密码构成，每个密钥加密对应位置的明文，例如维吉尼亚密码。

下面介绍恺撒密码。

恺撒密码又叫循环移位密码。它的加密方法就是把明文中所有字母都用它右边的第 k 个字母替换，并认为 Z 后边又是 A。例如，图 3.4 所示就是循环移动 3 位的恺撒加密法。

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

明文字母	n	o	p	q	r	s	t	u	v	w	x	y	z
密文字母	q	r	s	t	u	v	w	x	y	z	a	b	c

图 3.4　恺撒加密法

这种映射关系表示为如下函数：

$$F(a)=(a+k) \bmod 26$$

其中，a 表示明文字母，k 为密钥。

设 $k=3$，则有如图 3.4 所示的字母替代关系。

对于明文

$$P=\text{computer systems}$$

密文为

$$C=\text{frpsxwhu vbvwhpv}$$

显然，由密文 C 恢复明文非常容易，只要知道密钥 k，就可以构造一张映射表。其加密和解密均可根据此映射表进行。

恺撒密码的优点是密钥简单易记。但它的密文与明文的对应关系过于简单，故安全性很差。

2. 置换密码

置换密码算法的原理是不改变明文字符，而是按照某一规则重新排列消息中的比特或字符顺序，从而实现明文信息的加密。置换密码有时又称为易位密码。

矩阵换位法是实现置换密码的一种常用方法。它将明文中的字母按照给定的顺序安排在一个矩阵中，然后根据密钥提供的顺序重新排列矩阵中的字母，从而形成密文。

其解密过程是：以密钥的字母数作为列数，将密文按照行的顺序写出，再根据由密钥给出的顺序进行置换，产生新的矩阵，从而恢复明文。

下面介绍密钥短语密码。

选择一组有助于记忆的英文字符串，从中筛选无重复的字符，按原顺序记下字符串，作为密钥短语，写在明文字母表下，然后将未出现在字符串的字母按顺序依次写在密钥短语后。

例如选择密钥短语 network security，则

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：n e t w o r k s cui ya b d f g hj l mp q v x z

若明文为 data access，则密文为 wnln nttojj。

3.1.4　现代密码学

密码学者多认为，除了传统上的加解密算法以外，密码协议，即使用密码技术的通信协议，也一样重要，两者为密码学研究的两大课题。

根据密钥类型的不同可将现代密码技术分为两类：对称加密算法(秘密密钥加密)和非对称加密算法(公开密钥加密)。

在对称加密系统中，加密和解密均采用同一个密钥，而且通信双方都必须获得这个密钥，并保持密钥的秘密。

非对称加密系统采用的加密密钥(公钥)和解密密钥(私钥)是不同的。

对称密钥密码学指的是传送方与接收方拥有相同的密钥。

现代的密码学研究主要集中在分组密码与流密码及其应用方面。

1. 分组密码

取明文的一个分组和密钥，输出相同大小的密文分组。由于信息通常比单一分组长，因此可以用多种方式将连续的分组组织在一起。DES 和 AES 是美国政府批准的分组密码标准(AES 将取代 DES)。尽管将被废除，DES 目前依然很流行(Triple-DES 变形仍然相当安全)，应用非常广泛，如自动交易机、电子邮件和远端存取等。

2. 流密码

制造一段任意长的密钥，与明文按位或字符结合，有点类似于一次性密码本。输出的串流根据加密时的内部状态而定。在一些流密码方案中，由密钥控制内部状态的变化。RC4 是相当有名的流密码方案。

3.1.5　加密技术分类

加密技术可以分为以下两类。

1. 对称加密技术

对称密码体制是一种传统密码体制，也称为私钥密码体制。在对称加密系统中，加密和解密采用相同的密钥。因为加密和解密的密钥相同，需要通信双方必须选择和保存他们共同的密钥，并且必须信任对方不会将密钥泄露出去，这样就可以实现数据的机密性和完整性。

对于具有 n 个用户的网络，需要 $n(n-1)/2$ 个密钥。在用户群不是很大的情况下，对称加密系统是有效的。但是对于大型网络，当用户群很大、分布很广时，密钥的分配和保存就成了问题。

对机密信息进行加密和验证是通过随报文一起发送报文摘要(或散列值)来实现的。比较典型的算法有 DES(Data Encryption Standard，数据加密标准)算法及其变体 3DES(三重 DES)、GDES(广义 DES)、IDEA、FEALN 和 RC5 等。

DES 标准由美国国家标准局提出，主要应用于银行业的电子资金转账领域。DES 的密钥长度为 56 位。3DES 使用 3 个独立的 56 位密钥对交换的信息进行 3 次加密。

RC2 和 RC4 是 RSA 数据安全公司的对称加密专利算法，它们采用可变密钥长度的算法。通过规定不同的密钥长度，RC2 和 RC4 能够提高或降低安全的程度。对称加密算法的优点是计算开销小，加密速度快，是目前用于信息加密的主要算法。它的局限性在于通信双方难以确保密钥的安全交换。

此外，一个用户和几个人分别通信，就要维护几个专用密钥。对称加密系统也无法鉴别通信发起方或通信最终方，因为通信双方的密钥相同。另外，对称加密系统仅能用于对数据进行加解密处理，保障数据的机密性，而不能用于数字签名，因而人们迫切需要寻找新的密码体制。

对称加密系统的安全性依赖于以下两个因素：

（1）加密算法必须是足够强的，仅仅基于密文本身去解密信息在实践上是不可能的。

（2）加密方法的安全性依赖于密钥的秘密性，而不是算法的秘密性，因此，没有必要确保算法的秘密性，而需要保证密钥的秘密性。

对称加密系统的优点是：对称加密算法使用起来简单快捷，密钥较短，且破译困难。

对称加密系统的缺点如下：

（1）密钥难以安全传送。

（2）密钥量太大，难以进行管理。

（3）无法满足互不相识的人进行私人谈话时的保密要求。

（4）难以解决数字签名验证的问题。

2. 非对称加密技术

非对称加密技术也称公开密钥技术。该技术需要两个密钥：公钥和私钥。与对称加密技术相比，非对称加密技术最大的特点在于加密和解密使用不同的密钥。非对称加密技术模型如图3.5所示。

