河北工业大学

硕士学位论文

数字签名技术的研究及应用

姓名：贾超

申请学位级别：硕士

专业：计算机应用技术

指导教师：武金木

20061101

数字签名技术的研究及应用

摘　要

随着信息技术的发展，特别是互联网技术的迅速普及，人们通过网络可以获取大量的信息，也可以通过网络传播大量的信息，那么如何确认这些消息的可靠性。特别是在涉及到诸如经济方面的问题比如进行商务活动，如何防止交易方抵赖避免自己蒙受损失。又比如，进行电子政务，如何避免一些人伪造文件以及签署文件等等。诸如此类问题，如何解决，课题的研究内容数字签名技术就是为了解决这方面问题的。

课题的研究内容主要包括以下方面：

首先，是对密码学基础的讨论。对于密码学其主要分为两大块：对称密码学和非对称密码学。课题中对它们的概念、原理及优缺点进行了分析和讨论。

其次，是对于目前几种比较常用的签名体制进行了分析和比较,包括 MD5、RSA、DSS、ECDSA 等签名体制。接着对数字签名技术在现实世界中的应用进行了分析和讨论,包括电子邮件、电子商务以及远程通信等。

最后，针对对称加密方法、公开密钥加密方法和单向散列加密方法的优点和不足，提出并实现了保密散列数字签名算法。保密散列数字签名算法利用 RSA 公钥密码体制的原理、MD5 单向散列函数和排列码加密的思想，构造了一种能够快速实现数据保密、数字签名的签名方案。

关键词：RSA 算法，排列码，数字签名，保密散列数字签名

THE RESEARCH AND APPLICATION OF SIGNATURE TECHNOLOGY

ABSTRACT

With the development of the information technology, especially with the quick popularization of the Internet technology, people can acquire much information through the network, while how to confirm the reliability of the information. Especially the problem concerning the economy sides, such as commercial activities, how to avoid the loss owing to the transaction side's disavowal, further more, while going to do some official tasks, people maybe hope not to be imitated. Facing these troubles, people often ask how to solve them. In this paper, the main task is to try to solve some problems through using digital signature technology.

This paper includes the content as follows.

Firstly, it discusses the foundation of cryptography. Cryptography is divided into two main parts: symmetric cryptography and public-key encryption scheme. It analyzes and discusses their conception, principle, advantage and disadvantage respectively.

Secondly, it analyzes and compares among several normal signature schemes, including MD5, RSA and DSS etc. Then it analyzes digital signature's application in the actual world. The content includes email, e-commerce and remote communication and so on.

Lastly, It puts forward and fulfills the secret hash digital signature algorithm according to the advantage and disadvantage of symmetric algorithm, public-key algorithm and one-way hash function. This algorithm makes use of the principle of RSA public-key algorithm, MD5 one-way hash function and permutation code at the same time. It constructs a signature scheme that can quickly realize the data security and data signature.

KEY WORDS: RSA, permutation code, digital signature, secret hash digital signature

第一章 绪 论

§1-1 课题的研究背景

随着计算机技术和通信技术的迅速发展和应用，人类进入了信息时代。信息技术正从政治、经济、和社会生活的各个方面改变着人类的生存和发展方式，无论是在打开电灯、乘坐飞机、应急求助等日常事务中，还是在国家经济运行、安全防御等方面，都需要依赖于复杂的信息系统的支持，然而这个充满希望的时代也充斥着危险，当人们通过网络来交流思想、吐露内心的秘密、商讨业务、支付金钱，甚至传递涉及国家的重要经济、外交和军事信息时，都可能被他人窃取秘密。因此，如果人们希望继续享用信息时代所带来的种种益处，那就必须建立有效的安全防护措施，这就是所谓的信息安全。

密码技术正是保障信息安全的有效方法之一，不仅可以利用密码来打乱所传达的信息，以便只有我们所希望的接受者才能真正获取信息，而且通过密码技术的使用，使得现实世界的许多事情，诸如签字、订合同、发送收据等都可以通过数字信号来完成。

由于互联网技术的普及，使得网络安全和信息安全息息相关，使得通常谈到信息安全，人们的第一反应就是互联网，不过确实由于互联网技术的如火如荼，网络信息安全因此特别重要。课题也是侧重互联网中的信息安全相关部分。

在信息技术的应用过程中，信息是最为宝贵的资源，因特网为信息的传播和获取提供了极大的便利，它可以使我们不受时间和空间的限制，和世界上任何一个角落的个人或组织进行信息交流，而且每天发生的各种政治事件较以往任何时候能以最快的速度、在最短的时间内向全世界传播，各种经济信息更是充斥网络让人应接不暇。

实际上，信息安全的范围是非常广泛的，不过目前对于信息安全也没有一个确切的定义。信息安全问题是在通信的过程中产生的，而通信又可分为物理层、控制层和服务层等多种层面。从不同的观察角度，所关心的信息安全问题又有不同：用户关心的是涉及个人隐私和商业利益的数据在通信过程中的秘密性、完整性和真实性问题；运营商主要关心的是系统可靠性问题、信息存取控制问题、病毒问题、黑客问题等等；国家安全保密部门和行政部门关心的是国家机密信息的有效过滤和防堵、避免非法泄露的问题。

总之，信息安全[1]应该是综合运用一切手段、使信息从信源到信宿的整个过程中在通信的各个层面和不同的观察角度，在获取、存贮、显示、变换、处理、传递等各个环节上保证信息的可靠性、可用性、可控性、完整性、秘密性和不可抵赖性的理论和技术。

今天的信息社会里，科学技术的研究和发展及商业等方面，无一不与信息息息相关。所以信息就是

生命，信息就是时间，信息就是财富。由于信息是共享的，信息的扩散会产生社会影响，但是如果错误的消息则可能起到相反的作用，不但浪费时间，而且还会浪费金钱。那么如何知道消息是准确的，可靠的？课题要讨论的数字签名问题就是针对此类问题而来的。

§1-2 国内外现状

密码和信息安全技术大量使用于实际的网络通信中，标准化必然是一项重要的工作，数字签名的标准制订是其必备部分之一。

影响较大的制订信息安全相关标准的组织有：ISO 和国际电子技术委员会(IEC)，美国国家标准与技术委员会(NIST)制定的美国联邦信息处理标准(FIPS)系列，Internet 研究和发展共同体制制订的标准，IEEE 制定的标准，RSA 公司制定的 PKCS 系列等等。

对于数字签名，比较有代表型的有：

美国的 NIST 制定的数字签名标准（DSS：Digital Signature Standard），对应的数字签名算法是 DSA。

公钥密码学标准(PKCS：Public-Key Cry Standard)是 RSA 数据安全公司制定的公钥密码学的工业标准接口，也是第一个公钥密码学的工业标准接口，其中 PKCS#7 对用 RSA 数字签名算法的通用语法及数字信封作出了规定。

椭圆曲线数字签名算法[2] (ECDSA：Elliptic Curve Digital Signature Algorithm)。相对于RSA等密码体制来说，椭圆曲线加密机制是比较新的技术。ECDSA是由IEEE工作组和ANSI X9 组织研究，于 1998 年作为ISO标准被采纳，1999 年被接受为ANSI标准，2000 年被IEEE和NIST采纳为标准。

对于数字签名算法来说，它们都要用到一个杂凑函数(Hash)，利用它们的单向性实现验证和鉴别的目的。

对于我国的数字签名现状：由于我国起步较晚，因此主要是借鉴国外的经验，而且在操作中还很不规范。鉴于此，我国于 2004 年 8 月 28 日在第十届全国人民代表大会常务委员会第十一次会议通过了《中华人民共和国电子签名法》，从而规范电子签名的应用。

§1-3 目的和意义

随着网络技术的发展，网上银行、网上合同、电子商务、电子政务等应用越来越广泛，而幕后重要的“英雄”就是电子签名，因为通过它实现了身份的确认，从而实现了人们以前需要现场处理的功能。而作为电子签名的支撑是密码学中的杂凑函数，是通过它实现了类似“指纹”的功能。

通过数字签名不但可以实现防伪的作用，而且还可以实现信息的防止抵赖的作用，这样可以加大诸

如网上交易的可信性。如果网络上也解决了诚信问题，那么必然会加大物流，实现人力资源以及自然资源等资源的合理流动，为社会繁荣做出巨大的贡献。

课题中以现代加密技术为基础，综合利用公开加密算法、对称加密算法和单向散列算法，提出具有保密功能的新型数字签名技术。在软件实现方面，采用了 Montgomery 大数模幂优化算法，以解决制约方案性能瓶颈的大数模幂运算问题，使得 RSA 公开密钥算法运算速度增加，使得软件实现公开加密方法成为可能，同时采用 MD5 算法和排列码算法实现保密散列数字签名。

§1-4 课题的组织与结构

课题的组织与结构如下：

第一章绪论概述了课题的选题背景，信息安全技术的重要性以及数字签名技术的国内外研究现状。

第二章主要内容是密码学的基本概念，密码技术发展现状，还分析了课题采用的加密技术，数字签名技术，时间戳技术。

第三章主要内容是对目前几种比较常用的签名体制进行了分析和比较,包括 MD5、RSA、DSS、ECDSA 等签名体制。

第四章对电子签名技术在现实世界中的应用进行了分析和讨论,包括电子邮件、电子商务以及远程控制等。

第五章主要内容是排列码加密算法以及保密散列数字签名算法。排列码是一种新型的分组密码加密方法，它提出多对多的加密新概念，即：加密过程中记住的不仅仅是加密的结果，而且加密的密钥中还包含着由明文到密文的路径，解密时再顺原路变换回去。在此基础上实现的保密散列数字签名方法具有更强的可信度，更强的防抵赖、防伪造和防复制能力。

第六章实现了一个保密散列数字签名的系统模型,并对在该系统中采用的一些相关的密码技术进行了详细的描述。

第二章 密码学与数字签名技术

§2-1 密码学基础

2-1-1 加密的历史

作为保障数据安全的一种方式，数据加密起源于公元前 2000 年。埃及人是最先使用特别的象形文字作为信息编码的人。随着时间推移，巴比伦、美索不达米亚和希腊文明都开始使用一些方法来保护他们的书面信息。对信息进行编码曾被凯撒大帝使用，也曾用于历次战争中，包括美国独立战争、美国内战和两次世界大战。最广为人知的编码机器是 German Enigma机，在第一次世界大战中德国人利用它创建了加密信息。此后，由于 Alan Turing和Ultra计划以及其他人的努力，终于对德国人的密码进行了破译。当初，计算机的研究就是为了破解德国人的密码，当时人们并没有想到计算机给今天带来的信息革命。随着计算机的发展，运算能力的增强，过去的密码都变得十分简单了，于是人们又不断地研究出了新的数据加密方式如对称密钥算法和公开密钥算法[3]。

今天，密码学已经从最初的军事和外交领域走向公开，它是结合数学、计算机科学、电子与通信等诸多学科于一身的交叉学科，它不仅具有保证信息机密性的信息加密功能，而且具有数字签名、身份验证、秘密分存、系统安全等功能。所以，使用密码技术不仅可以保证信息的机密性，而且可以保证信息的完整性和确证性，防止信息被篡改、伪造和假冒。

密码学是以研究秘密通信为目的的。即研究对传输信息采用何种秘密的变换以防止第三者对信息的窃取。它作为数学的一个分支，包括密码编码学（cryptography）和密码分析学（cryptanalysis）两部分。使消息保密的技术和科学叫做密码编码学，密码分析学就是破译密文的科学和技术，即揭穿伪装。这两者是互相促进，共同发展的。

根据一个算法被破译的难易程度，不同的密码算法具有不同的安全等级。如果破译该算法的代价大于加密数据的价值，或者破译算法所需要的时间比加密数据要求保密的时间更长，或者用单密钥加密的数据量比破译算法需要的数据量少得多，则加密的数据可能就是安全的。密码学就是为了在可能的代价范围内尽量保障一个加密算法的加密强度，使之难以破译。

2-1-2 数据加密

尽管加密是为了安全目的，对信息进行编码和解码这个概念十分简单，但在这里仍需对其进行解释。数据加密的基本过程包括加密和解密。对被称为“明文”的可读信息采用某种方法进行伪装以隐藏它的内容，变换成称为“密文”或“密码”的代码形式，这个过程叫做加密（encryption）。该过程的逆

过程为解密(decryption)，即将该编码信息转化为其原来的形式的过程[4]。如图 2.1 所示。

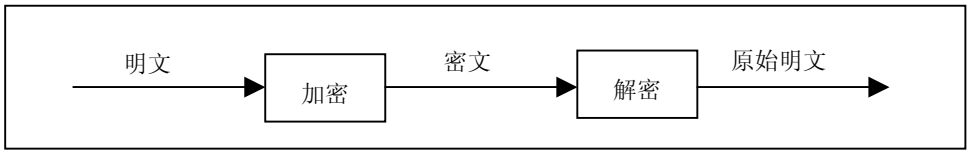

图 2.1 加密和解密

Fig.2.1 Encryption and decryption

2-1-3 加密算法和密钥

密码算法(algorithm)是用于加密和解密的数学函数。(通常情况下，有两个相关的函数：一个用于加密，另一个用于解密)

如果算法的保密性是基于保持算法的秘密，即算法必须保密，这种算法称为受限制的(restricted)算法。这种算法属于最初级的的加密方法，不能满足现代的加密要求。大的或经常变换的用户组织不能使用它们，因为如果有一个用户离开这个组织，其他的用户就必须改换另外不同的算法。如果有人无意暴露了这个秘密，所有人都必须改变他们的算法。而且，受限制的密码算法不可能进行质量控制或标准化。每个用户组织必须有他们自己的唯一算法。这样的组织不可能采用流行的硬件或软件产品，因为窃听者可以买到这些流行产品并学习算法。

