摘 要

随着通信网络的迅速发展，各种网络服务广泛应用在日常工作与生活当中，而利用电子邮件进行信息交流，已经成为人们联系沟通的重要手段，与此同时，越来越重要的角色也使得电子邮件的安全问题也得到使用者越来越多的重视。

本文以安全的电子邮件系统为研究对象，研究了电子邮件的基本原理和几种有代表性的安全电子邮件协议；分析了影响电子邮件服务的各种安全因素，针对现有电子邮件系统所表现出来的保密性和身份鉴别方面的问题，设计了安全电子邮件系统的体系结构；针对电子邮件应用需求提出了一种新的有代理的多重签名方案；将平衡度的理论引入 hash 函数的设计中，从而从理论上，以一种新的增强的 hash 函数来提高信息杂凑值的抗碰撞能力；数据的加解密方案采用混合方案来提供对信息的保密性，网络通信和加解密处理分别设计并独立实现，各自作为独立组件以实现可重用、可升级。

将上述算法和安全电子邮件系统结构结合，设计并实现了一种比较安全和实用的安全电子邮件系统，并针对此原型系统对未来更进一步的工作进行了展望。

关键词 信息安全，多重签名，代理，离散对数，杂凑函数

ABSTRACT

With the rapid development of communications network, various network services are used in people's daily life, as E-mail has become one of the most important way for people to communicate between each other, by the way, of course, the more and more crucial role which the E-mail act as, make users put more and more attention on security problems.

In this paper, the basic principle and some typical secure E-mail protocol are researched, and each security factor in E-mail service are analyzed, our research focus on both authentication and confidentiality which a secure E-mail system should offered, design the architecture of secure E-mail system, proposed a new kind of proxy multi-signature to satisfy applied demand, and by using balance theory in hash design, proposed a new hash algorithm combining SHA and AES, so as to strengthen the security and anti-attacking ability of hash function ,and uses a mixed scheme to ensure the confidentiality of information, design operations of network communications and security as two normal component respectively, so can be repeatable and be updated.

In the end, through researching and designing, combine various encryption algorithms with the architecture of secure mail, a secure electronic mail system which be applied is taken up and, at the same time, there are some things need to be done in the future.

KEY WORDS information security, multi-signature, proxy signature, discrete logarithm problem, hash function

原 创 性 声 明

本人声明，所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了论文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得中南大学或其他单位的学位或证书而使用过的材料。与我共同工作的同志对本研究所作的贡献均已在论文中作了明确的说明。

作者签名：罗俊　　日期：2007 年 5 月 27 日

关于学位论文使用授权说明

本人了解中南大学有关保留、使用学位论文的规定，即：学校有权保留学位论文，允许学位论文被查阅和借阅；学校可以公布学位论文的全部或部分内容，可以采用复印、缩印或其它手段保存学位论文；学校可根据国家或湖南省有关部门规定送交学位论文。

作者签名：罗俊　导师签名：张晓伟　日期：2007 年 5 月 27 日

第一章 绪论

1.1 研究背景

电子邮件是在20世纪70年代产生的，在20世纪80年得以兴起，到20世纪90年代中期，随着因特网的普及，电子邮件被广为使用。

如今通过电子邮件进行信息交流，已经成为人们联系沟通的重要手段，而电子邮件的安全问题也越来越得到使用者的重视。一般的电子邮件中的信息，就像明信片后面的信息一样，机器与机器之间传输的信息都是公开和可获取的，机器旁的每个人都可以查看上面的内容，并且信件的内容可能在不为人知的情况下被篡改，不怀好意的人甚至还可以冒充身份发送邮件。考虑到电子邮件所传送信息的敏感性，保证其通信的安全性自然成为人们高度关心的问题。然而平常使用的电子邮件的安全性远远达不到要求，因此如何保障邮件服务的安全是一项非常重要的课题。

电子邮件以其方便、快捷、便宜、容易存储和管理的特点，得到了越来越广泛的应用。然而传统的电子邮件存在的若干不安全因素(如邮件可能在不为通信双方所知的情况下被读取、篡改或截获；发信者的身份可能被人伪造等)，使真正重要的信息仍然不宜或不敢使用电子邮件来传递。

对电子邮件的安全性问题研究虽然出来已久，并出现了很多实际应用产品，但由于包括电子邮件的通信技术的广泛使用和快速发展，对电子邮件的安全提出了更多更高的新的要求等多方面因素，以及许多研究文献都因商业利益而保密未与公开，理想的安全电子邮件系统至今仍是一个相当困难的研究任务。

1.2 国内外完全电子邮件研究现状

1.2.1 电子邮件的安全性问题

安全的电子邮件主要是解决身份鉴别和保密性的安全问题。涉及到的问题：

- 安全算法的选择
- 系统邮件的信息格式

– 如何实现认证和信任管理

– 邮件服务器的可靠性

安全算法，是保障信息保密性的最基本要素，通过对安全算法的选择，能对所需进行保密的信息内容进行处理，实现隐藏，达到保密效果。而如何选定合适的安全算法，又采取什么样的保密处理方式，从而达到什么样的保密性要求，这将是安全算法或保密方案选择所需考虑的内容。

邮件作为信息交互的载体，承载着从特定发端到特定收端的特定信息，如何借助于邮件系统，将此特定信息通过信道完成信息交互，那么，如何组织或表示此类信息，从而合理化、规范化，方便进行数据存储与管理，并能以较好的信息结构满足功能扩展与功能附加的要求，而不必牵一发而动全身，即简单而高效的信息结构，对于整个邮件服务而言是最基本的要求。

认证与信任度，显然，是在保证数据基本保密要求后的必然扩展，正如电子邮件应用范围与领域的不断延伸，数字签名技术的认可与应用需求，对于电子邮件原本简单的初衷，必然提出了更多的要求，如何辨识信息的真伪与信息交互双方身份，势必将成为我们必须面对的问题。

邮件服务，是对整个电子邮件系统所涉及各方所有功能和业务的集合，包括基本的数据生成、收发、存储，数据保密，数字签名等，而所有这些都需由服务提供端，即邮件服务器承担或协同实现，而客户端通过 Web 或 Client 来与邮件服务器进行业务往来，因而，作为网络信息服务提供者的邮件服务器必须做的有效、安全、可靠，那么如何构筑并实现邮件服务器也是相当重要的。

1.2.1 国内外的研究现状

从目前国际上安全电子邮件问题的研究，将介绍现在常用的一些 Email 标准，包括官方的标准和事实上的标准，最常见而让人熟知的有：

SMTP[1] (Simple Mail Transfer Protocol)，即：简单邮件传输协议。它是一组用于从源地址到目的地址传输邮件的规范，通过它来控制邮件的中转方式。SMTP 协议属于 TCP/IP 协议族，它帮助每台计算机在发送或中转信件时找到下一个目的地。

POP[2](Post Office Protocol)，是一种电子邮局传输协议，而 POP3 是它的第三个版本，是规定了怎样将个人计算机连接到 Internet 的邮件服务器和下载电子邮件的电子协议。它是 Internet 电子邮件的第一个离线协议标准。简单点说，POP3 就是一个简单而实用的邮件信息传输协议。

MOSS[3](MIME Object Security Services)；

PEM(Privacy Enhanced Mail)；

PGP[4,5](Pretty Good Privacy)；

MIME[7,8,9,10,11](Multipurpose Internet Mail Extension protocol)；

PGP/MIME[12](MIME Security with Pretty Good Privacy)；

S/MIME[13,14](Secure/Multipurpose Internet Mail Extensions)；

其中 MIME 对象安全服务(MOSS)和保密增强邮件(PEM)是没有被广泛实现的标准。目前许多软件厂商都使用 S/MIME 作为安全 E-mail 的标准。S/MIME 是在 PEM 的基础上建立起来的，但是它发展的方向与 MOSS 不同。它选择使用 RSA 的 PKCS#7 标准，同 MIME 一起使用来保密所有的 Internet E-mail 信息。S/MIME 的标准化工作是一个由 RSA 数据安全组织协调的工业联合会进行的。PGP 既是一个特定的安全 E-mail 应用，又是一个安全 E-mail 标准。尽管标准委员会并没有规定它是安全 E-mail 的标准，但 PGP 在全球的广泛应用已经使它成为事实上的标准。

而对于国内方面的研究和应用，亦有相当部分，天融、紫光等分别提出了安全信使 SecurityMessager、安全电子邮件 SecMail；金笛电子邮件系统；北大天正开发了第二代电子邮件系统 Mail2G，是一种宽带的电子邮件系统，它包含了一个新制定的协议、相关软件及所需的硬件环境，可以完全实现以前 E-mail 等的所有功能。但是目前国内的电子邮件安全理论和技术与国际上还有一定的差距，因此深入进行有关电子邮件的安全性理论研究和实际应用是非常必要的。

1.2.3 目前电子邮件中安全性的限制和缺陷

电子邮件作为信息的一种载体，当信息遭遇安全问题时，从使用者角度而言，这意味着如何去确保收发信息的真实性，并在尽可能低的开销下方便快捷的使用信息；从研究者的角度则意味着如何利用现有理论和技术去建立更具安全性各要素的安全应用，提供满足广大应用群体对于信息安全没有止境的需求，以新的安全的应用系统来实现，而在新的应用系统中，设计者将通过有其新的考虑角度和着眼点来开展工作，当前电子邮件的安全问题，应该说，正是随着应用的深度和广度扩展而来的，是一贯就在考虑的问题，但却是需要以新的方式面对和解决的问题。

E-mail 的安全从四个角度分析：(1)算法，(2)信息格式，(3)认证格式，(4)信任管理。详细比较了以上各种 E-mail 安全标准在这四个方面的不同表现，并对与这四个方面相关的标准进行了讨论。之所以选择这个方面进行介绍是因为这四个方面能够准确的反映出这些 E-mail 安全标准的本质特点，因此最能比较出它们的异同。

国外出现比较著名的安全电子邮件协议及其产品 PEM、PGP、S/MIME、

MOSS 等都有其缺陷或不足，PEM 认证机构的结构太严格而缺乏足够的灵活性；PGP 采用社会信任链的方式进行密钥传播，更适合于一般场合下的安全通信；S/MIME 针对企业级用户设计，由于认证机制依赖于层次结构的证书认证机构，仍然不适合国内的普通用户的使用。

归纳起来，从前面所了解到的目前电子邮件安全方案，可以看到，当前的安全性问题，主要反映在邮件系统高效实用的保密方案，在新业务需求下的认证和信任管理，高效可靠的邮件服务和安全服务提供方，因为：

尽管有众多邮件服务提供商，能提供邮件基本服务，但是由邮件内容泄露而导致的隐私侵犯或泄露案例越来越多被报道，尤其涉及金融机密的私人信函一旦流失将造成很大的影响；

电子商务、政务等对于电子邮件提出更多更高的要求，比如在共同决策领域，对于集体决策的真实有效性如何得以保证；共同决策各方如何获得公平而便捷的实施环境；如何能对其共同决策进行鉴别，以对其进行验证；

服务提供方提供便捷邮件业务的同时，如何控制服务访问，即以何种方式来提供邮件相关业务，并对所提供服务进行高效管理，控制访问，对数据存储与提取，用户及用户信息处理与冗余，而目前尽管提出方案中有很多有针对性的设计与处理，但如何实现一个比较合理控制强度和便捷实用的邮件服务系统，依然是要努力的方向。

1.3 课题的目的以及意义

目的：在学习讨论目前比较著名的安全电子邮件协议基础上，如 PEM、PGP、S/MIME、MOSS 等，实现基本的电子邮件协议，并针对现实中对电子邮件实际应用的需求，在共同决策中考虑有代理的多重签名方案，同时，信息杂凑安全应用中的重要地位及其目前所面临的问题，对现有的杂凑算法进行可行范围内的研究和讨论，进而实现安全的电子邮件系统。

意义：电子邮件作为 Internet 上应用最广泛和使用最频繁的服务，如何保证其安全性，是网络安全领域中的一个重要课题，尤其在如今网络通信技术的飞速发展和应用，但由于众所周知的原因，不能够直接采用国外的安全产品，而现存的电子邮件系统存在这方面或那反面的缺陷， 因此研究电子邮件的安全问题并开发适合我们自己实际情况的安全电子邮件系统软件，不仅具有重要的政治意义， 而且会产生巨大的经济效益。

1.4 本文的结构安排

以更为安全的电子邮件系统作为我们的出发点，研究公钥密码学理论与技术，构造出安全高效的有序多重数字签名方案；研究数字签名在具体实现中涉及到的关键技术及其解决方案，并将有序多重数字签名技术应用在工程实践中，设计相应程序。

论文共分为五章，各章内容简介如下：

第一章为绪论，主要介绍了课题的研究背景，电子邮件的发展应用，电子邮件的安全性问题的发展和研究现状，课题的研究目的和意义；

第二章主要研究和分析了电子邮件的基本原理和几种有代表性的安全电子邮件协议，分析它们各自的安全性，并且考虑实际电子邮件应用过程中存在的问题，提出如何在现实情况下期待从哪几个角度来考虑增强邮件系统的安全性；介绍相关的密码学理论基础，包括经典密码体制，公钥密码体制，杂凑函数以及多重数字签名的概念；

第三章主要分析网络通信中信息保密技术，提出了安全电子邮件系统的数据加解密方案；对于杂凑函数的广泛应用和发展现状，从平衡度的角度，结合 SHA 和 AES 算法，对杂凑函数的安全性因素在一个不同的角度进行了讨论，以提高其在实际应用中的抗碰撞能力；针对数字签名在共同决策中的日益重要的应用需求，提出了一种新的有代理的多重签名方案；

