“一卡多用”智能卡安全性的实现①

Research on the Security of Multi - application Smart Card

李成华　张新访　吴俊军　（华中科技大学信息与系统技术研究所 430074）

摘要：本文从应用的传输、应用的卡内操作、应用的存储管理、应用安全运行、用户数据和业务安全等核心环节对“一卡多用”的安全需求进行了分析，并给出了实现方案。

关键字：智能卡　一卡多用　安全性

1　引言

从智能卡的生命周期来看，“一卡多用”可分为静态和动态两种。静态“一卡多用”是指卡片制造商将几个应用放在一张卡上发行，用户在使用过程中不能更新应用。由于它是单个应用的功能组合，国内已有不少 COS（Chip Operating System）供应商能实现。后文将这种静态的单应用和多应用智能卡称为传统的智能卡。动态“一卡多用”是指用户在使用过程中可根据自己的服务需求下载、更新或删除特定的应用。这是智能卡技术领域发展的必然趋势。

传统的 COS 直接将一个或多个应用包含在其中，它是一个专用系统而不是通用系统。和 PC 上的操作系统相比较，COS 在本质上更加接近于监控程序，而不是一个真正意义上的操作系统，它不涉及到共享、并发的管理及处理。“一卡多用”智能卡卡上系统如图 1 所示。

2　“一卡多用”智能卡的安全需求分析

由于“一卡多用”智能卡应用是在卡片发行后加载（或更新或删除）的，所以产生了许多由应用带来的直接的和潜在的安全需求。在“一卡多用”项目中一个具体的应用是由应用提供商提供的应用代码和由卡片发行商提供的应用配置数据组成的。

2.1　应用代码及其配置数据的安全

（1）应用代码及其配置数据在卡外非安全的网络上传输时的机密性、完整性和不可抵赖性。在“一卡多用”智能卡应用模式中允许用户从任意地方下载应用，包括使用不安全的网络。所以应用在传输的过程中，必须保证其内容的完整性和机密性。

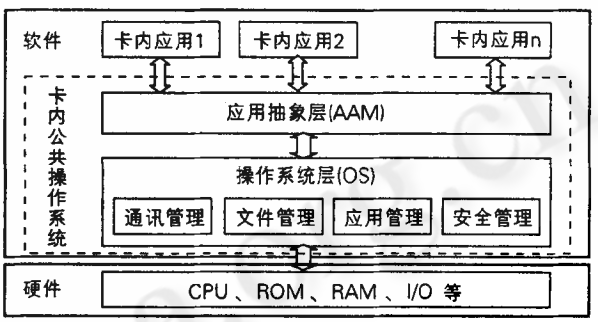

图 1　“一卡多用”的卡上系统结构

（2）在卡内对操作应用代码及其配置数据时的安全。从本质上讲，存储器管理是卡上数据的安全的关键。在“一卡多用”中，存储器在操作系统的控制下被分配给已经授权的应用。“一卡多用”系统必须采取某种措施对应用进行授权，在应用下载时对其进行合法性验证，认可这些应用能否被下载。

（3）应用在卡内存储时的安全。由于多应用的存在，系统必须设计应用合理的安全的存储管理方式。应用在卡内的存储结构决定了对应用的选择、安装和删除方法。当卡片下载、更新或删除一种应用软件时，