现代密码学用密钥(key)解决了这个问题，密钥用 K 表示。K 可以是很多数值里的任意值。密钥 K 的可能值的范围叫做密钥空间(keyspace)。加密和解密运算都是用这个密钥(即运算都依赖于密钥，并用 K 作为下标表示)，这样，加/解密函数，如式(2.1,2.2,2.3)所示。

$$E_K(M) = C \tag{2.1}$$

$$D_K(C) = M \tag{2.2}$$

$$D_K(E_K(M)) = M \tag{2.3}$$

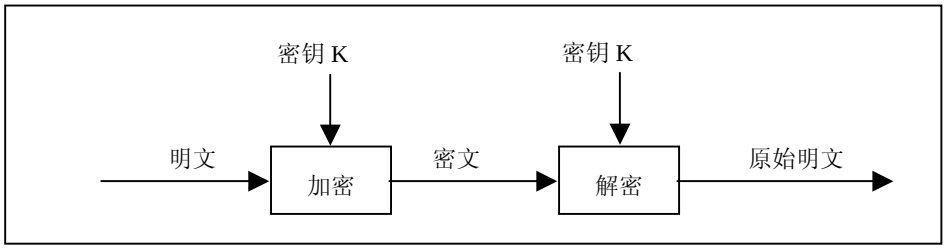

图 2.2 使用一个密钥的加/解密

Fig.2.2 Encryption/decryption by single key

有些算法使用的加密密钥与解密密钥不相同，如图 2.3 所示，也就是说加密密钥 K_1 与相应的解秘

密钥 K_2 不同，在这种情况下，如式(2.4,2.5,2.6)所示。

$$E_{K_1}(M) = C \tag{2.4}$$

$$D_{K_2}(C) = M \tag{2.5}$$

$$D_{K_2}(E_{K_1}(M)) = M \tag{2.6}$$

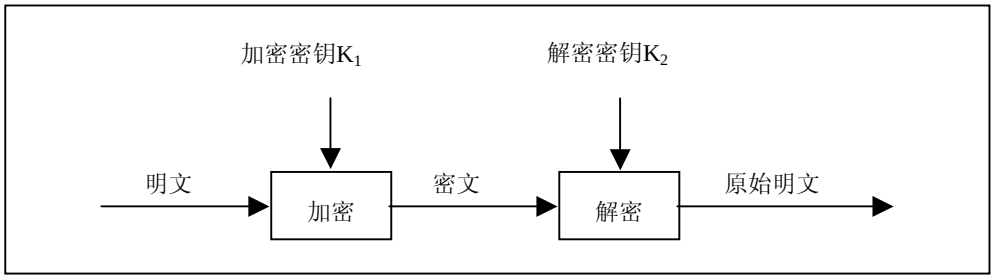

图 2.3 使用两个密钥的加/解密

Fig.2.3 Encryption/Decryption by two keys

所有这些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，即使偷听者知道算法也没有关系。只要他不知道使用的具体密钥，就不可能阅读消息。

2-1-4 对称密钥算法

基于密钥的算法通常有两类：对称密钥算法和公开密钥算法[5]。

对称密钥算法(symmetric-key algorithm)有时又叫传统密码算法，就是加密密钥能够从解密密钥中推算出来，反过来也成立。在大多数对称密钥算法中，加/解密密钥是相同的。这些算法也叫秘密密钥算法或单密钥算法，它要求发送者和接收者在安全通信之前，商定一个密钥。对称密钥算法的安全性依赖于密钥，只要通信需要保密，密钥就必须保密。

对称算法可分为两类。对位流或字节流进行操作运算的算法称为序列密码或流密码(stream cipher)。另一类算法是对明文的一组位进行运算，这些位组称为分组或块(block)，相应的算法称为分组密码或块密码(block cipher)。课题中使用的排列码加密算法属于分组密码，是一种新型的对称加密算法。

2-1-5 公开密钥算法

公开密钥算法(public-key algorithm)出现于 1976 年。它最主要的特点就是加密和解密使用不同的密钥，每个用户保存着一对密钥——公开密钥 PK 和私有密钥 SK，因此，这种体制又称为双钥或非对称密钥密码体制[6]。

在这种体制中，PK 是公开信息，而 SK 需要由用户自己保密。加密算法 E 和解密算法 D 也都是公开的。虽然 PK 与 SK 是成对出现，但却不能根据 PK 计算出 SK 。

公开密钥算法的特点如下：

用公开密钥 PK 对明文 M 加密后，再用私有密钥 SK 解密，即可恢复出明文，如式(2.7)所示。

$$D_{SK}(E_{PK}(M)) = M \tag{2.7}$$

加密密钥不能用来解密，如式(2.8)所示。

$$D_{PK}(E_{PK}(M)) \neq M \tag{2.8}$$

明文同样可以用私有密钥加密而用公开密钥解密，如式(2.9)所示。

$$D_{PK}(E_{SK}(M)) = M \tag{2.9}$$

这就是用作数字签名(Digital Signature)。

在公开密钥密码体制中，最著名的一种是RSA体制，是根据它的三个发明者来命名的，即Rivest，Shamir和Adleman。下面简要介绍RSA体制[7]的原理。

1) 用户选择 2 个足够大的秘密素数 p 和 q (至少为 100 位以上的十进制数，最好是 300 位)。

2) 令 $n = pq$ ， n 是公开的。以目前的数学知识，由 p 和 q 计算 n 很容易，但从 n 分解出因子 p 和 q 是极其困难的，这正是公开密钥加密方法的前提和基础。

3) 计算 n 的欧拉函数 $\varphi(n) = (p-1)(q-1)$ ， $\varphi(n)$ 即小于等于 n 并与 n 互素的数的个数。选择一个相对较大的整数 e 作为加密指数，使 e 与 $\varphi(n)$ 互素。

4) 用欧几里德扩展算法解同余方程 $ed \equiv 1 \bmod \varphi(n)$，求出解密指数 d 。

若用整数 M 、 C 分别表示明文、密文，则以下公式可用于加密和解密(M 、 C 均小于 n)：

加密： $C = M^e \bmod n$ 。

解密： $M = C^d \bmod n$ 。

每个用户都有一组密钥(e, d, n)。对这种体制，只有(e, n)是出现在公开手册上的(即公开密钥 PK)。 d 则是需要用户保密的(即私有密钥 SK)。

RSA 体制的保密性在于对大数的因数分解很花时间。因此，当 n 足够大时，在目前情况下，对 n 进行因数分解实际上是无法实现的。

§2-2 数字签名技术

在文件上手写签名长期以来被用作作者身份的证明，或至少同意文件的内容。那么在计算机上呢？可以用数字签名(Digital Signature)来实现与文件上手写签名相同的功能。所谓数字签名，就是只有信息发送者才能产生的别人无法伪造的一段数字串，这段数字串同时也是对发送者发送信息真实性一个证

明。数字签名也称为电子签名，是公钥密码系统的一种重要应用方式。

2-2-1 数字签名的特征和作用

作为一种签名方式，数字签名与书面文件上的手写签名有着共同的特征和作用[8]：

1) 签名是可信的：如果接收者能够用签名者的公开密钥解密，他就能够确定签名者的身份。

2) 签名不可伪造：只有签名者知道他的私人密钥，别人无法伪造他的签名。

3) 签名不可重用：签名是文件的一部分，不法之徒不可能将签名移到另一个文件上。

4) 被签名的文件是不可改变的：如果被签名的文件有任何改变，那么该签名文件就不可能用签名者的公开密钥进行解密。

5) 签名是不可抵赖的：因为别人不知道签名者的私人密钥，不可能产生同样的签名文件，因此签名是不可能抵赖的。

数字签名的原理与此相似，信息的发送者即签名者利用自己的私人密钥对要发送的消息进行签名，也就是通过对消息或者作为消息函数的少量数据应用加密算法，从而使文件得以签署。由于数字签名是只有信息的发送者本人才能产生的别人无法伪造的一段数字串，因此这段数字串可作为对发送者发送信息真实性的一个证明，即身份验证得以证实。除了对消息来源的确认，数字签名的另一个重要功能就是对数据完整性的验证。也就是说数字签名可以用于防止下列情况的发生：冒用他人身份发送电子信息；发出或收到某个电子信息后又加以否认。

2-2-2 数字签名原理

目前，数字签名是建立在公开密钥体制基础上[9]的，现有的多种数字签名算法都是公开密钥算法，用秘密消息对文件签名，用公开消息去验证，是公开密钥加密技术的另一类应用。在实际的实现过程中，采用公开密钥密码算法对长文件签名效率太低，为了节约时间，数字签名协议经常和单向散列函数一起使用。现在以图 2.4 来说明数字签名方案的原理。

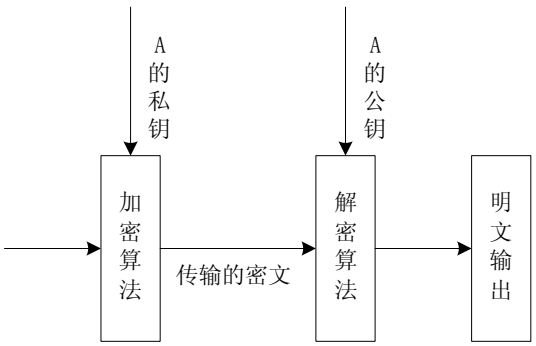

图 2.4 数字签名原理

Fig.2.4 Principle of digital signature

如图 2.4 所示，如果 A 要向 B 发送一个消息，尽管该消息本身的保密性可能并不重要，但 A 希望 B 能够确认该消息确实是 A 发出的，并且消息在传输过程中没有被改动，即要实现消息真实来源的验证和消息的完整性验证。

在这种情况下 A 使用自己的私人密钥来加密消息。如果 B 收到 A 的密文消息后，能够用 A 的公开密钥进行解密，这样就验证了该消息一定是由 A 发出的。因为除了 A 以外，没有其他人能够创建出可以用 A 的公开密钥来解密的密文来。并且因为如果没有 A 的私人密钥就不可能对消息进行改动，因此在消息的真实来源得以验证的同时，消息的数据完整性也能够得到验证。

数字签名是不可抵赖的。即使 A 以后声称他没有发送这个消息给 B，但由于除了 A 以外，没有人能够生成同样的密文，这就说明 A 在说谎。

§2-3 数字时间戳

2-3-1 数字时间戳

在书面合同中，文件签署的日期和签名一样是十分重要的防止文件被伪造和篡改的关键性内容。数字文件同样也有这个要求，因此，数字签名经常包括时间标记，通常称为时间戳（time-stamp）[10]。

数字时间戳协议满足：

1) 数据本身必须有时间标记，与存储媒介无关。

2) 对文件的任何更改都会被发现。

3) 不可能用不同于当前日期和时间的日期和时间来标记文件，即使用一个公认的或标准的时间。

2-3-2 时间戳协议

下面给出一个时间戳协议，在协议中引入一个仲裁 Tom，这个协议中还包括一个用户 Alice。假设 Alice 希望对信息 M 标记时间，使用的单向散列函数为 $H()$，设时间为 t，Tom 对数据的签名函数为 $S()$：

1) Alice 产生信息 M 的单向散列值 $H(M)$。

2) Alice 将散列值 $H(M)$ 传送给 Tom。

3) Tom 将收到的散列值时的日期和时间附在散列值后，得到 $(H(M),t)$。

4) Tom 对 $(H(M),t)$ 进行数字签名，得到 $S(H(M),t)$。

5) Tom 将签名的散列值和时间标记 $S(H(M),t)$ 发送给 A。

很显然，只要 Tom 是可信的，任一得到 $(M,S(H(M),t))$ 的用户，都可确定信息 M 在时间存在 t。这个方法是可行的。首先，它提供了数据的保密性，Alice 将信息的单向散列值而非信息本身发送给 Tom；

其次，Tom 接受到的是数据的单向散列值，这并不是海量数据，因此该方法从效率上考虑是可行的；最后，Tom 无须保存标记时间值 $S(H(M),t)$，这使系统的风险进一步下降。

2-3-3 权威机构提供实证

为了提供时间标记服务，需要一个可信赖的第三方机构—时间标记权威机构(TSA：Time Stamp Authority)，它为指定数据(或信息)提供即时的存在证明服务。在实际的时间标记服务中，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TSA，请求TSA对其所发送请求中的指定数据进行时间标记，TSA在接受到该请求后，在此指定的数据后加上当时的日期、时间，并对整个包进行数字签名，然后TSA将此数据，时间及签名发送给请求实体[11]。

可以看出，时间戳事实上是一个经加密后形成的凭证文档，它包括三个部分：需加时间戳的文件的摘要（digest）、TSA 收到文件的日期和时间、TSA 的数字签名。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，它是由认证单位 TSA 来加的，以 TSA 收到文件的时间为依据。

§2-4 数字签名的安全性

2-4-1 数字签名可能存在的攻击

攻击者攻击一个密码算法通常采取的攻击方法可以分为两大类[13]。其一是确定性分析法，其二是统计分析法。确定性分析法是指利用一个或几个已知量，用数学关系式表示所求未知量，已知量和未知量的关系视签名算法和验证算法而定，寻求这种关系是确定性分析法的关键步骤，现在对数字签名方案的攻击绝大多数采用这种方式。统计分析法是指利用消息及其相应签名的某种概率关系来进行攻击的方法。这种攻击方法目前对基于公钥体制的方案来说是不大适用的，它主要是用来攻击分组密码和流密码。