第四章以电子邮件系统理论为基础，设计与实现了一个具有邮件信息服务基本功能的安全电子邮件系统，以工程化的方法进行系统设计与实现；

第五章是总结与展望，总结了本文所做的工作，指出不足之处，提出以后的努力方向。

第二章 安全电子邮件理论基础

在讨论如何构建安全的电子邮件系统之前，对将涉及的密码学部分进行简单的提及，具体的分析和特定的讨论在后续内容中作详细叙述。

信息以编码的形式进行表示，而信息的保护必然地从其表现形式进行考虑，如何从其表现方式来隐蔽和保护需要保密的消息，使得未授权者不能提取信息也不能篡改信息。被隐蔽的消息称作明文，隐蔽后的消息称作密文。将明文变换成密文的过程称作加密，其逆过程，即由密文恢复出原明文的过程称作解密。信息发送方对明文进行加密时采用的一组规则称作加密算法，而信息接收方对密文进行解密时所采用的一组规则称作解密算法，加密与解密算法的操作通常都是在一组密钥控制下进行的，分别称为加密密钥和解密密钥。而根据密钥的特点，密码体制分为对称和非对称密码体制两种。对称密码体制又称私钥或传统密码体制，非对称密码体制又称为单钥或公钥密码体制。

2.1 电子邮件基本结构

电子邮件系统包含3个部分：用户代理(User Agent)、传输代理(Transfer Agent)和接收代理(Delivery Agent)。用户代理是一个用户端发信和收信的程序，负责将信按一定的标准包装，送至邮件服务器或将信件从邮件服务器收回，传输代理则负责信件的交换和传输，将信件转发至适当的邮件服务器，再由接收代理将信件发至不同的邮件箱。传输代理能够解读收信人的地址，并根据 SMTP[1](Simple Mail Transport Protocol)将它转送至邮件主机，待邮件到达邮件主机后再接收 POP[2](Post Office Protocol)，使邮件被用户读取至自己的主机上。

Email 在 Internet 上传送时会经过许多站点，在一些邮件服务器上作短暂停留。邮件服务器要查看信头，以确定该信息是否发给自己，如不是，邮件服务器将会转到下一个最有可能的地址。实际的操作是：邮件服务器有一个“路由表”列出了其它邮件服务器和目的地的地址，当服务器读完信头，发现不是发给自己时，就会迅速将信件发至目的地服务器或离目的地最近的服务器。

2.2 电子邮件基本协议

2.2.1 SMTP 简单邮件传输协议

SMTP 协议是 TCP/IP 协议族中的一员，主要对如何将电子邮件从发送方地址传送到接收方地址，也即是对传输的规则做了规定。SMTP 协议的通信模型并不复杂，主要工作集中在发送 SMTP 和接收 SMTP 上：首先针对用户发出的邮件请求，由发送 SMTP 建立一条连接到接收 SMTP 的双工通讯链路，这里的接收 SMTP 是相对于发送 SMTP 而言的，实际上它既可以是最终的接收者也可以是中间传送者。发送 SMTP 负责向接收 SMTP 发送 SMTP 命令，而接收 SMTP 则负责接收并反馈应答。可大致用下面的通讯模型示意图来表示：

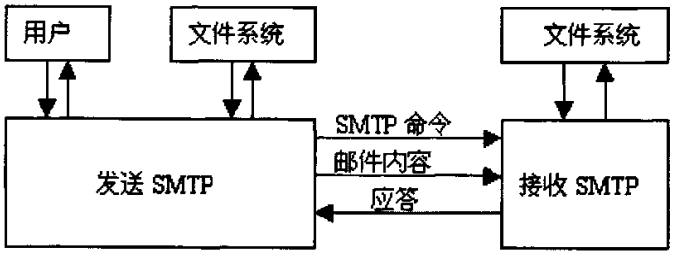


图 2-1 SMTP 通讯模型示意图

2.2.2 IMAP 消息访问协议

IMAP[15](即 Internet 消息访问协议)是与我们通常熟知的 POP3 对应的另一种协议，为美国斯坦福大学在 1986 年开始研发的多重邮箱电子邮件系统。它能够从邮件服务器上获取有关 E－mail 的信息或直接收取邮件，具有高性能和可扩展性的优点。IMAP 为很多客户端电子邮件软件所采纳，如 OutlookExpress、NetscapeMessenger 等，支持 IMAP 的服务器端的软件也越来越多，如 CriticalPath、Eudora、iPlanet、Sendmail 等。当然，正如大家所知，POP3 也是接收邮件的协议，现在不是用得很好么，为何还要用 IMAP 协议呢？

POP3 协议的不足在于，尽管 POP 作为 Internet 上邮件的第一个离线协议标准，允许用户从服务器上把邮件下载到本地主机上，同时删除保存在邮件服务器上的邮件，从而使用户不必长时间地与邮件服务器连接，很大程度上减少了服务器和网络的整体开销。但 POP3 有其天生的缺陷，即当用户接收电子邮件时，所有的信件都从服务器上清除并下载到客户机。在整个收信过程中，用户无法知道邮件的具体信息，只有照单全收入硬盘后，才能慢慢浏览和删除。这使用户几乎

没有对邮件接收的控制决定权。一旦碰上邮箱被轰炸，或有比较大的邮件，用户不能通过分析邮件的内容及发信人地址来决定是否下载或删除，从而造成系统资源的浪费。而 IMAP 协议不但可以克服 POP3 的缺陷，而且还提供了更强大的功能。

2.2.3 MIME 协议

下例是一个邮件的标准 MIME 头：

```
//版本号：1.0
Mime-Version: 1.0
//内容类型是多种的
Content-Type: multipart/mixed; boundary="IMA.Boundary.750407228"--I
MA.Boundary.750407228
//内容类型：文本，字符是 ASCII 的
//编码方式：7 位
Content-Type: text/plain; charset=US-ASCII Content-Transfer-Encoding: 7
bit //编码方式：7 位
Content-Description: cc:Mail note part
```

MIME(Multipurpose Internet Mail Extension)，即多用途的网络邮件扩充协议。现在它已经演化成一种指定文件类型(Internet 的任何形式的消息：E-mail，Usenet 新闻和 Web)的通用方法。在使用 CGI 程序时你可能接触过 MIME 类型，其中有一行叫做 Content-type 的语句，它用来指明传递的就是 MIME 类型的文件(如 text/html 或 text/plain)。

2.3 电子邮件的安全理论基础

安全性的考虑在简单电子邮件协议之后，相继提出过了很多与邮件信息处理相关的安全增强型方案，比如 MOSS，PEM，PGP，MIME，S/MIME 等，其中 PGP 作为应用最为广泛的电子邮件方案，由于其公开的特点，并结合多种加密手段来进行安全保障，有着相当范围的影响力，当然，同样基于我们主旨的考虑，以及更新的现实安全需要，尽管 PGP 软件的实现自身并没有应用独有的信息安全的新技术，但由于它组合了加密，数字签名、压缩、基数 64 转换、报文分段等多种先进技术，小巧而强大，使得它在加密重要文件、电子邮件以及为签发的文件作数字签名等信息安全方面应用日益广泛，其设计思想值得借鉴。

2.3.1 对称加密体制

1. DES

DES（Data Encryption Standard）算法，即数据加密标准，作为美国联邦信息处理标准，在通常所使用的加密算法中，应该是迄今最为广泛应用的加解密算法，其最初密钥长度为 128 位，采用回合、排列和置换操作，将明文的顺序打乱(排列)和替换(置换)的重复循环(回合)，在于一个随机密钥组合起来，从而来提供安全保障，后来随着计算能力的增强，三重 DES 作为临时的标准，亦即通过对明文块的 3 次 DES 加密来增强其安全性，当然，可想而知其速度受限。

2. AES

AES（Advanced Encryption Standard）算法，即高级加密标准算法，是由美国国家标准技术研究所（NIST）着手开发的用于替代 DES 的新标准。经过三年的评估、分析和攻击测试，来自比利时的密码学专家 Joan Daemen 和 Vincent Rijmen 设计的 Rijndael 算法最终获胜，成为美国推荐的高级加密标准算法。

AES 算法是一种分组密码算法，分组长为 128 比特，密钥长可以为 128、192 或 256 比特三种。AES 算法的数据处理单元是字节，128 比特的分组信息被分成 16 个字节。AES 算法中引入了矩阵的概念，分组的 16 个字节按顺序被复制到一个 4*4 的矩阵中，称为状态（state），AES 的所有变换都基于状态的变换。

AES 变换是由轮变换通过多轮迭代实现的，根据密钥长度的不同，轮函数的迭代次数也不一样，对应上面的三种密钥长度，迭代次数分别为 10、12、14 轮。轮函数的构成包括非线性、扩散和密钥调度几种元素。

非线性变换的目标就是通过较小较简单的非线性元素得到大的复杂化的非线性部件。在轮函数的每一轮迭代中，包括 4 步变换，分别是字节代替运算（ByteSub()）、行变换（ShiftRow()）、列混合（Mixcolumn()）以及轮密钥加变换（AddRoundKey()），其作用就是通过重复简单的非线性变换、混合函数变换，将字节代替运算产生的非线性扩散达到充分的混合，使加密后的分组信息统计特性分布更均匀，在每轮迭代中引入不同的密钥，这样便以最简单的运算代价得到最好的加密效果，实现加密的有效性。

由此可见，AES 算法设计简单，密钥安装快，需要的内存空间少，在所有平台上运行良好，支持并行处理，并且 AES 算法的差分均匀性和线性偏差都达到了最佳，可以很好的抵御差分密码分析和线性密码分析。

2.3.2 公钥密码体制

以 RSA[16]，ElGamal 和 NTRU[17]为代表的公钥密码体制，是从 1976 年公钥密码的思想提出以来国际上提出的，这几种是众多种公钥密码体制中比较典型的。

用抽象的观点来看,公钥密码就是一种陷门单向函数。说一个函数 f 是单向函数,即若对它的定义域中的任意 x 都易于计算 f(x),而对 f 的值域中的几乎所有的 y,即使当 f 为已知时要计算 $f^{-1}(y)$在计算上也是不可行的。若当给定某些辅助信息(陷门信息)时则易于计算 $f^{-1}(y)$,就称单向函数 f 是一个陷门单向函数。公钥密码体制就是基于这一原理而设计的,将辅助信息(陷门信息)作为秘密密钥。这类密码的安全强度取决于它所依据的问题的计算复杂度。

目前比较流行的公钥密码体制主要有两类:一类是基于大整数因子分解问题的,其中最典型的代表是 RSA 体制。另一类是基于离散对数问题的,如 ElGamal 公钥密码体制和影响比较大的椭圆曲线公钥密码体制。

2.3.3 数字签名

在现实生活中，长期以来文件上的手写签名一直被用作签名者身份的证明。这是因为：

签名是可信的；签名是不可伪造的；签名是不可重用的；签名的文件是不可改变的；签名是不可抵赖的在未来社会的生活中，电子文档将逐步代替纸质的文件成为信息交流的主体。证明某一个电子文件是某作者所做的有效办法是模拟普通的手写签名在电子文档上进行电子签名，即在电子化文件中添加可以标记自己的一段特征数据来实现签名。作者可以通过数字签名表明自己的身份，读者可以通过数字签名验证作者的身份。

电子邮件是互联网上最重要的应用之一，传统的电子邮件都是明文传输并且发送方可以轻松地伪造自己的身份。随着电子邮件的应用扩展到各种信息敏感领域，如:政府间来往、商业谈判等，电子邮件的内容保密和发送方身份确认的重要性便逐步凸显出来。数字签名能很好地解决电子邮件的身份确认问题，同样电子商务是互联网上发展最快的应用方向，它是借助于互联网的快速信息传输能力来完成各种商务活动，包括电子数据交换、在线交易系统、网上银行、商业增值网等。互联网是一个开放的空间，任何人都可以进入，而重要的商务信息具有敏感性和保密性，所以通常的商务信息在传输中要进行加密，同时，为了进一步防止欺骗性的篡改，数字签名是必不可少的。电子商务活动中的电子订单、电子帐单、电子收据、电子合同等电子文档都需要作数字签名以确保真实性。

基于公钥密码体制的数字签名技术随着公钥密码体制的诞生而产生，DSS是1991年8月由美国国家标准技术学会(National Institute of Standards and Technology)提出并于1994年12月被采纳的一个数字签名标准(Digital Signature Standard)[20]，该标准包括一个数字签名算法(DSA, Digital Signature Algorithm)[21]，它曾引起大量的争论。自从数字签名的概念提出以后，每年都有与之相关的众多的研究论文发表，随着研究的深入，数字签名的应用领域也在不断扩大，许多基于网络的新应用都离不开数字签名技术，如:匿名电子选举，电子拍卖，电子商务等。

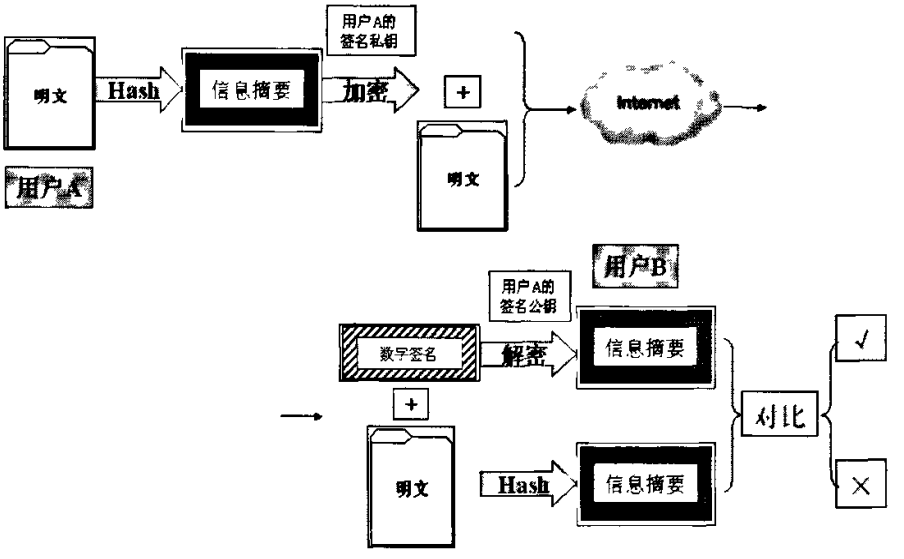


图 2-2 普通数字签名

在我们的通信安全模型里，通常有四个角色。第一是信息的发送者，通常将之叫做 Alice。第二个角色是接收者，或者说是验证者，通常将之称作 Bob。第三个角色是攻击者，或者说是分析者，通常将之叫做 Oscar。Oscar 一般会试图获取 Alice 和 Bob 之间的秘密，或者是试图使他们接受他所伪造的一些信息。Oscar 这个角色对于 Alice 和 Bob 来说是敌对者，而注意到，正当的部门(如军事或公安部门)也可能以这个角色出现。所以更倾向于使用分析者这个词语。第四个角色是可信任的部门(Trust Center)，通常做一些发行证书，充当仲裁者等等工作。他们之间的关系如图 2-3。