① 信息产业部发展基金项目：基于国产芯片和一卡多用的卡操作系统及其开发工具（编号：XDJZ－0317－07）

不得影响其它已经存在的应用的操作环境，即要保证每一个应用的物理“独立性”。

（4）应用在卡内运行时的安全。当一种应用软件运行时，不得以如何方式影响其它应用的操作环境（公共数据区的数据除外），即要保证每一个应用的逻辑“独立性”。

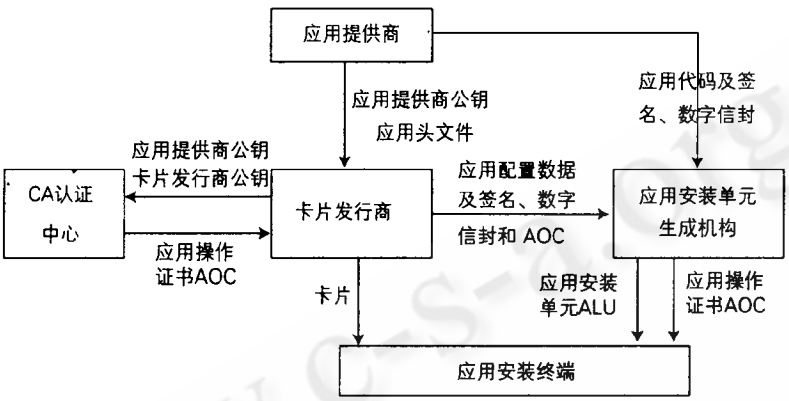

图 2 “一卡多用”应用下载时各实体的关系

2.2 用户数据的安全和业务（交易）的安全

对每一个具体应用而言必须和传统智能卡的 COS 一样要保证用户数据和业务的安全，包括用户与智能卡的鉴别、核实功能以及对传输数据的加密与解密操作等。“鉴别”是指对智能卡本身的合法性进行验证，“核实”是指对智能卡持有人的合法性验证，机密数据应该以密文形式通讯线上传输，对内部安全文件和存放用户机密数据文件要实施访问控制。

3 主要实现方案

3.1 基于 PKI 技术卡片发行商控制机制

公钥基础设施 PKI 是以公开密钥技术为基础，以数据的机密性、完整性和不可抵赖性为安全目的而构建的认证、授权、加密等硬件、软件的综合设施。在“一卡多用”的项目中，我们采用 PKI 技术设计了一个多应用管理系统来保证应用代码及其配置数据传输和操作的安全。

这个系统主要涉及注册认证（CA）中心、卡片发行商、应用提供商、应用安装终端、智能卡（用户）5 个实体。其中卡片发行商处于中心地位：卡片制造商向卡片发行商提供智能卡；智能卡由卡片发行商向最终用户发放；注册/认证中心制作的应用操作证书 AOC（Application Operation Certificate）同样将提供给卡片发行商。图 2 给出了应用下载时各实体的关系。

3.1.1 关键密钥及其关系

（1）应用提供商 RSA 密钥对。应用提供商提供应用代码。为了保证应用代码在网络传输的过程中不被篡改，应用提供商必须产生一对 RSA 非对称密钥。私钥被用来对应用代码进行签名，公钥继而被用来验证应用代码的真实性和完整性。

（2）卡片发行商 RSA 密钥对。卡片发行商提供应用配置数据。为了保证应用配置数据在网络传输的过程中不被篡改，卡片发行商必须产生一对 RSA 非对称密钥。私钥被用来对应用配置数据进行签名，公钥继而被用来验证应用配置数据的真实性和完整性。

（3）卡片 RSA 密钥对和两个对称密钥。要使应用代码和应用配置数据在网络传输过程中不会泄露，应用代码和应用配置数据必须被加密。由于用 RSA 密钥来加密应用代码和应用配置数据的速度太慢，应用代码和应用配置数据采用 3DES 对称密钥算法来加密。为了保证加密使用的会话密钥的安全，卡片提供一对卡片 RSA 密钥对，卡片公钥被应用提供商和卡片发行商用来对会话密钥制作数字信封；卡片私钥在发卡时植入卡内，只有卡片自己才能从数字信封中取得会话密钥，这样就保证了应用代码和应用配置数据的机密性。

（4）CA 认证中心 RSA 密钥对。要验证应用代码和应用配置数据的完整性，智能卡必须知道卡片发行商公钥和应用提供商公钥。卡片发行商公钥和应用提供商公钥是通过 CA 认证中心制作的 AOC 传输到卡内的。为了保护卡片发行商公钥和应用提供商公钥在传输过程中不被篡改，CA 认证中心要提供一对 RSA 密钥对。私钥是用来对 AOC 进行签名，公钥在发卡时植入卡内，卡用它来恢复和验证由 CA 认证中心制作的 AOC。