与讨论加密方案时的情况一样，对签名方案也有不同类型的攻击。敌手攻击一个数字签名方案通常采用以下基本攻击类型[14]。

1) 唯密钥攻击(Key-only attack)：敌手仅知道签名者的公开密钥，而没有其它任何信息。

2) 已知签名攻击(Know-signature attack)：敌手知道签名者的公钥并且看到了一些签名者产生的消息签名对。

3) 消息攻击：敌手在试图攻击一个签名方案之前，能够获得某些消息或选定的消息及其签名。这其中又包括选择消息攻击和适应性选择消息攻击两类：

(1) 选择明文攻击(Chosen-message attack)：敌手可选择一个消息列表并要求合法签名人签名这些消息。

(2) 适应性选择明文攻击(Adaptively-chosen-message attack)：敌手可以适应性地选择消息让签名人

签名，他可以选择一些消息并得到相应的签名，然后进行密码分析，并根据他的分析结果，选择下一个要签名的消息，然后继续上述过程。

如果一个攻击者能够以不可忽略的概率达到如下的至少一项的目的，那么我们说这个攻击者成功的攻破了他所试图攻击的数字签名方案。攻破数字签名方案的攻击类型[15]：

1）完全攻击：攻击者计算出签名者的秘密陷门信息即签名密钥。

2）一般伪造：找到一个功能等效于签名者的签名算法的有效算法(基于可能不同的但等效的陷门信息)，伪造合法签名。

3）存在性伪造（Existential forgery)：敌手有能力伪造至少一个消息的签名，敌手对他获得的消息及其签名没有任何控制作用。

4）选择性伪造(Selective forgery)：敌手能成功伪造她优先选择的一些消息的签名选择伪造。

5）完全伪造(Universal forgery)：虽然不能找到签名人的私钥，但攻击者能够伪造所有消息的签名。

虽然大多数数字签名方案的安全性并没有得到证明，但是多年来也并没有明显的验证表明这些方案是不安全的，例如基于大数分解问题的RSA方案。因此，把一些数字签名方案的安全性转化为这些公认安全的数字签名方案的安全性，是一个实际有效的保证数字签名方案安全性的方法。同时，利用计算复杂度理论，将数字签名方案的安全性转化为一些已知不可解的数学难题，就是通常所说的可证明安全性的研究[16]，也是近年来的研究热点之一。但是，如此证明安全的数字签名方案普遍存在效率偏低的问题。在可证明安全性的研究中，还有一种是将数字签名算法中常用到的Hash函数看作一个Random oracle模型。假设Hash函数没有弱点的情况下，基于该模型的证明能够保证一个签名方案的安全性。例如著名的Schnorr签名方案在这一模型下被证明是安全的。同时因为Schnorr[17]签名方案是一个基本的数字签名方案，因此有许多具有特殊性质的签名方案的安全性都可以基于Schnorr签名来进行设计和分析。

2-4-2 对策

有两种特别的措施可以提高数字签名的安全性[18]。一种是签名前首先增加消息的冗余度。此时，只有满足某种断言的消息才予以签名，这使得选择明文攻击变得更复杂。而且，只有当消息满足这个断言时签名才是有意义的，这样，在原方案中的扩展伪造看起来将不可能产生有效的消息。第二个措施是签名前对消息使用单向Hash函数。此时，攻击者不可能为选择明文攻击找到对他有用的消息，而且消息的Hash值就是实际签名的值。在原来的方案中，扩展伪造应该产生一段消息，其Hash函数下的原像是不知道的。使用Hash函数还有一个额外的优点，就是如果Hash函数的执行速度快，它将使得整个方案更有效。因此，选择好的冗余函数或者Hash函数，能够对方案的安全性提供更好的保障。目前有一些新的数字签名方案试图摈弃冗余函数或者单向函数的应用，通常都会导致所给方案能够很容易地通过唯密钥攻击进行存在性伪造。

还有一种保障数字签名算法安全性的方法是应用可证明安全性的思想，通过引进随机预言机模型(Random oracle model)，形式化数字签名算法中的单向函数。

数字签名的安全性主要有以下三种证明方法。

1）可证明安全性

（1）标准模型下：利用计算复杂性理论，将数字签名的安全性转化为一些已知的难题，如大整数分解问题，离散对数问题或者一般NP[18]完全问题。

（2）随机预言机模型下：数字签名经常使用Hash函数，为了给签名方案提供安全性证明，一些学者建议将Hash函数看成是一个随机函数，并给出了相应的模型，即随即预言机模型[19]。假设Hash函数是安全的情况下，基于此模型的证明能够保证一个签名方案的总体设计的安全性。

2）转化证明

把需要分析和确定其安全性的数字签名方案的安全性转化为一些公认安全的数字签名方案的安全性，也是一个有效可行的方法。

第三章 几种数字签名方案及分析

§3-1 RSA 签名体制

1 RSA 算法

RSA 算法是目前网络上进行保密通信和数字签名的最有效安全算法。RSA 算法的安全性基于数论中大素数分解的困难性。所以，RSA 需采用足够大的整数。因子分解越困难，密码就越难以破译，加密强度就越高。其公开密钥和私人密钥是一对大素数的函数。从一个公开密钥和密文中恢复出明文的难度等价于分解两个大素数之积。因式分解理论的研究现状表明：所使用的 RSA 密钥至少需要 1024 比特，才能保证有足够的中长期安全。

2 RSA 签名体制的实现

用 RSA 算法做数字签名也很简单。总的来说，就是签名者用私钥参数 d 加密，也就是签名；验证者用签字者的公钥参数 e 解密来完成验证。如图 3.1，3.2 所示。

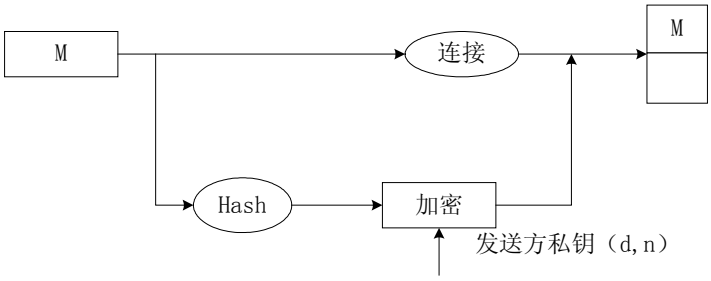

图 3.1 发送方

Fig.3.1 Transmission

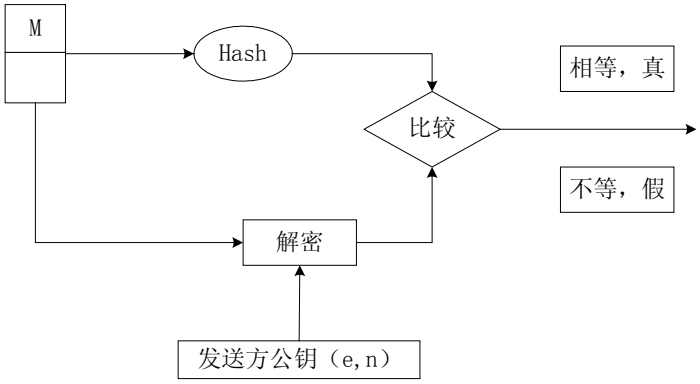

图 3.2 接收方

Fig.3.2 Acception

使用发送方的私钥(d,n)对消息M的散列值加密，既有身份辨别又有数据完整性保护的功能。

§3-2 DSS 签名体制

讨论DSS签名体制实际上讨论其对应的DSA算法。DSA（Digital Signature Algorithm，数字签名算法，用作数字签名标准的一部分），它是另一种公开密钥算法，它不能用作加密，只用作数字签名[13]。DSA使用公开密钥，为接受者验证数据的完整性和数据发送者的身份。它也可用于由第三方去确定签名和所签数据的真实性。DSA算法的安全性基于解离散对数的困难性，这类签字标准具有较大的兼容性和适用性，成为网络安全体系的基本构件之一。

DSA签名算法[21]中用到了以下参数：

p是L位长的素数，其中L从512到1024且是64的倍数。q是160位长且与$p-1$互素的因子。其中h是小于$p-1$并且满足大于1的任意数。x是小于q的数。

另外，算法使用一个单向散列函数$H(m)$。标准指定了安全散列算法（SHA）。三个参数p，q和h是公开的，且可以被网络中所有的用户公有。私人密钥是x，公开密钥是y。

对消息m签名时：

1）发送者产生一个小于q的随机数k。

2）发送者产生：r和s就是发送者的签名，发送者将它们发送给接受者。

3）接受者通过计算来验证签名：如果$v=r$，则签名有效。

DSA 签名：

1）公开密钥：

（1）p：512 位到 1024 位的素数。

（2）q：160 位长，并与$p-1$互素的因子。

其中，h是小于$p-1$并且满足大于 1 的任意数。

2）私人密钥：x小于q。

3）签名：k选取小于q的随机数。

4）验证： 如果$v=r$，则签名被验证。

§3-3 ECDSA 签名体制

椭圆曲线数字签名算法(ECDSA)[22]设计的数学原理是基于椭圆曲线离散对数问题的难解性[23]。EC点上离散对数的研究现状表明：所使用的ECDSA密钥至少需要 192 比特，才能保证有足够的中长期安

全。

椭圆曲线是指由韦尔斯特拉斯(Weierstrass)方程，如式(3.1)所示。

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \tag{3.1}$$

所确定的平面曲线。定义 F 为一个域，其中 $a_i \in F$ ， $i = 1,2,\cdots 6$。 F 可为有理解域、实数域、复数域，也可为有限域 GF(q)。在椭圆曲线密码体制中， F 一般为有限域。由有限域椭圆曲线上的所有点外加无穷远点组成的集合，连同按照“弦切法”所定义的加法运算构成一个有限 Abel 群。在此有限 Abel 群上，定义标量乘法(Scalar Multiplication)为： $mP = P + P + \cdots P$ (m 个 P 相加)；若 $mP = Q$，定义： $\mathrm{m}\, m = \log_P Q$ 为椭圆曲线点群上的离散对数问题，此问题无多项式时间内的求解算法。ECDSA 的设计正是基于这一问题的难解性。

在此，讨论定义在有限域 $GF(2m)$ 上的椭圆曲线数字签名算法。今定义椭圆曲线方程如式(3.2)所示。

$$y^2 + xy = x^3 + ax^2 + b, a, b \in GF(2m) \tag{3.2}$$

则椭圆曲线的域参数为 $D(m, f(x), a, b, P, n)$。其中 $f(x)$ 为 $GF(2m)$ 的多项式基表示的不可约多项式。 P 表示椭圆曲线上的一个基点， n 为素数且为点 G 的阶。

ECDSA 算法密钥对的生成过程为：在区间 $[1, n-1]$ 上选择一个随机数 d ，计算 $Q = dP$ ，则 Q 为公钥， d 为私钥。

ECDSA 算法的签名生成过程可简述如下：若签名的消息为 e，则在区间 $[1, n-1]$ 上选择一个随机数 k ，计算 $kG = (x_1, y_1)$； $r = x_1 \bmod n$； $s = k - 1(e + dr) \bmod n$。如果 r 或 s 为零，则重新计算，否则生成的签名信息为 (r, s)。

ECDSA 算法的签名验证过程可简述如下：

若公钥为 Q ，签名的消息为 e 计算： $w = s - 1 \bmod n$ ； $u_1 = ew \bmod n$ ； $u_2 = rw \bmod n$ ； $X = u_1P + u_2Q = (x_1, y_1)$。

如果 X 为无穷远点，则拒绝签名，否则计算： $v = x_1 \bmod n$。

如果 $v = r$ ，则接受签名，否则拒绝签名。

§3-4 其它数字签名体制

由于签名体制很多，所以只列出其它的一些比较有代表性的签名方案或算法。

1 shamir 背包数字签名

公钥密码的第一个算法就是基于背包难题的MH算法，但是只能用于加密，而不能用于数字签名。后来Adi Shamir[24]将它改进为能够用于数字签名。改进后的算法就只能数字签名，而不能用于加密。

2 Rabin 数字签名

该算法是Rabin[25]在 1979 年提出的一个公钥密码体制算法，其安全性是基于分解因子的计算困难性之上。该算法已经作为ISO/IEC9796 建议的数字签名标准算法。

3 GOST 数字签名

该算法与DSA非常相似，是俄罗斯采用的数字签名标准[26]，于 1995 年启用，其官方的名称为GOST R34. 10-94。DSA中使用的散列函数是 SHA，GOST算法中使用的散列函数称为 $H(x)$。 $H(x)$ 是在分组密码算法GOSY78147-89 基础上建立的。

4 OSS 数字签名

本数字签名算法由 Ong，Schnorr 和 Shamir 于 1984 年提出的，是基于模 n 的多项式的算法。

5 基于有限自动机的数字签名

我国学者陶仁骥、陈世华于 1985 年公开提出了基于有限自动机[27]（FiniteAutomation）理论的公钥密码体制。一般称为FA密码体制。FA密码体制其建立在有限状态自动机可逆性研究基础之上。从功能上讲，有限自动机就是将输入序列转换为输出序列。如果有限自动机 M 满足一定的结构要求，那么 M 的输出序列可逆，而可逆的过程可以用另外一个有限自动机 M' 来完成。更进一步，如果 M 、M' 满足安全性要求，那么就可以用 M 作为加密器， M' 可作为相应的解密器，从而构成一个FA密码体制。当然在此基础上可以构成数字签名系统。

6 ESINGN 数字签名

本算法是由日本 NTT 实验室提出的一个数字签名方案。对同样长度的密钥和签名长度，它的安全性至少和 RSA 和 DSA 一样，并且其运算速度也比这两个算法快。2001 年，该算法被 IEEE 采纳，作为公钥加密和认证标准，编号为 IEEE P1363，同年，该算法入围了 NESSIE 的第二阶段的筛选。

7 Lamport-Diffie 数字签名

该签名是由Lamport提出的的，又由Diff和Hellman给出了算法的具体的描述，所以称为Lamport-Diffie数字签名[28]。签名算法的基础是利用一组密钥，而密钥的个数是由报文分组的长度决定的。值得一提的是，该算法是利用传统密码即对称算法建立的数字签名。

8 ELGamal 签名方案

ELGamal签名方案[20]是非确定性的(ELGamal公钥密码体制也是非确定性的)，该方案的变型被美国国家标准技术研究所采纳为数字签名算法(DSA)。

然而最初的ELGamal签名方案不具有消息的自动恢复的特性[29]。1994 年，Nyberg和Reppel对ELGamal签名方案进行了改进，提出了六种具有消息自动恢复的签名方案，称为N-R数字签名方案。

§3-5 几种常用数字签名方案的比较及分析

1 MD5 签名体制与 DSS 签名体制的比较

MD5 签名体制中，MD5 算法是对 MD4 消息摘要算法的扩展，对任意长度的输入计算出一个 128 比特的散列值。而 DSS 签名体制中，其对应的散列算法有 SHA-1、SHA-256、SHA-384、SHA-512，这些算法分别产生 160、256、384、512 比特的散列值。数字签名为了实现防伪的作用，其原理就应用到了单向散列函数的不可逆性，即对于任意的输入不可能产生两个相同的散列值，从而起到指纹的作用。一旦发现产生相同散列输出的任意的两个输入，就认为该散列的生日攻击是成功的。所以，只有采取提高摘要的长度，从而提高生日攻击的难度。在 MD5 签名体制和 DSS 签名体制签名体制中，只有后者提供了扩展摘要的散列算法，结论不言而明。当然这并不意味着可以高枕无忧了，而是应该未雨绸缪,寻找新的杂凑函数和改进措施，提高安全性。

