CHAPTER 1

第 1 章 密码技术及实验

1.1 密码技术

密码学是一门古老的科学。它的起源可以追溯到 4000 多年前的古埃及、巴比伦、古罗马和古希腊，大概自人类社会出现战争时起便出现了密码。在 1949 年之前，密码技术更多地只能称为艺术而不是科学，密码的设计和分析是凭直觉和经验来进行的，而不是靠严格的理论证明。而随着电子计算机的诞生以及香农 (Shannon) 发表了《保密系统的通信理论》一文，密码学的研究才真正进入现代科学研究的范畴。

密码学又是一门年轻的科学。随着科学技术的进步，密码学的研究也日新月异。首先，密码学越来越依赖于数学知识，现代密码学离开数学几乎是不可想象的；其次，密码学还与别的学科相互渗透，如量子力学、光学、混沌学、生物学等，并且互相促进。

1.1.1 密码学基本概念

自古以来，密码主要应用于军事、政治、外交等机要部门，因而密码学的研究工作本身也是秘密进行的。然而随着计算机科学、通信技术、微电子技术的发展，计算机网络的应用进入了人们的日常生活和工作中，从而产生了保护隐私、敏感甚至秘密信息的需求，而且这样的需求在不断扩大，于是密码学的应用和研究逐渐公开化，并呈现出了空前的繁荣。

研究密码编制的科学称为密码编制学（Cryptography），研究密码破译的科学称为密码分析学（Cryptanalysis），它们共同组成了密码学（Cryptology）。

密码技术的基本思想就是伪装信息，即对信息做一定的数学变换，使不知道密钥的用户不能解读其真实的含义。变换之前的原始数据称为明文（Plaintext），变换之后的数据称为密文（Ciphertext），变换的过程就叫做加密（Encryption），而通过逆变换得到原始数据的过程就称为解密（Decryption），解密需要的条件或者信息称为密钥（Key），通常情况下密钥就是一系列字符串。

一个密码系统主要由以下五部分构成：

(1) 明文空间 M—— 所有明文的集合；

(2) 密文空间 C—— 全体密文的集合；

(3) 密钥空间 K—— 全体密钥的集合，其中每一个密钥 k 均由加密密钥 K_{e} 和解密密钥 K_{d} 组成，即 $K=(K_{\mathrm{e}},K_{\mathrm{d}})$，在某些情况下 $K_{\mathrm{e}}=K_{\mathrm{d}}$；

(4) 加密算法 E—— 一组以 K_{e} 为参数的由 M 到 C 的变换，即 $C=E(K_{\mathrm{e}},M)$，可简写为 $C=E_{K_{\mathrm{e}}}(M)$；

(5) 解密算法 D—— 一组以 K_{d} 为参数的由 C 到 M 的变换，可表示为 $M = D(K_{\mathrm{d}}, C)$ 或 $M = D_{K_{\mathrm{d}}}(C)$。

密码系统模型如图 1.1 所示。

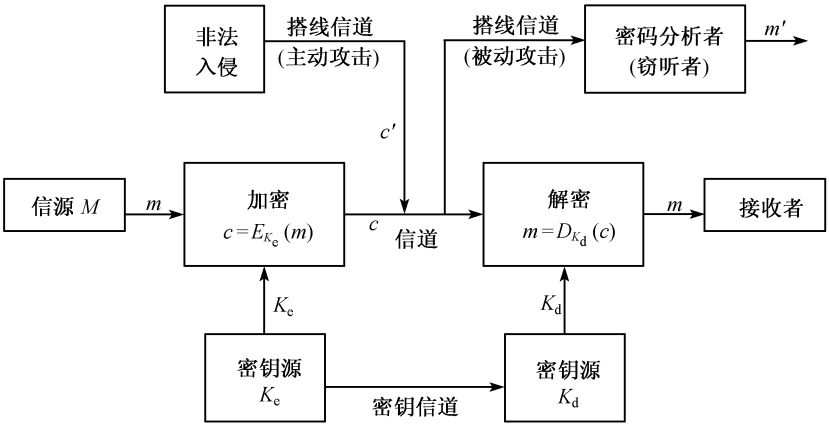

图 1.1　密码系统模型

从图 1.1 中可以了解密码系统工作的大体流程以及可能存在的被攻击的情形：信息的发送者通过一个加密算法将消息明文 m 加密为密文 c，然后通过不安全的信道传送给接收者，接收者接到密文 c 后用已知的密钥 K 来进行解密得到明文 m。而在信息的传输过程中，可能会有主动攻击者冒充发送者传送 c' 给接收者，干扰或者破坏通信；也可能会有被动攻击者盗取密文 c，那么密码分析者的工作就是在不知道 K 的情况下通过 c 来恢复出 m。以上两种攻击行为在现实生活中非常常见，因此对作为信息安全关键技术的密码学的研究就显得尤为重要和迫切。

1.1.2　信息论和密码学

现代信息论是由香农于 1948 年首先确立的，他在论文《通信的数学理论》中详细阐述了如何用信息论的观点处理存在随机干扰的通信系统中的信息传输问题。

1949 年香农发表了题为《保密系统的通信理论》的著名论文，从信息论的角度对信息源、密钥、加密和密码分析进行了数学分析，用不确定性和唯一解距离来度量密码体制的安全性，阐述了密码体制、完善保密性、纯密码、理论保密和实际保密等重要概念，把密码置于坚实的数学基础之上，标志着密码学作为一门独立学科的成立。从此，信息论成为密码学的重要理论基础之一。关于这部分内容的详细讨论请参考相关资料。

1.1.3　密码编制学

密码编制学是对消息进行编码以隐藏明文消息的一门学问。

从现代密码学的观点来看，许多古典密码都是不安全的，或者说是很容易被破译的。替代和置换是古典密码中常用的变换形式。

1. 替代密码

首先需要构造一两个或者多个密文字母表，然后用密文字母表中的字母或字母组来替

代明文字母或字母组，各个字母或字母组的相对位置不变，但其本身改变了。下面来看一下罗马皇帝 Julius Caesar 在公元前 50 年左右所使用的 “恺撒密码”，这其实就是一种典型的替代密码。他将字母按字母表中的顺序循环排列，将明文中的每个字母用其后面的第三个字母代替以得到对应的密文。

以英文为例，恺撒密码所使用的明文字母表和密文字母表分别为：

明文字母表：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
密文字母表：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

那么，对于明文 attack postoffice，经恺撒密码变换后得到的密文为：

dwwdfn srvwriilfh

恺撒密码可以说是替代密码的最简单的例子。在替代密码中，密文中的字母顺序与明文中的字母顺序一致，只不过各密文字母是由相应的明文字母按某种映射变换得到的。

按照映射规则的不同，替代密码可分为 3 种：单表替代密码、多表替代密码和多字母替代密码。在此不再详述，有兴趣的读者可查看相关资料。

2. 置换密码

将明文中的字母重新排列，字母表示不变，但其位置改变了，这样编成的密码就称为置换密码。换句话说，明文与密文所使用的字母相同，但是它们的排列顺序不同。最简单的置换密码就是把明文中的字母顺序颠倒一下。

可以将明文按矩阵的方式逐行写出，然后再按列读出，并将它们排成一排作为密文，列的阶就是该算法的密钥。在实际应用中，人们常常用某一单词作为密钥，按照单词中各字母在字母表中的出现顺序排序，用这个数字序列作为列的阶。

【例 1-1】 若以 coat 作为密钥，则它们的出现顺序为 2、3、1、4，对明文 attack postoffice 加密的过程如图 1.2 所示。

密钥	c o a t
阶	2 3 1 4
	a t t a c k p o s t o f f i c e

图 1.2　对明文 attack postoffice 加密的过程

按照阶数由小到大逐列读出各字母，所得密文为：

t p o c a c s f t k t i a o f e

对于这种列变换类型的置换密码，密码分析很容易进行：将密文逐行排列在矩阵中，并依次改变行的位置，然后按列读出，就可得到有意义的明文。为了提高它的安全性，可以按同样的方法执行多次置换。例如对上述密文再执行一次置换，就可得到原明文的二次置换密文：

o s t f t a t a p c k o c f i e

还有一种置换密码采用周期性换位。对于周期为 r 的置换密码，首先将明文分成若干组，每组含有 r 个元素，然后对每一组都按前述算法执行一次置换，最后得到密文。

【例 1-2】 一个周期为 4 的换位密码，密钥及密文同例 1-1，加密过程如图 1.3 所示。

密钥	c o a t	c o a t	c o a t	c o a t
阶	2 3 1 4	2 3 1 4	2 3 1 4	2 3 1 4
明文	a t t a	c k p o	s t o f	f i c e
密文	t a t a	p c k o	o s t f	c f i e