以上通信模型仍然适用于数字签名。在签名协议里面，Alice 是签名者，她把一段信息附上签名后发送给验证者。设 Bob 是验证者，他收到信息及其签名后可以验证其真伪。当任何人都可以验证 Alice 的签名时，Bob 这个角色就不是指特定的某个人，而代表任何可以收到签名信息的人。Oscar 在签名协议里是分

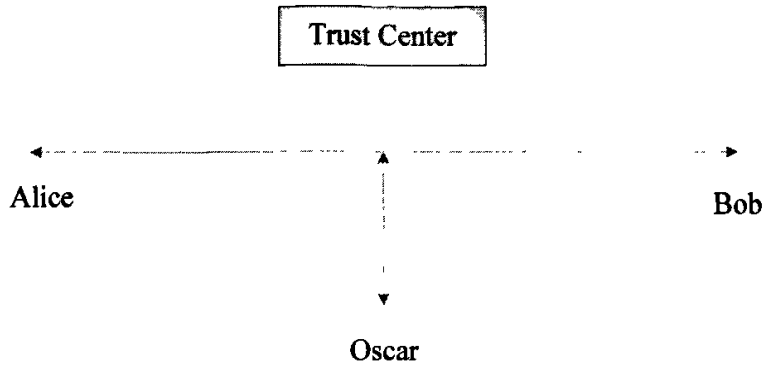


图2-3 通信安全模型的四种角色

析者。在某些情况下，他只是窃听 Alice 和 Bob 之间的通信。在另一些情况，他伪装成 Alice 的角色，企图让 Bob 认为自己所收到的信息是 Alice 所发。Oscar 也可以伪装成 Bob 的角色，他和 Alice 进行直接通信，并企图从中获取 Alice 的秘密信息。

签名协议可以保证数据的完整性，这就隐含了签名协议有一个解决争端的机制。因为，签名者 Alice 可能会否认她对某个文件签过名(而实际上她签过)。Bob 或 Oscar 也有可能声称 Alice 对某个文件签过名(而实际上 Alice 没有签过)。这些争端的解决可以通过签名协议中的验证机制来保证。

当前最主要的公钥密码体制有 RSA 密码体制、ElGamal 密码体制和椭圆曲线密码体制，它们都是基于某一个计算困难问题，一旦能有有效的多项式时间算法解出该计算困难问题，则相应的公钥密码体制也就被破解。数字签名技术是依赖于公钥密码体制，其安全性同样也是基于某个计算困难问题。历史上，曾经提出的基于背包问题的大多数背包密码体制都先后遭到破译，仅有 Chor-Rivest 背包密码体制被认为是安全的，但计算量太大，不实用。椭圆曲线公钥密码体制被认为是能产生更强更快密码算法的一种密码体制，该密码体制下密钥的长度要明显地小于另外两种密码体制。

数字签名技术发展到今天，其理论研究和应用开发上作都得到长足的发展。在基础理论研究方面，许多新型的数字签名概念被提出，各式各样的数字签名新算法也是层出不穷。

多重签名[22]，一种由多人参与对同一文件进行分别签名的特殊数字签名。多重签名是一种基本的签名方式，它与其它数字签名形式相结合又派生出许多其它签名方式，如代理多重签名，且有相当多的文献与之相关。

群签名[23]，由个体代表群体执行签名，验证者从签名不能判定签名者的真实身份，但能通过群管理员查出真实签名者。这是近几年一个研究热点，研究重点

放在群公钥的更新、签名长度固定和群成员加入与撤消上。最近，一种新的动态群签名模型也已提出。

盲签名[24]，是一种让签名人不知道所签名文件内容的签名形式。它能使所签名文件的内容不被签名者获知，保护了个人的隐私。盲签名这一性质能结合到其它签名方式中，形成新的签名方式。如:群盲签名，盲代理签名，代理盲签名，盲环签名等。

代理签名[25]，是将签名权委托给代理签名者，由他代替自己行使签名。它在电子商务中有着广泛的应用，相关研究主要集中在对代理签名者签名权的控制问题上。它也是一个与其它签名技术结合较多的一种签名形式。

2.3.4 信息杂凑

在现今的信息安全领域，借助于传统密码体制和现代密码体制，能构造出很多现实研究和实际需求的安全方案，当然，不乏各种加解密应用[18]。在日益广泛和必要的信息安全应用中，信息完整性的确保又是这其中很多种现实应用中至关重要的部分，如何来获得完整性对于信息而言，是否已经受到攻击而被篡改。

通过单向散列函数获得特定信息的消息摘要，将为我们对于完整性的要求提供可能。

单向散列函数满足的条件[19]：

对任意长度的明文 m，经由散列函数 h 可产生固定长度的散列值，散列函数值是对明文的一种“指纹”或是“摘要”。所以对散列值的数字签名，就可以视为对此明文的数字签名。使用散列函数可以提高数字签名的效率。使用在数字签名上的散列函数必须满足以下条件：

1．散列函数必需对任意长度的明文，产生固定长度的散列函数值。

2．对任意的明文 m，散列函数值 h(m)通过软件或硬件很容易实现。

3．对任意的散列函数值 x，要找到一个明文 m 与之对应，即 x=h(m)在计算上是不可行的。

4．对一个明文 m1，要找到另一个不同的明文 m2，而且具有相同的散列函数值 h(m1)=h(m2)，在计算上是不可行的。

5．要找到任意一对不同的明文(m1,m2)，而且具有相同的散列值 h(m1)=h(m2)，在计算上是不可行的。

能同时满足条件 1~5 的称为“强散列函数”。在数字签名上，必须是强散列函数。

2.4 本章小结

从电子邮件基本结构和协议入手，分析了 SMTP，IMAP 和 MIME 各自的特点，从而为系统设计和邮件安全性的分析与设计进行理论准备，并基于此，为设计安全的电子邮件系统做准备；相关的对称加密体制和公钥密码体制的简单介绍，以各自特点而言，综合这两种体制来进行安全方案的设计，被认为是实用而可行的，并简要的对信息杂凑和数字签名进行说明，在以后的内容中，将依照这些主题对课题展开讨论和设计。

第三章 安全电子邮件研究与设计

邮件服务作为基本网络服务之一，是最原始也是最广泛的网络应用之一，网络的安全问题和理论随着网络应用的日益扩展也不断发展，邮件安全问题其实是网络安全问题这个范畴里的一个特定主题，网络安全问题的出现和解决具有其普遍性，而电子邮件服务特定情况和应用下的安全性讨论将从电子邮件通信所涉及各方面来进行网络安全中某些安全因素的讨论。

3.1 Secure E-mail 中一种新的有代理的多重签名方案

人们为了实现共同决策，如商业合同的多方参与共同签署提出了多重签名；为了在签名人因故而不能行使合法签名权时，将签名权委派给某人以代替自己行使签名权，提出了代理签名[25]；后来又提出了多重代理签名[28]和代理多重签名[29]，前者是指原始签名人可以授权给一组代理签名人，只有代理签名组中的所有签名人才能完成代替原始签名人的代理签名；后者是指一组原始签名人授权给一个代理签名人，并且只有原始签名人组成员同时授权给此代理签名人时，代理签名人才能代替原始签名人组完成签名。

在这里将提出一种新的有代理的多重签名方案。在这种方案中，多个原始签名人将分别对自己选定的可信任的代理签名人进行代理授权，在签名过程中，若代理签名人所获授权未过期，则原始签名人与其对应的代理签名人均可实现原始签名人的合法签名；若代理签名人所获授权已过期或已被原始签名人取消代理授权，则只有原始签名人能实现其合法签名。在签名认证中，验证人能验证签名者身份，即签名为原始签名人还是其代理签名人所签。

显然，这种新的有代理的多重签名方案不同于一般的多重签名方案和代理签名方案；也不同于多重代理签名方案和代理多重签名方案,因为后者或者是多人代替一人签名即多重代理；或者一人代替多人签名即代理多重。新的有代理的多重签名方案始终为多方共同参与进行多重签名，同时，在保障安全的条件下，我们引入代理人的参与，从而实现签名的灵活性。只要代理签名人权限有效，那么原始签名人与其代理签名人均可作为代表参与签名协议的实施，并生成合法签名。生成的数字签名是可验证的，若有需要，亦可验证签名者身份，即可确认该签名为原始签名人还是其代理签名人所签。

3.1.1 系统参数

p 是大素数；q 是 $p-1$ 或 $p-1$ 的大素因子；

$g \in_R Z_p^*$，且 $g^q \equiv 1 (\mathrm{mod}\, p)$；

可信中心为 T，x 为其秘密密钥，$x \in_R Z_q^*$，y 为其公开密钥，$y = g^x (\mathrm{mod}\, p)$；

设 $U_i (1 \le i \le N)$ 为第 i 个原始签名人，则 x_{U_i} 为原始签名人 $U_i (1 \le i \le N)$ 的秘密密钥，$x_{U_i} \in_R Z_q^*$，y_{U_i} 为原始签名人 $U_i (1 \le i \le N)$ 的公开密钥，$y_{U_i} = g^{x_{U_i}} (\mathrm{mod}\, p)$；设 $P_i (1 \le i \le N)$ 为对应的原始签名人 $U_i (1 \le i \le N)$ 的代理签名人，x_{P_i} 为代理签名人 $P_i (1 \le i \le N)$ 的秘密密钥，$x_{P_i} \in_R Z_q^*$，y_{P_i} 为代理签名人 $P_i (1 \le i \le N)$ 的公开密钥，$y_{P_i} = g^{x_{P_i}} (\mathrm{mod}\, p)$。

3.1.2 代理授权

1．每个原始签名人 $U_i (1 \le i \le N)$ 选取一个随机数 $k_{U_i} \in Z_q^*$，计算：

$$K_{U_i} = g^{k_{U_i}} (\mathrm{mod}\, p)$$

每个代理签名人 $P_i (1 \le i \le N)$ 选取一个随机数 $k_{P_i} \in Z_q^*$，计算：

$$K_{P_i} = g^{k_{P_i}} (\mathrm{mod}\, p)$$

2．对于单向哈希函数 H（），每个原始签名人及其代理签名人分别计算：

$$U_i (1 \le i \le N)：\ h_{U_i} = H(K_{U_i})；$$

$$P_i (1 \le i \le N)：\ h_{P_i} = H(K_{P_i})；$$

3．接着，每个原始签名人及其代理签名人分别计算：

$$U_i (1 \le i \le N)：\ \dot{O}_{U_i} = (x_{U_i} + k_{U_i}) h_{U_i} (\mathrm{mod}\, q)$$

$$P_i (1 \le i \le N)：\ \dot{O}_{P_i} = (x_{P_i} + k_{P_i}) h_{P_i} (\mathrm{mod}\, q)$$

4．$U_i (1 \le i \le N)$ 将各自计算的结果：$(K_{U_i}, h_{U_i}, \dot{O}_{U_i})$ 通过安全信道发送给 T；$P_i (1 \le i \le N)$ 将各自计算的结果：$(K_{P_i}, h_{P_i}, \dot{O}_{P_i})$ 通过安全信道发送给 T。

5．T 对于来自 $U_i (1 \le i \le N)$ 的 $(K_{U_i}, h_{U_i}, \dot{O}_{U_i})$，验证等式成立与否：

$$g^{\dot{O}_{U_i}} = (y_{U_i} \cdot K_{U_i})^{h_{U_i}} (\mathrm{mod}\, p)，否则中止；$$

T 对于来自 $P_i(1 \le i \le N)$ 的 $(K_{P_i}, h_{P_i}, \dot{O}_{P_i})$，验证等式成立与否：

$$g^{\dot{O}_{P_i}} = (y_{P_i} \cdot K_{P_i})^{h_{P_i}} \pmod p$$，否则中止。

若均成立，则可信中心 T 对每对 (U_i, P_i) $(1 \le i \le N)$ 对应的 (k_{U_i}, k_{P_i}) 进行交叉存取控制（具体在 3.1.3 中介绍），参考图示如下：

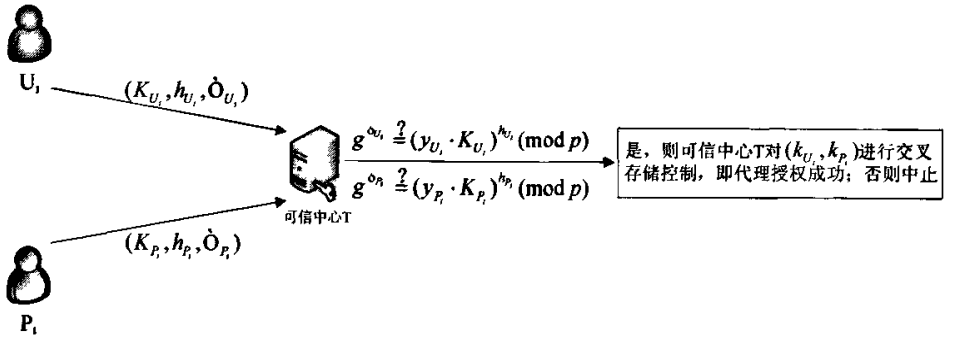


图3-1　原始签名人Ui与其代理签名人Pi间通过可信中心T进行代理授权

3.1.3 签名生成

可信中心 T 在签名开始之前，对 $(1 \le i \le N)$ 分别随机选取 r_i'，

$R_i' = g^{r_i'} \pmod p$。

1．N 个签名人（注：此时 N 个签名人中，可根据实际情况，允许有 N' 个原始签名人和 $N - N'$ 个代理签名人参与）在签名前，分别向可信中心 T 发送签名请求，出示 K_{U_i} 或 K_{P_i}。T 在确认其身份后，计算：

