摘 要

随着经济的发展和信息基础设施的不断完善，电子商务以其快捷、便利等优点得到了快速的发展。而电子商务中网上支付的安全性和通用性是目前制约电子商务发展的两大主要障碍。本文在研究分析网上支付系统和 USBKey 安全性能的基础上提出了一种基于 USBKey 的网上支付方案，该方案的基本架构是公钥基础设施（PKI）体系，采用基于 PKI 的安全支付协议，具有一定的安全性和通用性。并结合现有的各种网上支付工具给出了 USBKey 网上支付的实际应用方案。文中还对 USBKey 的攻击与防护进行了分析，提出了一种改进的用户 PIN 输入方案，进一步提高了 USBKey 应用系统的安全性。

关键词： **USBKey　电子商务　网上支付　公钥基础设施　安全**

Abstract

With the development of economic and continuous improvement of information infrastructure, E-commerce developed with high speed dues to the advantages of being swift and convenient. However, E-commerce online payment security and commonality is two major obstacles, which restrict the progress of electronic commerce currently. On base of study online payment system and USBKey, this thesis presents a new online payment scheme based on USBKey, the basic framework of this scheme is PKI system, which adopt secure payment protocols based on PKI, have some security and versatility. This thesis Combined with various existing online payment tool gives the practical application of online payment options of USBKey. This thesis also analyzes the attack and protection of USBKey, provides an improved user PIN input scheme to further improve the system's security.

Keywords: USBKey Electronic Commerce Online Payment PKI Security

目 录

第一章 绪论

1.1 研究的背景意义

作为全球使用范围最大的信息网，Internet 自身协议的开放性极大地方便了各种互联的设备，拓宽了资源的共享。因特网已经遍及世界各地，为亿万用户提供了多样化的网络与信息服务。由中国互联网络信息中心 CNNIC[1]的调查报告可知，截止到 2010 年 6 月我国网民规模已达到 4.2 亿，较 2009 年底增加 3600 万人。互联网普及率达到了 31.8%，与 2009 年底相比提高了 2.9 个百分点。手机网民用户达到了 2.77 亿，宽带网民数已达到 3.64 亿。更值得关注的是，互联网商务化程度正迅速提高，全国网络购物用户达到 1.4 亿，网上支付、网络购物和网上银行的半年用户增长均在 30%左右，远远超过其他的网络应用。随着信息技术的发展，以网络方式获得信息和交流信息已成为现代信息社会的一个重要的特征，网络正在逐步改变人们的工作和生活方式。

然而随着电子商务的大力发展，安全问题却日益突出。由 CNNIC 调查报告可知，仅 2010 年上半年，有 59.2%的网民曾遭遇过病毒或木马的攻击，近三成的网民账号或密码被黑客盗取过，电子商务网站访问者中有 89.2%的人担心假冒网站。其中，86.9%的人表示如果无法获得该网站进一步的确认信息，将会选择退出交易。由此看出，网络安全问题已经成为影响电子商务更深层次发展的最大制约因素，互联网向商务交易型应用的发展，急需建立更加可信、可靠的网络环境。

目前，网上支付的安全性已经成为用户的主要顾虑，安全问题已成为电子商务发展面临的最大障碍。

1.2 发展现状

1.2.1 USBKey 的发展现状

USBKey 产品最早是由加密锁厂商提出来的，原先的 USBKey 主要用来防止软件破解和复制，保护软件不被盗版。随着电子商务和 PKI（Public Key Infrastructure）应用的兴起，数字证书作为鉴别用户身份和保护用户数据的有效手段越来越受到人们的青睐。现在 USBKey 主要用于网络身份认证、保存数字证书和用户私钥等。由于用户数字证书和私钥保存在 USBKey 中，理论上使用任何方式都无法读取，因此保证了用户认证的安全性。

在美国、加拿大、欧洲等这些国家的 PKI 技术起步早，使用 USBKey 进行身份认证的技术相对更成熟。比如有使用生物特性识别技术，利用指纹、虹膜等取

代传统的PIN口令来获得对USBKey的使用权限，更加的安全和可靠。

在中国，2000年，由中国人民银行牵头，13家商业银行共同出资成立了中国最具权威的CA（Certificate Agency）机构中国金融认证机构（CFCA）。CFCA的建成成为中国PKI技术发展的推动力，大量的商业银行和银联组织等将在CFCA的PKI框架下实施电子商务和网络金融。由此可见，PKI和USBKey的应用潜力在金融领域前景广泛。

目前USBKey的应用主要分布在网上银行、电子口岸、网上报税及电子网络认证等领域。随着国家《电子签名法》[2]的实施，各地数字证书的应用快速展开。随着电子商务、电子政务的飞速发展，越来越多的CA和用户选择了USBKey作为他们的证书存储介质。

1.2.2 网上支付的发展现状

随着互联网的发展和普及，国外在20世纪80年代就开始了电子商务的研究与开发。现代密码技术的不断更新，使得电子商务，尤其是网上支付在安全算法和协议等方面已经十分成熟，网络通信技术的不断发展更给电子商务和电子支付提供了坚实的物质基础。目前国外网上支付系统已经进入实用化阶段。

网上支付方式主要有银行卡支付、电子钱包、电子支票、电子现金等。在我国，应用最为广泛的是银行卡网上支付。中国银行依托首都电子商务工程在国内率先推出了SET借记卡支付。目前，金融认证中心已经支持SET、NON-SET等多种支付活动。

在国内招商银行、中国银行首先开通了网上支付业务。由中国人民银行牵头成立的基于PKI机制的中国金融认证中心（CA中心）为中国电子商务的发展奠定了坚实的基础。目前金融CA下的网络银行支持的电子商务应用有B2B应用模式和B2C应用模式。

当前，电子商务支付模式的发展方向是综合化、集成化、安全化、通用化。而国内目前的电子支付主要是网络银行卡支付，多数企业的网上支付还是通过离线转账完成。目前通用的标准主要有SSL、SET以及基于PKI的安全支付协议。其中SSL协议是端的端的安全传输层协议，机制简单、便于应用，是现在网上支付系统主要方式，但是SSL协议尚不支持进行多方认证和数字签名，所以不能通用。SET协议的出现虽然克服了SSL协议的缺陷，但由于其过于复杂、成本较高，且一般只能支持拥有电子钱包客户使用，目前在支付领域中使用较少。随着PKI技术的发展，基于PKI的安全协议也在不断的普及，综合PKI技术的各种应用，可以建立一个可信任和足够安全的网络环境，在这里有可信的CA认证中心，典型的银行、政府或第三方机构，提供了身份认证、数据安全性、不可否认性、数据

完整性等功能，从而能保证信息的安全传输。目前 PKI 技术从电子商务到电子政务得到了广泛的应用，从发展趋势看，PKI 技术将会有很大的市场应用前景。

1.3 研究的内容及安排

论文分为六章，各章内容及安排如下：

第一章介绍了本文的研究背景、意义及发展现状。

第二章对介绍网上支付进行的相关基础知识，包括网上支付的概念、安全需求、支付工具、支付协议并分析了现有网上支付系统存在的问题。

第三章主要研究了目前网上支付系统中所应用到的安全技术，包括密码技术、PKI 技术、身份认证技术和数字签名技术。

第四章首先研究了 USBKey 体系结构，并对其攻击与防护进行逐一分析。最后结合 USBKey 的实际应用情况提出了一种改进的用户 PIN 输入方案。

第五章研究提出了基于 USBKey 网上支付方案，对其性能和安全性进行了分析，最后结合现有网上支付工具给出了其实际应用方案。

第六章对全文进行总结，分析了本文存在的不足和提出以后进一步的研究建议和对工作的展望。

第二章 网上支付系统

2.1 引言

随着互联网的日益普及，在电子商务中，网上支付系统是整个电子商务交易中的一个重要环节，同时也是电子商务准确性、安全性要求最高的业务过程。电子支付的资金流是一种业务过程，在进行网上支付活动的过程中，会涉及到很多安全技术问题，如交易双方的安全、保密等。网上支付是通过网络进行货币支付的，其实质是在网上把现有的支付结构转化为电子形式，从而提高了货币支付的效率，进而也避免了大量的纸张处理。

网上支付是指进行电子交易的当事人，包括消费者、厂商和金融机构，使用安全的网上支付手段通过网络进行货币支付或资金流转。网上支付又称在线支付，简单的说就是网上交易中的无纸化电子结算手段。从广义上说，网上支付就是资金或与资金有关的信息通过网络进行交换的行为。在普通的电子商务中就表现为消费者、商家、企业、中间结构和银行等通过互联网进行的资金流转，主要通过信用卡、电子支票、数字现金、智能卡等方式来实现[3][4]。

2.2 网上支付系统的构成

网上支付系统的基础设施是金融电子化网络，流通的支付工具是各类电子货币，支付功能的实现需要通过在线商用电子化工具以及互联网中的交易信息来体现，网上支付的交易安全保证则通过网络交易安全认证机构全过程以及互联网自身的安全设施来实现。其系统的构成如图 2.1 所示：

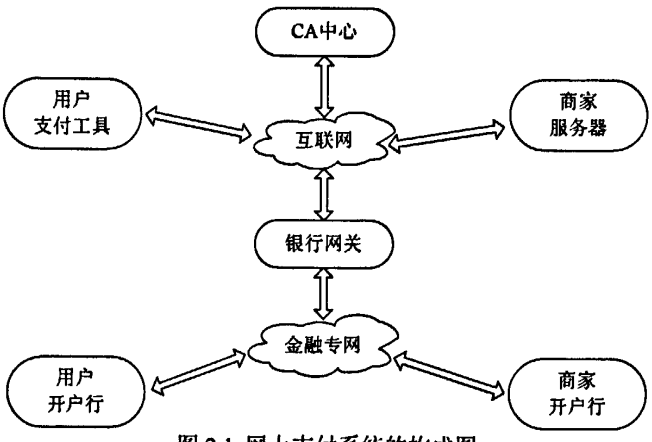

图 2.1 网上支付系统的构成图

(1) 用户：是指与商家有交易关系的一方，用户用自己拥有的支付工具（如信用卡、电子钱包、电子支票等）发起支付指令，是整个支付系统运作的源头。

(2) 商家：商家是商品交易的另一方，它可以用自己的商家服务器根据用户的购物所发起的支付指令向银行请求获取货币支付。

(3) 银行支付网关：支付网关是金融专网和公共互联网之间的纽带，所有的支付信息只能通过支付网关才能进入银行的支付系统，完成支付的授权和获取。

(4) 用户开户行：是指用户在其中拥有账户的银行，用户所拥有的支付工具就是由开户行提供的。用户开户行在提供支付工具的同时也提供了银行的信用，即对支付工具进行签名，承认支付工具的兑付。

(5) 商家开户行：是指商家在其中拥有账户的银行，其账户是整个支付系统中电子资金流向的终点。商家将用户的支付指令提交给其开户行后，就由开户行进行支付授权的请求以及银行之间的结算等工作。

(6) CA 中心：即认证机构，负责为参与电子商务活动的各方发放数字证书、进行身份认证，保证电子商务的安全性。

2.3 网上支付的安全需求

网上支付是电子商务中的重要环节，涉及到用户与银行等金融机构的交互和接口，其安全性是整个电子商务安全中很重要的一个方面。网上支付系统应有安全电子交易协议或安全套接层协议等安全控制协议，这些涉及安全的协议构成了网上交易的可靠环境。不断加强网上支付的信息保密性、完整性和不可否认性是电子商务应用中的重中之重。

在整个电子商务交易过程中网络金融服务都是通过网上支付来完成的，但是由于网上支付是在开放的 Internet 上实现的，支付信息很容易受到网络黑客的攻击和破坏，这些信息的泄露和损坏直接影响到用户或企业的切身利益，所以安全性一直是网上支付实现所要考虑的核心问题之一。

网上支付系统的各种安全需求在整体考虑的基础上，采取综合的安全解决办法，建立起完整的安全控制体系和保证体系，以保障支付信息的保密性、完整性、可用性和可靠性等安全指标[5]。目前网上支付的安全需求如下：

(1) 保密性

网上支付保密性主要指保证数据不受未授权的非法访问而导致数据失密，主要分为数据存储保密性和数据网络传输的保密性。系统需要及时对传输信息进行加密处理，防止交易中信息被非法截获或者读取。就是防止通过非法拦截会话数据获得账户、金额等有效信息。保密性意味着在参与者之间的通信通道具有保密性，仅允许目标支付方可以看到支付数据。

(2) 完整性

网上支付完整性是指数据不接受未授权的非法修改，从而使得数据保持未受

损害的完整状态，主要分为数据存储完整性和数据网络传输的完整性。系统需要防止对支付信息未经授权的随意改动，防止数据传输过程中支付信息的丢失或者重复，并且保证信息传递次序的统一。完整性意味着在网上支付系统中的支付数据，不能被未经授权的参与者修改或者破坏，保证一旦支付交易提交，支付数据不能非法修改，同时也确定了支付数据的一致。

(3) 可靠性

网上支付系统能够提供对用户身份的鉴别方法，确保用户身份信息的可靠与合法，实现系统对用户身份的有效确认，对私有密钥和口令有效保护，对非法攻击能够防范，防止假冒身份进行交易和诈骗，确保系统的无故障和无差错。

(4) 抗否认性

抗否认性也称抗抵赖性，系统应该有效防止支付欺诈行为的发生，保证支付信用和行为的不可否认性，保证支付参与方对已做交易无法抵赖。

(5) 可用性

可用性是指信息可被授权实体访问并按需求使用的特性。网上支付存在着由于计算机病毒或者其他人为的原因可能造成对合法使用人的拒绝服务，也可能存在被他人滥用机器或者信息的情况，这就是可用性需求。网上支付的可用性与硬件的可用性、软件的可用性、人员的可用性、环境的可用性等方面有关。

2.4 网上支付工具

(1) 信用卡

信用卡是一种可在一定范围内替代传统现金流通的电子货币，具有支付和信贷功能。在我国目前通用的主要是借记卡，在消费前需在卡内存入一定的金额，不提供信贷业务。

(2) 电子钱包

电子钱包是电子商务购物活动中常用的一种支付工具，其适于小额购物。在电子钱包内存放的电子货币，如电子现金、电子零钱、电子信用卡等。

(3) 电子支票

电子支票是纸质支票的电子替代物，电子支票将纸质支票改变为带有数字签名的电子报文，或利用其它数字电文代替纸质支票的全部信息。支票上除了必须的收款人姓名、账号、金额和日期外，还隐含了加密信息。电子支票通过电子邮件直接发送给收款方，收款人从电子邮箱中取出电子支票，并用电子签名签署收到的证实信息，再通过电子函件将电子支票送到银行，把款项存入自己的账户。

