摘要

在世界范围信息化趋势中，信息安全问题成为学术界和工业界共同关注的焦点。密钥管理是解决信息安全问题的关键。然而，当前密钥管理系统很少从全局和系统的角度出发，所以从系统整合的角度提出了广义密钥管理思想。

广义密钥管理定义密码系统的安全因子 $S = F_m(Alg,Ptl,Key)$，F_m 是密钥管理技术的参考因子，密码算法 Alg、密码协议 Ptl 和密钥 Key 是密钥管理技术的实现因子。密码算法是基本的安全要素，它确保了原子级的信息保密性、完整性和不可否认性。密码协议是应用的安全逻辑，它确保了应用级的信息安全，是整个信息系统安全的框架。密钥是密钥管理的核心。密码管理技术通过分析物理环境，选择基本安全算法和协议，建立管理策略，规划整个系统安全结构，参照密码管理参考因子，统筹安排密码管理实现因子，把握密钥管理系统的结构关键，提高系统整体安全性能。

基于广义密钥管理思想，联系智能卡资源条件和离线应用环境约束的具体实际，构造基于 PKI 协议的智能卡离线公钥管理模型，它倾向于从理论的角度讨论 PKI 框架下公钥管理功能的实现，涉及五个实体，包括密码系统的建立，签名验证机制，证书和密钥的相应操作，CRL 的维护和获取，以及信任关系建立等相关内容。同时，按照公钥管理模型，实现了离线状态下智能卡具体的密钥管理。在给出了签名校验机制的基础上，实现了离线状态下的静态数据鉴别和动态数据鉴别机制，用于建立智能卡和终端间的信任关系，同时通过用户 PIN 保护实现访问控制管理，以及对终端的密钥管理的安全需求进行处理等。

总之，引入了系统整合的观念，提出了广义密钥管理思想，给出了广义密钥管理体系结构，在考虑智能卡资源约束和离线状态约束的前提下，从全局和系统的角度构造了基于 PKI 协议的智能卡离线密钥管理模型，并在实际的智能卡离线应用环境中予以实现，达到保护信息系统安全的目的。

关键词： 智能卡；PKI；密码系统；密钥管理系统；离线模型

Abstract

In the worldwide trend of information, information security becomes the focus that academe and industry both pay attention to. The secret management is the key of resolving information security problems. However the current secret management system seldom sets out from the whole system, so the paper puts forward the generalized secret management ideology with the system idea.

The security factor of cryptography could be defined as: $S = F_m(Alg,Ptl,Key)$ in the generalized secret management, the symbol F_m means secret management reference factors, algorithms, protocols and key are secret management implementation factors. Algorithms are base security factors, they ensure privacy, integrity and accountability of infomations at the atom level. Protocols are applications' security logics, they ensure information security at the application level, and they are framework of the whole information security. Key is the kernel of secret management. Cryptogram technology selects basic secure algorithms and protocols, sets up management policy, plans the entire security system after analysing the physical environments, consults the reference factors, plans secret management implementation factors as a whole, and holds the structure key of secret management to improve the whole security performance.

Thinking about resources of smart cards and off-line application environment restricts based on those secret management ideology, the off-line model of secret management of smart cards is constructed. It concerns the implement of secret management under PKI framework from the theory that constructing the off-line model of secret management based on PKI, involved five entities, including the creation of cryptogram System, signing and verifying, the operations of certificates and keys, maintaining CRL or retrieving CRL, and creating trust relations so on. After that, secret management in the off-line status is concretely implemented with refering to the public-key management model. Based on signing and verifying *mechanisms, it*

implements the static data authentication and dynamic data authentication in the off-line status about the smart card to set up trust relations between the smart cards and the terminals, realizes accessing control with users PIN protections, and handles terminal security requires for secret management.

In a word, the paper has introduced the system idea, put forward the generalized secret management ideology, gived the generalized secret management architecture, constructed the off-line model of secret management of smart cards based on PKI, and impleted it in real off-line application about smart cards to ensure information system secure in the architecture point of view after thinking about resources of smart cards and off-line application environment restricts.

Key words: Smart Card; Public Key Infrastructure; Cryptogram System; Secret Management System; Off-line Model

1 绪论

1.1 课题背景

当今社会信息技术的迅猛发展，特别是计算机技术、网络技术、通信技术和信息安全技术的发展，使人们的生活方式和生产方式发生了深刻的变化。信息产业部副部长吕新奎在'98 国际电子商务论坛中曾指出：国民经济信息化，企业信息化是基础，金融电子化是保证，信息安全是关键。信息技术在给人们的工作和生活带来便利的同时，也给人们平添了许多困扰。安全问题首当其冲，密钥管理是解决信息安全问题的关键。

密钥管理实现对密钥的生成、分发、保存、使用、注销和更新等密钥操作的管理，它在信息安全中占据着十分重要的地位[1]。密钥管理不仅影响系统的安全性，而且涉及到系统的可靠性、有效性和经济性。

1.1.1 密钥管理的目标

密钥管理的目标就是确保信息安全。一般来说，信息安全包括访问控制、通讯安全和信息监察，具体包括信息传输的安全、信息存储的安全以及对网络传输信息内容的审计三个方面，如表 1-1：

表 1-1 密钥管理目标

信息安全	信息传输安全（动态安全）	数据加密	数据完整性的鉴别	防抵赖
	信息存储安全（静态安全）	数据库安全	终端安全	
	信息的防泄密	信息内容审计		
	用户	鉴别	授权	

a) 信息传输安全

信息传输安全是一种动态的安全。它是通过数据加密、数据完整性的鉴别和防抵赖等手段来实现的。这是论文讨论的一个重点。至于在智能卡系统中如何实现信息传输安全以及相关的密钥管理工作，论文后面作了具体的讨论。

b) 信息存储安全

信息存储安全是一种静态的安全。它包括数据库的安全存储[2]和终端安全。在智能卡应用系统中一些敏感信息存储在智能卡介质和终端中，所以论文着重介绍了敏感信息如何分发到智能卡和终端中，并且说明了这些信息对密钥管理所发生的作用。

c) 信息的防泄密

信息的防泄密是对信息内容的审计和信息传递的跟踪监察。通过审计和跟踪监察及时发现安全问题或安全漏洞，并提供补救措施，将损失降低到最低限度。

d) 访问控制

访问控制就是对用户访问权限的控制，包括授权和鉴别。

1.1.2 密钥管理的基本任务

a) 制定管理策略

制定密钥管理策略是根据信息系统的安全要求从宏观角度确定信息系统安全体系，包括系统结构和规模，各个结构元素的功能，以及在管理上需要完成的任务。同时，根据体系结构确定使用的密码管理体制，选择密码算法和密码协议，以完成基本的密钥管理任务。

b) 选择密码算法

选择密码算法时应该考虑算法的业界标准化、公开化、先进性，强度和模式等属性，以及实现时的资源要求等。

c) 选择密码协议

选择密码协议应该考虑选择公开的标准协议。他们都是经过密码专家分析和实际项目试验的协议。

d) 基本管理任务

密钥管理的基本任务就是管理密钥的生成、保存、传递、废止和更新以及相应

的证书管理等。

1.2 国内外概况

1.2.1 国内外密钥管理研究现状

在智能卡技术应用研究领域，比较著名的国外公司有 Gemplus、Schlumberger 和 EMVCo.公司，国内公司有北京握奇数据股份公司 WatchData、深圳明华澳汉科技股份有限公司和武汉天喻信息股份有限公司等。

Gemplus 公司长期进行智能卡产品的开发和推广应用，具有很强的研发实力，被誉为世界智能卡技术产品第一供应商。Gemplus 开发了自己的密钥管理系统——GemKeyWiz(Gemplus Key Management System)，允许对密钥的产生、传递、存储和转移进行手工控制。GemKeyWiz 是针对电信业务开发的，同时推出了相关的安全访问控制模块 SAM MPCOS-EMV。

Schlumberger 公司是一个经营石油、智能卡和电信等多种业务的跨国集团。在智能卡应用领域也有相对强的实力，主要致力于多应用安全智能卡电子商务解决方案。为公共电话和停车场开发了密钥管理系统——KeyOps(Key Open Platform System) 是一个用于安全控制的模块化的灵活的密钥管理系统。在 KeyOps 系统中允许发卡方管理整个安全流程。

EMVCo.公司是有三大信用卡提供商 Europay、MasterCard 和 Visa 在 1999 年共同创建的，旨在管理、维护和增强用于支付的 EMV 集成电路卡规范。EMVCo.公司在密钥管理领域没有开发具体的密钥管理系统，主要是进行理论研究，提出符合广泛需求的密钥管理规范。EMVCo.公司密钥管理规范：EMV2000 Integrated Circuit Card Specification for Payment Systems Book2 – Security & Key Management 2000（12），被许多智能卡技术开发机构所才用，差不多成为业界的标准。密钥管理规范是 EMVCo. 公司制定的系列规范之一，并随着技术的发展不断更新。该规范阐明密钥管理的一些相关细节和实现策略，具有一定的指导作用。

至于国内的智能卡技术开发公司，例如北京握奇、深圳明华和武汉天喻公司，密钥管理的研究基本上处于相同状态，主要是借鉴 EMVCo. 公司制定的密钥管理规

范，根据自己的需要，制定相应的密钥管理方案，没有开发出广泛适应的密钥管理系统产品。只是根据客户的售后服务要求，协助用户开发适合用户要求的具体的密钥管理系统。

根据使用的密码体制来划分，密钥管理大致可以分为对称密钥管理和公开密钥管理两种密钥管理机制。因为不同的密码体制，使用不同的密码算法，具有不同的密钥生成、分发、使用和更新等机制，所以在管理方式上也大相径庭。

在对称加密算法密钥管理方式中，通讯双方拥有相同的秘密密钥，他们之间的会话就是在这个密钥的基础上进行的。会话秘密完全寓于秘密密钥之中，密钥的泄漏必将导致整个安全体系的瓦解。下面以 Kerberos 协议[3-5]为例说明对称加密算法密钥管理的特点。

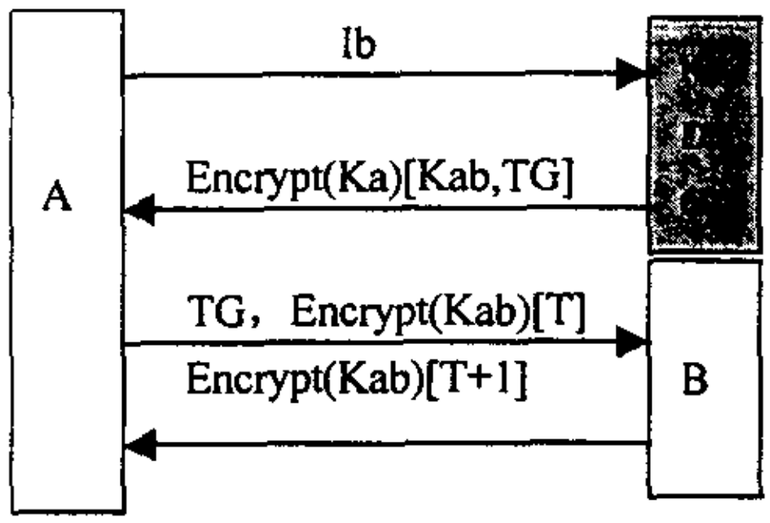

图 1-1 对称加密算法密钥管理模型

图 1-1 是一个简化的 Kerberos 模型示意图， KDC 为密钥分配中心，具有认证和授权服务的功能，它保存了各个实体的秘密密钥，A、B 为两个实体，Ia、Ib 分别是实体 A、B 的标识符，Ka、Kb、Kab 分别为实体 A、B 的秘密密钥和两者之间的会话密钥，T 为时间戳。TG=Encrypt(Kb)[Ia,Kab]就是票据，用来传递 Kab 给实体 B。T 的任务就是在两者之间实现会话通道的建立。秘密寓于票据之中，A 和 B 并不知道对方的密钥。整个模型的安全由 KDC 保证，Kab 由 KDC 在通讯请求时产生，通过票据 TG 传递、分发，会话结束后自动销毁。A 和 B 之间建立通讯联系的过程被简化成图 1-1 描述的四个步骤[6]。

以上 Kerberos 采用了 KDC 实现密钥的管理，但在 IC 卡系统中采用这种方案不符合 IC 卡具体应用要求。IC 卡应用采用了一个比较巧妙的方法，既实现了对称加密算法密钥管理，又避免了建立 KDC，优化了整个系统体系，方便了 IC 卡的应

用。

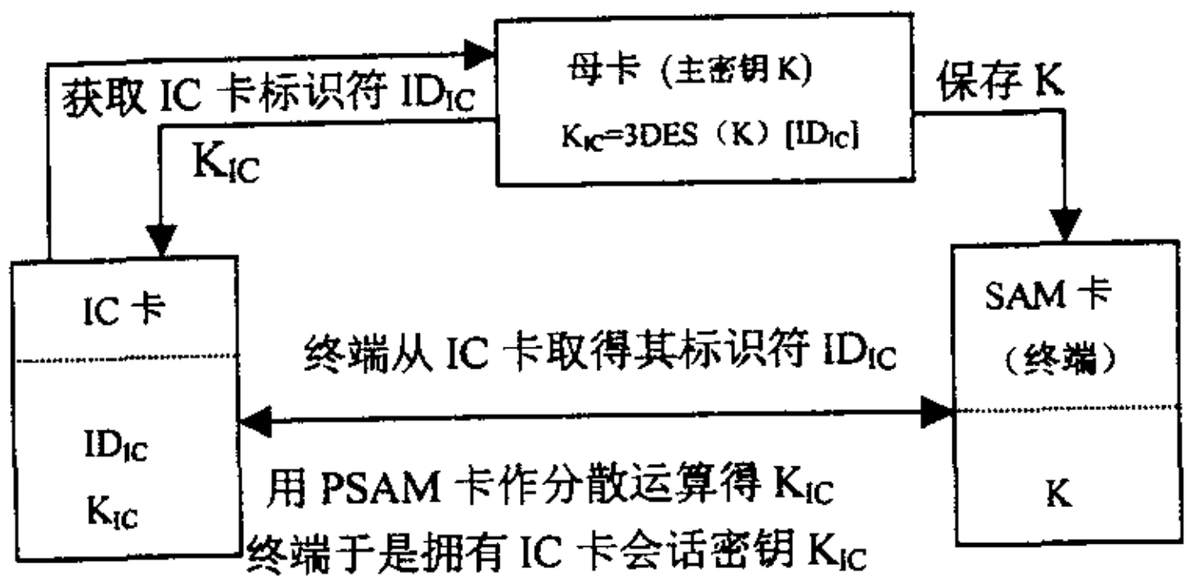

图 1-2 对称加密算法密钥管理实现

如图 1-2，K 为母卡主密钥，ID_{IC} 为 IC 卡标识符，K_{IC} 为 IC 卡主密钥，且 K_{IC} = 3DES（K）[ID_{IC}]。总的思路是密钥分散，即母卡通过使用自身携带的密钥 K，对 IC 卡的唯一标识号 ID_{IC} 进行 3DES 加密，得到的子密钥 K_{IC} 作为 IC 卡的主密钥。在进行加密信息传递应用时，IC 卡使用主密钥 K_{IC} 对信息进行加密后,密文传递给终端；终端为了解密密文首先从 IC 卡取得其标识符 ID_{IC}，接着用 SAM 卡密钥 K（即保存到 SAM 卡中的母卡主密钥）作分散运算得 K_{IC}，然后终端使用密钥 K_{IC} 解密密文得到信息。IC 卡并没有向终端传递 K_{IC}，而是传递 ID_{IC}，K_{IC} 在 SAM 卡内部生成，IC 卡和终端 SAM 卡拥有共同密钥 K_{IC} 进行信息传递[6]。

相对对称密钥管理，公钥密钥管理比较简单。公开密钥管理又称作非对称密钥管理。在公开密钥管理中，每一个用户有一个各不相同的名字，一个可信的证书认证中心（CA）给每个用户分配一个唯一的名字并签发一个包含名字和用户公开密钥的证书[7]。

如果甲和乙通信，他首先必须从数据库中取得乙的证书，然后对它进行验证。如果他们使用相同的 CA，甲只需验证乙证书上 CA 的签名；如果他们使用不同的 CA，甲必须从 CA 的树形结构底部开始，从底层 CA 往上层 CA 查询，一直追踪到同一个 CA 为止，找出共同的信任 CA。

证书可以存储在网络中的数据库中。用户可以利用网络彼此交换证书。当证书撤销后，它将从证书目录中删除，然而签发此证书的 CA 仍保留此证书的副本，以

备日后解决可能引起的纠纷。

如果用户的密钥或 CA 的密钥被破坏，从而导致证书的撤销。每一个 CA 必须保留一个已经撤销但还没有过期的证书废止列表（CRL）。当甲收到一个新证书时，首先应该从证书废止列表（CRL）中检查证书是否已经被撤销。

