第 2 章　密码学基础

本章学习要求

- 掌握对称密码(凯撒密码、栅栏密码、DES)使用方法、DES 对称密码算法；
- 掌握 RSA 加密用法和签名用法、RSA 公钥密码算法；
- 掌握椭圆曲线加法规则和倍点计算方法、椭圆曲线密码原理；
- 掌握混合密码系统使用方法、对称密钥分发方法、密码学语言；
- 掌握身份认证和消息认证、MD5 算法；
- 掌握国密密码使用方法，认识国密密码的优点；
- 掌握应用密码学知识保护网上银行支付安全。

2.1　密码学定义

密码学是研究数据变换(加密、解密、摘要等)的科学，是数学和计算机的交叉学科，和信息论密切相关。

密码学常用的两个数据变换方法是：

(1) 替代(substitution cipher)，按照一定规则将一组字符(字母)换成其他字符。例如“fly at once”变成“gmz bu podf”(每个字母用下一个字母替代，凯撒密码)。替代的特点是字母形式发展了变化，但字母所在位置不变，即字符变，位置不变。

(2) 置换(transposition cipher)，将字符位置(字母顺序)重新排列，例如“come here”变成“choemree”(4 个字母一行写成两行，按列重新书写，栅栏密码)。置换的特征是字母所在位置发展了变化，但字母形式不变，即位置变，字符不变。

直观地讲，变换就是把具有明确含义的字符串(英文单词)变换为无明确含义的字符串，从而实现隐藏信息涵义的目的。或者说是研究如何将机密信息进行特殊的编码，以形成不可识别的密码形式(密文)进行传递。

根据信息论，变换增加了字符串含义的不确定性，信息熵值增加。例如字符串“fly at once”具有明确含义，字符串“gmz bu podf”则需要多次尝试才能确定其含义。

密码学的发展经历了古典密码、近代密码、现代密码 3 个发展阶段：

(1) 古典密码阶段，从古代到 19 世纪末，长达几千年。密码体制为纸、笔或者简单器械实现的简单替代及置换，通信手段为信使，例如凯撒密码、栅栏密码等。

(2) 近代密码阶段，从 20 世纪初到 20 世纪 50 年代。密码体制为手工或电动机械实现的复杂的替代及置换，通信手段为电报通信，例如 DES 密码。这一阶段密码只在很小范围内使用，如军事、外交、情报等部门。

(3) 现代密码阶段，从 20 世纪 50 年代至今。密码体制：分组密码、序列密码以及公开密钥密码，有坚实的数学理论基础。通信手段包括无线通信、有线通信、计算机网络等。

1949 年 Shannon 发表题为“保密通信的信息理论”的论文，将数学（概率论）引入了密码学，为密码系统建立了理论基础，从此密码学成了一门科学，实现了第一次飞跃。

1976 年后，美国数据加密标准（DES）的公布使密码学的研究公开，标志着密码学从军用转向军民两用，扩大了密码学应用范围，密码学得到了迅速发展。

1976 年，Diffie 和 Hellman 在文章“密码学新方向”（New Direction in Cryptography）中首次提出了公开密钥密码体制的思想，1977 年，Rivest、Shamir 和 Adleman 三个人实现了公开密钥密码体制（RSA 公开密钥体制，RSA 为三人名字首字母的缩写，三人共同获得 2015 年图灵奖者），解决了密钥分配、身份认证问题，实现了密码学的第二次飞跃。

公钥密码体制算法以数学难题为基础，要求进行复杂的计算，使得必须将计算机引入密码学。

经过发展，密码学不仅仅是编码与破译的学问，而且包括安全管理、安全设计、秘密分存、Hash 函数等内容，已被有效地、系统地用于保证信息的保密性、完整性和真实性。保密性是对信息进行加密，使非法用户无法读懂数据信息。完整性是对数据完整性的鉴别，以确定数据是否被非法篡改，保证合法用户得到正确完整的信息。真实性是数据来源的真实性、信息本身真实性的鉴别，可以保证合法用户不被欺骗。

密码学广泛应用于日常生活，包括自动柜员机的芯片卡、计算机使用者存取密码、电子商务等。

2.1.1　凯撒密码

凯撒密码是历史上第一个密码技术，是古罗马凯撒（Caesare）大帝在营救西塞罗战役时用来保护重要军情的加密系统。

凯撒密码使用替代变换方法，可分为古典凯撒密码和广义凯撒密码两种。

1. 古典凯撒密码

古典凯撒密码的变换规则是：将字符串中当前字符用字母表中其后的第 3 个字母替代，x、y、z 依次用 a、b、c 替代，实现数据加密；将字符串中当前字符用字母表中其前的第 3 个字母替代，a、b、c 依次用 x、y、z 替代，实现数据解密。

本章约定字符串仅由英文小写字母构成，不考虑 ASCII 码表其他字符，统一将大写字母转化为小写字母。

定义变换前的字符串为明文（plain text），变换后的字符串为密文（cipher text），则变换规则可以直观地表示为一张字符替代表，如表 2-1 所示。

表 2-1　古典凯撒密码明文、密文字符替代表

明文字符	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
密文字符	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

数据加密时，用户对明文中的每一个字母，依次查找表 2-1 中“明文字符”行所在的位置，然后写出“密文字符”行对应的字母，就可生成密文；数据解密时，用户通过对密文中的每一个字母，依次查找表 2-1 中“密文字符”行所在的位置，然后写出“明文字符”行对应的字母，就可生成明文。因此古典凯撒密码是单表替代密码的典范。

古典凯撒密码可以用字母移位操作实现：数据加密时，将字母按字母表顺序向右移3位；数据解密时，将字母按字母表顺序向左移3位。

【例2-1】 明文 p_1=this is a book 使用古典凯撒密码加密，求密文 c_1。

解：查表2-1可得，t→w，h→k，i→l，i→l，s→v，a→d，b→e，o→r，k→n，所以 c_1=wklv lv d errn。

【例2-2】 古典凯撒密码加密的密文 c_2=zh duh vwxghqwv，求明文 p_2。据此说明替代密码的特征。

解：查表2-1可得，z→w，h→e，d→a，u→r，v→s，w→t，x→u，g→d，h→e，q→n，所以 p_2=we are students。

替代密码的特征是字符变，位置不变。

由例2-1和例2-2可知，古典凯撒密码明文和密文存在一一对应关系，密文的安全性依赖于保密变换规则。

2. 广义凯撒密码

广义凯撒密码的变换规则是：将字符串中当前字母用字母表中其后的第 n 个字母替代，实现数据加密；将字符串中当前字母用字母表中其前的第 n 个字母替代，实现数据解密。广义凯撒密码也称为移位密码。

密码学中一般约定凯撒密码指广义凯撒密码，使用古典凯撒密码会明确标注。

变换规则中的 n 称为密钥 k，规定 $k\in\{1,2,3,4,5,\cdots,23,24,25\}$。$k$=3 时，广义凯撒密码退化为古典凯撒密码。$k$ 取不同值时，对应不同的字符替代表，数据加密、解密需要查找对应字符替代表，因此凯撒密码属于多表替代密码。k=5、21 时，字符替代表如表2-2、表2-3所示。

表2-2 凯撒密码明文、密文字符替代表(k=5)

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

表2-3 凯撒密码明文、密文字符替代表(k=21)

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

【例2-3】 p_1=this is a book，使用凯撒密码加密，求 k=3、5 时 c_1、c_2。

解：∵ k=3 ∴查找表2-1，可得 c_1=wklv lv d errn；

∵ k=5 ∴查找表2-2，可得 c_2=ymnx nx f gttp。

【例2-4】 k=5 时，凯撒密码加密的密文 c_2=bj fwj xyzijsyx，求 p_2。

解：∵ k=5，∴查找表2-2，可得 p_2=we are students。

【例2-5】 凯撒密码加密的密文 c_2=bj fwj xyzijsyx，求 k=21 时使用凯撒密码加密的结果 p_2。

解：∵ k=21，∴查找表2-3，可得 p_2=we are students。

对比例2-4和例2-5可知，解密既可以用解密变换规则实现，也可以用加密变换规则实

现。因此凯撒密码加密和解密是相对的，可以互相替代，加密变换规则和解密变换规则本质上都是移位，只是移位的方向不同；或者说都是查表，只是查的表不同而已。

对比表 2-2 和表 2-3 可知，将两表中任意一张的上下两行互换重新按字母顺序排列就可得出另一张表，或者说两张表中字母的对应关系是固定的，例如 a 对应 v，b 对应 w，……

真实的密码解密是不知道密钥的，这也是解密的魅力所在。探索不知道密钥前提下准确解密凯撒密文，对于提高读者深入理解密码学、锻炼分析问题能力、提高编程能力大有益处。

【例 2-6】 凯撒密码加密的密文 c_3 = yt gj tw sty yt gj，ymfy nx ymj vzjxynts，求明文 p_3。

解：∵密钥未知。

∴使用暴力破解方法（穷举法）破解密文。

依次查找 k = 1、2、……、25、26 时对应字符替代表得出对应明文。

key = 1，p_3 = xs fi sv rsx xs fi，xlex mw xli uyiwxmsr；

key = 2，p_3 = wr eh ru qrw wr eh，wkdw lv wkh txhvwlrq；

key = 3，p_3 = vq dg qt pqv vq dg，vjcv kpub vjg swguvkqp；

key = 4，p_3 = up cf ps opu up cf，uibu jt uif rvftujpor；

key = 5，p_3 = to be or not to be，that is the question；

key = 6，p_3 = sn ad nq mns sn ad，sgzs hr sgd ptdrshnm；

key = 7，p_3 = rm zc mp lmr rm zc，rfyr gq rfc oscqrgml；

key = 8，p_3 = ql yb lo klq ql yb，qexq fp qeb nrbpqflk；

key = 9，p_3 = pk xa kn jkp pk xa，pdwp eo pda mqaopekj；

key = 10，p_3 = oj wz jm ijo oj wz，ocvo dn ocz lpznodji；

key = 11，p_3 = ni vy il hin ni vy，nbun cm nby koymncih；

key = 12，p_3 = mh ux hk ghm mh ux，matm bl max jnxlmbhg；

key = 13，p_3 = lg tw gj fgl lg tw，lzsl ak lzw imwklagf；

key = 14，p_3 = kf sv fi efk kf sv，kyrk zj kyv hlvjkzfe；

key = 15，p_3 = je ru eh dej je ru，jxqj yi jxu gkuijyed；

key = 16，p_3 = id qt dg cdi id qt，iwpi xh iwt fjthixdc；

key = 17，p_3 = hc ps cf bch hc ps，hvoh wg hvs eisghwcb；

key = 18，p_3 = gb or be abg gb or，gung vf gur dhrfgvba；

key = 19，p_3 = fa nq ad zaf fa nq，ftmf ue ftq cgqefuaz；

key = 20，p_3 = ez mp zc yze ez mp，esle td esp bfpdetzy；

key = 21，p_3 = dy lo yb xyd dy lo，drkd sc dro aeocdsyx；

key = 22，p_3 = cx kn xa wxc cx kn，cqjc rb cqn zdnbcrxw；

key = 23，p_3 = bw jm wz vwb bw jm，bpib qa bpm ycmabqwv；

key = 24，p_3 = av il vy uva av il，aoha pz aol xblzapvu；

key = 25，p_3 = zu hk ux tuz zu hk，zngz oy znk wakyzout；

key = 26，p_3 = yt gj tw sty yt gj，ymfy nx ymj vzjxynts。

比较以上答案发现：

(1) key=5 时,p_3=to be or not to be,that is the question,每个字符串都有确切含义(都是单词),p_3 具有确定的含义,所以为正确答案。

(2) key=25 时,$p_3=c_3$,完成了一轮循环,结果开始重复,暴力破解结束。

(3) 26 种 p_3 中任一位置的字符(如第一个字符)会规律性遍历 26 个字母。

从多个可能明文中选择出正确明文的原则是:统计可能明文中出现单词个数的多少,以个数多的明文为正确答案。理想状态下,正确答案是一个具有确定含义的句子。

由例 2-6 可知凯撒密码引入了密钥的概念,强调了密钥的重要性:

(1) 明文和密文变为一对多的关系。同一个明文使用不同的密钥,会产生多个不同的密文,增加了根据密文解密明文的难度。

(2) 密码加密、解密不仅与变换规则有关,而且与密钥相关,仅仅知道变换规则和密文、不给定密钥(无密钥),就不能唯一确定明文(不确定性增加,熵增加)。

(3) 密文的安全性可以不依赖于保密变换规则,而是依赖于密钥的保密。因此保密变换规则可以公开,以扩大密码使用范围,这是密码学发展的必然。

一个密码系统可能使用的密钥集合,称为密钥空间,记为大写 K。凯撒密码的密钥空间 $K=\{1,2,3,\cdots,25\}$,密钥空间大小为 25。这意味着,对于任意一个凯撒密码加密的密文,解密都需要考虑 25 种情况。

为了加大凯撒密码的密钥空间,可以采用单字母替代密码。单字母替代密码也是一种多表替代算法,是将密文字母的顺序打乱后与明文字母对应生成字符替代表,例如表 2-4。

表 2-4 单字母替代密码明文、密文字符替代表

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

【例 2-7】 p_1=this is a book 使用单字母替代密码(见表 2-4)加密,求 c_1。

解:查表 2-4 可得,t→h,h→a,i→l,s→e,a→o,b→g,o→t,k→u,所以 c_1=hale le o gttu。

单字母替代密码的密钥空间大小为 26! $\approx 4\times10^{26}$(明文字符 a 可从 26 个字母中任选 1 个字母替代,明文字符 b 从明文字符 a 选剩下的 25 个字母中任选 1 个字母替代,……)。使用人脑进行暴力破解显然不可能。即使使用每微秒尝试一个密钥的计算机进行暴力破解,也需要花费约 10^{10} 年才能穷举所有的密钥,显然也不可行。因此扩大密钥空间是增加密码安全性的方法之一。

综上所述,替代密码可以使用移位运算或查表运算实现,具有字符变,位置不变的特征。加密、解密规则可以互相推出,加密密钥、解密密钥相同。

2.1.2 栅栏密码

典型的置换密码有栅栏密码和矩阵置换密码。

1. 栅栏密码

栅栏密码的加密变换规则是首先将明文分成 n 个字符一栏,一栏为一行构成一个行列式;然后行列式行列转置后按列上下排列字符生成密文。

将 $n=2$ 的栅栏密码称为 2 栏栅栏密码,要求明文去除空格后长度为偶数。

【例 2-8】　p_1＝there is a cipher，使用 2 栏栅栏密码加密，求 c_1。

解：(1) 去空格，p_1＝thereisacipher。

(2) 2 个字符一栏，th，er，ei，sa，ci，ph，er。

(3) 生成行列式：$\begin{pmatrix} th \\ er \\ ei \\ sa \\ ci \\ ph \\ er \end{pmatrix}$，矩阵行列转置，得 $\begin{pmatrix} teescpe \\ hriaihr \end{pmatrix}$。

(4) 合并，c_1＝teescpehriaihr。分出空格，c_1＝teesc pe h riaihr。

解密变换规则是首先将密文分成 n 行构成一个行列式；然后行列式行列转置后按行排列字符生成密文。

【例 2-9】　已知使用 2 栏栅栏密码加密的密文 c_1＝teesc pe h riaihr，求明文 p_1。

解：(1) 去掉 c_1 中的空格，c_1＝teescpehriaihr。

(2) 密文从中间分开，变为两行：$\begin{pmatrix} teescpe \\ hriaihr \end{pmatrix}$，矩阵行列转置，得 $\begin{pmatrix} th \\ er \\ ei \\ sa \\ ci \\ ph \\ er \end{pmatrix}$。

(3) 合并，p_1＝thereisacipher。分出空格，p_1＝there is a cipher。

不是所有的栅栏密码都是 2 栏的，还有多栏栅栏密码。2 栏栅栏密码通常会明确标出。

多栏栅栏密码加密、解密首先需要计算字符串(明文或密文)可以分解为几栏，然后分别进行加密、解密。如字符串长度 L＝14＝2×7，说明可分成 2 个字符一栏和 7 个字符一栏两种情况。

【例 2-10】　已知明文 p_1＝there is a cipher，使用多栏栅栏密码加密，求密文 c_1。

(1) 去空格，p_1＝thereisacipher。

(2) 分栏，L＝14＝2×7，说明可分成 2 个字符一栏和 7 个字符一栏两种情况。

(3) 7 个字符一栏，thereis，acipher 写成两行：

$\begin{pmatrix} thereis \\ acipher \end{pmatrix}$，矩阵行列转置，得 $\begin{pmatrix} ta \\ hc \\ ei \\ rp \\ eh \\ ie \\ sr \end{pmatrix}$。

合并，p_1＝tahceirpehiesr。分出空格，p_1＝tahce ir p ehiesr。

(4) 2 个字符一栏时,结果见例 2-8。

【例 2-11】 已知使用多栏栅栏密码加密的密文 c_1=teesc pe h riaihr,求明文 p_1。

解:(1) 去掉 c_1 中的空格,c_1=teescpehriaihr。

(2) 计算 c_1 的长度 L,并分解为 L 为两个自然数的乘积。$L=14=2\times7$,说明可分成 2 个字符一栏和 7 个字符一栏两种情况。

(3) 7 个字符一栏:

变为两列:$\begin{pmatrix}\text{th}\\\text{er}\\\text{ei}\\\text{sa}\\\text{ci}\\\text{ph}\\\text{er}\end{pmatrix}$,矩阵行列转置,得 $\begin{pmatrix}\text{teescpe}\\\text{hriaihr}\end{pmatrix}$。

合并,p_1=thereisacipher。分出空格,p_1=there is a cipher。

(4) 2 个字符一栏,结果见例 2-9。

2. 矩阵置换密码

矩阵置换密码通过置换矩阵控制其输出方向和输出顺序来获得密文。例如,置换矩阵 $\begin{bmatrix}12345\\24512\end{bmatrix}$ 表示将明文矩阵的第 1 列置换为输出矩阵的第 2 列,明文矩阵的第 2 列置换为输出矩阵的第 4 列……

加密变换规则:

(1) 首先计算置换矩阵列数 n,将明文分成 n 个字符一行构成明文矩阵;

(2) 将明文行列式按置换矩阵生成输出矩阵;

(3) 输出矩阵按行次序顺序排列生成密文。

【例 2-12】 加密置换矩阵 $\boldsymbol{E}=\begin{bmatrix}12345\\24513\end{bmatrix}$,$p_1$=now we are having a test,求 c_1。

解:(1) 由 $\boldsymbol{E}$ 可知 $n=5$,明文矩阵为:$\begin{bmatrix}\text{nowwe}\\\text{areha}\\\text{vinga}\\\text{testx}\end{bmatrix}$。

(最后一行不足长度 5,加原文中未使用的任一字符,如 x。)

(2) 置换,得输出矩阵 $\begin{bmatrix}\text{wneow}\\\text{haare}\\\text{gvain}\\\text{ttxes}\end{bmatrix}$。

置换过程为：$\begin{bmatrix}12345\\ \text{nowwe}\\ \text{areha}\\ \text{vinga}\\ \text{testx}\end{bmatrix}\underset{\boldsymbol{E}}{\begin{bmatrix}12345\\ 24513\end{bmatrix}}\begin{bmatrix}12345\\ \text{wneow}\\ \text{haare}\\ \text{gvain}\\ \text{ttxes}\end{bmatrix}$。

(3) 输出密文，c_1= wne ow haa regvai n ttxe。

解密变换规则：

(1) 计算置换矩阵列数 n，将密文分成 n 个字符一行构成密文矩阵；

(2) 将密文矩阵按置换矩阵生成输出矩阵；

(3) 输出矩阵按行次序顺序排列生成明文。

【例 2-13】 解密置换矩阵 $\boldsymbol{D}=\begin{bmatrix}12345\\ 41523\end{bmatrix}$，$c_1$=wne ow haa regvai n ttze，求 p_1。

解：(1) 由 $\boldsymbol{D}$ 可知 $n=5$，密文矩阵为：$\begin{bmatrix}\text{wneow}\\ \text{haare}\\ \text{gvain}\\ \text{ttzex}\end{bmatrix}$。

(最后一行不足长度 5，加原文中未使用的任一字符，如 x。)

(2) 置换，得输出矩阵 $\begin{bmatrix}\text{nowwe}\\ \text{areha}\\ \text{vinga}\\ \text{zestx}\end{bmatrix}$。

置换过程为：$\begin{bmatrix}12345\\ \text{wneow}\\ \text{haare}\\ \text{gvain}\\ \text{ttxez}\end{bmatrix}\underset{\boldsymbol{D}}{\begin{bmatrix}12345\\ 41523\end{bmatrix}}\begin{bmatrix}12345\\ \text{nowwe}\\ \text{areha}\\ \text{vinga}\\ \text{testx}\end{bmatrix}$。

(3) 输出明文，p_1= now we are having a test。

比较以上两例，可得加密矩阵 $\boldsymbol{E}$ 和解密矩阵 $\boldsymbol{D}$ 可相互转换：将 $\boldsymbol{E}$ 两行互换，再从小到大排列，即得 $\boldsymbol{D}$；反之亦然。

$$\boldsymbol{E}=\begin{bmatrix}12345\\ 24513\end{bmatrix}\Leftrightarrow\boldsymbol{D}\begin{bmatrix}12345\\ 41523\end{bmatrix}$$

【例 2-14】 $n=5$，c_1=wynoo reuha inagv esatt，求 p_1。

解：(1) $n=5$，密文矩阵为：$\begin{bmatrix}\text{wynoo}\\ \text{reuha}\\ \text{inagv}\\ \text{esatt}\end{bmatrix}$。

(2) 置换,得输出矩阵 $\begin{bmatrix}\text{nowyo}\\ \text{uareh}\\ \text{aving}\\ \text{atest}\end{bmatrix}$。

置换过程为:$\begin{bmatrix}\text{wynoo}\\ \text{reuha}\\ \text{inagv}\\ \text{esatt}\end{bmatrix}\begin{bmatrix}12345\\ ?????\end{bmatrix}\boldsymbol{D}\begin{bmatrix}?????\\ ?????\\ ?????\\ ?????\end{bmatrix}$。

在本题解题过程中,我们不知道具体的解密矩阵,因此需要尝试各种排列找到正确的解密矩阵。$n=5$ 时可能的解密矩阵总数为 5! =120,从中找出正确的解密矩阵对人脑而言已具有一定难度,需要使用计算机辅助选择。正确结果为:

$$\begin{bmatrix}\text{wynoo}\\ \text{reuha}\\ \text{inagv}\\ \text{esatt}\end{bmatrix}\begin{bmatrix}12345\\ 34152\end{bmatrix}\boldsymbol{D}\begin{bmatrix}\text{nowyo}\\ \text{uareh}\\ \text{aving}\\ \text{atest}\end{bmatrix}$$

置换密码位置变,字符不变;替代密码字符变,位置不变。组合置换密码和替代密码就可实现位置、字符同时改变,增加密码解密难度。

【例 2-15】 解密困在栅栏中的凯撒 c_1=av\EnZZpZ)ZgbZpo/ai++x,求 p_1。

解:依题意可知密文使用了凯撒密码和栅栏密码加密技术。

∴首先进行栅栏密码解密。

∵c_1 长度为 22=2×11,∴需要分 2 栏和 11 栏两种情况分别讨论。

(1) 2 栏解密时,密文由中心分开为每栏 11 个字符,上下两栏:$\begin{pmatrix}\text{av\EnZZpZ)Z}\\ \text{gbZpo/ai++x}\end{pmatrix}$。

先上后下逐一合并得:agvb\ZEpnoZ/ZapiZ+)+Zx。

然后进行凯撒密码暴力破解:

Key=1,p_1=bhwc…;

Key=2,p_1=cixd…;

Key=3,p_1=djye…;

Key=4,p_1=ekzf…;

Key=5,p_1=flag{_Just_4_fun_0.0_},解密结束。

(2) 11 栏解密时分析同上,结果不符合题意。

一次置换和一次替代的组合称为一轮加密/解密。一轮加密后明文中字符和字符的位置均发生了变化,因而其解密难度增加。所以多轮加密是增强密码系统保密性的方法之一,解密运算量的增加,导致需要引入计算机进行计算。

【例 2-16】 p_1=now we are having a test,分析 3 轮加密后的解密难度。

解:∵一次矩阵置换后为:wne ow haa regvai n ttxe

一次凯撒替代后为:zqh rz kdd uhjydl q wwah

∴一轮解密强度:25×120=3000,数量级:千,一次成功概率:1/3000。

二轮:3000 × 3000=9 000 000,数量级:百万,一次成功概率:1/9 000 000。

三轮：9 000 000×600＝5 400 000 000，数量级：十亿，一次成功概率：1/5 400 000 000。

综上所述，置换密码使用矩阵变换运算实现，具有位置变，字符不变的特征。加密、解密规则相似度非常高，加密矩阵和解密矩阵可以相互推出。

2.1.3　密码学语言

一个密码系统(Crypto system)由算法以及所有可能的明文、密文和密钥组成，定义为一个五元组(P,C,E,D,K)，对应的加密方案称为密码体制。

明文(plain text)是密码系统可以处理的输入数据，用小写 p 表示。明文的有限集合构成明文空间 P，$p\in P$。例如 P＝{"this is a book", "i am a student"}，p_1＝this is a book，p_2＝i am a student。

严格地讲，明文是一串二进制数，代表字符串、文本文件、图形图像、数字化的语音流或者数字化的视频图像。本书中明文多以字符串形式出现，以方便读者直接识别其明确含义。实际的明文具有多种形式，如图 2-1(a)为图像形式的明文。

密文(cipher text)是明文被加密处理后的形式，特征是没有明确含义或其含义具有二义性，用小写 c 表示。密文的有限集合构成密文空间 C，$c\in C$。例如 C＝{"wklv lv d errn","xlmw mw e fsso","k co c uvwfgpv","m eq e wxyhirx"}，c_1＝wklv lv d errn，c_2＝xlmw mw e fsso，c_3＝k co c uvwfgpv，c_4＝m eq e wxyhirx。图 2-1(b)为图 2-1(a)图像对应的密文。

(a) 明文　　(b) 密文

图 2-1　图像加密

将明文变换为密文的变换规则(函数)，称为加密算法 E(Encrypt)。相应的变换过程称为加密，用数学公式表示为：$c=E(p)$，表示 E 作用于 p 得到 c。例如规定古典凯撒加密算法 E＝{将当前字母用字母表中其后的第 3 个字母替代，x、y、z 依次用 a、b、c 替代}，则加密 p_1 为 c_1 表示为：

$$c_1=\text{wklv lv d errn}=E(\text{this is a book})=E(p_1)$$

将密文恢复为明文的变换规则(函数)，称之解密算法 D(Deciphering)。相应的变换过程称为解密，用公式表示 $p=D(c)$，表示 D 作用于 c 产生 p。例如规定古典凯撒解密算法 D＝{将当前字母用字母表中其前的第 3 个字母替代}，则解密 c_1 为 p_1 表示为：

$$p_1=\text{this is a book}=D(\text{wklv lv d errn})=D(c_1)$$

加密、解密互为逆过程，对于有实用意义的密码系统而言，总是要求它满足：$D(E(p))=p$，即用加密算法得到的密文总是能用对应的解密算法恢复出原始的明文。例如：

$$D(E(p_1))=D(E(\text{this is a book}))=D(c_1)=D(\text{wklv lv d errn})=\text{this is a book}=p_1$$