2.5 网上支付协议

2.5.1 SSL 协议

安全套接层协议（Secure Socket Layer，SSL）[6]最先是由著名的 Netscape Communication 公司开发的，现在被广泛应用于 Internet 上的身份认证与 Web 服务器和用户浏览器之间的数据安全通信。

指定 SSL 协议的宗旨是为通信双方提供安全可靠的通信协议服务，协议包括两个层次，其较低的 SSL 记录层协议位于传输层协议 TCP/IP 之上。SSL 记录协议用来对其上层的协议进行封装。握手协议就在这些被封装的上层协议之中，它允许用户端与服务器彼此认证对方，并且在应用协议发出或收到第一个数据之前协商加密算法和加密密钥。这样做的原因就是保证应用协议的独立性，使低级协议对高级协议是透明的。

目前的 WEB 服务器和浏览器都集成了 SSL 协议，使得基于 SSL 协议的电子支付系统中交易的任何一方都无需安装专门的软件，操作简单、运行速度快，因此基于 SSL 协议的电子支付得到了广泛应用。但是由于 SSL 协议只是端到端的传输层加密协议，在 SSL 协议中只有商家服务器的认证是必须的，而用户端的认证则是可选的，因此它不能解决交易过程中的多方身份认证问题。在安全性方面，SSL 协议采用了公钥加密、信息摘要和 MAC 检测，可以提供保密性、完整性和端到端的身份认证功能，但不能构建一个安全可靠的支付体系[8]。

2.5.2 SET 协议

SET（Secure Electronic Transaction 即安全电子交易协议）[7]是美国 Visa 和 MasterCard 两大信用卡组织联合其它厂商制定的一个能保证在公共互联网上进行安全支付的电子支付协议。为此，SET 得到了 IBM、HP、Microsoft、Netscape 等很多公司的支持并已作为开放的工业标准发布。

SET 是一种应用于开放网络环境下，以信用卡为基础的安全电子支付系统的协议，它给出了一套电子交易的过程规范。通过 SET 这一套完备的安全电子交易协议可以实现电子商务交易中的加密、认证机制、密钥管理机制等，保证在开放网络上使用信用卡进行网上购物的安全。

SET 是为解决用户、商家和银行之间使用信用卡网上支付交易而设计的，以确保支付信息的保密性、支付过程的完整性、商家及持卡人身份的合法性以及可操作性。SET 协议中使用了对称密钥和非对称密钥的密码技术、数字签名、数字信封、数字证书等安全技术，实现了信用卡在电子商务交易中的信息保密性、完整性、身份认证等，为电子商务交易在互联网中的运行提供一种安全机制。

SET 协议位于应用层，不仅规范了整个商务活动的流程，而且制定了严格的加密和认证标准，具备商务性、协调性和集成性功能。但是基于 SET 协议的电子支付系统目前只能适用于电子钱包的网上支付，因此需要在用户端安装专门的电子钱包软件，在商家服务器和银行网络上也需要安装相应的软件，这就使得整个支付系统的运作成本增加，且使用 SET 协议交易起来过程非常复杂[9]。

不论是 SET 协议还是 SSL 协议，所采用的加密算法都是国外的进口加密算法，一直受到国外算法出口限制。协议规定的对称加密算法为 DES 算法，非对称加密算法为 RSA 算法，但是目前的密码技术研究表明 DES 算法和短密钥的 RSA 算法存在被破解可能。因此，我们亟待寻求如何把我们自主研发的国产商用加密算法应用到网上支付协议中的方法。利用目前新兴的 USBKey 设备可以将国产商用密码、DES、RSA 等现代密码内置到自身的安全芯片中，这样一来我们可以将几种密码相结合使用，提高系统的安全性。再者，USBKey 不仅可以通过安全的身份认证、数字签名的功能解决了 SSL 协议的不可否认性问题，也可以集成电子钱包等功能避免了 SET 协议中只能单纯依靠客户端软件来实现网上支付的缺点。本文正是基于这些问题提出了一种基于 USBKey 的网上支付方案。

第三章 网上支付相关安全技术研究

3.1 密码学基础

3.1.1 对称密码技术

(1) 概述

对称密码体制又称单钥或私钥（Privare-key）[10]，是从传统密码学中的简单移位、代替发展而来的。对称密码体制的特点是加密和解密采用相同的密钥。因此在通信过程中，通信双方需要使用安全信道来传输并保存他们的共同密钥。对称密码体制的模型如下图 3.1 所示：

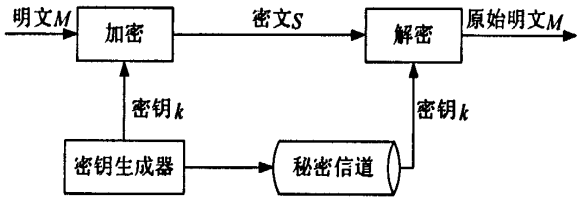

图 3.1 对称密码模型

对称密码体制的安全性依赖于密钥的安全性。它的主要特点就是计算开销小，加密速度快。它的缺点在于双方通信时怎么样确保密钥安全交换的问题。所以对称密码体制主要用于信息加密。

目前，广泛使用的对称加密算法有：数据加密标准 DES、3DES、高级加密标准 AES、国际数据加密标准 IDEA 和 BLOWFISH 等。

(2) DES 算法

DES[11]（Data Encryption Standard）即数据加密算法。它是 IBM 公司于 1975 年研究出来的一种加密算法算法，1977 年由美国国家标准局公开发布的。DES 是一种特殊类型的迭代密码，叫做 Feistel 型密码。它是以 64 比特为分组对数据进行加密的。64 比特为一组的明文作为算法的输入，64 比特的密文作为算法的输出。密钥的长度为 64 比特，但实际上只有 56 比特，其中 8 比特作为奇偶校验位，可以被忽略。DES 加密算法是混乱和扩散的组合，要进行 16 次组合（即替代和置换）。DES 算法是公开的，系统的安全性主要依赖密钥的保密性。

为了提高 DES 的安全性并充分利用有关 DES 的现有软件和硬件资源，人们提出了 3DES。3DES 又称 Triple DES，是 DES 加密算法的一种更安全的模式。其优点就是密钥长度增加到 112 比特或 168 比特，可以有效的克服 DES 面临的穷举搜索攻击；相对于 DES 增强了抗差分攻击和线性攻击的能力。

3.1.2 非对称密码技术

(1) 概述

在非对称密码体制之前，如何进行对称密码体制的密钥安全分发，一直困扰着密码学家。计算机网络的发展，电子商务的兴起，给网络通信的安全带来了更多的挑战。如果没有良好的保密机制，会给商家和用户带来巨大的经济损失。而非对称密码体制的提出和使用解决了上面的问题。

1976年，W.Diffie 和 M.Hellman 在 IEEE Trans.on Information 刊物上发表了"New Direction in Cryptography"文章[12]，提出了非对称密码体制即公开密钥密码体制的概念，开创了密码学研究的新方向，引起了广泛的关注。非对称加密算法又叫公开密钥加密算法。它需要两个密钥：公开密钥和私有密钥。公开密钥和私有密钥组成一个密钥对，如果用公开密钥进行加密，那么只有用对应的私有密钥才能解密；如果用私有密钥进行加密，那么只有用对应的公开密钥才能解密。非对称密码体制的模型如下图3.2所示：

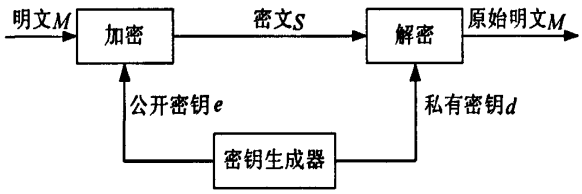

图3.2 非对称密码模型

实现公钥有多种方法和算法，大多数都是基于求解难题的。常用的非对称密码算法有RSA、Diffie-Hellman、ElGamal、DSA等等。其中，RSA算法的安全性是基于大素数分解的困难性，常被用于信息加密和数字签名；Diffie-Hellman算法产生的密钥可用于加密、密钥管理或任何其它的加密方式，通常被用作密钥交换的基础；ElGamal加密法主要是依靠求解离散对数问题的操作难度来保证其安全性，既能用于数据加密也能用于数字签名；DSA安全性与RSA相比差不多，虽然DSA也可以用于加密，但它通常仅用作数字签名。

1978年，麻省理工学院的Rivest、Shamir 和 Adleman[13]实现了基于Diffie 和 Hellman[14]思想的一种算法，简称RSA算法。它是基于大数分解难题的一种算法。它的安全基础是欧拉定理，其安全性依赖于大数的因数分解的困难性。目前RSA算法广泛应用于数字信封、数字签名、数字证书等领域，是PKI体系的基础。

(2) RSA算法

假设M为明文，C为密文。

1) 选择两个满足需要的大素数p和q，并且p和q保密；

2) 计算$n=q\times p$（公开），$\varphi(n)=(p-1)(q-1)$（保密），其中$\varphi(n)$是n的欧拉

函数值；

3) 随机选取整数e，使其满足$\gcd(e,\varphi(n))=1$（公开），$e\in(0,\varphi(n)-1]$；

4) 计算d的值，满足$de\equiv 1 \bmod(\varphi(n))$（保密），$d\in(1,\varphi(n))$；

5) 加密运算：$C=E(M)=M^e(\bmod n)$；

6) 解密运算：$M=D(C)=C^d(\bmod n)$。

其中在利用RSA算法加密前将明文M数字化，并取长度小于$\log_2 n$位的数字做明文块。最终的公钥为(n,e)，私钥为(n,d)。

由于RSA算法的安全性依赖于大数的因子分解，就目前的技术而言，解决大数的分解问题是比较困难的。在RSA算法计算过程中有大量的大数分解运算、模幂运算、模乘运算、模逆运算，速度很慢，随着密钥位数的增大，所需时间呈指数增长。产生密钥麻烦，因为受到素数产生技术的限制，所以难以做到一次一密。RSA加密算法比较适合于信息量较少的信息。

3.1.3 Hash函数

Hash函数又叫散列函数、哈希函数或杂凑函数，在现代密码学中扮演很重要的角色。Hash函数是一个公开的函数，记为H。Hash函数将任意长度的消息M映射为较短的、固定长度的一个值$H(M)$，记为$h=H(M)$。称$H(M)$为杂凑值或消息摘要。Hash函数是一种单向密码体制，从明文到密文是不可逆的，即只有加密过程，没有解密过程。给定M，很容易能算出h。但是给定h，由$h=H(M)$很难计算出M。散列值是消息中所有比特的函数值，即使改变消息中任何一个比特或几个比特都会使散列值发生改变，所以散列函数是实现安全可靠的数字签名和认证的重要工具。

Hash函数最初是为了验证数据的完整性的，后来被用来保证信息的合法性。目前常用的Hash函数有两大系列。

(1) MD（MessageDigest）系列

MD系列是由国际著名密码学家图灵奖获得者兼公钥加密算法RSA的创始人Rivest设计[15]，主要算法包括MD4、MD5。

(2) SHA（SecurityHashAlgorithm）系列

1995年SHA-1算法被提出，产生160位的杂凑值[16]。目前有SHA-1，SHA-256，SHA-384和SHA-512这几种单向散列算法。

3.2 公钥基础设施（PKI）

3.2.1 PKI 技术

PKI 即公钥基础设施，其核心技术基础就是公钥密码学的加解密与签名技术。它是国际上解决开放式互联网络信息安全需求的一套体系。它能够为所有的网络应用提供加解密、数字签名和身份认证等密码服务及所必需的密钥和证书管理体系。PKI 体系支持身份认证、消息传输和存储的完整性、消息传输和存储的机密性和不可否认性[18]。

3.2.2 PKI 的构成

完整的 PKI 系统必须具有 CA、数字证书库、密钥备份及恢复系统、证书作废系统、应用接口 API（Application Program Interface）等基本构成部分。

(1) CA：即数字证书的申请及签发机构，是 PKI 应用中权威的、可信任的、公正的第三方，是 PKI 中的核心。

(2) 数字证书库：数字证书是由权威的、可信任的、公正的第三方机构 CA 签发的，是权威性的电子文档。它符合 X.509 标准，是网上实体身份的证明。

数字证书库是 CA 颁发证书和撤消证书的集中存放地，用于存储已签发的数字证书及公钥。用户可由此证书库查询所需的其他用户的证书及公钥，从而能知道与之通信方的公钥，验证对方的证书是否有效和真实，也能对对方的身份进行鉴别。

(3) 密钥备份及恢复系统：密钥备份及恢复是密钥管理的主要内容，用户由于某些原因将解密密钥丢失，从而使已被加密的密文无法解开。为了解决这一问题，PKI 提供了密钥备份与密钥恢复机制。当用户证书生成时，解密密钥就会被 CA 备份存储。当需要恢复时，用户只需向 CA 提出申请，CA 就会为用户恢复密钥。但是密钥的备份与恢复必须由可信的机构来完成并且密钥备份与恢复只是针对解密密钥，用于签名的加密密钥为确保其唯一性是不能够作备份的。

(4) 证书作废系统：证书作废系统是 PKI 的一个必备组件。证书在有效期以内也可能由于密钥介质丢失或用户身份变更等原因需要作废。为实现这一点，PKI 提供了证书作废的一系列机制。

(5) API：是 PKI 必须提供的应用接口系统，使得各种应用能够以安全可信的方式与 PKI 交互，确保网络环境的完整性和易用性。

PKI 作为一种安全技术，已经深入到网络的各个层面。虚拟专用网络、安全电子邮件、WEB 安全、电子商务等服务都是以 PKI 技术为前提的。

3.2.3 PKI 的优势

(1) 采用公钥密码技术：能够支持公开验证并且无法伪造数字签名，在支持可追溯的服务上具有不可替代的优势。这种可追溯的服务也为原发数据完整性提供了更高级别的担保。

(2) 采用密码技术：PKI 不仅能够为相互认识的实体之间提供机密性服务，也可以为陌生用户之间提供保密支持。

(3) 用户独立验证数字证书：用户验证证书时不需要在线查询，原理上能够保证服务范围的无限制扩张，这就使得 PKI 能够成为一种服务于庞大用户群的基础设施。PKI 通过第三方颁发的数字证书证明终端用户的密钥，而不是过去的在线查询或在线分发，突破了过去必须在线进行安全验证服务的限制。

(4) 证书撤销机制：证书的撤销机制为用户提供了在意外情况下的补救措施，让用户在各种安全环境下都可以放心使用。此外，因为有撤销技术，不论固定身份、还是经常变换的身份，都不用担心证书被盗窃后其身份被作废或被他人恶意利用，这使得其应用领域不受具体应用的限制。