持证人甲向持证人乙传送数字信息，为了保证信息传送的真实性、完整性和不可否认性，需要对要传送的信息进行数字加密和数字签名[8-12]。

1.2.2 国内外智能卡密钥管理的缺陷和不足

对称加密算法密钥管理具有速度快和易于实现的优点，但是不具备防抵赖的能力[13]。而实现防抵赖功能是信息交互的基本要求之一，非对称加密算法密钥管理能很好地解决这个问题[14]。对称加密是基于共同保守秘密来实现的。采用对称加密技术的通讯双方必须要保证采用的是相同的密钥，要保证彼此密钥的交换是安全可靠的，同时还要设定防止密钥泄密和更改密钥的程序。这样，对称密钥的管理和分发工作将变成一件潜在危险的和繁琐的过程。通过公开密钥加密技术实现对称密钥的管理使相应的管理变得简单和更加安全，同时还解决了纯对称密钥模式中存在的可靠性问题和鉴别问题[15]。

公开密钥管理应用数字证书技术、签名校验技术和加密技术，理论上比较完善，实际应用却不多，原因在于公开密钥管理在实现上需要相应的政策和配套设施。不但要制定一个合理的密钥管理策略，规划一个完善的密钥管理体系，而且要有一个值得信赖的功能齐全的证书认证中心[14]。

无论那种密钥管理，其研究都仅局限于狭义的管理范围，很少从系统的角度作广义的探讨和理解，而且在智能卡领域尚没有成熟的公钥管理模型，以及在公钥管理模型下相关的离线密钥管理问题。

1.2.3 密钥管理发展趋势

a) 规范化

目前国际有关的标准化机构都着手制定关于密钥管理的技术标准规范。ISO与IEC下属的信息技术委员会(JTC1)已起草了关于密钥管理的国际标准规范。

该规范主要由 3 部分组成：第 1 部分是密钥管理框架；第 2 部分是采用对称技

术的机制；第 3 部分是采用非对称技术的机制。该规范现已进入到国际标准草案表决阶段，并将很快成为正式的国际标准。

b) 公开化

从系统的整体角度来看，密钥管理技术包括密码算法、密码协议和管理技术[16]。密码算法和密码协议在密码体制中都有一个从秘密到公开的发展过程，现在的系统大都采用公开的密码算法和密码协议。因为密码算法和密码协议的公开接受实践的检验和完善，有助于提高信息系统的安全性。

c) 互通性

信息交流是现代信息社会发展的要求，信息安全问题就是在信息互通的过程中产生的。为解决信息交流的安全问题，各种组织按照自己内部的需要，制定组织内部安全策略和方案，实现组织内部信息的安全交流。随着信息社会的发展，这种区域限制给信息交流带来了诸多不便。在信息系统安全设计时，互通性是要求考虑的一个方面。

1.3 课题主要研究工作

1) 提供新的视角

a) 从狭义走向广义

对传统密钥管理忽视的方面，进行重新思考和界定。重新思考它们对密钥管理所发挥的作用，重新界定它们在密钥管理中所处的地位。在系统整合中，它们不再显得无足轻重，它们得到应有的角色。

b) 从分散走向整合

不再以分散的孤立的眼光对待各种相关的安全因素，而是从系统的角度，整合各种安全因素，以构建一个完备的系统。

c) 从局部走向全局

不再局限于个别的安全因素，个别的安全因素固然重要，但是它们要服从全局的安全需求，同时不能忽视那些微不足道的安全因素，千里之堤，溃于蚁穴，就系统的角度来说，它们是必不可少的。

d) 从静态走向动态

建立密钥管理系统不可能一劳永逸，应该能够动态地适应技术发展、攻击监测和跟踪审计带来的挑战。而且，系统的安全是一个相对的安全状态，系统外界的任何变化均可以打破这种安全上的平衡，所以系统要具有能够从不平衡到平衡的动态适应能力。

e) 从封闭走向开放

密钥管理系统要不断地吐故纳新，能够感知外界的变化，并作出应对的策略。

2) **构造一个抽象模型**

基于新的视角和研究方法，将以往密钥管理忽视的安全因素纳入系统进行重新整合，给出一个广义密钥管理体系结构。在此基础之上，面对智能卡资源条件和离线应用环境，构造出一个基于 PKI 协议的智能卡离线密钥管理抽象模型。

3) **实现具体的离线密钥管理**

根据抽象模型，在智能卡离线应用环境中具体实现了密钥管理功能，确保离线应用环境的安全。

4) **目标实现及其意义**

为了改变以往密钥管理存在的范畴狭隘、因素分散局部和系统静态封闭的局限性，论文在第二部分提出了广义密钥管理思想，融合广义、整合、全局、动态和开发的观念，建立相应的密钥管理体系结构，把密码协议、密码算法、资源条件、技术发展、密码分析、试验测试、攻击、监测和审计等相关因素纳入密钥管理范畴，参考以上各要素，从整体上把握密钥管理。并且，可以根据密钥管理系统外界的因素变化，例如技术发展、攻击、监测和审计等变化，作出动态的反应，使得系统处于吐故纳新的开放状态。基于这种体系结构，考虑智能卡资源限制和离线交易环境，论文第三部分在 PKI 协议的基础上，构建了“智能卡＋PKI 协议”的离线模型，并对模型细节，例如证书的签名校验、分发、保存、获取等进行了讨论。同时，论文在第四部分具体实现了离线应用环境下密钥管理的部分功能，实现了签名校验机制，离线静态数据鉴别，离线动态数据鉴别和 PIN 码保护，这些都是以证书的管理和应用为基础的。

2 密钥管理的体系结构与方案思路

密钥管理具有自身的结构要素和功能模块。各种结构要素组合而成密钥管理系统；各个功能模块的协调配合才完成了最终的密钥管理功能。任何结构要素必须在密钥管理系统中才能发挥其应有的功能特性；任何单独的功能模块只能实现密钥管理的局部功能，只有相互支持配合才可以完成全局的密钥管理功能，否则密钥管理系统安全性能就会降低。所以，本章需要从系统整合的角度综合密钥管理各个因素，构建广义密钥管理体系结构，为智能卡密钥管理模型的构架提供指导和依据。

2.1 狭义密钥管理

狭义密钥管理指的是只实现了局部功能的密钥管理，系统具有了基本的密钥管理特性。现在大多数密钥管理系统应该被归为这个层次。狭义密钥管理没有系统和全局的观念，在应用系统安全管理中处于从属地位。狭义密钥管理孤立地看待各个安全元素，虽然也可以完成应有的安全操作，但是系统没有整合优势。对于狭义密钥管理的安全因子可以定义为式 2.1：

$$S = Fm(Key) \tag{2.1}$$

其中，Key 指密钥，Fm 指密钥管理技术参考因子，包括资源条件、技术发展、密码分析、试验测试、攻击监测和跟踪审计等因素。

2.2 广义密钥管理

广义密钥管理是相对于局部密钥管理提出的。广义密钥管理使用系统和全局的观念来看待密钥管理要素的作用，而且在应用系统安全中处于核心地位。在广义密钥管理中，系统具有安全结构和功能模块。各个安全元素既分工完成各自的安全功能，又互相配合以实现整个应用系统的安全特性。广义密钥管理的安全因子可以定

义为式 2.2：

$$S = Fm(Alg, Ptl, Key) \quad (2.2)$$

其中，Alg 指密码算法，Ptl 指密码协议，Key 指密钥。Fm 具有和狭义密钥管理相同的含义，只是在广义密钥管理中 Fm 加强了反馈和指导的作用。

引入了系统整合的观念，使得密钥管理具有了更加清晰的层次结构，更加明确的目标导向，而且对从宏观的角度把握密钥管理具有指导意义。

2.3 广义密钥管理体系结构

广义密钥管理具有开放式的体系结构，如图 2-1 所示：

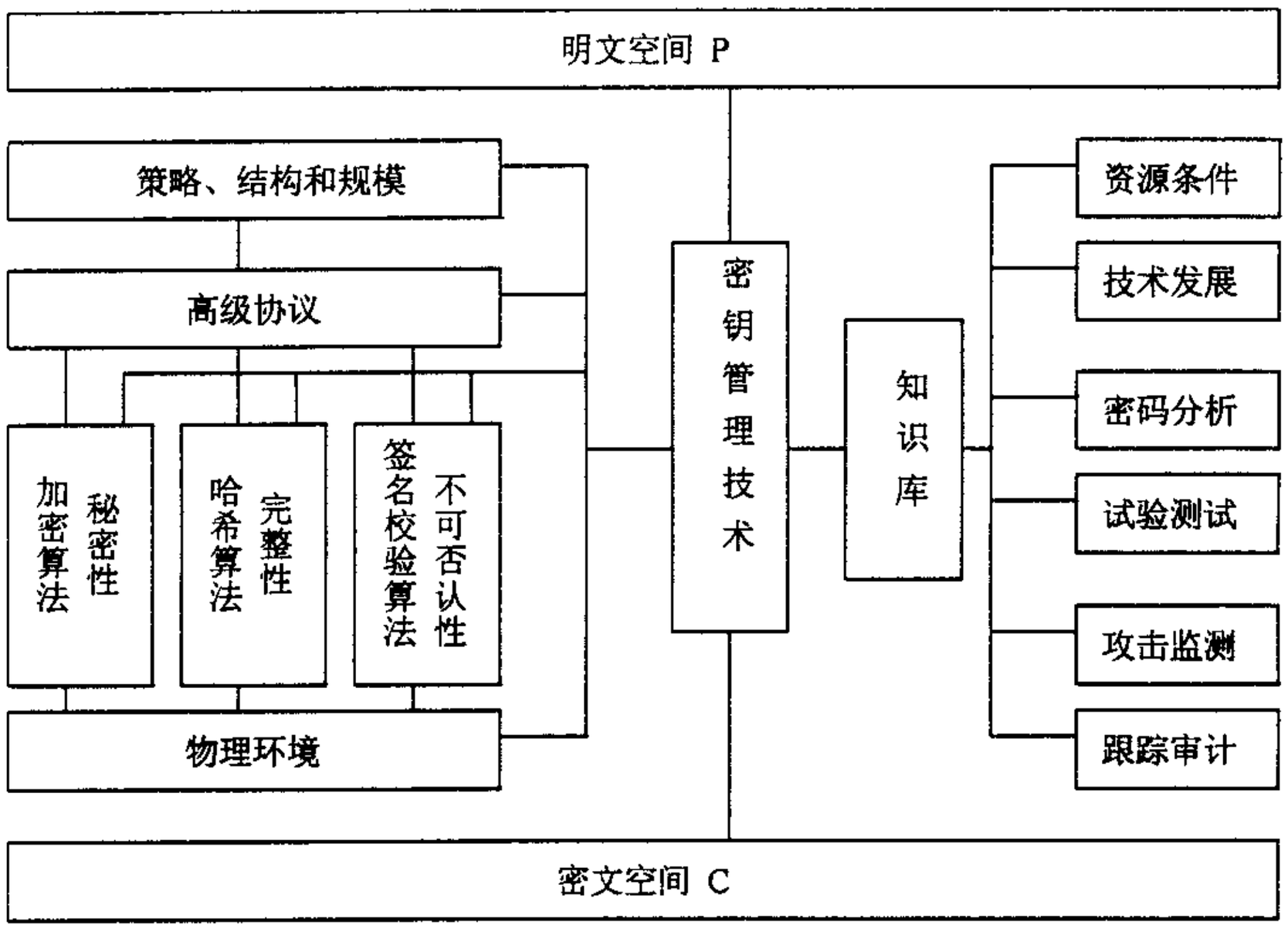

图 2-1 广义密钥管理体系结构

2.3.1 体系结构的结构要素

体系结构大体上可以分为三个组成部分：明文空间（P）、密文空间（C）和密钥管理模块（S）。三者之间的关系可以用等式 2.3 表示：

$$C = S(P) \text{ 或者 } P = S^{-1}(C) \qquad (2.3)$$

密钥管理模块是明文空间 P 和密文空间 C 之间的置换因子，也是整个应用系统的安全核心，位于密钥管理体系的中心地位。

应用系统的所有明文组成明文空间 P，所有的密文组成密文空间 C。

密钥管理模块又由两个主要部分构成：密钥管理技术实现因子和密钥管理技术参考因子。其中，密钥管理实现因子包括应用系统的物理环境、密码算法、高级协议及应用体系结构、策略和规模等；密钥管理参考因子包括资源条件、技术发展、密码分析、试验测试、攻击监测和跟踪审计等因素。在密钥管理参考因子中，资源条件和密码分析相对稳定，属于稳定因子；其它因子处于动态变化之中，属于活动因子。稳定因子和活动因子一起形成了密钥管理的知识库。

2.3.2 体系结构的功能分解

在建立应用体系的密钥管理系统之初，要对应用系统的物理环境进行分析，包括系统采用什么样的硬件，系统采用的硬件能力如何，以及硬件计算能力对算法有怎样的要求等。分析的结果放入知识库作为资源条件因子；由资源条件因子来选择硬件支持的密码算法，并对密码算法进行密码分析，存入知识库作为密码分析因子。物理环境制约了应用系统的体系结构、规模和安全策略，同时高级协议的选择依赖于体系结构、策略和密码算法等，密码管理技术据此初步建立密钥管理系统。随着技术发展、试验测试、攻击监测和跟踪审计对知识库的更新，密钥管理技术不断完善密钥管理中的密码算法、协议、体系结构和策略等要素，使得密钥管理系统臻于完美。

物理环境是密钥管理系统赖以生存的基础，也是设计密钥管理系统的起点，对密码算法、协议和应用体系具有决定意义。

密码算法是基本的安全要素，它是整个信息安全的基础。密码算法确保了原子级的信息保密性、完整性和不可否认性，它实现了最低级的安全需要。密码算法通常采用公开的经受住许多次和各种方式攻击考验的安全算法，在特定的计算环境下理论上和实践上均是安全的。

密码协议是应用的安全逻辑，它是整个信息安全的框架。密码协议确保应用级的信息安全，它是高级的应用安全规范。在某种意义下，密码协议是商业逻辑，是

解决问题的规则[17,18]。例如 PKI 协议和电子商务的 SET 协议[19]。在密码技术的管理下，应用密码算法基本安全元素构造面向应用的解决现实问题的可操作的高级应用安全规范。密码协议也是公开的，接受实践的检验。密码算法只有被密码协议使用，才具有了应用价值和实际意义；密码协议也只有使用了密码算法，才具有了安全的特性。密码协议扩展了密码算法的安全功能，使密码算法在应用级发挥用武之地。

应用体系结构、规模和策略是对具体应用的构架密钥管理系统的宏观把握，影响密钥管理系统的宏观特性，例如可扩展特性等。

知识库的稳定因子建立起最初的密钥管理系统，知识库的活动因子调节密钥管理系统，使其从一个不稳定和不安全的状态过渡到相对稳定和安全的状态。

2.3.3 体系结构的特征

体系结构具有以下六个方面的特征：

1) 系统性：使用系统整合的观点来组织系统密钥管理要素，发挥系统的组织优势。
2) 开放性：密钥管理系统不断地吐故纳新，使其处于相对稳定状态。
3) 层次性：密钥管理具有较为明显的层次特点，从物理底层、算法层次、协议层次到宏观结构和策略，具有清晰的结构层次。
4) 可操作性：密钥管理系统可以按照计划有步骤的建立起来，而且更强调实际的构建过程。
5) 可评估性：每种要素都有相应的评估标准，同时可以建立相应的参考知识库。
6) 动态适应性：每次系统的外界变化，无论是新技术的产生、技术发展、试验测试的突破还是攻击监测等所引发的密钥管理系统处于相对不安全状态，密钥管理系统都会调节自身的功能模块以适应系统外界的动态变化，保持自己处于相对稳定安全的状态。

2.4 密钥管理中的木桶效应

木桶效应指的是由长短不同的木块围绕而成的一个木桶，其容量由最短的那块木块决定。因此，最短的木块是木桶容量的关键。

在密钥管理系统中，密钥管理系统就是一个木桶，其薄弱环节就是决定密钥管理系统安全性的关键。

信息安全大致经历了算法安全、协议安全到密钥安全的演变过程。

在算法安全阶段，通信实体之间的安全薄弱环节是由安全算法造成的。此时的安全算法都是专用的和秘密的，安全算法的泄漏将会导致信息安全体系的瓦解。因此，这种算法是受限制的算法，而它们的保密性远远不够。大的或经常变换用户的组织不能使用它们，因为如果一个成员离开这个组织，其它成员必须换用其它算法。如果有人无意暴露了秘密，所有人必须改变他们的算法。受限制的算法不可能进行质量控制或标准化。每个用户组织需要有自己的唯一算法，所以不可能采用流行的硬件或软件产品，用户不得不设计自己的算法并予以实现。所以，算法必然走向公开化的道路，采用公开化算法的密码系统的安全性将不再依赖于算法的细节[14,20,21]。