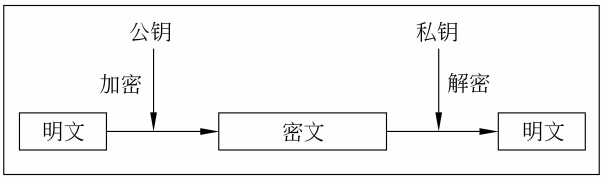

图3.5　非对称密钥技术模型

非对称加密技术的优点是：易于实现，使用灵活，密钥较少。其缺点在于：要取得较好的加密效果和强度，必须使用较长的密钥。

公钥算法大多基于计算复杂度较高的数学难题，通常来自数论。例如，RSA源于整数因子分解问题，DSA源于离散对数问题。近年来快速发展的椭圆曲线密码学则基于与椭圆曲线相关的数学难题，与离散对数问题的难度相当。由于这些问题多涉及模数乘法或指数运算，因此，公开密钥系统通常是复合式的，内含一个高效率的对称密钥算法，用以加密信息，再以公钥加密对称密钥系统所使用的密钥，以提高效率。

在对称加密体系中，加密和解密使用相同的密钥，也许对不同的信息使用不同的密钥，但都面临密钥管理的难题。由于每对通信方都必须使用异于他组的密钥，当网络成员的数量增加时，密钥数量以指数级增加。更尴尬的难题是：当双方没有安全的通道时，如何建立一个共有的密钥以保证安全的通信？如果有通道可以安全地建立密钥，何不使用现有的通道？这个“鸡生蛋、蛋生鸡”的矛盾多年以来在密码学界一直无法解决。

非对称加密技术的特点如下：

（1）密钥分配简单。由于加密密钥与解密密钥不同，且不能由加密密钥推导出解密密

钥，因此，加密密钥表可以像电话号码本一样分发给各用户，而解密密钥则由用户自己掌握。

（2）密钥的数量少。网络中的每个通信成员只需秘密保存自己的解密密钥，n 个通信成员只需产生 n 对密钥，便于密钥管理。

（3）可以满足互不相识的人之间进行私人谈话时的保密性要求。

（4）可以完成数字签名和数字鉴别。发信人使用只有自己知道的私钥进行签名，收信人利用公钥进行检查，既方便又安全。

在实际应用中，非对称加密系统并没有完全取代对称加密系统，因为非对称加密系统计算非常复杂，虽然它的安全性更高，但实现速度却远远赶不上对称加密系统。在实际应用中可利用二者的各自优点，采用对称加密系统加密文件，采用非对称加密系统对加密文件的密钥进行加密，这就是混合加密系统。

非对称加密技术通常被用来加密关键性的、核心的机密数据，而对称加密技术通常被用来加密大量的数据。

3. 两种加密技术的比较

两种加密技术的比较如表 3.1 所示。

表 3.1　两种加密技术的比较

加密技术	代表标准	密钥关系	密钥传递	数字签名	加密速度	主要用途
对称加密	DES	加密密钥与解密密钥相同	不必要	容易	快	数据加密
非对称加密	RSA	加密密钥与解密密钥不同	必要	困难	慢	数字签名、密钥分配加密

3.2　现代加密算法

现代采用的加密算法有 DES、RSA、SHA 等。随着对加密强度要求的不断提高，后来又出现了 AES 和 ECC 等。

3.2.1　加密算法

1. 对称加密算法

在对称加密算法中，只用一个密钥来加密和解密信息，即加密和解密采用相同的密钥。常用的对称加密算法包括以下 3 种：

（1）DES(Data Encryption Standard，数据加密标准)。该算法速度较快，适用于加密大量数据。

（2）3DES。是基于 DES 的变体，对一块数据用 3 个不同的密钥进行 3 次加密，强度更高。

（3）AES(Advanced Encryption Standard，高级加密标准)。是下一代的加密算法标准，速度快，安全级别高。

2000年10月，NIST（美国国家标准和技术协会）宣布了从15个候选算法中选出的一个新的密钥加密标准，由Joan Daemen和Vincent Rijmen设计的Rijndael密钥加密算法被选中，成为新的AES。AES正日益成为加密各种形式的电子数据的实际标准。NIST于2002年5月26日制定了AES规范。

AES算法基于排列和置换运算。排列是对数据的顺序重新进行安排，置换是将一个数据单元替换为另一个。AES使用几种不同的方法来执行排列和置换运算。

AES是一个迭代的、对称密钥分组的密码，它可以使用128、192和256位密钥，并且用128位（16B）分组加密和解密数据。通过分组密码返回的加密数据的位数与输入数据相同。迭代加密使用一个循环结构，在该循环中重复排列和置换输入数据。

AES与3DES的比较如表3.2所示。

表3.2　AES与3DES的比较

算　法	算法类型	密钥长度/位	速度	解密时间/亿年（每秒尝试255个密钥）	资源消耗
AES	对称block密码	128、192、256	高	1 490 000	低
3DES	对称feistel密码	112、168	低	46	中

2. 非对称加密算法

常见的非对称加密算法包括RSA、DSA、ECC和散列算法。

1）RSA

RSA算法由RSA公司发明，是一个支持变长密钥的公钥算法，需要加密的文件块的长度也是可变的。

1976年，由于对称加密算法已经不能满足需要，Diffie和Hellman发表了一篇名为《密码学新动向》的文章，介绍了公钥加密的概念。

RSA算法是1978年由Ron Rivest、Adi Shamir和Leonard Adleman发明的，该算法是以3个发明者姓氏的首字母命名的。RSA是第一个既能用于数据加密也能用于数字签名的算法，但RSA的安全性一直未能得到理论上的证明。