若签名人为 U_i：

$$R_i = R_i' \cdot g^{k_{P_i}} = g^{r_i'} \cdot g^{k_{P_i}} = g^{r_i' + k_{P_i}} = g^{r_i} \pmod p$$

若签名人为 P_i：

$$R_i = R_i' \cdot g^{k_{U_i}} = g^{r_i'} \cdot g^{k_{U_i}} = g^{r_i' + k_{U_i}} = g^{r_i} \pmod p$$

因为可信中心 T 随机选取 r_i'，所以：

$r_i = r_i' + k_{P_i}$ 或 $r_i = r_i' + k_{U_i}$ 亦可视为随机选取，$r_i \in_R Z_q^*$，有 $R_i = g^{r_i} \pmod p$。

2．可信中心 T 计算：

$$h_i = H(R_i) \qquad (1 \le i \le N)$$

$$\dot{O}_i = x \cdot h_i + r_i (\mathrm{mod}\, q) \qquad (1 \le i \le N)$$

并将$(R_i, \dot{O}_i, h_i)$通过安全信道发送给第i $(1 \le i \le N)$签名人；

第i $(1 \le i \le N)$签名人验证：

$$g^{\dot{O}_i} = y^{h_i} \cdot R_i (\mathrm{mod}\, p)$$

是否成立，否则中止。

3．第i $(1 \le i \le N)$签名人随机选取$t_i \in_R Z_q^*$，并计算出：

$$B_i = g^{t_i} (\mathrm{mod}\, p)$$

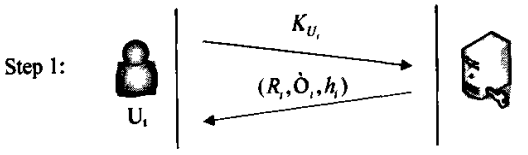


图 3-2 用户与密钥服务器间确认身份并进行初始化

并将B_i发送给其他$N-1$个签名人；

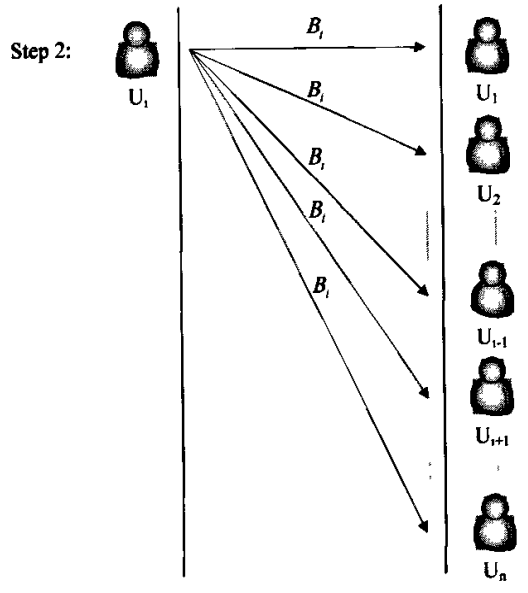


图3-3　用户将个人中间签名量分发给其余签名人

4．第i $(1 \le i \le N)$签名人在收到其他$N-1$个B_j后，计算出：

$$B = B_1 B_2 \cdots B_N (\mathrm{mod}\, p)$$

5．第i $(1 \le i \le N)$签名人计算出：

$$s_i = m\dot{O}_i - Bt_i (\mathrm{mod}\, q)$$

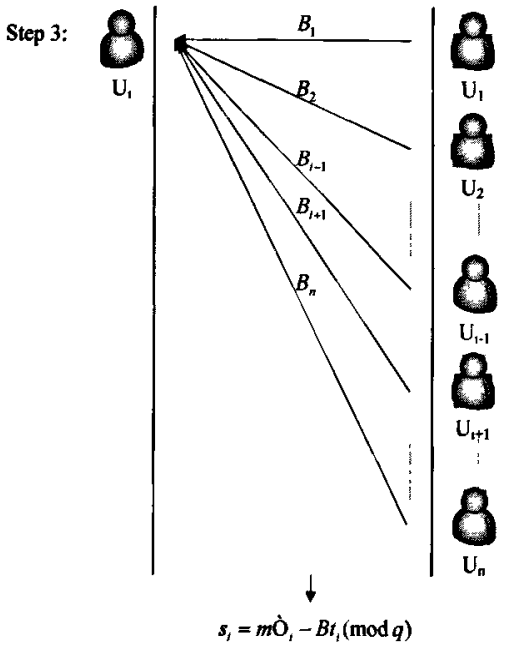


图3-4 用户在收集齐其余签名人所分发的个人中间签名后生成个人签名向量

并将(R_i, B_i, s_i)发送给某指定的签名人Q。

6. Q接收所有N个签名人发送来的$(R_i, B_i, s_i)\ (1 \le i \le N)$，计算出：

$$B = B_1 B_2 \cdots B_N \pmod p$$

验证等式：

$$(y^{h_i} \cdot R_i)^m = g^{s_i} \cdot B_i^B \pmod p$$

是否成立，否则中止。

如果所有验证等式：

$$(y^{h_i} \cdot R_i)^m = g^{s_i} \cdot B_i^B \pmod p\ (1 \le i \le N)$$

均成立，将由Q计算：

$$s = s_1 + s_2 + \cdots + s_N \pmod q$$

最后以：

$$(R_1, R_2, \cdots, R_N, B, s)$$

作为对消息m的有代理的多重签名。如图3-5所示：

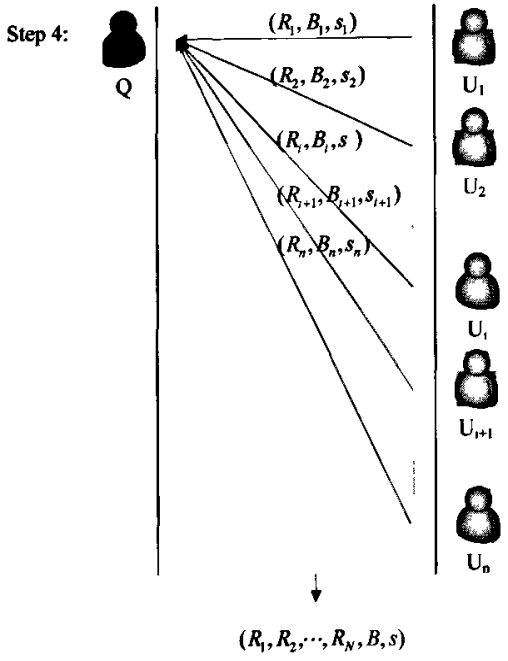


图3-5 签名接收人接收所有签名人的个人签名向量，并生成最终的消息签名

3.1.4 签名验证

验证人V，对消息m的有代理的多重签名$(R_1, R_2, \cdots, R_N, B, s)$，计算：

$$\bar{y} = (y^{h_1} \cdot y^{h_2} \cdots y^{h_N}) \cdot R_1 \cdots R_{N-1} \cdot R_N \pmod{p}$$

然后验证等式：

$$\bar{y}^m = B^B \cdot g^s$$

是否成立，成立则消息m的签名：

$$(R_1, R_2, \cdots, R_N, B, s)$$

为有效的有代理的多重签名；否则签名无效。

3.1.5 方案讨论

从上述方案的描述中，原始签名人对自己选定的可信任的代理人进行代理授权，在可信中心T的参与下实现代理授权和登记；在签名过程中，若代理签名人所获授权有效，则原始签名人与其对应的代理签名人均可实现原始签名人的合法签名；若代理签名人所获授权已被原始签名人取消或已由新指定的代理人取代，则原始签名人和其新代理人能实现合法签名，在 3.1.2 中，原始签名人可根据其

需要方便地更换其代理人，因为代理授权，是在可信中心T的协助下，原始签名人与代理人之间的协议活动，不涉及系统其他成员，从而不会对整体造成影响，保证了系统的稳定性；而系统基于离散对数进行构建，从而以求解离散对数的难度保证了此签名方案的安全性；在签名认证中，验证人能验证签名者身份，即签名为原始签名人还是其代理签名人所签。这种新的有代理的多重签名方案，始终为多方参与进行多重签名，并且提出只要代理签名人权限有效，那么原始签名人与其代理签名人均可参与此签名，并生成合法签名，不同于一般的多重签名方案和代理签名方案；也不同于多重代理签名方案和代理多重签名方案。若有需要，能由可信中心T验证签名，若有需要亦可验证签名者身份，即签名为原始签名人还是其代理签名人所签。

不同的数字签名方案的提出都是着眼于解决现实环境下不同的身份认证问题，这种新的有代理的多重签名方案也无非是面对应用的多样化而提出的，相信能满足在共同决策等领域的特定要求，如在电子邮件服务中应用于商务或政务决策等。

3.2 Secure E-mail 的杂凑方案

3.2.1 杂凑函数发展现状及常用杂凑算法

国际密码学会议（Crypto 2004）上，Eli Biham 和 Antoine Joux 相继做了对 SHA-1 的分析与给出 SHA-0 的一个碰撞，分析表明，于 1994 年替代 SHA-0 成为联邦信息处理标准的 SHA-1 的减弱条件的变种算法能够被破解；但完整的 SHA-1 并没有被破解，也没有找到 SHA-1 的碰撞。研究结果说明 SHA-1 的安全性暂时没有问题，但随着技术的发展，技术与标准局计划在 2010 年之前逐步淘汰 SHA-1，换用其他更长更安全的算法（如 SHA-224、SHA-256、SHA-384 和 SHA-512）来替代。

也是在本次会议上，Xiaoyun Wang 等人做了题为《Collisions for Hash Functions MD4,MD5,HAVAL-128 and RIPEMD》[30]的研究成果报告，并在发表的论文《How to Break MD5 and Other Hash Functions》[31]中已经就 hash 可能碰撞的产生进行了讨论，并对 MD4,MD5,HAVAL-128 和 RIPEMD，这四种 HASH 函数都给出了碰撞。而 MD5 就是这样一个在国内外有着广泛的应用的杂凑函数算法，它曾一度被认为是非常安全的。然而，Xiaoyun Wang 等人，可以很快的找到 MD5 的“碰撞”，并且有望以更低的复杂度完成对 SHA-0 的攻击，能在任何初始值下用 2^{40} 次 hash 运算找出 SHA-0 的碰撞,就是两个文件可以产生相同的“指纹”。这意味着，当你在网络上使用电子签名签署一份合同后，还可能找到另外

一份具有相同签名但内容迥异的合同，这样两份合同的真伪性便无从辨别。Xiaoyun Wang 等人的研究成果证实了利用 MD5 算法的碰撞可以严重威胁信息系统安全，这一发现使目前电子签名的法律效力和技术体系受到挑战。

而安全的杂凑函数在设计时必须满足两个要求：

其一是寻找两个输入得到相同的输出值在计算上是不可行的，这就是通常所说的抗碰撞的；

其二是找一个输入，能得到给定的输出在计算上是不可行的，即不可从结果推导出它的初始状态。现在使用的重要计算机安全协议，如 SSL、PGP 都用杂凑函数来进行签名，一旦找到两个文件可以产生相同的压缩值，就可以伪造签名，给网络安全领域带来巨大隐患。

可以发现领域似乎已经给出了最直接的解决方案，即采用更长的杂凑函数值，256 位，384 位更长可以是 512 位，诚然，面对当前情形，不失为一种救急的方法，而能不能尝试从另外的角度来提高 hash 函数性能，就成了我们将要进行的一部分工作。

那么，如何在此条件下，进行 Secure E-mail 的杂凑值方法设计，是一个要考虑的问题。首先，我们讨论目前有哪些主要的方案以及各 HASH 方案的特点，常用的散列函数，HASH 函数 MD4 的设计是不基于任何假设和密码体制的，为了增强 MD4 的安全性和抗攻击程度，由 MD4 衍生出多种 HASH 函数如 MD5、SHA-1、RIPEMD-160 等，而由 MD4 函数衍生出的 MD5、SHA-1、RIPEMD-160 算法设计及其安全性，对这些常用函数做适当的了解和分析。

目前常用的散列函数有 MD5 杂凑算法和 SHA[26]安全杂凑算法，RIPEMD-160 算法。

1. MD5 可以输入任意长度的明文，产生 128 位的摘要。任意长度的明文首先需要添加位的数目，使明文总长度与 448(521-64)在模 512 中同余，在明文后添加位的方法是第一个添加位是“1”外，其余都是“0”。然后再将真正明文的长度(没有添加位以前，以 64 位表示)附加于前面已添加过位的明文后，此明文的长度正好是 512 位的倍数。用 32 位软件易于高速实现。对纯强力攻击寻找一个消息具有给定 Hash 值的计算困难性为 2^{128}，若采用生日攻击法，寻找有相同 Hash 值的两个消息需要试验 264 个消息。

2. SHA 算法是美国 NIST 和 NSA 为配合数字签名标准(DSA)设计的一种标准算法，对任意长度明文的预处理和 MD5 的过程一样，输出 160 位的消息摘要，分别存储在 5 个 32 位的记录单元中，MD5 与 SHA 的差异如表 3-1 所示。

表 3-1 MD5 与 SHA 的差异

差异处	MD5	SHA
摘要长度	128 位	160 位
运算步骤数目	64	80
基本逻辑函数数目	4	4
常数数目	64	4

(1) 安全性：SHA 所产生的摘要比 MD5 长 32 位，因此 SHA 比 MD5 更安全。使用“生日攻击法”来伪造签名，要找到两个不同明文，但有相同摘要值的复杂度，在 MD5 中要 264 次运算，但在 SHA 中要 280 次运算。