密钥(key)是参与数据变换的参数，用小写 k 表示。一切可能的密钥构成的有限集，称

为密钥空间 K,$k \in K$。例如古典凯撒密码的密钥空间 $K=\{3\}$,$k=3$。

综上所述,可定义古典凯撒密码系统为:

({"this is a book", "i am a student"},{"wklv lv d errn","xlmw mw e fsso","k co c uvwfgpv","m eq e wxyhirx","o gs g yzajktz" },{将当前字母用字母表中其后的第 3 个字母替代,x、y、z 依次用 a、b、c 替代},{将当前字母用字母表中其前的第 3 个字母替代。a、b、c 依次用 x、y、z 替代},{3})

从数学的角度来讲,一个密码系统就是一组映射,它在密钥的控制下将明文空间中的每一个元素映射到密文空间上的某个元素。这组映射由密码方案确定,具体使用哪一个映射由密钥决定。图 2-2 示意了凯撒密码系统的映射关系。

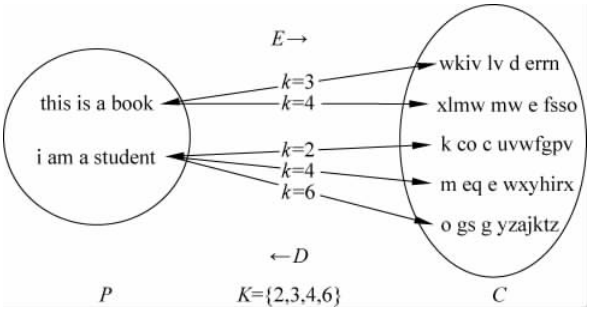

图 2-2 凯撒密码系统

定义字母和自然数存在如表 2-5 所示的一一对应关系,则凯撒加密算法可表示为函数形式:$c=f(p,k)=(p+k) \bmod 26$。凯撒解密算法可表示为函数形式:$c=f(p,k)=(p-k) \bmod 26$。

表 2-5 字母数字对照表

英文字母	a b c d e f g h i j k l m n o… v w x y z
数字	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14… 21 22 23 24 25

【例 2-17】 已知 c_1=xlmw mw e fsso,求 p_1。

解:∵英文句子中单个字母是 a 的可能性最大。

∴假定 e 为 a,e 为第 5 个字母,a 为第 1 个字母,k=e−a=4−0=4,k=4。

∴p_1=this is a book。

加解密算法可分为基于算法保密的算法和基于密钥保密的算法两类。

(1) 基于算法保密的加解密算法。这类算法的保密性取决于保持算法的秘密,使用范围有限,也称为受限制的算法,多用于安全性较高的军事、涉密部门,不适合民用。这种算法不可能进行质量控制或标准化,大的或经常变换的用户组织不能使用这种算法,因为如果有一个用户离开这个组织,其他的用户就必须更换另外不同的算法。如果有人无意暴露了这个秘密,所有的人都必须改变他们的算法。

(2) 基于密钥保密的加解密算法的安全性都基于密钥的安全性,而不是基于算法细节的安全性。这就意味着算法可以公开,也可以被分析,可以大量生产使用算法的产品,即使偷窃者知道用户的算法也没有关系。如果他不知道用户使用的具体密钥,他就不可能阅读

用户的消息。

这类算法是公开的，因而广泛使用于军民两用领域，其对信息安全传输的保护依赖于密钥空间的大小，通过扩大密码空间、增加算法运算量等方法增加解密难度，实现信息保密。

基于密钥保密的加解密算法可进一步分为对称密钥算法和公开密钥算法两类。前者的代表是 DES，后者的代表是 RSA。

密码学的首要目的是隐藏信息的涵义，实现数据通信的机密性。著名的密码学者 Ron Rivest 解释道："密码学是关于如何在敌人存在的环境中通信。"

在如图 2-3 所示模型中，还存在一个密码攻击者或破译者可从普通信道上拦截到的密文 c，其工作目标就是要在不知道密钥 k 的情况下，试图从密文 c 恢复出明文 p 或密钥 k。

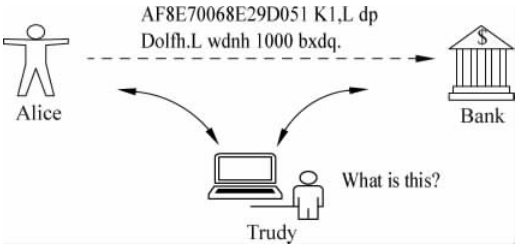

图 2-3　加密 AB 模型

一个安全的密码系统应该满足：

(1) 非法截收者很难从密文 c 中推断出明文 p。

(2) 加密和解密算法应该相当简便，而且适用于所有密钥空间。

(3) 密码系统的保密强度只依赖于密钥。

(4) 合法接收者能够检验和证实消息的完整性和真实性。

(5) 消息发送者无法否认其所发出的消息，同时也不能伪造别人的合法消息。

(6) 必要时可由仲裁机构进行公断。

2.1.4　凯撒密码解密

密码学包括密码编码学和密码分析学。密码体制的设计是密码编码学的主要内容，密码体制的破译是密码分析学的主要内容。密码编码技术和密码分析技术是相互依存、相互支持、密不可分的两个方面。

密码编码学的主要目的是保持明文(或密钥，或明文和密钥)的秘密以防止偷听者(对手、攻击者、敌人)知晓。这里假设偷听者完全能够截获收发者之间的通信。

密码分析学是在不知道密钥的情况下，恢复出明文的科学。成功的密码分析能恢复出消息的明文或密钥。密码分析也可以发现密码体制的弱点，最终得到上述结果(密钥通过非密码分析方式的丢失叫做泄露)。

常用的密码分析攻击有以下几种，当然，每一类都假设密码分析者知道所用的加密算法的全部知识：

(1) 唯密文攻击(Cipher Text-Only Attack)。密码分析者有一些消息的密文，这些消息都用同一加密算法加密。密码分析者的任务是恢复尽可能多的明文，或者最好是能推算出加密消息的密钥来，以便采用相同的密钥解密出其他被加密的消息。

已知：$C_1=E_k(P_1),C_2=E_k(P_2),\cdots,C_i=E_k(P_i)$

推导出：$P_1,P_2,\cdots,P_i$，或者密钥 k，或者找出一个算法从 $C_{i+1}=E_k(P_{i+1})$推出 P_{i+1}。

(2) 已知明文攻击(Known Plain Text Attack)。密码分析者不仅可得到一些消息的密文，而且也知道这些消息的明文。分析者的任务就是用加密信息推出用来加密的密钥或导出一个算法，此算法可以对用同一密钥加密的任何新的消息进行解密。

已知：$P_1,C_1=E_k(P_1),P_2,C_2=E_k(P_2),\cdots,P_i,C_i=E_k(P_i)$

推导出：密钥 k，或从 $C_{i+1}=E_k(P_{i+1})$推出 P_{i+1}的算法。

(3) 选择明文攻击(Chosen Plain Text Attack)。分析者不仅可得到一些消息的密文和相应的明文，而且可选择被加密的明文。这比已知明文攻击更有效。因为密码分析者能选择特定的明文块去加密，那些块可能产生更多关于密钥的信息，分析者的任务是推出用来加密消息的密钥或导出一个算法，此算法可以对用同一密钥加密的任何新消息进行解密。

已知：

$$P_1,C_1=E_k(P_1),P_2,C_2=E_k(P_2),\cdots,P_i,C_i=E_k(P_i)$$

其中 $P_1,P_2,\cdots,P_i$ 只可由密码分析者选择。

推导出：密钥 k，或从 $C_{i+1}=E_k(P_{i+1})$推出 P_{i+1}的算法。

(4) 自适应选择明文攻击(Adaptive Chosen Plain Text Attack)。这是选择明文攻击的特殊情况。密码分析者不仅能选择被加密的明文，而且也能基于以前加密的结果修正这个选择。在选择明文攻击中，密码分析者可以选择一大块被加了密的明文。而在自适应选择密文攻击中，攻击者可选取较小的明文块，然后再基于第一块的结果选择另一明文块，以此类推。

如果密码分析者可以仅由密文推出明文或密钥，或者可以由明文和密文推出密钥，那么就称该密码系统是可破译的。相反地，则称该密码系统不可破译。

衡量密码分析效果的指标有两个：一是准确率；二是时间效率。

下面以凯撒密码为例，进行密码分析。

首先无密钥凯撒密码是可以破解的，因为凯撒密码只有25种密钥，所以最直接的方法就是对这25种可能性逐一检测，这就是我们所说的暴力破解法。具体实例见例2-6。

(1) 初级解密。初级解密是指密文用空格分隔单词，且有单个字母。此时单字母是突破口，英文句子中出现单字母是a的概率非常大，因此可据此将密文中单字母假定为a从而推出密钥实现解密。具体实例见例2-17。

因此实际使用时，需要将明文中的空格去掉后加密生成密文，增加密文破解难度。

(2) 中级解密。中级解密是指密文允许出现空格，但没有单个字母，使用暴力破解方法列出所有25种可能，人工识别正确结果。具体实例见例2-6。

从多个可能明文中选择出正确明文的原则是：统计明文中出现单词个数的多少，以个数多的明文为正确答案。理想状态下，正确答案是一个具有确定含义的句子。

(3) 高级解密。在中级破解的基础上，提高计算机程序的智能化程度，方法是加入常用词词典或完整词典，通过字符串与词典中单词匹配的个数，最终选中正确的一个结果或缩小正确的结果为2～3个句子，减少人工干预，实现计算机智能选择正确结果。

例如增加常用词词典 dict=["a","i","to","be","am","we","are","you","the"]，选取可能明文的前 n 个字符串与 dict 比较，逐步缩小正确明文的范围。

具体密码分析请在实验1中自己实现。

2.2　DES 对称密码

对称密钥算法的特点是加解密算法使用的密钥相同(例如凯撒密码)或加密密钥能够从解密密钥中推算出来,反过来也成立(例如矩阵置换密码),所以也叫做单密钥算法。

对称密钥算法的安全性依赖于密钥,泄露密钥就意味着任何人都能对消息进行加解密。为了强调算法依赖于密钥,用 K 作为下标表示,这样加/解密函数就变成:

$$E_K(P) = C, \quad D_K(C) = P, \quad D_K(E_K(P)) = E_K(D_K(P)) = P$$

根据一次能处理数据的位数,对称算法可细分为两类:

(1) 一次只对明文中的单个位(有时对字节)运算的算法称为序列算法或序列密码。

(2) 对明文的一组位进行运算,这些位组称为分组,相应的算法称为分组算法或分组密码。现代计算机密码算法的典型分组长度为 64 位——这个长度既考虑到分析破译密码的难度,又考虑到使用的方便性。后来,随着破译能力的提高,分组长度又增加到 128 位或更长。

DES(Data Encryption Standard,数据加密标准)算法是在美国 NSA 资助下由 IBM 公司开发的一种对称密码算法,其初衷是为政府非机密的敏感信息提供较强的加密保护。它是美国政府担保的第一种加密算法,并在 1977 年被正式作为美国联邦信息处理标准。

DES 主要提供给非军事性质的联邦政府机构和私营部门使用,并迅速成为名声最大、使用最广的商用密码算法。

在 1972 年和 1974 年美国国家标准局两次向公众发出了征求加密算法的公告。对加密算法提出了以下几点要求:

(1) 提供高质量的数据保护,防止数据未经授权的泄露和未被察觉的修改;

(2) 具有相当高的复杂性,使得破译的开销超过可能获得的利益,同时又要便于理解和掌握;

(3) DES 密码体制的安全性应该不依赖于算法的保密,其安全性仅以加密密钥的保密为基础;

(4) 实现经济,运行有效,并且适用于多种完全不同的应用;

(5) 实现算法的电子器件必须经济、运行有效;

(6) 必须能够验证,允许出口。

2.2.1　DES 算法

DES 算法是一个分组加密算法,典型的 DES 以 64 位为分组对数据加密,加密和解密用的是同一个算法。它的密钥长度是 56 位(因为每个字节的第 8 位都被用作奇偶校验以保证密钥本身正确,不会在密钥分发过程中出错),密钥可以是任意的 56 位的数,而且可以在任意时候改变。其中有极少数被认为是易破解的弱密钥,但是很容易避开它们不用。所以 DES 算法保密性依赖于密钥。

简单地说,算法只不过是加密的两个基本技术——混乱和扩散的组合。

DES 组建分组是这些技术的一个组合(先代替后置换),它基于密钥作用于明文,这就

是众所周知的轮(round)。DES有16轮,这意味着要在明文分组上实施16次相同的组合技术。此算法只使用了标准的算术和逻辑运算,而其作用的数也最多只有64位,因而运算速度非常快。

DES算法的入口参数有3个:Key、Data、Mode。其中Key为8个字节共64位,是DES算法的工作密钥;Data也为8个字节64位,是要被加密或被解密的数据;Mode为DES的工作方式,有两种:加密或解密。

DES算法是这样工作的:如Mode为加密,则用Key把数据Data进行加密,生成Data的密码形式(64位)作为DES的输出结果;如Mode为解密,则用Key把密码形式的数据Data解密,还原为Data的明码形式(64位)作为DES的输出结果。

在通信网络的两端,双方约定一致的Key,在通信的源点用Key对核心数据进行DES加密,然后以密码形式在公共通信网(如电话网)中传输到通信网络的终点,数据到达目的地后,用同样的Key对密码数据进行解密,便再现了明码形式的核心数据。这样,便保证了核心数据在公共通信网中传输的安全性和可靠性。通过定期在通信网络的源端和目的端同时更换用新的Key,便能进一步提高数据的保密性,这是现在金融交易网络的流行做法。

DES算法实现加密需要3个步骤:

第一步,变换明文。对给定的64位比特的明文x,首先通过一个置换表IP表来重新排列x,从而构造出64位比特的x_0,$x_0=\mathrm{IP}(x)=L_0R_0$,其中$L_0$表示$x_0$的前32比特,$R_0$表示$x_0$的后32位。

第二步,按照规则迭代。规则为$L_i=R_{i-1}$

$$R_i = L_{i-1} \oplus f(R_{i-1},K_i) \quad (i=1,2,3,\cdots,16)$$

经过第一步变换已经得到L_0和R_0的值,其中符号⊕表示的数学运算是异或,f表示一种置换,由S盒置换构成,K_i是一些由密钥编排函数产生的比特块。f和K_i将在后面介绍。

第三步,对$L_{16}R_{16}$利用IP^{-1}作逆置换,就得到了密文y。加密过程如图2-4所示,f函数的处理流程如图2-5所示。

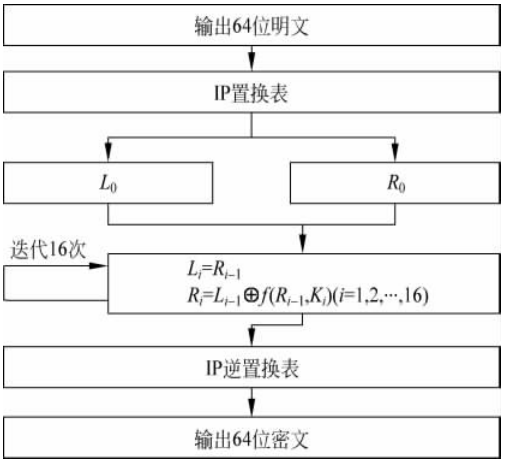

图2-4 DES加密系统

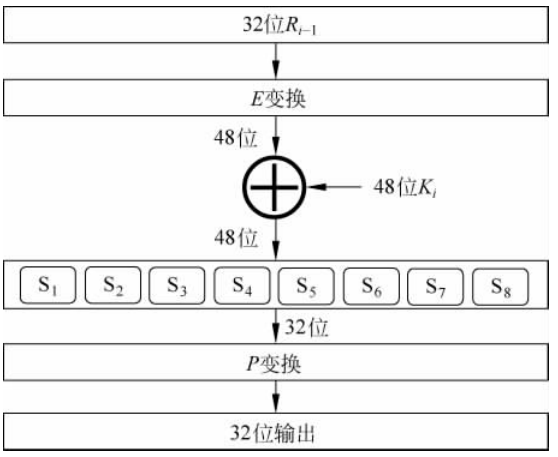

图 2-5　函数 f 的处理流程

1. DES 加密的 4 个关键点

从图 2-4 中可以看出，DES 加密需要 4 个关键点：IP 转换表和 IP^{-1} 逆转换表、函数 f 子密钥 K_i 和 S 盒的工作原理。

1）IP 置换表和 IP^{-1} 逆置换表

输入的 64 位数据按 IP 置换表进行重新组合，并把输出分为 L_0、R_0 两部分，每部分各长 32 位，其 IP 置换表如表 2-6 所示。

表 2-6　IP 置换表

58	50	12	34	26	18	10	2	60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6	64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1	59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5	63	55	47	39	31	23	35	7

将输入 64 位比特的第 58 位换到第 1 位，第 50 位换到第 2 位，以此类推，最后一位是原来的第 7 位。L_0、R_0 则是换位输出后的两部分，L_0 是输出的左 32 位，R_0 是右 32 位。比如：置换前的输入值为 $D_1D_2D_3\cdots D_{64}$，则经过初始置换后的结果为：$L_0=D_{58}D_{50}\ldots D_8$，$R_0=D_{57}D_{49}\ldots D_7$。

经过 16 次迭代运算后。得到 L_{16}、R_{16}，将此作为输入，进行逆置换，即得到密文输出。逆置换正好是初始置换的逆运算，例如第 1 位经过初始置换后，处于第 40 位，而通过逆置换 IP^{-1}，又将第 40 位换回到第 1 位，其逆置换 IP^{-1} 规则如表 2-7 所示。

表 2-7　逆置换表 IP^{-1}

40	8	48	16	56	24	64	32	39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30	37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28	35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26	33	1	41	9	49	17	57	25

2) 函数 f

函数 f 有两个输入：32 位的 R_{i-1} 和 48 位的 K_i，f 函数的处理流程如图 2-5 所示。

E 变换的算法是从 R_{i-1} 的 32 位中选取某些位，构成 48 位。即 E 将 32 比特扩展变换为 48 位，变换规则根据 E 位选择表，如表 2-8 所示。

表 2-8 E 位选择表

32	1	2	3	4	5	4	5	6	7	8	9	8	9	10	11
12	13	12	13	14	15	16	17	16	17	18	19	20	21	20	21
22	23	24	25	24	25	26	27	28	29	28	29	30	31	32	1

K_i 是由密钥产生的 48 位比特串，具体的算法下面介绍。将 E 的选位结果与 K_i 作异或操作，得到一个 48 位输出。分成 8 组，每组 6 位，作为 8 个 S 盒的输入。

每个 S 盒输出 4 位，共 32 位，S 盒的工作原理将在第 4 步介绍。S 盒的输出作为 P 变换的输入，P 的功能是对输入进行置换，P 换位表如表 2-9 所示。

表 2-9 P 换位表

16	7	20	21	29	12	28	17	1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9	19	13	30	6	22	11	4	25

3) 子密钥 K_i

假设密钥为 K，长度为 64 位，但是其中第 8、16、24、32、40、48、64 用作奇偶校验位，实际上密钥长度为 56 位。K 的下标 i 的取值范围是 1～16，用 16 轮来构造。构造过程如图 2-6 所示。

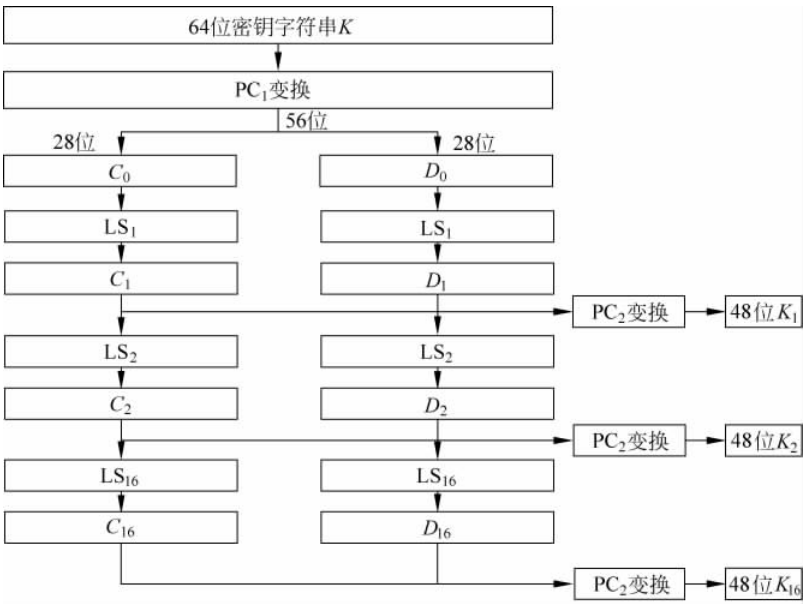

图 2-6 子密钥生成

首先，对于给定的密钥 K，应用 PC_1 变换进行选位，选定后的结果是 56 位，设其前 28 位为 C_0，后 28 位为 D_0。PC_1 选位如表 2-10 所示。

表 2-10　PC_1 选位表

57	49	41	33	25	17	9	1	58	50	42	34	26	18
10	2	59	51	43	35	27	19	11	3	60	52	44	36
63	55	47	39	31	23	15	7	62	54	46	38	30	22
14	6	61	53	45	37	29	21	13	5	28	20	12	4

第一轮，对 C_0 作左移 LS_1 得到 C_1，对 D_0 作左移 LS_1 得到 D_1，对 C_1D_1 应用 PC_2 进行选位，得到 K_1。其中 LS_1 是左移的位数，如表 2-11 所示。

表 2-11　LS 移位表

1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

表 2-11 中的第一列是 LS_1，第二列是 LS_2，以此类推。左移的原理是所有二进位向左移动，原来最右边的比特位移动到最左边。其中 PC_2 如表 2-12 所示。

表 2-12　PC_2 选位表

14	17	11	24	1	5	3	28	15	6	21	10
23	19	12	4	26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40	51	45	33	48
44	49	39	56	34	53	46	42	50	36	29	32

第二轮：对 C_1，D_1 作左移 LS_2 得到 C_2 和 D_2，进一步对 C_2D_2 应用 PC_2 进行选位，得到 K_2。如此继续，分别得到 K_3、K_4、……、K_{16}。

4）S 盒的工作原理

S 盒以 6 位作为输入，而以 4 位作为输出，现在以 S_1 为例说明其过程。假设输入为 $a=a_1a_2a_3a_4a_5a_6$，则 $a_2a_3a_4a_5$ 所代表的数是 0～15 的一个数，记为：$k=a_2a_3a_4a_5$；由 a_1a_6 所代表的数是 0～3 的一个数，记为 $h=a_1a_6$。在 S_1 的 h 行，k 列找到一个数 B，B 在 0～15 之间，它可以用 4 位二进制表示，为 $B=b_1b_2b_3b_4$，这就是 S_1 的输出。S 盒是由 8 张数据表组成（这里不详细给出）。

例如 $a=110011$，$h=a_1a_6=(11)_2=(3)_{10}$对应于第三行，$k=a_2a_3a_4a_5=(1001)_2=(9)_{10}$对应于第 9 列，$S$ 盒的第 3 行第 9 列的数是 14（记住行、列的记数从 0 开始而不是从 1 开始），则值 $b=b_1b_2b_3b_4=1110$，1110 将代替 110011。

2. DES 加密具有雪崩效应

在密码学中，雪崩效应（Avalanche effect）指当输入发生最微小的改变（如反转一个二进制位）时，也会导致输出的剧变（如输出中一半的二进制位发生反转）。在高品质的块密码中，无论密钥或明文的任何细微变化都应当引起密文的剧烈改变；否则，如果变化太小，就可能找到一种方法减小有待搜索的明文和密文的空间的大小。

严格雪崩准则（Strict Avalanche Criterion，SAC）是雪崩效应的形式化。它指出，当任何一个输入位被反转时，输出中的每一位均有 50％的概率发生变化。严格雪崩准则建立于密码学的完全性概念上，由 Webster 和 Tavares 在 1985 年提出。

DES加密具有雪崩效应:

(1) 用同样密钥加密只差一位的两个明文。例如 Key="11111111",p_1=hellodes,p_2=hellodet。

c_1=0101110110011011000100111100110110100110001011110110100000101110,

c_2=0010000010011000010101000011000000110001100110110100011110101000。

两者的汉明距离为36,说明64位中有36位不同。也就是说,用同样密钥加密只差1位的两个明文,密文相差36位,变化比率为56%。

(2) 用只差一比特的两个密钥加密同样明文。例如 key_1=12345678,key_2=12345677,p_1=hellodes。

c_1=0011001011001010110010100111100110010001001010101001101111101110

c_2=0110000011011001111110000000000101111001011011000010011001000011

两者的汉明距离为31,说明用只差一比特的两个密钥加密同样的明文,密文相差31位,变化比率为48%。

3. DES解密

DES的算法是对称的,既可用于加密又可用于解密,具有一个非常有用的性质:加密和解密可使用相同的算法。

DES使得能够用相同的函数来加密或解密每个分组,二者唯一不同之处是密钥的次序相反,算法本身并没有任何变化。这就是说,如果各轮的加密密钥分别是 $K_1K_2K_3,\cdots,K_{16}$,那么解密密钥就是 $K_{16},K_{15},K_{14},\cdots,K_1$。为各轮产生密钥的算法也是循环的。密钥向右移动,每次移动的个数为0,1,2,2,2,2,2,2,1,2,2,2,2,2,2,1。这使得DES硬件加密、解密可以使用同一硬件设备,极大地降低了硬件制造成本。

2.2.2 DES算法的优缺点

DES算法的优点是加解密速度快,具有比较高的安全性,目前还没有发现这种算法在设计上的破绽,在理论上DES算法仍然是不可解的。实践中,对于DES算法都是通过穷举密钥的方式破解的,56位长的密钥的穷举空间为 2^{56},这意味着如果一台计算机的速度是每一秒钟检测一百万个密钥,则它搜索完全部密钥就需要将近2285年的时间,可见这是难以实现的。但随着计算能力的增强,DES解密所需时间越来越短,安全性日益受到挑战。

1997年1月28日,美国RSA数据安全公司开展了一项名为秘密密钥挑战(Secret Key Challenge)的竞赛,悬赏10 000美元破解一段DES密文。RSA发起此次活动旨在测试DES算法的安全强度。美国科罗拉多州的程序员Rocke Verser设计了一个通过Internet分段运行的密钥搜索程序,采用穷举密钥的方式对DES进行破解。搜索行动被称为DESCHALL,成千上万的志愿者加入到计划中,每个加入的志愿者会被分配一部分密钥空间进行测试。破解活动从1997年3月13日开始,在1997年6月17日,也就是破解活动的第96天,美国盐湖城一个公司职员Michael Sanders成功地找到了密钥,解密出了明文:"Strong cryptography makes the world a safer place(高强度的密码技术使世界更安全)"。此次破解一方面说明DES算法绝非坚不可摧,另一方面也显示了Internet上分布式计算的强大能力。