RSA的安全性依赖于大数分解。公钥和私钥是两个大素数（大于100个十进制位）的函数，从密钥和密文推断出明文的难度等同于分解两个大素数的积。

简言之，找两个很大的素数，一个作为公钥公开，另一个作为私钥秘密保存。这两个密钥是互补的，即用公钥加密的密文可以用私钥解密，反过来也可以。

2）DSA

DSA（Digital Signature Algorithm，数字签名算法）是一种数字签名标准。

除了加密外，公钥密码学最显著的成就是实现了数字签名。数字签名是手写签名的数字化，两者的特性都是他人难以仿冒。数字签名可以永久地与被签名的信息结合，无法从信息中移除。

数字签名主要包含两个算法：一个是签名算法，使用私钥处理信息或信息的散列值而产生签名；另一个是验证算法，使用公开密钥验证签名的真实性。RSA和DSA是两种最流行的数字签名机制。数字签名是PKI以及许多网络安全机制的基础。

3）ECC

随着分解大整数方法的进步及完善、计算机速度的提高及计算机网络的发展，为了保障数据的安全，RSA 的密钥长度不断增加。但是，密钥长度的增加导致了其加解密的速度大为降低，硬件实现也变得越来越复杂，这给 RSA 的应用带来了很大的障碍，因此需要一种新的算法来代替 RSA。

1985 年，Koblitz 和 Miller 提出将椭圆曲线用于加密算法，其依据是有限域上的椭圆曲线上的点群中的离散对数问题。这类问题是比因子分解问题更难的问题，其难度是指数级的。这就是 ECC(Elliptic Curves Cryptography，椭圆曲线密码编码学)。

其基本原理为：基于椭圆曲线上的难题——椭圆曲线上离散对数问题，将椭圆曲线中的加法运算与离散对数中的模乘运算相对应，将椭圆曲线中的乘法运算与离散对数中的模幂运算相对应，就可以建立基于椭圆曲线的密码体制。

ECC 在许多方面都有绝对的优势，主要体现在以下几方面：

(1) 抗攻击能力强。

(2) 计算量小，处理速度快。

(3) 占用的存储空间小。ECC 的密钥长度和系统参数与 RSA、DSA 相比要小得多，意味着它所占的存储空间要小得多。这对于加密算法在 IC 卡上的应用具有特别重要的意义。

(4) 带宽要求低。当对长消息进行加解密时，这 3 类密码系统有相同的带宽要求，但应用于短消息时 ECC 的带宽要求却低得多。这使 ECC 在无线网络领域具有广泛的应用前景。

ECC 的这些特点使它必将取代 RSA，成为通用的公钥加密算法。例如，SET 协议的制定者已把 ECC 作为下一代 SET 协议中默认的公钥加密算法。

表 3.3 和图 3.6 是 RSA/DSA 和 ECC 的安全性和速度的比较。

表 3.3　RSA/DSA 和 ECC 的安全性比较

攻破时间(MIPS 年)	RSA/DSA 密钥长度(位)	ECC 密钥长度(位)	密钥长度比	攻破时间(MIPS 年)	RSA/DSA 密钥长度(位)	ECC 密钥长度(位)	密钥长度比
10^4	512	106	5∶1	10^{20}	2048	210	10∶1
10^8	768	132	6∶1	10^{78}	21 000	600	35∶1
10^{11}	1024	160	7∶1				

4) 散列算法

散列算法也叫哈希算法，通过散列算法可以把任意长度的输入(又叫作预映射，pre-image)变换成固定长度的输出，该输出就是散列值。这种变换是一种压缩映射，也就是说，散列值的空间通常远小于输入的空间，不同的输入可能会映射成相同的输出，而不可能从散列值唯一地确定输入值。简单地说，散列函数就是一种将任意长度的消息压缩到某一固定长度的消息摘要的函数。

散列算法主要用于信息安全领域中的信息加密，它把一些不同长度的信息转化成杂乱的 128 位的编码，这些编码就是散列值。也可以说，散列就是找到一种数据内容和数据存放地址之间的映射关系。散列值是信息的提炼，通常其长度要比信息小得多，且为一个固定长度。

加密性强的散列算法一定是不可逆的，这就意味着通过散列值无法推出任何一部分原始信息。任何输入信息的变化，哪怕仅一位，都将导致散列值的明显变化，这称为雪崩效应。

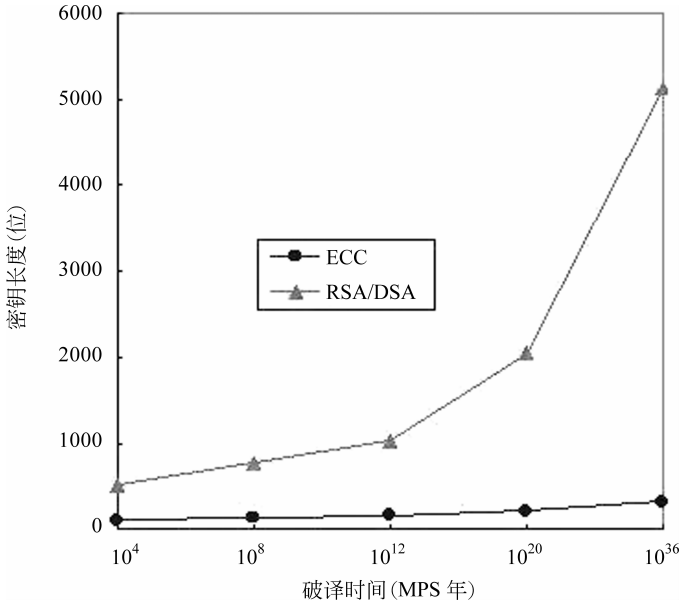

图 3.6　RSA/DSA 和 ECC 的速度比较

散列值还应该是防冲突的，即找不出具有相同散列值的两条信息。具有这些特性的散列值就可以用于验证信息是否被修改。

一般用于产生消息摘要和密钥加密等，常见的单向散列函数有以下两种：

(1) MD5(Message Digest Algorithm 5，信息摘要算法第5版)。是RSA公司开发的一种单向散列算法。

(2) SHA(Secure Hash Algorithm，安全散列算法)。可以对任意长度的数据进行运算，生成一个160位的数值。

1993年，安全散列算法(SHA)由NIST提出，并作为联邦信息处理标准(FIPS PUB 180)公布。1995年，NIST又发布了一个修订版(FIPS PUB 180-1)，通常称之为SHA-1。SHA-1是基于MD4算法的，并且它的设计在很大程度上是模仿MD4的。现在它已成为公认的最安全的散列算法之一，并被广泛使用。