(2) 速度：以 32 位处理器为基础，SHA 运算步骤比 MD5 多了 16 个步骤，而且 SHA 记录单元的长度较 MD5 多了 32 位，因此若以硬件来实现 SHA，其速度大约较 MD5 慢了 25%。

(3) 简易性：两种方法都相当简单，在实现上不需要很复杂的程序或是大量的存储空间，但是从总体上来讲，SHA 对每一步骤的操作描述较 MD5 简单。

因此综合各方面的考虑，在本文数字签名的实现中采用 SHA 算法作为单向散列函数，将明文经过 SHA 算法变换形成固定长度的消息摘要，然后采用 DSA 算法计算此消息的签名。计算接收消息的 Hash 值，并将该 Hash 值与给定的签名证实值相比较进行验证。

SHA 算法描述如下：

以五个 32bit 变量(A,B,C,D,E)作为初始值，消息经填充成 512bit 的整数倍。每组有 16 个 32bit 字。每送入 512bit，就进行四轮迭代，每轮完成 20 个运算，每个运算对 A,B,C,D,E 中三个进行非线性运算，而后作移位运算。

3. RIPEMD-160算法[33,34]也采用MD5算法的总体结构，允许任意长度的报文输入，输出160比特的报文摘要，算法中报文分组长度也为512位。RIPEMD-160的算法步骤如下。

报文填充和附加长度值与 MD5 相同。

初始化消息摘要(MD)缓存器

RIPEMD-160 使用 160 位的缓存来存放算法的中间结果和最终的散列值。这个缓存由 5 个 32 位的寄存器 A，B，C，D，E 构成。寄存器的初始值如下。

A0=67452301,B0=EFCDAB89,Co=98BADCFE,Do=10325476,E0=C3D2E1F0

数据存储时采用低位字节存放在低地址上的形式。

(1) 处理报文分组序列

处理算法的核心是一个10个循环的压缩函数模块，其中每个循环由16个处理步骤组成。在每个循环中使用不同的原始逻辑函数，算法的处理分为两条独立的

路线，分别以相反的顺序使用5个原始逻辑函数。每一循环以当前比特分组和160比特的缓存值A，B，C，D，E为输入，更新缓存的内容。每个循环使用一个额外的常数值K。在最后一个循环结束后，两条路线的计算结果 A,B,C,D,E ,和 A',B',C',D',E' ,以及链接变量的初始值经过一次相加运算产生最终的输出。

(2) 输出

对所有 L 个 512 位的分组处理完成后，第 L 阶段产生的输出就是 160 位的报文摘要。

(3) RIPEMD-160 的安全性

由于 RIPEMD-160 算法的核心是一个 10 个循环的压缩函数模块，其中每个循环由 16 个处理步骤组成，算法的处理分为两条独立的路线，分别以相反的顺序使用 5 个原始逻辑函数等特点，对于随机选取的报文，即使它们绝大部分信息相同，也很难产生相同的散列码，因此 RIPEMD-160 是很安全的。分析找出具有相同散列码的两个报文需要 $O(2^{80})$ 次操作，而要找到匹配特定散列码的报文需要 $O(2^{160})$ 次的操作。

3.2.2 平衡度

从另外的角度而言，众所周知，Hash 函数的密码学特性包括以下 3 个：不可逆性、抗二次碰撞性、抗碰撞性，这 3 个特性的强度是依次增高的，但却是相互独立的。由于 hash 函数主要用于数字签名和消息验证码，所以现实中对 hash 函数的密码分析主要集中于 hash 函数的抗碰撞性。而生日攻击是寻找 hash 函数碰撞的通用攻击方法。

生日攻击的概念如下所述:

设hash 函数为 $f:X\to Y$,从 X 中任意选取 q 个消息 $x_1,x_2,\cdots,x_q$,计算 $y_i=f(x_i),\ i=1,2,\cdots,q$ 。

如果存在一个碰撞,即存在一对 $i,j,i\neq j$,但 $y_i=y_j$,则表明攻击成功。其中, q 是攻击所选取的消息个数,用来衡量攻击的复杂度。

通常认为,根据生日攻击寻找hash 函数的碰撞对的复杂度大约是 $2^{r/2}$,其中 r 为hash 函数输出结果的比特长度. 比如对于输出长度为 m 比特的hash 函数,寻找其碰撞的复杂度是 $2^{m/2}$ 。这个复杂度是设计hash 函数时选择输出长度的主要依据,即设计的hash 函数的输出长度要足够长,使得生日攻击在计算上是不可行

的。但这个生日攻击的复杂度只是一个大略的估计,并不精确。因为这个复杂度是在假定hash 函数输出是均匀分布的前提下得到的,即在hash 函数值域的每一个值都有相同数目的原象,但事实并非如此。

Bellare等人对这一问题进行了探讨,提出了hash 函数输出分布的衡量指标——平衡度[35]的概念,并将生日攻击的复杂度表示为平衡度的简单函数。

Hash 函数的平衡度定义为:

设hash函数$h: D \to R$的值域中有$r(>2)$各点$R_1, R_2, \cdots, R_r$.对于$i = 1,2,\cdots,r$,设$h^{-1}(R_i)$为R_i的原象集,令$d_i = |h^{-1}(R_i)|, d = |D|$;称$\mu(h) = \log_r \left[\frac{d^2}{d_1^2 + \cdots + d_r^2} \right]$为函数$h$的平衡度。

由定义可知,平衡度是0~1之间的一个实数;如果对于任意i, j,有$d_i = d_j$,即hash函数是完全均匀分布时,平衡度取最大值1。当$d_i = 1, d_j = 0 (j \neq i)$,即hash 函数为常数函数时,平衡度取最小值0。

我们更为感兴趣的是函数的平衡度对生日攻击的复杂度的影响,对于$Q = r^{\mu(h)/2}$,Q设为攻击成功所需要的试验次数。这个等式说明,生日攻击的复杂度可以精确地表达为平衡度的函数,从这个等式我们可以看出:当$\mu(h) = 1$时,$Q = r^{1/2}$,即通常认为的生日攻击的复杂度. 当$\mu(h) = 0$时,$Q = 1$,即当函数为常数函数时,只需要一次试验即可找到碰撞. 当$\mu(h) = 1/2$时,$Q = r^{1/4}$,其复杂度远低于$r^{1/2}$。由此我们看到,hash 函数的平衡度对生日攻击复杂度的影响是不可忽视的。特别的,当hash 函数的平衡度很低时,攻击的复杂度将远远低于设计的期望。

因此hash 函数的平衡度应该作为hash 函数的一个重要的设计标准,即为了达到hash 函数预想的安全性,应该保证其有较高的平衡度。

此前,我们了解了Hash函数的安全现状,以及各种经典Hash的基本构造方式及其特点,并引入平衡度度量。回顾前面,我们在进行信息加解密中应用AES方案。

那么,我们知道Hash函数根据其构造方法大致分为三类:一类是精巧的Hash算法,如MD5和SHA-1;二类是基于分组密码算法的Hash函数,如MDC-2和MDC-4;三类是基于模运算的Hash函数,如基于RSA的Hash算法。其中第三类

由于速度太慢，在实际中基本上不适用。第一类的优点是速度较快，但无法正式证明其安全性。而其中研究最早的是第二类的基于分组密码的Hash函数，它们中许多均基于DES，由于DES的输出为64比特，因此这些Hash函数的输出长度为64比特或128比特。但由于计算机速度的日益提高，目前基本上认为输出长度为128比特的Hash函数已不能满足安全性的要求，必须设计出输出长度至少为160比特的Hash函数。例如近来提出的SHA-256和SHA-512，它们的输出长度即分别为256比特和512比特。

而在SHA安全性受到前所未有冲击，SHA-0被破解，SHA-1面临着被最终废弃的命运时，我们考虑，能不能将AES和SHA进行结合，尝试构造安全性能满足现实需要的Hash函数呢？

在数据加解密中采用AES算法，同时考虑到试图增强或提出个改进的有较高安全性的hash函数，可否利用AES的分组特性，参照以往hash函数，如MD4、MD5，以及SHA-1等，将平衡度纳入考虑，借此，从平衡度的角度，设计或提高抗攻击能力，SHA_AES的安全性，从而使数据安全，并将此增强hash，应用到新的签名方案中去，即采用AES和SHA结合技术并从平衡度角度增强Hash函数安全。

这里回顾常见Hash函数的基本算法步骤是，1、填充，2、附加长度值，3、初始化消息摘要(MD)缓冲器，4、处理报文，5、报文输出。

那么在先前的Hash方法中利用具备高安全性的分组密码，亦即利用有着非常坚实和成熟的理论作为基础的AES，这样能够从理论上证明所设计的分组密码具有哪些方面的安全性，从而能抵抗各种已知攻击。

那么从以下几个步骤来考虑如何提高hash函数的安全，

从安全性及运算效率的角度来说，一个好的哈希算法应具备以下特性：

1．对任意长度的消息能够计算出一个定长的且唯一的消息摘要；

2．由一个已知的摘要不能反推出产生该摘要的消息；

3．要找到两个具有相同摘要的不同消息在计算上是不可行的，尽管理论上是存在的；

4．能够抗弱冲突，也能抗强冲突；

5．具有较快的运算速度。

从分组长度、散列值的计算方式、逻辑函数的表达式和压缩函数逻辑结构4个方面对SHA-1进行改进。

3.2.3 AES 对于杂凑函数的平衡度

AES 算法即 Rijndael 算法的加密过程由以下四个变换组成[36]：

1．ByteSub 操作

ByteSub 操作是按字节进行的代替变换，也称为 S 盒子变换。它是作用在状态中每个字节上的一种非线性字节变换。S 盒子变换如下进行：

(1) 把字节的值用它的乘法逆代替，其中‘00’的逆就是它自己。

(2) 处理后的字节值进行下式定义的仿射变换。值得注意的是：

① 所加的常数列向量中含有 1，而模 2 加 1 运算等于非运算，所以它是非线性变换；

② 矩阵中每列都含有 5 个 1，说明改变输入的任意一位，将影响输出中的 5 位变化；

③ 矩阵中每行都含有 5 个 1，这说明输出中的每一位，都与输入中的 5 位相关。

$$\begin{bmatrix} y0 \\ y1 \\ y2 \\ y3 \\ y4 \\ y5 \\ y6 \\ y7 \end{bmatrix} = \begin{bmatrix} 10001111 \\ 11000111 \\ 11100011 \\ 11110001 \\ 11111000 \\ 01111100 \\ 00111110 \\ 00011111 \end{bmatrix} \begin{bmatrix} x0 \\ x1 \\ x2 \\ x3 \\ x4 \\ x5 \\ x6 \\ x7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

图 3-6

2．ShiftRow 操作

ShiftRow 操作是将状态行循环移位，0 行不移，第一行移 C1 字节，第二行移 C2 字节，第三行移 C3 字节，移位量 C1，C2，C3 与数据块长度 Nb 有关，如表 3-2 所示。

表 3-2 移位量与数据块长度的关系

Nb	C1	C2	C3
4	1	2	3
6	1	2	3
8	1	3	4

3．Mixcolumn 操作

Mixcolumn 操作是将状态列看作域 GF(2^8)中的多项式与一固定多项式 c(x)相乘，再模上 x^4+1，其中的 c(x)为：

$$c(x)='03'x^3+'01'x^2+'01'x+'02'$$

4．AddRoundKey 操作

在这个操作中，轮密钥被简单的异或，轮密钥根据密钥表得到，其长度为数据块的长度 Nb。

3.2.4 AES 算法与 SHA 算法结合

假定，消息m，分为若干长度为128 比特的明文块m_i（如果整个消息的长度不是128 比特的整数倍，则对消息进行填充，填充最高位为1，其余各位均为0），k 为密钥，长度也选取为128 比特。

而k 为 $k_1 \| k_2$，其中的k_1和k_2均是64 比特。

第一步：初始化，i从1开始，k_1作为H_0的初始值，k_2作为HA_0的初始值。

第二步：

$$R_i \leftarrow AES(m_i, H_{i-1} \oplus me_i) \oplus me_i,$$

$$RE_i \leftarrow AES(m_i, HA_{i-1} \oplus m_i \oplus R_i) \oplus m_i,$$

$$H_i \leftarrow H_{i-1} \oplus HA_{i-1} \oplus RE_i \oplus R_i,$$

$$HA_i \leftarrow H_{i-1} \oplus HA_{i-1} \oplus R_i,$$

$$i = i+1;$$

第三步：若i ≤ 1，则转第二步，否则作

$$H(k,m) \leftarrow H_h \| HA_h.$$

这里“‖”符号表示连接操作，用于把H_h与HA_h连接起来。

此仅为函数的抽象描述，具体实现通过AES算法结合SHA算法完成。

3.2.5 引入平衡度后的哈希函数的各项性能分析

1．散列性

已知有三条相似的明文信息，对它们求取散列值来观察该哈希函数的散列性。

Plaintext1 =“Test of Balance theory in hash function.”

Plaintext2 =“Test of Balance theory in hash function!”

Plaintext3 ="Test of Balance theory in hash function?"