1998年电子前线基金会(Electronic Frontier Foundation,EFF)为了告诫政府DES的保密性只是在一定范围内,花费了25万美元制造了一台专用于暴力破解DES算法的计算

机，这台计算机被命名为 DES 深层破解(Deep Crack)。1998 年 7 月 13 日早上 9 点，计算机开始破解了一段 DES 密文，在 7 月 15 日下午 5 点成功找到 DES 密钥，一共只花费来了 56 小时。1999 年，EFF 用 22 小时 15 分就完成了一段 DES 密文的破解工作。

DES 算法的缺点是：

(1) DES 算法中只用到 64 位密钥中的 56 位，而第 8,16,24,…,64 位 8 个位并未参与 DES 算法，这一点提出了一个应用上的要求，即 DES 的安全性基于除了 8,16,24,…,64 位外的其余 56 位的组合变化 2^{56} 才得以保证的。因此，在实际应用中，应避开使用第 8,16,24,…,64 位作为有效数据位，而使用其他的 56 位作为有效数据位，才能保证 DES 算法安全可靠地发挥作用。如果不了解这一点，把密钥 Key 的 8,16,24,…,64 位作为有效数据使用，将不能保证 DES 加密数据的安全性，对应用 DES 来达到保密作用的系统产生数据破译的危险，这正是 DES 算法在应用上的误区，留下了被人攻击、破译的极大隐患。

(2) DES 的唯一密码学缺点就是密钥长度较短。解决密钥长度问题的办法之一是采用三重 DES。

三重 DES 方法需要执行 3 次常规的 DES 加密步骤，但最常用的三重 DES 算法中仅仅用两个 56 位 DES 密钥。设这两个密钥为 K_1 和 K_2，其算法的步骤如下：

- 用密钥 K_1 进行 DES 加密；
- 对上面的结果使用密钥 K_2 进行 DES 解密(实为加密，因为密钥不同)；
- 对上一步的结果使用 K_1 进行 DES 加密。

这个过程称为 EDE，因为它是由加密—解密—加密步骤组成的。

在 EDE 中，中间步骤是解密，所以，可以使用两个密钥 K_1 和 K_2 执行常规的 DES 加密、解密实现三重 DES。

三重 DES 的缺点是时间开销较大，三重 DES 的时间是 DES 算法的 3 倍。但从另一方面看，三重 DES 的 112 位密钥长度在可以预见的将来可认为是合适的。

(3) 存在密钥分发难题。对称密钥算法的优点是加解密速度快，因而适合用作大量数据的加解密；缺点是发送者和接收者在安全通信之前必须先协商一个密钥，存在密码分发问题，信息泄密后不能区分责任者。这是由于加密密码和解密密钥相同，且发送方 A、接收方 B 拥有同一个密码。所以当一个密文被破解后，无法确定是 A 还是 B 泄露了密钥。

对称密钥分发问题具体包括：

- 密钥分发效率问题。一个 n 个人构成的组织，为了保证两两间能够相互保密通信，需要 $n\times(n-1)/2$ 个密钥，即密码空间为 n^2 量级。显然这会导致密码空间爆炸，不利于普遍推广使用。
- 密钥分发路径问题。线下分发(如使用 U 盾)需要当面交付，时效性较差；线上分发，第三方能够轻易获得对称密码从而使 DES 加密失去意义。

为了彻底解决以上问题，产生了以 RSA 为代表的公钥密码。

2.3　RSA 公钥密码

1976 年，Diffie 和 Hellman 在“密码学的新方向”一文中提出了公钥密码的思想：密码系统中的加密密钥和解密密钥可以不同。由于不能容易地通过加密密钥和密文来求得解密

密钥或明文,所以可以公开这种系统的加密算法和加密密钥,此时,用户只要保管好自己的解密密钥即可。

非对称密码体制的优点是可以适应网络的开放性要求,密钥管理相对简单,尤其是可以实现数字签名功能。与对称密码算法相比,非对称密码算法一般比较复杂,加、解密速度慢。

非对称密码体制问世以来,密码学家们提出了许多种非对称密码算法,它们的安全性大都基于复杂的数学难题。根据所基于的数学难题来分类,只有大数因子分解系统(代表算法RSA)、离散对数系统(代表算法ElGamal)、椭圆曲线密码系统(Elliptic Curve Cryptography,ECC)3类系统目前被认为是安全和有效的。

RSA算法是迄今为止最容易理解和实现的公开密钥算法(Public Key Algorithm),是第一个实用的公钥加密方案,同时也是历史上最成功的,直到现在仍有广泛应用的公钥加密体制,已经受住了多年深入的攻击,其理论基础是一种特殊的可逆模幂运算,其安全性基于分解大整数的困难性,但安全性一直未能得到理论上的证明。

RSA算法的基本原理是:由一个密钥进行加密的信息内容,只能由与之配对的另一个密钥才能进行解密。公钥可以广泛地发给与自己有关的通信者,私钥则需要十分安全地存放起来。规定加密密钥叫做公开密钥(public key),解密密钥叫做私人密钥(private key)。

2.3.1 模运算

定义:给定一个正整数 p,任意一个整数 n,一定存在等式 $n=k\times p+r$,其中 k、r 是整数,且 $0\leqslant r<p$,称 k 为 n 除以 p 的商,r 为 n 除以 p 的余数。

p 对于正整数和整数 a、b,定义如下运算:

(1) 取模运算:$a \bmod p$,表示 a 除以 p 的余数。

模运算是整数运算,有一个整数 a,以 p 为模做模运算,即 $a \bmod p$。运算是让 a 去被 p 整除,只取所得的余数作为结果,这就叫做模运算。例如 $10 \bmod 3=1$;$26 \bmod 6=2$ 等。

(2) 同余式:正整数 a、b 对 p 取模,它们的余数相同,记作 $a\equiv b \bmod p$ 或者 $a\equiv b \pmod p$。公式中≡符号的左边必须和符号右边同余,也就是两边模运算结果相同。

(3) 模 p 加法:$(a+b) \bmod p$,结果是 $a+b$ 算术和除以 p 的余数,也就是说,$(a+b)=k\times p+r$,则 $(a+b) \bmod p=r$。

(4) 模 p 减法:$(a-b) \bmod p$,其结果是 $a-b$ 算术差除以 p 的余数。

(5) 模 p 乘法:$(a\times b) \bmod p$,其结果是 $a\times b$ 算术乘法除以 p 的余数。

(6) 模指数运算就是先做指数运算,取其结果再做模运算。如 $5^3 \pmod 7=125 \bmod 7=6$。

(7) $n \bmod p$ 得到结果的正负由被除数 n 决定,与 p 无关。例如,$7 \bmod 4=3$,$-7 \bmod 4=-3$(商为-1)。

注意 对于任何同号的两个整数,其取余结果没有争议,所有语言的运算原则都是使商尽可能小;对于异号的两个整数,C++/Java语言的原则是使商尽可能大,很多新型语言和网页计算器的原则是使商尽可能小。本书使用前者。

基本性质

(1) 若 $p|(a-b)$,则 $a\equiv b \pmod p$。例如 $11\equiv 4 \pmod 7$,$18\equiv 4 \pmod 7$。

(2) $(a \bmod p)=(b \bmod p)$意味 $a\equiv b \pmod p$。

(3) 对称性:$a\equiv b \pmod p$等价于 $b\equiv a \pmod p$。

(4) 传递性：若 $a \equiv b \pmod{p}$且 $b \equiv c \pmod{p}$，则 $a \equiv c \pmod{p}$。

运算规则

模运算与基本四则运算有些相似，但是除法例外。其规则如下：

$(a + b) \bmod p = (a \bmod p + b \bmod p) \bmod p$

$(a - b) \bmod p = (a \bmod p - b \bmod p) \bmod p$

$(a \times b) \bmod p = (a \bmod p \times b \bmod p) \bmod p$

$(a^b) \bmod p = ((a \bmod p)^b) \bmod p$

结合律

$((a+b) \bmod p + c) \bmod p = (a + (b+c) \bmod p) \bmod p$

$((a \times b) \bmod p \times c) \bmod p = (a \times b \times c) \bmod p // (a \bmod p \times b) \bmod p = (a \times b) \bmod p$

交换律

$(a + b) \bmod p = (b+a) \bmod p$

$(a \times b) \bmod p = (b \times a) \bmod p$

分配律

$((a + b) \bmod p \times c) \bmod p = ((a \times c) \bmod p + (b \times c) \bmod p) \bmod p$

重要定理

若 $a \equiv b \pmod{p}$，则对于任意的 c，都有$(a+c) \equiv (b+c) \pmod{p}$

若 $a \equiv b \pmod{p}$，则对于任意的 c，都有$(a \times c) \equiv (b \times c) \pmod{p}$

若 $a \equiv b \pmod{p}$，$c \equiv d \pmod{p}$，则$(a+c) \equiv (b+d) \pmod{p}$，$(a-c) \equiv (b-d) \pmod{p}$

$(a \times c) \equiv (b \times d) \pmod{p}$

等价类

模运算是一种哈希函数，余数不唯一。对每个给定的模数 m 和整数 a，可能同时存在无限多个有效的余数，这些整数的集合称为等价类。

对 $a=12$，$m=9$ 的而言：

模数为 9，余数为 0 的等价类为：{…，−27，−18，−9，0，9，18，27，…}；

模数为 9，余数为 1 的等价类为：{…，−26，−17，−8，1，10，19，28，…}；

模数为 9，余数为 2 的等价类为：{…，−25，−16，−7，2，11，20，29，…}；

模数为 9，余数为 3 的等价类为：{…，−24，−15，−6，3，12，21，30，…}；

……

模数为 9，余数为 8 的等价类：{…，−19，−10，−1，8，17，26，35，…}。

等价类中所有成员的行为等价：对于一个给定模数 m，选择等价类中任何一个元素用于计算的结果都是一样的。因此在固定模数的计算中(这是密码学中最常见的情况)，我们可以选择等价类中最易于计算的一个元素。

例如计算 $3^8 \bmod 7$，直接计算为 $3^8 = 6561 \equiv 2 \bmod 7$，因为 $6561 = 937 \times 7 + 2$。这个计算中，尽管我们已经知道最后的结果不会大于 6，但还是会用到相当大的中间结果 6561。更加巧妙的方法是首先执行两个部分指数运算：$3^8 = 3^4 \times 3^4 = 81 \times 81$；然后将中间结果 81 替换为同一等价类中的其他元素。在模数 7 的等价类中，最小的正元素是 4(由于 $81 = 11 \times 7 + 4$)，所以 $3^8 = 81 \times 81 \equiv 4 \times 4 = 16 \bmod 7 \equiv 2 \bmod 7$。计算过程为：

$$3^8 = 3^4 \times 3^4 = 81 \times 81 \equiv 4 \times 4 = 16 \bmod 7 \equiv 2 \bmod 7$$

此时计算可以不使用计算器,因为它所涉及的所有数字都不会大于 81。而第一种方法中计算 6561 除以 7 就已经具有一定的挑战性。

我们必须牢记的通用规则就是:应该尽早使用模约简,使计算的数值尽可能小,这样做总是极具计算优势。当然,不管在等价类中怎么切换,任何模数计算的最终结果都是相同的。

同余式的存在说明:$y=x \bmod p$,对于同一个 y,x 的取值范围是等价类集合,不具有唯一性,因此 y 与 x 的对应关系是 1 对 n 的关系,由 x 计算 y 是唯一确定的,而由 y 推断 x 是不确定的,不具有唯一性。这正是密码学中大量使用模运算的原因所在。

费马定理:若 p 是素数,a 是正整数且不能被 p 整除,则:$a^{(p-1)} \bmod p = 1 \bmod p$。

推论:若 p 是素数,a 是正整数且不能被 p 整除,则:$a^p \bmod p = a \bmod p$。

要求 $a/c \bmod p$,当 c 与 p 互质时,使用费马定理可得:

$$a/c \bmod p = a/c \bmod p \times 1 = a/c \bmod p \times c^{(p-1)} \bmod p = a \times c^{(p-2)} \bmod p$$

例如,$13 \times 14^9 \bmod 11 = ((13 \bmod 11) \times (14 \bmod 11)^9) \bmod 11 = (2 \times 3^9) \bmod 11 = (2/3) \bmod 11 = 8$。

2.3.2 RSA 算法

RSA 算法就是找两个很大的素数:一个公开给所有人,称之为公钥;另一个不告诉任何人,称之为私钥。两把密钥互补——用公钥加密的密文可以用私钥解密,反过来也一样。

这里特别强调,私钥不允许告诉任何人,也就是具有唯一性。计算机产生私钥后只需存储于本地,不能也不需要在网上传输,从而保证了其唯一保密性。

RSA 算法如图 2-7 所示。

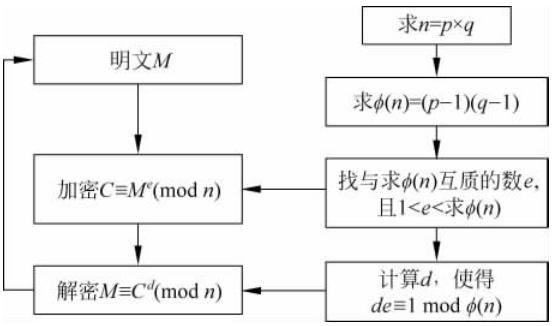

图 2-7 算法流程

(1) 选择一对不同的、足够大的素数 p 和 q(目前两个数的长度都接近 512bit 被认为是安全的)。

(2) 计算 $n=p \times q$。

(3) 计算欧拉函数 $\phi(n)=(p-1)(q-1)$,同时对 p、q 严加保密,不让任何人知道。

(4) 找一个与 $\phi(n)$互质的数 e,且 $1<e<\phi(n)$。

(5) 计算 d,使得 $d \times e \equiv 1 \bmod \phi(n)$。这个公式也可以表达为 $d \equiv e^{-1} \bmod \phi(n)$,显而易见,不管 $\phi(n)$取什么值,符号右边 $1 \bmod \phi(n)$的结果都等于 1;符号的左边 d 与 e 的乘积做模运算后的结果也必须等于 1。这就需要计算出 d 的值,让这个同余等式能够成立。

(6) 公钥 $K_{pub}=(e,n)$，私钥 $K_{pri}=(d,n)$。

(7) 加密过程为：$C\equiv M^e \pmod n$ (M 为待加密的明文)。加密时，先将明文变换成 0 至 $n-1$ 的一个整数 M。若明文较长，可先分割成适当的组，然后再进行交换。

(8) 解密过程为：$M\equiv C^d \pmod n=(M^e)^d \bmod n=M^{ed} \bmod n$($C$ 为待解密的密文)。

RSA 遭受攻击的很多情况是因为算法实现的一些细节上的漏洞所导致的，所以在使用 RSA 算法构造密码系统时，为保证安全，在生成大素数的基础上，还必须认真仔细选择参数，防止漏洞的形成。根据 RSA 加解密过程，其主要参数有 3 个：模数 n，加密密钥 e，解密密钥 d。

1) 模数 n 的确定

(1) p 和 q 之差要大(当 p 和 q 相差很小时，在已知 n 的情况下，可假定二者的平均值为 n，然后利用等式右边开方则得到 n，即 n 被分解)。

(2) $p-1$ 和 $q-1$ 的最大公因子应很小。

(3) p 和 q 必须为强素数。

2) 参数 e 的选取原则

在 RSA 算法中，e 和互质的条件容易满足，如果选择较小的 e，则加、解密的速度加快，也便于存储，但会导致安全问题。

一般地，e 的选取有如下原则：

(1) e 不能够太小。一般选择 e 为 16 位的素数。

(2) e 应选择使其在 mod $\phi(n)$的阶为最大。即存在 i，使得 $e^i\equiv 1 \bmod \phi(n)$，$i\geqslant(p-1)\times(q-1)/2$ 可以有效抗击攻击。

3) d 选取原则

一般地，私密密钥 d 要大于 $n^{\frac{1}{4}}$。在许多应用场合，常希望使用位数较短的密钥以降低解密或签名的时间。例如，在 IC 卡应用中，IC 卡 CPU 的计算能力远低于计算机主机。长度较短的 d 可以减少 IC 卡的解密或签名时间，而让较复杂的加密或验证预算(e 长度较长)由快速的计算机主机运行。一个直接的问题就是：解密密钥 d 的长度减少是否会造成安全性的降低？很明显地，若 d 的长度太小，则可以利用已知明文 M 加密后得 $C=M^e \bmod n$，再直接猜测 d，求出 $C^d \bmod n$ 是否等于 M。若是，则猜测正确，否则继续猜测。若 d 的长度过小，则猜测的空间变小，猜中的可能性加大，已有证明当 $d<n^{\frac{1}{4}}$ 时，可以由连分式算法在多项式时间内求出 d 值。因此其长度不能过小。

【例 2-18】 使用 RSA 算法实现用户 A 将明文“key”加密后传递给用户 B。

解：(1) 计算公私密钥(e,n)和(d,n)。令 $p=3$，$q=11$，得出 $n=p\times q=3\times 11=33$；$\phi(n)=(p-1)(q-1)=2\times 10=20$；取 $e=3$，(3 与 20 互质)则 $e\times d\equiv 1 \bmod \phi(n)$，即 $3\times d\equiv 1 \bmod 20$。$d$ 怎样取值呢？可以用试算的办法来寻找，结果如表 2-13 所示。

表 2-13　$3\times d\equiv 1 \bmod 20$ 试算结果

d	$e\times d=3\times d$	$(e\times d)\bmod(p-1)(q-1)=(3\times d) \bmod 20$
1	3	3
2	6	6
3	9	9

续表

d	$e\times d=3\times d$	$(e\times d)\bmod(p-1)(q-1)=(3\times d) \bmod 20$
4	12	12
5	15	15
6	18	18
7	**21**	**1**
8	24	3
9	27	6

通过试算可知,当 $d=7$ 时,$e\times d\equiv 1 \bmod \phi(n)$同余等式成立。因此,可令 $d=7$。从而我们可以设计出一对公私密钥:

加密密钥(公钥)$k_{pub}=(e,n)=(3,33)$;解密密钥(私钥)$k_{pri}=(d,n)=(7,33)$。

(2) 英文数字化。如表 2-14 所示,将明文信息数字化,并将每块两个数字分组。则得到分组后的 key 的明文信息为:11,05,25。

表 2-14 英文数字化

字母	a	b	c	d	e	f	g	h	i	j	k	l	m
码值	01	02	03	04	**05**	06	07	08	09	10	**11**	12	13
字母	n	o	p	q	r	s	t	u	v	w	x	y	z
码值	14	15	16	17	18	19	20	21	22	23	24	**25**	26

(3) 明文加密。用户加密密钥(3,33)将数字化明文分组信息加密成密文。由 $C\equiv M^e \pmod n$得:

$C1\equiv(M1)^e \pmod n\equiv 11^3 \pmod{33}\equiv 11$　$C2\equiv(M2)^e \pmod n\equiv 5^3 \pmod{33}\equiv 26$

$C3\equiv(M3)^e \pmod n\equiv 25^3 \pmod{33}\equiv 16$

因此,得到相应的密文信息为:11,26,16,对应字符串为 K2P。

(4) 密文解密。用户 B 收到密文,若将其解密,只需要计算 $M\equiv C^d \pmod n$,即:

$M1\equiv(C1)^d \pmod n=11^7 \pmod{33}=11$　$M2\equiv(C2)^d \pmod n=26^7 \pmod{33}=05$

$M3\equiv(C3)^d \pmod n=16^7 \pmod{33}=25$

用户 B 得到明文信息为:11,05,25。根据上面的编码表将其转换为英文,我们又得到了恢复后的原文 key。

当然,实际运算要比这复杂得多,由于 RSA 算法的公钥私钥的长度(模长度)要到 1024 位甚至 2048 位才能保证安全,因此,p、q、e 的选取,公钥私钥的生成,加密解密模指数运算都有一定的计算量,需要使用计算机才能高速完成。

RSA 算法的安全

密码分析者攻击 RSA 体制的关键点在于如何分解 n。若分解成功使 $n=pq$,则可以算出 $\phi(n)=(p-1)(q-1)$,然后由公开的 e,解出秘密的 d。所以说 RSA 算法的安全性基于分解大整数的困难性。

RSA 公钥密码系统基于“大整数分解”这一著名数论难题:将两个大素数相乘十分容易,但要将乘积结果分解为两个大素数因子却极为困难。举例来说,将两个素数 11 927 和 20 903 相乘,可以很容易地得出其结果 249 310 081;但是要想将 249 310 081 分解因子得到

相应的两个素数却极为困难。

Rivest、Shamir、Adleman 提出，两个素数的乘积如果长度达到了 130 位，则将该乘积分解为两个素数需要花费近百万年的时间。为了证明这一点，他们找到 1 个 129 位数，向世界挑战找出它的两个因子，这个 129 位的数被称为 RSA129，其值为 114 381 625 757 888 867 669 235 779 976 146 612 010 218 296 721 242 362 562 561 842 935 706 935 245 733 897 830 597 123 563 958 705 058 989 075 147 599 290 026 879 543 541。世界各地 600 多名研究人员通过 Internet 协调各自的工作向这个 129 位数发起进攻。花费了 9 个月的时间，终于分解出了 RSA129 的两个素数因子。两个素数因子一个长为 64 位，另一个长为 65 位。64 位的素数是 3 490 529 510 847 650 949 147 849 619 903 898 133 417 764 638 493 387 843 990 820 577，65 的素数是 32 769 132 993 266 709 549 961 988 190 834 461 413 177 642 967 992 942 539 798 288 533。RSA129 虽然没有如 RSA 三位专家预计的那样花费极长的时间破解，但它的破解足以说明两方面的问题：一是大整数的因子分解问题需要高昂的计算开销，计算不可行；二是通过 Internet 让大量的普通计算机协同工作可以获得强大的计算能力。

RSA 的安全性依赖于大整数分解，但是否等同于大整数分解一直未能得到理论上的证明，因为没有证明破解 RSA 就一定需要作大整数分解。假设存在一种无须分解大数的算法，那它肯定可以修改成为大整数分解算法。目前，RSA 的一些变种算法已被证明等价于大整数分解。

不管怎样，分解 n 是最直接的攻击方法，人们已能分解多个十进制位的大素数。因此，模数 n 必须选大一些，如 1024、2048 位二进制数。

由于进行的都是大数计算，使得 RSA 最快的情况也比 DES 慢上数倍，无论是用软件还是硬件实现。速度一直是 RSA 的缺陷。一般来说只适用于少量数据加密。

2.3.3　RSA 用法

RSA 算法是第一个既能用于数据加密也能用于数字签名的算法。每个用户拥有一个仅为本人所掌握的私钥，用它进行解密和签名；同时拥有一个公钥用于文件发送时加密。

1. RSA 数据加密用法

当发送一份保密文件时，发送方使用接收方的公钥对数据加密，而接收方则使用自己的私钥解密，这样，信息就可以安全无误地到达目的地了，即使被第三方截获，由于没有相应的私钥，也无法进行解密。

如图 2-8 所示，A 向 B 发信息，A 需要 B 的公钥。A 用 B 的公钥加密信息发出，B 收到后用自己的私钥解密还原出原文，这样就保证了信息传输的安全性。

密文在传输过程中，任何人都可以获得，但由于密文是用 B 的公钥加密的，只有 B 的私钥能够解密，其他人无法解密密文，因而实现了信息传输的机密性。

RSA 数据加密的特征是：先公后私（发送方先用接收方公钥加密信息发送，接收方接收加密信息后用接收方私钥解密）。

错误的使用方法是用发送方公钥加密。因为接收方没有发送方的私钥，无法解密使用发送方公钥加密的信息，将导致接收的加密信息无法使用。

使用 RSA 对通信的双方进行信息加密保护时，其实际步骤如下：

- 每个用户产生一对密钥用于加密和解密消息。

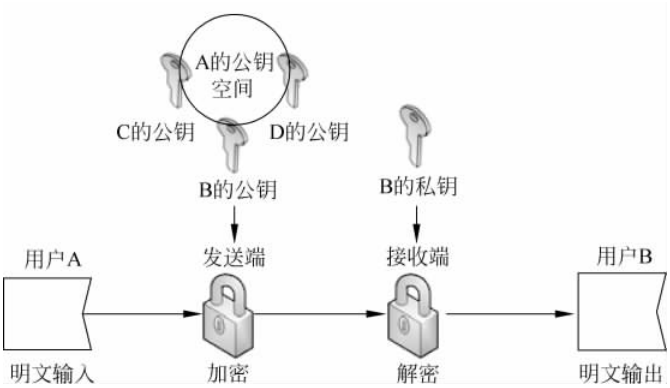

图 2-8 RSA 加密用法

- 每个用户将其中的一个密钥放入公共寄存器或者其他可访问的文件中,这个密钥就是公钥。该用户把另一个密钥自己保存,这个密钥就是私钥。如图 2-8 所示,用户 A 拥有从其他人那里获得的公钥的集合。
- 如果 A 希望向 B 发送一条私人信息,那么 A 使用 B 的公钥加密消息。
- 当 B 收到该信息的时候,B 使用 B 的私钥解密信息。没有其他的接收者能够解密消息,因为只有 B 知道私钥。

在这种方法中,所有的参与者都能够访问公钥,而私钥是由每个参与者在本地产生,因此不需要分配。只要用户保护好他的私钥,接收的通信信息就是安全的。在任何时候,用户都能够改变私钥,并公布相应的公钥值以替换旧的公钥值。

通常约定在对称加密中使用的密钥为密钥,而在公钥加密中使用的两个密钥分别为公钥和私钥。

2. 对称密钥分发

由于公钥加密系统效率较低,几乎不会用于大量数据的直接加密,而是经常用在少量数据的加密上,其最重要的应用之一就是用于对称密钥分发,例如密钥分发中心(Key Distribution Center,KDC)的主密钥分发。

对称密钥交换主要用下述两种方法来生成密钥对:

第一种方法是密钥对在客户端系统上生成,然后公钥以安全的方式传递给其他用户。

(1) A 产生公私钥对{PUa,PRa},然后将 A 的公钥 PUa 和身份信息 IDa 发给 B。

(2) B 产生一个会话密钥 K_b,并用 A 的公钥 PUa 加密发给 A。由于只有 A 的私钥能解密,因此 A 能得到会话密钥。

这个方案很容易被中间人攻击。即有中间人同时扮演 B 和 A 的角色分别与 A 和 B 通信,从而获取会话密钥。根本原因就是这个方案缺乏认证。

第二种方法是由一个可信的第三方(如 KDC 或 CA 或 CA 的授权 RA)生成密钥对,然后以安全的方式传递给用户。该方案的基本原理是密钥分发中心 KDC 和每个终端用户都共享一对唯一的主密钥(用物理的方式传递,如 U 盾)。终端用户之间每次会话,都要向 KDC 申请唯一的会话密钥,会话密钥通过与 KDC 共享的主密钥加密来完成传递。