图 1.3 周期性换位密码

相比之下，现代密码算法的编制需要考虑的因素要比古典密码多得多，在设计方案上也要复杂得多。以最常见的数据加密标准 DES 为例，它就综合运用了置换、替代、代数等多种密码技术，堪称近代密码的一个典范。关于 DES 的更多详细内容，请参看《现代密码技术》一书。

1.1.4 密码分析学

密码分析学就是研究密码破译的科学。如果能够根据密文系统确定出明文或密钥，或者能够根据明文密文对系统确定出密钥，则称这个密码系统是可破译的。常用的密码分析方法主要有 3 种。

(1) 穷举攻击：对截获的密文，密码分析者试遍所有的密钥，以期得到有意义的明文；或者使用同一密钥，对所有可能的明文加密直到得到的密文与截获的密文一致。穷举攻击也称强力攻击或完全试凑攻击。

(2) 统计分析攻击：密码分析者通过分析明文与密文的统计规律，得到它们之间的对应关系。

(3) 数学分析攻击：密码分析者根据加密算法的数学依据，利用数学方法 (如线性分析、差分分析及其他一些数学知识) 来破译密码。

根据密码分析者可利用的数据，可将常见的密码分析攻击分为 4 类，由弱到强分别是唯密文攻击、已知明文攻击、选择明文攻击和选择密文攻击。

(1) 唯密文攻击：密码分析者有一些用同一密钥加密的密文，他们试图恢复出尽可能多的明文，或者推算出加密密钥以解出更多的密文。

(2) 已知明文攻击：密码分析者不仅得到了一些明文，而且也知道相应的密文，他们的任务是据此推出加密密钥或算法，而该算法可以对用同一密钥加密的任何密文进行解密。

(3) 选择明文攻击：密码分析者不仅可得到一些消息的密文和相应的明文，而且也可选择被加密的明文。通过选择特定的明文进行加密，有可能产生更多的关于密钥的消息，这比已知明文攻击更有效。如果分析者不仅能选择被加密的明文，还能基于以前的结果修正这个选择，那么就是自适应选择明文攻击。

(4) 选择密文攻击：密码分析者可选择不同的密文，并可得到对应的明文。这种攻击主要用于公钥算法。

一个密码系统，如果无论密码分析者截获多少密文和用什么技术方法进行攻击都不能被攻破，则称为绝对不可破译的。绝对不可破译的密码在理论上是存在的，这就是著名的 “一次一密” 密码。但是，由于密钥管理上的困难，“一次一密” 密码是不实用的。从理论上来说，如果能够拥有足够多的资源，那么任何实际使用的密码都是可以破译的。

1.1.5　小结

密码学是一门专业性很强的学科，现代密码学更是涉及数学、计算机、信息资讯等多个学科，其复杂性不言而喻。本章接下来的内容主要就是为了通过一些基本密码算法实验让学生对密码技术有更深刻的理解和认识。

本章接下来的内容一共安排了 12 个实验，分别是素数生成实验、恺撒密码算法、线性反馈移位寄存器、DES 算法实验、MD5 算法实验、RSA 算法实验、SHA-1 算法实验、AES 算法实验、DSA 数字签名实验、ECC 算法实验以及密码算法分析设计实验和密码技术应用实验。

通过这些实验，可以了解分组密码的主要流程，掌握分组密码的设计原则、散列函数的应用以及公开密钥算法的原理和应用，从而对密码学知识有更深入的理解。

1.2　素数生成实验

只能被 1 和它本身除尽的整数称为素数，不是 1 且非素数的整数称为合数。素数是无限的，在密码学中，经常要用到大素数，判定一个整数是否为素数就是一项关键技术。实际上，由于一个合数总是可以分解成若干个素数的乘积，因此如果把素数（最初只知道 2 是素数）的倍数都去掉，那么剩下的就是素数了。这一方法称为 Eratosthenes 筛法，是一种寻找素数的确定性方法。

判断某一个整数是否为素数的方法有很多，Rabin-Miller 算法就是其中较为简单有效的一种。Rabin-Miller 素性检验的过程如下：

首先随机选择一个待测奇整数 $n \geqslant 3$，计算 s 和 t 使得 $n-1=2^s t$，其中 t 为奇数。

(1) 随机选取一个整数 b，使得 $2 \leqslant b \leqslant n-2$;

(2) 计算 $r_0 \equiv b^t (\mathrm{mod}\ n)$;

(3) 如果 $r_0 = 1$ 或者 $r_0 = n-1$，则通过检验，n 可能为素数，回到步骤 (1) 继续选取另一个随机整数 b 做检验，如果 $r_0 \neq 1$ 或者 $r_0 \neq n-1$，则计算 $r_1 \equiv {r_0}^2 (\mathrm{mod}\ n)$;

(4) 如果 $r_1 = n-1$，则通过检验，n 可能为素数，回到步骤 (1) 继续选取另一个随机整数 b 做检验，如果 $r_1 \neq n-1$，则计算 $r_2 \equiv {r_1}^2 (\mathrm{mod}\ n)$;

(5) 依次类推, 直到如果 $r_{s-1} = n-1$，则通过检验，n 可能为素数，回到步骤 (1) 继续选取另一个随机整数 b 做检验，如果 $r_{s-1} \neq n-1$，则 n 为合数。

1.2.1　Eratosthenes 筛法实验

【实验目的】

掌握 Eratosthenes 筛法的基本步骤。

【实验内容】

编写 VC++ 程序，运用 Eratosthenes 筛法寻找所有 10000 以内的素数。

【实验结果】

10000 以内的素数共有 1229 个。

1.2.2 Rabin-Miller 素性检验实验

【实验目的】

(1) 掌握 Rabin-Miller 素性检验的原理和步骤。

(2) 复习模重复平方计算法。

【实验内容】

以 2、3、5、7、11、13 作为基（即 b）对下列整数做素性检验：

131、133、141、163、181、197、203

【实验结果】

素数：131、163、181、197。

合数：133、141、203。

1.3 恺撒密码算法

前面已经提到过恺撒密码是一种典型的替代密码，它是将字母按字母表中的顺序循环排列，将明文中的每个字母用其后面的第三个字母代替以得到对应的密文。

以英文为例，恺撒密码所使用的明文字母表和密文字母表分别为：

明文字母表：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

密文字母表：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

那么，对于明文 attack postoffice，经恺撒密码变换后得到的密文为：

dwwdfn srvwriilfh

恺撒密码可以说是替代密码的最简单的例子。如果对字母 a∼z 做一个 0∼25 的映射，那么对于明文 m 和密文 c 就会有以下关系：

$$c \equiv m + 3 (\mathrm{mod}\ 26)$$

【实验目的】

(1) 掌握恺撒密码的使用和破译。

(2) 对替代密码有更深入的理解。

【实验内容】

编写 VC++ 程序，实现恺撒密码加密和解密过程（对于非字母符号不做处理），然后对下列明文进行加密，对下列密文进行解密。

明文：when, in the course of human events, it becomes necessary for one people to dissolve the political bands which have connected them with another, and to assume among the powers of the earth the separate and equal station to which the laws of nature and of nature's god entitle them, a decent respect to the opinions of mankind requires that they should declare the causes which impel them to the separation.

密文：zh krog wkhvh wuxwkv wr eh vhoi-hylghqw; wkdw doo phq duh fuhdwhg htxdo, wkdw wkhb duh hqgrzhg eb wkhlu fuhdwru zlwk fhuwdlq xqdolhqdeoh uljkwv; wkdw dprqj wkhvh duh olih, olehuwb, dqg wkh sxuvxlw ri kdsslqhvv.