在协议安全阶段，通讯实体之间的安全薄弱环节是由安全协议造成的。此时的密码协议是通信实体间的秘密协议，因此协议是受限制的协议[17]。受限制的协议具有和受限制的算法一样的弱点。所以，协议也会走向公开化，采用密码协议专家制定的严谨、缜密、实用和公开的标准协议[22]。

在密钥安全阶段，通信实体之间的安全取决于密钥的安全。密钥管理成为系统的薄弱环节，因此，需要构建缜密的密钥管理系统。

2.5 密钥管理的方案思路

广义密钥管理体系结构给出了制定密钥管理方案的整体框架和实现思路，确定智能卡密钥管理方案不能脱离这个框架和思路，在联系具体的智能卡资源和应用环境下，要充分考虑密钥管理体系结构的各个结构元素及其功能实现，发挥系统整合的优势。

密钥管理的木桶效应思想强调在制定密钥管理方案时要关心结构效应，要优化系统结构，合理配置资源，避免产生某个薄弱环节，造成系统整体安全性能的下降。

根据广义密钥管理思想和密钥管理木桶效应，制定健全的智能卡密钥管理方案，思路上首先要分析实施系统的特点和要求，考虑智能卡的硬件环境和离线应用环境，不应该从每个孤立的系统元素角度进行局部设计，而应该从如图 2-1 系统体系的高度加以解决，参考智能卡密钥管理技术知识库及相关参考因子，例如资源条件、技术发展、密码分析、试验测试、根据监测和跟踪审计等，统筹安排密码算法、密码协议和密码技术等密码管理实现因子，把握密钥管理系统的结构关键，提高系统整体安全性能。

在此基础上论文选择了“智能卡＋PKI 协议”的密钥管理模型方案。在智能卡物理资源中实现了密码算法和协议的支持，同时选择 PKI 协议支持离线分布应用环境。

3 基于 PKI 协议的智能卡离线公钥管理模型

本章的目的在于建立一个基于 PKI 协议的智能卡离线公钥管理模型，为下一章 PKI 构架下的智能卡离线公钥管理提供实现蓝本。根据前面提出的广义密钥管理体系结构和智能卡密钥管理方案思路，本章在分析了智能卡密钥管理的硬件环境和离线应用环境基础上，同时考虑体系结构的稳定因子，建立了智能卡离线公钥管理模型，并对模型进行了一定的分析。

3.1 智能卡密钥管理环境

3.1.1 硬件环境

硬件环境主要是指智能卡。智能卡具有自己的 CPU、ROM、RAM 和 EEPROM 资源，相对于一般的 PC 机来说，智能卡的处理能力十分有限，存储容量也比较小，一般只有几十 K。智能卡 COS（Chip Operation System）系统掩膜在 ROM 中，保证代码的安全，一般的 COS 系统只有十几 K。RAM 中存放中间的命令参数和返回结果等，COS 将用户的过程密钥生成后放在 RAM 中，掉电后自动丢失，保证其安全性。EEPROM 一般存放创建的应用系统和用户数据，EEPROM 被加密逻辑保护，在满足用户规定的安全条件时，可以进行读写操作。有的智能卡 CPU 带有 RSA 运算协处理器，这样的 CPU 支持模数较大的 RSA 算法。因此，这些硬件特性决定了密钥管理系统所采用的密码算法，也决定了密钥管理系统会在效率和安全强度之间进行折中。

3.1.2 离线应用环境

当前智能卡大都是工作在离线状态下，不可能要求在交易过程中去寻求 CA 和发卡方的联机支持，因为这样在效率上是不允许的。离线环境主要是由用户所持的智能卡和商家安装的终端组成，只有两者相互信任，方可安全使用，所以选择了 PKI

协议来建立分布式实体之间的信任关系。

3.2 智能卡离线公钥管理模型

离线模型可以简单概括为：

离线模型 = 智能卡 + PKI 协议[23]

智能卡组成了物理基础，同时实现了基本的安全密码算法；PKI 协议是根据分布式应用环境要求而确定的应用系统结构。

基于 PKI 的智能卡离线密钥管理模型[23-28]如图 3-1 所示：

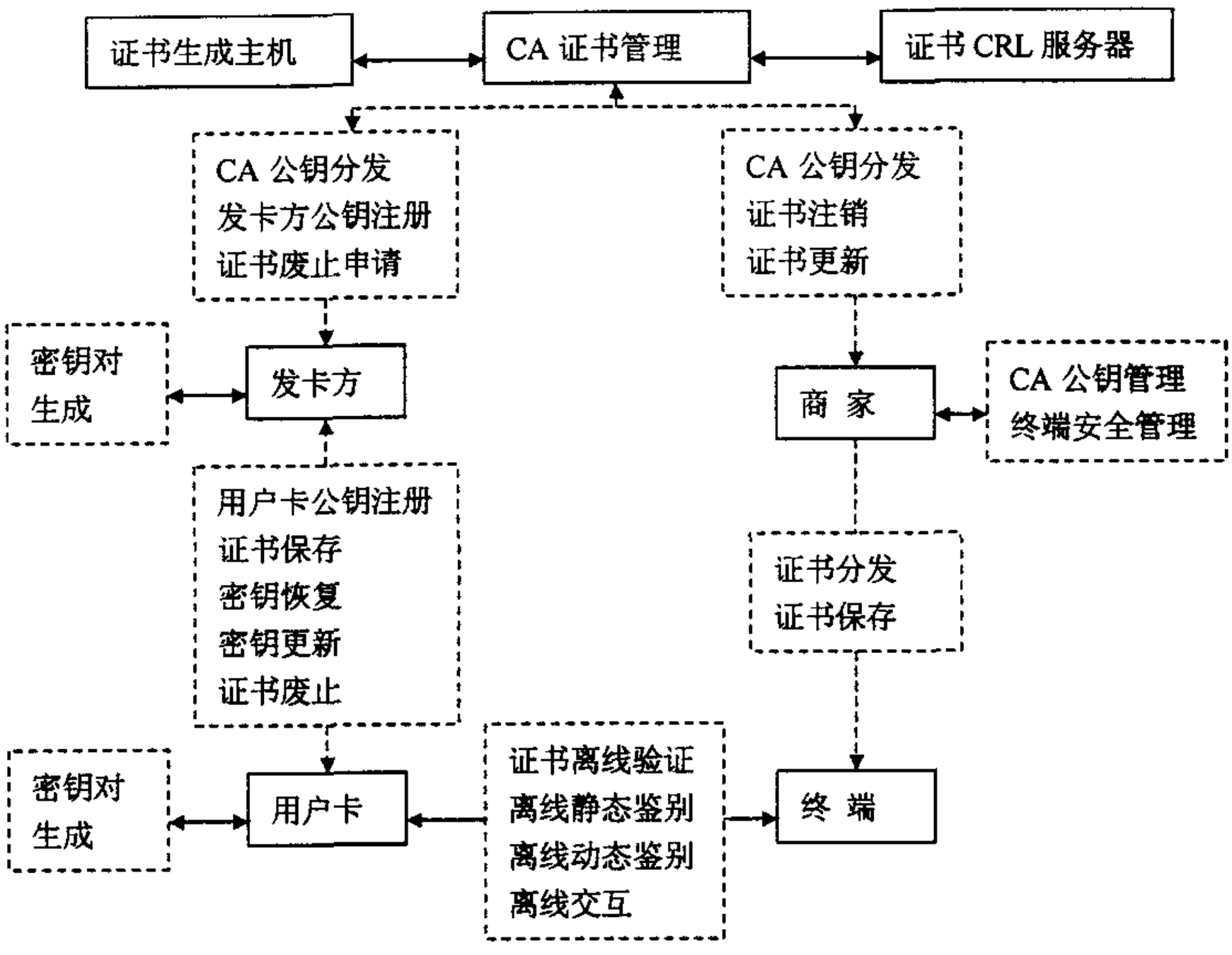

图 3-1 智能卡离线公钥管理模型

智能卡离线公钥管理模型，涉及到 CA、发卡方、用户卡、商家和终端五个角色，包括 CA 公钥的产生和分发，发卡方的注册，证书的获取、验证、保存、废止、

注销和更新，密钥的生成、更新和恢复，CRL 的获取，和信任关系的建立等功能。下面对其中部分给予了说明。

3.2.1 签名和验证

在 PKI 体系中，用户卡和终端对信息和文件的签名，以及对数字签名的认证是很普遍的操作。

PKI 成员（CA、用户卡和终端等）对数字签名和认证可以采用多种算法的，如 RSA，DSA 等等，这些算法可以由硬件、软件或硬软结合的加密模块(固件)来完成，智能卡系统中大都是在 COS 中用软件实现的。

在 4.1 节将会对基于 IC 卡的具体的签名与鉴别机制作详细的说明。

3.2.2 证书的获取

在验证信息的数字签名时，终端必须事先获取用户卡的公钥证书，以对信息进行解密验证，同时还需要 CA 和发卡方的公钥证书，以便验证用户卡身份的有效性。

证书的获取可以有多种方式：

1) 发送者发送签名信息时，附加发送自己的证书；
2) 从获取的相关信息中恢复得到证书；
3) 建立单独发送证书信息的通道；
4) 从访问发布证书的目录服务器处获得；
5) 从证书的相关实体(如 RA)处获得；

在 PKI 体系中，可以采取某种或某几种的上述方式获得证书。

在 4.2.3.1 节和 4.3.3.1 节将会说明在 IC 卡系统中如何获取 CA 公钥证书。

在 4.2.3.2 节和 4.3.3.2 节将会说明在 IC 卡系统中是如何通过证书恢复获取发卡方公钥证书的。

3.2.3 证书的验证

验证证书的过程是迭代寻找证书链中下一个证书和它相应的上级 CA 证书的过程。 用户检查证书的路径是从最后一个证书(即用户已确认可以信任的 CA 证书)所签发的证书有效性开始，检验每一个证书，一旦验证后，就提取该证书中的公钥，用于检验下一个证书，直到验证完发送者的签名证书，并将该证书中包括的公钥用

于验证签名。

在使用每个证书前，必须检查相应的 CRL。

在 4.2.3 节静态数据鉴别和 4.3.3 节动态数据鉴别中可以知道在 IC 卡应用中是如何验证证书的。

3.2.4 证书的保存

保存证书是指 PKI 实体在本地储存证书，以减少在 PKI 体系中获得证书的时间，并提高证书签名的效率。

在存储每个证书之前，应该验证该证书的有效性。PKI 实体可以选择存储其证书链上其他实体所接收到的所有证书，也可以只存储数字签名发送者的证书。

证书存储单元应对证书进行定时管理维护，清除已作废的或过期的证书及在一定时间内未使用的证书。证书存储数据库还要与最新发布的 CRL 文件相比较，从数据库中删除 CRL 文件中已发布的作废证书。

证书存储区存满之后，一般采取删除最少使用的那些证书(LRU)。

在 IC 卡模型中，证书（链）及相关数据大都存放在终端和 IC 卡中（见第 4.2.1 节、4.2.2 节、4.3.1 节和 4.3.2 节）。

3.2.5 证书的废止

当 PKI 中某实体的私钥被泄漏时，被泄密的私钥所对应的公钥证书应被作废。

对 CA 而言，私钥的泄密不大可能，除非有意破坏或恶意攻击所造成；对一般用户而言，私钥的泄密可能是因为存放介质的遗失或被盗。

另外一种情况是证书中所包含的证书持有者已终止或与某组织的关系已经中止，则相应的公钥证书也应该作废。

终止的方式[29]：

1) 如果是密钥泄露，证书的持有者以电话或书面的方式，通知相应的 CA；

2) 如果是因关系中止，由原关系中组织方面出面通知相应的 ORA 或 CA。

处理过程：

如果 ORA 得到通知，ORA 应通知相应的 CA，作废请求得到确认后，CA 在数据库中将该证书记上作废标志，并在下次发布 CRL 时加入证书作废列表，并标

明作废时间。在 CRL 中的证书作废列表时间有规定，过期后即可删除。

在 IC 模型中通常使用更新的方式解决私钥的泄漏，使用卡片回收的方式解决关系中止。

3.2.6 密钥的恢复

在密钥泄密、证书作废后，为了恢复 PKI 中实体的业务处理和产生数字签名，泄密实体将获得(包括个人用户)一对新的密钥，并要求 CA 产生新的证书。

泄漏密钥的实体是 CA 的情况下，它需要重新签发以前那些用泄密密钥所签发的证书。

每一个下属实体将产生新的密钥时，获得 CA 用新私钥签发新的证书，而原来用泄密密钥签发的旧证书将作废，并被放入 CRL。

在具体做法上可采取双 CA 的方式来进行泄密后的恢复，即每一个 PKI 实体的公钥都由两个 CA 签发证书，当一个 CA 泄密钥后，得到通知的用户可转向另一个 CA 的证书链，可以通过另一个 CA 签发的证书来验证签名。这样可以减少重新产生密钥时和重新签发证书的巨大工作量，也可以使泄密 CA 的恢复和它对下属实体证书的重新发放工作稍慢进行，系统的功能不受影响。

3.2.7 CRL 的获取

每一个 CA 均可以产生 CRL，CRL 可以定期产生也可以在每次有证书作废请求后，实时产生，CA 应将其产生的 CRL 及时发布到目录服务器上去。

CRL 的获取[29]就可以有多种方式：

1) CA 产生 CRL 后，自动发送到下属各实体；

2) 大多数情况是：由使用证书的各 PKI 实体从目录服务器获得相应的 CRL。

在 IC 模型中，CA 或发卡方证书被作废，则实体所属域均要重新更新，所以 CRL 使用不多。

3.2.8 密钥的更新

在密钥泄密的情况下，将产生新的密钥和新的证书。其中有一个问题要引起注意：密钥的更换时间，无论是签发者或是被签者的密钥作废时间，要与每个证书的有效截止日期保持一致。

如果 CA 和其下属的密钥同时到达有效期截止日期，则 CA 和其下属实体同时更换密钥，CA 用自己的新私钥为下属成员的新公钥签发证书；

如果 CA 和其下属的密钥不是同时到达有效截止期，当用户的密钥到期后，CA 将用它当前的私钥为用户新的公钥签发证书；而 CA 密钥先到达截止日期时，CA 用新私钥为所有用户的当前公钥重新签发证书。

不管哪一种更换方式，PKI 中的实体都应该在密钥截止之前，取得新密钥对和新证书。在截止日期到达后，PKI 中的实体便开始使用新的私钥进行对数据的签名，同时应该将旧密钥对和证书归档保存。

3.2.9 审计

PKI 体系中的任何实体都可以进行审计操作，但一般而言是由 CA 来执行审计，CA 保存所有与安全有关的审计信息。如：产生密钥对、证书的请求、密钥泄漏的报告和证书中包括的某种关系的中止等等。

3.2.10 存档

出于政府和法律的要求以及系统恢复的需要，CA 产生的证书和 CRL 应被归档，作历史文件保存。另外有关文件和审计信息出于调整或法规的规定也需要存档。

3.2.11 PKI 体系的互通性

随着互联网程度的提高，世界范围内将出现多种多样的证书管理体系结构。所以，PKI 体系的互通性也不可避免地成为 PKI 体系建设时必须考虑的问题，PKI 体系中采取的算法的多样性更加深了互通操作的复杂程度。

PKI 的互通性首先必须建立在网络互通的基础上，才能保证在全球范围内在任何终端用户之间数据的传送；其次是用户必须借助于 X500 目录服务获得对方签名使用的算法。PKI 在全球互通可以有两种实现途径：

1) 交叉认证方式

需要互通的 PKI 体系中的根 CA 在经过协商和政策制定之后，可以互相认证对方系统中的根 CA。

认证方式是根 CA 用自己的私钥为其他的需要交叉认证的根 CA 的公钥签发证书。

这种认证方式减少了操作中的政策因素，对用户而言，也只在原有的证书链上增加一个证书而已。但对于每一个根 CA 而言，需要保存所有其它需要与之进行交叉认证的根 CA 的证书。