2 DSS 签名体制与 RSA 签名体制

DSS 签名体制中，它指定了一种数字签名算法（DSA，Digital Signature Algorithm）, 是美国政府的 Capstone 计划的一部分，该签名算法是 Schnorr 和 ElGamal 签名算法的变种。DSA 算法的安全性也依赖于整数有限域上的离散对数问题，安全强度和速度均低于 RSA 算法，其优点是不涉及专利问题。DSA 的一个重要特点是两个素数公开，这样，当使用别人的 P 和 Q 时，即使不知道私钥，也能确认它们是否是随机产生的，还是做了手脚，而 RSA 做不到。另外，对于 DSA，它是另一种公开密钥算法，但它不能用作加密，只能用作数字签名。

3 RSA 签名体制与 ECDSA 签名体制

RSA 签名体制的数学基础是基于数论中大素数分解的困难性。椭圆曲线数字签名算法(ECDSA)设计的数学原理是基于椭圆曲线离散对数问题的难解性。相对于 RSA 等密码体制而言，椭圆曲线加密体制是比较新的技术。椭圆曲线数字签名算法(ECDSA)和 RSA 与 DSA 的功能相同，并且数字签名的产生与认证速度要比 RSA 与 DSA 快。

于 1985 年，N.Kobliz[22]和Miller[30]提出将椭圆曲线用于密码算法，分别利用有限域上椭圆曲线的点构成的群实现了离散对数密码算法。椭圆曲线密码其根据是有限域上椭圆曲线上的点群中的指数级的难度。从目前已知的最好求解算法来看，160 比特的椭圆曲线密码算法的安全性相当于 1024 比特的RSA 算法。此后，有人在椭圆曲线密码算法实现了类似ELGamal的加密算法，以及可以恢复明文的数字签名方案。另外，人们也在探索在椭圆曲线上实现类似RSA算法。

总之，ECDSA 算法是蛮有前途的算法，不过在本课题中采用的还是用 RSA，因为 RSA 应用的已经非常广泛了，而且已经是免费的了，不再具有专利性质，把 RSA 研究好了，一样可以提供借鉴和参考。

第四章 数字签名的应用分析

目前数字签名已经广泛应用于商业、金融、军事等领域，尤其是在网络通信、电子邮件、电子商务、电子政务方面。

§4-1 电子邮件和数据交换

S/MIME 是安全电子邮件的一种规范。全称“安全的多功能互联网邮件扩展”协议(Secure/Multipurpose Internet Mail Extensions)[31]。它在原有的MIME邮件规范的基础上，新增了许多强有力的安全功能。安全电子邮件提供邮件的加密和签名，利用公钥算法保证用户的签名邮件不会被篡改，而加密邮件除了邮件接收者以外(甚至发送者)的任何人无法阅读其中的内容。

使用安全电子邮件之前要先申请二个安全证书，用户可以在证书颁发机构(CA)处申请证书，也可以到其他使用标准证书格式的颁发机构处申请。有的颁发机构会提供短期有效的免费证书供用户试用。若要经常使用，则要申请一个长期有效的证书，此时用户通常需要提交足够的身份证明和交纳一定的费用。

用户成功申请数字证书后，将会获得一对公私密钥。该密钥对存储于用户本地 PC 中。若要使用 AIMC 的安全邮件服务，用户还需要将该密钥对导入到 AIMC 中。AIMC 支持常用的 PKCS12 导入文件格式。导入的过程分两步：用户导出密钥对，并保存为 PKCS #12 格式的文件：用户将导出的密钥对文件通过浏览器导入到 WebMail 中。

电子邮件软件使用用户的私钥加密邮件，并在邮件中添加签名 PKCS#7 Signature，这个签名包括了用户的证书，它就会对邮件的签名进行验证。当邮件接收者使用的邮件工具支持 S/MIME，验证签名需要对证书链进行验证，这需要已经安装颁发给你的 S/MIME 证书的 CA 的证书，通常其它人不会有你的根证书，所以用户必须首先将根证书发给别人并安装，否则签名邮件将无法验证。根证书安装方法同 SSL 根证书的安装，通常 SSL 和 S/MIME 根证书会是同一个，因为 CA 一般是不分用途的，它只有一个用途就是签发证书。当接收者收到用户的签名邮件后，通过查看邮件属性可以看到用户证书(包含在签名中)，他可以把用户的证书添加到地址簿中去，并同地址簿中的邮件地址绑定。这样接收者就可以给用户发送加密邮件了，这是因为加密邮件使用的是邮件接收方的公钥即数字证书。加密邮件的前提是在用户的地址簿中有与发送地址绑定的数字证书。

另外，利用数字签名技术的邮件标准还有 PEM(Privacy-Enhanced Mail), PGP(PrettyGood Rrivacy)等。人们还可以利用数字签名的非否认特性保证电子数据交换(EDI)安全运行。而且数字签名已经成功

的运用于电子资金转账(EFT)等。

§4-2 数字签名在电子商务中的应用

SET(Secure Electronic Transaction)，安全电子交易[32]，是VISA和MasterCard两大信用卡公司和多家科技公司制定的一个在Internet上进行的在线交易的安全标准，于 1997 年推出。SET提供了消费者、商家和银行间的认证，确保了交易数据的安全、完整可靠和交易的不可否认性，特别是保证了消费者的隐私。SET已经成为目前最流行通用的安全电子商务标准，它的核心技术主要有公开密钥加密、数字签名、数字信封、数字证书等。图 4.1 是SET协议的描述。

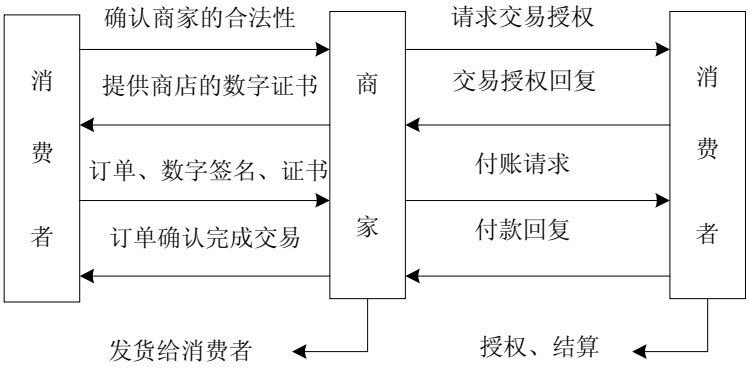

图 4.1 SET 协议流程

Fig.4.1.SET protocol flow

从协议中可以看出，SET 协议中应用到很多的数字签名技术，如数字签名、数字信封、双重签名(数字双签名)等等。

下面讨论一下在电子商务中应用的双重签名。

设 Alice, Bob, Carol 为通信实体，Alice 有一对用于数字签名算法的密钥对 (k_S, k_P)，$E(.)$ 和 $D(.)$ 分别为加/解密算法，$S(x) = E_{k_S}(H(x))$ 为 x 的签名，其中 $H(x)$ 为消息 x 的杂凑。

签名验证方程为 $H(x) = D_{k_P}(S(x))$，若方程成立 $S(x)$ 为 Alice 对 x 的有效数字签名。

m_1，m_2 为 Alice 产生的两个相互独立的消息，Alice 准备把 m_1 发送个 Bob，把 m_2 发送给 Carol。Alice 不希望 Bob 和 Carol 解读不想给他们看到的消息。

Alice 可以对 m_1，m_2 签名，得到 $D_{S_A}(m_1, m_2)$，Alice 在发送消息给 Bob 和 Carol 的同时也将

$D_{S_A}(m_1, m_2)$ 发送给他们。

Alice 产生数字双签名的步骤如下：

1）计算 $M_1 = H(m_1)$。

2）计算 $M_2 = H(m_2)$。

3）计算 $M_{12} = H(M_1 \| M_2)$，其中‖表示串接。

4）计算 $D_S = E_{k_S}(M_{12})$，则 D_S 为 Alice 对 m_1 ，m_2 的数字双签名，$D_S = D_{S_A}(m_1, m_2)$。

在发送消息时，Alice 把 m_1，$D_{S_A}(m_1, m_2)$ 及 $M_2 = H(m_2)$ 发送给 Bob，把 $m_2, D_{S_A}(m_1, m_2)$ 及 $M_1 = H(m_1)$ 发送给 Carol。

这样，Bob 只能得到消息 m_2 的杂凑，而得不到 m_2；同理，Carol 只能得到 m_1 的杂凑而不能得到 m_1。

Bob 和 Carol 收到相应消息后，均可以验证数字双签名。Bob 的验证步骤如下：

1）计算 $M'_{12} = D_{k_P}(D_{S_A}(m_1, m_2))$。

2）计算 $M_1 = H(m_1)$。

3）验证 $M'_{12} = H(M_1 \| M_2)$ 是否成立，若等式成立则表示数字双签名有效。

数字双签名的特性就是把发给两个不同的通信实体的两个不同消息联系在一起，两个通信实体都可以验证数字双签名。

在典型的B-C电子商务中[26]，有顾客、商家、支付网关（银行）等角色。顾客选定所需商品后向商家发出订购信息 OI（Order Information），并向银行发出支付信息 PI（Payment Information）。在这样的交易过程中，银行没有必要知道顾客订单的详情，或者顾客不想让银行知道。顾客可以把 OI 和 PI 分离而保护其隐私。然而 OI 和 PI 又必须以某种方式关联在一起，以便发生争议时方便公正第三方仲裁。顾客需要 OI 与 PI 之间的这种关联，以使顾客确信他是支付的这笔订单，而不是其它订单。

这种关联是必需的。假设顾客向商家发送两个消息——经过签名的 OI 和 PI 并由商家把 PI 转发给银行。如果商家能从顾客的另一笔中得到另一个 OI，商家就可以作弊，辩称这一 OI 与原 PI 配对，伤害顾客的利益。而引入这种关联就可以避免这样的清况发生，数字双签名就可以用来构造这种关联。

假设顾客的密钥对为 (k_{cs}, k_{cp})，银行的密钥对为 (k_{bs}, k_{bp})，顾客按以下步骤构造数字双签名：

1）计算 $H_{PI} = H(PI)$。

2）计算 $H_{OI} = H(OI)$。

3）计算 $H_{PO} = H(H_{PI} \| H_{OI})$。

4）计算 $D_S = E_{k_{cs}}(H_{PO})$，则 D_S 为顾客对 PI 、OI 的数字双签名。

顾客发给商家的消息构造如下：

1）随机生成一会话密钥k_s，用于单钥加密算法。

2）计算$M_b = E_{k_s}(PI \| D_S \| H_{OI})$。

3）计算$k = E_{k_{bp}}(k_s)$。

4）令$M = (M_b \| k \| H_{PI} \| OI \| D_s)$，则$M$为顾客发给商家的消息。

顾客把M发给商家，M中含有顾客的数字双签名，表示顾客愿意为这次交易付款。商家收到M后，执行以下步骤验证顾客的数字双签名：

1）计算$H_{OI} = H(OI)$。

2）计算$H_{PO} = H(H_{PI} \| H_{OI})$。

3）验证方程$H_{PO} = D_{k_{cp}}(D_S)$是否成立，若等式成立则表示数字双签名有效。

商家验证了顾客的数字双签名的有效性后，表明顾客已经承诺为OI所对应的该次交易付款，而商家对顾客是否具有支付能力还要通过银行验证。这时，商家把(M_b, k)发给银行。

银行收到(M_b, k)后，执行以下步骤：

1）利用银行的私钥k_{bs}，计算$k_s = D_{k_{bs}}(k)$。

2）利用在步骤1)得到的k_s再进一步计算$D_{k_s}(M_b)$，得到$(PI \| D_S \| H_{OI})$。

3）计算$H_{PI} = H(PI)$。

4）计算$H_{PO} = H(H_{PI} \| H_{OI})$。

5）验证方程$H_{PO} = D_{k_{cp}}$是否成立，若等式成立则表示数字双签名有效。

银行验证顾客的数字双签名有效后，表明顾客同意为H_{PI}所对应的交易付款，银行将检查顾客的账号是否具备支付该次交易的能力。若有，则从顾客的账号中扣除该次交易的款项并划到商家账号里，同时通知商家。商家在得到消息后就可以向顾客发货或提供顾客购买的服务。

从上述交易支付过程可以看到:顾客发给商家的消息M中[27]，M_b、k是通过商家转发给银行的，由于经过了加密处理，商家不能得到其中的PI；而银行得到的M_b、k中又不保护OI，因此银行不可

能知道OI。但由于利用了数字双签名将PI、OI关联在一起，使得商家、银行可以验证数字双签名，表明顾客愿意为交易付款。当发生争议时，可以由公正的第三方仲裁。因此保护了电子商务参与各方的合法权益。

§4-3 数字签名在远程控制中的应用

随着网络的日益普及，远程管理正日益成为真正意义上的远程管理。除了某些大的组织使用专网对其系统进行管理外，大部分的组织都可以通过 Internet 对自己的系统进行远程管理。由于 Internet 是一个开放的网络环境，因此，数据的安全将成为一个非常重要的问题。数据的安全包括三个方面：

机密性：非法用户不能读懂数据所表示的意义。

完整性：数据不被非法篡改。

数据源认证：数据来自正确的用户。

在远程管理[33]这个特定的应用环境中，数据的机密性将不成为主要的安全问题。主要的安全问题是数据的完整性和数据源认证(数据不可抵赖性)。远程管理需要保证最根本的两点：只有合法用户才能对系统进行管理；管理配置信息不被篡改；而使用数字签名就可以很好的解决这些问题。

具体到远程管理系统中，数字签名的过程为：在远程管理系统的客户端，先对管理信息进行摘要，再用这个管理员的私钥对管理信息的摘要进行签名(加密)。根据管理信息的敏感程度，可以选择是否对管理信息本身进行加密。最后把管理信息(明文或密文)和对管理信息的签名一起发送给远程管理系统的服务器端。服务器端在执行完管理指令之后也对管理结果进行签名，把管理指令执行结果和结果的签名发送给管理指令请求的管理员。

数字签名技术的应用，往往都是跟数字证书配合使用的。数字证书是由公认的第三方权威机构(CA)颁发的在网络通信中用以标识通信双方身份的数字信息的集合。除了基本的身份信息之外，证书还包含了证书持有者的公钥，用以对用与这个公钥相匹配的私钥加密的信息进行解密。因此，在远程管理系统的服务器端必须保存有管理员的证书。