采用SHA-1算法的散列结果分别为：

Hash1 = 7c154f708b6245a8d230810803eade9738c9aa21

Hash2 = a0289c23bb07307f5272fa17f7405c7f2c7cea12

Hash3 = 52c143187d779c6ab26008609dea6b9fc4f57359

采用SHA-256算法的散列结果分别为：

Hash1 =
b2f31af872e2a5c9a4035350b39c76b198acecf0c651dd91807f532085071f22

Hash2 =
74d5775c17f2807cd059784c6cc350ee76b9f7b334e4f82dfb251bb03a64f84b

Hash3 =
43020d59778259ab1e33a5dec9ef6c32329ed5320a132b2dd5047d10d6cd44fc

采用SHA_AES算法的散列结果分别为：

Hash1 =
5b26faeed8f2f6fa302ead80e5b53e2ff3f72b2d358d573605083d846da97980

Hash2 =
08a992e3c0c74a05cb9706da3171f94db825fe6744bedd0e5f0abde0b962c235

Hash3 =
80001a66465b64f00aa0b0f6e547a3d2d4a4a1556c54765c32abc82dcbf419e2

由明文信息和其相应的散列结果可以看出，明文的任何细微变化都使散列结果发生很大的变化，由此可得该哈希函数具有很好的初值敏感性和单向散列性。

2. 混乱和扩散特性

混乱和扩散是两种隐蔽明文消息中的冗余度的基本技术，是衡量Hash 性能的两个重要指标。

Hash 函数要尽量做到明文与其对应的散列结果不相干，这就要求明文的任何细微变化散列结果都以接近50% 的概率发生变化。这里用平均变化比特数和平均变化概率来衡量该散列函数的混乱和扩散特性。

定义平均变化比特数（明文1比特变化的情况下，散列结果变化的比特数）：

$$\overline{A}=\frac{1}{N}\sum_{n=1}^{W}A_n$$

平均变化概率：

$$P=(\overline{A}/256)\times 100\%$$

其中W为统计次数，An为第n(其中n = 1～W)次测试时结果的变化比特数。随机选取一段明文计算出其散列结果h1，然后改变明文中任意1 比特得出散列结果h'，对h1 和h'进行比较，不同的比特位数即为An。选取W = 768，计算出每次的An。由此求得该算法的平均变化比特数和平均变化概率分别为127. 44 比特和49.78%，非常接近理想状况下的128 比特和50%。这说明明文的任何细微变化，密文从统计结果上看都是均匀分布的，因此可以有效的抵御已知明文攻击和选择明文攻击。

3．抗冲突性

所谓冲突，是指不同的初值Hash 结果却相同，即发生了多对一映射。如果在256 比特的应用尺度上来进行碰撞分析，计算量过于庞大而不现实。现选取下面的算法来衡量本文构造的哈希函数的抗冲突性。由于该算法的设计与结果长度无关，可截取任意长度的一段文本来进行冲突程度的定量分析。

取一段长度为一个字节的初始文本，其对应的ASCII 值为0-255，其散列的结果也取为一个字节，即也为0-255。记初值空间为Ri，终值空间为Rf，终值空间任一值对应初值空间中原像的个数为K，终值空间中具有K个原像的点的个数为n(k)，n(1)越大，n(0)和n(k)(k>1)越小，说明该函数的碰撞越少，散乱能力越强。定义L 的值如下：

$$L=\frac{256-n(0)}{256}$$

所以，可以看到，当L 的值越接近于1，说明该算法的碰撞程度越低，当L 的值等于1 时，则完全没有碰撞。对散列结果进行统计可得：n(0) = 98，n(1) = 95，n(2) = 46，n(3) = 20，n(4) = 5，n(5) = 3，n(6) = 1，n(7) = 0，n(k) = 0 (k > 7，为整数)，通过计算得L = (256 – 98) / 256 = 0. 617。可见该算法的碰撞程度还是很低的。

4．执行速度

AES算法作为新一代的高级数据加密标准，将安全、高效、实用和灵活性于

一体，在不同的硬件和软件条件下表现出始终如一的良好性能，并且AES算法内部循环结构使得它有益于并行实现，Mixcolumn变换和Bytesub变换可先造表再查表，这样可以进一步提高该散列函数的执行速度。该哈希函数已经在P4 机上用软件加以实现，加密速度高达85Mb/s，完全能够满足哈希函数的速度要求。

目前大多数域扩展都采用增强的MD方法，R. Merkle和I. Damgard证明如果初始值是固定的，且填充内容包含原始消息的长度信息，那么，如果压缩函数是抗冲突攻击的，对应的Hash函数也是抗冲突攻击的，从不可辨性(indifferentiable)角度可以证明了它并不能保持模型的随机性。同时作者证明：对前缀固定(prefix-free)的消息，使用MD方法构建出来的Hash函数与随机模型是不可辨的。在构建中，通过和fz在每块分组的头一位分别填充0、0、0、1和1、1、1、0，来保证消息前缀固定。

5. 双倍长度的Hash函数及其安全性

在设计过程中，同时注意了相关方面的情况的，比如有人提出的双倍长度的hash函数等，采用参照Whirlpool[37]，在分析近来提出的新的攻击方法的基础上构建了一个基于AES的双倍长度Hash函数DH，它产生512位散列值，这种方法可以在后续研究中进行更进一步的讨论。

3.3 Secure E-mail 的信息加解密方案

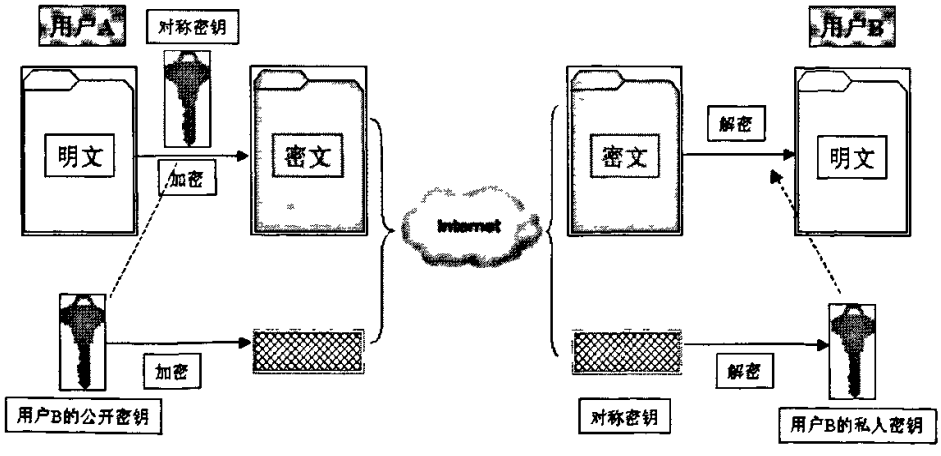


图 3-7 数据加解密

利用.NET 平台，将 AES 算法和 SHA 算法等相关加解密操作进行封装，设计成 cryptolib,以静态链接库的形式，进行代码重用；将网络通信部分，如 smtp 和 imap，以及 auth 分别进行功能实现，并模块化，生成动态链接库，底层 socket 操作利用.net 的 socket 类进行实现。

信息加解密采用 AES 进行，代码如下：

```
struct Rijndael_Info : public FixedBlockSize<16>, public VariableKeyLength<16, 16,
32, 8>
{
        CRYPTOPP_DLL static const char * CRYPTOPP_API
StaticAlgorithmName() {return "Rijndael";}
};//构造基本的安全函数 AES 信息结构体
class CRYPTOPP_DLL Rijndael : public Rijndael_Info, public
BlockCipherDocumentation
{//类定义
        class CRYPTOPP_DLL CRYPTOPP_NO_VTABLE Base : public
BlockCipherImpl<Rijndael_Info>
        {
        public:
                void UncheckedSetKey(CipherDir direction, const byte *userKey,
unsigned int length);

        protected:
                static const word32 Te0[256];
                static const word32 Te1[256];
                static const word32 Te2[256];
                static const word32 Te3[256];
                static const word32 Te4[256];

                static const word32 Td0[256];
                static const word32 Td1[256];
                static const word32 Td2[256];
                static const word32 Td3[256];
                static const word32 Td4[256];

                static const word32 rcon[];

                unsigned int m_rounds;
                SecBlock<word32> m_key;
        };

        class CRYPTOPP_DLL CRYPTOPP_NO_VTABLE Enc : public Base
        {
        public:
                void ProcessAndXorBlock(const byte *inBlock, const byte
*xorBlock, byte *outBlock) const;
        };
```

```
        class CRYPTOPP_DLL CRYPTOPP_NO_VTABLE Dec : public Base
        {
        public:
                void ProcessAndXorBlock(const byte *inBlock, const byte
*xorBlock, byte *outBlock) const;
        };

public:
        typedef BlockCipherFinal<ENCRYPTION, Enc> Encryption;
        typedef BlockCipherFinal<DECRYPTION, Dec> Decryption;
};

typedef Rijndael::Encryption RijndaelEncryption;
typedef Rijndael::Decryption RijndaelDecryption;
```

其具体实现包括如下基本函数块：

加密函数

AES定义中只包含一些固定的数据，其加密的功能由AESEncryption来完成。

解密函数

AES定义中只包含一些固定的数据，其解密的功能由AESDecryption来完成。

AES核心部分-加密算法块

AES核心部分-解密算法块

3.4 安全电子邮件的系统结构

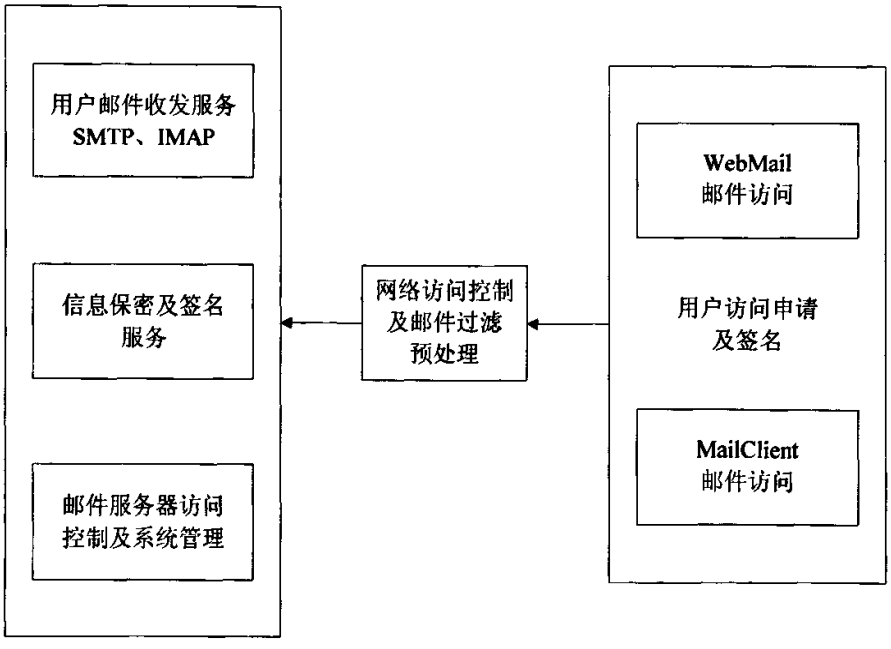


图3-8 安全的电子邮件系统结构

3.5 本章小结

通过前续内容的准备，结合网络通信特点及网络安全现状，和电子邮件现实使用和安全性需求，分别做了以下工作：

1、提出了一种新的有代理的多重签名方案，并根据电子邮件越来越多的参与到电子商务以至共同决策领域的要求，应用到本电子邮件系统中。

2、考虑到杂凑函数抗碰撞能力目前所受到的前所未有的冲击，我们分析各基本杂凑算法后，通过引入平衡度的概念，以构造一种比较好的结合型杂凑函数，从而满足系统设计中对杂凑函数安全性的要求。

3、信息加解密方案，以实用、可重用为基本设计理念，通过构建加解密通用组件和网络操作通用组件的方式，利用工程化设计的方法，进行系统的设计与实现。

第四章 安全电子邮件系统 Secure E-mail 的实现

以Microsoft Visual Studio .NET 2003为平台，进行开发设计，系统使用IIS作为网络发布服务平台，以Lenovo台式机作为系统服务器，IE或Navigator作为客户端浏览器。

4.1 Secure E-mail 保密与签名

安全电子邮件基本体系由邮件服务器和邮件收发端组成，提供Web mail作为邮件系统的用户。邮件服务器包括系统邮件服务和邮件系统控制端，前者提供邮件发送、用户从服务器接收邮件以及提供包括签名在内的信息安全服务，后者提供对邮件服务器所处理业务进行管理。