2) 全球建立统一根方式

这种方式是将不同的 PKI 体系组织在同一个全球根 CA 之下，这个全球 CA 可由一个国际组织，如联合国等来建设。

考虑到各个 PKI 体系管理者一般都希望能保持本体系的独立性，全球统一根 CA 实现起来有一些具体的困难。所以，PKI 体系之间的互通性一般用交叉认证来实现。

3.2.12 CA 公钥管理原则

1) 规划

在规划阶段，需要对新的 CA 密钥对的引入需求进行调查研究。这些需求与密钥的数目和参数相关。

应用系统通过对当前 CA 密钥对的安全性进行评估，确定新引入的密钥对的模长和使用期限等其它属性。

2) 生成

评估后如果有必要引入新的密钥对，则要通过一个安全的方式生成预期的新的 CA 密钥对。

生成密钥对以后，要保证私钥的秘密性；同时要保证 CA 公钥和私钥的一致性。

3) 分发

CA 必须把生成的公钥分发给 CA 的所有成员发卡方及商家终端。如图 3-2 所示。

对于发卡方，为了在密钥使用阶段能够校验 CA 支持的发卡方公钥证书。对于商家，需要在其终端安全安装 CA 公钥。

注意：

为了使 CA 公钥能够正确引入，需要确保 CA、发卡方和商家之间 CA 公钥分发的一致性。

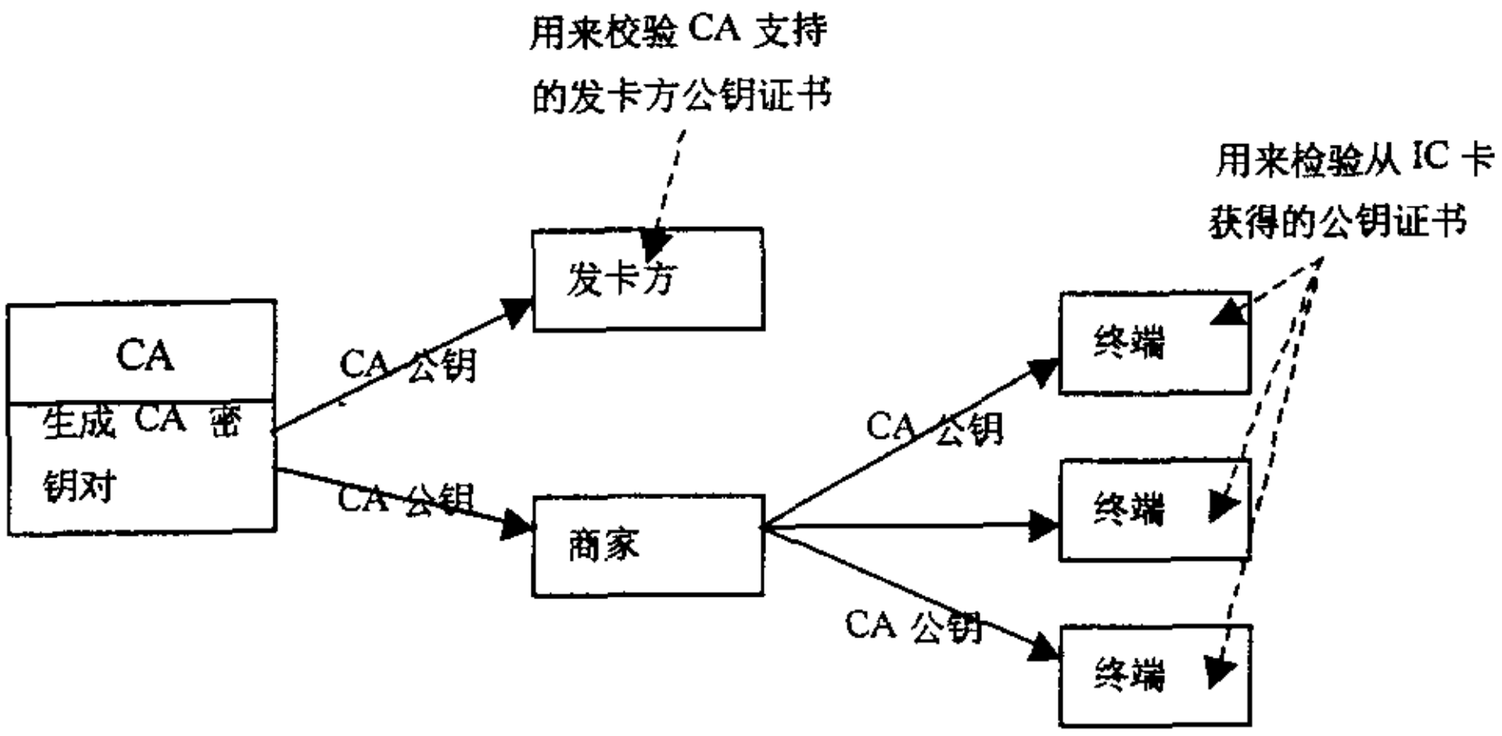

图 3-2 CA 公钥分发

4) 使用

CA 公钥通常用在终端进行静态或者动态数据鉴别。CA 私钥用于为发卡方公钥生成证书（如图 3-3）。

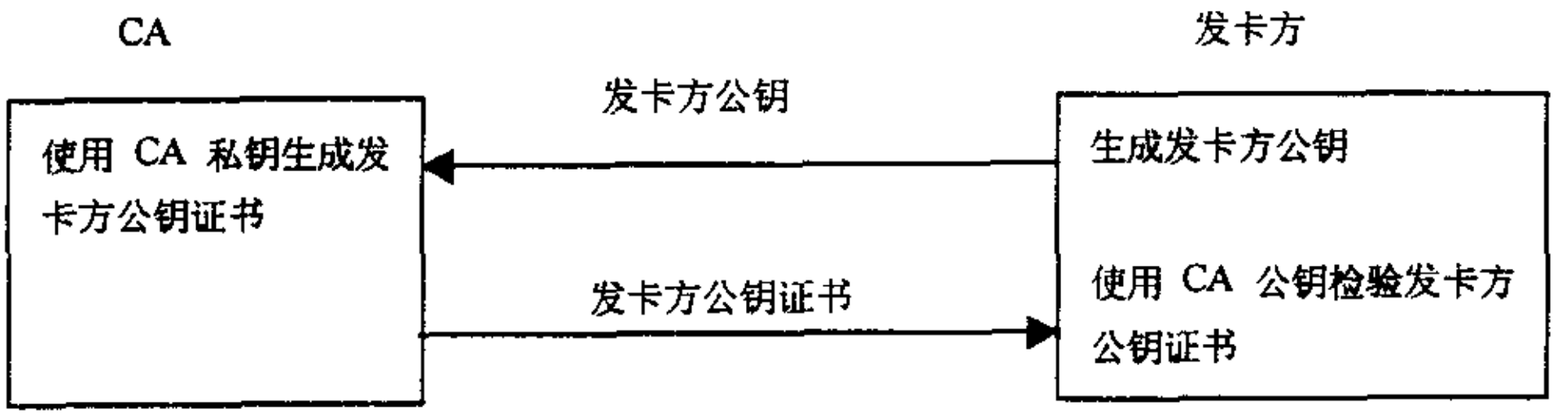

图 3-3 CA 签发发卡方公钥证书

说明:

发卡方生成自己的公钥，并把它送给 CA；

CA 使用自己的私钥生成发卡方公钥证书，并返回给发卡方；

发卡方使用 CA 公钥检验所收到的发卡方公钥证书，如果正确，发卡方就可以保存起来，并把它作为 IC 卡个人化数据的一部分。

注意:

为了确保发卡方公钥正确的引入，发卡方与 CA 需保证提交给 CA 的发卡方公钥的一致性。

5) 更新

对达到使用期限或其安全性受到破坏的 CA 密钥对，应用系统要进行密钥更新，以适应新的规划要求。

当 CA 密钥对到达使用期限，由 CA 私钥签发的发卡方公钥证书将不再有效。因而，发卡方应该确保使用该发卡方公钥证书个人化的 IC 卡的有限期不迟于 CA 密钥对的有限期。

在先于 CA 密钥对的有限期的适当时间内，CA 应该停止使用私钥签发发卡方公钥证书。

到有限期为止，商家必须撤换它们所有终端的 CA 公钥。

6) 监测

CA 密钥对的安全漏洞可能是 CA 本身可以得到证实的，也可能是密钥分析造成的，这些都是可以被证实的漏洞，还有一些是潜在的安全隐患：一方面来自系统管理员、成员和持卡人对系统信息安全产生的疑虑；一方面是密码分析技术的新发展带来的威胁。为此，需要监测信息系统，收集错误信息，进行安全防范。

7) 评估

CA 公钥对漏洞的评估内容将包含技术、风险、欺骗和商业冲突等。评估结果将对漏洞进行核实，以此支持作出判断和决定。

8) 决策

基于评估的结果，决定是否需要在 CA 密钥对有限期到来前对其进行更新。

9) 撤消

当作出撤消的决定后，通知整个系统，并在计划的时间内撤消存在安全漏洞的 CA 密钥对，同时对整个系统进行更新[28,30]。

3.3 小结

按照广义密钥管理体系结构和密钥管理方案思路，本章分析了智能卡的管理环境：

第一，智能卡具有自己的 CPU、ROM、RAM 和 EEPROM 等资源，但是计算

能力和存储容量都比较有限，这些限制了对算法和协议的选择，智能卡的 COS 系统实现了对这些算法和协议的支持。

第二，智能卡大都在分布式离线环境下使用，为此需要选用支持 PKI 协议的智能卡，并使用 PKI 协议建立通讯实体之间的信任关系。

在这些前提下，提出智能卡离线公钥管理模型，即：

离线模型 = 智能卡 + PKI 协议

并对模型进行了分析：智能卡离线公钥管理模型涉及到 CA、发卡方、用户卡、商家和终端五个角色，包括 CA 公钥的产生和分发，发卡方的注册，证书的获取、验证、保存、废止、注销和更新，密钥的生成、更新和恢复，CRL 的获取，和信任关系的建立等功能。

4 PKI 构架下的智能卡离线公钥管理实现

按照智能卡密钥管理模型，整个智能卡应用系统是由认证中心 CA、发卡方、终端和用户卡几个部分组成的。终端可以支持多个应用系统，它要保存所有被其支持的应用系统 CA 公钥，终端上使用的用户卡要保存相应的 CA 公钥索引。在用户卡与终端交互过程中，不需要 CA 参与，建立交互的信息均被保存在终端和卡中，所以用户卡需要和终端之间进行相应的信息鉴别，以决定卡片和终端的合法性。因此，PKI 框架下智能卡密钥管理具体实现涉及到这些相关细节：

1) 在离线状态下用户卡和终端应需要保存相应的安全信息，用来实现两者之间的校验；
2) 发卡方密钥对的产生、保存或更新，公钥证书的生成、保存、分发和更新，公钥证书由 CA 签发；
3) 用户卡密钥对的生成和保存，公钥证书的生成、保存；公钥证书由发卡方签发；
4) 在离线状态下合法性校验的内部机制；
5) 在离线状态下用户卡和终端之间的合法性校验；

智能卡离线公钥管理实现基本上是围绕以上内容展开的。首先，在第 4.1 节给出了智能卡在离线状态下完成合法检验的签名校验机制，它是公钥证书的验证，静态和动态数据鉴别的实现基础。接着，在第 4.2 节和第 4.3 节分别讨论了静态数据鉴别和动态数据鉴别有关细节，这是建立用户卡和终端信任关系的关键；同时，也完成了密钥和证书的生成、分发、保存和更新过程，给出了实现这种信任校验用户卡和终端分别需要保存的安全信息。在第 4.4 节实现了密钥管理中的用户 PIN 保护，这有关用户鉴别管理。后面两节涉及到密钥管理对终端的要求和 CA 公钥管理的实施策略，这些都关系到密钥管理的全局利益。

4.1 智能卡实现的签名与检验机制

签名/校验机制是 PKI 体系的基础，公钥证书的生成和校验、静态数据鉴别和动态数据鉴别等安全机制均是在它的基础上实现的。它是理解 PKI 体系的关键。

A. 签名机制原理

给定私钥 SK（模数长 N 字节）和报文 MSG（L 字节，L > N - 22），计算签名 S。

1. 计算 MSG 的 Hash 值 H（20 字节）。

   H = Hash[MSG]

2. 将 MSG 分成 MSG_1 和 MSG_2 两部分。

   $MSG = (MSG_1 \| MSG_2)$

   其中 —— MSG_1 由最左边 N-22 字节组成

   —— MSG_2 由其余 L - (N - 22)字节组成

3. 定义字节

   B = '6A'

4. 定义字节

   E = 'BC'

5. 定义数据块 X（N 字节）。

   $X = (B \| MSG_1 \| H \| E)$

6. 计算签名 S（N 字节）。

   $S = Sign(SK)[X] := X^d \bmod n$

签名 S（报文 MSG 的证书）不能单独使用，因为它包含的信息并不完备。如果同时获得签名 S 和它的配套数据（报文余留 MSG_2），只要知道与私钥 SK 对应的公钥 PK，就可以校验签名的正确性并复原报文 MSG。

B. 校验机制原理

给定公钥 PK（模数长 N 字节）、签名 S 和它的配套数据（报文余留 MSG_2），检验签名 S 的正确性。

1. 检查签名 S 的长度是否为 N 字节。

2. 计算数据块 X（N 字节）。

$X = Verify(PK)[S] := S^e \bmod n$

3. 将 X 分成 B、MSG_1、H 和 E 四部分。

$X = (B \| MSG_1 \| H \| E)$

其中 —— B 的长度为 1 字节

—— H 的长度为 20 字节

—— E 的长度为 1 字节

—— MSG_1 由其余 N-22 字节组成

4. 检查字节 B 是否为 '6A'。

5. 检查字节 E 是否为 'BC'。

6. 计算 $MSG = (MSG_1 \| MSG_2)$

检查是否满足 H == Hash[MSG] ，若是，报文合法。

4.2 离线静态数据鉴别的实现

终端使用基于公钥技术的数字签名/鉴别机制进行静态数据鉴别，以验证驻留在 IC 卡中的静态数据的合法性。目的在于判断 IC 卡个人化后数据是否在未授权的情况下发生改变。

静态数据鉴别机制[31,32]如图 4-1 所示。

符号说明：

RID	注册的应用提供者标识
ICC	IC 卡
PAN	主帐号
N_{CA}	CA 公钥模长
N_I	发卡方公钥模长
PK_{CA}	CA 公钥
SK_{CA}	CA 私钥
PK_I	发卡方公钥

SK_I　　发卡方私钥

$Cert_I$　　发卡方公钥证书

PK_{IC}　　IC 卡公钥

SK_{IC}　　IC 卡私钥

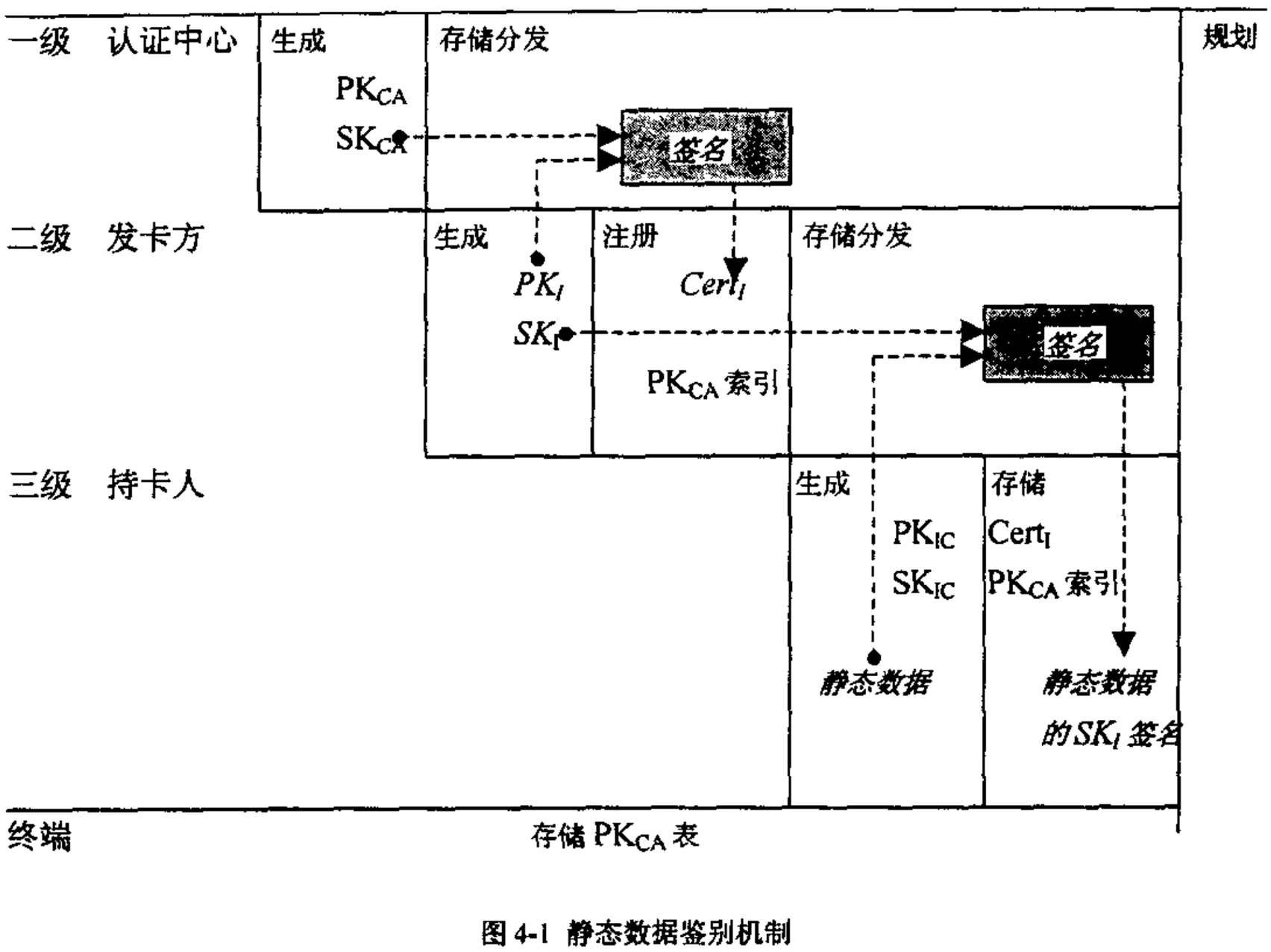

图 4-1 静态数据鉴别机制

IC 卡提供给终端：

——PK_{CA} 索引

——PK_I 证书 $Cert_I$

——静态数据 SK_I 签名

终端：

——从 PK_{CA} 表找到 PK_{CA}

——使用 PK_{CA} 校验 $Cert_I$

——使用 PK_I 校验静态数据签名

终端首先从 IC 卡取得 PK_{CA} 索引，从 PK_{CA} 表找到 PK_{CA} ；再从 IC 卡取得 SK_{CA} 签名的 PK_I 证书 $Cert_I$，用 IC 卡指定的 PK_{CA} 进行校验；然后从 IC 卡取得静态应用数据的 SK_I 签名，用 PK_I 进行校验。校验成功则认为 IC 卡是合法的。