【实验结果】

密文：zkhq, lq wkh frxuvh ri kxpdq hyhqwv, lw ehfrphv qhfhvvdub iru rqh shrsoh wr glvvroyh wkh srolwlfdo edqgv zklfk kdyh frqqhfwhg wkhp zlwk dqrwkhu, dqg wr dvvxph dprqj wkh srzhuv ri wkh hduwk wkh vhsdudwh dqg htxdo vwdwlrq wr zklfk wkh odzv ri qdwxuh dqg ri qdwxuh'v jrg hqwlwoh wkhp, d ghfhqw uhvshfw wr wkh rslqlrqv ri pdqnlqg uhtxluhv wkdw wkhb vkrxog ghfoduh wkh fdxvhv zklfk lpsho wkhp wr wkh vhsdudwlrq.

明文：we hold these truths to be self-evident; that all men are created equal, that they are endowed by their creator with certain unalienable rights; that among these are life, liberty, and the pursuit of happiness.

1.4　线性反馈移位寄存器

移位寄存器在一个流密码系统中是产生密钥序列的主要部分，如图 1.4 所示。

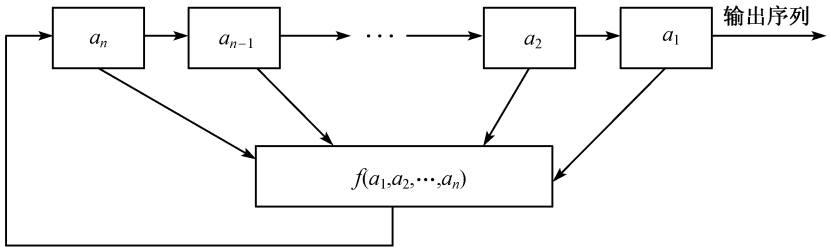

图 1.4　反馈移位寄存器

图中存储单元的个数 n 称为反馈移位寄存器的级数，在某一时刻 n 个存储单元的内容构成的向量 $(a_1, a_2, \cdots, a_n)$ 称为该移位寄存器的状态，函数 $f(a_1, a_2, \cdots, a_n)$ 为其反馈函数，当反馈函数为线性函数时，称其为线性反馈移位寄存器 (LFSR)，如果不是线性函数则称其为非线性反馈移位寄存器 (NLFSR)。一般考虑二值序列，即每个寄存器单元的值非 0 即 1，那么在有限域 GF(2) 中共有 2^{2^n} 种反馈函数。

由于线性反馈移位寄存器的反馈函数 $f(a_1, a_2, \cdots, a_n)$ 是 $a_1, a_2, \cdots, a_n$ 的线性函数，因此函数 f 可表示为：

$$f(a_1, a_2, \cdots, a_n) = c_n a_1 \oplus c_{n-1} a_2 \oplus \cdots \oplus c_1 a_n$$

其中 $c_i (i = 1, 2, \cdots, n)$ 为反馈系数，在二进制下可取 0 或 1。这样的线性函数共有 2^n 个。

如果以断开或闭合来分别表示 0 或 1，则线性反馈移位寄存器可用图 1.5 来表示。

对于 n 级线性反馈移位寄存器最多有 2^n 个状态，而全 0 状态不会转入其他状态，因此线性反馈移位寄存器的最大周期为 2^n-1，输出序列的周期与状态周期相同，也小于或等于 2^n-1。可以将这些非 0 序列的全体记为 $\Omega(f(x))$。周期为 2^n-1 的 LFSR 序列称为 m 序列。

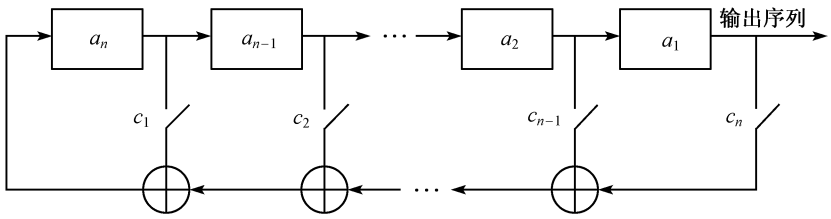

图 1.5　线性反馈移位寄存器

定义　以线性反馈移位寄存器的反馈系数决定的多项式如下：

$$f(x) = c_0 + c_1x + c_2x^2 + \cdots + c_{n-1}x^{n-1} + c_nx^n$$

称做 LFSR 的特征多项式或联系多项式，其中 $c_0 = c_n = 1$。因此 $\Omega(f(x))$ 是特征多项式为 $f(x)$ 的 LFSR 的所有输出序列集。

定义　设 $f(x)$ 为 GF(2) 上的多项式，使 $f(x)|x^n - 1$ 最小的 n 称为 $f(x)$ 的周期。

定义　设 $f(x)$ 是 n 次即约多项式，若其周期为 $2^n - 1$，则称 $f(x)$ 是 n 次本原多项式。

定理　以 $f(x)$ 为特征多项式的 LFSR 的输出序列是 m 序列的充要条件是 $f(x)$ 为本原多项式。

1.4.1　线性反馈移位寄存器周期计算实验

【实验目的】

掌握分别使用反馈参数和特征多项式求解线性反馈移位寄存器周期的方法。

【实验内容】

(1) 令 $n = 4$，$c_4 = c_3 = c_0 = 1$，$c_2 = c_1 = 0$，初始值 1011，计算该移位寄存器的输出序列，并计算周期；写出其特征多项式，并求出该特征多项式的周期。

(2) 令 $n = 5$，$c_5 = c_3 = c_1 = c_0 = 1$，$c_4 = c_2 = 0$，初始值 01010，计算该移位寄存器的输出序列，并计算周期；写出其特征多项式，并求出该特征多项式的周期。

【实验结果】

(1) 输出序列（从左至右）为 110101111000100，周期为 15；特征多项式为 $f(x) = 1 + x^3 + x^4$，周期为 15。

(2) 输出序列（从左至右）为 010100001110110，周期为 15；特征多项式为 $f(x) = 1 + x + x^3 + x^5$，周期为 15。

1.4.2　反馈参数计算实验

【实验目的】

掌握根据输出序列推导反馈参数的方法。

【实验内容】

已知一个 5 级线性反馈移位寄存器，输出序列（从左至右）为 0101111000，计算与此相对应的反馈参数（$c_5c_4c_3c_2c_1c_0$ 的值）。

【实验结果】

$$c_5 = c_3 = c_1 = c_0 = 1，c_4 = c_2 = 0$$

1.5　DES 算法实验

对称密码算法又称为传统密码算法，是应用较早的加密算法，技术比较成熟。在对称加密算法中，数据发信方将明文（原始数据）和加密密钥一起经过特殊加密算法处理后，使其变成复杂的加密密文发送出去。收信方收到密文后，若想解读原文，则需要用加密时使用的密钥及相同算法的逆算法对密文进行解密，才能使其恢复成可读明文。在对称加密算法中，使用的密钥只有一个，发收信双方都使用这个密钥对数据进行加密和解密，这就要求解密方事先必须知道加密密钥。对称加密算法的特点是算法公开、计算量小、加密速度快、加密效率高；其不足之处是，通信双方都使用同样的钥匙，安全性得不到保证。此外，每对用户每次采用对称加密算法时，都需要使用其他人不知道的唯一钥匙，这会使得发收信双方所拥有的钥匙数量成几何级数增长，密钥管理成为用户的负担。对称加密算法在分布式网络系统上应用较为困难，主要是因为密钥管理困难，使用成本较高。目前使用较多的对称密码算法有 DES 算法、3DES 算法以及 AES 算法。

DES 的前身是 1971 年由 IBM 公司的 Horst Feistel 领导研制的 LUCIFER 算法，其密码设计思想 Feistel 网络充分体现了香农提出的混淆和扩散原则，DES 沿用了这一思想。DES 的数据分组长度是 64b，密文分组长度也是 64b，不存在数据扩展问题。密钥长度是 64b，但有 8bit 是奇偶校验位，因此有效密钥长度实际上是 56b。

DES 是迄今为止应用最广泛的一种密码算法，也是最有代表意义的分组加密体制。虽然它也受到了很猛烈的批评，而且随着 AES 的提出，它不会长期成为数据加密标准，但是对它的基本原理、安全性分析、实际应用等进行较为深入的研究，对于掌握分组密码理论及进行相关领域的研究都是很有帮助的。