SHA-1算法的原理：接收一段明文，然后以一种不可逆的方式将它转换成一段(通常更小的)密文。也可以简单地理解为：取一串输入码(称为预映射或信息)，并把它们转化为长度较短、位数固定的输出序列，即散列值(也称为信息摘要或信息认证代码)。

单向散列函数的安全性来源于其产生散列值的操作过程具有较强的单向性。如果在输入序列中嵌入密码，那么任何人在不知道密码的情况下都不能产生正确的散列值，从而保证了信息的安全性。SHA将输入流按照每块512b(64B)进行分块，并产生20B的散列值。

该算法输入报文的最大长度不超过264b，产生的输出是一个160b的报文摘要。输入是按512b的分组进行处理的。SHA-1是不可逆的、防冲突的，并具有良好的雪崩效应。

通过散列算法可实现数字签名。数字签名的原理是：将要传输的明文通过散列函数运算转换成报文摘要(不同的明文对应不同的报文摘要)，将报文摘要加密后与明文一起传输给接收方；接收方利用接收的明文产生新的报文摘要，与发送方发来的报文摘要(需要解密)进行比较，如结果一致表示明文未被改动，如果不一致表示明文已被改动。

MAC（信息认证代码）就是一个散列值，其中部分输入信息是密码，只有知道这个密码的接收方才能再次计算和验证 MAC 的合法性。

SHA-1 和 MD5 均由 MD4 导出，因此二者很相似。相应地，它们的加密强度和其他特性也很相似，但两者有以下几点不同：

（1）抗强行攻击的安全性。最显著和最重要的区别是 SHA-1 摘要比 MD5 摘要长32b。使用强行技术，产生任何一个报文使其摘要等于给定报摘要的难度对 MD5 是 2^{128} 数量级的操作，而对 SHA-1 则是 2^{160} 数量级的操作。这样，SHA-1 对强行攻击有更强的抵抗能力。

（2）抗密码分析的安全性。MD5 抗密码分析攻击的能力弱，SHA-1 抗密码分析攻击的能力强。

（3）速度：在相同的硬件上，SHA-1 的运行速度比 MD5 慢。

3.2.2　加密算法的选择与应用

1. 对称加密算法与非对称加密算法的比较

以上综述了对称加密算法与非对称加密算法的原理，总体来说，两者主要有以下几方面的不同：

（1）管理方面。非对称加密算法只需要较少的资源就可以实现目的，在密钥的数量上，两者之间相差很大（非对称加密算法是 n 级别的，对称加密算法是 n^2 级别的）。对称加密算法不适用于广域网，更重要的一点是它不支持数字签名。

（2）安全方面。非对称加密算法基于数学难题，在破解上几乎是不可能的。而对于对称加密算法，发展到 AES，虽然从理论上来看是不可能破解的，但从应用角度来看，非对称加密算法无疑更具有优越性。

（3）速度方面。如果用软件实现，AES 的加解密速度是非对称加密算法的 100 倍；而如果用硬件来实现，这个比值将提高到 1000 倍。

2. 加密算法的选择

前面已经介绍了对称加密算法和非对称加密算法。在实际使用中，应该根据应用需求来选择：

（1）由于非对称加密算法的运行速度比对称加密算法的速度慢很多，当需要加密大量的数据时，建议采用对称加密算法，以提高加密速度。

（2）对称加密算法不能实现签名，因此签名只能使用非对称加密算法。

（3）对称加密算法的密钥管理是一个复杂的过程。密钥的管理直接决定安全性，因此，当数据量很小时，可以考虑采用非对称加密算法。

在实际的操作过程中，通常采用非对称加密算法管理对称加密算法的密钥，然后用对称加密算法加密数据，这样就集成了这两种加密算法的优点，既体现了对称加密算法加密速度快的优点，又体现了非对称加密算法密钥管理安全、方便的优点。

在选定了加密算法后，应该采用多少位的密钥呢？一般来说，密钥越长，运行的速度就越慢，应该根据实际需要的安全级别来选择。RSA 建议采用 1024 位，ECC 建议采用 160 位，AES 采用 128 位即可。

3. 密码学在现代的应用

随着密码学商业应用的普及，公钥密码学受到前所未有的重视。除传统的密码应用系统外，PKI系统以公钥密码技术为主，提供以下功能。

1）保密通信

保密通信是密码学产生的动因。使用公钥密码体制进行保密通信时，信息接收者只有知道对应的密钥才可以解密该信息。

2）数字签名

数字签名技术可以代替传统的手写签名，而且从安全的角度考虑，数字签名具有很好的防伪造功能。它在政府机关、军事领域和商业领域有广泛的应用。

3）秘密共享

秘密共享技术是指将一个秘密信息利用密码技术拆分成 n 个称为共享因子的信息，分发给 n 个成员，只有利用 $k(k\leqslant n)$ 个合法成员的共享因子才可以恢复该秘密信息，其中任何一个或 $m(m\leqslant k)$ 个成员合谋，都无法知道该秘密信息。利用秘密共享技术可以控制任何需要多个人共同控制的秘密信息或命令等。