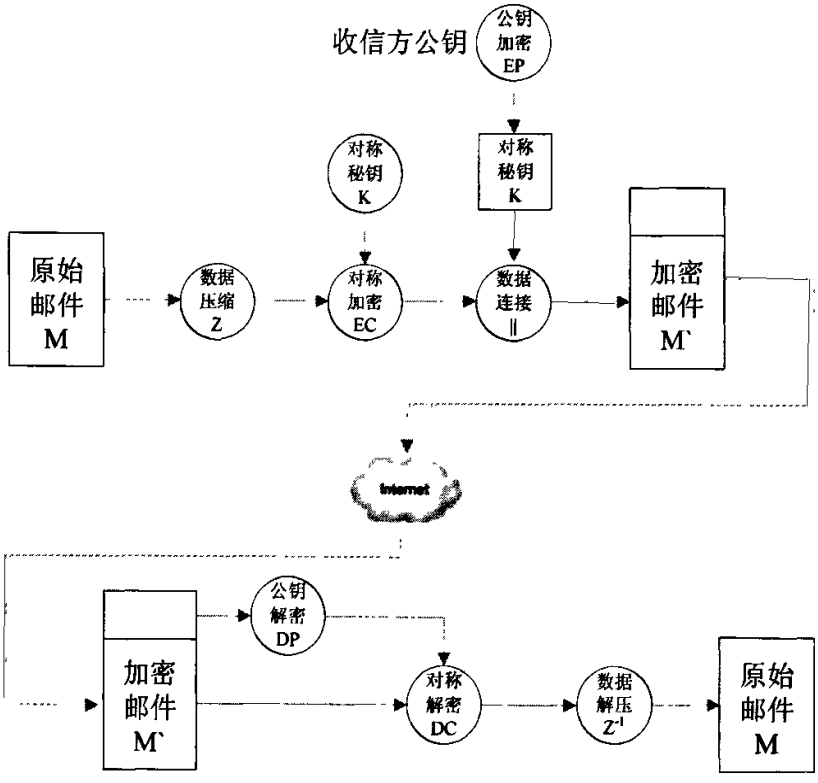


图4-1 邮件保密性

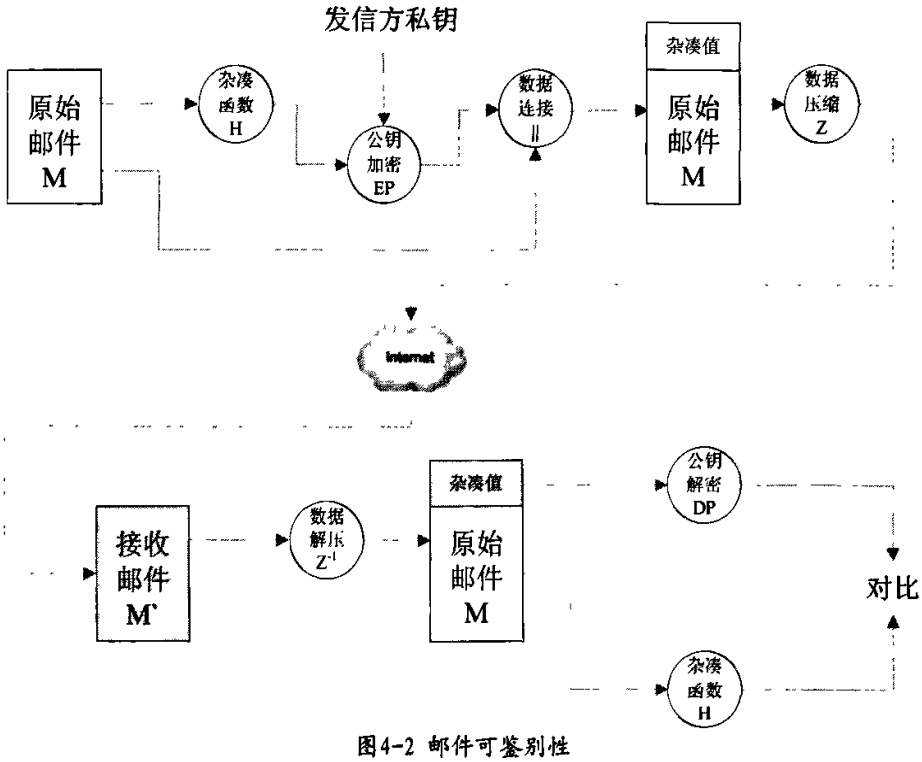


图4-2 邮件可鉴别性

符号说明见页底[1]。

如图4-1与图4-2分别来实现信息的保密性和可鉴别性，当我们需要在邮件通信时对双方进行特定身份的鉴别，那么首先发信方产生邮件M，然后利用SHA_AES对M生成一个160位的散列码H，再用发送者的私钥对H加密，并与M连接；而接收方则利用发送者的公钥解密并恢复散列码H，然后对邮件M生成一个新的散列码，与H比较。如果一致，则邮件M被鉴别。这里我们采用离散对数的强度保证了发送方的身份真实，而hash函数SHA_AES的强度保证了签名的有效性，当然，我们为方便操作处理将可以签名与邮件分离，这样对邮件进行单独的日志记录，对可执行程序的签名记录，进行检查，对于文档多方签名，可以避免嵌套签名。

当发送方需要发送的邮件信息安全到达对方，那么首先生成邮件M并为该邮件生成一个随机数作为会话密钥，然后用该会话密钥加密M，再将接收者的公钥加密会话密钥并与邮件M结合。而接收方用自己的私钥解密恢复会话密钥，用会话密钥解密恢复邮件M来。由于我们采用AES进行加密，并对一次性密钥，单向分发，结合公钥算法，进行保护。

[1] Ks: session key; KRa: 用户A的私钥; KUa: 用户A的公钥; EP: 公钥加密; DP: 公钥解密; EC: 常规加密; DC: 常规加密; H: 散列函数; ||: 连接; Z: 用VCLZIP算法数据压缩; R64: 用radix64转换到ASCII格式

1．对称加密算法和公钥加密算法的结合可以缩短加密时间。

2．用公钥算法解决了会话密钥的分配问题，因为不需要专门的会话密钥交换协议，而且由于邮件系统的存储-转发的特性，用握手方式交换密钥不太可能。

3．每个邮件都有自己的一次性密钥，进一步增强了保密强度。所以，每个密钥只加密很小部分的明文内容。

4．公开密钥算法的长度决定安全性。

实际的邮件应用中，保密与可鉴别这两种安全服务都是需要的，发信者先用自己的私钥签名，然后用会话密钥加密，再用收信者的公钥加密会话密钥。

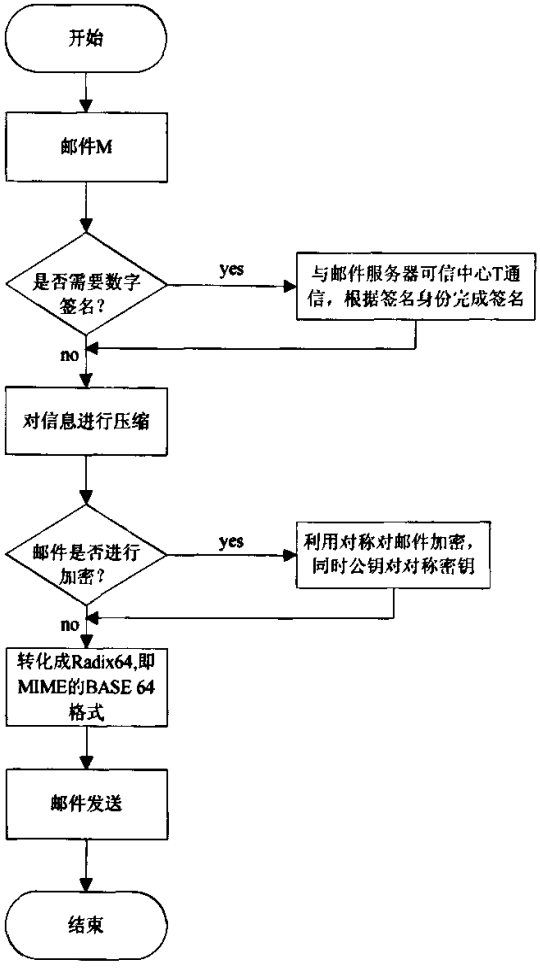


图4-3　邮件消息发送

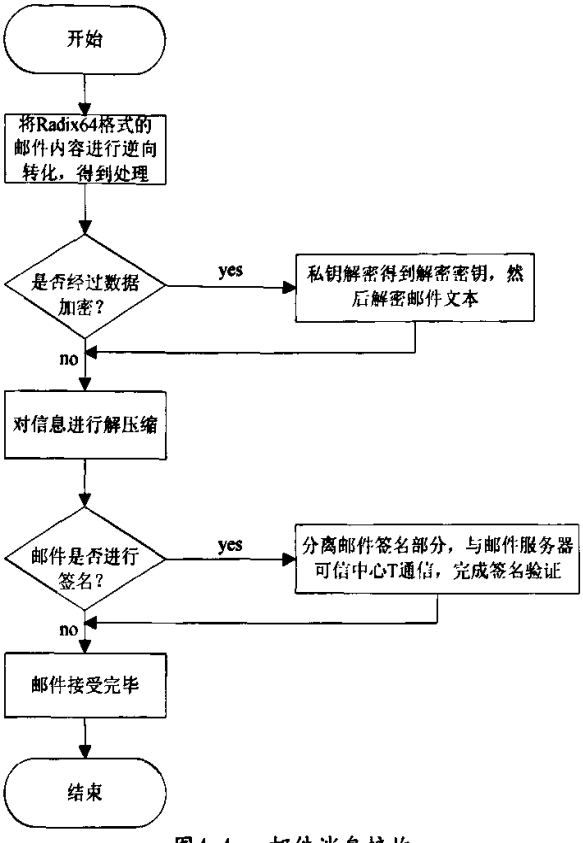


图4-4　邮件消息接收

1．发信方处理邮件的过程是，针对要生成的签名信息，利用UserID作为索引，并根据提示输入口令短语，从私钥库中得到私钥，根据其身份，即为原始签名人或代理签名人，依据方案规则，经过与邮件服务器可信中心T通信，构造其个人签名部分。而对于加密，产生一个会话密钥，利用对称算法AES加密邮件，并根据邮件接收者的UserID从公钥库中获取其对应公钥，用此公钥加密用于邮件加密的对称密钥,并构造邮件的会话密钥部分。

2．收信方处理邮件的过程是，解密邮件前，先用邮件的会话密钥部分中的KeyID作为索引，同样结合提示输入口令短语从私钥库中获取私钥，并解密出用于解密邮件的对称密钥，获得解密邮件。而对于附带的验证部分，用邮件的签名部分中的KeyID作为索引，从公钥库中获取发送者的公钥恢复被传输过来的消息摘要，对于接收到的摘要进行暂存，这时，有两情况：如果接收方为签名最终获取者，那么当其获得所有N个签名人(当然，其中有N'个原始签名人和N－N'个代理签名人)的签名信息后，即可生成最终签名；如果接收方为签名人之一，那么其接收到其余N－1人的信息后，生成自己的个人签名部分，然后发送给最终签名接收人，依据方案规则完成签名过程。

系统的公钥管理中，由于本安全电子邮件系统主要是为了研究和设计实现系

统安全性因素，并不重在广泛地在正式环境下应用，没有建立严格的公钥管理模式。

要是采用PGP[38,39]的模式，那么如果A的公钥环上有一个从BBS上获得B发布的公钥，但已被C替换，这是就存在两条通道。

C可以向A发信并冒充B的签名，A以为是来自B；A与B的任何加密邮件C都可以读取。

因此，我们利用邮件服务器可信中心的存在，采用公钥库，利用用户的UserID来进行公钥对应，并结合提示输入口令短语，来验证，这样即使有公钥更新，也保证其安全使用，为了防止用户公钥在库中上包含错误的公钥，有比较多的方法可用于降低风险，比如通过特性物理上得到公钥，或者在各方都信任的实体，比如可信中心处，利用公开算法获得公钥等。

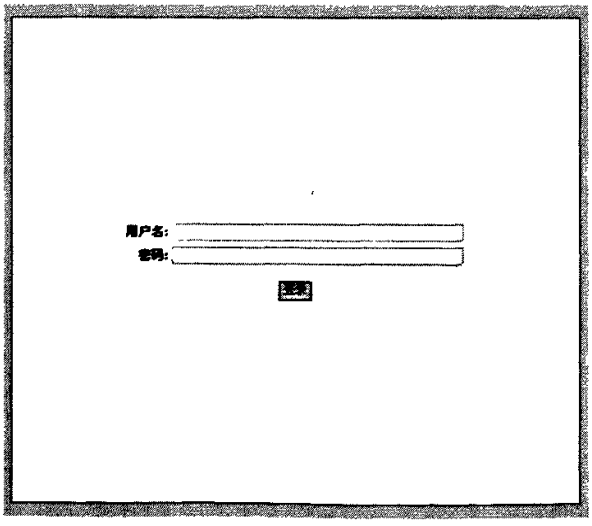
4.2 邮件服务器系统密钥管理

采用集中式系统密钥管理模式，即邮件服务器所提供的所有系统服务，包括四个部分，系统安全邮件服务、邮件服务管理器、邮件WEB收发和安全邮件处理客户端，此四部分构成了邮件系统的应用和安全体系。

4.3 个人密钥管理

4.3.1 账户管理

用户个人帐户，采用服务器本地XML存储，进行申请后，获取个人邮件地址，成为系统用户。



4.3.2 产生密钥对

1．安全电子邮件系统的口令系统

口令采用注册密码加密存储，登录时口令密文对比，防止明文密码的网络泄露，从而避免了安全隐患，登录入口采用随机字符的验证登录，避免强力攻击中机器重试对用户密码的威胁。

2．产生密钥的过程

在申请成为安全电子邮件系统用户后，用户将被要求登记个人与系统间安全协议所要求的用户密钥，如有应用需要，同时登记为某团体的授权安全成员，并提供其对应代理人的安全信息，包括在行使特定商务信函签名时的代理人密钥信息。

3．程序具体实现

参照上文所提签名方案中的细节部分。

4.3.3 分发公钥

公钥的分发，为了实现方便的获取，通过用户登录询问方式获得，并没有采用公告牌形式，为的是以注册用户为服务对象，将访问控制在一个可控范围之内，当然，根据系统规模和需求的变化，可以动态的改变公钥分发的方式。

4.3.4 修改口令

用户口令为用户登录系统基本安全控制，采用AES加密，存储密文，修改密

码是首先验证密码，并以注册安全问题进行再次确认，并登记口令修改事务，从而对用户口令变更进行管理。

4.3.5 备份和恢复密钥

密钥的备份和恢复对于系统的可用性和稳定性是非常重要的，因此，在用户密钥登记注册时便对所注册密钥进行交叉存储控制，并同步备份。

4.4 地址簿

用户地址统一存储在服务器XML地址数据中，利用XML便捷的使用方式进行管理，按标签进行特定信息存取，格式如下：

```
<Users>
  <UserID>d8985f9b-b2d6-41db-bcec-e1f8096d7b25</UserID>
  <FullName>napoleon</FullName>
  <UserName>napoleon</UserName>
  <Password>napoleon</Password>
  <Description/>
  <DomainName>server1.securemail.com</DomainName>
  <Mailbox_Size>20</Mailbox_Size>
  <Enabled>true</Enabled>
  <AllowRelay>true</AllowRelay>
</Users>
```

4.5 发送邮件

4.5.1 邮件编辑器

邮件编辑采用通用文本编辑，支持 MIME，即多媒体形式，但出于对课题各方面因素考虑，本原型系统只对文本信息进行处理，在系统的应用中，可以通过扩展 UI 接口，来扩展系统处理功能。

4.5.2 签名邮件生成并格式化

```
private string          m_MsgID      = "";
private string[]        m_To          = null;
private string[]        m_Cc          = null;
private string[]        m_Bcc         = null;
private string          m_From        = "";
private string          m_DSN         = "";
private string          m_Subject     = "";
private string          m_Body        = "";
private string          m_BodyHtml    = "";
private string          m_CharSet     = "";
private DateTime        m_MsgDate;
private Attachments     m_pAttachments = null;

public MimeConstructor()
{
m_pAttachments = new Attachments();

m_MsgDate = DateTime.Now;
m_MsgID = "<" + Guid.NewGuid().ToString().Replace("-","") + ">";
m_CharSet = "utf-8";
}
```