1.5.1　DES 单步加密实验

【实验目的】

(1) 掌握 DES 算法的基本原理。

(2) 了解 DES 算法的详细步骤。

【实验环境】

(1) 本实验需要密码教学实验系统的支持。

(2) 操作系统为 Windows 2000 或者 Windows XP。

【实验预备知识点】

DES 算法的概念。

【实验内容】

(1) 掌握 DES 算法的原理及过程。

(2) 完成 DES 密钥扩展运算。

(3) 完成 DES 数据加密运算。

【实验步骤】

(1) 打开 “DES 理论学习”，掌握 DES 算法的加解密原理。

(2) 打开 “DES 算法流程”，开始进行 DES 单步加密实验，如图 1.6 所示。

图 1.6　DES 单步加密实验界面

(3) 选择密钥输入为 ASCII 码或十六进制码模式，输入密钥；若为 ASCII 码模式，则输入 8 个字符的 ASCII 码；若为十六进制码模式，则输入 16 个字符的十六进制码（0～9，a～f，A～F）。

(4) 单击 “比特流” 按钮，将输入的密钥转化为 64 位比特流。

(5) 单击 “置换选择 I” 按钮，完成置换选择 I 运算，得到 56b 有效密钥位，并分为左右两部分，各 28b。

(6) 单击 C0 下的 “循环左移” 按钮，对 C0 进行循环左移运算。

(7) 单击 D0 下的 “循环左移” 按钮，对 D0 进行循环左移运算。

(8) 单击 “选择置换 II” 按钮，得到扩展子密钥 K1。

(9) 进入第二部分 —— 加密，选择加密输入为 ASCII 码或十六进制码模式，输入明文。若为 ASCII 码模式，则输入 8 个字符的 ASCII 码；若为十六进制码模式，则输入 16 个字符的十六进制码（0～9，a～f，A～F）。

(10) 单击 “比特流” 按钮，将输入明文转化为 64 位比特流。

(11) 单击 “初始 IP 置换” 按钮，对 64bit 明文进行 IP 置换运算，得到左右两部分，各 32bit。

(12) 单击 “选择运算 E” 按钮，将右 32b 扩展为 48b。

(13) 单击 “异或运算” 按钮，对扩展的 48bit 与子密钥 K1 进行按位异或。

(14) 依次单击 S1、S2、S3、S4、S5、S6、S7、S8 按钮，对中间结果分组后进行 S 盒运算。

(15) 单击 “置换运算 P” 按钮，对 S 盒运算结果进行 P 置换运算。

(16) 单击 “异或运算” 按钮，对 P 置换运算结果与 L0 进行按位异或，得到 R1。

(17) 单击 “逆初始置换 IP_1” 按钮，得到最终的加密结果。

【实验思考题】

(1) DES 算法中大量的置换运算的作用是什么？

(2) DES 算法中 S 盒变换的作用是什么？

1.5.2　DES 加解密实验

【实验目的】

(1) 掌握 DES 运算的基本原理。

(2) 了解 DES 运算的实现方法。

【实验环境】

(1) 本实验需要密码教学实验系统的支持。

(2) 操作系统为 Windows 2000 或者 Windows XP。

【实验预备知识点】

(1) DES 算法的特点。

(2) DES 算法的加解密过程。

(3) DES 的工作模式及其特点。

【实验内容】

(1) 掌握 DES 算法的原理及过程。

(2) 完成字符串数据的 DES 加密运算。

(3) 完成字符串数据的 DES 解密运算。

【实验步骤】

(1) 打开 “DES 理论学习”，掌握 DES 算法的加解密原理。

(2) 打开 “DES 实例”，进行字符串的加解密操作，如图 1.7 所示。

(3) 选择 “工作模式” 为 ECB 或 CBC 或 CFB 或 OFB。

(4) 选择 “填充模式” 为 ISO_1 或 ISO_2 或 PAK_7。

(5) 输入明文前选择 ASCII 码或十六进制码输入模式，然后在明文编辑框内输入待加密的字符串。

(6) 输入密钥前选择 ASCII 码或十六进制码输入模式，然后在密钥编辑框内输入密钥：若为 ASCII 码模式，则输入不超过 8 个字符的 ASCII 码，不足部分将由系统以 0x00 补足；若为十六进制码模式，则输入不超过 16 个字符的十六进制码（0~9，a~f，A~F），不足部分将由系统以 0x00 补足。

(7) 单击 “加密” 按钮，进行加密操作，密钥扩展的结果将显示在列表框中，密文将显示在密文编辑框中。

(8) 单击 “解密” 按钮，密文将被解密，显示在明文编辑框中，填充的字符将被自动除去；也可以修改密钥，再单击 “解密” 按钮，观察解密是否正确。

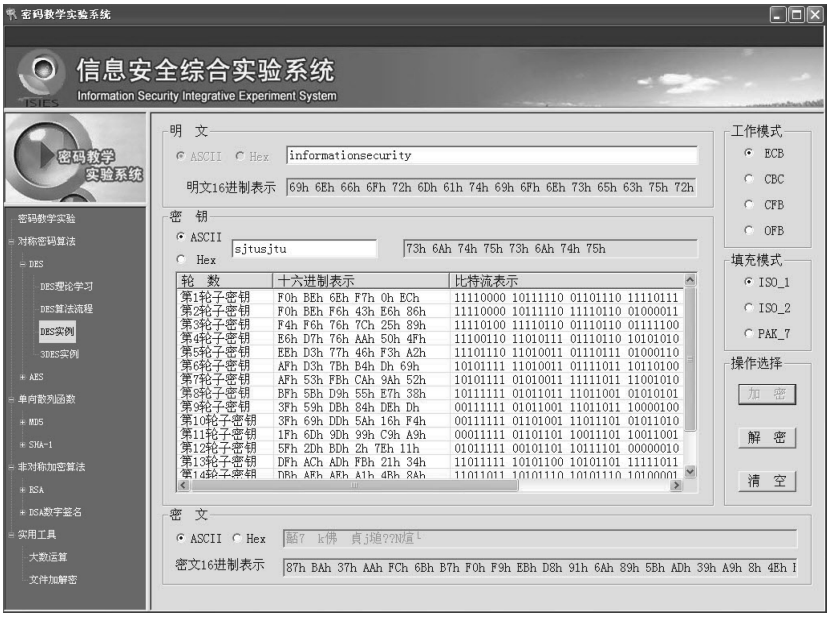

图 1.7　DES 算法实验界面

(9) 单击 “清空” 按钮即可进行下次实验。

【实验思考题】

在 DES 算法中哪些是弱密钥? 哪些是半弱密钥?

1.5.3　3DES 算法实验

【实验目的】

(1) 了解 3DES 算法的基本原理。

(2) 掌握 3DES 算法的实现方法。

【实验环境】

(1) 本实验需要密码教学实验系统的支持。

(2) 操作系统为 Windows 2000 或者 Windows XP。

【实验预备知识点】

(1) DES 之后，为什么要有 3DES?

(2) 就密钥的长度而言，3DES 的有几种加密方式?

【实验内容】

(1) 完成单块的数据的 3DES 3 密钥加密运算。

(2) 完成单块的数据的 3DES 2 密钥加密运算。

【实验步骤】

(1) 熟悉 3DES 运算原理。

(2) 掌握在不同密钥数量的情况下，3DES 的数学公式表示。

(3) 在密码教学系统中打开 “3DES 实例”，如图 1.8 所示。

图 1.8　3DES 算法实验界面

(4) 选择 “工作模式” 为 ECB 或 CBC 或 CFB 或 OFB。

(5) 选择 “填充模式” 为 ISO_1 或 ISO_2 或 PAK_7。

(6) 输入明文前选择 ASCII 码或十六进制码输入模式，然后在明文编辑框内输入待加密的字符串。

(7) 选择密钥长度为 16 字节或 24 字节，分别代表双密钥或三密钥。

(8) 输入密钥前选择 ASCII 码或十六进制码输入模式，然后在密钥编辑框内输入密钥：若为 ASCII 码模式，则输入 16 个或 24 个字符的 ASCII 码，不足部分将由系统以 0x00 补足；若为十六进制码模式，则输入不超过 32 个或 48 个字符的十六进制码（0∼9，a∼f，A∼F），不足部分将由系统以 0x00 补足。