4）身份认证

在公开的信道上进行敏感信息的传输时，可以采用签名技术对消息的真实性和完整性进行验证，通过验证公钥证书实现对通信主体的身份认证。

5）密钥管理

密钥是保密系统中最为脆弱而重要的环节，公钥密码体制是密钥管理的有力工具。利用公钥密码体制协商和产生密钥，保密通信双方不需要事先共享秘密信息。可以利用公钥密码体制进行密钥分发、保护、密钥托管和密钥恢复等。

基于公钥密码体制除了可以实现以上通用功能以外，还可以实现以下的系统：安全电子商务系统、电子现金系统、电子选举系统、电子招投标系统和电子彩票系统等。

公钥密码体制的产生是密码学由传统的政府和军事等应用领域走向商用、民用的基础，同时互联网和电子商务的发展为密码学的发展开辟了更为广阔的前景。

4. 加密算法的未来

随着计算方法的改进、计算机运行速度的加快和网络的发展，已经有越来越多的算法被破解。

历史上有3次对DES有影响的攻击实验。1997年，有人利用当时各国7万台计算机，历时96天破解了DES的密钥。1998年，电子边境基金会(EFF)用一台花费了25万美元制造的专用计算机历时56小时破解了DES的密钥。1999年，EFF用22小时15分完成了DES密钥破解工作。因此，曾经有过卓越贡献的DES也不能满足日益增长的网络安全需求了。

最近，一组研究人员成功地对一个512位的整数进行了因子分解，宣告了RSA的破解。

数据的安全是相对的，可以说，所有的加密算法都只在一定时期、一定条件下是安全的。随着硬件和网络的发展，目前的常用加密算法都有可能在短时间内被破解，那时就不得不使用更长的密钥或更加先进的算法，才能保证数据的安全。因此，加密算法依然需要不断发展和完善，提供更高的加密强度和运算速度。

纵观这两种加密算法，一个从DES到3DES再到AES，另一个从RSA到ECC，其发展

都是从密钥的简单性、成本的低廉性、管理的简易性、算法的复杂性、保密的安全性及计算的快速性这几个角度去考虑的。因此，未来算法的发展也必定是从这几个角度出发的，而且在实际操作中往往把这两种加密算法结合起来。也许未来集两种加密算法优点于一身的新型算法将会出现，到那个时候，物联网各项应用必将更加快捷和安全。

3.3　对称密码技术

3.3.1　对称密码技术简介

最古老的加密方法已经用了几千年，这种方法被称为对称加密。在这种方法中，同一密钥既用于加密明文，也用于解密密文。

密钥使用的机制非常多样化，但它们共同的弱点是：因为需要共享密钥，所以如果密钥落入坏人之手将很危险。一旦未经授权的人得知了密钥，就会危及基于该密钥的安全系统。如果只涉及一条消息，可能不要紧；但是，同一个密钥很可能被重复使用，而通信双方未必知道密钥已不再是保密的。

这种简单方案的变体涉及使用一个任意排序的字母表，它和用于明文消息的字母表有同样的长度。在这种情况下，密钥可能是由数字组成的一个长序列，例如，5，19，1，2，11，…，表明 A 应该映射为 E，B 为 S，C 为 A，D 为 B，E 为 K……当然，这样的系统是极其脆弱的，而现代的系统则使用基于难解的数学问题的复杂算法，因而使系统极其强壮。

对于一个查看用对称密码加密的数据的人来说，如果对用于加密数据的密钥根本没有访问权，那么他完全不可能查看加密数据。如果这样的密钥落入坏人之手，那么就会马上彻底地危及使用该密钥加密的数据的安全性。因此，使用密钥方法的这个组中的所有系统所共享的内容是密钥管理的难点。

1. 密钥长度

通常提到的密钥都有特定的长度，如 56 位或 128 位，这些长度都是指对称密钥密码的长度，而非对称密钥中至少私钥是相当长的。而且，这两组密钥长度之间没有任何相关性，除非偶尔在使用某一给定系统的情况下，达到某一给定密钥长度提供的安全性级别。

在任何特定组中，所用密钥的长度通常是确定安全性的一个重要因素。而且，密钥空间并不是随着密钥长度线性增长的，而是密钥每增加一位，密钥空间就加倍。Giga Group 对此作了一个简单的比喻：如果一个茶匙足够容纳所有可能的 40 位的密钥组合，那么所有 56 位的密钥组合需要一个游泳池来容纳，而所有可能的 128 位的密钥组合的体积与地球的体积相当。一个用十进制表示的 128 位的值大概有 3.40×10^{38} 个。

2. 加密速度

对称密钥方法比非对称密钥方法快得多，因此加密大量文本时，对称密钥方法是首选机制。密钥密码最适合用于在单用户或小型组的环境中保护数据，通常都是通过使用密码实现的。实际上，正如在前面已提到的，广为散布或大规模实际使用的最令人满意的方法往往都同时组合了对称加密和非对称加密系统。

3. 对称密钥密码的类型

现在，通常使用分组密码（block cipher）或流密码（stream cipher）实现对称密码。下面讨论这两种密码。

1）分组密码

分组密码根据“电码本密码”获得，其特点如下：

（1）分组密码的密钥决定电码本。

（2）每个密钥生成一个不同的电码本。

（3）混淆和扩散都得到了利用。

2）流密码

流密码根据一次一密获得，其特点如下：

（1）密钥较短。

（2）密钥被扩展为更长的密钥流（keystream）。

（3）密钥流被用作一次一密的密钥。

（4）只用到了混淆。

3.3.2 分组密码

分组密码将定长的明文块（称为分组）转换成等长的密文，这一过程在密钥的控制之下。使用逆向变换和同一密钥来实现解密。对于当前的许多分组密码，分组大小是64b，但这很可能会增加。分组密码的基本模型如图3.7所示。