(1) A 以明文形式向 KDC 发送会话密钥请求包。包括通话双方 A、B 的身份以及该次

传输的唯一标识 n_1，称为临时交互号(nonce)。临时交互号可以选择时间戳、随机数或者计数器等。KDC 可根据临时交互号设计防重放机制。

(2) KDC 返回的信息包括两部分。第一部分是 A 想获取的信息，用 A 的主密钥 K_A 加密，包括通话密钥 K_s 和 KDC 收到的请求包内容以验证消息到达 KDC 前是否被修改或者重放过。第二部分是 B 想获取的信息，用 B 的主密钥 K_B 加密，包括通话密钥 K_s 和 A 的身份。A 收到后这部分消息便原样发给 B。

(3) 为保证 A 发给 B 的会话密钥信息未被重放攻击，A、B 使用会话密钥进行最后的验证。B 使用新的会话密钥 K_s 加密临时交互号 n_2 并发给 A。A 对 n_2 进行一个函数变换后，用会话密钥发给 B 验证。

在 RSA 密钥体制中，当 A 用户发文件给 B 用户时，A 用户用 B 用户公开的密钥加密明文，B 用户则用解密密钥解读密文，其特点如下。

(1) 密钥配发十分方便，用户的公用密钥可以像电话号码簿那样公开，使用方便。对网络环境下众多用户的系统，密钥管理更加简便，每个用户只需持有一对密钥就可实现与网络中任何一个用户的保密通信。

(2) RSA 加密原理基于单向函数，非法接收者利用公用密钥不可能在有限时间内推算出秘密密钥，这种算法的保密性能较好。

3. RSA 数字签名用法

同加密用法相反，RSA 数字签名用法的特征是先私后公：发送方先用自己的私钥签名(加密)信息发送，实现自身身份证明，接收方接收后将签名(加密)信息用发送方公钥验证身份(解密)。

如图 2-9 所示，第三方 C 得到加密的信息后，虽然可通过 A 公钥得到明文的消息，并对消息进行更改，但是第三方无法得到 A 私钥，即使将更改后的消息发给 B，但 B 无法用 A 公钥进行解密，从而达到识别 A 身份(身份认证)的效果。

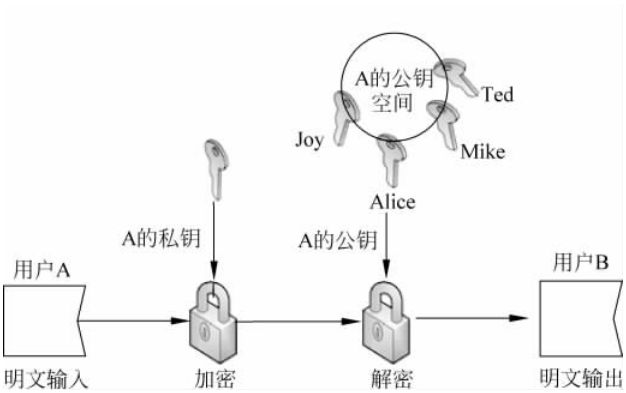

图 2-9　RSA 签名用法

数字签名用于发送方身份认证和验证消息的完整性，要求具有唯一性、不可抵赖、不可伪造等特性。RSA 的私钥是仅有发送方知道的唯一密钥，具有唯一性；使用该密钥加密消息(即数字签名)，加密者无法抵赖，具有不可抵赖性；RSA 加密强度保证了私钥破译计算不可行，因而难以伪造，从而具有保密性。因而 RSA 符合数字签名的要求，能够实现数字签名。

RSA在用户确认和实现数字签名方面优于现有的其他加密机制。RSA数字签名是一种强有力的认证鉴别方式,可保证接收方能够判定发送方的真实身份。另外,如果信息离开发送方后发生变更,它可以确保这种变更能被发现。更为重要的是,当收发方发生争执时,数字签名提供了不可抵赖的事实。

综上所述,RSA的用法可表示为:

给定 $n=pq$,p 和 q 是大素数,$ed \bmod \Psi(n)=1$,公开密钥为(n,e),秘密密钥为(n,d)。

加密用法:

公钥加密 $m\in[0,n-1]$,$\gcd(m,n)=1$,则 $c=m^e \bmod n$

私钥解密 $m=c^d \bmod n=(m^e \bmod n)^d \bmod n=m^{ed} \bmod n=m$

签名用法:

私钥签名 $s=m^d \bmod n$

公钥验证 $m=s^e \bmod n=(m^d \bmod n)^e \bmod n=m^{ed} \bmod n=m$

【例2-19】 给定 $n=55=11\times5$,$\Psi(n)=40$,选 $d=11$,则 $e=11$,$m=3$,$c=3^{11} \bmod 55=47$,验证:$m=c^{11} \bmod 55=47^{11} \bmod 55=3$。

【例2-20】 给定 $n=65$,$\Psi(n)=48$,选 $d=5$,则 $e=29$,$m=3$,$s=3^5 \bmod 65=48$,验证:$m=48^{29} \bmod 65$。

综上所述:

(1) 公钥密码学与其他密码学完全不同,使用这种方法的加密系统,不仅公开加解密算法本身,也公开了加密用的密钥。

(2) 公钥密码系统与只使用一个密钥的对称传统密码不同,算法是基于数学函数而不是基于替换和置换的。

(3) 公钥密码学是非对称的,它使用两个独立的密钥,即密钥分为公钥和私钥,因此称双密钥体制。双钥体制的公钥可以公开,因此称为公钥算法。算法加密密钥能够公开,即陌生者能用加密密钥加密信息,但只有用相应的解密密钥才能解密信息。规定加密密钥叫做公开密钥,解密密钥叫做私人密钥。

由于算法的加密与解密由不同的密钥完成,并且从加密密钥得到解密密钥计算不可行,所以算法也称为非对称加密算法。

那么什么是理论安全(无条件安全)?什么是实际安全(计算上安全、计算不可行)?

理论安全是指攻击者无论截获多少密文,都无法得到足够的信息来唯一地决定明文。Shannon用理论证明:欲达理论安全,加密密钥长度必须大于等于明文长度,密钥只用一次,用完即丢,即一次一密(One-time Pad),不具有实用价值。

实际安全是指如果攻击者拥有无限资源,那么任何密码系统都是可以被破译的;但在有限的资源范围内,攻击者都不能通过系统地分析方法来破解系统,则称这个系统是计算上安全的或破译这个系统是计算上不可行的(computationally infeasible)。

公钥密码使用的数学难题都是可解的,因此不具有理论安全性;但破解公钥密码计算量巨大,相对于现有的计算机计算能力而言,计算时间都以年为单位,不具有计算可行性,而且可以随着计算能力的提高不断增加秘钥的长度来进一步提高破解难度,因此公钥密码是计算不可行的。

(4) 公钥算法的出现,给密码的发展开辟了新的方向。公钥算法虽然已经历了多年的

发展，但仍具有强劲的发展势头，在鉴别系统和密钥交换等安全技术领域起着关键的作用。

2.4　椭圆曲线密码

大多数使用公钥密码学进行加密和数字签名的产品和标准都使用 RSA 算法。为了保证 RSA 使用的安全性，最近这些年来密钥的位数一直在增加，伴随而来的是运算负担越来越大，这对使用 RSA 的应用是很重的负担，对进行大量安全交易的电子商务更是如此。

1985 年，Neal Koblitz 和 Victor Miller 将椭圆曲线引入密码学，提出了椭圆曲线密码系统 ECC(Elliptic Curves Cryptography，椭圆曲线密码)。ECC 算法的优点在于：

(1) 安全性高。安全性基于椭圆曲线上的离散对数问题的困难性。目前还没找到解决椭圆曲线上离散对数问题的亚指数时间算法。而大数因子分解和离散对数的求解都存在亚指数时间算法。

(2) 短密钥。随着密钥长度的增加，求解椭圆曲线上离散对数问题的难度，比同等长度大数因子分解和求解离散对数问题的难度要大得多。如表 2-15 所示，椭圆曲线密码算法仅需要更小的密钥长度就可以提供 RSA 相当的安全性，因此可以减少处理负荷。

表 2-15　ECC 和 RSA 对比

ECC 密钥尺寸(bit)	106	132	160	220	600
RSA 密钥尺寸(bit)	512	768	1024	2048	21000
破解时间(MIPS 年)	104	108	1011	1020	1078
ECC/RSA 密钥尺寸比例	1∶5	1∶6	1∶7	1∶10	1∶35

(3) 灵活性好。可以改变曲线的参数得到不同的曲线，形成不同的循环群，构造密码算法具有多选择性。

因此 ECC 和 RSA 相比，在许多方面都有绝对的优势，主要体现在以下方面：

(1) 抗攻击性强。相同的密钥长度，其抗攻击性要强很多倍。

(2) 计算量小，处理速度快。ECC 总的速度比 RSA、DSA 要快得多。

(3) 存储空间占用小。ECC 的密钥尺寸和系统参数与 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，两类密码系统有相同的带宽要求，但应用于短消息时 ECC 带宽要求却低得多。带宽要求低使 ECC 在无线网络领域具有广泛的应用前景。

ECC 应用方面已被 IEEE 公钥密码标准采用。它既可以加密又可以实现数字签名，便于软硬件实现，密钥生成速度快，特别适合于对计算能力要求不高的系统，如智能卡、手机。

2.4.1　实数域上的椭圆曲线

密码学中的椭圆曲线绘出的图像并非椭圆，之所以称为椭圆曲线是因为曲线方程与计算椭圆周长的方程相似，也是用三次方程来表示的。

一般形式的椭圆曲线三次方程是一个具有两个变量 x 和 y 的魏尔斯特拉斯方程：y^2+

$axy+by=x^3+cx^2+dx+e$,其中 a、b、c、d、e 是实数,x 和 y 在实数集上取值。

密码中普遍采用实数域上的椭圆曲线,实数域上的椭圆曲线是指曲线方程定义中,所有系数都是某一实数域 GF(p)中的元素(其中 p 为大于 3 的素数)。

常用椭圆曲线形式是:

$$y^2 = x^3 + ax + b(a,b \in \mathrm{GF}(p), 4a^3 + 27b^2 \neq 0)$$

常用椭圆曲线由上述方程确定的所有点(x, y)的集合,加一个无穷远点 O(认为其 y 坐标无穷大)所定义。给定 a 和 b,常用椭圆曲线用集合 $E(a, b)$表示。

对 x 的每一个值,$y=\pm\sqrt{x^3+ax+b}$,即 y 都有一个正值和负值,这样每一曲线都关于 $y=0$ 对称,如图 2-10 所示。

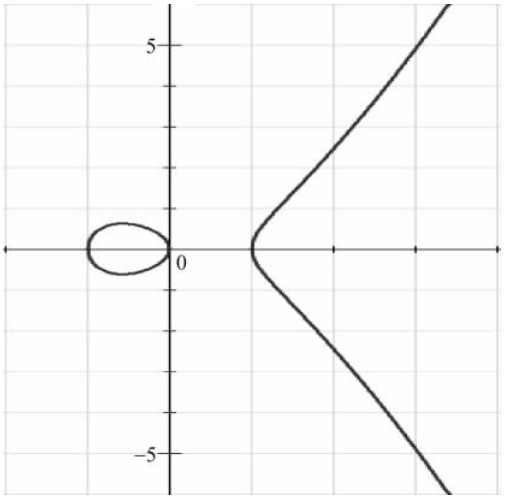

图 2-10 $y^2 \equiv x^3 - x$

需要提醒大家注意的是,一般椭圆曲线不一定关于 x 轴对称,例如 $y^2-xy=x^3+1$ 对应的图像椭圆曲线就不关于 x 轴对称。

实数域上椭圆曲线的加法运算

椭圆曲线的参数 a 和 b,如果满足条件:$4a^3+27b^2 \neq 0$,则可基于集合 $E(a, b)$定义一个群。该群按以下加法运算规则构成阿贝尔群:如果椭圆曲线上的 3 个点 A、B、C 在同一直线上,则它们的和等于零元 O,即 $A+B+C=O$。

定义无穷远点是加法零元 O。由弦切法可以定义椭圆曲线加法的运算规则:

(1) O 为加法零元(单位元),即对椭圆曲线上任一点 P,有 $P+O = P$;

(2) 设 $P_1=(x, y)$是椭圆曲线上的一点,其加法逆元 $P_2 = -P_1 = (x, -y)$;

(3) 设 Q、P 是椭圆曲线上的两点,它们相加后的点是 Q 和 P 的连线与椭圆曲线的交点的加法逆元。设交点是 R,由 $Q+P+R= O$,得 $Q+P=-R$;

使用弦切法的做法是:任意取椭圆曲线上两点 P,Q(若 P,Q 两点重合,则做 P 点的切线),过两点做直线交于椭圆曲线的另一点 R',过 R'做 y 轴的平行线交于 R,规定 $P+Q=R$。

(4) P 的倍点是 P 点所做的椭圆曲线的切线与椭圆曲线的交点的加法逆元。设交点是 S,定义 $2P = P+P=-S$。类似地,可定义 $3P = 2P+P$ 等。

(5) 若存在最小正整数 n,使得 $nP=O(P \in E(a, b))$,则 n 为椭圆曲线 E 上点 P 的阶。

以上运算规则的几何意义是：

(1) 无穷远点 O 的坐标无穷大，为一虚拟的点，在图像上无法显示。

(2) 一条与 x 轴垂直的直线和曲线相交于两个点 P_1 和 P_2，这两个点的 x 坐标相同，即 $P_1=(x, y)$ 和 $P_2=(x, -y)$，同时它也与曲线相交于无穷远点 O，因此 $P_2=-P_1$。故 P 与其逆元 $-P$ 成对出现在椭圆曲线上；求一点的逆元方法是过该点做 y 轴的平行线，其与椭圆曲线的另一交点即为逆元。

(3) 横坐标不同的两个点 Q 和 P 相加，先在两点之间画一条直线并求直线与椭圆曲线的第三个交点 R'，然后过 R' 做 y 轴的平行线交于 R，则 R 为两个点 Q 和 P 的和，即 $P+Q=R$。

(4) 两个相同的点 P 相加，通过该点画一条切线，切线与椭圆曲线相交于另一点 R'，然后过 R' 做 y 轴的平行线交于 R，则 R 为 $2P$ 点，即 $2P=R$。

【例 2-21】 求椭圆曲线 $E(-10, 15)$：$y^2=x^3-10x+15$ 上点 $P(-3.23, 3.70)$ 和 $Q(3.45, 4.66)$ 的和。

(1) 如图 2-11 所示，选择点 $P(-3.23, 3.70)$ 和点 $Q(3.45, 4.66)$。

(2) 根据加法规则，设通过点 P 和 Q 的直线与椭圆曲线相交于点 $-R$，$-R$ 是点 R 关于 x 轴的镜像。R 就是点 P 和 Q 的和，如图 2-11 所示点 $R=(-0.21, 4.13)$。

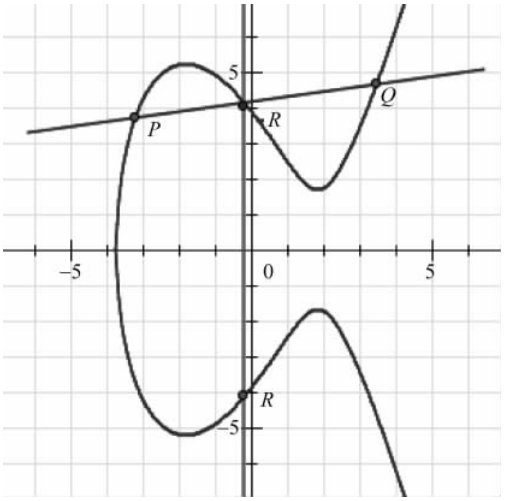

图 2-11　$y^2=x^3-10x+15$　$P+Q=R$

【例 2-22】 求椭圆曲线 $E(-10, 15)$：$y^2=x^3-10x+15$ 上点 $P(-1.27, 5.07)$ 的 2 倍点和 3 倍点。

(1) 如图 2-12 所示，选定椭圆曲线上一点 $P(-1.27, 5.07)$。

(2) 根据加法规则，设点 P 的切线与椭圆曲线交于点 $-R$，$-R$ 是点 R 关于 x 轴的镜像。R 点就是点 P 的 2 倍。如图 2-12 所示，2 倍点 $R=2P=(2.80, -3.00)$。

(3) 根据加法规则，设点 $2P$ 的切线与椭圆曲线交于点 $-R$，$-R$ 是点 R 关于 x 轴的镜像。R 点就是点 P 的 3 倍。如图 2-13 所示，3 倍点 $R=2P+P=(2.39, 2.17)$。

2.4.2　有限域上的椭圆曲线

由方程 $y^2 \bmod p=(x^3+ax+b) \bmod p$ 定义的椭圆曲线，若方程的变元和系数均为有

限域 F_p 中的元素(其中 p 为素数),即 x、y、a、$b \in F_p$,运算为模 p 运算,则该椭圆曲线为有限域 F_p 上的椭圆曲线。

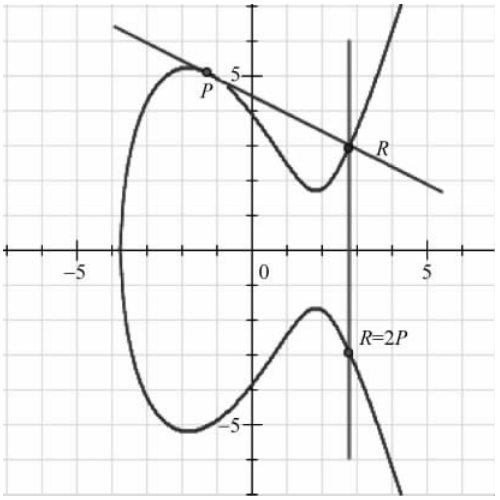

图 2-12　2 倍点

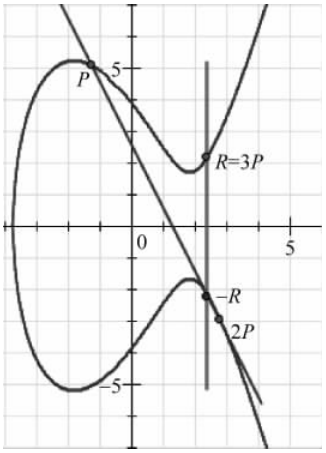

图 2-13　3 倍点

所有满足上述方程的整数对(x, y)和无穷远点 O 组成的集合 $E_p(a, b)$即有限域上 F_p 的椭圆曲线。类似实数域上的椭圆曲线,有限域 F_p 上的椭圆曲线关于 $y=p/2$ 对称。

图 2-14 给出了有限域 F_{23}椭圆曲线 $E_{23}(11, 20)$的可视化结果。从图中可以看出,椭圆曲线分别关于 $y=11.5$ 对称。

【例 2-23】 求有限域 F_{23}椭圆曲线 $E_{23}(11, 20)$:$y^2 \equiv x^3+11x+20$ 第一象限整数点的集合。

解:$\because x=0, \therefore (x^3+11x+20) \bmod 23=20 \quad \therefore y^2 \bmod 23 \equiv 1$

$\therefore$ y 依次取正整数 1,2,3,…,23,试算:

$y=1, y^2 \bmod 23 \equiv 1$; $y=2, y^2 \bmod 23 \equiv 4$; $y=3, y^2 \bmod 23 \equiv 9$;

$y=4, y^2 \bmod 23 \equiv 16$; $y=5, y^2 \bmod 23 \equiv 25 \bmod 23 \equiv 2$; …;

$y=22, y^2 \bmod 23 \equiv 484 \bmod 23 \equiv 1$; $y=23$ 时,$y=p$,本次试算结束。

$\therefore$无满足条件的解。

$\because x=1, \therefore (x^3+11x+20) \bmod 23=9 \quad \therefore y^2 \bmod 23 \equiv 9$

$\therefore$ y 依次取正整数 1,2,3,…,23,试算:

$y=1, y^2 \bmod 23 \equiv 1$; $y=2, y^2 \bmod 23 \equiv 4$; $y=3, y^2 \bmod 23 \equiv 9$;

$y=4, y^2 \bmod 23 \equiv 16$; $y=5, y^2 \bmod 23 \equiv 25 \bmod 23 \equiv 2$; …;

$y=22, y^2 \bmod 23 \equiv 484 \bmod 23 \equiv 1$; $y=23$ 时,$y=p$,本次试算结束。

$\therefore$ $y_1=3, y_2=20$。

类似地,使用表 2-16 可得其他运算结果。当 $x=23$ 时,$x=p$,本题运算结束。所以方程在第一象限整数点的集合为表 2-17 中的 27 个点加上无穷点 O,共 28 个点。具体的可视化结果见图 2-14。

根据计算结果可知,椭圆曲线中 $y \to x$ 的对应关系是 1 对 n 的关系,如 $y=16$,$x=6$、7、

10，因此根据 y 不能具体确定 x，这就是椭圆曲线能够用于密码学的原因。

表 2-16　$y^2 \equiv x^3+11x+20$ 试算

x	(x^3+x+1)	y_1	$(y_1)^2$	y_2	$(y_2)^2$
1	32＝23＋9	3	9	20	400＝391＋9
2	50＝23×2＋4	2	4	21	441＝437＋4
4	128＝23×5＋13	6	36＝23＋13	17	289＝23×12＋13
5	200＝23×8＋16	4	16	19	361＝345＋16
6	302＝23×13＋3	7	49＝46＋3	16	256＝23×11＋3
7	440＝23×19＋3	7		16	
10	1130＝23×49＋3	7		16	
11	1472＝23×64	0	0		
15	3560＝23×154＋18	8	64＝46＋18	15	225＝23×9＋18
18	6050＝23×263＋1	1	1	22	484＝23×21＋1
19	7088＝23×308＋4	2		21	
20	8240＝23×358＋6	11	121＝115＋6	12	144＝23×6＋6
21	9512＝23×413＋13	6		17	
22	10910＝23×474＋8	10	100＝92＋8	13	169＝23×7＋8

表 2-17　(x,y)值

第一象限点集			
(1,3)	(1,20)	(0,0)	
(2,2)	(2,21)	(15,8)	(15,15)
(4,6)	(4,17)	(18,1)	(18,22)
(5,4)	(5,19)	(19,2)	(19,21)
(6,7)	(6,16)	(20,11)	(20,12)
(7,7)	(7,16)	(21,6)	(21,17)
(10,7)	(10,16)	(22,10)	(22,13)

有限域 F_p 上椭圆曲线的加法运算

椭圆曲线的参数 a 和 b，如果满足条件：$4a^3+27b^2 \pmod p \neq 0$，则可基于集合 $E_p(a,b)$ 定义一个有限阿贝尔群。$E_p(a,b)$ 上的加法运算构造与定义在实数域上的椭圆曲线中描述的代数方法是一致的。对任何点 $P,Q \in F_p$，加法运算的代数描述如下：

(1) O 为加法单位元，$P+O=P$；

(2) 若 $P=(x,y)$，$(x,-y)$是 P 的加法逆元，表示为 $-P$。$-P$ 也是 $E_p(a,b)$中的点。

(3) 若 $P=(x_1,y_1)$，$Q=(x_2,y_2)$，$P \neq -Q$，则 $P+Q=(x_3,y_3)$按以下规则确定：

$$x_3 \equiv \lambda^2-x_1-x_2 \pmod p,\ y_3 \equiv \lambda(x_1-x_3)-y_1 \pmod p,$$

$$\lambda=\begin{cases}\dfrac{y_2-y_1}{x_2-x_1}(P \neq Q)\\ \dfrac{3x_1^2+a}{2y_1}(P=Q)\end{cases}$$

用 $E_p(a,b)$表示方程所定义的椭圆曲线上的整数点集，由点集合$\{(x,y)$：$0 \leqslant x<p$，$0 \leqslant y<p$，x、y 为正整数$\}$并上无穷点 O 所得。

(4) 乘法定义为重复相加。如 $4P=P+P+P+P$。k 个相同的点 P 相加,记作 kP,称为 k 倍点,表示为:$Q=kP$,其中 Q、P 为 $E_p(a,b)$ 上的点,k 为小于 n(n 是点 P 的阶)的整数,说明 kP 仍然是椭圆曲线上的某一点。

k 倍点运算是指:给定一点 P 和一个整数 k,计算 kP,即 k 个 P 点的和。不难发现,给定 k 和 P,根据加法法则,计算 Q 很容易;但给定 K 和 Q,求 k 就相对困难了。

这就是椭圆曲线加密算法采用的数学难题——椭圆曲线离散对数(ECDLP)问题。我们把点 P 称为基点(base point),k($k<n$,n 为基点 P 的阶)称为私有密钥(public key),K 称为公开密钥(public key)。

椭圆曲线上的离散对数问题表示为:给定点 P 和 kP,计算整数 k。椭圆曲线密码体制的安全性便建立在椭圆曲线离散对数问题之上。

【例 2-24】 求椭圆曲线 $E_{23}(11,\ 20)$:$y^2=x^3+11x+20$ 上点 $P(7,\ 16)$和点 $Q(15,\ 8)$之和。

解:$\because P(7,\ 16)$,$Q(15,\ 8)$,$P\neq Q$

$\therefore \lambda=\dfrac{8-16}{15-7}=\dfrac{-8}{8}\equiv -1 \bmod 23=22$,$x_3=(22)^2-7-15=462 \bmod 23\equiv 2$,$y_3=22(7-2)-16=94 \bmod 23\equiv 2$,$P+Q=(2,2)$,如图 2-14 所示。

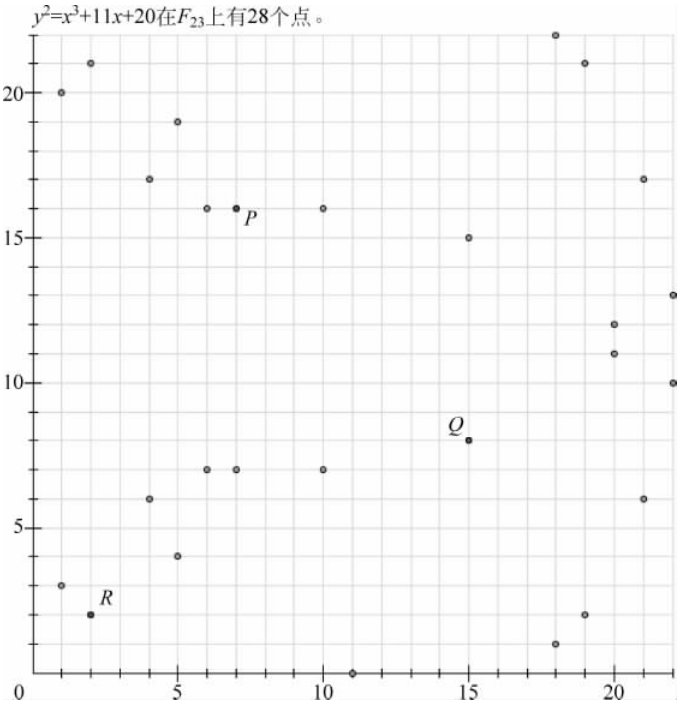

图 2-14 计算点 P 和 Q 的和

【例 2-25】 求椭圆曲线 $E_{23}(11,\ 20)$:$y^2=x^3+11x+20$ 上点 $P(7,\ 16)$的 2 倍点。

解:$\because P(7,\ 16)$,$P=Q$

$\therefore \lambda=\dfrac{3\times 7^2+11}{2\times 16}=\dfrac{158}{32} \bmod 23=15$,$x_3=(15)^2-7-7=211 \bmod 23\equiv 4$,$y_3=15(7-4)-16=29 \bmod 23\equiv 6$,$2P=(4,6)$,如图 2-15 所示。