(9) 单击 “加密” 按钮，进行加密操作，密钥扩展的结果将显示在列表框中，密文将显示在密文编辑框中。

(10) 单击 “解密” 按钮，密文将被解密，显示在明文编辑框中，填充的字符将被自动除去；也可以修改密钥，再单击 “解密” 按钮，观察解密是否正确。

(11) 单击 “清空” 按钮即可进行下次实验。

【实验思考题】

将下面两个密钥中的有效比特列出来：

k1：12345678　　k2：23456789

1.6　MD5 算法实验

单向散列函数又称为哈希（Hash）函数，将任意长度的消息 M 映射/换算成固定长度值 h（散列值，或消息摘要 MD, Message Digest），其最大的特点为具有单向性。Hash 函数用于

消息认证（或身份认证）以及数字签名。其特性如下：

(1) 给定 M，可以很容易算出 $h = H(M)$。

(2) 给定 h，根据 $H(M) = h$ 反推出 M 是非常困难的。

(3) 给定 M，要找到另外一个消息 M_*，使其满足 $H(M_*) = H(M) = h$ 是非常困难的。

散列函数可以与密钥一起使用，也可以不与密钥一起使用。如果使用了密钥，则可能会同时使用对称密钥（单密钥）和非对称密钥（公钥/私钥对）。对称密钥要求加密和解密过程使用相同的密钥，这样，密钥必须只能被加解密双方所知道，否则就不安全。这种技术安全性不高，但是效率高。非对称密钥的加密和解密使用不同的密钥，分别叫做 “公钥” 和 “私钥”。顾名思义，“私钥” 就是不能让别人知道的，而 “公钥” 就是可以公开的。二者必须配对使用，用公钥加密的数据必须用与其对应的私钥才能解开。这种技术安全性高，应用广泛，但是效率太低。常用的散列算法有 MD5 和 SHA-1。

MD5 是由 RSA 公钥加密方案发明人之一 ——Ron Rivest 设计的一个散列函数。MD5 可以对不同长度的数据块进行暗码运算，得到一个 128 位的数值。目前人们已经了解到 MD5 具有一些缺点，应尽量避免使用它，因此通常建议使用 SHA-1。SHA-1（安全散列算法 -1）是一种类似于 MD5 的算法, 该算法旨在与数字签名标准（DSS）配合使用。美国的两个机构 NIST（国家标准和技术研究所）和 NSA（国家安全局）负责 SHA-1。SHA-1 可接纳一个和多个 512 位（64 字节）的数据块，并生成一个 160 位（20 字节）的散列结果，一般认为这种较长的输出比 MD5 更安全。

【实验目的】

(1) 了解 MD5 算法的基本原理。

(2) 掌握 MD5 算法的实现方法。

【实验环境】

(1) 本实验需要密码教学实验系统的支持。

(2) 操作系统为 Windows 2000 或者 Windows XP。

【实验预备知识点】

(1) 散列函数 MD5 的作用。

(2) MD5 算法的原理和过程。

【实验内容】

(1) 掌握 MD5 算法的原理及过程。

(2) 完成字符串数据的 MD5 运算以及完整性检验。

(3) 完成文件数据的 MD5 运算以及完整性检验。

【实验步骤】

(1) 单击 “MD5 理论学习”，掌握 MD5 算法的基本原理。

(2) 单击 “MD5 实例” 开始进行实验，如图 1.9 所示。

(3) 选择 “字符串”，在报文 1 编辑框中输入字符串，如 abcdefghijklmnopqrstuvwxyz，单击 “计算 MD5 值” 按钮，计算结果显示在对应的编辑框中。

图 1.9　MD5 算法实验

(4) 在报文 2 编辑框中输入对比字符串，如 aacdefghijklmnopqrstuvwxyz，单击 “计算 MD5 值” 按钮，计算结果显示在对应的编辑框中。

(5) 单击 “异或比较” 按钮，两个报文的 MD5 值的异或值将显示出来。

(6) 选择 “文件”，单击报文 1 后面的 “浏览” 按钮，选择文件，单击 “计算 MD5 值” 按钮，计算结果显示在对应的编辑框中。

(7) 单击报文 2 后的 “浏览” 按钮，选择对比文件，单击 “计算 MD5 值” 按钮，计算结果显示在对应的编辑框中。

(8) 单击 “异或比较” 按钮，两个文件的 MD5 值的异或值将显示出来，若为全 0 则表示文件内容相同。

【实验思考题】

改变报文中的一个比特值最多有可能影响 MD5 值中的多少比特？

1.7　RSA 算法实验

非对称密码术也被称做公钥密码术，其思想是由 W.Diffie 和 Hellman 于 1976 年提出的。不同于以往的加密技术，非对称密码技术是建立在数学函数基础上的，而不是建立在位方式的操作上的。更重要的是，与只使用单一密钥的传统加密技术相比，它在加解密时，分别使用了两个不同的密钥：一个可对外界公开，称为 “公钥”；一个只有所有者知道，称为 “私钥”。公钥和私钥之间具有紧密联系，用公钥加密的信息只能用相应的私钥解密，反之亦然。同时，要想由一个密钥推知另一个密钥，在计算上是不可能的。

非对称加密算法的基本原理是，如果发信方想发送只有收信方才能解读的加密信息，发信方必须首先知道收信方的公钥，然后利用收信方的公钥来加密原文；收信方收到加密密文

后，使用自己的私钥才能解密密文。显然，采用不对称加密算法，在通信之前，收信方必须将自己早已随机生成的公钥送给发信方，而自己保留私钥。由于不对称算法拥有两个密钥，因而特别适用于分布式系统中的数据加密。广泛应用的不对称加密算法有 RSA 算法和美国国家标准局提出的 DSA。

下面就来看一下在一个 RSA 加密体制中，某一个用户 i 的公钥及私钥的生成过程。

首先该用户随机地选取两个大素数 p_i 和 q_i，并计算

$$n_i = p_i \times q_i$$

及其欧拉函数值

$$\Phi(n_i) = (p_i - 1) \times (q_i - 1)$$

然后随机地选取一整数 e_i，满足 $1 \leqslant e_i \leqslant \Phi(n_i)$，且 $(e_i, \Phi(n_i))A\# = 1$。因此在模 $\Phi(n_i)$ 下，e_i 有逆元。可以利用欧几里德算法计算 d_i，使得

$$d_i \cdot e_i = 1 \bmod \Phi(n_i)$$

至此，用户 i 就可以公布 (n_i, e_i)，将其作为公钥；而 d_i 是私钥，予以保密，p_i 和 q_i 也要保密，或者立刻销毁。

相应的，加密算法为

$$c = E(e_i, m) = m^{e_i} \bmod n_i$$

而解密算法为

$$m = D(d_i, c) = c^{d_i} \bmod n_i$$

只要能够证明由解密运算可以恢复出明文，就可以证明该加解密机制是正确的。证明过程如下：

$$\begin{aligned} D(d_i, c) &= c^{d_i} \bmod n_i = (m^{e_i})^{d_i} \bmod n_i = m^{k\Phi(n_i)+1} \bmod n_i \\ &= (m^{k\Phi(n_i)} \times m) \bmod n_i \end{aligned}$$

若 $(m, n_i) = 1$，则由欧拉定理 $m^{\Phi(n_i)} = 1 \bmod n_i$，所以上式 $= m \bmod n_i = m$。

若 $(m, n_i) \neq 1$，因为 $n_i = p_i \times q_i$，所以 (m, n_i) 必含 p_i 或 q_i，不妨设为 p_i，即 $(m, n_i) = p_i$，则有 $m = c \times p_i$，$1 \leqslant c < q_i$，故