4.2.1 ICC 中的数据

1) PK_{CA} 索引号。

2) $Cert_I$，亦即部分 PK_I 数据（长度为 N_{CA} - 22）的 SK_{CA} 签名，长度为 N_{CA}。
3) PK_I 余留（如果有的话）。若 $N_I > (N_{CA} - 36)$，则 PK_I 模数可分为两部分，左边 N_{CA} - 36 字节称为 PK_I 最左边数字，剩余的 $N_I - (N_{CA} - 36)$字节称为 PK_I 余留。
4) PK_I 指数。
5) 静态应用数据签名，亦即部分静态数据（长度为 N_I - 22）的 SK_I 签名，长度为 N_I。
6) 待鉴别的静态数据。

4.2.2 终端中的数据

各种 PK_{CA}（按 RID 和索引号查询，每个 RID 允许有 6 个）以及相关信息（如算法）。详见 4.5.2.2 节 CA 公钥的存放。

4.2.3 静态数据鉴别过程

4.2.3.1 获取 CA 公钥

终端从 ICC 取得 PK_{CA} 索引号。用此前取得的 RID 和 PK_{CA} 索引号查找 PK_{CA} 及其相关信息，若未查到，鉴别失败。

4.2.3.2 获取发卡方公钥

1) 终端从 ICC 取得 $Cert_I$、PK_I 余留和 PK_I 指数。如果 $Cert_I$ 的长度不等于 N_{CA}，鉴别失败。
2) 查询 CRL，检查 $Cert_I$ 是否已经废止，如果废止，则鉴别失败。
3) 用 PK_{CA} 按指定算法复原 $Cert_I$，得到复原数据，见表 4-1。若复原数据尾字节不是'BC'，鉴别失败。
4) 检查复原数据首字节，若不是'6A'，鉴别失败。
5) 检查证书格式，若不是'02'，鉴别失败。
6) 将数据从左到右按一定格式连接起来，对连接的结果执行 hash 运算，所用

算法由 Hash 算法标识(Hash Algorithm Indicator)指定。

表 4-1 发卡方公钥复原数据表[33-35]

域名	长度（字节）	说明
复原数据头	1	‘6A’
证书格式	1	‘02’
发卡方标识数	4	PAN 的最左边 3-8 字节
证书有效期	2	证书在此时间后无效
证书序列号	3	由 CA 分配的证书唯一二进制数据
哈希算法标识	1	标识产生哈希结果的算法
发卡方公钥算法标识	1	标识发卡方产生数字签名的算法
发卡方公钥长度	1	发卡方公钥模长（字节数）
发卡方公钥指数长度	1	发卡方公钥指数长度（字节数）
发卡方公钥或者发卡方公钥最左边的数字	$N_{CA}-36$	如果 $N_I<=N_{CA}-36$,本域由发卡方公钥和 $N_{CA}-36-N_I$个‘BB’组成 如果 $N_I>N_{CA}-36$,本域由发卡方公钥的最左边 $N_{CA}-36$ 个数字组成
哈希值	20	发卡方公钥信息的哈希结果
复原数据尾	1	‘BC’

7) 将上一步得到的 hash 值与复原的 hash 值进行比较，若不相同，鉴别失败。
8) 检查发卡方 ID 号是否与 PAN 的最左边 3-8 字节匹配，若否，鉴别失败。
9) 检查证书失效日期所载明的月份的最后一天是否为今天或更晚，若否，鉴别失败。
10) 检查 RID、PK_{CA} 索引号和证书系列号的连接是否合法，若否，鉴别失败。
11) 若发卡方公钥算法标识不可用，鉴别失败。
12) 如果上述所有检查都是正确的，将 PK_I 最左边数字与 PK_I 余留（如果有的

话）连接起来，获得 PK_I 模数，然后执行下一步，校验静态应用数据的签名。

4.2.3.3 静态数据鉴别

1) 终端从 ICC 取得静态应用数据签名和待鉴别的静态数据。如果静态应用数据签名的长度不等于 N_I，鉴别失败。
2) 用 PK_I 按指定算法复原静态应用数据签名，得到复原数据，见表 4-2。若复原数据尾字节(Recovered Data Trailer)不是'BC'，鉴别失败。

表 4-2 静态数据复原数据表

域名	长度（字节）	说明
复原数据头	1	‘6A’
签名数据格式	1	‘03’
哈希算法标识	1	用来标识计算哈希值的算法
数据认证码	2	发卡方分配的代号
填充值	N_I-26	由 N_I-26 个‘BB’组成
哈希值	20	静态数据的哈希值
复原数据尾	1	‘BC’

3) 检查复原数据首字节(Recovered Data Header)，若不是'6A'，鉴别失败。
4) 检查数据签名格式(Signed Data Format)，若不是'03'，鉴别失败。
5) 将数据按一定格式从左到右连接起来，对连接的结果执行 hash 运算，所用算法由 Hash 算法标识(Hash Algorithm Indicator)指定。将上一步得到的 hash 值与复原的 hash 值进行比较，若不相同，鉴别失败。

如果上述所有步骤都已成功执行，则静态数据鉴别成功。

4.3 离线动态数据鉴别的实现

终端使用基于公钥技术的数字签名/鉴别机制进行动态数据鉴别，以验证驻留在 IC 卡中的动态数据和从终端接收的动态数据的合法性。目的在于认证 IC 卡。

动态数据鉴别机制[31,32]，如图 4-2 所示。

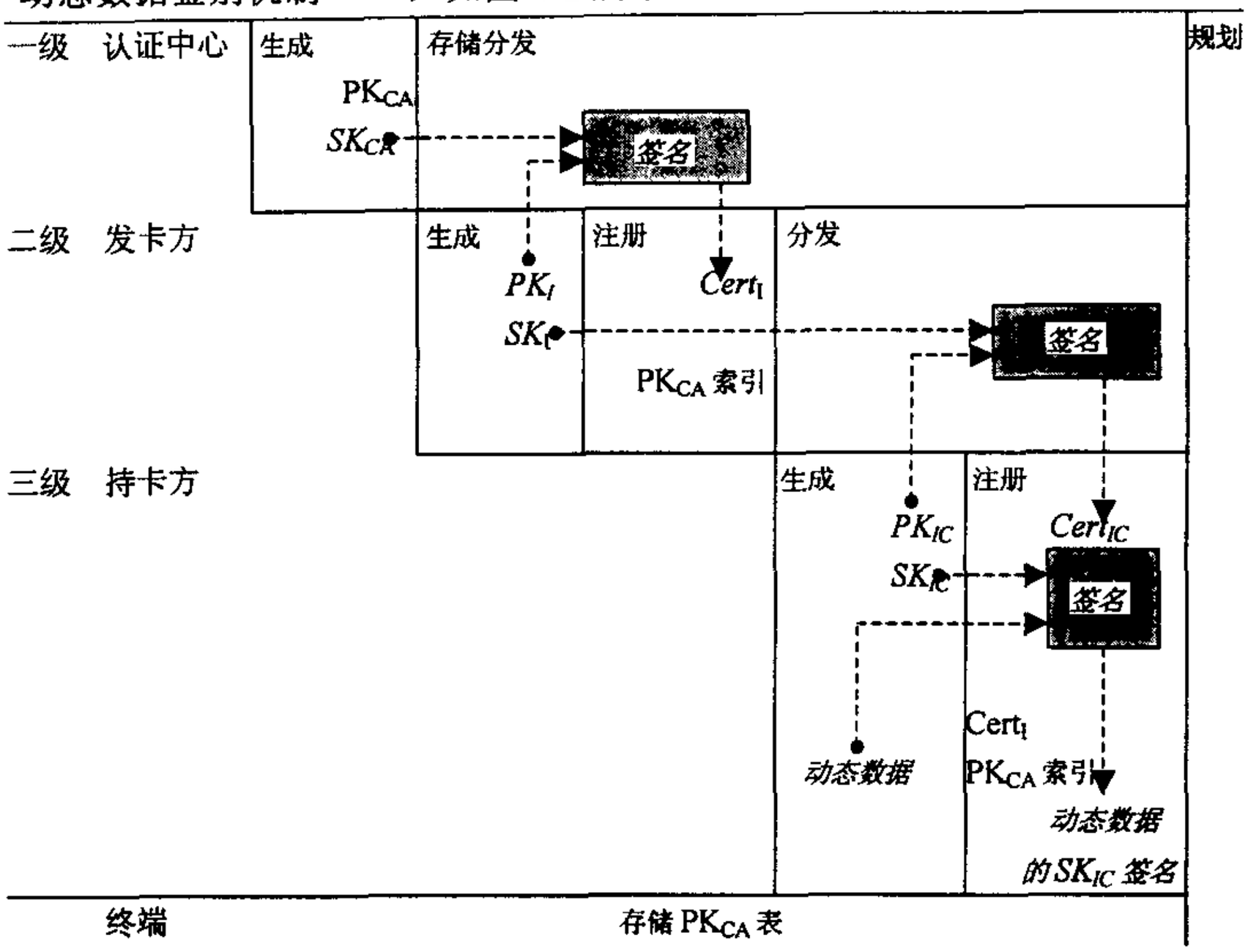

图 4-2 动态数据鉴别机制

IC 卡提供给终端：

——PK_{CA} 索引

——PK_I 证书 $Cert_I$

——PK_{IC} 证书 $Cert_{IC}$

——卡和终端动态数据

终端：

——从 PK_{CA} 表找到 PK_{CA}

——使用 PK_{CA} 校验 PK_I 证书 $Cert_I$

——使用 PK_I 校验 PK_{IC} 证书 $Cert_{IC}$

——使用 PK_{IC} 校验动态数据签名

符号说明：

RID	注册的应用提供者标识
ICC	IC 卡
PAN	主帐号
N_{CA}	CA 公钥模长
N_I	发卡方公钥模长

PK_{CA}	CA 公钥
SK_{CA}	CA 私钥
PK_I	发卡方公钥
SK_I	发卡方私钥
$Cert_I$	发卡方公钥证书
PK_{IC}	IC 卡公钥
SK_{IC}	IC 卡私钥
$Cert_{IC}$	IC 卡公钥证书
L_{DD}	动态数据长度

终端首先从 IC 卡取得 PK_{CA} 索引，从 PK_{CA} 表中找到 PK_{CA} ；再从 IC 卡取得 SK_{CA} 签名的 PK_I 证书 $Cert_I$，用 IC 卡指定的 PK_{CA} 进行校验；然后从 IC 卡取得 SK_I 签名的 PK_{IC} 证书 $Cert_{IC}$，用 PK_I 进行校验；最后 IC 卡生成动态应用数据的 SK_{IC} 签名，送给终端用 PK_{IC} 进行校验。如校验成功则认为终端是合法的。

4.3.1 IC 卡中的数据

1) PK_{CA} 索引号。
2) $Cert_I$，亦即部分 PK_I 数据（长度为 N_{CA} - 22）的 SK_{CA} 签名，长度为 N_{CA}。
3) PK_I 余留（如果有的话）。若 $N_I > (N_{CA} - 36)$，则 PK_I 模数可分为两部分，左边 N_{CA} - 36 字节称为 PK_I 最左边数字，剩余的 $N_I - (N_{CA} - 36)$字节称为 PK_I 余留。
4) PK_I 指数。
5) $Cert_{ICC}$，亦即部分 PK_{ICC} 数据（长度为 N_I - 22）的 SK_I 签名，长度为 N_I。
6) PK_{ICC} 余留（如果有的话）。若 $N_{ICC} > (N_I - 42)$，则 PK_{ICC} 模数可分为两部分，左边 N_I - 42 字节称为 PK_{ICC} 最左边数字，剩余的 $N_{ICC} - (N_I - 42)$字节称为 PK_{ICC} 余留。
7) PK_{ICC} 指数。
8) SK_{ICC}。
9) 动态应用数据签名，亦即部分动态数据（长度为 N_{ICC} - 22）的 SK_{ICC} 签名，长度为 N_{ICC}。

10) 待鉴别的动态数据。

4.3.2 终端中的数据

各种 PK_{CA}（按 RID 和索引号查询，每个 RID 允许有 6 个）以及相关信息（如算法）。详见 4.5.2.2 节 CA 公钥的存放。

4.3.3 动态数据鉴别过程

4.3.3.1 获取 CA 公钥

终端从 ICC 取得 PK_{CA} 索引号。用此前取得的 RID 和 PK_{CA} 索引号查找 PK_{CA} 及其相关信息，若未查到，鉴别失败。

4.3.3.2 获取发卡方公钥

1) 终端从 ICC 取得 $Cert_I$、PK_I 余留和 PK_I 指数。如果 $Cert_I$ 的长度不等于 N_{CA}，鉴别失败。
2) 查询 CRL，检查 $Cert_I$ 是否已经废止，如果废止，则鉴别失败。
3) 用 PK_{CA} 按指定算法恢复 $Cert_I$，得到复原数据，见表 4-3。若复原数据尾字节不是'BC'，鉴别失败。
4) 检查复原数据首字节，若不是'6A'，鉴别失败。
5) 检查证书格式，若不是'02'，鉴别失败。
6) 将数据按一定格式从左到右连接起来，对连接的结果执行 hash 运算，所用算法由 Hash 算法标识(Hash Algorithm Indicator)指定。
7) 将上一步得到的 hash 值与复原的 hash 值进行比较，若不相同，鉴别失败。
8) 检查发卡方 ID 号是否与 PAN 的最左边 3-8 字节匹配，若否，鉴别失败。
9) 检查证书失效日期所载明的月份的最后一天是否为今天或更晚，若否，鉴别失败。
10) 检查 RID、PK_{CA} 索引号和证书系列号的连接是否合法，若否，鉴别失败。
11) 若发卡方公钥算法标识不可用，鉴别失败。

12) 如果上述所有检查都是正确的，将 PK_I 最左边数字与 PK_I 余留（如果有的话）连接起来，获得 PK_I 模数，然后执行下一步，检索 PK_{ICC}[36]。

表 4-3 发卡方公钥复原数据表[33-35]

域名	长度（字节）	说明
复原数据头	1	'6A'
证书格式	1	'02'
发卡方标识数	4	PAN 的最左边 3-8 字节
证书有效期	2	证书在此时间后无效
证书序列号	3	由 CA 分配的证书唯一二进制数据
哈希算法标识	1	标识产生哈希结果的算法
发卡方公钥算法标识	1	标识发卡方产生数字签名的算法
发卡方公钥长度	1	发卡方公钥模长（字节数）
发卡方公钥指数长度	1	发卡方公钥指数长度（字节数）
发卡方公钥或者发卡方公钥最左边的数字	$N_{CA}-36$	如果 $N_I<=N_{CA}-36$,本域由发卡方公钥和 $N_{CA}-36-N_I$ 个'BB'组成 如果 $N_I>N_{CA}-36$,本域由发卡方公钥的最左边 $N_{CA}-36$ 个数字组成
哈希值	20	发卡方公钥信息的哈希结果
复原数据尾	1	'BC'

4.3.3.3 获取 IC 卡公钥

1) 终端从 ICC 取得 $Cert_{ICC}$、PK_{ICC} 余留和 PK_{ICC} 指数。如果 $Cert_{ICC}$ 的长度不等于 N_I，鉴别失败。
2) 查询 CRL，检查 $Cert_{ICC}$ 是否已经废止，如果废止，则鉴别失败。
3) 用 PK_I 按指定算法复原 $Cert_{ICC}$，得到复原数据，见表 4-4。若复原数据尾字

节不是'BC'，鉴别失败。

表 4-4　IC 卡公钥复原数据表[33-35]