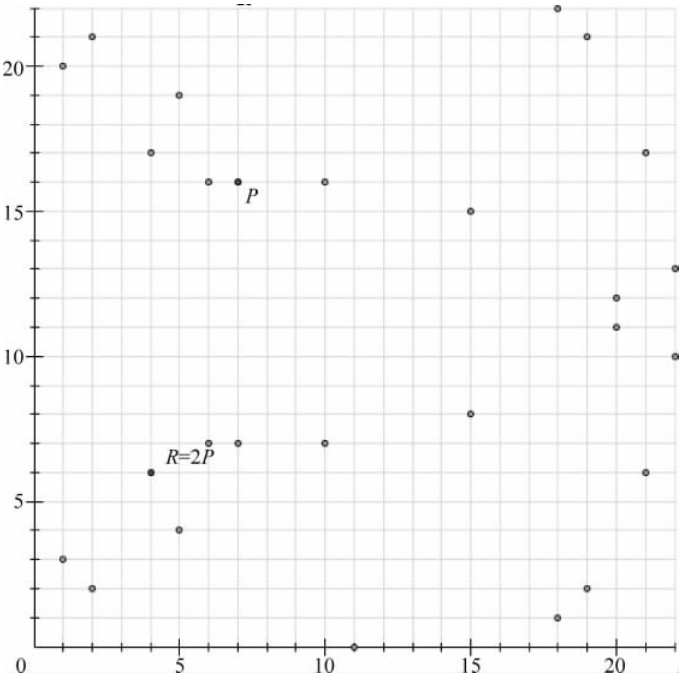

图 2-15　计算 2*P*

解题指导：分数(除法)取模运算需要转换为乘法取模运算。

难点是如何求 $\lambda=\frac{158}{32} \bmod 23$。

$\because \lambda=\frac{79}{16} \bmod 23 \quad \therefore 79 \bmod 23=16\lambda \bmod 23, 10=16\lambda \bmod 23$，取 $\lambda=1,2,\cdots,23$ 试算，可得 $\lambda=15$ 时等式成立。$\therefore \lambda=15$。

【例 2-26】　已知 $E_{11}(1,6)$上一点 $P(2,7)$，求 n 倍点 nP 的所有值。

解：$2P=(x_3,y_3)$，　$\lambda=\frac{3x_1^2+a}{2y_1}=\frac{13}{14}=8 \bmod 11$，

$x_3=64-2-2=60\equiv 5 \bmod 11$，　$y_3=8\times(2-5)-7\equiv 2 \bmod 11, 2P=(5,2)$

$3P=P+2P=(2,7)+(5,2)$，　$\lambda=\frac{y_2-y_1}{x_2-x_1}=-\frac{5}{3}=2 \bmod 11$

$x_3=4-2-5=-3\equiv 8 \bmod 11$，　$y_3=2\times(2-8)-7\equiv 3 \bmod 11, 3P=(8,3)$

$4P=2P+2P=(5,2)+(5,2)$，　$\lambda=\frac{3x_1^2+a}{2y_1}=19=8 \bmod 11$

$x_3=64-5-5=54\equiv 10 \bmod 11$，　$y_3=8\times(5-10)-2\equiv 2 \bmod 11, 4P=(10,2)$

$5P=P+4P=(3,6)$，　$6P=P+5P=(7,9)$，　$7P=P+6P=(7,2)$

$8P=4P+4P=(10,2)+(10,2)=(3,5)$，　$\lambda=\frac{3x_1^2+a}{2y_1}=\frac{301}{4}\equiv 1 \bmod 11$

$x_3=1-10-10=-19\equiv 3 \bmod 11$，　$y_3=1\times(10-3)-2\equiv 5 \bmod 11, 8P=(3,5)$

$9P=P+8P=(10,9), 10P=P+9P=(8,8)$，　$11P=P+10P=(5,9)$

$12P=8P+4P=(3,5)+(10,2)$，　$\lambda=\frac{y_2-y_1}{x_2-x_1}=-\frac{3}{7}=9 \bmod 11$

$x_3=81-3-10=68\equiv 2 \bmod 11$，　$y_3=9\times(3-2)-5=4\equiv 4 \bmod 11$，$12P=(2,4)$

$13P=P+12P=(2,7)+(2,4)=(0,0)$，此时 $\lambda=\dfrac{4-7}{2-2}=\dfrac{-3}{0}$ 趋向于无穷，表示无穷远点。根据定义，该点为零元，计算到此结束。

2.4.3 椭圆曲线加密解密

同 RSA 一样，ECC 也属于非对称公开密钥算法。

如图 2-16 所示，一个利用椭圆曲线进行加密通信的过程如下：

(1) 用户 A 选定一条椭圆曲线 $E_p(a,b)$，并取椭圆曲线上一点，作为基点 P。

(2) 用户 A 选择一个私有密钥 k，并生成公开密钥 $K=kP$。

(3) 用户 A 将 $E_p(a,b)$ 和点 K，P 传给用户 B。

(4) 用户 B 接到信息后，将待传输的明文编码到 $E_p(a,b)$ 上一点 M，并产生一个随机整数 $r(r<n)$。

(5) 用户 B 计算点 $C_1=M+rK$；$C_2=rP$。

(6) 用户 B 将 C_1、C_2 传给用户 A。

(7) 用户 A 接到信息后，计算 $C_1-k\,C_2$，结果就是点 M。

因为 $C_1-k\,C_2=M+rK-k(rP)=M+rK-r(kP)=M$，再对点 M 进行解码就可以得到明文。

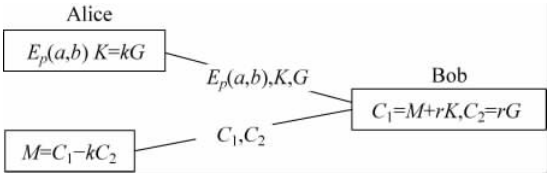

图 2-16　椭圆曲线加密解密

在这个加密通信中，如果有一个偷窥者 H，他只能看到 $E_p(a,b)$、K、P、C_1、C_2，而通过 K、P 求 k 或通过 C_2、P 求 r 都是计算不可行的。因此，H 无法得到 A、B 间传送的明文信息。

1. 明文消息到椭圆曲线上的嵌入

在使用椭圆曲线构造密码前，需要将明文消息镶嵌到椭圆曲线上，作为椭圆曲线上的点。设明文消息是 $m(0\leqslant m\leqslant M)$，给一个足够大的整数 k，假如取 $k=30$，对明文消息 m，计算一系列的 x：$x=\{mk+j,j=0,1,2,\cdots\}=\{30m,30m+1,30m+2,\cdots\}$，直到 $x^3+ax+b \pmod p$ 是平方根，即得到椭圆曲线上的点 $\sqrt{x^3+ax+b}$，反过来，为了从椭圆曲线上的点 (x,y) 得到明文消息 m，只需要计算 $m=\left\lfloor\dfrac{x}{30}\right\rfloor$。

【例 2-27】 设椭圆曲线为 $y^2=x^3+3x$，$p=4177$，$m=2174$，则 $x=\{30\times 2174+j,j=0,1,2,\cdots\}$。当 $j=15$ 时，$x=30\times 2174+15=652\ 35$，$x^3+3x=65\ 235^3+3\times 65\ 235=1444 \bmod 4177=38^2$，所以得到椭圆曲线上的点为(65 235,38)。

由椭圆曲线上的点(65 235,38)，则可求得明文信息 m，$m=\left\lfloor\dfrac{65\ 235}{30}\right\rfloor=\lfloor 2174.5\rfloor=2174$。

2. 椭圆曲线密钥交换算法

(1) 双方协商两个公开的参数,素数域 GF(p)上的椭圆曲线 E 和基点 P。

(2) A 选择一个保密的随机数 x,并计算出 $x\in[1,n-1]$,$R_1=x\times P$。

(3) B 选择一个保密的随机数 y,并计算出 $y\in[1,n-1]$,$R_2=y\times P$

(4) A 将 R_1 发送给 B; B 将 R^2 发送给 A。

(5) A 计算出 $K_1=x\times R_2$; B 计算出 $K_2=y\times R_1$,K_1、K_2 即为双方协商的密钥。

$\because K_1=x\times R_2=x\times(y\times P)=y\times(x\times P)=y\times R_1=K_2$

$\therefore K_1=K_2$

【例 2-28】 设有曲线,$E_{13}(1,6)$,在曲线上取一点 $P(2,7)$,求 A、B 协商的公钥 Q。

解: 取 A 的随机数 x 为 1,B 的随机数 y 为 7,则:

$R_1=x\times P=1\times(2,7)=(2,7)$,　$R_2=y\times P=7\times(2,7)=(7,2)$

$K_1=x\times R_2=1\times(7,2)=(7,2)$,　$K_2=y\times R_1=7\times(2,7)=(7,2)$

$\therefore K_1=K_2\therefore$A、B 所协商的公钥 $Q=K_1=K_2$

2.5　数据完整性认证

数据加密只是实现了密码学要求的机密性,密码学还同时要求提供其他方面的功能:身份认证、完整性和抗抵赖性。这些功能是人类通过计算机进行社会交流的至关重要的需求。

(1) 身份认证: 数据的接收者应该能够确认数据的来源,入侵者不可能伪装成他人。

(2) 完整性: 数据的接收者应该能够验证在传送过程中数据没有被修改,入侵者不可能用假数据代替合法数据。

(3) 抗抵赖性: 发送数据者事后不可能虚假地否认他发送的数据。

认证的目的是为了防止传输和存储的数据被有意无意地篡改,包括数据内容认证(即数据完整性认证)、数据的源和宿认证(即身份认证)及操作时间认证(即时间戳)等。它在税务的金税系统、银行的支付密码器等票据防伪中具有重要应用。

数据内容认证使用数据摘要方法(如 MD5)实现; 身份认证是识别通信对方的身份,以防止假冒,使用数字签名方法实现。

2.5.1　MD5 算法

消息摘要又称数字摘要(Digital Digest)。它是一个唯一对应一个数据的固定长度的值,它由一个单向 Hash 函数对数据进行作用而产生。如果数据在途中改变了,则接收者通过对收到数据的新产生的摘要与原摘要比较,就可知道数据是否被改变了。因此数据摘要保证了数据的完整性。

最常用的单向 Hash 算法是 MD5,全称是 Message Digest Algorithm 5(数据摘要算法 5),MD5 是经 MD2、MD3 和 MD4 发展而来。MD5 算法的使用不需要支付任何版权费用,可用于提供数据的完整性保护,确保信息传输完整一致。

MD5具有以下特征:

(1) 唯一性。单向Hash函数将需加密的明文摘要成一串128位的密文。它有固定的长度,且不同的明文摘要成密文,其结果总是不同的,输入的微小变化将导致输出的巨大变化(称为雪崩效应),同样的明文其摘要必定一致。这样摘要便可成为验证明文是否是在传输、存储过程中发生改变的"指纹"。

任何人都有自己独一无二的"指纹",这常常成为公安机关鉴别罪犯身份最值得信赖的方法;与之类似,MD5可以为任何信息(不管其大小、格式、数量)产生一个独一无二的MD5值,如果任何人对信息做了任何改动,其MD5值都会发生变化,从而实现数据完整性检测。因此MD5值也称为信息或文件的数字指纹(Digital Finger Print)。

(2) 单向性。根据128位的输出结果不可能反推出输入的信息(不可逆,单向性);MD5将任意长度的字节串映射为一个128位的大整数,并且通过该128位整数反推原始字符串是困难的,换句话说就是,即使你看到源程序和算法描述,也无法将一个MD5的值变换回原始的字符串,从数学原理上说,是因为原始的字符串有无穷多个,这有点像不存在反函数的数学函数。

(3) 结果定长。输入任意长度的信息,经过处理后均输出128位的MD5值,易于处理。例如:

MD5 ("") = d41d8cd98f00b204e9800998ecf8427e

MD5 ("a") = 0cc175b9c0f1b6a831c399e269772661

MD5 ("abc") = 900150983cd24fb0d6963f7d28e17f72

MD5 ("message digest") = f96b697d7cb7938d525a2f31aaf161d0

MD5 ("abcdefghijklmnopqrstuvwxyz") = c3fcd3d76192e4007dfb496cca67e13b

MD5 ("ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz") = f29939a25efabaef3b87e2cbfe641315

对于特定的文件而言,数据摘要是唯一的。数据摘要可以被公开,它不会透露相应文件的任何内容。

MD5的作用是让大容量数据在用数字签名软件签署私人密钥前被压缩成一种保密的格式(就是把一个任意长度的字节串变换成一定长的十六进制数字符串)。除了MD5以外,比较有名的还有sha-1、RIPEMD以及Haval等。

MD5算法以512位分组作为输入的信息,且每一分组又被划分为16个32位子分组,经过了一系列的处理后,算法的输出结果由4个32位分组组成,将这4个32位子分组级联后将生成一个128位散列值(即128位的数据摘要)。

总体流程为:

(1) 填充。如果输入信息的长度对512求余的结果不等于448,就需要填充使得对512求余的结果等于448。填充的方法是填充1个1和n个0。填充完后,信息的长度就为$n\times 512+448$(位),n为一个非负整数,可以是零;

(2) 记录信息长度。用64位来存储填充前信息长度。这64位加在第一步结果的后面,这样信息长度就变为$n\times 512+448+64=(n+1)\times 512$位。

(3) 装入标准的幻数(4个整数):$A=(01234567)_{16}$,$B=(89ABCDEF)_{16}$,$C=(FEDCBA98)_{16}$,$D=(76543210)_{16}$。如果在程序中定义应该是($A=0X67452301L$,$B=$

0XEFCDAB89L，C=0X98BADCFEL，D=0X10325476L）。

（4）4 轮循环运算：循环的次数是分组的个数（n+1）。

① 将每一 512 字节细分成 16 个小组，每个小组 64 位（8 个字节）。

② 先认识 4 个线性函数（& 是与，| 是或，～是非，^ 是异或）。

$F(X,Y,Z)=(X\&Y)|((\sim X)\&Z)$，$F$ 是一个逐位运算的函数。即，如果 X，那么 Y，否则 Z。

$G(X,Y,Z)=(X\&Z)|(Y\&(\sim Z))$

$H(X,Y,Z)=X\wedge Y\wedge Z$，函数 H 是逐位奇偶操作符。

$I(X,Y,Z)=Y\wedge(X|(\sim Z))$

这 4 个函数的说明：如果 X、Y、Z 的对应位是独立和均匀的，那么结果的每一位也应是独立和均匀的。

③ 设 M_j 表示数据的第 j 个子分组（从 0～15），$<s$ 表示循环左移 s 位，4 种操作为：

FF(a,b,c,d,M_j,s,t_i)表示 $a=b+((a+F(b,c,d)+M_j+t_i)<s)$

GG(a,b,c,d,M_j,s,t_i)表示 $a=b+((a+G(b,c,d)+M_j+t_i)<s)$

HH(a,b,c,d,M_j,s,t_i)表示 $a=b+((a+H(b,c,d)+M_j+t_i)<s)$

II(a,b,c,d,M_j,s,t_i 表示 $a=b+((a+I(b,c,d)+M_j+t_i)<s)$

假设 M_j 表示数据的第 j 个子分组（从 0～15），常数 t_i 是 4 294 967 296×abs(sin(i))的整数部分，i 取值从 1～64，单位是弧度（4 294 967 296 等于 2^{32}）。

第一分组需要将上面 4 个幻数复制到另外 4 个变量中：A 到 a，B 到 b，C 到 c，D 到 d。从第二分组开始的变量为上一分组的运算结果。

主循环有 4 轮，每轮循环都很相似。第一轮进行 16 次操作。每次操作对 a、b、c、d 中的其中 3 个作一次非线性函数运算，然后将所得结果加上第四个变量。

所有这些完成之后，将 A、B、C、D 分别加上 a、b、c、d，然后用下一分组数据继续运行算法，最后的输出是 A、B、C、D 的级联。

MD5 算法的优势在于使用不需要支付任何版权费用，所以在非绝密应用领域应用广泛。2004 年 8 月 17 日，我国的王小云院士破译 MD5、HAVAL-128、MD4 和 RIPEMD 算法，公布了 MD 系列算法的破解结果。

这种破解并非是真正的破解，只是加速了杂凑冲撞。杂凑冲撞是指两个完全不同的数据经杂凑函数计算得出完全相同的杂凑值。根据鸽巢原理，以有长度限制的杂凑函数计算没有长度限制的数据必然会有冲撞情况出现。一直以来，专家都认为要任意制造出冲撞需要太长时间，在实际情况中不可能发生，而王小云等的发现打破了这个必然性。2009 年，冯登国院士、谢涛二人利用差分攻击，将 MD5 的碰撞算法复杂度从王小云的 2^{42} 进一步降低到 2^{21}，极端情况下甚至可以降低至 2^{10}。2^{21} 的复杂度意味着在目前计算机上，只要几秒便可以找到一对碰撞。

王小云院士的研究成果具有重要意义。它表明从理论上讲电子签名可以伪造，必须及时添加限制条件，或者重新选用更为安全的密码标准，以保证电子商务的安全。

2.5.2　MD5 应用

数据内容认证时，采用数据摘要函数计算数据摘要，将数据摘要用发送方的私钥加密后

和原数据一起发送至目的端,目的端通过执行相应操作,就可实现数据认证。

1. 完整性验证(一致性验证)

典型应用是发送一个文件前,先计算其MD5的输出结果a,将a附加在文件后面发送;对方收到文件后,重新计算其MD5的输出结果b;如果a与b一样,就代表文件在传输中途未被篡改。这样,一旦这个文件在传输过程中,其内容被损坏或者被修改,那么这个文件的MD5值就会发生变化,通过对文件MD5的验证,可以得知获得的文件是否完整。

UNIX,软件下载的时候都有一个文件名相同,文件扩展名为.md5的文件,在这个文件中通常只有一行文本,例如MD5(tanajiya.tar.gz)=0ca175b9c0f726a831d895e269332461。这就是tanajiya.tar.gz文件的数字签名。MD5将整个文件当作一个大文本信息,通过其不可逆的字符串变换算法,产生了这个唯一的MD5数据摘要。它的作用就在于我们可以在下载该软件后,对下载回来的文件用专门的软件(如Windows MD5 Check等)做一次MD5校验,以确保我们获得的文件与该站点提供的文件为同一文件。

在网上下载软件时,为了防止不法分子在软件中添加木马,应在网站上公布由软件得到的MD5值供下载用户验证软件的完整性。

2. 安全访问认证

MD5广泛用于操作系统(UNIX、各类BSD系统)的登录认证上。在UNIX系统中,用户的密码是以MD5(或其他类似的算法)运算后存储在文件系统中的。当用户登录时,系统把用户输入的密码进行MD5运算,然后再去和保存在文件系统中的MD5值进行比较,进而确定输入的密码是否正确。通过这样的步骤,系统在并不知道用户密码明文的情况下就可以确定用户登录系统的合法性。这可以避免用户的密码被具有系统管理员权限的用户知道,而且还在一定程度上增加了破解密码的难度。这种加密技术被广泛的应用于UNIX系统中,这也是为什么UNIX系统比一般操作系统更为安全的一个重要原因。

现在很多网站在数据库存储用户的密码的时候都是存储用户密码的MD5值。这样就算不法分子得到数据库用户密码的MD5值,也无法知道用户密码。

MD5密码如何破解?比较有效的办法是使用系统中的md5()函数重新设一个密码,如admin,把生成的一串密码的Hash值覆盖原来的Hash值就行了。

正是因为这个原因,现在被黑客使用最多的破译密码的方法就是一种被称为跑字典的方法。先用MD5程序计算出这些字典项的MD5值,然后再用目标的MD5值在这个字典中检索。有两种方法得到字典:一种是搜集用作密码的字符串表,另一种是用排列组合方法生成。假设密码的最大长度为8字节(8 Bytes),同时密码只能是字母和数字,共26+26+10=62个字符,排列组合出的字典的项数则是$P(62,1)+P(62,2)+\cdots+P(62,8)$,已经是一个很天文的数字了,存储这个字典就需要TB级的磁盘阵列,而且这种方法还有一个前提,就是能获得目标账户的密码MD5值的情况下才可以。因此直接使用这种方法的效率不高。

3. 数据内容认证

数据内容认证常用的方法:发送者在数据中加入一个鉴别码并经加密后发送给接收者,接收者利用约定的算法对解密后的数据进行鉴别运算重新计算鉴别码,将得到的鉴别码与收到的鉴别码进行比较,若二者相等则接收,否则拒绝接收。

相对于密码系统，认证系统更强调的是完整性。消息由发送者发出后，经由密钥控制或无密钥控制的认证编码器变换，加入认证码，将数据连同认证码一起在公开的无扰信道进行传输，有密钥控制时还需要将密钥通过一个安全信道传输至接收方。接收方在收到所有数据后，经由密钥控制或无密钥控制的认证译码器进行认证，判定数据是否完整。

数据在整个过程中以明文形式或某种变形方式进行传输，不要求加密，也不要求内容对第三方保密。

攻击者能够截获和分析信道中传送的数据内容，而且可能伪造数据送给接收者进行欺诈。攻击者不再像保密系统中的密码分析者那样始终处于消极被动地位，而是主动攻击者。

在数据认证中，数据源和宿的常用认证方法有两种：

(1) 通信双方事先约定发送数据的数据加密密钥，接收者只需要证实发送来的数据是否能用该密钥还原成明文就能鉴别发送者。如果双方使用同一个数据加密密钥，那么只需在数据中嵌入发送者识别符即可。

(2) 通信双方事先约定各自发送数据所使用的通行字，发送数据中含有此通行字并进行加密，接收者只需判别数据中解密的通行字是否等于约定的通行字就能鉴别发送者。为了安全起见，通行字应该是可变的。

数据认证中常见的攻击和对策。

(1) 重放攻击：截获以前协议执行时传输的信息，然后在某个时候再次使用。对付这种攻击的一种措施是在认证数据中包含一个非重复值，如序列号、时戳、随机数或嵌入目标身份的标志符等。

(2) 冒充攻击：攻击者冒充合法用户发布虚假数据。为避免这种攻击可采用身份认证技术。

(3) 重组攻击：把以前协议执行时一次或多次传输的信息重新组合进行攻击。为了避免这类攻击，把协议运行中的所有数据都连接在一起。

(4) 篡改攻击：修改、删除、添加或替换真实的数据。为避免这种攻击，可采用数据认证码 MAC 或 Hash 函数等技术。

2.6　数 字 签 名

2.6.1　数字签名定义

在计算机通信中，当接收者接收到数据时，往往需要验证数据在传输过程中有没有被篡改；有时接收者需要确认数据发送者的身份。所有这些都可以通过数字签名来实现。

数字签名是公开密钥加密技术的一种应用。其使用方式是：报文的发送方从报文文本中生成一个 128 位的散列值(即 Hash 函数值，有时这个单向值也叫做报文摘要，与报文的数字指纹或标准校验相似)来验证发送报文的完整性和不可抵赖性。

数字签名可以用来证明数据确实是由发送者签发的，而且当数字签名用于存储的数据或程序时，可以用来验证数据或程序的完整性。它和传统的手写签名类似，应满足以下条件。

- 签名是可以被确认的，即接收方可以确认或证实签名确实是由发送方签名的。

- 签名是不可伪造的,即接收方和第三方都不能伪造签名。
- 签名不可重用,即签名是数据(数据文件)的一部分,不能把签名移到其他数据上。
- 签名是不可抵赖的,即发送方不能否认他所签发的数据。
- 第三方可以确认收发双方之间的数据传送但不能篡改数据。

1. 基于密钥的数字签名

使用对称密码系统可以对文件进行签名,但此时需要可信任的第三方仲裁。设 CA 是 A 和 B 共同依赖的仲裁人。K_A 和 K_B 分别是 A 和 B 与 CA 之间的密钥,而 K_{CA}是只有 CA 掌握的密钥,P 是 A 发给 B 的数据,t 是时间戳。CA 解读了 A 的报文$\{A,K_A(B,t,P)\}$以后产生了一个签名的数据 $K_{CA}(A,t,P)$,并装配成发给 B 的报文$\{K_B(A,t,P,K_{CA}(A,t,P))\}$。B 可以解读该报文,阅读数据 P,并保留 $K_{CA}(A,t,P)$。

由于 A 和 B 之间的通信是通过中间人 CA 的,所以不必怀疑对方的身份。又由于证据 $K_{CA}(A,t,P)$的存在,A 不能否认发送过数据 P,B 也不能改变得到的数据 P,因为 CA 仲裁时可能会当场解密 $K_{CA}(A,t,P)$,得到发送人、发送时间和原来的数据 P。

2. 基于公钥的数字签名

利用公钥加密算法的数字签名系统如图 2-17 所示。如果 A 否认了,B 可以拿出 $D_A(P)$,并用 A 的公钥 E_A 解密得到 P,从而证明 P 是 A 发送的。如果 B 把数据篡改了,当 A 要求 B 出示原来的 $D_A(P)$时,B 拿不出来。

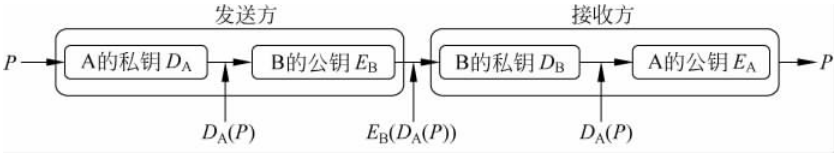

图 2-17 基于公钥的数字签名

在实际应用中,由于公开密钥算法的效率较低,发送方并不对整个文件签名,而只对文件的 Hash 值签名。

3. 直接方式的数字签名技术

直接方式的数字签名只有通信双方参与,并假定接收方知道发送方的公开密钥。数字签名的形成方式可以用发送方的密钥加密整个数据。

如果发送方用接收方的公开密钥(公钥加密体制)或收发双方共享的会话密钥(密钥加密体制)对整个数据及其签名进一步加密,那么对数据及其签名提供了更高保密性。而此时的外部保密方式(即数字签名是直接对需要签名的数据生成而不是对已加密的数据生成,否则称为内部保密方式)则对解决争议十分重要,因为在第三方处理争议时,需要得到明文数据及其签名才行。但如果采用内部保密方式,那么,第三方必须在得到数据的解密密钥后才能得到明文数据。如果采用外部保密方式,那么,接收方就可将明文数据及其数字签名存储下来以备之后可能出现的争议时使用。

直接方式的数字签名有一个弱点,即方案的有效性取决于发送方密钥的安全性。如果发送方想对自己已发出的数据予以否认,就可声称自己的密钥已丢失被盗,认为自己的签名是他人伪造的。对这一弱点可采取某些行政手段,在某种程度上可减弱这种威胁,如要求每