(5) 互联能力：PKI 能够按照人们信任的方式进行多种形式的互联互通，从而使 PKI 能够很好地服务于符合人们习惯的大型网络信息系统。PKI 中各种互联技术的结合使建设一个复杂的网络信任体系成为可能。

3.3 数字证书

3.3.1 概述

数字证书是一种权威的电子文档，是由权威、公正的第三方机构（CA）签发的证书。这个电子文档的所有内容必须有权威机构的数字签名。数字证书的主要内容有用户的真实姓名、身份唯一标识 ID 和用户的公钥。以数字证书为核心的加密技术，可以对网络上传输的信息进行加密和解密、数字签名和签名验证，确保网上传递信息的机密性、数据交换的完整性、发送信息的不可否认性、交易者身份的确定性。

从证书的应用角度来分，数字证书可以分为以下几种：

(1) 服务器证书：在服务器上安装证书后，用户端浏览器可以与服务器证书建立 SSL 连接，在 SSL 连接上传输的任何数据都会被加密。

(2) 电子邮件证书：它可以证明电子邮件发信人的真实性。

(3) 用户端个人证书：主要用来进行身份验证和数字签名。

最简单的证书包括用户的公开密钥、名称以及证书授权中心的数字签名。一般情况下还包括密钥的有效时间、发证机关的名称、证书的序列号等信息。证书

的格式有好些种，目前被大量接受的证书格式是由ITU-T（国际电信联盟电信标准化组织）定义的X.509[19]国际标准。

3.3.2 数字证书的存储

以往PKI/CA为用户签发证书时，一般用户在自己的浏览器中通过非对称算法RSA在内存中运算产生密钥对。这一操作会将用户的私钥读取到PC机的内存中再进行签名运算，将私钥的安全依托于PC机操作系统的安全，这必然是一个很大的漏洞。但是，往往PC机的操作系统很容易受到网络病毒或木马的攻击，攻击者会利用系统漏洞盗取用户的私钥和数字证书，就可以假冒用户进行非法签名等。

要想防止这一漏洞，唯一解决的办法就是不能将私钥存储在PC机上，只能保存在其他外部设备中，而且为了防止网络病毒或木马的攻击也不能将私钥读取到PC机的内存中进行运算，还要保证利用私钥能够进行数字签名。在智能卡基础上发展起来的USBKey设备，配合PKI体系就能很好的解决上述问题。既能生成密钥对，存储私钥也可以进行数字签名，整个过程只在USBKey完成，提高了系统的安全性。

3.3.3 数字证书的生成

数字证书颁发的过程一般为用户先产生自己的密钥对，并将公共密钥及部分个人身份信息传送给认证机构，身份核实后认证机构会给用户一个数字证书，里面包含用户的个人信息和公钥信息，同时还有认证机构的签名信息。数字证书生成步骤如图3.3。

图3.3 数字证书生成步骤

(1) 密钥对的产生

密钥对的产生有两种，一种是终端用户自己用软、硬件的方法来生成密钥对。一种是让认证机构代理用户来产生密钥对，这个方案的缺点就是在给用户发送的时候有可能中途被攻击者截获。在这里密钥对的产生是在用户终端产生，并保存在USBKey内，不能导出和复制。

(2) 注册

终端用户生成密钥对后要向RA发送公钥和相关的注册信息（比如电子邮箱，用户名称等）。正确填写信息后发送到RA。

(3) 验证

RA 要验证用户材料的真实性和用户是不是私钥的拥有者。

一般有三种方式来验证：

1) RA 要求用户用私钥对证书签名请求进行数字验证，如果 RA 能够用用户的公钥验证签名的正确性，则相信该用户是私钥的拥有者。

2) RA 也可以生成随机数，用用户的公钥进行加密后发给用户，用户能解密则相信这个用户是私钥的拥有者。

3) RA 可以给用户生成一个哑证书，用用户的公钥进行加密，将其发给用户，用户用私钥能正确解密这个证书才能取得明文证书。

如果以上步骤都成功，则 RA 把用户的材料交给证书机构 CA，CA 进行必要的验证后将对用户生成数字证书，并会存储一份证书记录。认证机构发给用户的数字证书里面包含用户的个人信息和用户的公钥，同时还有 CA 的签名信息。

(4) 证书生成后用户就可以下载到 USBKey 中进行保存，且 USBKey 使用时需要输入保护密码 PIN，构建成一个双因子保护体系。用户就可以利用 USBKey 中的数字证书安全地在网络中向其他的接收方证明自己的身份。另外因为每个数字证书中都有用户的公钥，证书也可以向接收方证实自己是公钥的持有者，也起到了公钥的分发作用。

3.4 身份认证

3.4.1 概述

身份认证技术是指计算机及网络系统确认操作者身份的过程所应用的技术手段，是信息安全理论和技术中非常重要的方面，它是网络应用系统中的第一道安全防线，是安全的网络系统的门户。在计算机网络的虚拟世界中，一切信息包括用户的身份信息都是用一组特定的数据来表示的，计算机只能识别用户的数字身份，所有对用户的授权也是针对用户数字身份的授权。如何保证以数字身份进行操作的用户就是这个数字身份合法拥有者，也就是说保证操作者的物理身份与数字身份相对应，就成为一个很重要的问题。身份认证的目的是验证通信双方的真实身份，防止非法用户假冒合法用户访问敏感的信息资源。

身份认证技术从使用的认证工具可分为软件认证和硬件认证；从认证的条件可以分为单因子认证和双因子认证；从认证信息可以分为静态认证和动态认证。身份认证技术的发展，经历了从软件认证到硬件认证，从单因子认证到双因子认证，从静态认证到动态认证的过程。

3.4.2 基于USBKey的身份认证

基于USBKey的身份认证方式是近几年发展起来的一种方便、安全的身份认证技术[20][21]。它采用软硬件相结合的强双因子认证模式，很好地解决了安全性与实用性之间的矛盾。利用USBKey内置芯片中的密码算法实现对用户身份的认证。USBKey身份认证的优点有以下三个方面：

(1) 双因子认证

每一个USBKey都具有硬件PIN码保护，用户只有同时拥有USBKey和用户PIN码，才可以用USBKey进行操作。相比之前的单一口令认证，很大程度上提高了系统的安全性。

(2) 安全存储空间

USBKey具有一定的安全数据存储空间，可以存储用户密钥、数字证书等秘密数据，对该存储空间的访问操作必须通过特定的程序来实现，且其中存储的用户私钥是不能导出的，这就杜绝了非法用户盗取用户密码和仿冒身份的可能。

(3) 硬件加密算法

USBKey内置安全芯片，可以结合PKI技术构建安全的数据处理体系，体系中使用的数据摘要、数据加解密和数字签名等各种算法的运算处理完全在USBKey内进行，极大地提高了安全性。并且其硬件运算效率相比软件运算有了很大的提高。

3.5 数字签名

3.5.1 数字签名的定义

在ISO7498-2标准中把数字签名定义为[18]："附加在数据单元上的一些数据，或是对数据单元所做的密码变换，这种数据和变换允许数据单元的接收者用以确认数据单元来源和数据单元的完整性，并保护数据，防止被人伪造"。

数字签名技术（又称公钥数字签名、电子签章）是一种给电子信息进行签名的方法，它可以等同于手写签名类似的作用。防止签名消息的伪造或篡改，保证了电子交易中信息的合法性及不可否认性。此外，数字签名也可用于通信双方的身份鉴别。一套数字签名通常定义两种互补的运算，一个用于数字签名，一个用于签名验证。目前，国际上都普遍使用基于PKI技术数字签名技术来进行电子签名。

3.5.2 数字签名的分类

(1) 基于数学难题的分类：基于大因子分解问题的方案（RSA算法）和基于有

限域离散对数问题的方案（DSA 算法）。

(2) 基于签名用户数量的分类：根据签名用户的数量分为：单个用户签名和多个用户签名（多重签名）。

(3) 基于特殊用途的分类：若一般的数字签名方案不能满足特殊的签名需求时，便需要寻求其他数字签名，主要有：盲签名、双重签名、群签名、门限签名、代理签名、门限代理签名、以及报文还原签名。

目前的网上支付系统中主要用到的是基于 RSA 公钥体制的数字签名。

3.5.3 公钥体制数字签名

(1) 数字签名原理

数字签名过程可分为数字签名和签名验证两个阶段。通信双方利用各自的数字证书相互进行身份合法性的验证后，再进行数字签名和签名验证流程，数字签名原理图如图 3.4。

1) 数字签名阶段

在数字签名阶段，发送方将原始消息 M 按双方之前约定的 Hash 算法计算得到一个固定位数的消息摘要 h。将该消息摘要值 h 用发送方的私钥 x_d 加密得到签名消息 C，即数字签名。然后把数字签名 C 和原消息 M 一起发送给接收方。

2) 签名验证阶段

签名验证阶段，接收方收到原始消息 M 和签名消息 C 后，用与发送方同样的 Hash 算法对原始消息 M 计算摘要，值假设为 h。再用发送方的公钥 x_e 对签名消息 C 进行解密，假设得到 h'。然后比较 h 和 h'，如果两者相等则说明消息确实来自发送方。这个过程即为签名验证。

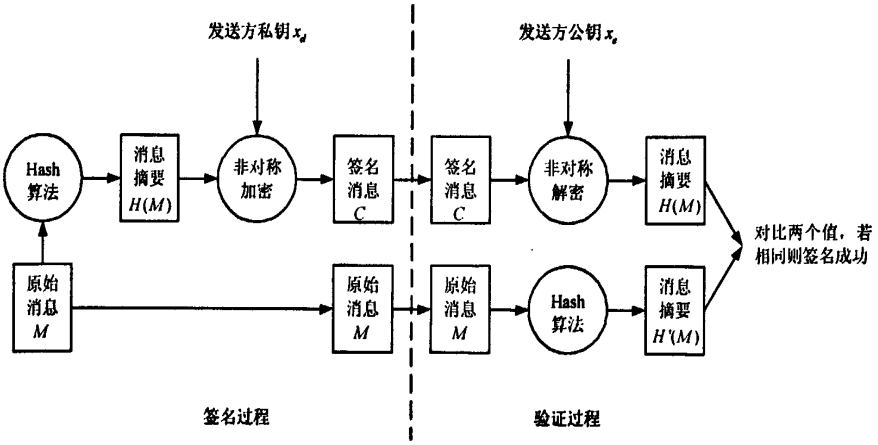

图 3.4 公钥体制数字签名原理图

(2) 安全性分析

1) 若攻击者截获了原始消息 M 和签名消息 C，也得到发送方的公钥 x_e，但由

于获取发送方的私钥x_d是不可能的，因为这将面临因子分解的数学难题。

2) 若对原始消息M或签名消息C进行修改，但是Hash函数是单向的，接收方便可通过摘要信息h和h'是否一致进行验证。需面临单向Hash函数的不可逆的难题。

3) 数字签名本身并不能对原始消息M提供保密性，若原始消息M是非常重要的消息，还需再对原始消息M进行加密或隐藏处理。

3.5.4 数字签名与数字信封

公钥密码体制在应用中实际包括数字签名和数字信封[18]两种方式，前面已经讨论过数字签名本身无法对消息进行保密，而数字信封的出现解决了该问题。数字信封采用密码技术保证了只有合法的接收者才能获取消息的明文。消息发送者先利用随机产生的对称密码加密消息，在利用接收方的公钥加密该对称密钥，被公钥加密后的对称密钥称为数字信封。只有接收方用自己的私钥才能解密数字信封获取对称加密密钥得到消息的明文，保证了消息的隐蔽性、真实性和完整性。在SET协议中就是用了数字签名和数字信封技术。

3.5.5 基于USBKey的数字签名

基于USBKey的数字签名过程是：用户通过USBKey产生密钥对，再向CA申请数字证书，将从CA处申请到的数字证书保存在USBKey中，CSP可以通过命令从USBKey中读取证书，但是任何命令都无法读出用户的私钥。在数字签名时，CSP向USBKey发送一个签名命令即可，USBKey内部便会自动进行签名运算，并将签名结果送出，这就保证了所有的运算都在USBKey内部进行，私钥永不出USBKey。

第四章 USBKey 的体系结构和安全研究

4.1 USBKey 概述

USBKey 是从智能卡技术发展而来的。它实质上就是硬件（安全芯片）和软件（芯片操作系统）组成的安全产品。它内置了现代密码算法，是新一代的身份认证新产品。USBKey 的操作系统标准和智能卡的兼容，他的最大优势就是通过 USB 接口进行通信，它的传输速度可以高达 480Mbps。另外，USBKey 体积小，便于携带，不易磨损，支持即插即用；安全性高，运算能力强，安全芯片带有密码协处理器，能够进行 DES、RSA 等多种算法的运算，是其他存储介质无法比拟的。

USBKey 具有独立的数据处理能力和良好的安全性，是用户数字证书和用户私钥存储的理想载体。用户的公私钥对是在 USBKey 内产生的，私钥是不能读取和复制的。整个数字签名过程是在 USBKey 内完成的，私钥永远不会暴露在 USBKey 外部，网络黑客也永远不可能得到用户的私钥，更加无法仿冒用户的身份，比传统方式下在主机端实现签名更加的安全。

目前，USBKey 是大多数银行采用的客户端解决方案，使用 USBKey 存放代表用户唯一身份的数字证书和用户私钥。随着电子政务、电子商务的大力普及，USBKey 有着很好的使用前景。

4.2 USBKey 体系结构

4.2.1 概述

USBKey 从硬件结构上可分为数据传输模块和智能芯片两大部分。数据传输模

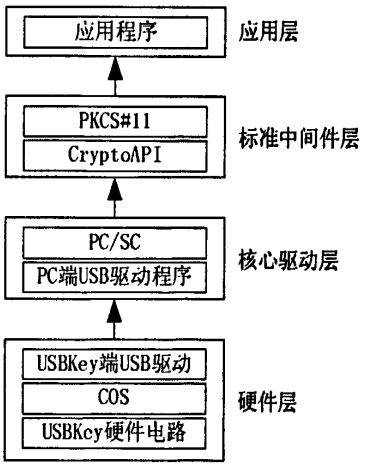

图 4.1 USBKey 的体系结构图

块负责PC机与智能芯片之间的数据传输。其中与PC机的通信使用的是USB的传输协议，而与智能芯片之间的信息交换必须遵循智能卡的国际规范ISO7816[22]。智能芯片部分负责应用程序的存储管理和运算。这些功能的实现都是通过芯片操作系统COS（Chip Operating System）来完成的。