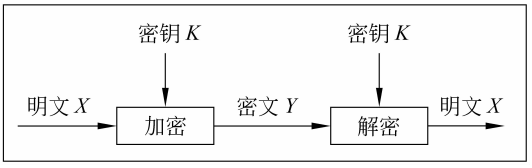

图3.7 分组密码的基本模型

明文消息通常要比特定的分组大得多，而且使用不同的技术或操作方式对分组进行加密。这样的方式示例有电码本（ECB）、密码分组链接（CBC）或输出反馈（CFB）。

（1）ECB使用同一个密钥简单地将每个明文块一个接一个地进行加密。

（2）在CBC方式中，每个明文块在加密前先与前一密文块进行异或运算，从而提高了复杂程度，可以使某些攻击更难以实施。

（3）OFB方式类似于CBC方式，但是它进行异或的量是独立生成的。

CBC得到广泛使用，例如在DES（qv）实现中。在有关密码技术的图书中深入讨论了各种方式。请注意：用户自己建立的密码系统的普遍弱点就是以简单的形式使用某些公开的算法，而不是以提供了额外保护的特定方式使用。

迭代的分组密码在加密过程中有多次循环，因此提高了安全性。在每次循环中，可以使用特殊的函数根据初始密钥派生出的子密钥进行适当的变换。该附加的计算需求必然会影响加密的速度，因此要在安全性需要和执行速度之间进行平衡。

分组密码方案包括DES、IDEA、SAFER、Blowfish和Skipjack，最后一个方案是美国国

家安全局限制器芯片中使用的算法。

3.3.3　流密码

与分组密码相比，流密码可以非常快速有效地运作。流密码作用于由若干位组成的一些小型组，通常使用称为密钥流的一个位序列作为密钥对它们逐位应用异或运算。有些流密码基于一种称作线形反馈移位寄存器(Linear Feedback Shift Register，LFSR)的机制，该机制生成一个二进制位序列。

流密码是由一种专业的密码——Vernam 密码(也称为一次性密码本)发展而来的。流密码的示例包括 RC4 和软件优化加密算法(Software optimized Encryption ALgorithm，SEAL)及 Vernam 密码的特殊情形。

3.3.4　对称密码的算法

1. 数据加密标准

数据加密标准(DES)源自 IBM 公司的研究工作，并在 1997 年被美国政府正式采纳为加密标准。它是使用最广泛的密钥系统，特别是在保护金融数据安全方面。最初开发的 DES 是嵌入硬件中的。通常，自动取款机(Automated Teller Machine，ATM)都使用 DES。

DES 使用一个 56 位的密钥以及附加的 8 位奇偶校验位，产生最大 64 位的分组。这是一个迭代的分组密码，使用称为 Feistel 的技术。DES 将加密的文本块分成两半，使用子密钥对其中一半应用循环功能，然后将输出结果与另一半进行异或运算，接着交换这两半。这一过程会继续下去，但最后一个循环不交换。DES 执行 16 次循环。

攻击 DES 的主要形式被称为蛮力破解或彻底密钥搜索，即重复尝试各种密钥，直到有一个正确为止。如果 DES 使用 56 位的密钥，则可能的密钥数量是 2^{56} 个。随着计算机系统能力的不断发展，DES 的安全性会逐渐减弱，然而，对于非关键性质的实际应用来说，仍可以认为它是足够安全的。DES 现在仅用于旧系统的鉴定，而当前的应用系统更多地选择新的加密标准——高级加密标准(Advanced Encryption Standard，AES)。

DES 的常见变体是 3DES，它使用 168 位的密钥对数据进行 3 次加密。它通常(但并非始终)具有极其强大的安全性。如果 3 个 56 位的子元素都相同，则 3DES 向后兼容 DES。

IBM 公司最初对 DES 拥有专利权，但是在 1983 年已到期。DES 目前处于公有领域，允许在特定条件下免除专利使用费而使用。

2. 国际数据加密算法

国际数据加密算法(International Data Encryption Algorithm，IDEA)是由苏黎世理工学院的两位研究员 Xuejia Lai 和 James L. Massey 开发的，由一家瑞士公司 Ascom Systec 拥有专利权。IDEA 是作为迭代的分组密码实现的，使用 128 位的密钥和 8 次循环。这比 DES 提供了更高的安全性，但是在选择用于 IDEA 的密钥时，应该排除那些被称为“弱密钥”的密钥。DES 只有 4 个弱密钥和 12 个次弱密钥，而 IDEA 中的弱密钥数相当可观，有 2^{51} 个。但是，如果密钥的总数非常大，达到 2^{128} 个，那么仍有大量密钥可供选择。

通过支付专利使用费(通常大约是每个副本6美元),可以在世界很多地区使用IDEA。这种费用在某些区域适用,而其他区域并不适用。IDEA被认为是极为安全的。使用128位的密钥,蛮力攻击需要进行的测试次数与DES相比会明显增大,甚至允许对弱密钥进行测试。而且,它尤其能抵抗专业形式的分析性攻击。

3. CAST

CAST是以它的设计者Carlisle Adams和Stafford Tavares命名的。它是一个64位的Feistel密码,使用16次循环并允许密钥最长可达128位。其变体CAST-256使用128位的分组大小,而且允许使用最长256位的密钥。

虽然CAST非常快,但是它的主要优势是安全性,而不是速度。在PGP的最新版本及IBM、Microsoft等厂商的产品中都使用了它。

Entrust Technologies公司拥有CAST的专利权。

4. 一次性密码本

一次性密码本(或Vernam密码)具有很高的安全性,所以在某些特殊情况中(通常是在战争中)有很高的应用价值。它使用与消息一样长的随机生成的密钥。通常使用位的异或运算,将其应用于明文,以产生加密文本。应用同一密钥和适当的算法,可以方便地解密消息。

一次性密码本加密和解密的简单例子如下:

00101100010…11011100101011 (原始明文消息)
01110111010…10001011101011 (与消息长度相等的随机生成的密钥)
01011011000…01010111000000 (加密后的消息)
01110111010…10001011101011 (重用于解密的密钥)
00101100010…11011100101011 (恢复的原始消息)

虽然一次性密码本是绝对安全的,但是它常常是不太实用的,因为需要以某种安全的方法将与消息长度相等的密钥传送给接收方用于解密。而且,密钥只使用一次,然后就被丢弃,虽然这明显对保证安全性有利,但加大了密钥管理的困难。目前使用一次性密码本的一个领域是MAC。

5. 高级加密标准

DES即将到了它的使用寿命尽头,预计高级加密标准(AES)会代替DES作为新的安全标准。1997年,美国国家标准和技术协会(National Institute of Standards and Technology,NIST)组织了一项竞赛,最终的获胜者是比利时的Joan Daemen和Vincent Rijmen提交的一个名为Rijndael的产品(当前正在处于大规模试验和评估中)。

从技术上讲,Rijndael结构复杂,而且有点不同寻常,却似乎非常安全且通用,因为它的执行速度很快,十分适合现代需求(如智能卡),而且能够使用的密钥大小范围很广。

3.4 非对称密码技术

非对称密码系统的解密密钥与加密密钥是不同的,一个称为公钥,另一个称为私钥,因此这种密码体系也称为公钥密码体系。公钥密码除可用于加密外,还可用于数字签名。

3.4.1　公钥密码算法概述

公钥密码系统体制采用了一对密钥——公钥和私钥，而且很难从公钥推导出私钥。公钥密码系统主要使用 RSA 公钥密码算法。

1. 公钥的起源

公钥密码体制于 1976 年由 W. Diffie 和 M. Hellman 提出，同时，R. Merkle 也独立提出了这一体制。这种密码体制采用了一对密钥——加密密钥和解密密钥（而且从解密密钥推出加密密钥是不可行的）。在这一对密钥中，一个可以公开（称为公钥），另一个为用户专用（称为私钥）。

公钥密码体制的产生主要有两个原因，一是常规密钥密码体制存在密钥分配问题，二是对数字签名的需求。

公钥密码体制算法的特点是：使用一个加密算法 E 和一个解密算法 D，它们彼此完全不同。对于已选定的 E 和 D，即使已知 E 的完整描述，也不可能推导出 D。

公钥密码体制如图 3.8 所示。

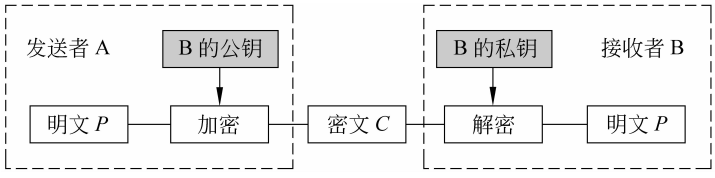

图 3.8　公钥密码体制

2. 单向陷门函数

公钥密码系统是基于单向陷门函数的概念提出的。

单向函数是易于计算但求逆困难的函数，而单向陷门函数是在不知道陷门信息时求逆困难，而在知道陷门信息时易于求逆的函数。

单向陷门函数是有一个陷门的特殊单向函数。它首先是一个单向函数，在一个方向上易于计算，而反方向却难以计算。但是，如果知道陷门，则也能很容易在另一个方向计算这个函数。即，已知 x，易于计算 $f(x)$；而已知 $f(x)$，却难以计算 x。然而，一旦给出 $f(x)$和一些秘密信息（即陷门）y，就很容易计算 x。在公钥密码系统中，计算 $f(x)$相当于加密，陷门 y 相当于私钥，而利用陷门 y 求 $f(x)$中的 x 则相当于解密。

在现实世界中，这样的例子是很普遍的。例如，将挤出的牙膏弄回管子里要比把牙膏挤出来困难得多；燃烧一张纸要比使它从灰烬中再生容易得多；把盘子打碎成数千个碎片很容易，把所有这些碎片再拼成一个完整的盘子则很难。

类似地，将许多大素数相乘要比对其乘积分解因式容易得多。数学上有很多函数具有单向函数的特点，人们能够有效地计算它们，但至今未找到有效的求逆算法。一般把离散对数函数和 RSA 函数作为单向函数来使用，但是，目前还没有严格的数学证明表明这些单向函数真正难以求逆，即单向函数是否存在还是未知的。