一个被签名的数据都包含有一个时间戳(日期和时间),并要求密钥丢失后立即向管理机构报告。这种方式数字签名还存在发送方的密钥真的被偷的危险,例如,敌方在时刻 T 获得发送方的密钥,然后可伪造一个数据,用偷得的密钥为其签名并加上 T 以前的时刻作为时间戳。

4. 安全 Hash 函数

通过单向 Hash 函数生成数字摘要,并用单向检验和函数 CK 对其作用,计算出 CK(M),发送者把这一 CK(M)和原数据一起发送到目的方。其实现过程总结如下:

(1) 被发送明文先用 MD5 方式产生 128 位的数字摘要;

(2) 发送方用自己的私有密钥对摘要加密,形成"数字签名";

(3) 将原文 m 和加密的摘要同时传给对方;

(4) 接收方用发送方的公钥对摘要进行解密,同时对接收到的文件用 MD5 编码重新产生摘要;

(5) 接收方将解码后的摘要和收到的原明文重新用 MD5 加密产生的摘要进行对比,如果两者一致,则明文信息在传递过程中没有被破坏或篡改。

采用公钥密码体制和单向 Hash 函数进行数字签名的具体过程如图 2-18 所示。

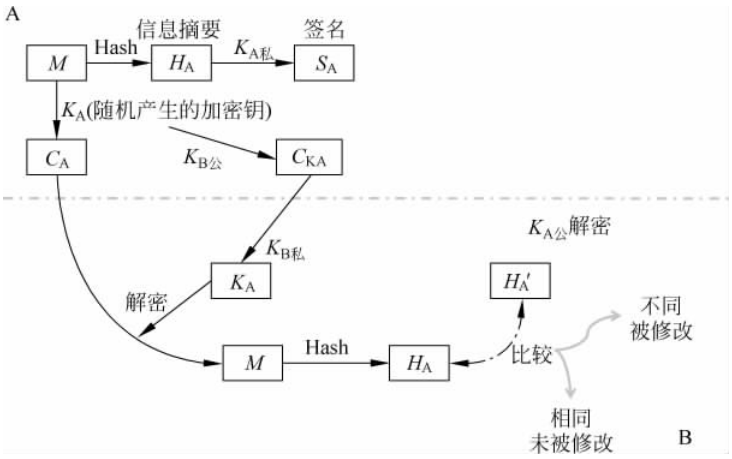

图 2-18　采用公钥密码体制和单向 Hash 函数进行的数字签名过程

(1) 将发送的信息 M 经过 Hash 运算产生信息 M 的数据摘要 H_A,将该数据摘要经过 A 的私钥 $K_{A私}$ 加密后产生 A 的签名 S_A;

(2) 将 A 要发送的信息用 A 随机产生的对称密钥 K_A 进行加密,产生密文 C_A;将 A 随机产生的密钥 K_A 用 B 的公钥 $K_{B公}$ 进行加密,得到加密的密钥 C_{KA};

(3) 用户 A 将签名 S_A、密文 C_A 和加密后的密钥 C_{KA} 发送给 B。

(4) 用户 B 收到这些信息后,先用 B 的私钥 $K_{B私}$ 将发送过来的加密密钥 C_{KA} 解密后得到密钥 K_A;

(5) 用该密钥解密密文 C_A 得到信息明文 M;对明文信息 M 计算其数据摘要得到摘要信息 H_A;将接收到的签名信息 S_A 用用户 A 的公钥 $K_{A公}$ 解密得到由用户 A 计算出的数据摘要,记为 H'_A。

(6) 用户 B 对两个数据摘要 H_A 和 H'_A 进行比较,若相同,则证明信息发送过程中未发

生任何改变;若不同,则有人进行了修改。

在这种签名机制中,用户B完全可以相信所得到的信息一定是用户A发送过来的,同时用户A也无法否认发送过信息,因此是一种安全的签名技术方案。

整个过程使用了3对密钥:用户A的非对称密钥$K_{A私}$和$K_{A公}$、用户B的非对称密钥$K_{B私}$和$K_{B公}$、A随机产生的对称密钥K_A。其中用户A的非对称密钥用于数字签名,实现用户A无法否认发送过信息;用户B的非对称密钥利用非对称密钥难于解密的优点对随机产生的对称密钥K_A进行加密、解密,保证对称密钥的安全分发;随机产生的对称密钥K_A利用对称密钥加密速度快的优点实现对大量的数据进行加密、解密。

混合密钥算法:使用公钥算法加密随机产生的对称密钥,利用公钥算法难以解密的优点进行对称密钥的安全分发;使用对称密钥加密速度快的优点加密传输数据,实现一次一密,保证信息传输保密性。

混合密钥算法是目前数据加解密系统普遍采用的方法,必须牢固掌握。

2.6.2 时间戳

在交易文件中,时间是十分重要的因素,需要对电子交易文件的日期和时间采取安全措施,以防文件被伪造或篡改。时间戳服务是计算机网络上的安全服务项目,由专门机构提供。

时间戳是一个经加密后形成的凭证文档,它包括以下3部分:

(1) 需要加时间戳的文件的数据摘要(digest);

(2) ETS(Electronic Timestamp Server,电子时间戳服务器)收到文件的日期和时间;

(3) ETS的数字签名。

时间戳产生的过程为:用户首先将需要加时间戳的文件用Hash编码加密形成摘要,然后将该摘要发送到DTS(Digital Time-stamp Service,数字时间戳服务)机构,DTS在加入了收到文件摘要的日期和时间信息后再对该文件加密(数字签名),最后送回用户。

DTS工作过程:加密时将摘要信息归并到二叉树的数据结构,再将二叉树的根值发表在报纸上,这样便有效地为文件发表时间提供了佐证。注意,书面签署文件的时间是由签署人自己写上的,而数字时间戳则不然,它是认证单位DTS加上的,以DTS收到文件的时间为依据。因此,时间戳也可以作为科学发明文献的时间认证。

2.7 公钥基础设施

越来越多的企业网和电子商务使用Internet作为通信基础平台,但都面临一个问题,就是如何建立相互之间的信任关系以及如何保证信息的真实性、完整性、机密性和不可否认性。PKI(Public Key Infrastructure,公钥基础设施)是解决这一系列问题的技术基础。

2.7.1 PKI定义

AB通信模型存在公钥替换问题:Alice使用自己的私钥签名和Bob通信,实现身份认证。但敌手Trudy出现了,他偷偷使用了Alice的计算机,用自己的公钥换走了Bob的公钥。此时,Alice实际拥有的是Trudy的公钥,但是还以为这是Bob的公钥。因此,Trudy就

可以冒充 Bob，用自己的私钥进行数字签名，写信给 Alice，让 Alice 用假的 Bob 公钥进行解密。

Alice 如何确定公钥是否真的属于 Bob？解决方法是引入第三方认证机构——认证中心(Certificate Authority，CA)。方法是要求 Bob 去找 CA 为公钥做认证：CA 用自己的私钥，对 Bob 的公钥和一些相关信息一起加密，生成数字证书(Digital Certificate)。

Bob 拿到数字证书以后再给 Alice 写信，只要在签名的同时，附上数字证书就可以了。Alice 收信后，用 CA 的公钥解开数字证书，就可以拿到 Bob 真实的公钥了，然后就能证明数字签名是否真的是 Bob 签的。这是因为 CA 以自己的信誉证明公钥的真实性(公钥持有人身份真实)，这就和我们使用身份证证明自己一样，公安部门相当于 CA。

必要时我们可以要求 CA 提供数字证书的真实性证明，就像坐飞机、火车进行安检一样，警察会比对我们身份证是否和原始身份证数据库一致。

与此类似，网络安全中身份验证和密钥协商要求的前提是合法的服务器掌握着对应的私钥。但服务器公钥并不包含服务器的信息，RSA 算法无法确保服务器身份的合法性，存在中间人攻击和信息抵赖的安全隐患，如图 2-19 所示。

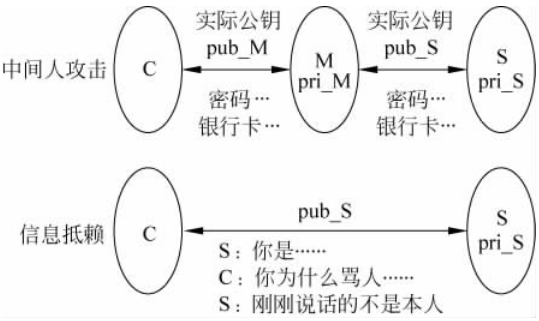

图 2-19　中间人攻击和信息抵赖

(1) 客户端 C 和服务器 S 进行通信，中间节点 M 截获了二者的通信；

(2) 中间节点 M 自己计算产生一对公钥 pub_M 和私钥 pri_M；

(3) C 向 S 请求公钥时，M 把自己的公钥 pub_M 发给了 C；

(4) C 使用公钥 pub_M 加密的数据能够被 M 解密，因为 M 掌握对应的私钥 pri_M，而 C 无法根据公钥信息判断服务器的身份，从而 C 和 M 之间建立了虚假的“可信”加密连接；

(5) 中间节点 M 和服务器 S 之间再建立合法的连接，因此 C 和 S 间通信被 M 完全掌握，M 可以进行信息的窃听、篡改等操作。

另外，服务器也可以对自己的发出的信息进行否认，不承认相关信息是自己发出。

解决上述身份验证问题的关键是确保获取的公钥途径是合法的，能够验证服务器的身份信息，为此需要引入权威的第三方机构 CA。CA 负责核实公钥的拥有者的信息，并颁发认证证书，同时能够为使用者提供证书验证服务，即 PKI 体系。

PKI 的基本原理为：CA 负责审核信息，然后对关键信息利用私钥进行签名，公开对应的公钥，客户端可以利用公钥验证签名。

PKI 就是利用公钥理论和技术建立的，为网络的数据和其他资源提供信息安全服务的基础设施。从广义上说，所有提供公钥加密和数字签名服务的系统都可以叫做 PKI 系统。

PKI 的实质是利用一对互相匹配的密钥进行加密、解密。PKI 的主要目的是通过自动

管理密钥和证书,为用户建立起一个安全的网络运行环境,使用户可以在多种应用环境下方便地使用加密和数字签名技术,从而保证网络通信中数据的机密性、完整性、有效性。

一个有效的PKI系统在提供安全性服务的同时,在应用上还应该具有简单性、透明性,即用户在获得加密和数字签名服务时,不需要详细了解PKI内部的实现原理和具体操作,如PKI怎样管理证书和密钥等。

目前被广泛认可的PKI是以X.509第三版为基础的结构。PKI所带来的保密性、完整性、不可否认性的重要意义日益突出。

PKI的概念和内容是动态的、不断发展的,完整的PKI系统必须具有权威认证机关(CA)、数字证书库、密钥备份及恢复系统、证书作废系统等基本组成部分,构建PKI也将围绕着五大系统来着手。

认证机构(CA):即数字证书的申请及签发机关,CA必须具备权威性这一特征,它是PKI的核心;

数字证书库:用于存储已签发的数字证书及公钥,用户可由此获得所需的其他用户的证书及公钥;

密钥备份及恢复系统:如果用户丢失了用于解密数据的密钥,则数据将无法被解密,这将造成合法数据丢失。为避免这种情况,PKI提供备份与恢复密钥的机制。但必须注意,密钥的备份与恢复必须由可信的机构来完成;并且,密钥备份与恢复只能针对解密密钥,签名私钥为确保其唯一性而不能够作备份。

证书作废系统:证书作废处理系统是PKI的一个必备的组件。与日常生活中的各种身份证件一样,证书有效期以内也可能需要作废,原因可能是密钥媒体丢失或用户身份变更等。为实现这一点,PKI必须提供作废证书的一系列机制。

应用接口(API):PKI的价值在于使用户能够方便地使用加密、数字签名等安全服务,因此一个完整的PKI必须提供良好的应用接口系统,使得各种各样的应用能够以安全、一致、可信的方式与PKI交互,确保安全网络环境的完整性和易用性。客户端软件的安装就可使客户方便地使用PKI系统。

对于构建密码服务系统的核心内容是如何实现密钥管理,公钥体制涉及一对密钥(即私钥和公钥),私钥只由用户独立掌握,无须在网上传输,而公钥则是公开的,需要在网上传送,故公钥体制的密钥管理主要是针对公钥的管理问题,目前较好的解决方案是数字证书机制。

2.7.2 认证中心

认证中心在网络通信认证中具有特殊的地位。例如,在进行电子商务时,认证中心是为了从根本上保障电子商务顺利进行而设立的,主要是解决电子商务活动中参与各方的身份、资质的认定,维护交易活动的安全。

CA机构又称为证书授证(Certificate Authority)中心,作为电子商务交易中受信任和具有权威性的第三方,承担公钥体系中公钥的合法性检验的责任。

CA中心为每个使用公开密钥的客户发放数字证书,数字证书的作用是证明证书中列出的客户合法拥有证书中列出的公开密钥。CA机构的数字签名使得第三者不能伪造和篡改证书。它负责产生、分配并管理所有参与网上信息交换各方所需的数字证书,因此是安全电子信息交换的核心。

CA 是提供身份验证的第三方机构，通常由一个或多个用户信任的组织实体组成。例如，持卡人要与商家通信，持卡人从公开媒体上获得了公开密钥，但无法确定不是冒充的，于是请求 CA 对商家认证。此时，CA 对商家进行验证，其过程由持卡人→商家转变为持卡人→CA；CA→商家。证书一般包含拥有的标识名称和公钥，并且由 CA 进行数字签名。

CA 的功能主要有接收注册申请、处理、批准/拒绝请求、颁发证书。

在实际动作中，CA 也可由大家都信任的一方担当，例如，在客户、商家、银行三角关系中，客户使用的是由某个银行发的卡，而商家又与此银行有业务关系（有账号）。在此情况下，客户和商家都信任该银行，可由该银行担当 CA 角色，接收和处理客户证书的验证请求。又如，对商家自己发行的购物卡，则可由商家自己担当 CA 角色。

如何知道在每次通信或交易中所使用的密钥对实际上就是用户的密钥对呢？这就需要一种认证公用密钥和用户之间关系的方法。

解决这一问题的方法是引入一种叫做证书或凭证的特种签名信息。数字签名很重要的机制是数字证书（Digital Certificat，或 Digital ID），数字证书又称为数字凭证，是用电子手段来证实一个用户的身份和对网络资源访问的权限。在网上的电子交易中，如双方出示了各自的数字凭证，并用它来进行交易操作，那么双方都可不必为对方身份的真伪担心。数字凭证可用于电子邮件、电子商务、群件和电子基金转移等各种用途。

数字证书是一个经证书授权中心数字签名的包含公开密钥拥有者信息以及公开密钥的文件。人们可以在互联网交往中用它来识别对方的身份。在数字证书认证的过程中，证书认证中心（CA）作为权威的、公正的、可信赖的第三方，其作用是至关重要的。数字证书由独立的证书发行机构发布。数字证书各不相同，每种证书可提供不同级别的可信度。

公开密钥技术解决了密钥发布的管理问题。最简单的数字证书包含一个公开密钥、名称以及证书授权中心的数字签名。一般情况下，数字证书还包括密钥的有效时间、发证机关的名称和证书的序列号等信息，证书的格式遵循 ITU-T X.509 国际标准。

（1）版本号：用于区分 X.509 的不同版本。

（2）序列号：由同一发行者（CA）发放的每个证书的序列号是唯一的。

（3）签名算法：签署证书所用的算法及其参数。

（4）发行者：指建立和签署证书的 CA 的 X.509 名字。

（5）有效期：包括证书有效期的起始时间和终止时间。

（6）主体名：指证书持有者的名称及有关信息。

（7）公钥：有效的公钥以及其使用方法。

（8）发行者 ID：任选的，名字唯一标识证书的发行者。

（9）主体 ID：任选的，名字唯一标识证书的持有者。

（10）扩展域：添加的扩充信息。

（11）认证机构的签名：用 CA 私钥对证书的签名。

数字证书有着广泛的现实作用，分为以下 3 种类型：

（1）个人凭证（Personal Digital ID），它仅仅为某一个用户提供凭证，以帮助个人进行安全交易操作。个人身份的数字凭证通常是安装在客户端的浏览器中的，并通过安全的电子邮件来进行交易操作。

（2）企业凭证（Server ID），它通常为网上某个 Web 服务器提供凭证，拥有 Web 服务器

的企业就可以用具有凭证的Web站点来进行安全电子交易。有凭证的Web服务器会自动地将其与客户端Web浏览器通信的信息加密。

(3) 软件(开发者)凭证(Developer ID),它通常为Internet中被下载的软件提供凭证,该凭证用于微软公司的Authenticode技术中,以使用户在下载软件时能获得所需的信息。

数字证书的工作过程如下:

用户首先产生自己的密钥对,并将公共密钥及部分个人身份信息传送给认证中心。认证中心在核实身份后,将执行一些必要的步骤,以确信请求确实由用户发送而来,然后,认证中心将发给用户一个数字证书,该证书内包含用户的个人信息和他的公钥信息,同时还附有认证中心的签名信息。之后用户就可以使用自己的数字证书进行相关的各种活动了。

网络的每个用户必须知道CA公钥,这就使任何一个想验证证书的人能采用用于验证上述信息和数字证书的相同程序。CA的公用密钥以证书格式提供,因而它也是可以验证的。

CA签发并管理正式使用公用密钥与用户相关联的证书。证书只在某一时间内有效,因而CA会保存一份有效证书及其有效期清单。有时,证书或许要求及早废除,因而CA会保存一份废除的证书有及有效证书的清单。CA把其有效证书、废除证书或过期证书的清单提供给任何一个要获得这种清单的人。

如图2-20所示,CA使用流程为:

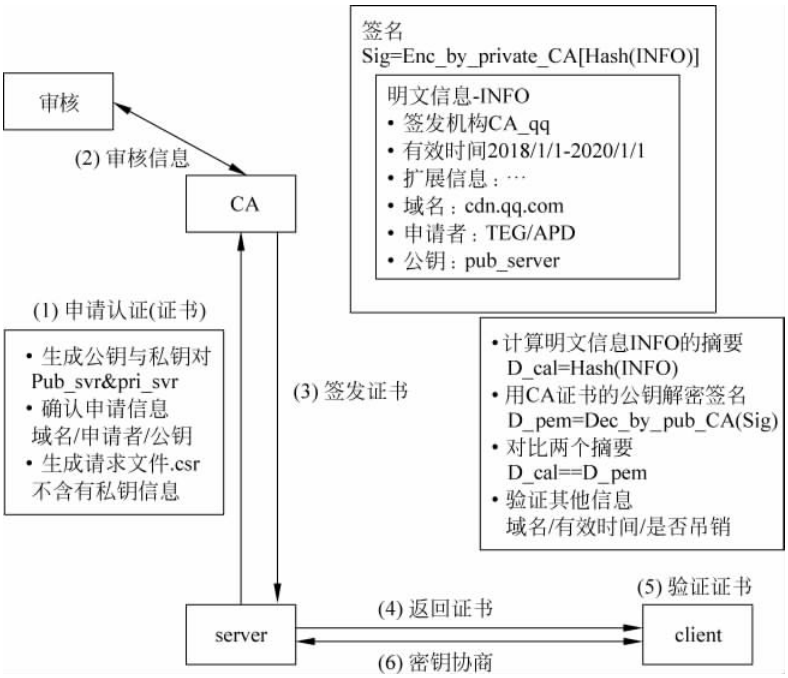

图2-20 CA使用流程

(1) 申请认证(证书)。服务器S向第三方机构CA提交公钥、组织信息、个人信息(域名)等信息并申请认证;

(2) 审核信息。CA通过线上、线下等多种手段验证申请者提供信息的真实性,如组织是否存在、企业是否合法,是否拥有域名的所有权等;

(3) 签发证书。如信息审核通过,CA 会向申请者签发认证文件——证书。

证书包含以下信息:申请者公钥、申请者的组织信息和个人信息、签发机构 CA 的信息、有效时间、证书序列号等信息的明文,同时包含一个签名;

签名的产生算法:首先使用 Hash 函数计算公开的明文信息的信息摘要,然后采用 CA 的私钥对信息摘要进行加密,密文即签名;

(4) 返回证书。客户端 C 向 S 发出请求时,S 返回证书文件;

(5) 验证证书。C 读取证书中相关的明文信息,采用相同的 Hash 函数计算得到信息摘要,然后,利用对应 CA 的公钥解密签名数据,对比证书的信息摘要,如果一致,则可以确认证书的合法性,即公钥合法;C 然后验证证书相关的域名信息、有效时间等信息;

客户端会预先内置信任 CA 的证书信息(包含公钥),如果 CA 不被信任,则找不到对应 CA 的证书,证书也会被判定非法。

在这个过程中应注意几点:

- 证书=公钥+申请者与颁发者信息+签名;
- 申请证书不需要提供私钥,确保私钥永远只能由服务器掌握;
- 证书的合法性仍依赖于非对称加密算法,证书主要是增加了服务器信息以及签名;
- 内置 CA 对应的证书称为根证书,颁发者和使用者相同,自己为自己签名,即自签名证书。

(6) 密钥协商。C 和 S 协商对称密钥。

下面看一个应用数字证书的实例:https 协议。

超文本传输协议 http 协议被用于在浏览器和网站服务器之间传递信息,http 协议以明文方式发送内容,不提供任何方式的数据加密,如果攻击者截取了浏览器和网站服务器之间的传输报文,就可以直接读懂其中的信息,因此,http 协议不适合传输一些敏感信息,例如信用卡号、密码等支付信息。

为了消除 http 协议的这一缺陷,需要使用另一种协议:安全超文本传输协议(Secure Hypertext Transfer Protocol)https。https 在 http 的基础上加入了 SSL 协议,SSL 依靠证书来验证服务器的身份,并为浏览器和服务器之间的通信加密,确保数据传输的安全。

https 能够加密信息,以免敏感信息被第三方获取,所以很多银行网站或电子邮箱等安全级别较高的服务都会采用 https 协议,百度、谷歌、淘宝等网站都已经使用了 https 进行保护,特征是浏览器左上角已经全部出现了一把绿色锁。iOS 9 系统默认把所有的 http 请求都改为 https 请求。现代互联网正在逐渐进入全网 https 时代。

https 通过在 http 上建立加密层,使用安全套接字层(SSL)对传输数据进行加密,用于在客户计算机和服务器之间交换信息。主要作用可以分为两种:一种是建立一个信息安全通道,来保证数据传输的安全;另一种就是确认网站的真实性。

简单来说,https 是 http 的安全版,是使用 TLS/SSL 加密的 http 协议,如图 2-21 所示。http 协议采用明文传输信息,存在信息窃听、信息篡改、信息劫持的风险;而 https 协议具有身份验证、信息加密和完整性校验的功能,采用密文传输信息,可以避免此类问题发生。

SSL(Secure Sockets Layer,安全套接层)及其继任者传输层安全(Transport Layer Security,TLS)是为网络通信提供安全及数据完整性的一种安全协议。TLS 与 SSL 在传输层对网络连接进行加密。

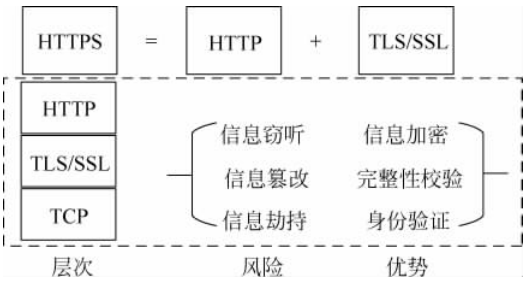

图 2-21 https 协议

https 协议的主要功能基本都依赖于 TLS/SSL 协议。TLS/SSL 是介于 TCP 和 http 之间的一层安全协议,不影响原有的 TCP 协议和 http 协议,所以使用 https 基本上不需要对 http 页面进行太多的改造。

如图 2-22 所示,TLS/SSL 的功能实现主要依赖于 3 类基本算法:Hash 函数算法、对称加密算法、非对称加密算法。其利用非对称加密算法实现身份认证和密钥协商,对称加密算法采用协商的密钥对数据加密,基于 Hash 函数验证信息的完整性。

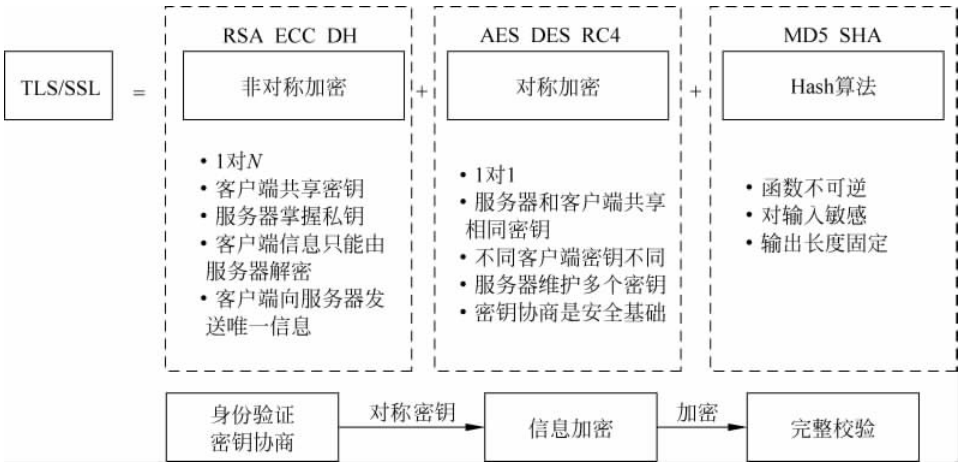

图 2-22 TLS/SSL 的功能实现

常见的 Hash 函数有 MD5、SHA1、SHA256,该类函数特点是函数单向不可逆、对输入非常敏感、输出长度固定,针对数据的任何修改都会改变 Hash 函数的结果,用于防止信息篡改并验证数据的完整性;对称加密算法常见的有 AES-CBC、DES、3DES、AES-GCM 等,相同的密钥可以用于信息的加密和解密,掌握密钥才能获取信息,能够防止信息窃听,通信方式是 1 对 1;非对称加密即常见的 RSA 算法,还包括 ECC、DH 等算法,算法特点是,密钥成对出现,公钥加密的信息只能私钥解开,私钥加密的信息只能公钥解开。因此掌握公钥的不同客户端之间不能互相解密信息,只能和掌握私钥的服务器进行加密通信,服务器可以实现1 对N 的通信,客户端也可以用来验证掌握私钥的服务器身份。

在信息传输过程中,Hash 函数不能单独实现信息防篡改,因为在明文传输中,中间人可以修改信息之后重新计算数据摘要,因此需要对传输的信息以及数据摘要进行加密;对称加密的优势是信息传输 1 对 1,需要共享相同的密码,密码的安全是保证信息安全的基

础，服务器和 N 个客户端通信，需要维持 N 个密码记录，且缺少修改密码的机制；非对称加密的特点是信息传输 1 对 N，服务器只需要维持一个私钥就能够和多个客户端进行加密通信，但服务器发出的信息能够被所有的客户端解密，且该算法的计算复杂，加密速度慢。