数字签名还可以应用到电子政务、访问控制、软件验证、病毒检测等不同领域。也就是其应用的范围会很广。用一句话来说，随着信息时代的突飞猛进，电子签名的应用将越来越广泛，到时候可能会成为人们的第二身份证，甚至是全权代理。

第五章 保密散列数字签名算法

本章提出了一种基于 RSA 公钥密码体制、排列码对称加密算法和 MD5 单向散列函数的数字签名方案，实现了端对端用户的双向身份认证、数据加密和数字签名的功能。本章的最后给出了关于保密散列数字签名算法的安全性评价。

§5-1 排列码加密算法

5-1-1 传统分组密码机制

根据密钥的特点，密码体制分为对称和非对称密码体制两种。对称密码体制又称单钥或传统密码体制，非对称密码体制又称双钥或公钥密码体制。在公钥密码体制中，加密密钥和解密密钥不同，从一个难于推出另一个，可将加密能力和解密能力分开。公钥密码体制因为加密速度较慢，不能用于对长文件进行加密，通常用来进行密钥传递或数字签名。而对文件的加密要由对称加密算法来实现。按加密方式又可将对称密码体制分为流密码和分组密码两种。在流密码中，将明文消息按字符逐位地加密。在分组密码中，将明文消息分组（每组含有多个字符），逐组地进行加密。

从分组密码的发展现状来看，目前分组密码的观念是 $A=A'$， $N=N'$， $L=L'$，则映射 f 可构造成一对一的映射用于加密变换，则逆映射 f^{-1} 可构造成一对一的逆映射用于解密变换。正是这一观念，飞速发展的计算机技术使得 DES 仅仅 21 年就走过了它的历史的辉煌。因为对于 n， f 或 f^{-1} 可构造成一对一的映射或逆映射充其量只不过是 2^n。可以预言，如果抱着这个观念不放，AES 的寿命也不会太长，因为它仅仅是加大了一倍的 n。面对人类的进步、计算技术的高速发展，必须创立新的加密思路，使得破密难度远远大于 2^n。

5-1-2 创新分组密码机制

如果在同一加密解密过程中，明文（0，1，1，1），（1，0，1，1），（1，1，0，1），（1，1，1，0）都可映射成密文（0，1，1，1），而密文（0，1，1，1），（1，0，1，1），（1，1，0，1），（1，1，1，0）都可映射成明文（0，1，1，1）。这样就形成了多对多的分组密码的新概念，这在旧观念中是完全不可能实现这样的加密解密。在多对多的情况下，如果加密过程中记住的不仅仅是加密的结果，而且加密的密钥中还包含着由明文到密文的路径，解密时再顺原路变换回去，多对多的分组密码的新概念就可以利

用了，对于 $A=A'$， $N=N'$， $L=L'$，由明文到密文的路径远远大于 2^n，而明文到密文的路径和密文到明文的路径都是一对一的。由这个新概念就产生了排列码加密解密方法。

5-1-3 排列码加密解密方法

如图 5.1 所示，A有 n 个比特,它的全排列共有 $n!$ 种不同的排列，重新排列后做密文，由明文到密文的路径就有 $n!$ 种。每个整数对应一种排列，当这个整数做密钥时，密钥的个数是 $n!$ 个。下一组 n 比特，可以加一个任意整数，再取 $n!$ 的模，将结果做这 n 比特的加密密钥，以此类推，每组 n 比特都这样处理。因为可加的整数共有 $n!$ 个，所以这样进行加密的加密强度是 $(n!)^2$ 。若每 $n*n$ 比特再作为一个分组施加此方法，那它的加密强度就是 $(n!)^4$ 。因为每一个比特可以交换到 n 个不同的方向去，进而 n 个比特可以有 $n*n$ 个彼此不同的方向交换到 n 个彼此不同的地方去，而每一去向都可以加一个非门，也可以不加一个非门，因此加非门的方法共有 2^{n*n}，所以总的由明文到密文的路径有 $(n!)^4*2^{n*n}$ $(n!)^4$种。

A有 n 个比特，它的全排列共有 $n!$ 种不同的排列，我们称为有 $n!$ 个排列码，对这 $n!$ 个排列码进行全排列，共有 $(n!)!$ 个排列码编码方案，每个排列码编码方案都对应着一个算法[26]，因此共有 $(n!)!$ 个算法，如图 5.1 所示。

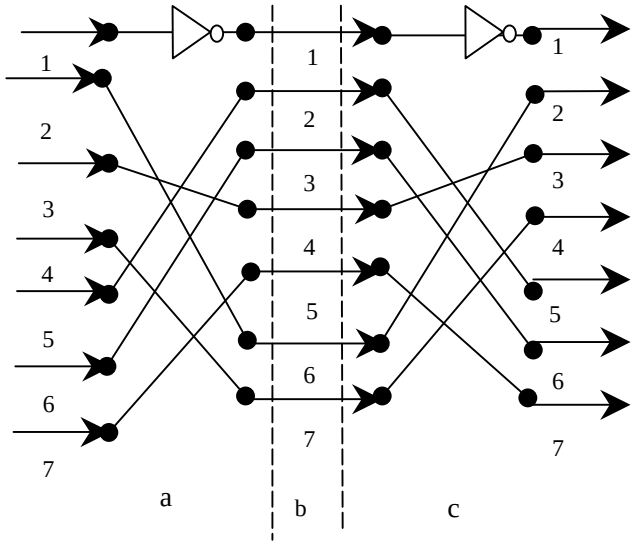

图 5.1 排列码加密解密原理

Fig.5.1 Principle of encrypt and decrypt by permutation code

明文寄存器中的明文的每一比特按排列码算法的计算结果重新排列并经过一次加非运算送入密文寄存器，就完成一个分组的加密，因此如果用硬件实现，明文信号最多只须经过一级与门、一级或门、一级非门就可以到达密文寄存器完成加密过程。

类似地，密文寄存器中的密文的每一比特按排列码算法的计算结果重新排列并经过一次加非运算送

入明文寄存器，就完成一个分组的解密，因此如果用硬件实现，密文信号最多只须经过一级与门、一级或门、一级非门就可以到达明文寄存器完成解密过程。

排列码加密解密过程都非常简单，仅是能计算排列码就可完成；硬件实现速度非常快，仅一级与门、一级或门、一级非门的时间就可以完成；加密强度非常高，当$n=128$时，它的加密强度是 DES 的10^{5781}倍；密钥处理非常灵活，密钥可以是数字串，还可以是任意的文字串，可定长、可变长，如一个成语，一段诗词，某本书中的一段话；可使用的算法非常多，可同一集团使用相同算法不同集团使用不同算法，因此即使算法全部公开，攻击者确定究竟使用者使用哪个算法也是一个难题。

5-1-4 排列码加密解密方法的密钥方案

第一个分组把可用任何方法存入到密钥寄存器中的若干字节信息当作整数处理，就可以计算出 *n*! 中的一个排列，再用若干字节来确定加非门的位置，以后的分组可用这个整数加一个任意整数，假如所得整数大于 *n*!，再取 *n*! 的模再加 1 做密钥，或加一个伪随机数做密钥，加一个伪随机数来改变加非门的位置。

§5-2 保密散列数字签名

数字签名实现的基础是加密技术，可以由对称密钥算法实现，也可以由非对称密钥（公开密钥）算法实现。对称密钥算法的速度较快，但涉及到密钥的交换问题以及仲裁人，安全性和保密性较差。公开密钥算法更加灵活多样，但是速度不如对称密钥算法快，不适用于长文件的数字签名[34]。实际应用当中可以结合两者的优点，扬长避短，形成一个安全性和签名速度都令人满意的系统。

5-2-1 数字签名常用的方法

1）对称密钥密码算法进行数字签名

对称密钥密码算法所用的加密密钥和解密密钥通常是相同的，或者可以很容易地由其中的任意一个导出另一个。对称密钥算法实现数字签名必须有仲裁人参与。用户A和仲裁人T共享密钥K_A，用户B和T共享另一个不同的密钥K_B。

⑴ A用K_A加密他准备发送给B的明文消息M，并把加密消息C_A传送给T：$C_A=E_{K_A}(M)$。

⑵ T用K_A解密C_A得到明文M：$M=D_{K_A}(C_A)$。

⑶ T把解密消息和他收到A消息的声明t，一起用K_B加密成C_B：$C_B=E_{K_B}(M,t)$。

⑷ T把加密的消息包C_B连同A用K_A加密的C_A一起传给B：$(C_B,K_A)\rightarrow B$。

(5) B用K_B解密消息包C_B之后，就可以读到A的消息M 和T 的签名证书t，证明消息来自A：$(M,t)=D_{K_B}(C_B)$。

B需要保留C_A以备发生分歧时裁决之用。这种签名方式要求仲裁人必须高度的完善和安全，而且得到所有人的信任。

2) 公开密钥数字签名算法

公开密钥系统用作数字签名的协议很简单[27]：

(1) A用他的私人密钥对文件加密，从而对文件签名。

(2) A将签名文件法送给B。

(3) B用A的公开密钥解密文件，从而验证签名。

公开密钥数字签名的算法很多，应用最为广泛的两种是：DSA 签名、RSA 签名。

早在 1991 年 8 月 30 日，美国国家标准与技术学会（NIST）就在联邦注册书上发表了一个通知，提出了一个联邦数字签名标准[35]，NIST称之为数字签名标准（Digital Signature Standard, DSS）。该标准为计算和核实数字签名指定了一个公开密钥数字签名算法（Digital Signature Algorithm, DSA）。

RSA 是最流行的一种加密标准，与 DSS 不同，RSA 既可以用来加密数据，也可以用于身份认证。和对称密钥算法实现的数字签名相比，在公钥系统中，由于生成签名的密钥只存储于用户的计算机中，安全系数大一些。

在实际的实现过程中，采用公开密钥密码算法对长文件签名效率太低。为了节省时间，数字签名协议经常和单向散列函数一起使用。该编码方法采用单向散列函数将需加密的明文“摘要”成一串128bit的密文，这一串密文亦称为数字指纹[38]（FingerPrint），它有固定的长度，并且不同的明文摘要必定不一致。这样这串摘要就可成为验证明文是否是“真身”的“指纹”了。这样，只对文件的摘要进行签名就可以了，不必对整个文件签名。采用这种方法使得计算速度大大提高，单向散列函数确保不同文件的散列值不相同，因此，对文件摘要签名和对文件签名一样安全。

这种方法还可以将文件和签名分开保存，接收者对文件和签名的存储量要求降低了。

5-2-2 当前数字签名的不足

对称密钥密码算法实现的数字签名，需有仲裁人参与，这是非常耗时的，容易对整个系统造成瓶颈。仲裁人的数据库中存有每一对签名人的密钥，这对计算机的存储量提出了较高的要求，而且一旦被攻破会造成很大的损失。文件的内容也不能对仲裁人保密。

公开密钥算法实现的数字签名可以解决这个问题，即签名人A用私有密钥加密文件实现签名，验证人用B用A的公开密钥解密，如果能够解密，就验证了签名，因为别人不知道A的私有密钥。但这里有另一个问题，任何人都可以用A的公开密钥解密文件看其内容。这样满足不了用户对文件机密性的要求。

公开密钥系统数字签名满足签名的不可重用，因为签名是文件的函数并且不可能转换成另外的文件，但是，可能被B把签名和文件一起重用。比如，A交给B一张签名数字支票，B可以复制该支票，在不同的时间或到不同的银行验证支票并兑现支票。

公开密钥系统对长文件的加密速度比对称密钥系统要慢近 1000 倍，效率太低。

5-2-3 保密散列数字签名

针对以上所述，可以如此解决各种签名方法的不足。

对公开密钥算法实现的数字签名而言，为避免任何人都可以用A的公开密钥解密文件看其内容，A可以进一步用B的公开密钥加密其签名文件，B收到之后，先用自己的私人密钥解密，然后用A的公开密钥验证签名。这样就实现了把数字签名的真实性和加密的安全性结合起来，就好像写信，签名说明了谁是写信人，信封提供了秘密性。

为防止B复制A签名的数字支票，在不同的时间或到不同的银行验证支票并兑现支票，可以加入时间标记可以防止类似的金融诈骗。即把日期和时间的签名附在消息中，并跟消息中其他的部分一起签名。银行可以将所兑现支票的时间标记存储在数据库中，兑现支票时检查时间标记是否和数据库中的一样。如果一样就证明该支票已经被兑现。

为提高效率，公开密钥数字签名还应该与单向散列函数一起使用。

总的来说，目前的数字签名算法都是以对称密钥密码体制或公开密钥密码体制来实现的，这就必然会有各种缺陷。我们全面综合各类数字签名方法的优点，把对称密钥算法、公开密钥算法及单向散列函数结合在一起，避免了单独使用一种算法的不足，提出以下的保密散列数字签名算法：

A和B事先协商他们采用的单向散列函数H，散列函数不必保密。

1) A产生明文M的单向散列值H_A（或称消息摘要）。

2) A选择密钥k，利用对称密钥加密方法对明文M加密成密文 C：$C = E_k(M)$。

3) A利用公开密钥算法以私人密钥k_{AS}对散列H_A、时间标记t、密钥k加密产生C_H，从而表示对文件签名：$C_H = E_{k_{AS}}(H_A, t, k)$。

4) A将签名C_H用B的公开密钥 k_{BP}加密成C_H'：$C_H' = E_{K_{BP}}(C_H)$。

5) A将密文C和签名C_H'一起传送给B。

6) B用自己的私人密钥解密签名C_H'，得到C_H。

7) B用A的公开密钥解密C_H得到文件摘要H_A、 时间标记t和密钥k。

8) B用k解密C得到文件原文M。

9) B用明文M产生单向散列值H_B，对比自己计算产生的H_B和从C_H解密得到的H_A，如果匹

配证明签名是有效的。

这种签名方式用对称密钥加密方法加密文件原文，保证了文件的秘密性，同时弥补了公开密钥算法加密长文件效率较低的不足。利用单向散列函数产生文件摘要进行公开加密数字签名，同时将加密文件的密钥 k 一起签名，避免密钥 k 的泄漏。这样做不需要仲裁人参与就解决了对称密钥算法的密钥传递问题，更加高效而且保密。进一步用接收方公开密钥加密又可以阻止其他人偷看文件原文及验证签名。