域名	长度（字节）	说明
复原数据头	1	‘6A’
证书格式	1	‘04’
应用 PAN	10	PAN
证书有效期	2	证书在此时间后无效
证书序列号	3	由 CA 分配的证书唯一二进制数据
哈希算法标识	1	标识产生哈希结果的算法
ICC 公钥算法标识	1	标识 ICC 产生数字签名的算法
ICC 公钥长度	1	ICC 公钥模长（字节数）
ICC 公钥指数长度	1	ICC 公钥指数长度（字节数）
ICC 公钥或者 ICC 公钥最左边的数字	N_I-42	如果 $N_{IC}<=N_I-42$,本域由发卡方公钥和 N_I-42-N_{IC} 个‘BB’组成 如果 $N_{IC}>N_I-42$,本域由发卡方公钥的最左边 N_I-42 个数字组成
哈希值	20	发卡方公钥信息的哈希结果
复原数据尾	1	‘BC’

4) 检查复原数据首字节，若不是'6A'，鉴别失败。

5) 检查证书格式，若不是'04'，鉴别失败。

6) 将如下数据按一定格式从左到右连接起来，对连接的结果执行 hash 运算，所用算法由 Hash 算法标识指定。

7) 将上一步得到的 hash 值与复原的 hash 值(Hash Result)进行比较，若不相同，鉴别失败。

8) 检查复原的 PAN(Application PAN)是否与从 ICC 中读取的应用 PAN 相同，若否，鉴别失败。

9) 检查证书失效日期所载明的月份的最后一天是否为今天或更晚，若否，鉴别失败。

10) 若 ICC 公钥算法标识不可用，鉴别失败。

11) 如果上述所有检查都是正确的，将 PK_{ICC} 最左边数字与 PK_{ICC} 余留（如果有的话）连接起来，获得 PK_{ICC} 模数，然后执行最后两步，真正进行动态数据鉴别。

4.3.3.4 IC 卡生成动态签名

ICC 用 SK_{ICC} 对数据进行签名，得到动态应用数据签名。ICC 动态数据长度 L_{DD} 满足 $0 \leqslant L_{DD} \leqslant N_{IC} - 25$。ICC 动态数据最左边 3-9 字节是由长 1 字节的 ICC 动态数字长度和长 2-8 字节的 ICC 动态数字值（标记'9F4C'，2-8 字节二进制数）组成的。ICC 动态数字是由 ICC 生成的一个随时间变化的参数

4.3.3.5 终端校验动态签名

1) 终端从 ICC 取得动态应用数据签名和待鉴别的动态数据。如果动态应用数据签名的长度不等于 N_{ICC}，鉴别失败。

2) 用 PK_{ICC} 按指定算法复原动态应用数据签名，得到复原数据，见表 4-5。若复原数据尾字节不是'BC'，鉴别失败。

3) 检查复原数据首字节，若不是'6A'，鉴别失败。

4) 检查数据签名格式，若不是'05'，鉴别失败。

5) 将如下数据按一定格式从左到右连接起来，对连接的结果执行 hash 运算，所用算法由 Hash 算法标识(Hash Algorithm Indicator)指定。

6) 将上一步得到的 hash 值与复原的 hash 值进行比较，若不相同，鉴别失败。

如果上述所有步骤都已成功执行，则动态数据鉴别成功。

表 4-5 动态数据复原数据表

域名	长度（字节）	说明
复原数据头	1	‘6A’
签名数据格式	1	‘05’
哈希算法标识	1	用来标识计算哈希值的算法
ICC 动态数据长度	1	标识 ICC 动态数据的长度
ICC 动态数据	L_{DD}	由 ICC 产生或存放的动态数据
填充值	$N_{IC}-L_{DD}-25$	由 $N_{IC}-L_{DD}-25$ 个‘BB’组成
哈希值	20	动态数据的哈希值
复原数据尾	1	‘BC’

4.4 个人识别码（PIN）保护的实现

PIN 码加密用于离线 PIN 码校验。终端使用非对称加密机制完成对 PIN 码的加密，以便在 PIN 码输入板和 IC 卡之间安全传输 PIN 码[37]。

IC 卡应该具有一对与 PIN 码加密相关的密钥对。其中，公钥被 PIN 码输入板用来加密传输的 PIN 码，私钥被 IC 卡用来校验被加密的 PIN 码。

A. 密钥与证书

如果支持离线 PIN 码校验，IC 卡必须拥有一对密钥对；

密钥对可以是 IC 卡具有的特定的 PIN 码加密密钥对，这个公钥应该以一个公钥证书的格式存放在 IC 卡中。如果 IC 卡没有特定的 PIN 码加密密钥对，用于动态数据鉴别的密钥对可以用来进行 PIN 码的离线检验。

B. PIN 加密与校验

在终端和 IC 卡之间交换和检验加密 PIN 码要进行如下操作：

1) 在 PIN 码输入板上以明文格式输入 PIN 码，生成 PIN 码块（8 字节）；

2) 终端从卡中取出 8 字节的随机数；

3) 终端生成 N－17 字节长的随机填充数（N 为公钥模长）；

4) 终端使用获取的 PIN 加密公钥或者动态数据鉴别公钥加密 N 字节的 PIN 数据，得到 PIN 数据密文；

5) 终端向 IC 卡发出包含 PIN 数据密文的检验命令；

6) IC 卡使用相应的私钥解密 PIN 数据密文；

7) IC 卡检验解密得到的随机数和 IC 自己生成的随机数是否相等，如果不相等，则校验失败；

8) IC 卡检验数据头字节是否为‘7F’，如果不是，则校验失败；

9) IC 卡检验输入的 PIN 数据和 IC 卡中存放的 PIN 码数据是否相等，如果不等，则校验失败；

10) 如果所有步骤成功，则校验 PIN 码成功。

4.5 终端安全与密钥管理需求

在处理敏感数据，例如明文 PIN 码等需要考虑终端安全与密钥管理的要求。

4.5.1 终端安全需求

终端包括读写器和 PIN 码输入板等，对于终端要保证其在物理上和逻辑上均是安全的。

终端要严格限制物理访问终端内部敏感数据。

终端严禁被盗窃、拆卸、改装和仿造。

终端要阻止非法使用。

4.5.2 密钥管理需求

1) CA 公钥的引入

当应用系统决定引入新的 CA 公钥，要确保新公钥安全地分发到应用系统的每个商家，然后由商家将新的公钥和相关数据传递到所属的每个终端设备。

a) 终端必须能够检验收到的 CA 公钥和相关数据来自合法的商家。

b) 商家必须能够证实 CA 公钥正确的引入到终端设备。

2) CA 公钥的存放

每个 CA 公钥在终端存放的信息如下表 4-6：

表 4-6　存放在终端的 CA 公钥信息

名称	长度（字节）	说明
注册的应用提供者标识（RID）	5	标识与 CA 公钥相关的应用系统
CA 公钥索引	1	标识与 RID 对应的公钥
CA 哈希算法标识	1	标识产生哈希的算法
CA 公钥算法标识	1	标识公钥使用的数字签名算法
CA 公钥模数	<=248	CA 公钥模数值
CA 公钥指数	1 或 3	CA 公钥指数值
CA 公钥检验和	20	使用 SHA1 算法对上述公钥信息作哈希运算的结果

3)　CA 公钥的使用

CA 公钥在终端通常用来进行静态或动态数据鉴别，本质上是对发卡方公钥证书的校验，确保发卡方证书的合法性。

4)　CA 公钥的撤消

当应用系统决定撤消 CA 公钥，商家必须保证 CA 公钥不再被终端用来进行静态或动态数据鉴别。

a)　终端必须能够检验收到的撤消通知来自合法的商家。

b)　商家必须保证 CA 公钥在终端中被正确的撤消。

4.6 密钥管理实施原则及策略

公钥管理原则是实现 CA 公钥管理的依据。由这些原则可以推演出不同应用系统间共享的策略，或者推演出单独的应用系统策略。每个应用系统将开发自己的一套程序以实现这些策略。

1)　计划阶段

计划阶段涉及对当前使用的 CA 密钥对进行总结和对新 CA 密钥对进行规划。通过总结现存密钥对的使用情况和抗攻击能力，规划新密钥。通过对现存和新密钥对的长度和有限期等进行风险和密码分析，确保新密钥对达到预期的安全要求。

a) 原则

密钥的大小应该发挥与终端能力相容的最大可能性的安全。

应用系统应该与特定长度的密钥的有效期保持同步。

应用系统要对密钥泄漏、撤销和更新进行全面计划。

b) 策略

评估机构运用计算机科学、密码学和数据安全的当前最新成果，对 CA 密钥对进行评估。

评估机构收集对现存密钥对安全强度估计的准确信息，用来获得新密钥参数的推荐值。

应用系统考虑自身的特殊要求。

应用系统根据评估机构的推荐决定自己的密钥长度、有效期和撤消日程表安排，并通知到所有成员。

所有现存 CA 公钥将被设置相同计划过期时间。

商家应该有足够的更换时间。

所有新 CA 公钥将先于现存 CA 公钥计划过期时间完成分发。

商家在现存 CA 公钥计划过期时间内完成新密钥的安装。

新 CA 公钥将在现存 CA 公钥计划过期时间开始有效。

2) 生成阶段

CA 公钥应该在安全的环境中生成。

在任何注册的应用供应商标识中，CA 公钥索引代表一个特定的 CA 公钥对，而且是不能改变的。

3) 分发阶段

CA 公钥必须分发到各个终端，CA 私钥将用来产生发卡方公钥证书，并必须保存在安全的地方。

a) 原则

密钥分发必须确保密钥的一致性。

b) 策略

应用系统支持从 CA 到发卡方和商家以物理的或者电子的方式分发公钥。

所有新 CA 公钥必须在原来公钥的计划过期时间内分发完毕。

CA 公钥将被分发到发卡方，以便用来鉴别发卡方公钥的有效性。

4) 使用阶段

CA 公钥被终端用来进行静态和动态数据鉴别；私钥保存在 CA 中心，用来签发发卡方公钥证书。

a) 原则

终端应该具有支持安装和撤消 CA 公钥的能力。

终端应该具有判断 CA 公钥一致性的能力。

应用系统有责任保证 CA 公钥对的安全。

b) 策略

应用系统在发布证书之前，应该判断发卡方公钥的一致性和来源是否有效。

任何发行的 IC 卡的有效期不迟于保存在 IC 卡中的发卡方公钥证书的有效期，而且不迟于公布的用来产生发卡方公钥证书的 CA 密钥对注销日期。

发卡方公钥证书的有效期应该不迟于公布的用来产生发卡方公钥证书的CA密钥对注销日期。

IC 卡公钥证书的有效期不迟于用来生成 IC 卡公钥证书的发卡方密钥的有效期。

5) 监测阶段

监测是为了让一个实体有能力发现 CA 密钥对是否将会或者已经产生漏洞。他们可能是物理的、逻辑的、怀疑的、潜在的或已经证实了的漏洞。

a) 原则

密钥一致性的控制和潜在的 CA 密钥对漏洞的监测是应用系统的责任。

b) 策略

成员应该对可能的或有疑虑的漏洞发出通知信息。

6) 评估阶段

评估内容包括证实漏洞，决定可能的解决方案，估算该方案相对于漏洞带来的风险所花费的成本，和支持作出决策。

a) 原则

应用系统负责对被怀疑的或潜在的 CA 公钥漏洞进行评估。

应用系统应该制定评估政策和过程。

不同级别的漏洞要求可以采取不同种的解决方案。

b) 策略

对应用系统和成员来说，应用系统评估将包含实际的和预期的花费。

7) 决策阶段

作为评估的结果，应用系统决定对 CA 密钥漏洞采取相应的措施。

撤消 CA 密钥对是应用系统在大多数情况下作出的决定。为此，应用系统需要制定并向成员公布一套策略和步骤细节。

8) 撤销阶段

撤消意味着 CA 私钥不再用来签发发卡方公钥证书，并且从终端中取消 CA 公钥服务。同时，由该 CA 私钥签发的发卡方公钥证书在 IC 卡中不再有效。

4.7 小结

本章是对智能卡离线模型的具体化。

智能卡离线模型涉及五个实体，实际离线应用则主要是在智能卡和终端两个实体之间完成。其它实体主要是在密钥管理系统形成之初完成密钥及证书的产生、分发和保存工作。本章首先给出了智能卡在离线状态下完成合法检验的签名校验机制，它是公钥证书的验证，静态和动态数据鉴别的实现基础。接着分别讨论了静态数据鉴别和动态数据鉴别有关细节，这是建立用户卡和终端信任关系的关键，同时给出了实现这种信任校验用户卡和终端分别需要保存的安全信息。最后实现了密钥管理中的用户 PIN 保护，这有关用户鉴别管理，并讨论了密钥管理对终端的要求和 CA 公钥管理的实施策略，这些都与密钥管理的全局利益密切相关，所以不容忽视。

5 基于 PKI 的智能卡密钥管理系统设计

本章从实际应用的角度设计了一个智能卡密钥管理系统的演示系统。首先讨论了系统设计的原则、前提和要求，接着对公钥管理、安全信息传送和信任检测进行了设计和实现。

5.1 设计原则

为了从实际应用中检验“基于 PKI 的智能卡离线密钥管理”理论的合理性，设计了实现相关功能的公钥演示系统。从论文前面部分的讨论中知道基于 PKI 的智能卡密钥管理涉及到 CA、发卡方和用户方等各层实体，这些实体在密钥管理中处于不同的管理地位，完成不同的管理功能，而且在密钥管理中相互配合、相互依赖共同完成整个系统密钥管理功能[38]。整个密钥管理系统可以分为 CA 中心、发卡方管理部分和用户密钥管理部分。

在密钥管理系统中，CA 处于核心地位，但是在公钥演示系统中，并没有实现一个 CA 中心，而是假定有一个安全、公正和值得信任的 CA 存在。因为，第一，在论文的模型中，对 CA 的设计并没有提出任何高见；第二，论文的重点是在 PKI 模型框架的规范下在离线状态中智能卡的密钥管理；第三，CA 中心是一个信任中介，应该作为一个独立机构存在，而且系统庞大，不是此处可以完成的。

因此公钥演示系统只是一种演示，不具备一般的管理条件，所以此处设计的公钥管理系统没有普适性。

在这些前提下提出了以下几个原则要求：

5.1.1 功能要求

公钥演示系统要能够完成基本的发卡流程的各项功能，并对其给予控制管理。发卡流程包括格式化卡片、生成加密和签名密钥对和注册相应的公钥证书等。通过发卡流程实现系统密钥体系的建立。与此同时，提供一个有效的应用来检验发卡等

一系列操作的正确性和安全性。在此，演示系统提供了类似于安全电子邮件的应用演示。

5.1.2 安全要求

公钥演示系统最重要的是要显示它能够达到基本的安全要求，实现信息的保密性、完整性和不可否认性要求[33,39,40]。为此，提供了一个具有代表性的信任关系鉴别演示。

在前面密钥管理体系结构的讨论中知道：密钥管理应该充分发挥密码算法、密码协议和密码管理技术的联合优势，从系统的角度加以解决。所以，公钥演示系统选择公开的标准算法（DES、RSA、MD5 和 SHA1 等）和公开的标准协议（PKI），确保密码算法和密码协议的安全性能，密码管理技术便成为系统的弱点，所以把它作为安全目标更具有说服力和代表性。

5.1.3 界面要求

公钥演示系统的界面要能够提供相应的功能操作界面，并能够将每一步操作及其结果显示出来。对于发卡流程、安全电子邮件的发送和信任关系鉴别要反映出清晰的思路。同时，应该提供对卡片的浏览功能界面，以便清楚的看到发卡结果。当然还应该提供对读写器设备的管理操作功能及其界面。完整的公钥演示系统界面如图 5-1。

整个演示系统界面可以分为三个部分：读写器管理、公钥管理及安全测试和信息显示。

读写器管理是用来显示系统所有的读写器插槽、插槽内有无卡片以及插槽与卡片是否建立连接和会话。同时也用来显示当前操作的卡片对象或者会话对象。演示系统提供了一个辅助线程自动的完成对插槽内卡片的搜索与确认。卡片的插入和拔出都会自动的显示出来，可见这样还可以提供防拔插的功能。对于会话的建立和断开是通过“连接”和“断开”两个功能按钮完成的。

功能界面主要反映在公钥管理界面和安全测试界面。在公钥管理界面提供了格式化、登录和注册功能按钮，分别完成创建卡片应用文件系统、用户鉴别和生成加密/签名密钥对及双证书的功能。同时，提供了一个树控件和一个 Grid 控件分别用