在密码学中最常用的单向函数有两类：一是公钥密码中使用的单向陷门函数，二是消息摘要中使用的单向散列函数。

单向函数不能用于加密。因为用单向函数加密的信息是无法解密的。但是，可以利用具有陷门信息的单向函数构造公钥密码。

3. 公钥密码系统的应用

公钥密码系统可用于以下3个方面。

1）通信保密

在通信保密中，将公钥作为加密密钥，将私钥作为解密密钥，通信双方不需要交换密钥就可以实现保密通信，如图3.9所示。

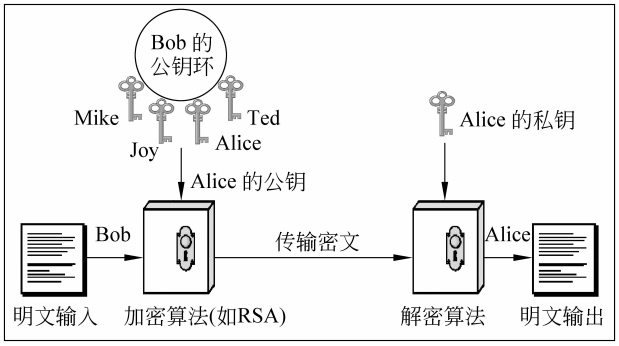

图3.9　公钥密码系统应用于通信保密

2）数字签名

将私钥作为加密密钥，将公钥作为解密密钥，可实现由一个用户对数据进行加密，而多个用户可以解读数据，如图3.10所示。

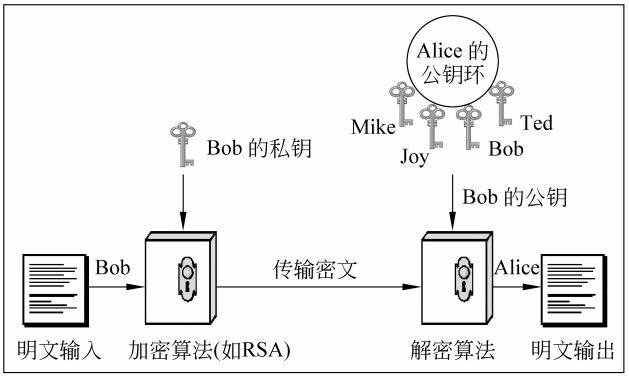

图3.10　公钥密码系统应用于数字签名

3）密钥交换

通信双方交换会话密钥，以加密通信双方后续传输的信息。每次逻辑连接使用一个新的会话密钥，用完就丢弃。

4. 公开密钥算法的特点

公开密钥算法有如下特点：

(1) 发送者用加密密钥 PK 对明文 X 加密后，接收者用解密密钥 SK 解密，即可恢复出明文，或写为

$$D_{SK}(E_{PK}(X))=X$$

解密密钥是接收者专用的私钥，对其他人都保密。

此外，加密和解密的运算可以对调，即

$$E_{PK}(D_{SK}(X))=X$$

(2) 加密密钥是公开的，但不能用它来解密，即

$$D_{PK}(E_{PK}(X))\neq X$$

(3) 在计算机上可以很容易地产生成对的 PK 和 SK。

(4) 从已知的 PK 实际上不可能推导出 SK，即从 PK 得到 SK 是在计算上不可行的。

(5) 加密和解密算法都是公开的。

3.4.2　RSA 算法

RSA 是一种基于公钥密码体制的优秀加密算法。

RSA 算法是一种分组密码算法，它的保密强度取决于具有大素数因子的合数的因子分解的难度，如表 3.4 所示。

表 3.4　具有大素数因子的合数因子分解的难度

整数的十进制位数	因子分解的运算次数	所需计算时间（每微秒一次）	整数的十进制位数	因子分解的运算次数	所需计算时间（每微秒一次）
50	1.4×10^{10}	3.9 小时	200	1.2×10^{23}	3.8×10^{9} 年
75	9.0×10^{12}	104 天	300	1.5×10^{29}	4.0×10^{15} 年
100	2.3×10^{15}	74 年	500	1.3×10^{39}	4.2×10^{25} 年

例如，整数的十进制位数达到 100 位时，进行因子分解的运算次数为 2.3×10^{15}，平均每微秒进行一次计算，求解所需要的时间为 74 年。而 74 年后几乎现在所有的资料都已经不具备保密的价值了。

公钥和私钥是一对大素数的函数，从一个公钥和密文中恢复出明文的难度等价于分解两个大素数之积。求一对大素数的乘积很容易，但要对这个乘积进行因子分解则非常困难，如图 3.11 所示。因此，可以把一对大素数的乘积公开作为公钥，而把素数作为私钥。

公钥密码系统一般都涉及数论的知识，如素数、欧拉函数和中国剩余定理等。

1. RSA 加密算法

若用整数 X 表示明文，用整数 Y 表示密文（X 和 Y 均小于 n），则加密运算为

$$Y = X \bmod n$$

解密运算为

$$X = Y \bmod n$$

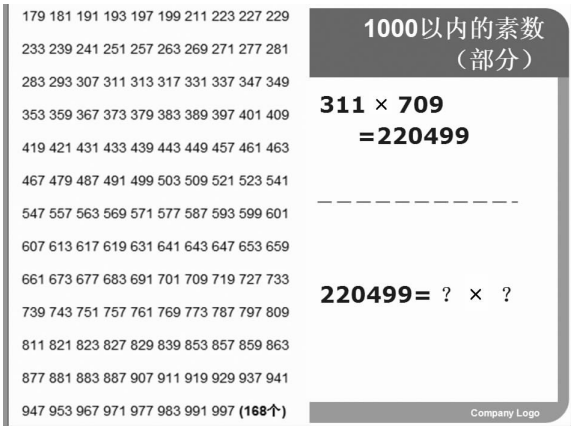

图 3.11　公开密钥算法 RSA-1

2. RSA 密钥的产生

现在讨论 RSA 公钥密码体制中每个参数是如何选择和计算的。

(1) 计算 n。

用户秘密地选择两个大素数 p 和 q，计算出 $n=pq$。n 称为 RSA 算法的模数。

(2) 计算 $\phi(n)$。

用户再计算出 n 的欧拉函数 $\phi(n)=(p-1)(q-1)$。

(3) 选择 e 作为加密指数。

用户从[1,$\phi(n)-1$]中选择一个与 $\phi(n)$互素的数 e 作为公开的加密指数。

(4) 计算 d 作为解密指数。

用户计算出满足下式的 d：

$$ed \equiv 1 \bmod \phi(n)$$

即

$$(ed-1) \bmod \phi(n)=0$$

由此推出

$$ed = t\phi(n)+1$$

其中，t 是大于或等于 1 的正整数。

(5) 得出所需的公钥和私钥：

$$\mathrm{PK}=\{e,n\}$$

$$\mathrm{SK}=\{d,n\}$$

其中，p、q、$\phi(n)$和 d 就是秘密的陷门(这 4 项并不是相互独立的)，这些信息不可以泄露。

3. RSA 加密消息

RSA 加密消息 m 时(这里假设 m 是以十进制表示的)，首先将消息分成大小合适的分组，然后对分组分别进行加密。每个分组的大小应该比 n 小。

设 c_i 为明文分组 m_i 加密后的密文，则加密公式为

$$c_i = m_i \bmod n$$