$$m^{\Phi(q_i)} = 1 \bmod q_i,\ mk\Phi(q_i)(p_i - 1) = 1 \bmod q_i$$

因此由 $m^{k\Phi(n_i)} = 1 + aq_i$ 得

$$m^{k\Phi(n_i)+1} = m \times (1 + aq_i) = m + acq_i p_i = m + acn_i$$

所以得 $m^{k\Phi(n_i)+1} = m \bmod n_i = m$。可以看出，RSA 算法的陷门在于模指数函数的单向性。

【实验目的】

(1) 了解 RSA 算法的基本原理。

(2) 掌握 RSA 算法的实现方法。

【实验环境】

(1) 本实验需要密码教学实验系统的支持。

(2) 操作系统为 Windows 2000 或者 Windows XP。

【实验预备知识点】

(1) RSA 密码系统所基于的数学难题是什么？

(2) RSA 密码系统可以取代 DES、3DES 等公钥密码系统吗？

【实验内容】

(1) 自行以 2 位小素数为 p、q，3 为公钥 e，构造一个小的 RSA 系统，对 “1、2、3、4” 这 4 个字母的 ASCII 码进行加密和解密。

(2) 在密码教学系统中实现 RSA 运算的大素数、公钥、私钥的生成、明文加解密、分块大小的选择。

(3) 了解在不同分块大小的情况下，RSA 系统的密文长度也会有所变化。

(4) 了解在不同参数的情况下，RSA 系统的性能变化。

【实验步骤】

(1) 熟悉 RSA 运算原理。

(2) 打开 “非对称加密算法” 中的 “加密” 选项下的 RSA，选择 “RSA 实例”，如图 1.10 所示。

图 1.10　RSA 算法实验

(3) 选择密钥长度为 128、256、512 或者 1024 比特。

(4) 单击 GetPQ 按钮，得到两个大素数。

(5) 单击 GetN 按钮，得到一个由两个大素数的积构成的大整数。

(6) 单击 GetDE 按钮，得到公钥和私钥。

(7) 在明文对话框中输入需要加密的明文字符串。

(8) 单击 “获得明文 ASCII” 按钮可得到明文的 ASCII 码。

(9) 输入分块长度，或者通过单击 “推荐值” 按钮直接获得。

(10) 单击 “加密” 按钮可获得加密后的密文，单击 “解密” 按钮可获得解密后的明文。

(11) 反复使用 RSA 实例，通过输入不同大小的分片，了解密文长度的变化。

(12) 反复使用 RSA 实例，通过输入不同的安全参数，了解 RSA 密码系统的性能与参数关系。

【实验思考题】

(1) 对于 128b 的 AES 算法，需要安全参数为多少的 RSA 系统与之相匹配？

(2) RSA 系统的安全参数是什么意思？安全参数为 1024b 的 RSA 系统，其模数 n 大约为多少比特？

1.8 SHA-1 算法实验

SHA（Secure Hash Algorithm）是由美国国家安全局（NSA）设计，美国国家标准与技术研究院（NIST）发布的一系列密码散列函数。正式名称为 SHA 的家族第一个成员发布于 1993 年。然而现在的人们给它取了一个非正式的名称 SHA-0 以避免与它的后继者相混淆。两年之后，SHA-1—— 第一个 SHA 的后继者发布了。另外还有 4 种变体，曾经被发布以提升输出的范围和变更一些细微设计:SHA-224、SHA-256、SHA-384 和 SHA-512（这些有时候也被称做 SHA-2）。关于 SHA-1 的更多细节请参考相关资料。

【实验目的】

(1) 了解 SHA-1 算法的基本原理。

(2) 掌握 SHA-1 算法的实现方法。

【实验环境】

(1) 本实验需要密码教学实验系统的支持。

(2) 操作系统为 Windows 2000 或者 Windows XP。

【实验预备知识点】

(1) 散列函数 SHA-1 的作用。

(2) SHA-1 算法的原理过程。

【实验内容】

(1) 掌握 SHA-1 算法的原理及过程。

(2) 完成字符串数据的 SHA-1 运算以及算法流程。

【实验步骤】

(1) 单击 “SHA-1 实例”，开始实验，如图 1.11 所示。

(2) 单击消息编辑框，输入要填充的消息，如 abcdefghijklmnopqrstuvwxyzSHA-1 实验。

图 1.11　SHA-1 算法实验

(3) 单击 “填充” 按钮，计算结果显示在对应的编辑框中，以十六进制显示。

(4) 单击 “计算第一个填充块的W[0]---W[79]”，编辑框中可以得到第一个填充块的W[0]---W[15]，以及计算 W[16] 所需的 W[0]、W[2]、W[8]、W[13] 的十六进制以及二进制显示。

(5) 单击 “W[16]=”，其下的编辑框中显示 W[16] 的十六进制，以及 W[16] 计算过程的二进制表示。

(6) 在 “第一次循环运算” 组合框中单击 “≪ 5”，右方编辑框得到 a 左移 5 位的十六进制表示，b 编辑框显示 a 的传递值。

(7) 单击 “≪ 30”，c 编辑框显示 b 左移 30 位的十六进制表示。

(8) 单击 f[0](b,c,d) 按钮即可在后面显示 f[0] 的计算结果，d、e 编辑框分别显示 c、d 的传递值。

(9) 单击 Temp，Temp 编辑框显示 Temp 的计算结果，并在 a 编辑框中同时显示。

(10) 单击 “再经过 79 次运算”，其下的编辑框中显示 80 次运算后的十六进制值。

(11) 单击 “摘要”，系统在其右的编辑框中显示第一个填充块的摘要的十六进制值，页面底部的 “摘要” 编辑框中显示总的摘要。

(12) 若消息长度大于 56 字节，则有两个填充块，单击 “第二填充块 80 次循环的计算结果”，其下的编辑框显示第二填充块 80 次循环计算的十六进制结果。

(13) 实验结束，可以进行下一次实验。

【实验思考题】

比较 SHA-1 算法与 MD5 算法的异同点。

1.9　AES 算法实验

为了确定美国政府在 21 世纪应用的数据加密标准，美国国家标准技术研究所于 1997

年 4 月 15 日发起了征集先进加密标准 AES（Advanced Encryption Standard）的活动，并于 1997 年 9 月 12 日在联邦登记处（FR）公布了征集 AES 候选算法的通告，目的是确定一个非保密的、公开披露的、全球免费使用的分组密码算法，用于保护 21 世纪政府的敏感信息，并希望能够成为秘密和公开部门的数据加密标准。1998 年 8 月，NIST 召开了第一次 AES 候选会议，并公布了 15 个符合基本要求的候选算法，1999 年 3 月，NIST 举行了第二次 AES 候选会议，从 15 个候选算法中筛选出 5 个候选者，2000 年 4 月，NIST 举行了第三次 AES 候选会议，对这 5 个候选算法又进行了讨论。本章将较为详细地分析 5 个 AES 候选算法。2000 年 10 月 2 日，NIST 公开了最终评选结果，将 Rijndael 算法作为 AES 标准算法。2001 年 11 月 26 日，NIST 正式公布了高级加密标准 AES，并于 2002 年 5 月 26 日正式生效。

对 AES 的基本要求是比三重 DES 快而且至少和三重 DES 一样安全，分组长度为 128 位，密钥长度为 128/192/256 位可选。NIST 对 AES 进行评估的主要准则是安全性、效率和算法的实现。其中安全性是第一位的，算法应能抵抗已有的密码攻击方法和可能的密码分析方法。在保证安全性的前提下，效率是最重要的评估因素，包括算法在不同平台上的计算速度和对内存的需求等。算法的实现主要是指其灵活性，如算法可以用软件和硬件实现、可以作为序列密码、杂凑算法实现、在不同的环境中都能够有效地实现和运行等。Rijndael 算法最终获胜，成为新的数据加密标准。关于该算法的详细过程请参考《现代密码技术》（李建华主编，机械工业出版社出版，2007 年）一书。

【实验目的】

(1) 了解 AES 算法的基本原理。

(2) 掌握 AES 算法的实现方法。

【实验环境】

(1) 本实验需要密码教学实验系统的支持。

(2) 操作系统为 Windows 2000 或者 Windows XP。

【实验预备知识点】

(1) AES 中有限域上的数学运算。

(2) AES 算法的特点。

【实验内容】

(1) 掌握 AES 算法的原理及过程。

(2) 完成字符串数据的 AES 加密运算。

(3) 完成字符串数据的 AES 解密运算。

【实验步骤】

(1) 打开 “AES 理论学习”，掌握 AES 加密标准的原理。

(2) 打开 “AES 实例”，如图 1.12 所示，进行字符串的加解密操作。

(3) 选择 “工作模式” 为 ECB 或 CBC 或 CFB 或 OFB。

(4) 选择 “填充模式” 为 ISO_1 或 ISO_2 或 PAK_7。

图 1.12　AES 算法实验

(5) 输入明文前选择 ASCII 码或十六进制码输入模式，然后在明文编辑框中输入待加密的字符串。

(6) 输入密钥前选择 ASCII 码或十六进制码输入模式，然后在密钥编辑框中输入密钥；若为 ASCII 码模式，则输入不超过 16 个字符的 ASCII 码，不足部分将由系统以 0x00 补足；若为十六进制码模式，则输入不超过 32 个字符的十六进制码（0~9，a~f，A~F），不足部分将由系统以 0x00 补足。