一个完整的USBKey体系结构包括四层：硬件层、核心驱动层、标准中间件层、应用层，结构如图4.1所示。

4.2.2 硬件层

硬件层包括USBKey的硬件电路、COS和USB驱动程序。COS主要用来控制USBKey和外界的信息交换，管理USBKey内的存储及安全保护。该层通过USBKey芯片集成的USB2.0控制器与用户主机以USB标准协议进行通信[23]。

(1) 硬件构成

无论是哪个厂商哪个系列的产品，USBKey的硬件逻辑结构构成都基本类似。它的硬件构成包括微处理器、密码协处理器、随机数生成器、存储器、I/O接口、USB控制器、调试接口和安全访问逻辑，其芯片的硬件结构图如图4.2所示。

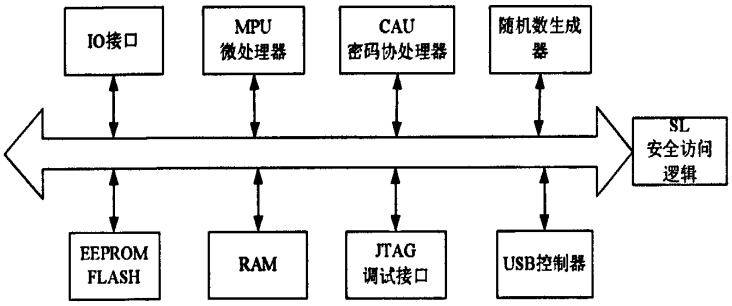

图4.2 USBKey芯片的硬件结构图

1) 微处理器MPU（Mini Processor Unit）是USBKey的核心，在功能上类似于PC机上的CPU。完成基本的指令执行、存储控制和逻辑控制等。

2) 密码协处理器CAU（Cipher Arithmetic Unit）用来执行各种加密算法。除了MPU提供基本的运算功能外，为了支持应用的安全，一些芯片中也提供了常用算法的协处理器CAU。

3) 随机数生成器用来生成任意长度的随机数。

4) 存储器用来存储数据和代码。一般包括易失性存储器RAM，非易失性ROM、EEPROM、FLASH等。易失性存储器RAM类似于PC机上的内存，是USBKey使用时临时数据空间。非易失性存储器ROM是USBKey内的只读存储区，COS代码、USB固件程序和一些基本常数都存储在这里，在芯片掩膜阶段这些代码和数据一起写入，在使用阶段不能更改。EEPROM和FLASH可以进行读、写、擦

除这三种操作，灵活性大。一般 USBKey 使用 EEPROM 和 FLASH。

5) 系统 I/O 接口，即输入输出端口，是 USBKey 和外界联系的唯一通道，并且提供芯片与外界之间传送数据所必须的状态和控制信息。

6) USB 控制器，控制 USBKey 和主机端的 USB 传输的硬件逻辑部件。

7) 调试接口 JTAG 是在 USBKey 芯片的开发阶段提供给开发人员的接口，COS 的程序通过该接口下载到 USBKey 中并且可以调试源代码。

8) 安全访问逻辑 SAL（Secutity Access Logic）是芯片自定义的一些硬件安全逻辑。例如存储区的分区，不同区域的访问控制，对 CAU 的操作权限、异常的外部光电信号检测等。

(2) COS

USBKey 之所以能够支持这么多复杂的应用，其便利和安全的基础便是其内部的操作系统 COS。USBKey 上的 COS 和 PC 机上的 Windows 等操作系统很相似，但是它们又有很大的区别。与 PC 机上的操作系统相比较，USBKey 中的 COS 更加接近于一个监控程序。USBKey 中的 COS 是一个专用的系统，它是为某种类型的 USBKey 和某种应用范畴而专门设计开发的。USBKey 中的 COS 主要解决的是对外部命令如何进行处理和响应的，并不涉及共享、并发的管理及处理复杂的问题。在 USBKey 的 COS 设计时一般要根据 USBKey 的存储器的分区情况，按照国际标准规定的功能进行设计和开发。从接口设备的角度来看，USBKey 的操作系统相当于一个命令解释器，它通过指令访问 USBKey 内部的命令解释器。当 USBKey 接收到命令后，USBKey 的 COS 经过指令分析后，转到相应的功能模块完成指定的操作。

1) COS 功能

USBKey 中 COS 的主要功能是控制 USBKey 和外界的信息交换，管理 USBKey 内的存储，并在 USBKey 内完成各种命令的处理及安全保护。具体功能如下：

- USBKey 的上电复位；
- 芯片底层硬件的驱动；
- USBKey 对外接口交互；
- USBKey 内存空间的维护；
- USBKey 内文件系统的维护；
- 系统运行安全的控制；
- 命令的处理；
- 应用代码的执行。

2) COS 模块

根据 COS 具体实现我们将 COS 划分为四个主要的功能模块：I/O 模块、文件系统模块、安全模块和命令模块。COS 的逻辑结构如图 4.3 所示。

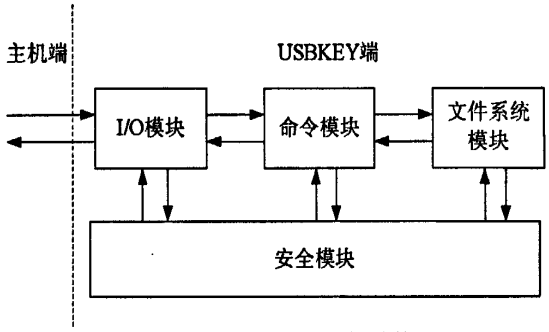

图4.3 COS的逻辑结构

I/O模块：是COS的基本模块之一，它的开发遵循传输协议的定义。根据ISO7816-3[22]接触式智能卡的电信号和传输协议，智能卡的通信传输协议有两种：T=1（异步半双工分组传输）协议和T=0（异步半双工字符传输）协议。T=1协议以定长的数据块为基本单位，T=0协议以单字节的字符为基本单位。

命令模块：对USBKey接收的命令的可执行性进行判断，然后执行相应的操作。ISO/IEC7816-4[22]国际标准中对COS的基本命令集做了规定，主要定义了文件操作命令、信道管理命令以及认证与校验命令。

文件模块：主要负责组织、管理、维护USBKey内存储的所有数据。文件系统的设计和实现既是COS系统中最灵活的部分，也是对系统整体结构影响最大的模块之一。在USBKey中，数据的管理和维护一般通过文件系统来进行。COS的文件系统的主要功能有：空间的管理维护、数据的有效性存储、数据的查找与定位、数据的修改和维护。COS的文件系统有三种类型：主文件MF（Main File）、专用文件DF（Definition File）、基本数据文件EF（Elementary File）。

安全体系：是USBKey最核心的模块。在USBKey中，COS的安全模块分散在COS的各层中，包括最底层的加密算法、系统安全服务和应用安全控制等。其中最核心的是安全服务中系统所提供的安全策略。USBKey的安全性就是对USBKey内数据对象访问的安全控制能力。USBKey安全体系结构的总体结构可以分为三个部分：安全属性、安全状态和安全机制。

安全状态：记录的是当前USBKey所处的安全级别，是一种中间状态信息。表示的是在USBKey完成复位或执行鉴别命令或是进行安全报文校验后可能达到的状态。COS的安全状态分别用两个寄存器来表示，一个称为MF的安全状态寄存器，表示整个USBKey所处的安全级别，另一个为当前DF的安全状态寄存器，表示当前DF所处的安全级别。

安全属性：数据对象的安全属性也就是对数据对象的访问控制，包括两个方面：文件访问安全属性、命令安全属性。

文件安全属性文件访问的安全属性包括了两方面的内容：一个是允许的操作类型，这个由COS提供的功能和应用需要来确定。比如对普通文件允许的读写操

作，对 PIN、私钥等文件读操作禁止，对应用的特殊文件只能通过应用命令来维护等。一个是进行操作所需要的满足的安全条件，也就是要求的安全状态。所以文件安全属性可以看做是对文件所能进行的操作及其需要满足的安全条件。文件的安全属性一般包含在文件的描述块中，在文件建立阶段定义，由 COS 来进行管理和维护，是一种显示机制。

命令安全属性比较简单，由具体的命令来定义，是一种隐性机制。命令的安全属性也包含两个方面内容。一是命令全部报文的安全控制，采用的是在命令数据域中添加命令安全报文的方式。二是命令数据域的安全控制，对某些敏感信息，如密钥在 USBKey 和主机间必须以密文形式传送，以此确保数据传输过程中的机密性。

安全机制：指的是在 USBKey 中与安全相关的所有元素，在 USBKey 内完成安全操作和安全信息传递所需要的一系列安全机制的集合被称为安全环境。安全环境指定了在接下来的命令中所要执行的加密算法、执行操作的类型、所使用的密钥等。在 USBKey 通电后将自动加载默认的全局安全环境，直到使用命令来改变。

4.2.3 核心驱动层

核心驱动层是指主机端的 USB 驱动程序和 PC/SC（Personal Computer/Smart Card）。PC 端的 USB 驱动是微软已经安装好的 MassStorage 设备的驱动。USBKey 使用起来更加方便。PC/SC 是微软为智能卡访问 Windows 平台而定义的一种标准用户接口 API，提供了一个从 PC 机到智能卡（Smart Card）的整合环境。根据微软定义的 PC/SC 标准开发的驱动接口，使得上层可以通过 Win32 标准函数集来访问 USBKey。PC/SC 接口是关于智能卡应用的国际标准，包括计算机操作系统的智能卡设备管理规范、应用接口规范等。该层负责协调用户主机与硬件层之间的数据交互操作和处理上层应用对 USBKey 的访问请求。

Windows 是唯一支持 PC/SC 标准的操作系统平台，PC/SC 规范也为智能卡业界所使用。

PC/SC 的体系结构如图 4.4 所示。

(1) ICC（Integrated Circuit Card）：通常叫做智能卡，主要用来存储数据和加密服务程序。

(2) IFD（Interface Device）：用来连接智能卡和 PC 机之间的通信的。一个读卡器可以接多张智能卡。

(3) IFD Handler（Handler of Interface Device）：就是读卡器底层驱动，使 PC 机和读卡器进行通信。

(4) ICC Resource Manager：管理和控制应用程序所有对任何读卡器中智能卡的访问。

(5) Service Provider：通常是建立的基于智能卡的应用服务。用户的智能卡用作什么用途就在这里实现。

PC/SC 驱动屏蔽了设备和其他一些通信协议方面的事情，用户要对 USBKey 或智能卡进行开发，只需要调用 Windows 提供的 PC/SC 函数，不用管具体的实现细节。

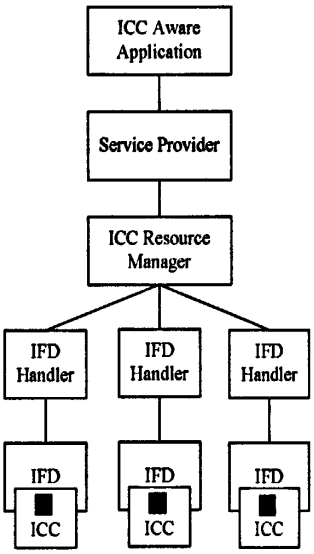

图 4.4 PC/SC 的体系结构

4.2.4 标准中间件层

标准中间件层是处于应用层和设备驱动之间，包括具有跨平台特性的公钥密码标准 PKCS#11 接口和 Windows 平台的密码服务提供商 CSP 接口。

CSP 是微软为 Windows 平台提供的最底层加解密接口。CSP 负责创建和销毁密钥并提供各种密码操作。CSP 可以用软件实现，也可以用硬件实现，比如 USBKey。

PKCS#11 规范为开发人员提供一套通用的跨操作系统和加密引擎的开发界面，并规定了一套独立于技术的程序接口，用于智能卡和 USBKey 之类的加密设备，该接口称作 Cryptoki。

(1) CSP

微软提出的 CSP 接口标准是一个包含了标准加密算法实现的软件模块。CSP 在微软操作系统安全体系中是加解密操作的实际执行者，它可以直接同硬件加密设备如 USBKey 进行交互。其他 CSP 可由加密设备厂商按照微软制定的标准接口

开发，要通过微软签名后安装到系统中才能使用。应用程序都是通过调用微软提供的 CryptoAPI 接口函数，把请求通过 CryptoAPI 传递给 CSP，让 CSP 来完成应用程序所要求的密码操作。CSP 的种类可以有签名型（Signature）、密钥交换型（KeyExchange）和安全通道型（SChannel）等。CryptoAPI 是 Windows 提供的统一安全 API 接口，它由一组函数构成，该 API 的全部功能都需要调用下层的 CSP 服务模块来实现。

(2) PKCS#11

PKCS 是由美国 RSA 数据安全公司及其合作伙伴制定的一组公钥密码学标准，其中包括证书申请、证书更新、证书作废表发布、扩展证书内容以及数字签名、数字信封的格式等方面的一系列相关协议。现在已经公布了 15 个标准，其中 PKCS#11 称为 Cyptoki，定义了一套独立于技术的程序设计接口，用于智能卡和 PCMCIA 卡之类的加密设备具有跨平台的功能。PKCS#11 的一个目标是一个低级的程序接口，该接口将设备的细节抽象化，并把密码设备的通用模型—密码令牌提供给应用程序。第二个目标是资源共享，当台式多任务操作系统越来越流行的时候，单个设备能为一个以上的应用程序所共享。

4.3 USBKey 的攻击与防范

USBKey 以其特有的安全可靠性，作为安全信息的载体被越来越广泛地应用于 PKI 体系中。USBKey 的安全固然成为整个 PKI 系统安全的基础。而 USBKey 的安全又依赖于自身的内置安全芯片和应用过程中是否安全。

目前针对 USBKey 的攻击和对其他智能卡的攻击方法基本一致，其攻击大致分为以下几个方面：对其物理攻击[24]、失效分析攻击[24]、边信道攻击及算法攻击。其中前三种攻击取决于 USBKey 内置安全芯片的安全性；算法攻击是指攻击者利用 USBKey 中算法参数设置的漏洞进攻击。下面就从这些攻击方法来研究分析 USBKey 的安全性。

4.3.1 物理攻击

(1) 攻击方法

1) 探针技术

攻击者通过破坏芯片封装，使芯片内部电路版图裸露，再使用微型探针获取芯片工作的信号，以便分析芯片的相关设计信息和存储结构，有些还可以直接读取出芯片内部存储器的信息来进行数据分析。