4.5.3 SMTP 协议的实现

```
//Webmail 邮件发送
private void SendMail()

//调用的 send 函数，在网络组件函数库中的申明如下
public bool Send(string[] to,string from,Stream message)
```

```
//Send 函数通过调用组件函数中的 SendMessageToServer 来实现邮件发送功能，即 SMTP implementation 实现
private bool SendMessageToServer(string[] to,string reverse_path,Stream message)
```

4.6 接收邮件

参见前一章解密方案及所引入的签名接收处理方案。

```
clnt.Connect(Application["IMAPServerName"].ToString(),Convert.ToInt32(Application["IMAPServerPort"]));
//获取 IMAP 服务器及其端口信息并连接
clnt.Authenticate(Session["Name"].ToString(),Session["Password"].ToString());
//提取用户名及密码，进行身份确认
clnt.SelectFolder(Request.Params["Folder"].ToString());
//选择文件夹
if(Request.Params["MessageID"] != null)
{
IMAP_FetchItem[] f = clnt.FetchMessages(
                        Convert.ToInt32(Request.Params["MessageID"]),
                        Convert.ToInt32(Request.Params["MessageID"]),
                        true,false,true);
if(f.Length > 0)
{
        p = new MimeParser(f[0].Data);//解析 MIME 格式邮件内容
}
Else
{
        m_ErrorText = m_WTxt["101"] + "!"; //邮件消息不存在
}
}
```

4.7 处理邮件及签名

4.7.1 邮件存储格式

以 MIME 格式进行邮件的存储，通过数据安全封装包，采用 AES 算法对数据进行处理，签名在多方进行处理后，邮件服务器通过本文采用的有代理的多重签名方案，进行并行验证。

4.7.2 邮件处理的实现

1．普通邮件将直接解析，以加密文件形式存储于邮件服务器，用户通过安

全认证后访问邮件，提取信息，解密阅读。

2．签名邮件，将对邮件主体和文件附加签名段分别进行处理，对于邮件主体按普通邮件进行处理实现，同时，将附件文件签名段交由邮件服务器进行签名提取与解析，若通过服务器确认，那么在数据库进行此签名登记，在获取完备签名集后，生成结果签名，并备案。

4.8 本章小结

通过前面部分对系统安全和架构的分析理解，将邮件系统基本功能、引入的新的数据签名方案、为获取比较好抗碰撞能力杂凑函数所进行的尝试，以及工程化设计的方法通过本安全的电子邮件系统的原型设计并实现，来更深入的理解网络数据通信与信息安全处理的若干环节。

第五章 工作总结与展望

5.1 工作总结

1．基于普通电子邮件特性及其功能的研究与设计；

对已有的电子邮件系统协议及相应的电子邮件系统，如PGP，PEM，S/MIME等进行分析，并对其各自特性，即优缺点进行全面和客观的评估。开发能提供基本安全的E-mail系统，实现基本的安全邮件的服务：

- 信息保密—— 保证只有希望的接收方能够阅读信息。
- 信息完整性—— 保证发出的信息与接收到的完全一样。
- 认证—— 保证信息的发起者不是冒名顶替的。
- 非否认—— 证实发送者确实发送了信息而不管他是否承认。

2．新的安全电子邮件在保密和认证方面的研究；

保密方面[13,16,22,24]，对称密码体制和公开密钥密码体制各有其优缺点，在实际应用中可将两种算法结合起来使用，以一种相对较优的组合来充分发挥两种算法的优势。

签名方面[15,17,18,20,21,26,29,30,31,32,33]，在这里提出一种新的有代理的多重签名方案。在这种方案中，多个原始签名人将分别对自己选定的可信任的代理签名人进行代理授权，在签名过程中，N个签名人，可根据实际情况，允许有N′个原始签名人和N－N′个代理签名人共同参与。在签名认证中，验证人能验证签名者身份，即签名为原始签名人还是其代理签名人所签。

对新的对电子邮件安全方面的要求，满足网络环境下多用户间实现如有代理的多重签名邮件的生成与认证等等，从而增强电子邮件系统的安全性能，提升其使用价值，满足社会发展过程中在电子邮件领域对安全性更高的要求；

3．对于杂凑函数的广泛应用和发展现状，从平衡度的角度，结合SHA和AES算法，对杂凑函数的安全性因素在一个不同的角度进行了讨论，以提高其在实际应用中的抗碰撞能力；

4．分析网络通信中信息保密技术，提出了安全电子邮件系统的数据加解密方案，从而满足电子邮件服务作为；

5．将研究成果应用于B/S模式和C/S模式或两者结合的系统之中；

通过将安全模块、网络操作模块分别进行独立的可重用设计，即以组件开发的形式实现；采用三层体系结构可以实现在多种开发模式系统中，利用安全的电子邮件系统进行可靠的信息通信。

5.2 工作展望

本文工作所涉及的内容，是在网络技术得到日益发展和安全技术越来越受到重视的情况下，围绕电子邮件安全这一网络服务而进行的，信息保密技术、数字签名技术以及信息杂凑技术等安全领域的研究主题，也正是伴随着信息技术的发展而不断发展的，因而还有待进一步完善，需要进一步研究的问题是：

1．效率：设计的保密方案和签名方案应尽量提高运算速度和执行效率。

2．安全性：设计一种安全性能高的有代理的多重签名尤其重要。数字签名的设计一般都是基于离散对数问题和因子分解理论，所设计的新算法的安全性在特定的情形，可能由于所在域的特定性质导致其安全特性发生变化，因而其安全性要在更广的范围内重新评估与改进；同样，杂凑理论所受到的冲击必定促使我们在将来寻求更为彻底和安全的设计思想和方案。

3．实用性：设计安全电子邮件系统应有比较好的实用性，并能适应安全性需求变更和升级情况下的良好适应能力。

4．产品化：进一步研究所涉及安全技术在电子邮件系统中的应用，使之产品化，真正具有商用价值。本文只实现了基本的安全电子邮件系统，要研究开发出成熟可产品化的安全电子邮件系统还需要不断地努力。

参考文献

[1] Jonathan B. Postel, Simple Mail Transfer Protocol, RFC821, August 1982.

[2] J. Myers, Post Office Protocol-Version 3, RFC1939, May 1996.

[3] S. Crocker, MIME Object Security Services, RFC1848, October1995.

[4] D. Atkins, PGP Message Exchange Formats, RFC1991, August 1996.

[5] PGP, Pretty Good Privacy, http://www.pgp.com.

[6] M. Elkins, MIME Security with Pretty Good Privacy (PGP), RFC2015, October 1996.

[7] N. Freed, Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies, RFC2045, November 1996.

[8] N. Freed, Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types, RFC2046, November 1996.

[9] K. Moore, MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text, RFC2047, November 1996.

[10] N. Freed, Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures, RFC2048, November 1996.

[11] N. Freed, Multipurpose Internet Mail Extensions (MIME) Part Five: Conformance Criteria and Examples, RFC 2049, November 1996.

[12] M. Elkins, MIME Security with Pretty Good Privacy (PGP), RFC2015, October 1996.

[13] B. Rams dell, S/MIME Version 3 Certificate Handling, RFC2632, June 1999.

[14] B. Rams dell, S/MIME Version 3 Message Specification, RFC2633, June 1999.

[15] M. Crispin, Internet Message Access Protocol-Version 4rev1, RFC2060, December 1996.

[16] Revist R L, Shamir A, Adleman L. A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 1978, 21(2):120-126.

[17] J. Hoffstein, J. Pipher, and J.H. Silverman. NTRU: a Ring based Public Key Cryptosystem. In Proc of ANTS III, LNCS 1423, Berlin: Springer-Verlag, 1998:267-288.

[18] Oppliger R, Rytz R. TCPA, Palladium, and NGSCB: Does trusted computing remedy the personal computer's security problems. IEEE Security & Privacy, to appear in 2004.

[19] J Damgard. A Design Principle for Hash Functions. In: Crypto 89. LNCS 435. 416-427.

[20] Proposed Federal Information Processing Standard for Digital Signature Standard (DDS). Federal Register, v.56, n.169, 30 Aug 1991:42980-42982.

[21] Atreya M 等著,贺军等译.数字签名.北京:清华大学出版社,2003.

[22] Boyd C. Digital multi-signature. Proceedings of conference on Coding and Cryptography, 1986:15-17.

[23] Chaum D, Heyst E. Group signatures. Advances in Cryptology-Eurocrypto'91, LNCS 547. Berlin: Springer-Verlag, 1991:257-265.

[24] Mitomi S, Miyaji A. A general model of multi-signature schemes with message flexibility, order flexibility, and order verifiability. IEICE Trans., Fundamentals.2001, E84-A (10):2488-2499.

[25] Mambo M, Usuda K, Okamoto E. Proxy signature Delegation of the power to sign messages. IEICE Trans on Fundamentals.1996, E79-A(9):1338-1354.

[26] 王育民,刘建伟.通信网的安全---理论与技术.西安电子科技大学出版社.2000.

[27] Itakura K, Nakamura K.A public-key cryptosystem suitable for digital multi-signature. NEC Research & Development, October 1983, (71):1-8.

[28] K.Ohta, T.Okamoto. A digital multi-signature scheme based on the Fiat-Shamir scheme. Advances in Cryptology-ASIACRYPT'91.

[29] 伊丽江,白国强,肖国镇. 代理多重签名:一类新的代理签名方案.电子学报,2001,29(4):569-570.

[30] Wang Xiaoyun, Feng Dengguo, Lai Xuejia, et al. Collisions for Hash Functions MD4, MD5, HAVAL-128 and RIPEMD. http://eprint.iacr.org/2004/199.pdf, 2005-01.

[31] Wang Xiaoyun, Yu Hongbo. How to Break MD5 and Other Hash Functions. EUROCRYPTO'05.2005.

[32] Bruce Schneier 著,吴世忠等译:应用密码学.北京:机械工业出版社,2000,1.

[33] 许崇祥.基于 Diffie-Hellman 算法的可否认认证协议.计算机工程,2002,28(10):145-146.

[34] 蔡文威.一种密钥管理协议的设计.计算机应用与软件, 2002,(12): 40-43.

[35] Stinson D R. Some observations on the theory of cryptographic hash functions. http ://eprint.iacr. org/2001/020.

[36] J Daemen,N Rijmen.The Rijindael Block Cipher. AES Proposal, Available at http://www.nist.gov/aes/.

[37] ISO/IEC 10118-3:2004, Dedicated hash functions.

[38] Oppliger, R. Secure Messaging with PGP and S/MIME. Artech House, Norwood,MA,2001.

[39] K.Yamamoto, An integration of PGP and MIME, In Proc of 1996 IEEE Symposium on Research in Network and Distributed System Security, 1996:17-24,.

[40] 施荣华,蔡立军.一种基于不同签名授权的组签名方案.小型微型计算机系统,2003,24(3):612-613.

[41] 施荣华.基于数字签名的安全认证存取控制方案.软件学报,2002,13(5):1003-1008.

[42] 伊丽江,白国强,肖国镇.代理多重签名:一类新的代理签名方案.电子学报,2001,29(4).

[43] 卢开澄.计算机密码学.第二版.北京.清华大学出版社.1998.

[44] 罗俊,施荣华.一种新的有代理的多重签名方案.计算机工程与应用,2006,42(34):137-138.

致 谢

时间总是短暂的，对于一个人而言。三年前，跨入中南大学，现在，却即将告别这里，走向另一个新的地方。在这些日子里，面对过问题，同时有过问题得以解决的快乐，让我更加自信；遇到过挫折，但最终找到方法克服困难的喜悦，让我收获坚强。在毕业论文完成之际，心中，怀着几分期待，几分激动，更怀着深深的谢意。

首先，要感谢我的导师，施荣华教授，三年来对我的悉心指导和谆谆教诲，学高为师，身正为范，他的渊博学识和严谨学风，为我折服，豁达的胸怀和求实的品格更让我受益匪浅。三年的研究生学习，在施老师的指导下，我的分析问题、解决问题的能力得到了很大的提高，这将为今后的工作和学习打下良好的基础，在此，我向施老师致敬！

感谢刘卫国、王国才、梁建武、肖大光和王果平老师在论文完成过程中给予的指导和建议，正是您们的帮助和鼓励，使我的论文能顺利完成。

感谢杨政宇、周诚、丁耀军、吴科桦、朱宁和所有的兄弟姐妹们，给予我的学习上的帮助，这些共同学习、共同进步的日子都将使我终生难忘。

感谢我慈爱的双亲，可爱的妹妹和亲戚朋友们，他们给了我巨大的支持和无微不至的关怀。他们的敦促、鼓励使我不断努力，勇往直前。

感谢所有给过我教诲的老师。

感谢所有帮助、鼓励和关心我的人。

最后，感谢在繁忙工作中抽出宝贵时间审阅本文的老师们，谢谢您们！

攻读学位期间主要的研究成果

一、发表论文情况：

罗俊,施荣华.一种新的有代理的多重签名方案.计算机工程与应用,2006,42(34):137~138

二、科研项目情况：

[1] 参与《科技部科技数据仓库和辅助决策信息服务系统》系统调研；

[2] 参与《广州港水运工程监理公司生产管理信息系统》项目，负责项目组织与管理；

[3] 参与《移动自组网中安全认证问题的研究》项目立项调研工作；

[4] 参与《中国铝业股份有限公司中州分公司设备管理系统》项目，独立完成多个功能模块的调研与开发；

[5] 参与《中国铝业股份有限公司河南分公司设备管理系统》项目，独立完成多个功能模块的调研与开发；

[6] 参与《长沙市雨花区地方国防动员管理信息系统》项目调研与系统分析。