以上的保密散列数字签名算法能够满足所要求的所有条件：

1）可信的。只有 A 才能用他的私人密钥对文件签名，否则 B 无法用 A 的公开密钥解密。

2）不可伪造的。只要 A 没有泄漏私人密钥，就不可能有人假借 A 的名义签名。

3）签名不能重复使用。如果 B 把签名附到另一个文件上，A 或仲裁人就要求他出示 A 用私人密钥签名过的文件，B 显然没有以 A 的私人密钥对伪造文件进行的签名。

4）文件不能被改变。用上述方法同样可以验证消息 M 是否被改动。

5）A 不能抵赖消息 M 。A 对文件摘要的签名密文可以证明 A 发送过消息 M 。

这个数字签名算法不但速度快，而且可以将文件与签名分开传送和保存，保密性强，减少存储空间，应用前景十分广阔。

第六章 保密散列数字签名系统实现

本章基于 RSA 公钥密码体制、排列码对称加密算法和 MD5 单向散列函数的数字签名方案，实现了数据加密和数字签名功能的保密散列数字签名系统。本章将详细描述系统的安全措施和运行流程。

§6-1 系统设计思想

随着信息科技的发展，数据的安全越来越重要。如何保护存储数据的安全也显得很重要，为了保护数据的安全，本系统采用排列码加密算法实现数据的加密，该算法目前没有安全方面的问题，至少在公开的密码分析中，尚未发现有成功的破译该算法的。同时随着电子文档的普遍应用，如何确认某一文档是某人认同的，或者该文档没有被改变，这些也显得很重要，在生活中，人们通过对纸上进行签名来实现，而在电子文档中，图形签名很容易被复制，而且无法区别复制与被复制图形签名，所以电子文档的签名不能使用图形签名来实现，解决的方法是数字签名，在前面我们已经讨论过了数字签名。在本系统的数字签名实现中，采用 MD5 算法，公钥密码算法 RSA 和排列码对称加密算法来实现。

§6-2 系统需求

全球网络的出现已经开始对个人交往、企业运作、政府对公民提供服务的方式产生巨大的影响。随着网上经济的蓬勃发展，在互联网上实现电子签约将是一个必然的趋势。网上签约系统的启动，将免去处于不同地域、国家的合作者往返奔波的舟车劳顿之苦，节省经费和时间，提高工作效率，大大方便世界范围内经济及科技合作与交流。合作者甚至不用见面便可完成签约的手续[40]。

电子签约的核心是必需保证签约双方不能改动合约内容、如果一方合约被改动，另一方可以通过鉴别程序予以揭穿。数字签名是实现网上电子签约的一项最重要的技术，它所带来的对传统签名方式的变革将会为人们从事电子商务活动提供方便、快捷、安全的手段。

§6-3 系统实现

经过整体考虑和现实要求，本系统提供以下功能（如图 6.1 所示），数据加密/解密、密钥管理、签

名和数据库管理。下面具体介绍一下这些功能以及实现时需要的数学方面的理论知识。

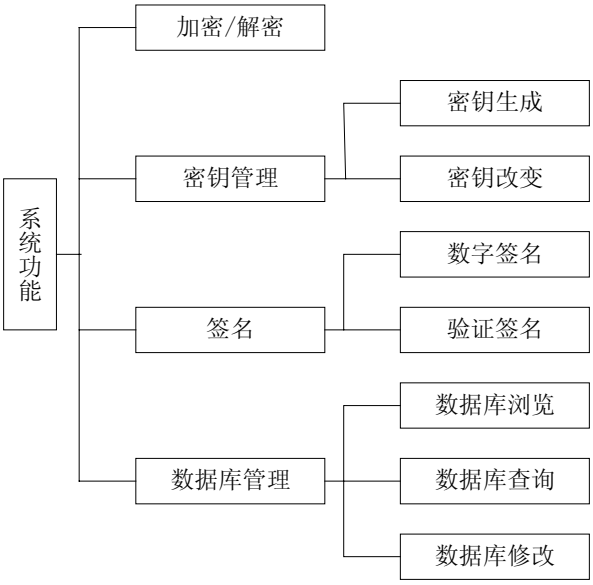

图 6.1 系统功能

Fig.6.1 System function

6-3-1 密钥管理功能

本系统密钥管理包括:密钥生成、密钥改变和密钥查看。

在本系统中，我们使用的是 RSA 公开密钥加密算法。该体制是密码学中第一个较完善的公钥密码体制，既可用于加密也能用于数字签名。RSA 公钥密码体制的基础在于大整数因子分解的困难性。

RSA公钥密码体制[17]的原理可简述于下：

(1) 随机选取两个大素数 p 、 q 。

(2) 计算 p 与 q 的乘积： $n = p * q$ 。

(3) 再由 p 与 q 计算： $z=(p-1)(q-1)$ 。

(4) 随机选取加密密钥 e ，使 e 和 z 互为素数。

(5) 然后用欧几里德扩展算法计算解密密钥 d ，满足 $e * d = 1 \bmod(z)$ 。

(6) 由此得到两组数 (n,e) 和 (n,d) ，分别被称为公开密钥和私人密钥。公开参数 e 和 n ，而 d 则应保密，两个素数 p 、 q 及 z 不再需要，应及时销毁不可泄露。

(7) 利用 RSA 加密第一步需将明文数字化，并取长度小于 $\log_2 n$ 位的数字作明文块，设 m 为明文，c 为密文，则下列两式成立：

加密算法： $c = E(m) = m^e \bmod n$ 。

解密算法：$D(c)=c^d \bmod n$。

RSA 的安全性基于大数分解的难度。其公开密钥和私人密钥是一对大素数(100 到 200 个十进制数或更大)的函数。从一个公开密钥和密文中恢复出明文的难度等价于分解两个大素数之积。素数的产生是 RSA 算法中的难点，也是关键之处。一般而言，判定一个数是素数比较困难，但判定一个数不是素数则要容易的多。对尺寸为n的数，一个随机数是素数的概率接近于$1/\ln(n)$，那么小于n的素数的总数为$n/\ln(n)$。由此可知，在长度为 512 位或略短一些的数中，有超过10^{151}个素数。因此在生成密钥时先产生两个大的随机数然后再测试其是否为素数是可行的。密钥的生成过程如下:

1）产生两个大的随机数p和q

RSA 算法对随机数的随机性要求很高，普通的编程语言中库函数产生的随机数毫无随机性可言，因 RSA 算法对随机数的随机性要求很高，普通的编程语言中库函数产生的随机数毫无随机性可言，因为每次运行程序时所得结果相同，不满足密钥的不可重复性要求。因此，在程序实现过程中应加入不确定因素，使之重复调用时不会产生相同的随机数。

本系统中把系统时间作为不确定因素，确保在一定范围内不会产生相同的随机数。因为系统时间随时在变化，所以每一次运行程序产生的随机数也不相同。

2）测试p和q是否是素数

日前有多种简单而且快速的素数判定定理，其中概率测试法是 RSA 算法中产生素数最实用的方法。常用的概率测试法有以下几种:

(1) Lehmann(勒曼)测试法

该测试方法是由Lehmann独自研究得出的[37]。测试算法如下:

① 选择一个小于p的随机数a。

② 计算$a^{(p-1)/2} \bmod p$。

③ 如果$a^{(p-1)/2} \bmod p \neq 1$或$-1$，那么$p$肯定不是素数。

④ 如果$a^{(p-1)/2} \bmod p \equiv 1$或$-1$，那么$p$不是素数的可能性至多是 50%。

⑤ 同样，如果p是合数，随机数a是p的证据的概率不小于 50%。对a选择t个不同的随机值，重复t次这种测试。如果计算结果等于1或-1，并不恒等于1，那么p可能是素数所冒的错误风险不超过$1/2^t$。当我们把t取成较大的数值时，基本上就能保证产生的是素数了。

Lehmann(勒曼)测试法流程图，如图 6.2 所示。

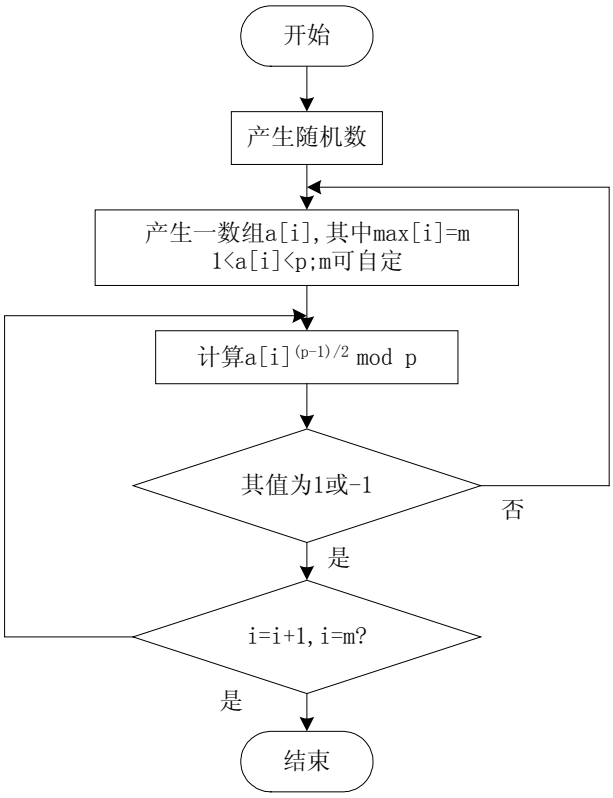

图 6.2 Lehmann(勒曼)测试素数法

Fig.6.2 Lehmann's tests for primality

(2) Rabin-Miller 测试方法

这是相对来说很容易使用的简单算法，它基于Gray Miller[38]的部分想法，由Michael Rabin发展。事实上，这是在NIST(国家标准和技术研究所)的DSS建议中推荐的算法的一个简化版[39]。

首先选择一个待测的随机数 p，计算 b，b 是2整除 $p-1$ 的次数(即 2^b 是能整除 $p-1$ 的2的最大幂数)。然后计算 m，使得 $n=1+2^b m$。

① 选择一个小于 p 的随机数 a。

② 设 $j=0$ 且 $z=a^m \bmod p$。

③ 如果 $z=1$ 或 $z=p-1$，那么 p 通过测试，可能是素数。

④ 如果 $j>0$ 且 $z=1$，那么 p 不是素数。

⑤ 设 $j=j+1$。如果 $j<b$ 且 $z\neq p-1$，设 $z=z^2 \bmod p$，然后回到第④步。如果 $z=p-1$，那么 p 通过测试，可能是素数。

⑥ 如果 $j=b$ 且 $z\neq p-1$，那么 p 不是素数。

(3) 在系统中，这样产生一个素数，速度是很快的。

① 产生一个 n 位的随机数 p，设高位位和低位位为 1(设高位位为 1 是为了确保该素数达到要求的长度，设低位位为 1 是为了确保该素数为奇数)。

② 检查以确保 p 不能被任何小素数整除：如 2，3，5，7 等。许多算法测试 p 对小于 256 的所有素数的整除性。最有效的测试整除性的方法是整除所有小于 2000 的素数，可排除不是素数的可能性为：$1-(1-1/2)(1-1/3)(1-1/5)...(1-1/1999)\approx 0.853$。

③ 对某随机数 a 进行素数测试。若 p 通过测试，则另外产生一个随机数 a，再重新进行测试。选取较小的 a 值，以保证较快的计算速度。可以作四五次这样的测试。

④ 如果 p 在其中的一次测试中失败，则重新产生一个 p，再进行测试。当然，我们可以不用每次都产生一个随机数 p，而是按递增的方式搜索以某随机数为起点的所有数，直到找到一个素数。素数产生的流程，如图 6.3 所示。

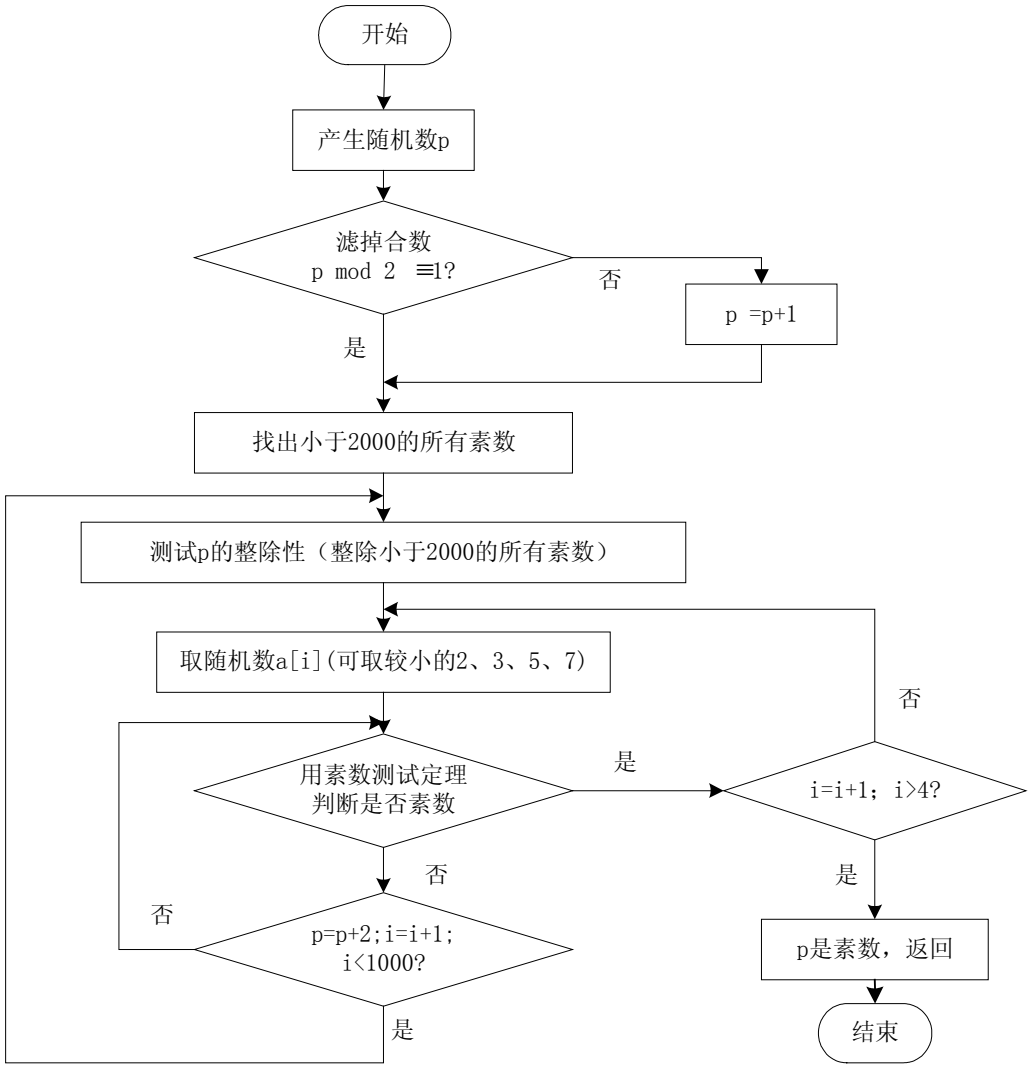

图 6.3 素数产生流程图

Fig.6.3 Flow chart for prime number

3）密钥的产生

首先选取两个大素数 p 和 q（使 $p>q$），为了获得最大程度的安全性，要求两个素数的长度相同。然后计算 p 和 q 的乘积 $n=pq$ 和 $z=(p-1)(q-1)$，再随机选取加密密钥 e，使 e 和 z 互素。就得到公开密钥 (e,n)。最后用欧几里德扩展算法计算解密密钥 d，以满足 $ed\equiv 1 \bmod (p-1)(q-1)$。注意 d 和 n 也互素，d 是私人密钥。两个素数 p 和 q 不再需要，可以舍弃，但决不能泄密。