2) 版图重构技术

借助半导体失效分析专用设备实施攻击，利用高倍显微镜和微型探针探测芯

片的内部总线、对芯片内部的ROM进行逆向或用聚焦离子束重构内部电路结构等来恢复ROM中的数据信息，以达到攻击的目的。

(2) 防护措施

1) 芯片制造工艺复杂化、精细化

物理攻击是实现成功探测的强有力手段，但其缺点是这种攻击是一种入侵式的攻击模式， 同时需要昂贵的高端实验室设备和专门的探测技术。应对物理攻击的关键在于提高芯片设计的复杂程度和芯片制造的精细程度。安全芯片尽量采用0.13μm以下的生产工艺。同时，芯片版图采用混乱布线的方法实现，在一定程度上提高了芯片版图的重构难度。

2) 表层保护技术

采用复杂的、微米级的超细保护线及专有的多层布局结构，形成可被微处理器控制的主动保护层覆盖整个芯片。这些保护线会被连续地监控，一旦芯片被刺探，芯片内部的微处理器可以及时探测到这些保护线的状态的变化，立即感知到外界入侵的危险，根据COS的设定采用不同级别的主动防御措施。

3) 存储器加密技术

将安全芯片内的所有类型的存储器RAM、ROM等所存储的信息经过特殊的加密机制处理，其结果是即便存储内容被攻击者非法获得，这些加密后的信息对攻击者也是毫无意义的。

4.3.2 失效分析攻击

安全芯片作为半导体器件其电性能会随不同的环境参数而改变。芯片的性能会随着不同的电压、温度、光、电离辐射以及周围电磁场的变化而改变，攻击者试图在芯片的程序逻辑中引入一些错误的行为，如数据复位或随机化，或改变操作码，或使密钥运算在期间出错为诱导攻击创造条件，或导致信息泄漏，输出非正常的数据，包括部分的软件、机密数据，甚至是储存的密钥等。

(1) 攻击方法

1) 尖峰和电涌攻击

电源和时钟信号是安全芯片运行的必需条件，在其电源或时钟信号上突然增加尖峰和电涌信号，会导致芯片的运行故障，使安全芯片暂时失效，攻击者借机实施错误的操作。

2) 电磁干扰攻击

电磁攻击就是通过放在芯片上的电磁线圈将攻击信号注入芯片内部，以干扰芯片的正常工作，使其发生故障。电磁攻击虽然更复杂，但比起传统的尖峰或电涌攻击方法来，一个明显的进步是可以把芯片放在一个特殊电磁环境下的攻击。

这就需要更难的防范策略以应对该方法的攻击。

3) 光和热攻击

光攻击是指利用光照射正在工作的安全芯片表面，由于光的入侵，硅片感受到外界光照内部会产生相应的电压和电流，从而导致故障。

热攻击修改芯片外界的环境温度也可被用作对安全芯片的攻击方法。通过改变周边环境的温度，在极低的温度下 RAM 内存的一些位会被修改或冻结，这种攻击无论芯片是否工作都是可以进行的。

(2) 防护措施

1) 提高电路抗干扰能力

对芯片的电源和信号端口，增加必要的检测和防护电路，防止电压的瞬时变化和时钟信号的异常等。

2) 传感器保护技术

采用传感器技术，在芯片表层增加高灵敏度传感器保护网，一旦芯片受到外界的人为干扰或攻击，如光、温度、电磁等，芯片内部会通过这些传感器立即感知到危险的存在，便会采取相应的自我保护措施，确保芯片内部数据的安全。

4.3.3 边信道攻击

边信道攻击，安全芯片内的逻辑门电路在工作时会不断的开关变化，其操作类型的不同以及所处理的数据不同会在能量消耗、电磁辐射、时间状态上产生一定的变化规律，利用的这些信息就被称为边信道信息，通过一定得相关性运算便可分析获取密钥。针对安全芯片实施的最典型的边信道攻击有：功耗攻击、电磁攻击、时间攻击等。

(1) 攻击方式

1) 功耗攻击

1998 年 Kocher 等三人提出了功率分析的攻击方法 [25]，即根据已知操作确定的功率值，直接分析密码算法运行时所记录到的功率数据。常见的有简单功耗攻击 SPA（Simple Power Analysis attacks）和差分功耗攻击 DPA（Differential Power Analysis attacks）。通过分析功率消耗信号得到处理器正在处理数据的汉明重量信息，利用这些信息可以构造出数据中的相关信息，以便实施攻击。

DPA 是一种对密码芯片的泄漏功耗进行统计分析而恢复密钥的攻击方法。差分功耗攻击就是对大量的曲线样点进行功耗统计测试，即根据大量功耗测试样本来分析可能的密钥值。

DPA 主要有两个步骤：数据采集和数据分析。数据采集需要采集大量的功耗波形，并且在采集同一数据的时要求芯片内部的密钥是固定不变的。数据分析就

是利用数学统计的方法对数据进行相关性处理，最后根据得出的芯片功耗信息来分析对应的密钥值。

2) 电磁攻击[26]

芯片工作时会产生一定量的电磁辐射。攻击者如果可以检测到电磁辐射并且找到它们与底层计算和数据之间的关系，就可能获得和这些计算和数据有关的信息。同功率分析相似，电磁辐射辐射攻击 EMA（Electromagnetic Analysis）也可以分成两大类：简单电磁辐射分析 SEMA（Simple Electromagnetic Analysis）和差分电磁辐射分析 DEMA（Differential Electromagnetic Analysis）。

3) 执行时间分析

由于芯片内操作命令的执行时间与密钥之间存在相关的信息，执行时间分析攻击就是测量加密系统基本运算单元执行时间的泄露来实施攻击的。例如，可以通过测量完成 RSA 私钥导入的操作时间，攻击者可以找到 Diffie-Hellman 密钥交换协议中固定的参数或者 RSA 算法中的密钥[25]，进而破译 RSA 密码系统。

(2) 防护措施

从上面的三种攻击方式可以看出，边信道攻击的主要方法就是捕获芯片泄露信息，再进行相关运算。只要采用合理的软硬件设计减小芯片对外的泄露信息和破坏运算过程中某个中间运算结果与密钥位要具有相关性以及密码模块的泄露信息与被运算处理的数据相关性就可以减小边信道攻击成功的几率。

我们可以从以下几个方面进行防护：

1) 采用平衡电路降低信号能量，使用多层布板技术抑制电磁发射。

2) 执行并行随机处理来增加噪声水平。例如，内部编程电压产生电路可用作并行噪声发生器。

3) 随时处理中断引入的时间噪声和不同的时钟频率。如对差分轨迹进行时间噪声处理会防止或打乱轨迹的排列规律。

4) 引入芯片工作时钟跳变技术，使芯片的工作频率随机的跳转，增加芯片功耗及电磁辐射的随机性。

5) 基于数据随机化掩盖算法的防护方法，利用线性反馈移位寄存器产生随机数，在运算前保证每个输入的关键数据都被随机数所掩蔽，防止密钥相关信息泄露出去。使得电路的功耗、运行时间以及电磁辐射等外界可探测的因素与内部运算数据无关。

6) 引入软件冗余技术，适当的对软件执行过程中的关键流程进行冗余，削弱软件的执行规律，使执行时间和功耗的相关性大大减小。

4.3.4 算法攻击

(1) 攻击方式

算法攻击是指黑客使用具有超强计算能力的计算机，并具有数学方面的特长，对智能卡的安全核心算法进行攻击。目前通用的 USBKey 中涉及的算法有 DES、3DES、RSA、MD5 和 SHA-1。其中 DES、3DES 属于对称密码体制，RSA 属于非对称密码体制，而 MD5 和 SHA-1 为 Hash 函数。

1) DES 算法的攻击

在 DES 算法中存在 12 个半弱密钥和 4 个弱密钥。由于在子密钥的产生过程中，密钥被分成了 2 个部分，如果这 2 个部分分成了全 0 或全 1，那么每轮产生的子密钥都是相同的，当密钥是全 0 或全 1，或者一半是 1 或 0 时，就会产生弱密钥或半弱密钥，DES 算法的安全性就会变弱。

目前对 DES 的攻击方法主要有[27][28]：穷举攻击、差分密码分析、线性密码分析这三种。

2) RSA算法的攻击

RSA 是第一个将安全性基于大数因子分解的密码系统，虽然迄今为止尚无法“证明”破解 RSA 系统等价于因子分解，但一般“相信”RSA 系统的安全性等价于因子分解。Alexi 等曾揭示，从 RSA 加密的密文恢复某些比特的困难性也和恢复整组明文一样困难[29]。RSA 面临的较大威胁主要来自于计算能力的持续提高和因子分解算法的不断改进。所以 RSA 模数也要相应的加长。

RSA 的攻击[27][28]最常用的攻击方法有分解模数攻击、选择密文攻击、低指数攻击、共模攻击、定时攻击等。

3) Hash函数攻击

评价一个 Hash 函数的安全性，是看是否存在有效的攻击方法来破译它，找出该 Hash 函数的一对或者多对碰撞。复杂度是评价 Hash 函数的一个度量尺度。目前已有一些攻击 Hash 函数和计算碰撞消息的方法有：生日攻击、中间相遇攻击[32]、差分攻击[30]及差模攻击[31]等。

(2) 防护措施

为了避免由于算法的应用不当导致 USBKey 的安全性降低，USBKey 应用这些密码算法时需注意以下几点：

1) 尽量选用 3DES 算法，加大密钥长度，增加攻击的时间复杂度；

2) RSA 算法参数的选取需合理，避免由于参数选取的不当造成算法的攻击漏洞。特别要注意对模数长度的选取，一般用户至少选用 1024bit 的 RSA，对安全性要求特别高的用户需选用 2048bit 的 RSA；

3) Hash 函数尽量采用更高级的 SHA-1 以上的算法，避免攻击者利用 Hash 函

数的碰撞性，破坏数字签名系统的安全。

在USBKey的应用中，一方面，将这几种算法相互结合，取长补短以增加攻击的时间复杂度和攻击代价；另一方面，可以通过进一步提高USBKey的硬件运算处理速度，选择更长的密钥或更为复杂的算法来增强其安全性。

4.4 USBKey应用安全分析

4.4.1 安全性分析

为了确保USBKey不被别人利用，比如当你的USBKey丢失后别人就可以直接利用你的USBKey进行非法签名，所以在USBKey使用时必须结合用户PIN（Personal Identification Number），才能完成一次签名操作，PIN的校验保证了只有合法用户才能使用USBKey进行操作，为USBKey的安全使用增加了一道新的防护屏障。

目前普遍的验证过程如下：用户通过与USBKey相连接的PC机键盘入PIN，然后通过APUD指令将PIN送给USBKey，USBKey在和提前存储在自己内部的PIN进行比较，比较的结果在以后访问存储器和执行指令时作为参考，用来判断是否允许访问或执行。例如设置PIN的错误输入次数，若在有限的输入次数内没有输入正确的PIN，USBKey就会自锁，禁止任何操作，这样可以防止非法用户对PIN的猜测攻击。虽然此方法完全可以防御USBKey丢失后被别人冒名利用的风险，但是还有一个安全漏洞就是网络上的黑客可以使用木马程序利用PC机系统的漏洞，盗取用户输入的PIN。这时若用户的USBKey正好连接在PC机上，黑客便可利用盗取PIN来悄悄地取得虚假身份认证远程控制USBKey进行非法签名等操作。如何防范网络黑客的攻击是USBKey使用过程中的首要解决的问题。解决方案有两种：

(1) 采用物理按键的方法阻拦黑客对USBKey的远程控制。

(2) 对用户PIN的保护，斩断黑客的盗取途径。

针对以上问题，按照上述第一种解决方案,目前也出现了一种增强型USBKey，即在即在传统的USBKey上增加液晶显示屏和手动按钮，在进行交易之前，交易关键数据必须在USBKey上显示，使用者通过显示信息来判断交易数据的正确性，如果交易数据正确,就按增强型USBKey上的“确认”键来签名，否则按“取消”键取消签名。

但是该方案存在固有的缺陷，就是显示数据的格式需要事先约定，只有满足事先约定格式的数据才能够显示，但现有通用PKI的应用系统中签名数据不是我们所需要的格式，甚至有些系统只提供摘要后的数据，不提供原始数据，由此导致增强型USBKey和现有的通用PKI系统不兼容。哪些数据才需要是格式化后显

示给用户，哪些数据不需要显示给用户呢？还有用户在什么情况下按键来签名，这是增强型 USBKey 和使用增强型 USBKey 系统必须面临的问题。如果考虑不周，可能会留下隐患给攻击者提供机会，或者给使用者带来极大不便。

4.4.2 一种改进的用户 PIN 输入方案

针对上述 USBKey 应用过程中出现的问题，本文基于第二种方案提出了一种改进的用户 PIN 输入方案，从 PIN 防护角度预防黑客的攻击。与增强型的 USBKey 一样也增加液晶显示屏和手动按钮，不过液晶显示屏和手动按钮的作用是来完成 PIN 输入用的，从源头上彻底斩断了网络黑客盗取用户 PIN 直接操纵 USBKey 的危险。

(1) 改进思路

是将原来在 PC 机上进行 PIN 的输入方式转移到 USBKey 上进行，步骤如下：

1) 在使用 USBKey 时首先会自动进入 PIN 输入模式，验证用户身份；

2) 用户利用 USBKey 上的选择按钮选择液晶屏中的字符并通过确认按钮完成 PIN 字符的选择；

3) USBKey 会将用户输入的 PIN 字符存入自己的缓存中，再与已经提前存储在自己内部的 PIN 进行比较，完成有效性验证；

4) 在验证完 PIN 的有效性后 USBKey 会自动将缓存中的信息清除。

(2) 安全性分析

在整个操作过程完全的避免了传统 USBKey 输入 PIN 时由于 PC 机系统的安全问题造成的 PIN 泄露的隐患，由于网络黑客无法深入到 USBKey 的缓存区，便彻底斩断了网络黑客盗取用户 PIN 码的途径。

(3) 方案的优点

与已有的增强型 USBKey 相比，改进的方案在硬件成本上与其基本持平，都采用了液晶显示屏和手动按钮，但是二者的防护措施却不相同。改进的方案针对 PIN 进行防护，只需在液晶显示屏显示 0 至 9 十个字符即可，不存在像增强型 USBKey 需要进行数据格式处理的问题，就完全可以和传统的通用 PKI 系统兼容，方案的设计不仅简单，也更加安全，使黑客根本就没有可乘之机，因为黑客没有办法盗取用户 PIN。

4.4.3 基于生物识别技术的 USBKey

随着基于生物识别技术的发展，也可以将人的指纹[33]、虹膜、语言等信息集成到 USBKey 中，可以取代 PIN 实现真正的身份验证。但这些都是以牺牲硬件成本为代价换取的，暂时只能用于高端场合，还没有得到大力普及。