结合 3 类算法的特点，TLS 的基本工作方式是：客户端使用非对称加密与服务器进行通信，实现身份验证并协商对称加密使用的密钥，然后对称加密算法采用协商密钥对信息以及数据摘要进行加密通信，不同的节点之间采用的对称密钥不同，从而可以保证信息只能通信双方获取。

https 通常只要求服务器有一个证书。主要解决了以下问题：

(1) 保证服务器就是他声称的服务器(信任主机的问题)。采用 https 的服务器必须从 CA 申请一个用于证明服务器用途类型的证书。该证书只有用于对应的服务器的时候，客户机才信任此主机。客户通过信任该证书，从而信任了该主机。其实这样做效率很低，但是银行更注重安全。所以目前所有的银行系统网站，关键部分应用都是 https 的。

(2) 服务端和客户端之间的所有通信，都是加密的，防止通信过程中数据的泄密和被篡改。具体来讲，就是客户端产生一个对称密钥，通过服务器的证书来交换对称密钥；接下来所有的信息往来就都是加密的。第三方即使截获，也没有任何意义。因为他没有密钥。当然篡改也就没有什么意义了。

https 在对客户端有安全要求的情况下，也要求客户端必须有一个证书。

(1) 这里客户端证书，其实就类似表示个人信息的时候，除了用户名/密码，还有一个 CA 认证过的身份。因为个人证书一般来说是别人无法模拟的，所有这样能够更进一步确认自己的身份。

(2) 目前个人银行的专业版就是这种做法，具体证书可以用 U 盘作为载体(即 U 盾)。大多数网上银行就是采取这种方式。

https 的缺点是工作效率不高，这是因为：

(1) http 协议只需要简单的一次请求/响应就完成了信息获取，而 https 需要有密钥和确认加密算法，单握手就需要 6/7 次请求/响应。在任何应用中，过多的请求/响应肯定影响性能。

(2) 接下来具体的 http 协议每一次响应或者请求，都要求客户端和服务端对会话的内容做加密/解密。尽管对称加密/解密效率比较高，可是仍然要消耗过多的 CPU。如果 CPU 性能比较低，肯定会降低性能，从而不能对更多的请求提供服务。

综上所述，https 实际上就是 SSL over http，它使用默认端口 443，而不是像 http 那样使用端口 80 来和 TCP/IP 进行通信。

https 协议使用 SSL 在发送方对原始数据进行加密，然后在接收方进行解密，加密和解密需要发送方和接收方通过交换共知的密钥来实现，因此，所传送的数据不容易被网络黑客截获和解密。然而，加密和解密过程需要耗费系统大量的开销，严重降低机器的性能，相关测试数据表明使用 https 协议传输数据的工作效率只有使用 http 协议传输的十分之一。

假如为了安全保密，将一个网站所有的 Web 应用都启用 SSL 技术来加密，并使用 https 协议进行传输，那么该网站的性能和效率将会大大降低，而且没有这个必要，因为一般来说并不是所有数据都要求那么高的安全保密级别，所以，只需对那些涉及机密数据的交互处理使用 https 协议，就能做到鱼与熊掌兼得。

2.7.3 CA结构

证书管理有两种常用的结构：CA的分级系统和信任网。

在分级证明中，顶部即根CA，它验证它下面的CA，第二级CA再验证用户和它下属的CA，以此类推。

在信任网络中，用户的公用密钥能以任何一个为接收证书的人所熟悉的用户签名的证书形式提交。一个企图获取另一个公用密钥的用户可以从不同来源获取，并验证它们是否全部符合。

1. 证书链

如图2-23所示，CA根证书和服务器证书中间增加一级证书机构，即中间证书，证书的产生和验证原理不变，只是增加一层验证，只要最后能够被任何信任的CA根证书验证合法即可。

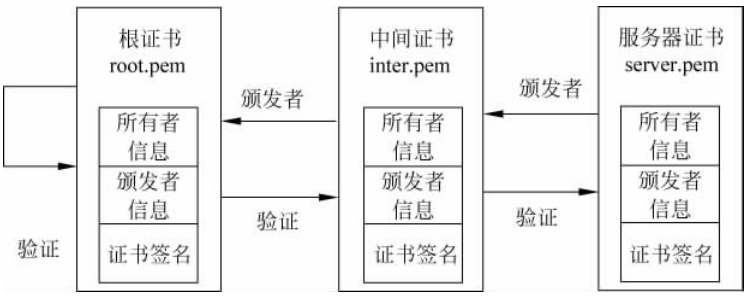

图2-23 CA证书链

(1) 服务器证书server.pem的签发者为中间证书机构inter，inter根据证书inter.pem验证server.pem确实为自己签发的有效证书；

(2) 中间证书inter.pem的签发CA为root，root根据证书root.pem验证inter.pem为自己签发的合法证书；

(3) 客户端内置信任CA的root.pem证书，因此服务器证书server.pem的被信任。

服务器证书、中间证书与根证书在一起组合成一条合法的证书链，证书链的验证是自下而上的信任传递的过程。

二级证书结构存在如下优势：

(1) 减少根证书结构的管理工作量，可以更高效地进行证书的审核与签发；

(2) 根证书一般内置在客户端中，私钥一般离线存储，一旦私钥泄露，则吊销过程非常困难，无法及时补救；

(3) 中间证书结构的私钥泄露，则可以快速在线吊销，并重新为用户签发新的证书；

(4) 证书链在四级以内一般不会对https的性能造成明显影响。

如图2-24所示，证书链具有以下特点：

(1) 同一本服务器证书可能存在多条合法的证书链。因为证书的生成和验证基础是公钥和私钥对，如果采用相同的公钥和私钥生成不同的中间证书，那么针对被签发者而言，该签发机构都是合法的CA，不同的是中间证书的签发机构不同。

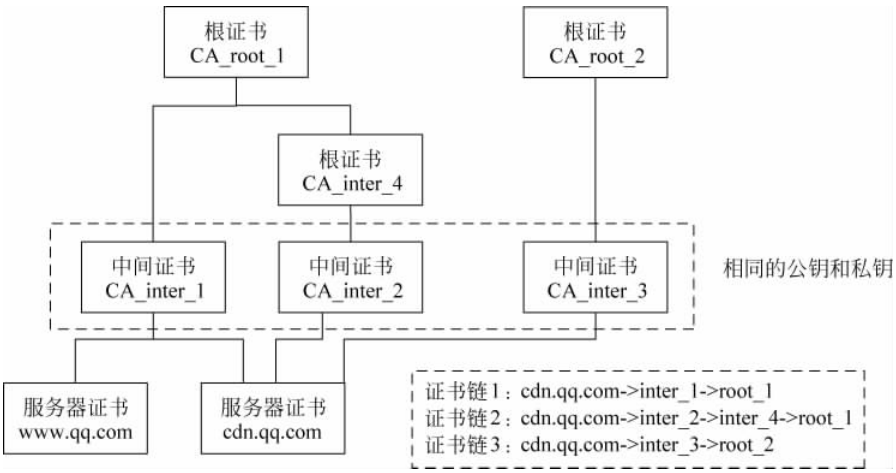

图 2-24　CA 证书链的特点

(2) 不同证书链的层级不一定相同,可能是二级、三级或四级证书链。中间证书的签发机构可能是根证书机构也可能是另一个中间证书机构。

2. CA 的主要功能

PKI 具有产生、验证和分发的密钥、签发和验证、获取证书、验证证书、保存证书、本地保存的证书的获取、证书的撤销、密钥的恢复、证书撤销列表(CRL)的获取、密钥的更新、审计以及存档等功能。这些功能大部分由 PKI 的核心组成部分 CA 完成。其中 CA 主要完成的功能有证书颁发、证书更新、证书和证书撤销列表的公布、证书状态的在线查询、证书认证和制定政策等。

(1) 证书颁发。申请者在 CA 的注册机构进行注册,申请证书。CA 对申请者进行审核,审核通过则生成证书,颁发证书给申请者。证书的申请可采取在线申请和亲自到注册机构申请两种方式。证书的颁发也可采取两种方式:一种是在线直接从 CA 下载;一种是 CA 将证书制作成媒体(如 IC 卡)后,由申请者带走。

(2) 证书更新。当证书持有者的证书过期、被窃取或丢失时,可通过更新证书的方式,使其使用新的证书,继续参与网上认证。证书的更新包括证书的更换和证书的延期两种情况。证书的更换实际上是重新颁发证书,因此证书更换的过程和证书的申请流程基本一致。证书的延期只是将证书有效期延长,其签名和加密信息的公钥/私钥没有改变。

(3) 证书撤销。证书持有者可以向 CA 申请撤销证书。CA 通过认证核实可执行撤销证书职责,通知有关组织和个人,并写入 CRL。

CA 机构能够签发证书,同样也存在机制用于宣布以往签发的证书无效:证书使用者不合法,CA 需要废弃该证书;或者私钥丢失,使用者申请让证书无效。主要存在两类机制:CRL 与 OCSP。

- CRL(Certificate Revocation List,证书吊销列表)是一个单独的文件。该文件包含了 CA 已经吊销的证书序列号(唯一)与吊销日期,同时该文件包含生效日期并通知下次更新该文件的时间,当然该文件必然包含 CA 私钥的签名以验证文件的合法性。证书中一般会包含名为 CRL 分发点(CRL Distribution Point)的 URL 地址,通

知使用者去哪里下载对应的CRL以校验证书是否吊销。该吊销方式的优点是不需要频繁更新,但是不能及时吊销证书,因为CRL更新时间一般是几天,但这期间可能已经造成了极大损失。

- OCSP(Online Certificate Status Protocol,证书状态在线查询协议)是一种实时查询证书是否吊销的方式。请求者发送证书的信息并请求查询,服务器返回正常、吊销或未知中的任何一个状态。证书中一般也会包含一个OCSP的URL地址,要求查询服务器具有良好的性能。部分CA或大部分的自签CA(根证书)都是未提供CRL或OCSP地址的,对于吊销证书会是一件非常麻烦的事情。

(4) 证书和CRL的公布。CA通过LDAP服务器维护用户证书和CRL。它向用户提供目录浏览服务,负责将新签发的证书或废止的证书加入到LDAP服务器上,这样用户通过访问LDAP服务器就能够得到其他人的数字证书或能访问CRL。

(5) 证书状态的在线查询。通常CRL的发布为一日一次,CRL的状态同当前状态有一定的滞后。证书状态的在线查询通过向OCSP服务器发送OCSP查询包实现,包中含有待验证证书的序列号、验证时间戳,OCSP服务器返回证书的当前状态并对返回结果加以签名。在线证书状态查询比CRL更具有时效性。

(6) 证书认证。CA对证书进行有效性和真实性的认证,在实际操作中,如果一个CA管理得用户太多,则很难得到所有用户的依赖并接受它所发行的所有用户的公钥证书,而且一个CA也很难对大量的用户有足够、全面的了解,为此需要采用一种多CA分层结构的系统。在多个CA的系统中,由特定CA发放证书的所有用户组成一个域,同一个域中的用户可以直接进行证书交换和认证,而不同域中用户的公钥安全认证和递送需要通过建立一个可依赖的证书链或证书通路实现。跨域证书的认证也可通过交叉认证来实现,这会大大缩短信任关系的路径,提高效率。

(7) 制定政策。普通用户信任一个CA除了它的技术因素之外,另一个极重要的因素就是CA的政策。CA的政策是指CA必须对信任它的各方负责,它的责任大部分体现在政策的制定和实施上。CA的政策越公开越好,信息发布越及时越好。

CA的政策包含:CA私钥的保护;证书申请时密钥对的产生方式;对用户私钥的保护;CRL的更新频率;通知服务;保护CA服务器;审计与日志检查。

2.8 网络银行支付安全

传统支付方式为柜台支付或银行卡支付。银行卡是一种内置集成电路的芯片,芯片中存有与用户身份相关的数据。银行卡由专门的设备生产,是不可复制的硬件。银行卡由合法用户随身携带,登录时必须将银行卡插入专用的读卡器读取其中的信息,以验证用户的身份。银行卡传统的安全保障措施就是用户名和密码,这是很不安全的——容易被留驻内存的木马或网络监听等黑客技术窃取。

近年来支付更加方便和安全的网络支付日益普及,但现在的网络支付中存在着很多缺陷:

(1) 网络数据流分析。在用户和银行交易的过程中,第三方可通过各种方法截获数据流,分析数据中的信息从而得到用户的信息。

(2) 木马窃听。用户计算机中了病毒或者木马之后,计算机被监听,用户和银行交易的信息被木马记录,用户的信息被盗取。

(3) 穷举攻击。攻击者使用有意义的数字作为密码来不断尝试持卡人的密码。如果持卡人的密码是未经过改动的初始密码或一个特殊、容易被分析的数字,则密码很容易被攻击者穷举出来。

简单密码包括:身份证号码中连续的 6 位数字、出生年月日 3 组数字的任意组合、连续 5 位以上(含)数字相同的、6 位连续数字递增或递减的排列组合。银行统一集中开卡的密码多为 6 位连续数字,如 111111 等。

因为 ATM 只有数字键盘,所以很多银行卡的密码都是 6 位数字密码,密码空间为 10^6 (100 万),理论上讲是可以暴力破解的。当然实际上银行的密码不仅仅是通过算法的安全性来保密的,而是通过保密制度来保密的。也就是说,银行把密码在银行的安全性建立在密文不被窃取的前提下,这和通常 IT 网站所认为的不一样。

银行要求密码必须以密文形式存放。密码根据不同的密钥组合生成,相同密码不同的客户、同一客户不同账号生成的密文密码都是不同的,例如可能客户编号和账号就是组合成密钥的一部分,并且这两者可以组成唯一索引,保证不重复;所有的存入数据库的密码密文的加密算法至少能保证在当前是不可逆的;在不知道算法的情况下,根据密文想穷举计算不可行;当前账户依存的媒体一般是银行卡等,在银行卡里不只包含账号,还有一些校验位,只有账号密码,银行卡还是没法单独使用。所以保管好银行卡十分重要。

银行保密制度规定银行中只有极少数人能直接接触储户密码(密文),而这些人的身份和操作全部会被记录在案,就算离职,这些信息也不会被清除;涉及存储及使用这些信息的电脑网络物理隔离,操作终端有摄像头全程监控,操作者别想着抄下来;攻击银行或者金融机构是属于犯罪行为。

但攻击者仍然可以通过撞库等方式获取密码。撞库指黑客通过收集互联网已泄露的用户和密码信息,生成对应的字典表,尝试批量登录其他网站后,得到一系列可以登录的用户信息。

很多用户在不同网站使用的是相同的账号密码,因此黑客可以通过获取用户在 A 网站的账户从而尝试登录 B 网址,这就是撞库攻击。2014 年 12 月 25 日,12306 网站用户信息在互联网上疯传。对此 12306 官方网站称,网上泄露的用户信息系经其他网站或渠道流出。据悉此次泄露的用户数据不少于 131 653 条。该批数据基本确认为黑客通过撞库攻击所获得。

(4) 网络钓鱼:第三方利用银行的身份给用户发信息,要求用户提供账号和密码,如果用户提供了的话就会泄露自己的信息。或者是第三方假冒银行或者交易的网站,在没有认真辨别的情况下,用户很容易上当从而泄露自己的信息。

为了提高网络支付安全性,硬件认证、指纹认证、虹膜认证、人脸认证等多种认证方式应运而生。

2.8.1　U 盾定义

硬件认证的代表是 U 盾。U 盾即移动数字证书 USB key,它存放着个人的数字证书,并不可读取。是目前网上银行客户端安全级别最高的一种安全工具。同样,银行也记录着

你的数字证书。

U盾(USB Key)外形酷似U盘,像一面盾牌,时刻保护着网上银行的资金安全。U盾采用了目前国际领先的信息安全技术,核心硬件模块采用智能卡CPU芯片,内部结构由CPU及加密逻辑、RAM、ROM、EEPROM和I/O五部分组成,是一个具有安全体系的小型计算机。除了硬件,安全实现完全取决于技术含量极高的智能卡芯片操作系统(COS),该操作系统就像DOS、Windows等操作系统一样,管理着与信息安全密切相关的各种数据、密钥和文件,并控制各种安全服务。

从技术角度看,U盾是用于网上银行电子签名和数字认证的工具,它采用1024位非对称密钥算法对网上数据进行加密、解密和数字签名,确保网上交易的保密性、真实性、完整性、不可否认性。

U盾中装有数字证书,数字证书是由可信任的第三方认证机CA颁发的一组包含用户身份信息(密钥)的数据结构,PKI体系通过采用加密算法构建了一套完善的流程,保证数字证书持有人的身份安全。唯一不同的是,你不会把U盾24小时放在USB接口上,要用的时候才会插上U盾,用完马上把U盾拔出。而这个过程时间是非常短的。就算你被最出色的黑客瞄上了或中了最新款的病毒木马等,也没可能在这么短的时间内对你的U盾证书进行解密并复制出来。1024位加密的证书除非知道密码否则暴力破解简直是天方夜谭。没有密码,对方就算远程登录到你的机器也不可能把你的证书复制到他的机器上。

使用U盾可以保障数字证书无法被复制,所有密钥运算在U盾中实现,用户密钥不在计算机内存出现也不在网络中传播,只有U盾的持有人才能够对数字证书进行操作,因此安全性有了保障。

使用U盾前,一般都要安装驱动,使U盾能正常工作。当用户需要交易向银行提交订单要求支付时,这时就需要验证用户的身份,系统提示用户插入U盾,并输入U盾的密码,系统会在后台验证(用户看不到该过程过程),一经验证通过,用户就可以使用继续输入网上支付密码和验证码,验证都正确后交易就完成。

U盾保证交易安全的措施有:

(1) 硬件PIN码保护。U盾使用以物理媒体为基础的个人客户证书,建立基于公钥PKI技术的个人证书认证体系。用户需要同时取得用户的U盾硬件以及用户的PIN码(U盾密码),才可以登录系统。即使用户的PIN码泄露,只要U盾没有丢失,合法用户的身份就不会被仿冒;如果用户U盾丢失,其他人不知道用户的PIN码,也是无法假冒合法用户的身份。因此只要登录卡号、登录密码、U盾和U盾密码不同时泄露给同一个人,就可以放心安全地使用网上银行。

(2) 安全的密钥存放。U盾具有硬件随机数发生器,对称密钥完全在硬件内生成并存储于内部的智能芯片中,用户无法从外部直接读取。硬件提供的加解密算法完全在加密硬件内运行,对密钥文件的读写和修改都必须由U盾内部的CPU调用相应的程序文件执行,能够保证密钥不出硬件,从而在U盾接口的外面,没有任何一条指令能对密钥区的内容进行读取、修改、更新和删除,这保证了黑客无法利用非法程序修改密钥。

(3) 双密钥密码体制。为了提高交易的安全,U盾采用了双钥密码体制保证安全性,在U盾初始化的时候,先将密码算法程序烧制在ROM中,然后通过产生公私密钥对的程序生成一对公私密钥。公私密钥产生后,密钥可以导出到U盾外,而私钥则存储于密钥区,不允

许外部访问。因此你的数字证书有一对密钥：一个是在 U 盾里的私钥，一个是在银行的公钥。进行数字签名时以及非对称解密运算时，凡是有私钥参与的密码运算只在芯片内部完成，全程私钥可以不出 U 盾媒体，从而保证以 U 盾为存储媒体的数字证书认证在安全上无懈可击。

(4) 硬件实现加密算法。U 盾内置 CPU 或智能卡芯片，可以实现数据摘要、数据加解密和签名的各种算法。加解密运算在 U 盾内进行，保证用户密钥不会出现在计算机内存中。

2.8.2　U 盾工作原理

U 盾使用基于 PKI 的数字证书认证模式进行银行和客户身份的双向认证来保证交易安全。

1. 基于冲击-响应的认证模式

U 盾内置 RSA 算法，初始化时，预先在 U 盾和服务器中存储一个证明用户身份的密钥对。

当 Alice 尝试进行网上交易时，需要在网络上验证用户 Alice 身份：先由客户端 Alice 向服务器 Bank 发出一个验证请求。Bank 接到此请求后向你发送由时间字符串、地址字符串、交易信息字符串、防重放攻击字符串组合在一起进行加密后得到的随机字符串 strA，并回传给 Alice 的 U 盾，此为冲击。

U 盾使用 strA 与存储在 U 盾中的密钥进行 RSA 运算得到一个运算结果 strB，作为认证证据传送给服务器，此为响应。

与此同时，服务器使用随机数 strA 与存储在服务器数据库中的该客户密钥进行 RSA 运算，如果服务器的运算结果与客户端传回的响应结果相同，则认为客户端是一个合法用户，完成对客户身份的认证，交易便可以完成；如果不一致便认为你不合法，交易便会失败。

理论上不同的字符串 strA 不会得出相同的字符串 strB，即一个字符串 strA 对应一个唯一的字符串 strB；字符串 strB 和字符串 strA 无法得出你的数字证书，而且 U 盾具有不可读取性，所以任何人都无法获得你的数字证书；并且银行每次都会发不同的防重放字符串(随机字符串)和时间字符串，所以当一次交易完成后，刚发出的 strA 字符串便不再有效。综上所述，理论上 U 盾是绝对安全的。理论上发生伪造概率大约为 $1/2^{80}$。

该模式类似于双向认证的 TLS(SSL)或者其他用到 RSA 的双向证书验证。具体步骤为：

(1) Alice 在客户端插入 U 盾，输入 PIN 码。

(2) Bank(服务器端)检查 PIN 码是否和 U 盾对应，成功则执行下一步；否则拒绝服务。

(3) Bank 收到 PIN 码后查找对应的公钥，给 U 盾(客户端)一个冲击，它包含一个随机数，以及该随机数的 Hash 值，它们都由 U 盾公钥加密，这样就可以保证只有 U 盾能解密这个冲击。这是一种典型的非对称加密用法。

(4) U 盾计算该随机数的 Hash 值，并和用私钥解出的 Hash 值比较，两者相同后，便可确认银行的身份正确和数据完整性(数据在传输过程中没有被篡改)，实现 Bank 身份认证。

(5) U 盾以一个只有 U 盾和 Bank 知道的算法(U 盾初始化时选定)，利用这个随机数和一些其他信息，生成响应和相应的 Hash 值，再用私钥加密后发回 Bank。这是典型的非

对称加密的签名用法,用于保证 Alice 不可抵赖。

(6) Bank 和(4)同步以相同的算法计算该响应。

(7) Bank 用 U 盾公钥解密(4),比较和(5)的响应是否相同,相同的话 Alice 的身份也就确定了,完成 Alice 身份认证,银行执行具体银行业务;否则拒绝服务。

该过程进行了 Alice 和 Bank 双方身份认证,实现了保密传输。至于私钥的保密性则由 U 盾来实现。U 盾的控制芯片被设计为只能写入证书,不能读取证书,并且所有利用证书进行的运算都在 U 盾中进行。所以,只能从 U 盾读出运算结果。

2. 基于 PKI 的数字证书认证模式

冲击-响应模式可以保证用户身份不被仿冒,但无法保证认证时数据在网络传输过程中的安全。通过在 U 盾中内置数字证书,采用上节介绍的基于 PKI 的数字证书认证方式可以有效保证用户的身份安全和数据传输安全。

由于 U 盾具有安全可靠,便于携带、使用方便、成本低廉的优点,加上 PKI 体系完善的数据保护机制,使用 U 盾存储数字证书的认证方式已经成为目前主要的认证模式。

2.9 国产密码算法

密码算法是保障网络信息安全的核心技术,在网络信息安全中发挥着至关重要的作用,主要用于网络身份认证以及数据存储、传输的保密,是金融服务等关键领域实现用户身份认证、信息传输加密、保障交易及用户数据安全的关键技术保障措施。

目前我国网银、支付等系统的密码应用中,存在着两方面的问题:一是我国银行业核心领域长期以来都是沿用 3DES、SHA-1、RSA 等国际通用的商用密码算法体系及相关标准(美国国家安全局发布),国际权威的密码机构已确认 RSA 算法不再安全,存在安全漏洞,可以被破解;二是密码应用体系存在安全隐患,当前网银等系统中采用的安全协议和加密算法均为国外制定,密码应用的关键环节存在着不可控因素,一旦被利用攻击,将对我国的金融安全造成重大冲击。

我国政府高度重视密码算法国产化工作。为了从根本上摆脱对国外密码技术和产品的过度依赖,国家商用密码管理局组织制定了一系列我国自主研发的密码算法,包括 SM2、SM3、SM4、SM9 算法等,并于 2012 年陆续公布了相关算法标准。

国产密码算法(简称国密算法)是指国家密码局认定的国产商用密码算法,主要有 SM1、SM2、SM3、SM4,相关文档和 C 代码可从网站下载。

2.9.1 SM4 分组密码算法

分组密码就是将明文数据按固定长度进行分组,然后在同一密钥控制下逐组进行加密,从而将各个明文分组变换成一个等长的密文分组的密码。其中二进制明文分组的长度称为该分组密码的分组规模。

分组密码的实现原则如下:

(1) 必须实现比较简单,知道密钥时加密和解密都十分容易,适合以硬件和(或)软件实现。

(2) 加解密速度和所消耗的资源和成本较低,能满足具体应用范围的需要。

分组密码的设计基本遵循混淆原则和扩散原则:

混淆原则就是将密文、明文、密钥三者之间的统计关系和代数关系变得尽可能复杂,使得敌手即使获得了密文和明文,也无法求出密钥的任何信息;即使获得了密文和明文的统计规律,也无法求出明文的任何信息。

扩散原则就是将明文的统计规律和结构规律散射到相当长的一段统计中去。也就是说,让明文中的每一位影响密文中尽可能多的位,或者说让密文中的每一位都受到明文中尽可能多位的影响。

SM1、SM4 均采用对称加密算法,加解密的分组大小为 128 位,故对数据进行加解密时,若数据长度过长,需要进行分组;若数据长度不足,则要进行填充。

要保证一个对称密码算法安全性的基本条件是其具备足够的密钥长度,SM1、SM4 算法与 AES 算法具有相同的密钥长度分组长度——128 位,因此在安全性上高于 112 位的 3DES 算法。

SM1 主要用于有线网络,其加密强度与 AES 相当。该算法不公开,仅以 IP 核的形式存在于芯片中。调用该算法时,需要通过加密芯片的接口进行调用。当使用特定的芯片进行 SM1 或其他国密算法加密时,若用多个线程调用加密卡的 API 时,要考虑芯片对于多线程的支持情况。

采用 SM1 算法已经研制了系列芯片、智能 IC 卡、智能密码钥匙、加密卡、加密机等安全产品,广泛应用于电子政务、电子商务及国民经济的各个应用领域(包括国家政务通、警务通等重要领域)。