(7) 单击 “加密” 按钮，进行加密操作，密钥扩展的结果将显示在列表框中，密文将显示在密文编辑框中。

(8) 单击 “解密” 按钮，密文将被解密，显示在明文编辑框中，填充的字符将被自动除去；也可以修改密钥，再单击 “解密” 按钮，观察解密是否正确。

(9) 单击 “清空” 按钮即可进行下次实验。

【实验思考题】

对于长度不足 16 字节整数倍的明文进行加密，除了填充这个办法，还有没有其他的方法？

1.10　DSA 数字签名实验

采用公钥密码体制的一个重要优点就是可以通过数字签名机制达到身份认证、防否认等目的。

假设用户 A 要将一消息 m 及其签名一起发给 B，则 A 用他的私钥对 m 签名为

$$S = m^{d_{\mathrm{A}}} \bmod n_{\mathrm{A}}$$

B 接收到 (m, S) 后，首先验证 S 是否正确，即验证

$$S^{e_{\mathrm{A}}} \bmod n_{\mathrm{A}} = m$$

是否成立。若成立，则 S 是 m 的签字；否则认为该消息不可信。

由于签名是采用签名者的私钥进行的，而该私钥是保密的，因此任何人都不能伪造签名；另一方面，对签名的验证采用签名者的公钥，因此易于证实该签名的合法性。而采用 RSA 公钥体制很容易实现这一过程。

【实验目的】

(1) 了解数字签名的基本原理。

(2) 掌握运用 RSA 算法实现数字签名的方法。

【实验环境】

(1) 本实验需要密码教学实验系统的支持。

(2) 操作系统为 Windows 2000 或者 Windows XP。

【实验预备知识点】

(1) 散列函数 MD5 的作用。

(2) MD5 算法的原理过程。

(3) RSA 算法的原理过程。

(4) 数字签名算法的基本原理。

【实验内容】

(1) 掌握 MD5 算法以及 RSA 算法的原理及过程。

(2) 完成字符串数据的 MD5 运算以及完整性检验。

(3) 掌握数字签名算法的基本原理及过程。

(4) 完成对字符串数据及文件的数字签名过程。

(5) 会计算 RSA 算法中的各个参数值。

【实验步骤】

(1) 单击 “DSA 数字签名理论学习”，学习 DSA 原理。

(2) 单击 “DSA 数字签名实例”，开始进行数字签名实验，如图 1.13 所示。

(3) 选择 “字符串” 或者 “文件”。选择 “字符串” 时，在报文输入框中输入字符串，选择 “文件” 时，单击 “浏览” 按钮，选择需要计算 MD5 值的文件。

(4) 单击 “计算 MD5 值”，系统在相应的编辑框中显示用户输入的字符串或者选择的报文的 MD5 值。

(5) 选择并计算签名所需的各个参数，包括 p、q 和 n 等。单击 “检验” 按钮，检查用户输入的正确性。

图 1.13　DSA 实验

(6) 单击 “数据清空”，以清空上次实验值。

(7) 计算并输入 MD5 值 RSA 算法签名的前 8 位，MD5 值的分块大小默认为 8 b，即 2 位十六进制数。计算过程中，取计算出的签名值的前 8 位输入。单击 “检验并生成签名”，系统检验用户输入的签名值的正确性。

(8) 单击 “签名并验证” 框中的 “验证” 按钮，系统显示签名验证值。

【实验思考题】

DSA 算法的安全性是建立在什么基础之上的？

1.11　ECC 算法实验

椭圆曲线已经被广泛研究了 100 多年，而且从中得出了非常广泛的研究课题。椭圆曲线现在已经成为很多重要应用领域的工具，如编码理论、伪随机比特生成以及数论算法（素性证明、整数分解）等。

椭圆曲线密码（ECC）系统自 1985 年由 Neal Koblitz 和 Victor Miller 分别独立提出以后，人们对它的安全性和实现的有效性进行了广泛和深入的研究。椭圆曲线密码系统是建立在椭圆曲线点群的离散对数问题上的。在有限域上椭圆曲线点群中，还没有关于寻找离散对数的诸如 Index-calculus 之类的亚指数时间算法出现。因此，可以利用规模更小的椭圆曲线群来达到相同的安全级别。这样，就可以拥有更小的密钥长度，更小的带宽需要，以及更快的实现。这些特性对于那些计算能力和集成芯片空间受限的安全应用特别具有吸引力。例如，应用在智能卡、PC（Personal Computer）卡，以及无线设备上。

1.11.1 椭圆曲线简介

椭圆曲线方程的一般形式为

$$E: y^2 + a_1xy + a_3y \ = \ x^3 + a_2x^2 + a_4x + a_6$$

这里，考虑 a_1, a_2, a_3, a_4, a_6 为域 K 中的元素。

域 K 上的点集为

$$E: \{(x, y) \mid y^2 + a_1xy + a_3y \ = \ x^3 + a_2x^2 + a_4x + a_6\} \cup \{O\}$$

其中 $a_1, a_2, a_3, a_4, a_6 \in K$；$\{O\}$ 为无穷远点，叫做域 K 上的椭圆曲线。

在对域 K 上的椭圆曲线 E 的研究中，我们通常取如下形式的椭圆曲线方程：

(1) 当域 K 的特征不为 2、3 时，椭圆曲线方程为

$$y^2 \ = \ x^3 + a_4x + a_6$$

(2) 当域 K 的特征为 2 时，椭圆曲线方程为

$$y^2 + xy \ = \ x^3 + a_2x^2 + a_6 \text{ 或 } y^2 + a_3y \ = \ x^3 + a_4x + a_6$$

(3) 当域 K 的特征为 3 时，椭圆曲线方程为

$$y^2 \ = \ x^3 + a_2x^2 + a_6 \text{ 或 } y^2 \ = \ x^3 + a_4x + a_6$$

椭圆曲线加法规则 设 E 是定义在域 K 上的椭圆曲线为

$$E: \{(x, y) \mid y^2 + a_1xy + a_3y \ = \ x^3 + a_2x^2 + a_4x + a_6\} \cup \{O\}$$

定义 E 上的运算法则，记为 $\oplus$。

运算法则 设 P 和 Q 是 E 上的两个点，L 是过 P 和 Q 的直线（过 P 点的切线，如果 $P = Q$），R 是 L 与曲线 E 相交的第三点。设 L' 是过 R 和 O 的直线，则 $P \oplus Q$ 就是 L' 与 E 相交的第三点。

定理 E 上的运算法则 $\oplus$ 具有如下性质：

(1) 如果直线 L 交 E 于点 P、Q 和 R（不必是不同的），则 $(P \oplus Q) \oplus R = O$。

(2) 对任意 $P \in E$，$P \oplus O = P$。

(3) 对任意 P、$Q \in (E)$，$P \oplus Q = Q \oplus P$。

(4) 设 $P \in E$，存在一个点，记做 $-P$，使得 $P \oplus (-P) = O$。

(5) 对任意 P、Q、$R \in E$，有 $(P \oplus Q) \oplus R = P \oplus (Q \oplus R)$。

这就是说，E 对于运算法则 $\oplus$ 构成一个交换群。更进一步，如果 E 定义在 K 上，则

$$E(K) = \{(x, y) \in K \times K \mid y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6\} \cup \{O\}$$

是 E 的子群。

下面给出群运算的具体计算公式。

定理　设椭圆曲线 E 的一般方程为

$$E:\{(x,y)\mid y^2+a_1xy+a_3y\ =\ x^3+a_2x^2+a_4x+a_6\}\cup\{O\}$$

设 $P_1=(x_1,y_1)$、$P_2=(x_2,y_2)$ 是曲线 E 上的两个点，则

(1) $-P_1=(x_1,-y_1-a_1x_1-a_3)$。

(2) 如果 $P_3=(x_3,y_3)=P_1+P_2\neq O$，则 x_3、y_3 可以由下列公式给出

$$\begin{cases} x_3=\lambda^2+a_1\lambda-a_2-x_1-x_2 \\ y_3=\lambda(x_1-x_3)-a_1x_3-y_1-a_3 \end{cases}$$