来显示卡片内的密钥对和证书以及密钥对和证书的属性。安全测试界面提供了安全发送信息的功能和一个信任关系鉴别演示。

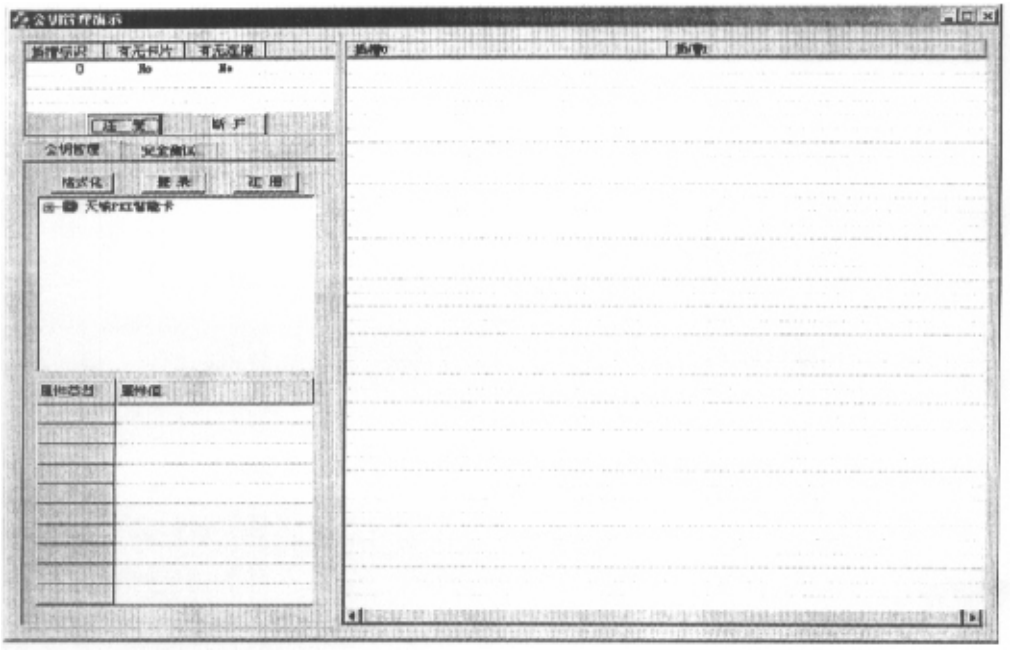

图 5-1 密钥管理演示系统

显示界面就是演示系统右边的列表视图。在此考虑到演示系统出现的最多插槽一般为两个，所以列表视图最多只给出两个插槽的结果显示信息。

5.2 公钥管理设计

公钥管理具有层次性，简单说来整个体系从上到下包括 CA 中心、发卡方和用户方，各个层次的管理任务不尽相同且各有偏重。在 5.1 节提过对于 CA 演示系统没有实现，在此只作理论上的分析，以保证系统的完整性。

5.2.1 CA 中心设计

数字证书作为网络用户的网上通行证，数字证书本身的可信度就相当重要。数字证书是由认证中心即 CA 中心发放和管理的，因此数字证书的可信任程度与发放

证书、提供与证书相对应的整套服务的认证中心的可信任程度有直接的关系。

目前，虽然有很多组织或公司都希望成为提供数字证书服务的认证中心(CA)，但它们很少真正认识和能够承担认证中心、以及一个可信任的、发布和管理数字证书的组织机构所应起的作用和承担的责任，同时也并不具备建立、运行和维护一个认证中心所必需的大量投资和专业队伍。设计认证中心时应该考虑 CA 应具备如下基本构架。

5.2.1.1 技术方案

认证中心所采用的技术方案是建立认证中心的基础，正确的技术方案可使数字证书可靠和容易使用，并容易被普遍接受。

加密技术是数字证书的核心。所采用的加密技术应考虑先进性、业界标准和普遍性。目前，较流行的是 RSA 数据安全加密技术，用 1024 位的加密算法。为保证加密体系和数字证书的互操作性，公钥加密系统和 X.509 也是广泛采用的标准，以实现认证中心的统一体系。

数字证书的有效周期管理对于数字证书是必须的，包括数字证书的发布、更新、作废的整个管理过程。数字证书的管理必须跟上证书持有的组织或个人情况的变化，及时更新或作废，对数字证书进行全过程的管理。此外，还有一些附加的对数字证书的管理，如证书目录查询、证书时间戳和证书管理情况的定期报告等。

为使数字证书在广泛应用领域内实现可互操作，数字证书需要与主要的 Internet 安全协议兼容，以支持应用环境，成为安全协议中所嵌入的数字证书。这些协议有安全电子交易协议 SET)、安全多用途 Internet 邮件扩展协议(S/MIME)和安全套层协议(SSL)等。

5.2.1.2 基础设施

这里的基础设施专指认证中心的安全设施、信息处理和网络的可靠性措施以及为客户服务的呼叫中心等。对于一个有长远规划的认证中心来说，无论是为公众还是为专用社团组织提供服务，都需要在基础设施方面进行周密的考虑和必要的投

资。

安全设施用来保护认证中心的计算机通信系统、证书签字单元、认证机构用于对每份证书进行数字签字的唯一私人密钥和客户的信息。为了使认证中心处于非常安全的环境，并使消费者相信他们的数字信息处于最高水平的保护之下，安全设施应设有多个关卡，进入认证中心的人员必须通过这些关卡。此外，还应有视频监视器、防护围栏和具有双向进入控制的安全系统来加强认证中心的安全程度。更为重要的是，只有可信任的、经过审查的认证中心人员才能接触和操作认证中心的设施。认证中心技术装备应是高可用性的，计算机系统、通信网络和呼叫中心必须是坚固的，使客户的需求随时得到满足。通信、数据处理和电源系统应通过多冗余备份系统来保证。网络安全应包括最新防火墙技术、通向最终用户的安全加密线路、IP 欺骗检测、可靠的安全协议和专家指挥系统。呼叫中心提供由专家支持的客户服务，并在任何时间都可以通过在线服务终端进行查询。

5.2.1.3 运作管理

运作管理是认证中心发挥认证功能的核心。运作管理包括数字认证的有关政策、认证过程的控制、责任的承担和对认证中心本身的定期检查等。

认证政策为数字认证过程建立行为准则，是认证中心的对外宣言，应包括在认证中心开始运作时对外公布的文件里。认证政策应在数字认证过程中，随着技术的进步和应用的发展适时地进行调整。

认证过程控制是认证政策的实施。认证机构必须有公正的、经过深思的运作控制，来管理数字认证过程。认证标准控制是重要的，它是认证一致性的保障。对于认证中心的工作人员必须提出要求，即证书是由经过训练的专业人员签发的，这些专业人员必须经过安全部门的检查，从而给客户以信心。

此外，认证机构应尽量使用数字证书的工业标准，比如由 WWW 协会、国家技术标准局和国际工程任务组等国际标准组织推荐的标准。

承担责任是一种需要，它帮助消费者确信他们的数字财富受到仔细的保护。如果因为某种原因，在认证机构保护下的消费者密钥丢失了，认证机构需要处理此事，并承担责任。责任保险是必须的，它能确保消费者的利益。

检查是一种重要手段，包括定期自我检查和接受第三方检查，它体现了认证中心对客户负责的精神。

5.2.2 发卡方与用户方设计

系统的演示部分是发卡方和用户方的管理功能。

发卡方应该提供密钥的产生、分发、存储、备份/恢复、归档和销毁功能，确定客户密钥生存周期、实施密钥吊销和更新管理功能，以及密码运算功能等。

用户方应该提供密钥的产生、存储、使用、申请吊销和密码运算功能等。

功能实现：

首先，建立连接。选中插槽，点击连接按钮，实现读写器管理，如图 5-2 所示。

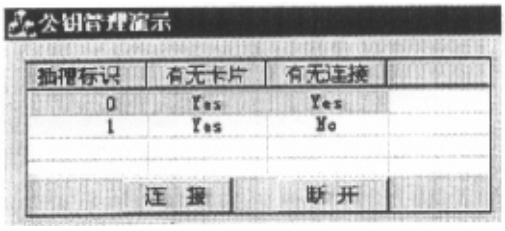

图 5-2 连接管理

第二，格式化卡片，建立卡片文件系统。

第三，登录，进行用户鉴别，实现访问控制，如图 5-3 所示。

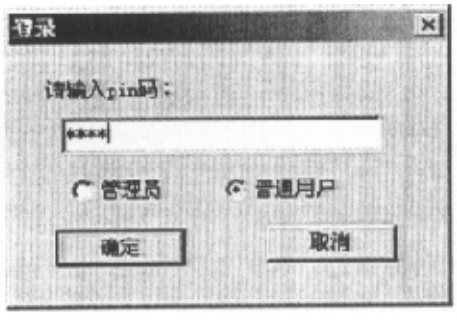

图 5-3 用户鉴别管理

此时卡片中仅仅建立了应用文件系统，建立密钥对和证书存储空间，如图 5-4 所示。

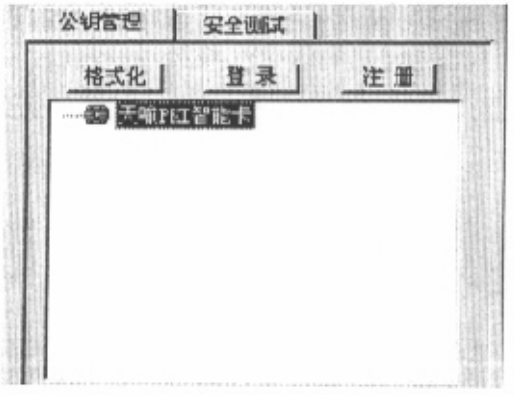

图 5-4 卡片内容显示

最后，生成密钥对和证书。密钥的生成、分发、存储、备份和归档在这部分完成；证书的生成、分发、存储、归档和密钥对生命期也是在这个时候确定的，如图 5-5 所示。

图 5-5 证书与密钥的管理

下面是注册函数实现的注册过程：

1) 在卡片内生成签名密钥对（使用属性模板）；
2) 获取签名公钥（使用属性模板）；
3) 封装成证书（公共的证书封装函数）；
4) 保存签名公钥证书（CTL:省略，卡中）；
5) 终端生成交换密钥对；
6) 在终端备份交换密钥对:用于恢复；
7) 向卡中写交换私钥（使用属性模板）；
8) 向卡中写交换公钥（使用属性模板）；
9) 封装成证书（公共的证书封装函数）；
10) 向卡中写交换公钥证书（使用属性模板）。

5.3 安全信息传送设计

安全信息传送用来检验在能够完成基本信息传送功能（如图 5-6 所示）的基础上，可以实现信息的秘密性、完整性和不可否认性要求。

图 5-6 安全信息传送

图 5-7 是信息发送函数显示发送过程：（插槽 0 向插槽 1 发送数据）

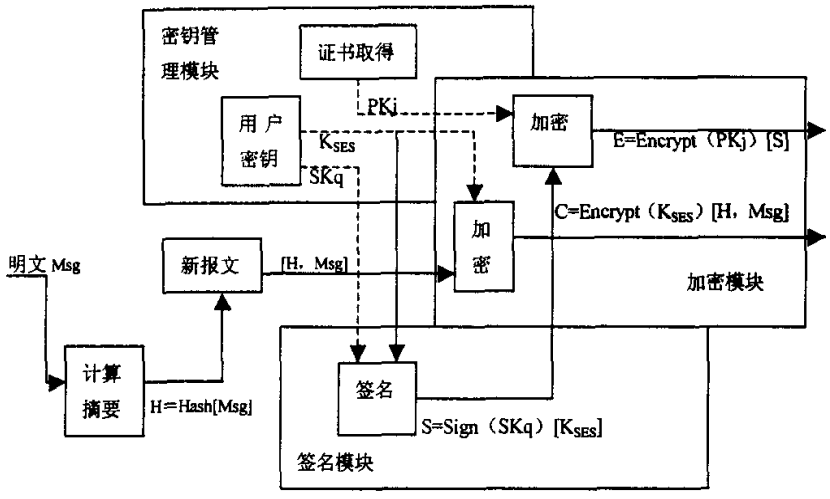

图 5-7 信息发送处理过程

其中，实线代表信息流，虚线代表作用因子。

1) 获取报文 Msg；
2) 插槽0计算实际报文摘要 H=Hash[Msg]；
3) 插槽0形成新报文 [H，Msg]；
4) 插槽0生成会话密钥 K_{SES}；
5) 插槽0用会话密钥 K_{SES}加密新报文 C=Encrypt（K_{SES}）[H，Msg]；
6) 用插槽0的签名私钥SKq对K_{SES}进行签名得到 S=Sign（SKq）[K_{SES}]；
7) 用插槽1的交换公钥PKj加密S得到 E=Encrypt（PKj）[S]；
8) 保存结果。

图 5-8 是信息接收函数显示信息接收过程：（插槽 1 接收插槽 0 数据，图中实线表示信息流，虚线表示作用因子。）

1) 用插槽1的交换私钥SKj解密E得到 S=Decrypt（SKj）[E]；
2) 用插槽0的签名公钥PKq对S进行校验得到 K_{SES} = Verify(PKq)[S]；
3) 用插槽1会话密钥K_{SES}解密C得到报文[H，Msg]=Decrypt（K_{SES}）[C]；

4) 解析报文[H，Msg]得到H和Msg；

5) 计算H'=Hash[Msg]；

6) 比较H 和 H'。

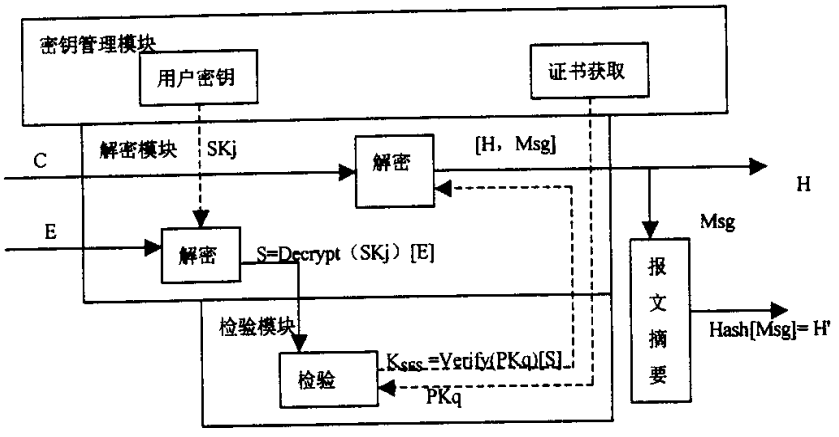

图 5-8 信息接收处理过程

在此，可以将信息传递过程按照功能分为加、解密模块，数字签名与验证模块和密钥管理模块。其中，加、解密模块完成信息的保密性；数字签名与验证模块保证信息的完整性、源发性和不可否认性；而密码管理模块则完成数据加密密钥的生成，用户所用公、私密钥对的存储，以及其他用户证书的取得和验证。无论加、解密模块还是数字签名与验证模块，其安全功能的实现在很大程度上依赖于密钥管理模块对密钥的保护与鉴别，因此，密钥管理模块是信息传递的安全核心。

5.4 信任检测设计

5.4.1 选择监测目标

虽然监测目标有多处，但是对于监测设计不可能面面俱到，只能选择有代表性的和论文密钥管理密切相关的信任监测。

鉴于论文阐述的信息安全层次结构和密钥管理模型，对于基本安全算法和高级安全协议，因为选择的都是标准算法和协议，所以它们并不是密钥管理模型的薄弱

环节。因此，最有说服力的安全目标应该在密钥管理技术层次，而且密钥管理技术也是论文所关心的部分。在密钥管理层次部分，涉及到与密钥分发传递有关的中间人攻击，这种攻击比较适合具有未知信任缺陷的系统。在基于 PKI 的密钥管理系统中，可以通过信任鉴别来对实体身份进行确认以使这种攻击无效，从而保护信息不被泄漏。

5.4.2 实现信任检测

在 5.3 节的安全信息传送中，信息的传送和接收都和密钥管理模块有密切的关系，系统的安全功能依赖于密钥管理模块对密钥的保护和鉴别。在未知信任缺陷的系统，对公钥证书缺乏鉴别机制，以致造成信任安全漏洞，受到中间人介入的攻击。中间人攻击[41]的原理如图 5-9：

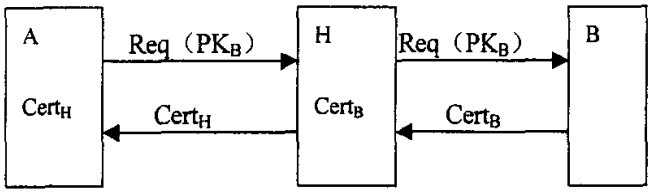

图 5-9 中间人攻击示意图

A. 中间人攻击：

在 A 发出获取 B 的交换公钥证书请求 Req（PK_B）时，H（Hacker）接收到请求信息，并将请求信息传递给 B；

B 发出自己的交换公钥证书 $Cert_B$ 作为应答；

H 处于 A 和 B 交互之间，先于 A 收到交换证书应答 $Cert_B$，保存 B 的交换公钥证书 $Cert_B$；

H 把自己的交换公钥证书 $Cert_H$ 发给 A 作为应答；

A 把 H 的交换公钥证书 $Cert_H$ 当作 B 的交换公钥证书保存一份，以备后来传递信息时加密会话密钥签名使用。

B. 在具有未知信任缺陷的系统：

当系统不能进行信任鉴别[42,43]时，A 无法知道自己获取的 B 的证书是否确定就是 B 的交换公钥证书。因此，就不能检测出中间人攻击，于是 H 就可以阅读 A