第五章 基于USBKey的网上支付方案

5.1 方案概述

本方案是基于USBKey的PKI架构下的网上支付方案，是基于PKI的安全支付协议。综合PKI技术的各种应用，建立一个可信任和足够安全的网络环境，系统由可信的CA认证中心，银行网关、用户、商家服务器、支付工具及银行组成，利用身份认证技术、数字签名技术和数字信封技术等提供了交易双方的身份认证、数据的安全传输、交易的不可否认性等功能，从而能保证网上支付的安全性。

方案将传统网上银行中使用的USBKey的功能进行了扩展，使USBKey不仅能够支持现有的银行卡支付，也能支持电子钱包、电子支票等网上支付工具。开户银行首先对用户和商家进行授权，颁发数字证书至各自的USBKey中，用户再使用USBKey的签名、认证等功能实现网上支付功能。商户也拥有自己的USBKey，只需要通过互联网将用户的消费记录发送给银行的服务器，通过认证后银行就可以安全的将这些用户消费划入商户对应的账户中。

基于USBKey的网上支付方案功能模块分为：USBKey模块、交易平台模块、支付网关模块和CA认证中心模块四大主要模块。系统的功能模块设计如图5.1。

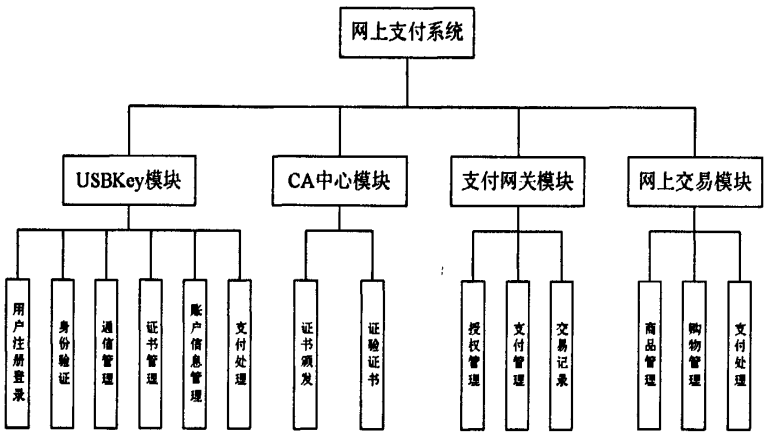

图5.1 基于USBKey网上支付方案的功能模块图

USBKey模块是系统组成的核心部分，主要功能包括：用户的注册登录管理、用户身份的认证、USBKey的通信管理、账户信息的管理，管理数字证书，支付处理等。

CA认证中心是系统中的重要参与方，它给用户、商家和支付网关提供数字证书并负责验证他们提交的数字证书的真实性来保证交易双方身份的可信性。CA认证中心模块采用B/S结构，持卡人、商家和支付网关通过浏览器就可以完成数字证书的申请、管理和验证等功能。CA认证中心模块按照功能可以细分为以下几个

子模块：证书申请、证书下载、证书发布、证书验证、证书存储、证书撤销。

支付网关的功能主要有两个，一个是接收商家的支付授权请求并转发到发卡行，然后把发卡行的授权响应结果转发给商家；另一个是接收商家的支付请款请求并转发到发卡行，然后把发卡行的请款响应结果转发给商家。

交易平台模块是用户和商家参与网上支付的主要工具。在平台中可以完成对商品的管理、购物的管理、支付信息的管理、个人账号的管理等。

5.2 支付流程

方案实现的总体支付流程如图5.2所示，其中涉及到的各个流程的具体实现将在下面逐一列出。

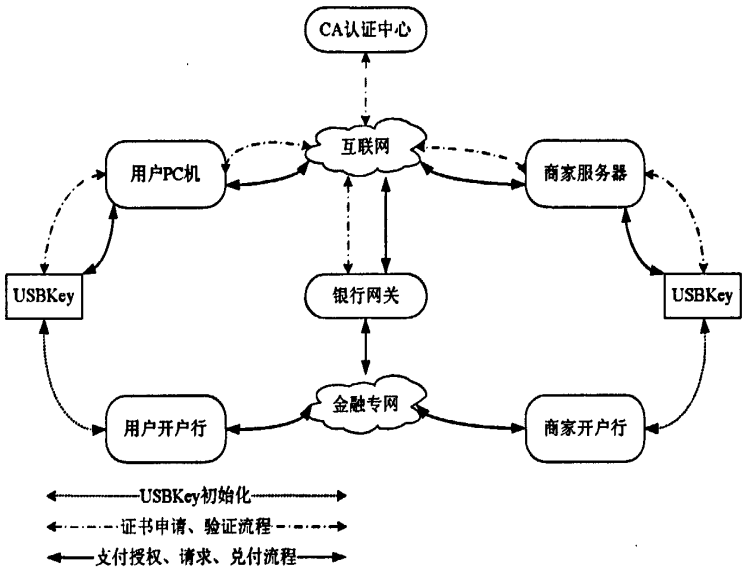

图5.2 基于USBKey的网上支付总体流程图

5.2.1 USBKey的授权

用户和商家的USBKey在使用前需进行授权设置，这一步骤必须在各自的开户行持本人有效证件完成。授权目的就是将USBKey与银行账户相关联，授权后的USBKey中应当包含个人资料、账户信息等关键信息，这时的USBKey在法律效力上等同于银行的账户，可以直接进行支付交易。

5.2.2 证书申请过程

(1) 用户得到授权设置的USBKey后登录网上银行填写个人资料并签名通过RA向CA中心请求颁发证书。这一步骤也可由银行代理完成，完成后银行直接给用户授权码即可，用户再用授权码进行证书的请求。

(2) CA中心经过资格和身份验证之后产生用户证书并对其用自己的私钥签名后发布；

(3) 用户再通过浏览器从网上下载证书，验证CA中心的身份及证书后保存至自己的USBKey中。

5.2.3 用户购物请求过程

(1) 用户浏览器在商家网站上选购商品，并生成购买订单，将购买请求消息R发往商家服务器系统；

(2) 商家服务器收到用户的购买请求消息R后，为用户的购买请求消息产生一个唯一的事务处理标识符，产生购买请求响应消息R'；

(3) 商家用自己拥有的USBKey将响应消息R'通过Hash函数得到数字摘要$h_{R'}=H(R')$；然后将这个数字摘要用自己的私钥y_d加密，形成数字签名$C_{R'}=h_{R'}^{y_d} \bmod n$，再将购买请求响应消息$R'$和数字签名$C_{R'}$及商家的数字证书打包一起发送给用户。

(4) 用户接收到商家的购买响应消息R'后，先验证商家的数字证书，再用户利用商家的公钥y_e验证商家对响应消息R'的数字签名，将接收到的购买响应消息R'通过Hash函数产生消息摘要$h'_{R'}=H(R')$，并将这个消息摘要$h'_{R'}$与解密后的购买响应消息签名$h_{R'}$做比较，来验证商家签名的正确性与完整性。

(5) 用户对商家的购买响应消息R'验证完成后（即双方交易达成一致后），便会自动产生一个购买订单消息O和一个支付消息P。用户用自己的USBKey对O和P进行双重数字签名，用Hash函数分别产生信息摘要$h_o=H(O)$和$h_P=H(P)$，然后将这两个信息摘要合并成一个新的消息M后再用Hash函数产生一个信息摘要$h_M=H(M)$；最后使用自己的私钥x_d将这个信息摘要h_M进行签名得到签名消息$C_M=h_M^{x_d} \bmod n$。

(6) 用户加密支付信息，利用USBKey产生一个随机的DES密钥k_1，用k_1加密支付消息P得到加密的支付消息P'。

(7) 产生支付数字信封，用户用银行网关的公钥z_e加密DES密钥k_1得到支付数字信封$S_{k_1}=k_1^{z_e} \bmod n$。

(8) 用户将购买订单消息O、签名消息C_M、加密的支付消息P'、支付数字信封S_{k_1}、支付指令消息P的信息摘要h_P随用户证书一起发送给商家服务器。

(9) 商家服务器收到这些信息后，验证用户数字证书合法性后，再使用用户公钥x_e对签名消息C_M解密得到$h_M=C_M^{x_e} \bmod n$；再将购买订单消息O通过Hash函数产生信息摘要$h_o=H(O)$；再对购买订单消息O的摘要消息h_o和接收到的支付指令消息P的信息摘要h_P进行合并得到新的消息M'并通过Hash函数得到摘要消息

$h_{M'}=H(M')$，再与解密后的原始摘要消息h_M做比较。

(10) 商家对购买订单消息O和支付消息P验证通过后即生成一个支付请求消息Q（其中包括交易ID号、支付金额、发货时间等交易信息），并利用自己的USBKey对这个支付请求消息Q进行数字签名得到签名消息$C_Q=h_Q^{y_d} \bmod n$。商家将支付请求消息Q和支付请求签名C_Q随商家数字证书一起发送给用户。

(11) 当用户接收到商家的支付请求消息Q后，验证商家数字证书及支付请求的数字签名C_Q，当用户完成确认后，至此用户的整个购物请求过程结束。

5.2.4 用户预支付过程

(1) 商家再将支付请求消息Q、支付请求消息的签名C_Q、用户的加密支付消息P'和支付数字信封S_{k_1}、订单消息O的信息摘要h_O随用户和商家签名证书发送给银行支付网关。

(2) 银行支付网关验证商家证书后，再利用商家的公钥y_e来验证商家签名的支付授权请求T。

(3) 银行支付网关验证用户证书后，用银行网关私钥z_d解密支付信封S_{k1}获得$k_1=S_{k_1}^{z_d} \bmod n$；再用$k_1$解密加密的支付消息$P'$得到支付消息$P$。

(4) 银行网关使用用户公钥x_e对支付消息P的双重签名消息C_M解密得到$h_M=C_M{}^{x_e} \bmod n$；再将支付消息P通过Hash函数产生信息摘要$h_P=H(P)$；再将支付消息摘要h_P和接收到的购买订单消息O的信息摘要h_O进行合并得到新的消息M''并通过Hash函数得到摘要消息$h_{M''}=H(M'')$，再与解密后的原始摘要消息h_M做比较。

(5) 银行网关对购买订单消息O和支付消息P通过验证后，将支付请求Q通过银行专网络发送到用户的开户行，开户行先对用户账户进行支出处理再等待用户从商家那里收到所购物品后再和商家进行兑付处理。

(6) 银行网关生成支付请求响应消息Q'，再用自己的私钥z_d进行签名得到签名消息$C_{Q'}=h_{Q'}^{z_d} \bmod n$。银行网关将支付响应消息$Q'$、支付请求签名$C_{Q'}$和网关签名证书一起分别发送给商家。

(7) 商家收到支付响应消息Q'后，先验证银行网关证书，再验证银行网关对支付请求响应消息的签名$C_{Q'}$。确认完成后并保存支付请求响应消息Q'，也就意味着用户的预支付过程的结束，此时商家便可以对用户进行发货处理。

5.2.5 商家兑付过程

(1) 当用户收到所购的物品后，产生一个收货确认消息V再用自己私钥x_d对收货确认消息进行签名，分别反馈给商家和银行网关，说明已经收到货物，银行可

以对商家进行兑付处理。

(2) 这时商家便可提请产生一个兑付请求W，并用自己的拥有的USBKey对兑付请求签名得到$C_W = h_W^{y_d} \bmod n$，将兑付请求W和签名消息C_W随商家证书一起发送给银行网关。

(3) 银行网关收到兑付请求W后，验证商户证书，再验证兑付请求的商家签名C_W。验证通过后，将兑付请求W通过金融专网发送到商家的开户行。由商家开户行和用户开户行之间进行结算。

(4) 银行网关再生成兑付响应消息，并签名。将兑付响应消息和兑付响应签名银行网关证书一起发送给商家。商家收到银行网关传来的兑付响应消息后，验证银行网关证书和兑付响应消息的签名。整个网上支付的交易过程至此结束。

5.3 方案分析

5.3.1 性能分析

(1) 方案中银行网关可看成是一个第三方平台，不仅可以进行交易中的仲裁，也可以起到第三方支付的作用，保证了商家和用户之间的公平交易，满足实物商品的原子性、确认发送原子性和合同原子性。该方案的实现不复杂，不会降低原有系统的运行效率，且在安全性方面有所提高。

(2) 利用USBKey设备能支持目前常用的支付工具，如银行卡、电子钱包、电子支票等，结合用户PIN实现了双因子认证，即可满足B2C交易中网上小额支付也可以满足B2B中大额支付的安全要求；

(3) 能够兼容现有网上支付系统并解决目前支付协议中密钥长度受限制的问题，USBKey中完全可以集成国产商用密码再结合DES、RSA等进行多重加密提高系统的安全性。

(4) 该方案还可以扩展至移动支付领域，将现有的USBKey设备改进成为SD卡形式的智能卡设备，可支持手机，PDA等便携式设备进行网上移动支付。

5.3.2 安全分析

(1) 不可否认性分析

整个方案的实现是在PKI体系下实现，交易各方均完全信任CA中心，并接受其进行身份认证。从协议的详细描述过程中可知，用户、商家和支付网关在进行消息发送时，都通过对方数字证书验证交易方的身份，而且交易者之间发送的消息也都用交易者的私钥进行了签名，能够有效防止任意一方对消息的发送的否认性，同时也方便确认消息发送者的身份。

(2) 不可伪造分析

在签名过程中采用单向散列Hash函数，对消息进行处理，若在传输过程中对消息进行任何修改，接收者都可以判断出消息的完整性。而且，任何一方没有其他参与者的私钥，无法以他人的名义伪造消息进行欺骗。

(3) 公平性分析

方案中提出了利用银行网关充当了第三方的角色，用户和商家完全信任银行支付网关。用户选中商品后，需先向银行网关发送预支付指令，银行网关验证后用户的预支付指令后并告知商家用户已付款，这时商家便可放心的给用户发送所购商品，防止了用户的虚假欺骗。若商家并不诚信没有给用户发送商品，那么支付网关也不会收到用户所授权的付款指令，这时商家就不能得到用户之前支付的商品货款。因此，该方案中如果交易中有一方有不诚实行为，都会由支付网关判断出来从而终止交易的，保证了交易双方的公平性。

(4) 安全性分析

方案采用RSA公钥密码体制，结合身份认证技术、数字签名技术和数字信封技术保证了支付方案的安全需求。方案中将用户的支付信息和购买订单信息等重要信息分别进行双重数字签名，使商家只能看到订单信息，银行只能看到支付信息，保护了用户购物信息的安全。