其中

$$\begin{cases} \lambda=\dfrac{y_2-y_1}{x_2-x_1}, & x_1\neq x_2 \\ \lambda=\dfrac{3{x_1}^2+2a_2a_1+a_1-a_1y_1}{a_1y_1+a_1x_1+a_3}, & x_1=x_2 \end{cases}$$

实数域 R 上椭圆曲线及其运算法则的几何意义是，因为实数域 R 的特征不为 2、3，所以实数域 R 上椭圆曲线 E 的方程可设为

$$E:y^2=x^3+a_4x+a_6$$

其判断式 $\Delta=-16(4{a_4}^3+27{a_6}^2)\neq 0$。这时，$E$ 在 R 上的运算规则如下：

设 $P_1=(x_1,y_1)$、$P_2=(x_1,y_1)$ 是曲线 E 上的两个点，O 为无穷远点，则

(1) $O+P_1=P_1+O$。

(2) $-P_1=(x_1,-y_1)$。

(3) 如果 $P_3=(x_3,y_3)=P_1+P_2\neq O$，则 x_3、y_3 可以由下列公式给出

$$\begin{cases} x_3=\lambda^2-x_1-x_2 \\ y_3=\lambda(x_1-x_3)-y_1 \end{cases}$$

其中

$$\begin{cases} \lambda=\dfrac{y_2-y_1}{x_2-x_1}, & x_1\neq x_2 \\ \lambda=\dfrac{3{x_1}^2+a_4}{2y_1}, & x_1=x_2 \end{cases}$$

运算法则的几何意义是：设 $P_1=(x_1,y_1)$、$P_2=(x_2,y_2)$ 是曲线 E 上的两点，O 为无穷远点，则 $-P_1$ 为过点 P_1 和点 O 的直线 L 与曲线 E 的交点，换句话说，$-P_1$ 是点 P_1 关于 x 轴的对称点。而点 P_1 与点 P_2 的和 $P_1+P_2=P_3=(x_3,y_3)$ 是过点 P_1 与点 P_2 的直线 L 与曲线 E 的交点关于 x 轴的对称点 $P_3=-R$。

素域 $F_p(p>3)$ 上有椭圆曲线 E，因为素域 F_p 的特征不是 2、3，所以素域 F_p 上椭圆曲线 E 的方程可设为

$$E:y^2=x^3+a_4x+a_6$$

其中 $\Delta=-16(4{a_4}^3+27{a_6}^2)\neq 0$。这时，$E$ 在 F_p 上的运算规则如下：

设 $P_1=(x_1,y_1)$、$P_2=(x_2,y_2)$ 是曲线 E 上的两点，O 为无穷远点，则

(1) $O+P_1=P_1+O$。

(2) $-P_1=(x_1,-y_1)$。

(3) 如果 $P_3=(x_3,y_3)=P_1+P_2\neq O$，则 x_3、y_3 可以由下列公式给出

$$\begin{cases} x_3=\lambda^2-x_1-x_2 \\ y_3=\lambda(x_1-x_3)-y_1 \end{cases}$$

其中

$$\begin{cases} \lambda=\dfrac{y_2-y_1}{x_2-x_1}, & x_1\neq x_2 \\ \lambda=\dfrac{3{x_1}^2+a_4}{2y_1}, & x_1=x_2 \end{cases}$$

域 $F_{2^n}(n\geqslant 1)$ 上有椭圆曲线 E，因为 F_{2^n} 的特征为 2，所以域 F_{2^n} 上椭圆曲线 E 的方程可设为

$$E:y^2+xy=x^3+a_2x^2+a_6$$

E 在域 F_{2^n} 上的运算规则如下：

设 $P_1=(x_1,y_1)$、$P_2=(x_2,y_2)$ 是曲线 E 上的两点，O 为无穷远点，则

(1) $O+P_1=P_1+O$。

(2) $-P_1=(x_1,-y_1)$。

(3) 如果 $P_3=(x_3,y_3)=P_1+P_2\neq O$，则 x_3、y_3 可以由下列公式给出

$$\begin{cases} x_3=\lambda^2+\lambda+x_1+x_2+a_2 \\ y_3=\lambda(x_1+x_3)+x_3+y_1 \end{cases}$$

其中

$$\begin{cases} \lambda=\dfrac{y_2+y_1}{x_2+x_1}, & x_1\neq x_2 \\ \lambda=\dfrac{{x_1}^2+y_1}{x_1}, & x_1=x_2 \end{cases}$$

域 $F_{3^n}(n\geqslant 1)$ 上椭圆曲线 E，因为域 F_{3^n} 的特征为 3，所以域 F_{3^n} 上椭圆曲线 E 的方程可设为

$$E:y^2=x^3+a_2x^2+a_6$$

E 在域 F_{3^n} 上的运算规则如下：

设 $P_1=(x_1,y_1)$，$P_2=(x_2,y_2)$ 是曲线 E 上的两点，O 为无穷远点，则

(1) $O+P_1=P_1+O$。

(2) $-P_1=(x_1,-y_1)$。

(3) 如果 $P_3=(x_3,y_3)=P_1+P_2\neq O$，则 x_3、y_3 可以由下列公式给出

$$\begin{cases} x_3=\lambda^2-x_1-x_2-a_2 \\ y_3=\lambda(x_1-x_3)-y_1 \end{cases}$$

其中

$$\begin{cases} \lambda = \dfrac{y_2 - y_1}{x_2 - x_1}, & x_1 \neq x_2 \\ \lambda = \dfrac{3{x_1}^2 + 2a_2x_2}{2y_1}, & x_1 = x_2 \end{cases}$$

1.11.2　椭圆曲线上的离散对数问题

椭圆曲线密码系统的安全是基于椭圆曲线离散对数问题（ECDLP）的，椭圆曲线离散对数问题可以描述如下。

给定一条定义于有限域 F_q 上的椭圆曲线 E，n 阶点 $P \in E(F_q)$，则有：

(1) 对任意整数 l，$0 \leqslant l \leqslant n-1$，计算点 $Q = lP$ 很容易。

(2) 对于点 Q，求解整数 x，$0 \leqslant x \leqslant n-1$，使得 $xP = Q$ 是很困难的。

1.11.3　椭圆曲线密码算法

将椭圆曲线离散对数问题应用到密码系统中，就可以得到椭圆曲线密码算法。

(1) 准备工作：有选择限域 F_q、椭圆曲线 E、基点 (x, y)，把明文编码到曲线上的点 (x_m, y_m)，即每个明文都对应一个二维表示的点，选择一个私钥 n，计算公钥 $P = n(x, y)$。

对于 A：私钥 n_A，公钥 $P_A = n_A(x, y)$；对于 B：私钥 n_B，公钥 $P_B = n_B(x, y)$。

(2) 加密 —— 对于任何想加密消息并发送给 A 的人：选择一个随机整数 k，生成密文 $C_m = \{k(x, y), (x_m, y_m) + kP_A\}$。

(3) 解密 ——A 用私钥恢复明文：计算 $n_A(k(x, y))$，计算 $(x_m, y_m) + kP_A - n_A(k(x, y)) = (x_m, y_m) + k(n_A(x, y)) - n_A(k(x, y)) = (x_m, y_m)$。

1.12　密码算法分析设计实验

本实验为拓展实验。

【实验目的】

掌握各种密码算法的分析以及编程实现的方法。

【实验预备知识点】

(1) 几种基本密码算法的原理与流程（DES、AES、RSA、MD5）。

(2) VC++ 编程基础。

【实验内容】

(1) 设计实现密码算法所需的函数。

(2) 编写 VC++ 程序实现加解密功能。

(3) 对各个算法的运行效率做比较分析。

(4) 总结各个算法的特点及差异。

1.13 密码技术应用实验

本实验为创新实验。

【实验目的】

(1) 了解 PGP 协议的具体内容。

(2) 掌握密码技术在工程实际中的应用。

【实验预备知识点】

(1) PGP 邮件加密标准 http://baike.baidu.com/view/7607.htm。

(2) RSA 加密算法原理及实现。

(3) VC++ 编程基础。

【实验内容】

(1) 设计实现 PGP 标准所需的函数。

(2) 编写 VC++ 程序实现该标准要求的功能。

(3) 对设计实现的程序进行测试。

(4) 对测试效果进行分析并提出改进方案。