和 B 之间传递的信息，而且 A 和 B 对此毫无知觉。

A 向 B 发送 Msg：

1) 获取报文 Msg；
2) A计算实际报文摘要 H_M=Hash[Msg]；
3) A形成新报文 [H_M，Msg]；
4) A生成会话密钥 K_{SES}；
5) A用会话密钥 K_{SES}加密新报文 C=Encrypt（K_{SES}）[H_M，Msg]；
6) A用自己签名私钥Sk_{Aq}对K_{SES}进行签名得到 S=Sign（Sk_{Aq}）[K_{SES}]；
7) A用H的交换公钥PK_{Hj}加密S得到 E=Encrypt（PK_{Hj}）[S]；
8) A向B传递秘密信息C和E。

H截获秘密信息C和E：

1) H用自己的交换私钥SK_{Hj}解密E得到 S=Decrypt（SK_{Hj}）[E]；
2) H用A的签名公钥Pk_{Aq}对S进行校验得到 K_{SES} = Verify(Pk_{Aq})[S]；
3) H用会话密钥K_{SES}解密C得到报文[H_M，Msg]=Decrypt（K_{SES}）[C]；
4) H解析报文[H_M，Msg]得到H_M和Msg；
5) H计算H'=Hash[Msg]；
6) 比较H_M 和 H'，获取信息。

H向B发送Msg：

1) H用B的交换公钥PK_{Bj}加密S得到 E=Encrypt（PK_{Bj}）[S]；
2) H向B传递秘密信息C和E。

C. 在基于 PKI 信任的系统：

在基于 PKI 信任的系统中，A 可以通过 CA 的认证鉴别证书身份，可以知道接收的证书是否是 B 的公钥证书，从而避免了中间人攻击。在智能卡应用环境，通常使用离线的静态数据鉴别和动态数据鉴别来确保通讯双方身份的真实性和合法性。

5.4.3 监测数据分析

演示主要过程包括建立密钥系统、安全信息发送和接收以及中间人攻击的信任检测，在此只对信任关系检测数据进行分析。图 5-10 是整个过程的结果显示：

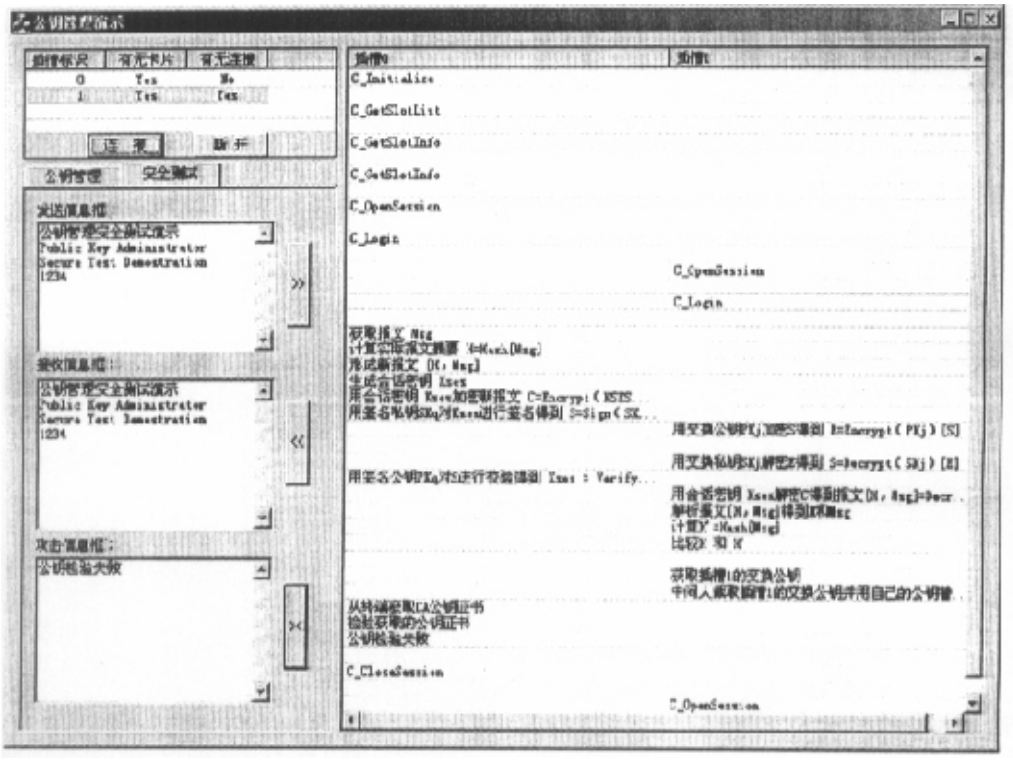

图 5-10 演示过程数据

信任关系检测过程分析：

Enter Function: C_FindObjectsInit

参数信息

CK_SESSION_HANDLE hSession = 13047296

CK_ATTRIBUTE_PTR pTemplate = 0012F488

CK_ULONG ulCount = 5

模板为:

No.0 类型 = 00000000 长度 = 4 值(指针) = 0012F4C4 数据= 02 00 00 00

No.1 类型 = 00000003 长度 = 14 值(指针) = 0012F4CC 数据= 70 6B 6D 73 5F 65 78 63 68 61 6E 67 65 00

No.2 类型 = 00000001 长度 = 1 值(指针) = 0012F4E0 数据= 01

No.3 类型 = 00000100 长度 = 4 值(指针) = 0012F4DC 数据= 00 00 00 00

No.4 类型 = 00000104 长度 = 1 值(指针) = 0012F4E0 数据= 01

CK_ULONG ulCount = 5

Exit Function: C_FindObjectsInit

==========

Enter Function: C_FindObjects

参数信息

CK_SESSION_HANDLE hSession = 13047296

CK_OBJECT_HANDLE_PTR phObject = 0012F5B8

CK_ULONG ulMaxObjectCount = 1

CK_ULONG_PTR pulObjectCount = 0012F4E4

找到的数目为: 1

句柄值为:

No.0 Handle = 2097154

执行成功

Exit Function: C_FindObjects

==========

Enter Function: C_FindObjectsFinal

参数信息

CK_SESSION_HANDLE hSession = 13047296

执行成功

Exit Function: C_FindObjectsFinal

==========

Enter Function: C_VerifyRecoverInit

参数信息

CK_SESSION_HANDLE hSession = 13047296

CK_MECHANISM_PTR pMechanism = 0012F470

机制为:

类型 = 00000001 长度 = 0　　值(指针) = 00000000

CK_OBJECT_HANDLE hKey = 0012F5B8

执行成功

Exit Function: C_VerifyRecoverInit

==========

Enter Function: C_VerifyRecover

参数信息

CK_SESSION_HANDLE hSession = 13047296

CK_BYTE_PTR pSignature = 015A1160

CK_ULONG ulSignatureLen = 64

0F 8A 36 39 D5 EA C7 B7 39 73 0E 37 01 35 36 18 98 26 56 6A 66 94 A5 36 97 98 B7 B8

52 8F 4E F4 DF 47 08 0A B3 5E 03 9E 98 F9 3D 09 DD 61 8C EE DE 87 89 98 75 CB 08 09

5B E0 BB A4 BD 6F 3E 84

CK_BYTE_PTR pData = 00000000

CK_ULONG_PTR pulDataLen = 0012F640

执行失败

Exit Function: C_VerifyRecover

结束语

面对当前国内外密钥管理方面的不足，论文作了三个方面的努力：

第一，参照系统论和木桶效应原理，融合广义、整合、全局、动态和开放的观念，提出了广义密钥管理体系结构思想，确定了智能卡密钥管理简要方案和思路。

第二，构造了基于 PKI 协议的智能卡离线密钥管理模型。讨论了密钥管理模型下 CA 公钥的产生和分发，发卡方的注册，证书的获取、保存、验证、废止、注销和更新，CRL 的获取，密钥的生成、更新和恢复，和角色间信任关系的确立等内容，为具体的密钥管理实现作准备。

第三，实现了在离线状态下的密钥管理。在给出签名校验机制的基础上，详细讨论了静态数据鉴别和动态数据鉴别的实现，同时也给出了密钥和证书相应操作过程。而且讨论了与密钥管理全局利益密切相关的用户 PIN 保护、密钥管理对终端的安全要求及公钥管理实施策略等内容。

论文基本上完成了预期的计划，由于水平和时间有限，还有一些问题有待改进：

首先，论文提出的密钥管理的体系结构与方案思路比较简单粗糙，没有提供深入的理论论证，作为一个思路尚可，还需要在今后的工作中进行精细的定量分析和全面的讨论，以便提出丰富完善的体系。

第二，在智能卡密钥管理实现中，基本上是在考虑密钥管理技术稳定因子的基础上实现密钥管理系统的，没有对活动因子进行收集和分析，还需要对所有参考因子实现库管理。

第三，在智能卡公钥管理模型中虽然提出了通过 PKI 解决应用体系的互通性问题，不过在实际中很少遇到，所以该模型下的离线公钥管理并没有实现互通性。

由于能力有限，错误和不当之处，恳请各位老师和专家批评指正，笔者在此深表谢意！

致谢

在此硕士学位论文完成之际，我要衷心地感谢我的导师：曹化工教授和童敏教授。两位教授在我的硕士学位攻读期间，给予我极大的教诲和鼓舞。他们严谨的治学作风、渊博的学识令我敬佩，他们不厌其烦的悉心指导、和蔼可亲的育人风范使我感动。他们对我的教导和关怀，将使我终生难忘。

感谢天喻信息技术中心欧阳由先生。在与他共事的日子里，他认真踏实的科研作风和朴实的生活态度也促使我努力提高自己的能力和水平，是我学习的榜样。

感谢本教研室的卢正鼎老师、秦友淑老师、胡和平老师、马光志老师、胡久乡老师和刘清老师等，他们以不同的方式给予了我支持和帮助。

感谢我的室友狄晓涛和邹畅，感谢天喻信息的同事付秦华、朱文显、吴俊军、程诗猛和刘辉，和他们的每次讨论都使我获益匪浅，也感谢他们的关心和鼓励。

感谢我的父母，是他们赋予了我生命和智慧，给予了我最无私的爱和关怀，是他们牺牲自我的精神才使我取得今天的成绩。感谢我的姐姐和哥哥，他们也给了我无尽的关爱。

最后，衷心感谢在百忙之中抽出时间审阅本论文的专家教授。

笔者：梁宗炼

二 OO 二年四月

参考文献

[1] 蒙杨,卿斯汉,刘克龙.等级加密体制中的密钥管理研究.软件学报,2001,12(08):1147-1153.

[2] 余祥宣,倪晓俊.加密数据库系统中的密钥管理.华中理工大学学报,1995,7,23(7):52-55.

[3] M Bellovin and M Merritt. Limitations of the kerberos authenication system. Computer Communication Review, October 1990:20(5):119-132.

[4] Steiner J,Neuman C,Schiller J I.Kerberos:An Authentication Service for Open Network Systems in Proc.Winter USENIX Conference, Dallas,1998:80-84.

[5]. T Kohl and B C Neuman. The Kerberos network authentication service. Internet RFC 1510, September 1993:150-156.

[6] 曹化工,梁宗炼,高小新等.基于智能卡的 PKI 体系实现框架.小型微型计算机系统,2002.

[7] Girault, M. Self_Certified public keys. In:Christoodolakis,ed. Proceedings of EuroCrypt'91. Lecture Notes in Computer Science 547, Berlin: Springer-Verlag, 1991, 490-497.

[8] 孙晓蓉,王育民.计算机分布式环境中的认证与密钥分配研究. 计算机学报,1999,6,22(6):577-581.

[9] Dolev D, Yao C A. On the Security of Public Key Protocol.IEEE Tansactions on Information Theory,1983,30(2):18-36.

[10] Wei-Bin Lee.Authenticity of public keys in asymmetric cryptosystems. Computer Communications ,1998,21:60-62.

[11] S. Chokhani.Toward a national public key infrastructure. IEEE Communications Magazine 4(1994):70-74.

[12] J.J. Tardo, K. Alagappan.SPX:global authentication using public key certificates.

In:Proceedings of the IEEE Symposium on Security and Privacy,Oakland, 1991,pp.232-244.

[13] National Bureau of Standards, NBS FIPS PUB 74. Guidelines for Implementing and Using the NBS Data Encryption Standard.U.S. Department of commerce, Apr 1981:74-77.

[14] R.C. Merkle. Secrecy, Authentication, and Public Key Systems. Ph.D. dissertation, Stanford University,1979:56-60.

[15] 许剑卓,戴英侠,左英男.一种混合认证体制.计算机工程,2000,3:63-87.

[16] （美）施奈尔（Schneier.B.）著. 应用密码学：协议、算法与 C 源程序.第二版. 吴世忠等译,何德全校.北京:机械工业出版社,2000,1:200-204.

[17] W. Mao and C. Boyd. Towards Formal Analysis of Security Protocols. Proceedings of the Computer Security Foundations workshop VI, IEEE Computer Society Press, 1993, pp. 147-158.

[18] 田梦瑾,李建华,杨宇航.基于 PKI 的电子商务安全密钥托管方案研究.计算机工程,1999,1:35-37.

[19] Visa and MasterCard. Secure Electronic Transaction, Book1: Business Description, 1997,5:40-50.

[20] National Bureau of Standards(U.S.).Data Encryption Standard(DES).Federal Information Processing Standards Publication 46,Springfield, VA,1977,4:58-64.

[21] National Institute of Standards and Technology. Secure Hash Algorithm. Federal Information Processing Standards Publication 180:115-120.

[22] 刘玉莎,张晔,张志浩等.公钥基础设施在网络安全中的研究与应用.计算机工程与应用,2000,3:133-137.

[23] Housley R, Ford W, Polk W. Internet X.509 Public Key Infrastruture Certificate and CRL profile.(1999-01) RFC 249-254.

[24] Adams C,Farrel S(SSE). Internet X.509 Public Key Infrastructure Certificate Management Protocols.RFC2510,1999,03:250-254.

[25] 韦为,杜炜,王行刚.构造基于 X.509 公钥证书的密钥管理系统.计算机工

程,1999,10,V25 Special Issue:133-135.

[26] C Adams,S Farrell. Internet X.509 Public Key Infrastructure Certificate Management Protocols.RFC 2587,1999,6:247-252.

[27] ISO/IEC 9594-8/ITU-T Recommendation X.509. Information Technology – Open Systems Interconnection.ITU ,1997:142-148.

[28] 卢昱,李勇奇.基于数字签名及智能卡的 Intranet 认证模型.计算机工程与应用,1999,3:107-109.

[29] 徐志大,南相浩.认证中心 CA 理论与开发技术.计算机工程与应用,2000,9:87-90.

[30] EMV2000 Integrated Circuit Card Specification for Payment Systems Book2 – Security & Key Management 2000,12 :43-50.

[31] Frankel M, Yung M. Secure and Eficient Off-Line Digital Money. Proceeding of the Twentith International Colloquium on Automata, Languages and Programming (ICALP 1993).Lund, Sweden: Springer-Verlag, 1993.265-276.

[32] Yacobi Y. On the Continuum between On-Line and Off-Line E-Cash Systems. Financial Cryptography'97. Anguilla,British West Indies: Springer-Verlag, 1997:193-202.

[33] 汪立东,余祥湛,方滨兴. PKI 中几个安全问题的研究.计算机工程,2000,1:14-16.

[34] A.Shamir. Identity-based cryptosystem and signature schemes. In:Proceedings of Crypto'84, Santa Barbara, CA, 1984,pp.47-53.

[35] Schnopr C P. Efficient identification and signature for smart cards. In: Advances in Cryptology –CRYPTO'98, Santa Barbara, 1985.239-252.

[36] R.L. Rivest, A.Shamir and L.Adelman. A method for obtaining digital signatures and public-key cryptosystem. Commun.of ACM 21(2)(1987):120-126.

[37] Lamport, L. Password authentication with insecure communication, Comm. ACM.1981;24(11):770-772.

[38] 董雨果,郑连清,赵雪岩.一种新型 Internet 网络密钥管理体制.纺织高校基础科学学报,1999.12 12(4):305-307.

[39] Foss JA.Multi-protacol Attacks and the Public-Key Infrastructure. In Proc.21st

National Information Systems Security Conference, Arlington, Va., 1998-10: 566-576.

[40] 张吉文,郭圣文.IC 卡系统设计中的安全性考虑.微型机与应用,2000,2:17-18.

[41] Kent S.Privacy Enhancement for Internet Electronic Mail: Part II:Certificate-based Key Management. RFC1422, 1993-02.

[42] Bruce Schneier.E-mail Security. New York:John Wily,1995:84-92.

[43] 许晓东,荆继武,杨炜.安全电子邮件的系统分析与密钥管理.计算机应用研究,1999,11:59-61.

附录 攻读硕士学位期间发表的论文

[1] 曹化工,梁宗炼,高小新等.基于智能卡的 PKI 体系实现框架.小型微型计算机系统,2002,已录用.

[2] 梁宗炼,欧阳由.基于 PKI 的卡基密钥管理.信息与开发,2001,4.