方案中的USBKey是一种集成安全芯片的设备，利用其进行身份认证和数字签名具有很高的安全保密性，具体的安全性能已在前章分析。

5.4 方案应用

5.4.1 银行卡中的应用

目前银行卡的网上支付基本都是基于网上银行进行交易的，在网上银行数据传输过程中，银行普遍采用128位密钥的SSL安全加密通信传常用版本为SSL3.0。SSL3.0通过数字签名和证书实现浏览器与Web服务器的双方身份验证，具有机密性、消息完整性、免重放攻击等功效。尽管该协议能够提供加密、认证等安全服务，但在SSL协议中只有商家服务器的认证是必须的，而用户端的认证则是可选的，因此不能解决用户与银行、用户与商家、商家与银行的多方身份认证问题。且SSL协议密钥长度受制于出口限制要求，随着攻击技术的发展其安全存在一定得风险。

通过分析可知基于银行卡的网上银行支付系统，其薄弱点在于客户端以及数据传输过程。目前攻击者攻击目标是在用户进行客户端操作以及系统数据传输过程中获取网上银行用户的隐秘信息。

在基于银行卡的网上银行支付系统引入USBKey支付方案，利用其特有的私

钥保密性，可以完全解决目前存在身份认证问题和数据加密密钥长度的问题。

(1) 可以利用USBKey设备的特点提供USBKey与用户PIN的双因子认证，也可以实现用户和服务器之间的动态身份认证方案。

(2) 利用USBKey中自带的加密算法，摆脱SSL加密密钥长度限制的问题，完全可以采用我国自有的国家商用密码进行加密。

(3) 解决了多方认证的问题，使其网上交易更加安全。

5.4.2 电子钱包中的应用

电子钱包可分为两种：一种是纯SET协议用户端软件，主要用于网上消费、帐户管理，这类软件通常与银行账户或银行卡账户是连接在一起的。电子商务活动中电子钱包的软件通常都是免费提供的。世界上有VISACash和Mondex两大在线电子钱包服务系统。另一种是用于小额支付的智能储值卡，持卡人预先在卡中存入一定的金额，交易时直接从储值帐户中扣除交易金额。如我们平时生活中使用的公交卡、购物卡等IC卡。

但是，目前的这两种电子钱包各有优缺点，前者需实时的依赖网络服务器才能完成网上支付，后者虽然可以实现脱机支付，但是基于卡的电子钱包不论在充值或支付时还需配置独立的读卡器，使其应用受到一定的限制。

基于USBKey的网上支付方案中，使用一种具有电子钱包功能的USBKey设备，不仅可以解决脱机支付的问题，也可以使其应用更加广泛、灵活。

具有电子钱包功能的USBKey其实就相当于在USBKey中充入了一定金额的电子现金，可在目前的USBKey基础上再集成一个电子钱包存储模块，使USBKey可以登录网上银行或在银行柜台自行办理充值业务，在使用时USBKey便可当做离线电子钱包进行支付。这一基于USBKey电子现金的离线支付方案将在下节将详细讨论。

该方案也满足电子钱包的基本安全需求，能够实现数字证书的管理、数字签名、交易数据的保存查询等功能。

5.4.3 电子支票中的应用

基于电子支票的电子支付手段要求传输资金和其他信息的高度安全性。在电子支票的支付中通常采用了多种安全措施来保证电子支票交易的安全，其中包括电子支票簿、防欺诈和私有性保障等。

本方案中我们可以采用USBKey来充当电子支票簿工具，不仅能够保障签名私有性，同时也增加了电子支票使用的方便性，解决了目前电子支票系统依赖智能卡和读卡器才能完成交易的不便。再者，USBKey上面还可以集成指纹、虹膜等

生物认证技术，大大的提高了电子支票系统的支付安全性和唯一性，可以使电子支票支付系统在 B2B 领域中的得到广泛应用，推进网上大额支付在电子商务中的发展。

USBKey 在电子支票系统中的应用可以完全兼容目前的现有的系统，只在客户终端对现有系统进行升级即可，在进行支付交易客户可利用 USBKey 能够自动产生电子支票，完成电子支票的填写后再用 USBKey 进行数字签名和加密。在电子支票系统中同样可以用 USBKey 进行离线支付。

5.5 基于 USBKey 电子现金的离线支付方案

5.5.1 概述

电子现金即数字现金，是电子商务模式下电子支付的有价凭证，它比传统现金更适合在数字化的网络背景下方便、实时地支付，是完全意义上的电子商务所追求的支付手段。其实质上是一种特殊的由银行签发的证书，用户根据特定的信息（嵌入的身份信息、面额及有效期等）产生电子现金的公私钥，银行对电子现金公钥进行签名。用户购物时出示电子现金，商家用银行的公钥验证电子现金公钥的合法性后要求用户用电子现金私钥对本次交易进行签名。商家存款时，将收到的电子现金及其相关交易信息发送给银行要求转账或兑换成现实中的纸币。

传统的基于智能卡的电子现金支付方案中，智能卡仅仅是安全的存储设备，核心的支付协议需要在作为智能卡的读写终端的服务器或 PC 机上实现，然而由于服务器或 PC 机都会连接至公共网络属于开放的不可信的平台，因此必须通过复杂安全协议来保证上面的数据和程序的安全。本节将通过利用 USBKey 设备来存储电子现金且利用 USBKey 进行电子现金的支付，提出一种新型的电子现金支付方式，不仅可以简化协议的复杂性，同时也支持电子现金的离线支付。

离线支付当一个用户在使用电子现金时，用户和商户的交易过程是以离线方式进行的，并不实时依靠银行的参与。

5.5.2 电子现金支付过程

基于 USBKey 电子现金的离线支付过程大致可分为四个阶段：

(1) 开户阶段

用户向其发卡银行申请 USBKey 的开户信息，银行对用户身份进行确认，验证通过后，银行向用户提供一个内部包含用户个人资料、账户信息等关键信息的 USBKey，这一过程依赖于传统的银行柜台模式进行。用户再通过网络浏览器对申请到的 USBKey 进行认证注册，并下载用户证书。商家的 USBKey 开户过程和用

户USBKey的开户过程基本相同。

(2) 存入阶段

用户通过USBKey向银行请求存入电子现金服务，双方相互进行身份验证后，同时协商产生临时会话密钥。用户在利用临时会话密钥和自己私钥组成数字信封将请求的电子现金数额发给银行，银行收到用户请求的电子现金数额后先进行解密验证，确认用户请求的安全性后再完成用户账户现金和电子现金的兑换，请求用户USBKey完成电子现金的存入。存入过程同样采用数字信封、数字签名技术，确保电子现金存入的安全性。电子现金中包含有银行的签名信息，用户是无法伪造的。

(3) 支付阶段

电子现金的支付过程，就是将用户USBKey中的一定数目电子现金转移到商家的过程。整个支付过程只由用户和商家就可完成，其间不需要银行的参与。实现了电子现金离线支付。利用电子现金进行购物的过程如下：

1) 用户使用浏览器在商家网上商城勾选商品生成购物请求；

2) 商家收到购物请求后进行确认商品的数量和价格等并用自己的私钥签名得到购物请求响应并后返回给用户；

3) 用户收到购物请求响应后验证商家身份并确认商品的数量和价格等，生成订单消息并用自己的私钥签名后返回给商家；

4) 商家收到用户的订单消息后，先验证用户身份，再确认用户订单消息，再生成支付请求消息并签名后发送给用户；

5) 用户收到商家的支付请求消息后，先验证商家身份，再确认商家支付请求，确认完成后用户便可通过口令取出USBKey要支付的电子现金，并用自己的私钥签名后连同用户发卡银行的证书一起发送给商家；

6) 商家得到用户的电子现金后，通过用户证书验证用户电子现金签名的完整性，再通过银行证书验证电子现金的真伪，是否存在用户伪造电子现金的可能。此时，商家得到的电子现金中包含有银行和用户共同的签名信息，也是无法伪造的。

商家收到用户电子现金验证通过后，便可向用户发送所购货物。至此，电子现金的支付过程就此结束，不过这里只讨论了双方交易，假设商家是诚信的，收到用户的电子现金后会履行合约按时保质保量发货的。也可引入第三方支付平台来监督双方的交易行为。

(4) 兑换阶段

兑换阶段是在商家和银行之间完成。无论是商家还是用户，只要持有合法的USBKey，就可以凭借自己数字证书通过银行将USBKey中的电子现金兑换为普通货币或者将电子现金存入自己普通帐户，且都无法伪造电子现金。

5.5.3 方案分析

该方案与目前传统的网上支付方案相比，脱离了必须实时依靠银行的支付方式，可以随时随地的进行离线支付，不过方案的缺点就是若USBKey丢失或其内部电子现金的丢失和现实中的纸币丢失一样都将是无法追回的，只能应用于小额支付。

从方案的安全性来说，满足了电子现金不可伪造、不可重复使用，不可否认等安全需求。

方案只是提出了一种基于USBKey的电子现金离线支付的可行性，对其具体的详细实现过程没有深入，如电子现金的可分性、不可追溯等这些问题已经在一些文章中进行了详细的探讨[37][38]，这里并没有涉及。

5.6 基于USBKey身份认证的技术实现

基于USBKey网上支付方案与传统的网上支付方案相比增加了USBKey设备并结合数字证书来实现身份认证和数字签名等功能。而在整个身份认证的过程中已经包含了数字签名过程。因此，这一小节将重点对基于USBKey的网上支付方案中身份认证技术进行具体实现。

5.6.1 技术平台

(1) USBKey设备

USBKey设备采用的是海泰方圆的一款基础型产品Haikey，该产品支持对称加密算法DES/3DES、非对称加密算法RSA以及国产分组算法；支持双因子认证，在使用时要求用户输入PIN码，同时提供数字证书进行认证，安全性高；采用硬件实现数字签名；Windows的PC/SC标准驱动程序、免驱两种驱动方式可选；这款产品可以运行在Windows 98以上操作系统的计算机上，采用标准USB接口，传输速率可达480Mbps。

(2) 软件平台

软件开发采用的编程语言是C++，编译环境使用的Visual Studio 2010。数字证书的产生和数字签名的实现调用了微软提供的CryptoAPI接口，图形界面的制作采用MFC作向导。对Haikey的二次开发要引用海泰公司提供的“HKAPI.h”头文件及“HKAPI.dll”动态链接库，调用CryptoAPI接口要引用“wincrypt.h”头文件。

微软的CryptoAPI的功能是为应用程序开发者提供Windows环境下使用加密、验证等安全服务时的标准加密接口。CryptoAPI接口允许调用函数来加密数据，交换公钥，散列一个消息来建立摘要以及生成数字签名。微软提供的CryptoAPI由简

单消息函数（Simplified Message Functions）、底层消息函数（Low-level Message Function）、基本加密函数（Base Cryptographic Functions）、证书编解码函数（Certificate Encode /Decode Function）和证书库管理函数（Certificate Store Function）五部分组成。其中前三者用于对敏感信息进行加密或签名处理，从而保证网络传输信息的保密、防篡改、防抵赖等，后两者是对证书的操作，实现身份认证。

(3) 数字证书格式

一般数字证书的格式采用X.509标准。X.509是被广泛使用的数字证书标准，是由国际电联电信委员会（ITU-T）为单点登录（SSO-Single Sign On）和授权管理基础设施（PMI-Privilege Management Infrastructure）制定的PKI标准。X.509定义了(但不仅限于)公钥证书、证书吊销清单、属性证书和证书路径验证算法等证书标准。X.509格式的证书被VeriSign、微软、网景和其他许多公司广泛应用于对电子邮件消息进行签名，对程序代码进行认证，以及对许多其他类型的数据进行认证等等。X.509证书格式见图5.3。

版本号
序列号
签名算法ID
发行者名称
有效期
主体名称
主体公钥
发行者唯一标识
主体唯一标识
扩展
签名

图5.3 X.509证书格式

5.6.2 具体步骤

本节将按流程介绍基于Haikey的身份认证技术的实现。共分为五个模块：

(1) CA根证书的生成

在为用户生成数字证书前必须先创建CA的根证书。安装根证书意味着对这个CA认证机构的信任。创建CA根证书的流程如图5.4，这里假设CA只有一层。

创建的根证书的主体名为RootCA，保存在本地磁盘的根证书文件名为rootcert.cer。为证书分配公私钥对时，要调用CryptAcquireContext函数获得系统CSP的密钥容器句柄，用函数CryptGetUserKey或CryptGenKey获取或生成密钥对。程序最后要用函数CryptReleaseContext释放CSP句柄，CertCloseStore断开系统证

书库。

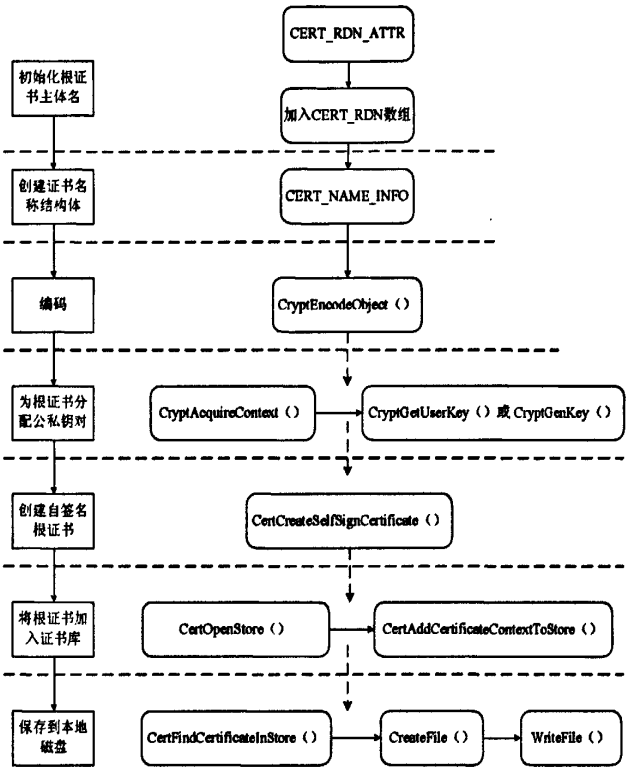

图 5.4 CA 根证书创建

(2) Haikey 初始化

Haikey 作为用户端先要进行初始化，主要工作为产生公私钥对，也包括获取 Haikey 的设备信息等。

1) 连接 Haikey 设备，获得对 Haikey 操作的句柄,用到函数 HK_ConnectDev;

2) HK_BeginTransaction 开始对 Haikey 进行操作，独占设备;

3) 获取设备信息，调用函数 HK_GenerateRSA 产生密钥对;

4) HK_EndTransaction 结束独占设备;

5) 断开设备，用到 HK_DisconnectDev。

(3) 用户证书申请

用户输入“姓名”、“国家”、“省份”、“城市”和“电子邮箱”，公钥已在之前的 Haikey 中生成，此时，会自动添加，显示的为 16 进制 der 编码形式，完成对证书的申请。