国际的 DES 算法和国产的 SM4 算法的目的都是为了加密保护静态储存和传输信道中的数据,主要特性对比如表 2-18。

表 2-18　DES 算法与 SM4 算法比较

比 较 项 目	DES　算　法	SM4　算　法
算法结构	使用标准的算术和逻辑运算,先替代后置换,不含非线性变换	基本轮函数加迭代,含非线性变换
加解密算法是否相同	是	是
计算轮数	16 轮(3DES 为 16 轮×3)	32 轮
分组长度	64 位	128 位
密钥长度	64 位(3DES 为 128 位)	128 位
有效密钥长度	56 位(3DES 为 112 位)	128 位
实现难度	易于实现	易于实现
实现性能	软件实现慢、硬件实现快	软件和硬件实现都快
安全性	较低(3DES 较高)	算法较新,还未经过现实检验

SM4 是无线局域网标准的分组数据算法,是我国自主设计的分组对称密码算法,用于实现数据的加密/解密运算,以保证数据和信息的机密性。

2006 年我国公布了无线局域网产品使用的 SM4 密码算法,这是我国第一次公布自己的商用密码算法。

SM4 加密算法与密钥扩展算法都采用 32 轮非线性迭代结构。每次迭代由一个轮函数

给出,其中轮函数由一个非线性变换和线性变换复合而成,非线性变换由 S 盒所给出。

解密算法与加密算法的结构相同,只是轮密钥的使用顺序相反,解密轮密钥是加密轮密钥的逆序。

从算法上看,国产 SM4 算法在计算过程中增加非线性变换,理论上能大大提高其算法的安全性,并且由专业机构进行了密码分析,民间也对 21 轮 SM4 进行了差分密码分析,结论均为安全性较高。

2.9.2 SM2 公钥密码算法

SM2 椭圆曲线公钥密码算法是我国自主设计的公钥密码算法,包括 SM2-1 椭圆曲线数字签名算法、SM2-2 椭圆曲线密钥交换协议、SM2-3 椭圆曲线公钥加密算法,分别用于实现数字签名、密钥协商、数据加密等功能。

SM2 算法与 RSA 算法不同的是,SM2 算法是基于椭圆曲线上点群离散对数难题,相对于 RSA 算法,ECC 256 位(SM2 采用的就是 ECC 256 位的一种)安全强度比 RSA 2048 位高,但运算速度快于 RSA。

SM2 为非对称加密,基于 ECC。该算法已公开。由于该算法基于 ECC,故其签名速度与密钥生成速度都快于 RSA。

SM2 算法由国家密码管理局于 2010 年 12 月 17 日发布,是一种椭圆曲线算法,使用的椭圆曲线方程为:$y^2=x^3+ax+b$

SM2 算法实现如下:

(1) 选择 $E_p(a,b)$的元素 G,使得 G 的阶 n 是一个大素数;

(2) G 的阶是指满足 $nG=O$ 的最小 n 值;

(3) 秘密选择整数 k,计算 $B=kG$,然后公开(p,a,b,G,B),B 为公钥,保密 k,k 为私钥;

加密 M:先把数据 m 变换成为 $E_p(a,b)$中一个点 P_m,然后,选择随机数 r,计算密文 $C_m=\{rG,P_m+rP)$,如果 r 使得 rG 或者 rP 为 O,则要重新选择 r。

解密 Cm:$(P_m+rP)-k(rG)=P_m+rkG-krG=P$。

SM2 算法的安全性基于一个数学难题“离散对数问题 ECDLP”实现,即考虑等式 $Q=kP$,其中 Q、P 属于 $E_p(a,b)$,$k<p$,则可证明由 k 和 P 计算 Q 比较容易,而由 Q 和 P 计算 k 则比较困难。由于目前所知求解 ECDLP 的最好方法是指数级的,这使得我们选用 SM2 算法作加解密及数字签名时,所要求的密钥长度比 RSA 要短得多。

国际的 RSA 算法和国产的 SM2 算法的主要特性对比如表 2-19。

表 2-19 RSA 算法与 SM2 算法比较

比 较 项 目	RSA 算法	SM2 算法
计算结构	基于特殊的可逆模幂运算	基于椭圆曲线
计算复杂度	亚指数级	完全指数级
相同的安全性能下所需公钥位数	较多	较少(160 位的 SM2 与 1024 位的 RSA 具有相同的安全等级)
密钥生成速度	慢	较 RSA 算法快百倍以上
解密加密速度	一般	较快
安全性难度	基于分解大整数的难度	基于离散对数问题、ECDLP 数学难题

2.9.3　SM3 摘要算法

摘要函数在密码学中具有重要的地位，被广泛应用在数字签名、数据认证、数据完整性检测等领域。2005 年，王小云等人给出了 MD5 算法和 SHA-1 算法的碰撞攻击方法，证明现今被广泛应用的 MD5 算法和 SHA-1 算法不再是安全的算法。

摘要函数通常被认为需要满足 3 个基本特性：碰撞稳固性、原根稳固性、第二原根稳固性。

SM3 摘要算法是中国国家密码管理局 2010 年公布的中国商用密码摘要算法标准，适用于商用密码应用中的数字签名、验证数据认证码的生成与验证、随机数的生成，可满足多种密码应用的安全需求。

SM3 摘要算法采用 Merkle-Damgard 结构，数据分组长度为 512 位，摘要值长度为 256 位。

SM3 摘要算法是我国自主设计的密码摘要算法，为了保证摘要算法的安全性，其产生的摘要值的长度不应太短，例如，MD5 输出 128 位摘要值，输出长度太短，影响其安全性；SHA-1 摘要算法的输出为 160 位摘要值；SM3 摘要算法的输出为 256 位摘要值，因此 SM3 摘要算法的安全性要高于 MD5 摘要算法和 SHA-1 摘要算法。

SM3 摘要算法可以与 MD5 摘要算法对比理解。SM3 摘要算法是在 SHA-256 摘要算法基础上改进实现的一种算法，其压缩函数与 SHA-256 摘要算法的压缩函数具有相似的结构，但是 SM3 摘要算法的设计更加复杂，例如压缩函数的每一轮都使用 2 个数据字。按目前情况来讲，SM3 摘要算法的安全性相对较高。

SM3 摘要算法大体上与 SHA-256 摘要算法相同，其算法过程如下：

(1) 填充。SM3 摘要算法对数据长度小于为 2^{64} 位进行运算，其填充方法与 SHA-256 摘要算法相同，假设数据 m 的长度为 l 比特。首先将比特 1 添加到数据的末尾，再添加 k 个 0，k 是满足 $l+1+k=448 \bmod 512$ 的最小的非负整数，然后再添加一个 64 位比特串。填充后的数据 m' 的比特长度为 512 的倍数。

(2) 迭代压缩。这个过程与其他 Hash 摘要算法类似，先进行数据扩展，之后迭代与压缩，其详细过程可参考标准文档。其扩展与压缩计算以循环移位为主，并有异或计算。

近年来国家有关机关和监管机构站在国家安全和长远战略的高度提出了推动国密算法应用实施、加强行业安全可控的要求。2010 年底，国家密码管理局公布了我国自主研制的“椭圆曲线公钥密码算法”(SM2 算法)。为保障重要经济系统密码应用安全，国家密码管理局于 2011 年发布了《关于做好公钥密码算法升级工作的通知》，要求自 2011 年 3 月 1 日期，在建和拟建公钥密码基础设施电子认证系统和密钥管理系统应使用 SM2 算法。自 2011 年 7 月 1 日起，投入运行并使用公钥密码的信息系统，应使用 SM2 算法。

2015 年 2 月国家商业密码管理办公室发布公告称：根据要求全国第三方电子认证服务机构(CA)针对电子认证服务系统和密钥管理系统公钥算法进行了升级改造完毕，已经全面支持国产算法，同时各认证服务机构正在积极推动国产算法的应用服务改造，淘汰有安全风险以及低强度的密码算法和产品。

目前，全国 30 多家 CA 认证机构已经具备发放基于自主非对称 SM2 算法的数字证书能力。在产品方面，经过上百家密码企业的努力，已经有近 400 个支持 SM2/SM3/SM4 算

法的产品通过了国家密码管理局的检测,产品涵盖密码芯片、加密卡、加密机、智能密码钥匙、ATM系统、安全网关以及各种专用安全终端等。支持国密算法的软硬件密码产品共699项,包括SSL网关、数字证书认证系统、密钥管理系统、金融数据加密机、签名验签服务器、智能密码钥匙、智能IC卡、PCI密码卡等多种类型,目前已初步形成形式多样、功能互补的产品链,并保持着持续增长的势头。

虽然在SSL VPN、数字证书认证系统、密钥管理系统、金融数据加密机、签名验签服务器、智能密码钥匙、智能IC卡、PCI密码卡等产品上改造完毕,但是目前的信息系统整体架构中还使用操作系统、数据库、中间件、浏览器、网络设备、负载均衡设备、芯片等软硬件,由于复杂的原因无法完全把密码模块升级为国产密码模块,导致整个信息系统还存在安全薄弱环节。

密码服务是信息化安全建设的基础服务,密码的国产化改造和推广就成为我们重要的历史使命。为了普及和推广国产密码,我们可以:一方面是产品升级改造,对于国外的产品,通过国产密码算法的标准出海战略,让国产密码算法成为国际标准,从而得到国外产品的支持;对于国产的产品,加快国产密码算法模块的改造和应用,真正让国产密码算法为信息系统的安全自主可控保驾护航;另一方面是应用的宣传和推广,国产密码算法虽然在安全圈里面是众所周知的事情,但是在其他领域根本就没有听说。所以对于从业者来说,就要不断对用户灌输使用国产密码算法以及尽快升级到国产密码算法的思想。只有从以上这两个方面入手并且持之以恒,才能使国家提出的信息安全领域的自主可控战略最终实现。

习 题 2

2.1 密码学定义

1. 已知古典凯撒密码加密密文 c_1=wr eh ru qrw wr eh,wkdw lv wkh txhvwlrq,求明文 p_1。

2. 已知经凯撒密码加密密文 c_2=opx zpv bsf ibwjoh b uftu,使用暴力破解法求 p_2,要求列出所有25种可能解密结果。

3. 已知明文 p_1=you are a student,使用2栏栅栏密码加密,求密文 c_1。

4. 用置换矩阵 $\boldsymbol{E}_k=\begin{bmatrix}0&1&2&3&4\\1&4&3&2&0\end{bmatrix}$ 对明文 Now we are having a test 加密,并给出其解密矩阵及求出可能的解密矩阵总数。

5. 置换密码的特点是(　　)变,(　　)不变;替代密码的特点(　　)变,(　　)不变。

6. 密码系统的5个要素是(　　)、(　　)、(　　)、(　　)、(　　)。

7. 密码学的目的是(　　)。

A. 研究数据加密　　B. 研究数据解密

C. 研究数据保密　　D. 研究信息安全

8. 假设使用一种加密算法，它的加密方法很简单：将每一个字母加 5，即 a 加密成 f。这种算法的密钥就是 5，那么它属于(　　)。

A. 对称加密技术　　B. 分组密码技术

C. 公钥加密技术　　D. 单向函数密码技术

9. 古典密码包括(　　)和(　　)两种。对称密码体制和非对称密码体制都属于现代密码体制，对称密码体制主要存在两个缺点：一是(　　)；二是(　　)。在实际应用中，对称密码算法与非对称密码算法总是结合起来的，对称密码算法用于加密，而非对称算法用于保护对称算法的密钥。

10. 密码学的发展经历了(　　)、(　　)、(　　)3 个发展阶段。密码学的两个分支是(　　)和(　　)。

11. 密码分析的 4 种方法是(　　)、(　　)、(　　)、(　　)。

2.2　DES 对称密码

1. 对称密钥算法的特点是(　　)。

2. 对称密钥算法的安全性依赖于(　　)。

3. DES 算法的入口参数有 3 个：(　　)、(　　)、(　　)。DES 算法实现加密需要 3 个步骤：(　　)、(　　)、(　　)。

4. DES 算法密钥是 64 位，其中密钥有效位是(　　)位。

5. 三重 DES 需要执行 3 次常规的 DES 加密步骤，但最常用的三重 DES 算法中仅仅用两个(　　)DES 密钥，因此密钥长度为(　　)位。

6. DES 的密码组件之一是 S 盒。根据 S 盒表计算 $S_3=110011$ 时的值，并说明 S 函数在 DES 算法中的作用。假定 S_3 盒对应行列交叉处的值为 8。

2.3　RSA 公钥密码

1. RSA 公钥密码系统基于(　　)这一著名数论难题：(　　)。

2. 计算：

(1) 400　mod 23；　(2) 5843 mod 23；　(3) -9 mod 4；　(4)5^{10} mod 7；

(5) 13/14 mod 11；　(6) $-1/2$ mod 11；　(7)301/4 mod 11　(8)$-7/3$ mod 11。

3. 写出 $a=10$，$m=5$ 的所有等价类，据此说明等价类的作用是(　　)。

4. 密码学中大量使用模运算的原因是什么？

5. 列举 RSA 的两种用法，并分别解释其定义。

6. (　　)是笔迹签名的模拟，是一种防止源点或终点否认的认证技术。

7. RSA 为什么能实现数字签名？

8. A 方有一对密钥($K_{A公开}$，$K_{A秘密}$)，B 方有一对密钥($K_{B公开}$，$K_{B秘密}$)，A 方向 B 方发送数字签名 M，对信息 M 加密为：$M'=K_{B公开}(K_{A秘密}(M))$。B 方收到密文的解密方案是_____。

A. $K_{B公开}(K_{A秘密}(M'))$　　B. $K_{A公开}(K_{A公开}(M'))$

C. $K_{A公开}(K_{B秘密}(M'))$　　D. $K_{B秘密}(K_{A秘密}(M'))$

9. 什么是计算安全性？什么是理论安全性？

10. DES 理论上(　　)，或者说具有(　　)安全性；RSA 理论上(　　)，但计算(　　)，或者说不具有(　　)，具有(　　)。

2.4 椭圆曲线密码

1. 椭圆曲线的三次方程形为(　　)。

2. 设 R 的坐标为(x_3,y_3),利用 P、Q 点的坐标(x_1,y_1)、(x_2,y_2)求 $R=P+Q$ 的计算公式是(　　)。

3. 已知 $y^2\equiv x^3-2x-3$ 是系数在 GF(7)上的椭圆曲线,$P=(3,2)$是其上一点,求 $10P$。

4. 椭圆曲线密码体制的安全性建立在椭圆曲线离散对数问题之上。椭圆曲线上的离散对数问题表示为(　　)。

5. 椭圆曲线密码体制的优点是(　　)、(　　)、(　　)。

6. 如何使用公钥加密技术解决对称密钥分发问题?

2.5 数据完整性认证

1. MD5 的特征是(　　)、(　　)、(　　)。

2. 杂凑碰撞是指(　　)。王小云院士的研究成果表明(　　)。

3. MD5 的应用包括(　　)、(　　)。

4. 数据认证中常见的攻击有(　　)、(　　)、(　　)、(　　)。

2.6 数字签名

1. 数字签名要预先使用单向 Hash 函数进行处理的原因是(　　)。

A. 多一道加密工序使密文更难破译

B. 提高密文的计算速度

C. 缩短签名密文的长度,加快数字签名和验证签名的运算速度

D. 保证密文能正确还原成明文

2. 身份鉴别是安全服务中的重要一环,以下关于身份鉴别叙述不正确的是(　　)。

A. 身份鉴别是授权控制的基础

B. 身份鉴别一般不用提供双向的认证

C. 目前一般采用基于对称密钥加密或公开密钥加密的方法

D. 数字签名机制是实现身份鉴别的重要机制

3. Alice 想将一份机密合同通过 Internet 发给 Bob,如何才能实现这个合同的完整、安全发送?

4. 简述采用公钥密码体制和单向 Hash 函数进行的数字签名过程。在这个过程中共使用了几对密钥?它们各自的作用是什么?

5. 使用(　　)解决身份认证问题和数据传输完整性问题,实现发送方身份不可抵赖性;使用(　　)绑定公钥和公钥所有人,保证公钥可信,实现发送方身份不可抵赖性;使用(　　)强化身份认证。

6. 使用(　　)保证时间不可抵赖性,防止重放攻击。

2.7 公钥基础设施

1. PKI 支持的服务不包括(　　)。

A. 非对称密钥技术及证书管理　　B. 目录服务

C. 对称密钥的产生和分发　　D. 访问控制服务

2. 简述 CA 使用流程。

3. http 协议采用(　　)传输信息,存在(　　)、(　　)和(　　)的风险,而 https

协议具有(　　)、(　　)、(　　)的功能,采用(　　)传输信息,可以避免此类问题发生。

4. https=http+TLS/SSL,TLS/SSL 的功能实现主要依赖于 3 类基本算法:(　　)、(　　)和(　　),其利用非对称加密算法实现(　　),对称加密算法(　　),(　　)验证信息的完整性。

5. 二级证书结构存在的优势是什么?

2.8　网络银行支付安全

1. U 盾即移动数字证书 USB key,它存放着用户个人的(　　),并不可读取。

2. USB key 采用了目前国际领先的信息安全技术,核心硬件模块采用智能卡 CPU 芯片,内部结构由 CPU 及加密逻辑、RAM、ROM、EEPROM 和 I/O 五部分组成,是一个具有安全体系的(　　)。

3. U 盾是用于网上银行电子签名和数字认证的工具,它采用(　　)位非对称密钥算法对网上数据进行加密、解密和数字签名,确保网上交易的保密性、真实性、完整性和不可否认性。

4. U 盾是如何保证网上银行支付安全的?

5. 简述 U 盾的工作原理。

2.9　国产密码算法

1. 分组密码设计需要遵循的混淆原则和扩散原则的具体含义是什么?

2. 国际的 DES 算法和国产的 SM4 算法的目的都是为了(　　)。

3. SM2 算法使用的椭圆曲线方程为(　　)。

4. 简述 SM2 算法。

5. SM3 算法采用 Merkle-Damgard 结构,数据分组长度为(　　)位,摘要值长度为(　　)位。MD5 算法的摘要值长度为(　　)位,SHA-1 算法的摘要值长度为(　　)位,SM3 算法的摘要值长度为(　　)位,因此 SM3 算法的安全性要高于 MD5 算法和 SHA-1 算法。

6. 国产密码和国际密码的对应关系是:DES 对应(　　),RSA 密码对应(　　),MD5 对应(　　)。

实验 1　凯撒密码解密进阶

【实验目的】

(1) 编程实现凯撒加密、解密算法,理解密码学基础知识,初步建立密码学思维方式。

(2) 通过不断增加凯撒解密难度,理解唯密文解密,提高解密智能化程度。

【实验内容和要求】

(1) 在允许输入密码的条件下,编程实现凯撒密码加解密。要求:

① 从一文本文件读入英文文章(明文或密文)。

② 对读入内容加密或解密后写入另一文本文件。

(2) 在不允许输入密码的条件下,编程实现解密凯撒密码加密密文。要求绘制3种情况下的解密程序流程图,说明不同解密程序存在的不足。程序需要计算、显示解密使用时间(单位:ms)。

① 已知 c_1=wklv lv d errn,求 p_1。(初级解密)

问:两次使用凯撒密码算法,能否正确解密?(字符串用凯撒加密后的结果再用凯撒加密一次。)

② 已知 c_1=go kbo cdenoxdc,或 c_1=zh duh vwxghqwv,求 p_1。(中级解密)

③ 已知 c_1=rxwvlgh wkh eleoh, wkhvh vla zrugv duh wkh prvw idprxv lq doo wkh olwhudwxuh ri wkh zruog. wkhb zhuh vsrnhq eb kdpohw zkhq kh zdv wklqnlqj dorxg, dqg wkhb duh wkh prvw idprxv zrugv lq vkdnhvshduh ehfdxvh kdpohw zdv vshdnlqj qrw rqob iru klpvhoi exw dovr iru hyhub wklqnlqj pdq dqg zrpdq. wr eh ru qrw wr eh, wr olyh ru qrw wr olyh, wr olyh ulfkob dqg dexqgdqwob dqg hdjhuob, ru wr olyh gxoob dqg phdqob dqg vfdufhob. d sklorvrskhu rqfh zdqwhg wr nqrz zkhwkhu kh zdv dolyh ru qrw, zklfk lv d jrrg txhvwlrq iru hyhubrqh wr sxw wr klpvhoi rffdvlrqdoob. kh dqvzhuhg lw eb vdblqj: "l wklqn, wkhuhiruh dp.",求 p_1。(高级解密)

(3) 对给定较长密文文件进行解密测试,将测试结果填入表2-20。

要求密文的内容不少于1000个英文单词,使用凯撒密码加密,加密密码保密。

正确率=正确单词数/单词总数,智能程度:优秀(解密结果正确与否不需要人工判断)、一般。

表2-20 凯撒密码解密结果

学 号	姓 名	时 间	正 确 率	智 能 程 度

(4) 选择测试结果前3名同学进行解密算法演讲和程序演示,学习交流不同解决算法。该工作可在下一次实验前进行。

实验2 DES算法和RSA算法性能测试

【实验目的】

(1) 通过实验让学生充分理解DES算法,掌握使用DES算法加密和解密数据。

(2) 通过实验让学生充分理解RSA算法,掌握使用RSA算法加密和解密数据。

(3) 通过对比(1)和(2)的实验结果,对比DES算法和RSA算法的性能,总结其各自使用的条件。

【实验内容和要求】

(1) 学习使用DES和RSA加密、解密函数使用方法。

使用DES和RSA加解密非常简单,常用语言都提供了相应函数(方法),只要正确填写

参数就能实现数据加解密。以 Java 为例，Java 提供了安全框架类和接口 java. security 以及软件包 javax. crypto，支持对称密码算法、不对称密码算法、块密码算法、流密码算法等多种加解密。

将明文 c 赋值给变量 msg，密钥赋值给变量 key，调用 encrypt(msg. getBytes()，key)方法就可实现 DES 加密，密文存入 enMsg；调用 decrypt(enMsg，key)方法就可实现 DES 解密。

RSA 加解密，首先需要产生密钥。选定密钥长度 keylength(256，512，1024 可选)，调用方法 generateKey(keylength)就可产生随机产生密钥对(RSAPublicKey，RSAPrivateKey)。使用(RSAPublicKey) keyPair. getPublic()可查看公钥，使用(RSAPrivateKey) keyPair. getPrivate()可查看私钥。

将明文 c 赋值给变量 msg，使用 encryptByPublicKey(msg，publicKey)公钥加密，使用 decryptByPrivateKey(plainText，privateKey)私钥解密，实现 RSA 加密用法；或使用 encryptByPrivateKey (msg，privateKey) 私钥加密，使用 decryptByPublicKey (privateplainText，publicKey)实现签名用法。

(2) 编程实现对字符串"hellodes"进行 DES 和 3DES 加解密，将对应结果填入表 2-21。

表 2-21　DES 加解密

算　法	明　文	密文(BASE64)	密文长度	十六进制表示	加密用时	解密用时
DES	hellodes					
3DES	hellodes					

明文：hellodes　　DES 密文长度：12

K_1：12345678(加密)　　加密结果：c_1＝?

K_2：12345678(解密)c_1　　解密结果：p_1＝?

K_3：87654321(加密)，c_1　　加密结果：p_2＝?

明文：hellodes　　三重 DES 密文长度：12

K_1：11111111 (加密)　　加密结果：

K_2：22222222(解密)　　解密结果：

K_3：11111111 (加密)　　加密结果：

明文：hellodes　　三重 DES 密文长度：12

K_1：11111111 (加密)　　加密结果：

K_2：11111111(解密)　　解密结果：

K_3：2222222(加密)　　加密结果：

【实验结论 1】

(1) DES 解密快，加密慢。

(2) 3DES 的速度与解密、解密次序密切相关。

(3) 编程实现对某个文本文件进行 DES 加密、解密，记录加密、解密时间和文件大小。要求加解密时间达到分钟级别。

(4) 验证 DES 加密会产生雪崩效应。

① 用同样密钥加密只差一比特的两个明文。

② 用只差一比特的两个密钥加密同样明文。

(5) 编程实现对字符串"hellorsa"进行 RSA 加解密,记录加密用法和签名用法加解密的结果。

待加密明文:hellorsa 密钥长度:256 密文长度:

公钥:

私钥:

==========RSA 加密用法==========

公钥加密后密文:

加密用时: 毫秒 解密用时: 毫秒

==========RSA 签名用法==========

私钥加密后密文:

加密用时: 毫秒 解密用时: 毫秒

【实验结论 2】

对于相同明文,使用相同密钥对,使用公钥加密的结果和私钥加密的结果不同。

(6) 编程实现对字符串"hellorsa"使用不同长度 RSA 密钥加解密,将加解密的结果填入表 2-22。

表 2-22 RSA 不同密钥长度加解密

算 法	明 文	密钥长度	密文长度	加密用时	解密用时
RSA 公钥加密	hellorsa	512			
RSA 公钥加密	hellorsa	1024			
RSA 私钥加密	hellorsa	512			
RSA 私钥加密	hellorsa	1024			

【实验结论 3】

随着密钥长度增加,加解密时间增加,密文长度增加。

(7) 编程实现对字符串"hellotest"分别进行 DES 和 RSA 加解密,将加解密的结果填入表 2-23。

表 2-23 DES 和 RSA 加解密对比

加密字符	密 钥	加密用时	解密用时	密 文	密文长度
RSA	hellotest				
DES	hellotest				

DES 密文为十六进制形式,RSA 为 BASE64 形式。

【实验结论 4】

对称密钥算法的加解密时间远远小于公钥算法的加解密时间。

【思考题】

1. 3DES 就执行效果而言执行了 3 次 DES，但形式上执行的是两次加密和一次解密，必须遵守 EDE 规则；3DES 使用了 2 个不同密钥，而不是 3 个不同密钥。设待加密信息为 m，加密密钥为 $k1$、$k2$，3DES 加密结果为：$E(D(E(m,k1),k2),k1)$。

(1) $D(E(m,k1),k1)$的结果是什么？

(2) $D(E(m,k1),k2)$形式上是解密，为什么效果上是加密？

(3) $E(D(E(m,k1),k1),k1)=E(m,k1)$成立吗？

2. RSA 算法。

(1) 令 $p=3$，$q=11$，使用 RSA 算法生成对应密钥对；

(2) 使用生成密钥对，用户 A 将明文“men”加密后传递给用户 B，实现数据传输的机密性。写出明文“men”加解密的具体过程，此时密钥对的所有者是谁？

(3) 使用生成密钥对，将明文“men”签名后传递给用户 B 实现源身份认证。写出明文“men”签名和验证的具体过程，此时密钥对的所有者是谁？

3. RSA 算法具有乘法同态性质。

(1) 加密算法具有乘法同态性质是指明文 a、b 满足：

$D(E(a,k_{公})\times E(b,k_{公}),k_{私})=a\times b$。证明 RSA 算法具有乘法同态性质。

(2) 设长方形的宽 $w=10$，高 $h=9$，验证 RSA 算法具有乘法同态性质。

(3) Alice 使用(2)生成的公钥加密 w、h 生成 $E(w,k_{公})$、$E(h,k_{公})$发送给 Cloud，Cloud 计算 $E(w,k_{公})\times E(h,k_{公})$返回结果给 Alice，Alice 使用私钥解密 $D(E(w,k_{公})\times E(h,k_{公}),k_{私})$得到结果 $w\times h$。据此应用场景说明同态加密可以保护云存储数据安全。