目前需要解决的问题是:如何找到与 z 互素的数 e。也就是说如何使 z，e 满足它们的最大公因数 $\gcd(z,e)=1$。

我们可以先选择一个数 n_1，然后判断 $\gcd(z,n_1)$ 是否等于 1，如果不等于 1，则改变 n_1 直到得到一个满足条件 $\gcd(z,e)=1$ 的数。而欧几里德算法可以迅速找出给定的两个整数 n_1，n_2 的最大公因数 $\gcd(n_1,n_2)$，因此我们可以通过该算法判断两个数是否互素。

欧几里德[40]求最大公因数 $\gcd(n_1,n_2)$ 算法描述如下:

（1）求 q 及 r，使其满足 $n_1=q*n_2+r$。

（2）若 $r=0$，则 $g\leftarrow n_2$，输出 g（g 即为 n_1，n_2 的最大公因数），结束。否则转下一步。

（3）$n_1\leftarrow n_2$，$n_2\leftarrow r$，转到(1)。

由此可见，整个密钥的产生过程，如图 6.4 所示。

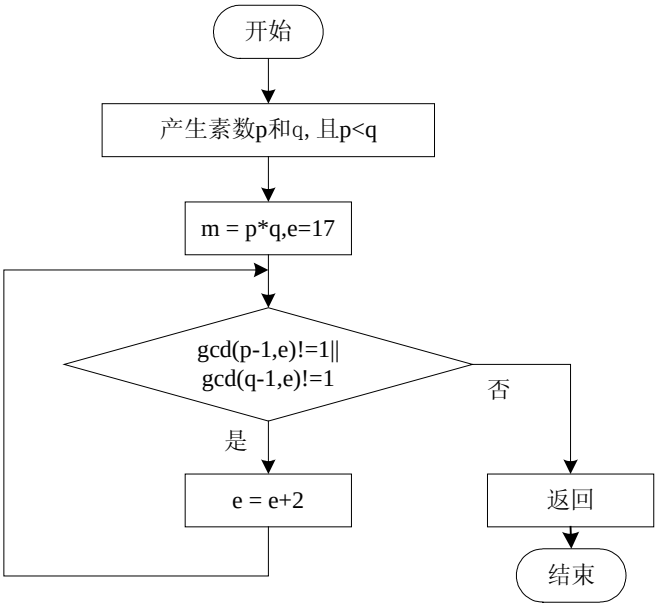

图 6.4 密钥产生过程

Fig.6.4 Process of generate key

在系统中用户可以选择2**512、2**768或2**1024三种密钥长度，密钥长度越长，加密强度越高，但加密所需时间也越长。选择密钥长度后，点击“开始”按钮，生成新密钥，如图6.5所示。新密钥生

成后，在系统运行的目录下将产生密钥文件rsa.key和对应该密钥的公开密钥文件rsa_pub.key两个文件。如果用户不想使用当前系统的设置（公钥/私钥），可以重新生成密钥，还可以查看公钥/私钥，如图6.6所示。

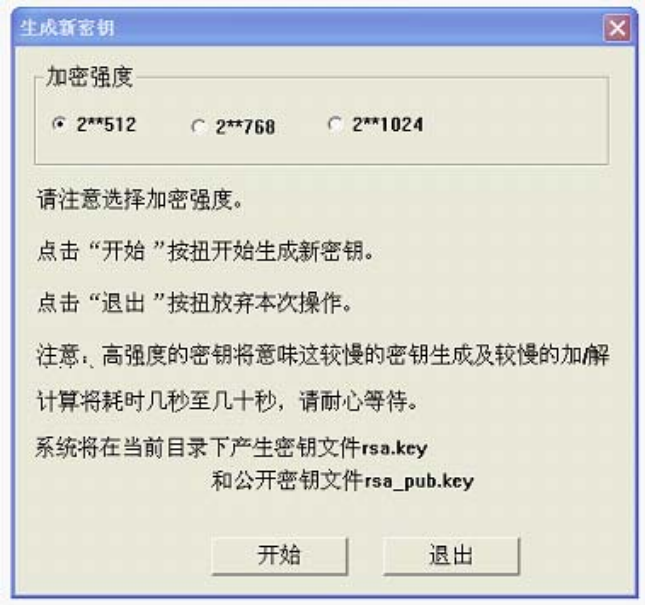

图 6.5 产生新密钥

Fig.6.5 Generate new key

图 6.6 查看密钥

Fig.6.6 Examine key

6-3-2 数字签名体制的实现

1）系统设置

甲乙双方进行通信之前，双方都应生成各自的公钥和私钥(rsa.key/rsa_pub.key(甲)和rsa.key/rsa_pub.key(乙))。然后双方交换各自的公钥。对于发送方甲来说，需要：

(1) 导入自己的私钥rsa.key(甲)，如图6.7所示。

图 6.7 导入私钥

Fig.6.7 Induct private key

(2) 添加通信用户乙的数据信息，并导入乙的公钥rsa_pub.key，如图6.8所示。

图 6.8 添加通信用户的数据信息

Fig.6.8 Increase correspondence user's data message

通信用户乙的操作与甲类似，导入自己的私钥，并添加甲的数据信息，导入甲的公钥。

2）产生消息摘要

对要传送的明文用 MD5 进行处理,获得消息摘要。

MD5 是Ron Rivest设计的单向散列函数，MD表示消息摘要（Message Digest），对输入消息，算法产生 128 位散列值（或称消息摘要）[25]。

MD5 以 512 位分组来处理输入文本，每一分组又划分为 16 个 32 位子分组，算法的输出有四个 32 位分组组成，将它们级联形成一个 128 位散列值。

算法实现步骤：

(1) 填充消息，使其长度恰好为一个比 512 的整数倍小 64 位的数。

填充方法是在消息后面附一个 1 和多个 0，然后在其后附上 64 位的填充之前原消息的长度。这样，消息的长度恰好是 512 位的整数倍，同时确保不同的消息在填充后不同。

四个 32 位变量初始化为：

A=0x01234567；B=0x89abcdef；C=0xfedcba98；D=0x76543210。

它们称为链接变量。

(2) 进行算法的主循环。循环次数为消息中 512 位消息分组的数目。

将上面四个变量复制到另外的变量中：A 到 a，B 到 b，C 到 c，D 到 d。

主循环有四轮。每一轮进行 16 次操作。每次操作对 a，b，c 和 d 中的其中三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，文本的一个子分组和一个常数。再将所得结果向右环移一个不定的数，并加上 a，b，c 或 d 中之一。最后用该结果取代 a，b，c 或 d 中之一，如图 6.9 所示。

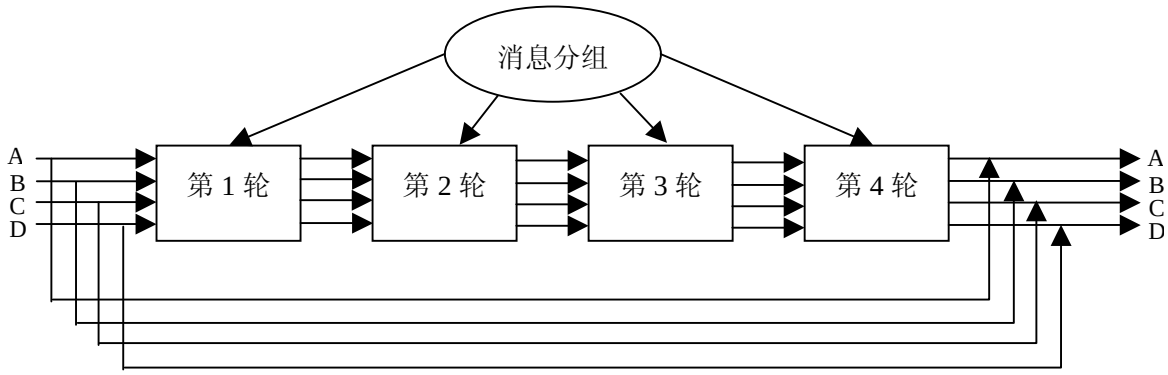

图 6.9 MD5 主循环

Fig.6.9 Main loop of MD5

以下是每次操作中用到的四个非线性函数（每轮一个）。

$$F(X, Y, Z) = (X \wedge Y) \vee ((\neg X) \wedge Z)$$

$$G(X, Y, Z) = (X \wedge Z) \vee (Y \wedge (\neg Z))$$

$$H(X, Y, Z) = X \oplus Y \oplus Z$$

$$I(X, Y, Z) = Y \oplus (X \vee (\neg Z))$$

（注：$\oplus$是异或，$\wedge$是与，$\vee$是或，$\neg$是反）

设M_j表示消息的第j个子分组（从 0 到 15），<<<s表示循环左移s位，则四种操作为：

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

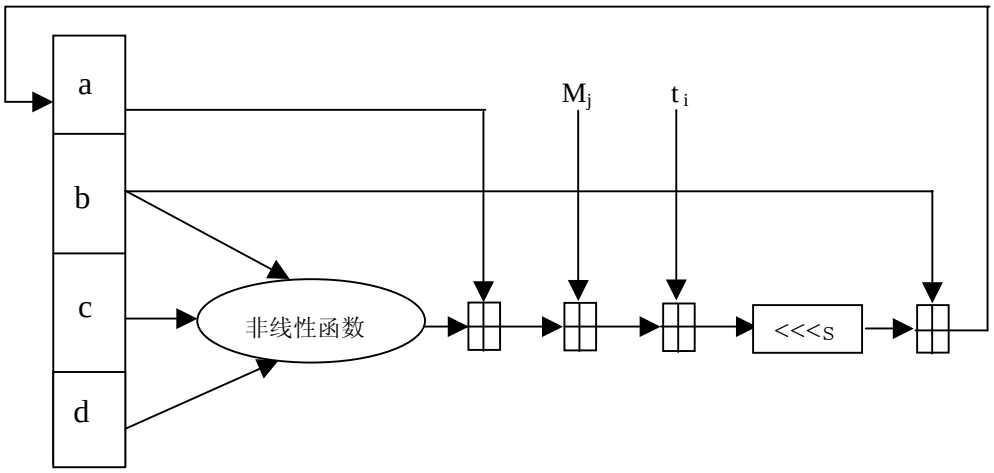

图 6.10 MD5 的一个执行过程

Fig.6.10 Process of MD5

这四轮（64 步）是：

第一轮

FF(a, b, c, d, M_0, 7, 0xd76aa478)

FF(d, a, b, c, M_1, 12, 0xe8c7b756)

FF(c, d, a, b, M_2, 17, 0x242070db)

FF(b, c, d, a, M_3, 22, 0xc1bdceee)

FF(a, b, c, d, M_4, 7, 0xf57c0faf)

FF(d, a, b, c, M_5, 12, 0x4787c62a)

FF(c, d, a, b, M_6, 17, 0xa8304613)

FF(b, c, d, a, M_7, 22, 0xfd469501)

FF(a, b, c, d, M_8, 7, 0x698098d8)

FF(d, a, b, c, M_9, 12, 0x8b44f7af)

FF(c, d, a, b, M_{10}, 17, 0xffff5bb1)

FF(b, c, d, a, M_{11}, 22,0x895cd7be)

FF(a, b, c, d, M_{12}, 7, 0x6b901122)

FF(d, a, b, c, M_{13}, 12, 0xfd987193)

FF(c, d, a, b, M_{14}, 17, 0xa679438e)