程序内部将用户信息建立成系统可接受的证书申请格式。该部分功能主要在 EncodedReqInfo 函数中实现。程序流程如图 5.5。“生成证书请求信息”模块，用户的每一条信息即是用户证书请求的一项属性。各项属性组成属性列表，作为用户证书请求信息的名称部分。用户的公钥信息导入到证书请求的公钥信息成员中，

添加随机数和Haikey对其的私钥签名作为附加信息，以便CA对证书请求的合法性进行验证。最后该请求保存在本地磁盘，文档名为“cert_req_info.txt”。

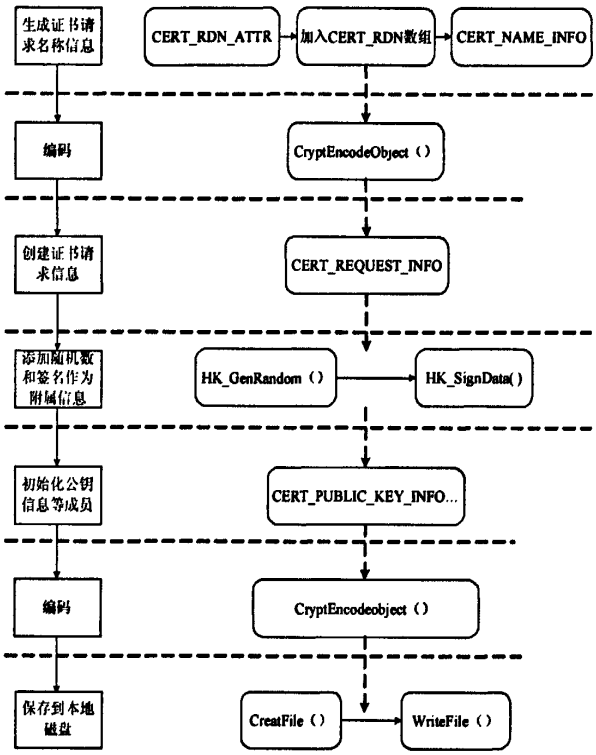

图5.5 证书请求格式生成

(4) 证书验证及颁发

CA在颁发用户证书之前先要验证证书签名请求的合法性及完整性，即CSR验证。分为三步：

1) 调用CryptDecodeObject函数对证书签名请求进行解码，解码内容放入CERT_REQUEST_INFO结构体；

2) 导出用户证书请求中的公钥信息，导入CA相应的CSP密钥容器；

3) 从附属信息中导出随机数和用户的私钥签名；

4) CryptDecrypt函数利用公钥信息进行签名验证。

签名验证无误后，CA开始颁发用户证书。用户证书的创建类似于根证书的创建。具体流程为：

1) 打开系统证书库，查找根CA，并创建上下文环境；

2) 根据解码后的用户请求信息，初始化证书结构CERT_INFO，填写证书版本号、起始日期、终止日期、公钥信息等。证书颁发者为根CA；

3) 调用CryptSignAndEncodeCertificate函数对证书结构签名和编码；

4) 调用CertCreateCertificateContext函数根据编码后的证书数据由CA根证书

创建用户证书上下文；

5) 调用 CertAddCertificateContextToStore 将证书加入系统证书库。

(5) 身份认证

身份认证分为对数字证书的验证和对证书持有人身份的验证两个过程。

1) 对数字证书的验证可分为证书拆封和证书链验证。

拆封证书

调用 CryptDecodeObject 函数解码证书，CertOpenSystemStore 函数打开系统证书库，用 CertFindCertificateInStore 函数在系统证书库中查找根证书，并获取根证书的公钥信息，调用 CryptVerifyCertificateSignature 函数验证用户证书。

证书链验证

声明和初始化 CERT_CHAIN_ENGINE_CONFIG 结构体；为创建一个新的非缺省的链引擎做准备；

调用 CertCreateCertificateChainEngine 为应用程序创建一个新的非缺省的链引擎；

调用 CertOpenSystemStore 打开系统证书库；

CertEnumCertificatesInStore 枚举证书库中的证书；

对每一个枚举得到的证书，用 CertGetNameString 获得其名称；

用 CertGetCertificateChain 对每一个证书创建证书链；

查看证书链上每一证书的状态，证书链上每一证书都要合法。

2) 对证书持有人的验证

服务器生成伪随机序列

CryptAcquireContext 获取 CSP 密钥容器句柄，调用 CryptGenRandom 函数产生随机序列 M，放入内存 Unin；

生成摘要值

Haikey 内的 HK_MD5 函数对 Unin 中的随机序列 M 进行 MD5 算法杂凑，生成杂凑值 $H(M)$ 放入 ucDigest；

用户签名过程

调用函数 HK_SignData 对 ucDigest 的 $H(M)$ 进行用自己的私钥进行签名得到签名值 $C=H(M)^d \bmod n$，再将随机序列 M 和签名 C 及用户的数字证书用数组的形式打包发送给服务器；

服务器验证签名

服务器先用 CryptImportKey 将用户证书中的公钥导入 CSP 密钥容器，再调用 CryptCreateHash 函数生成 Hash 对象，调用 CryptHashData 对随机序列 M 生成摘要值 $H'(M)$，再用 CryptVerifySignature 公钥解密签名，同时比较 $H(M)$ 和 $H'(M)$ 的值是否一致来判断签名的完整性和有效性。若验证成功，则该用户端 Haikey 为合

法的。

同样，用户也可以使用相同的方法对服务器进行身份认证，这样通信双方之间便完成了基于USBKey和数字证书的身份认证。该方式也可以结合PIN口令完成双因子认证。

5.7 小结

本章在前面研究分析的基础上提出了一种基于USBKey的网上支付方案，给出了该方案的具体流程，并对方案的性能和安全性进行了分析。再将方案结合现有的网上支付系统，分别研究了基于USBKey的网上支付方案在银行卡、电子钱包、电子支票网上支付系统中的新应用，并逐一进行了可行性分析，同时也提出了一种基于USBKey电子现金离线支付的方案。最后基于海泰方圆的一款基础型产品Haikey，结合数字证书实现了基于USBKey身份认证的功能。

第六章 总结与展望

随着经济的发展和信息基础设施的不断完善，电子商务得到了快速的发展。而电子商务中的网上支付的安全性和通用性是目前制约电子商务发展的两大主要障碍。当前的网上支付中，普遍使用的仍旧是在网上银行中基于银行卡的支付手段，其他支付手段由于其自身的缺点应用受到一定得限制。另外，随着一些电子支付安全案件的发生，网上支付业务的安全问题也日益受到关注。

本文的重点是在研究分析网上支付系统和 USBKey 的基础上提出了提出一种新的网上支付方案，方案不仅能够满足安全的需要也能够兼容目前常用的网上支付工具，提高了网上支付工具的通用性。研究取得的成果如下：

(1) 研究了 USBKey 体系结构，并对其攻击与防护进行逐一研究分析。结合 USBKey 的实际应用安全情况提出了一种改进的用户 PIN 输入方案；

(2) 重点研究分析了现有网上支付系统，基于 USBKey 提出一种新的网上支付方案，并详细介绍了支付方案的支付流程；

(3) 对方案的性能和安全性进行了分析，最后结合现有网上支付工具给出了其各种实际应用方案。

(4) 基于海泰方圆的一款基础型产品 Haikey，结合数字证书实现了基于 USBKey 的身份认证技术。

本文研究的不足之处就是提出的方案只是一个概括性的方案，也可以说是一种思路，对其中方案实施的具体细节性问题没有深入探讨。这是后续的要研究的工作。

总之，随着信息安全技术的不断发展，USBKey 等智能设备将会以其安全、便携等独特之处在电子商务中得到广泛的应用，促进电子商务朝更加安全可靠的方向发展。

致 谢

首先，要感谢我的导师——马文平教授，在本文的研究与撰写过程中得到马老师了的悉心指导，使我学会了如何用所学的知识去解决实际的问题能力。马老师一丝不苟的工作作风、分析问题的能力以及敏锐的洞察力对我的学习和今后的工作给予了莫大的帮助和启发。他严谨的学风、诲人不倦的精神令我终身难忘。他那正直无私的品格、豁达的胸襟、严谨的学术修养对我以后的生活树立了榜样。再次感谢马老师对我谆谆的教导和孜孜不倦的言传身教，这一切都将让我终生受益！

感谢实验室的秦好磊、杨康、刘维博、贺国强、姬国珍、王余刚、何平平、寇振华、冯赟、杨元原、高胜、何叶锋、陈和风、余旺科、傅佩龙、徐明、殷浩、张晓、陈秋丽、刘宝成、冯佳、郭娜等所有的兄弟姐妹们，在读研的学习生活中，正是在他们的精诚团结下为我创造了一个温暖的大家庭，给了我许多帮助。

感谢我的舍友杨丹丹、王玲玲，感谢他们对我日常生活和学习上的关心和帮助。

感谢我所有的朋友们，是他们在我最困难的时候帮助我，在我最失落的时候给予我关怀。有了他们使我的生活更加精彩，谢谢！

最后要感谢我的家人，感谢他们对我的鼓励与支持！

参考文献

[1] CNNIC. 第 26 次中国互联网络发展状况统计报告, 2010.

[2] 中华人名共和国电子签名法, 2004.

[3] 唐晓东. 电子商务中的信息安全. 北京: 清华大学出版社, 2006.

[4] 张卓其, 史明坤. 网上支付与网上金融服务. 大连: 东北财经大学出版社, 2002.

[5] 林松. 电子支付安全体系结构的研究与实现. 四川大学博士论文, 2005.

[6] Master Card and VISA. SET Secure Electronic Transaction Specification, Bookl: Business Description Version l.0, May, 1997.

[7] David Wagner, Bruce Schneier. Analysis of the SSL3.0 protocol. The Sencond USENXI Workshop on Electronic Commerce Proceedings, USENIX Press, Nov, 1996.

[8] 翟东锴, 徐孟春, 赵欣等. 一种加强 SSL 协议安全性的解决方案. 计算机应用与软件. 2005, 22(6).

[9] 牛云, 罗守山. 安全电子交易协议的一种改进. 计算机工程与设计. 2004, 25(7).

[10] 冯登国, 裴定一. 密码学导引. 北京: 科学出版社, 1999.

[11] National Institute of Standards and Technology. Data Encryption Standard(DES), 1999, FIPS 46-3.

[12] W. Difie, M. E. Hellman. New Directions in Cryptology. IEEE Transactions on Information Theory. pp.644-654, 1976, 22(6).

[13] R. L. Rivest, A. Shamir, L. Adleman. A method for obtaining digital signatures and public key cryptosystem. Communications of the ACM.. pp.120-126, 1978, 21.

[14] R. L. Rivest, RFC1320: The MD4 Message-Digest Algorithm. 1992.

[15] FIPS 180-1. Secure Hash standard. NIST, US Department of Commerce, Springer-Verlag.Washing DC, 1996.

[16] 管有庆. 电子商务安全技术. 北京: 北京邮电大学出版社, 2007.

[17] 关振胜. 公钥基础设施 PKI 及其应用. 北京: 电子工业出版社, 2008.

[18] ISO/IEC 959428, ITU-T (formerly CCITT) Information technology-Open systems Interconnection-The Directory: Authentication Framework Recommendation X. 509.

[19] 程宇贤. 网上银行身份认证系统的安全性研究. 上海交通大学硕士论文, 2009.

[20] 曹喆. 基于 USBKey 的身份认证机制的研究与实现. 东华大学硕士论文, 2010.

[21] Wolfgang Rankl, Wolfgang Effing. 智能卡大全: 智能卡的结构. 功能. 应用. 北京: 电子工业出版社, 2002.

[22] USB 相关资料和协议. http://www.usb.org.

[23] Stefan Mangard, Elisabeth Oswald, Thomsa Popp. Power Analysis Attacks. Springer-Verlag, 2007.

[24] Paul Kocher, joshua Jaffe, Benjamin Jun. Differential Power Analysis. M. Wiener, Eds. Advances in Cryptology-CRYPTO'99. LNCS, 1666, pp.388-397, Springer, 1999.

[25] Jean-Jacques Quisquater, David Samyde. Electromagnetic analysis (EMA): measures and countermeasures for smart cards. Lectures Notes in Computer Sciencevol. 2140, pp. 200-210, 2001.

[26] A. Behrouz, Forouzan. 密码学与网络安全. 北京: 清华大学出版社, 2009.

[27] Douglas R. Stinson. 密码学原理与实践. 北京: 电子工业出版社, 2003.

[28] W. Chor, Alexi, b.z., Goldreich, O.,Schorr,C.P.. RSA and Rabin functions: certain parts are as hard as the whole. SIAM Journal on Computing, Vol.17, No.2, pp,194-209, Apr. 1988.

[29] E. Biham, A. Shamir. Differential cryptanalysis of DES-like crypto systerns Journal of Cryptology, V01.4, No.1, pp.3-72, 1991.

[30] X. Y. Wang, D. G. Feng, X. J. L, H. B. Yu. Collisions for Hash Functions MD4, MD5, HAVAL-128 and RIPEMD, Rump Session of Crypto'04, Eprint, 2004.

[31] K. Nishimura, M. Sibuya. Probability to Meet in the Middle, Journal of Cryptology. No.2, pp.13-22, 1990.

[32] 王明波, 张海松. 基于指纹加密的 USBKey 安全方案. 微计算机, 2009, (25).

[33] 刘守义. 智能卡的技术. 西安电子科技大学出版社, 2004.

[34] 浪潮信息安全事业部. 我国网上支付安全现状与发展趋势. 信息安全与通信保密, 2010, (2).

[35] E. Foo, C. Boyd, W. Caelli, et al. A Taxonomy of Electronic Cash Schemes. In: L. Yngstron, J. Carlsen eds. Proceedings of the IFIP TCII 13th International Conference on Information Security. Copenhagen, Denmark. London: Chapman & Hall, Ltd. 1997.

[36] 董威. 多智能卡新技术研究. 北京邮电大学博士论文, 2008.

[37] 彭冰. 离线电子现金的协议研究. 华中科技大学博士论文, 2004.

[38] 马臣云, 王彦. 精通 PKI 网络安全认证技术与编程实现. 北京: 人民邮电出版社, 2008.

[39] 宋坤, 刘锐宁等. MFC 程序开发参考大全. 北京: 人民邮电出版社, 2007.

作者读研期间参与的科研项目

参与的科研项目：国家 863 项目“密码算法和安全协议检测分析技术与测评系统”。

项目编号：2007AA01Z472