FF（b, c, d, a, M_{15}, 22, 0x49b40821）

第二轮

GG（a, b, c, d, M_1, 5, 0xf61e2562）

GG（d, a, b, c, M_6, 9, 0xc040b340）

GG（c, d, a, b, M_{11}, 14, 0x265e5a51）

GG（b, c, d, a, M_0, 20, 0xe9b6c7aa）

GG（a, b, c, d, M_5, 5, 0xd62f105d）

GG（d, a, b, c, M_{10}, 9, 0x02441453）

GG（c, d, a, b, M_{15}, 14, 0xd8a1e681）

GG（b, c, d, a, M_4, 20, 0xe7d3fbc8）

GG（a, b, c, d, M_9, 5, 0x21e1cde6）

GG（d, a, b, c, M_{14}, 9, 0xc33707d6）

GG（c, d, a, b, M_3, 14, 0xf4d50d87）

GG（b, c, d, a, M_8, 20, 0x455a14ed）

GG（a, b, c, d, M_{13}, 5, 0xa9e3e905）

GG（d, a, b, c, M_2, 9, 0xfcefa3f8）

GG（c, d, a, b, M_7, 14, 0x676f02d9）

GG（b, c, d, a, M_{12}, 20, 0x8d2a4c8a）

第三轮

HH（a, b, c, d, M_5, 4, 0xfffa3942）

HH（d, a, b, c, M_8, 11, 0x8771f681）

HH（c, d, a, b, M_{11}, 16, 0x6d9d6122）

HH（b, c, d, a, M_{14}, 23, 0xfde5380c）

HH（a, b, c, d, M_1, 4, 0xa4beea44）

HH（d, a, b, c, M_4, 11, 0x4bdecfa9）

HH（c, d, a, b, M_7, 16, 0xf6bb4b60）

HH（b, c, d, a, M_{10}, 23, 0xbebfbc70）

HH（a, b, c, d, M_{13}, 4, 0x289b7ec6）

HH（d, a, b, c, M_0, 11, 0xeaa127fa）

HH（c, d, a, b, M_3, 16, 0xd4ef3085）

HH（b, c, d, a, M_6, 23, 0x04881d05）

HH（a, b, c, d, M_9, 4, 0xd9d4d039）

HH（d,a,b,c,M_{12},11,0xe6db99e5）

HH（c, d, a, b, M_{15}, 16, 0x1fa27cf8）

HH（b, c, d, a, M_2, 23, 0xc4ac5665）

第四轮

II（a, b, c, d, M_0, 6, 0xf4292244）

II（d, a, b, c, M_7, 10, 0x432aff97）

II（c, d, a, b, M_{14}, 15, 0xab9423a7）

II(b, c, d, a, M_5, 21, 0xfc93a039)

II(a, b, c, d, M_{12}, 6, 0x655b59c3)

II(d, a, b, c, M_3, 10, 0x8f0ccc92)

II(c, d, a, b, M_{10}, 15, 0xffeff47d)

II(b, c, d, a, M_1, 21, 0x85845dd1)

II(a, b, c, d, M_8, 6, 0x6fa87e4f)

II(d, a, b, c, M_{15}, 10, 0xfe2ce6e0)

II(c, d, a, b, M_6, 15, 0xa3014314)

II(b, c, d, a, M_{13}, 21, 0x4e0811a1)

II(a, b, c, d, M_4, 6, 0xf7537e82)

II(d, a, b, c, M_{11}, 10,0xbd3af235)

II(c, d, a, b, M_2, 15, 0x2ad7d2bb)

II(b, c, d, a, M_9, 21, 0xeb86d391)

常数t_i可以如下选择：

在第i步中，t_i是 2^{32}*abs(sin(i))的整数部分，i的单位是弧度。

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 和 D 的级联。

3）数字签名的生成

(1) 发送方对要传送的明文用MD5进行处理,获得消息摘要。

(2) 发送方用自己的私用密钥对摘要加密，形成了数字签名，并附在要发送的信息原文后面。

(3) 发送方产生一个通信密钥，对带有数字签名的信息原文进行加密。

(4) 发送方用接收方的公开密钥对自己的通信密钥进行加密。

如图6.11所示。

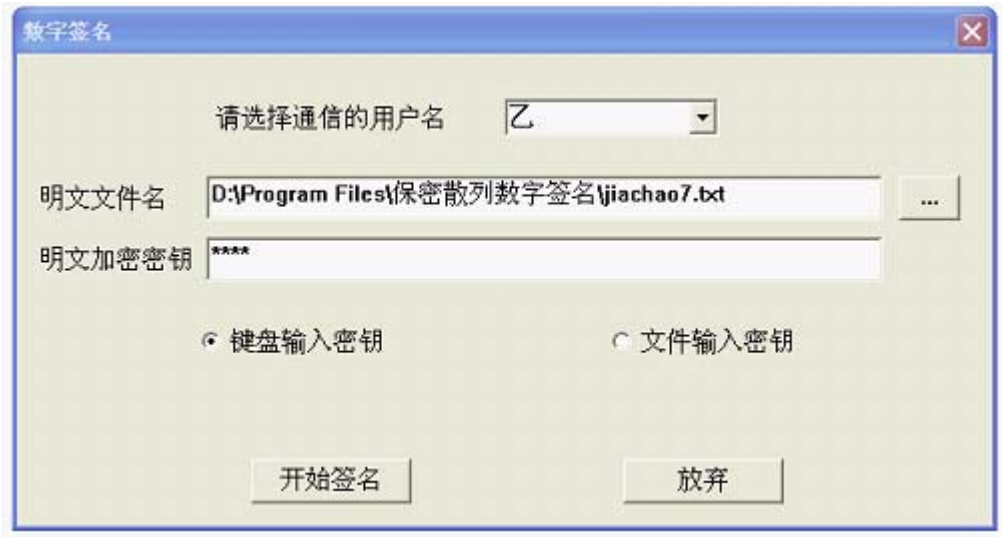

图 6.11 数字签名

Fig.6.11 Digital signature

签名后将会产生一些信息摘要,其中“.all”文件可以通过一般的电子邮件进行发送给要通信的用户，这是您不必为您的文件被别人窃取或篡改而担心；“.md5”文件是明文文件的“指纹”文件；“.sec”文件是通过对明文文件加密后的文件。

4）验证签名，点击“签名—验证签名”，可以对发送方甲所发过来的“.all”文件进行验证。

(1) 接收方收到发送方加密后的通信密钥后，用自己的私用密钥对其进行解密，得到发送方的通信密钥。

(2) 接收方用发送方的通信密钥对收到的经加密的签名原文解密，得到数字签名和信息原文。

(3) 接收方用发送方的公共密钥对数字签名进行解密，得到摘要；同时将收到的原文用 MD5 函数编码，产生另一个摘要。

(4) 接收方将两个摘要进行比较，如果两者一致，则说明发送的信息原文在传送过程中信息没有被破坏或篡改过，从而得到准确的原文。如图6.12所示。

图 6.12 验证签名

Fig.6.12 Confirm digital signature

第七章 总结和展望

§7-1 课题总结

课题对数字签名算法及具体应用进行了讨论与研究。

首先，对于各种有代表性的数字签名算法进行了讨论和研究，包括 RSA、MD5、DSA、ECDSA 等。探讨的基本上都是非对称的数字签名算法，对称的数字签名算法由于其传递密钥不像公钥密码体制方便，所以大多不采用。

然后，对数字签名的应用进行了一定的归纳和具体的阐述，并进行了分析和讨论，其主要包括以下方面：电子邮件，电子商务以及远程控制等。

最后，针对对称加密方法、公开密钥加密方法和单向散列加密方法的优点和不足，采用了一种新型的分组加密算法—排列码加密算法，并在此基础上实现了保密散列数字签名算法。保密散列数字签名算法利用 RSA 公钥密码体制的原理、MD5 单向散列函数和排列码加密的思想，构造了一种能够快速实现数据保密、数字签名的签名方案。

§7-2 展望

针对课题提出的一种基于 RSA 的保密散列数字签名，在 windows XP 操作系统平台上，用 VC++ 实现了该方案的一个原型系统，进行了初步的实验。目前更进一步的代码测试实验还在进行之中。

方案的设计尚存在一些不足，如因时间所限，方案的实现没有采用符合 RSA 体制增强安全性所要求的强素数等。为进一步增强本方案的安全性和提高效率，可考虑对本方案进行如下改进：

采用符合 RSA 体制和 Diffie—Hellman 密钥交换协议增强安全性所要求的强素数。

RSA 体制中强素数 p 、 q 应满足如下条件：

① p 、 q 位数相差不大，这样 $p-q$ 不会太小，大致与 p 或 q 位数相近。

② $p-1$ 与 $q-1$ 的最大公约数 $d=\gcd(p-1,q-1)$ 应尽量的小。

③ $p-1$ 与 $q-1$ 都应该至少含有一个大的素因数， $p+1$ 与 $q+1$ 也该至少含有一个大的素因数。

参考文献

[1] 卢开澄. 计算机密码学. 第二版. 北京:清华大学出版社,1998. 23-25.

[2] Nyberg, K. and R. A. New digital signature scheme based on discrete logarithm（comment）. Electronic Letters, 2004, 30(5):481.

[3] P. Smith and L. Skinner. A public-key cryptosystem and a digital signature system based on the Lucas function analogue to discrete Iogarithms. In:Proceedings of Asiacrypt'94, Springer, 2004.

[4] William Stallings. 密码编码学与网络安全:原理与实践(第二版). 北京:电子工业出版社,2004. 35-37.

[5] He J. and Keisler, T. Enhancing the security of EIGamal's signature scheme. IEEE proc. Comput. Digit. Tech. 2004, 141(4):249-252.

[6] 冯登国. 计算机通信网络安全. 北京:清华大学出版社, 2001. 56-58.

[7] Merkle, R. C. , "Secure Communications over Insecure Channels", Communications of the ACM, Vo l. 2l, pp. 2003. 294-299.

[8] Harn, L. and Xu, Y. Design of generalized EIGamal type digitai signature scheme based on discrete logarithm. Electronic Letters, 2004, 31(6):321-326.

[9] 张先红. 数字签名原理及技术. 第一版. 北京:机械工业出版社, 2004. 108-109.

[10] C. C. Chang, Y. F. Chang. Signing a digital signature without using one-way hash function and message redundancy schemes[J]. IEEE Commun. Lett. , 2005. vol. 8(8):485-487.

[11] 龚俭. 计算机网络安全导论. 南京:东南大学出版社, 2000. 58-62.

[12] Cao Zhenfu. Several issues in modern cryptography[DB/OL]. International Conference on Information Security, Shanghai, 2004.

[13] F. Hess, Efficient identity based signature schemes based on pairings[C]. SAC2002, Springer-Verlag, 2002, LNCS2595:310-324.

[14] 周明天, 汪文勇. TCP/IP 网络原理与技术. 北京:清华大学出版社, 1998. 56-58.

[15] 段钢等编著. 加密与解密－软件保护技术及完全解决方案. 北京：电子工业出版社, 2001. 100-102.

[16] 刘世栋, 程存学, 杜亚超. 基于PKI的电子商务安全中间件技术. 信息安全与通信保密. 2005. 26-29.

[17] WilliamStallings. 网络安全要素—应用与标准. 北京:人民邮电出版社, 2000. 35-37.

[18] K. Shum, V. Wei. A Strong Proxy Signature Scheme with Proxy Signer Privacy Protec-tion[C]. WETICE'02, 2002.

[19] K. G. Perterson. ID-based signature from pairings on elliptic curves[J]. Electron. Lett. , 2002, 38(18): 1025-1026.

[20] Malkin T, Obana S, Yung M. The Hierarchy of Key Evolving Signatures and a Charac-terization of Proxy Signatures[C]. EUROCRYPT'04, Heidelberg:Springer-Verlag, 2004.

[21] 冯登国. PKI 的基本特征及其相关标准. 密码与信息, 1999(3):200-204.

[22] 杨义先, 孙伟, 钮心忻. 现代密码新理论. 第一版. 北京:科学出版社, 2002. 112-113.

[23] 刘振华, 尹萍. 信息隐藏技术及其应用. 北京:科学出版社, 2002. 192-203.

[24] 汪旦华. 盲签名及其应用研究. 信息技术, 2005(2):35-38.

[25] Harn, L. and Xu, Y. Design of generalized EIGamal type digitai signature scheme based on discrete logarithm. Electronic Letters, 2000, 31(6):28-29.

[26] 武金木, 武优西. 建立分组密码加密技术的新概念. 河北工业大学学报, 2001(1).

[27] D. Johanson, A. Menezes, S. Vanstone. The elliptic curve digital signature algorithm (ECDSA). Internatinal Journal on Information Security, 2004, 1:36-63.

[28] N. Koblitz. A Course in Number Theory and Cryptosystems. Mat- hematics of Computation, 2005, 48: 203-209.

[29] M. O. Rabin. Digitalized signatures and public-key functions as intractable as factorization. MIT Laboratory for Computer Science Technical Report. LCS/TR-212, 1979.

[30] C. P. Schnorr. Efficient Signature Generation by Smart Card. Journal of Cryptophy, 1991, 3:161-174.

[31] 王书浩. 计算机网络中数字签名的实现方案. 微型机与应用, 1998(6):36-38.

[32] T. EIGmal, A Public Key Cryptosystem and a signature Scheme Based on Discrete Logarithms. IEEE Tractions on Information Theory IT, 1985, 31:469-472.

[33] W. Difffie. M. E. Hellman. Multiuser cryptographic technique-s. Federal Information Processing Standard Conference Proceedings, 2005, 45:109-112.

[34] N. Koblitz. A Course in Number Theory and Cryptosystems. Mat-hematics of Computation, 2004, 48: 203-209.

[35] V. Miller. Uses of elliptic curves in cryptography. Lecture Notes in Computer Science, 1986, 218: 417-426.

[36] B. Ramsdell S/MIME Version 3 Certificate Handling, RFC 2632. Internet Activity Board, 2005.

[37] 欧阳锋, 陈朝荣. 电子商务技术. 第一版. 北京:中国财经经济出版社, 2001. 270-294.

[38] 祁明, 许伯桐等. 基于弱盲签名的新型代理签名方案. 计算机工程与设计, 2001(5):57-58.

[39] Advanced Encryption Standard. Federal Information Processing Standard Publication 197, 2001.

[40] N. Ferguson R. Schroeppel D. Whiting. A simple algebraic rep-resentation of Rijndael. Lecture Notes in Computer Science, 2001, 2259:15-23.

致谢

时光如梭，转眼间硕士的学习生涯即将结束。在两年多的学习生活中，河北工业大学计算机科学与软件学院众多知识渊博的老师，浓厚的学术气氛，令我受益匪浅，在此表示感谢。

特别要感谢的是我的恩师武金木老师，他博学多才，谦虚谨慎，勤奋刻苦，治学严谨，乐观奋进，这些都潜移默化地影响着我的为人和治学态度。从论文的选题、开题、写作，直到论文的结构，甚至标点符号，都倾注了老师的心血。在生活上，武老师对我们也是关心倍至。在此再次感谢武老师，我将在今后的学术生涯中更加努力，用更出色的学术成就回报恩师。

最后，向所有评阅论文的老师、教授、专家和学者们表示最真挚的谢意！

攻读学位期间所取得的相关科研成果

贾超，武金木，吴际达．数字签名技术及其在电子商务中的应用．四川师范大学学报