3.1.2　应用在卡外的拆分封装成 AOC 和 ALU

一个编写好的应用，被多应用管理系统拆分成应用操作证书 AOC 和应用下载单元 ALU（Application Load Unit）两种类型的数据包。这两种格式的数据包可在非安全的网络环境下传输。

应用下载单元 ALU 分为应用代码下载单元 CALU 和应用配置数据下载单元 DALU 两类。图 3 给出了 CALU 的生成过程及其数据结构。应用配置数据下载单元 DALU 的生成过程与 CALU 的类似，其处理的数据对象是应用配置数据，对应的内容是应用配置数据签名、应用配置数据密文和对应用配置数据加密密钥的数字信封。

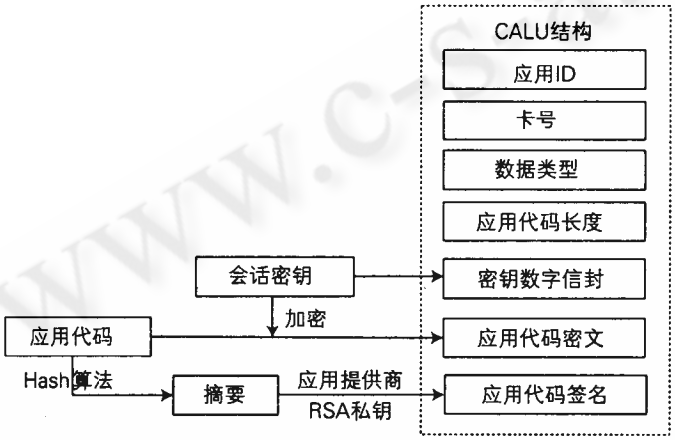

图 3　CALU 的生成及其数据结构

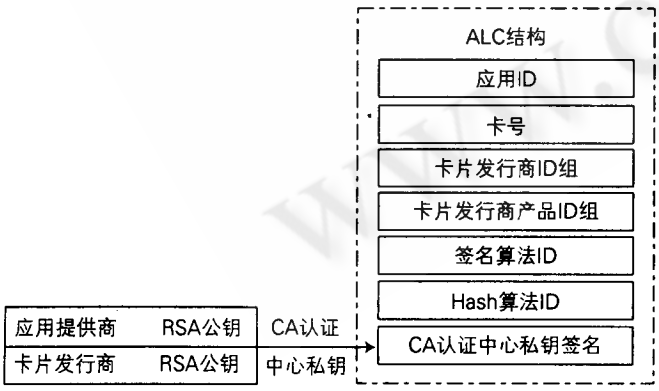

图 4　ALC 的生成及其数据结构

应用操作证书 AOC 包括应用下载证书 ALC 和应用删除证书 ADC。图 4 给出了 ALC 的生成过程及其数据结构。CA 认证中心利用得到的卡片发行商公钥、与应用代码相对应的应用代码头文件和应用提供商公钥及其相关的应用配置数据信息来完成应用的注册，并用这些信息来制作 AOC，这些证书再送给卡片发行商，由卡片发行商将 AOC 提供给应用安装单元生成机构。

3.1.3　卡内 AOC 和 ALU 还原成应用

应用操作数据 ALU 和应用操作证书 ALC 最终被提交给卡片进行处理，只有它们三者相互确认通过后才能还原成一个可供用户使用的应用。

ALU 和 ALC 在应用安装终端首先要进行完整性外部校验：即对在 ALU、卡片和 ALC 中的应用 ID、卡号、卡片发行商 ID、卡片发行商产品 ID、签名算法 ID 和 HASH 算法 ID 等进行三方比较。应用 ID 的比较保证了 ALU 与 ALC 都是相对于同一个应用的。卡号的比较保证了 ALU 和 ALC 都是特定于这张卡。ALC 中的卡片发行商 ID 组或卡片发行商产品 ID 组包含有多个卡片发行商 ID 或者多个卡片发行商产品 ID，通过比较，判断卡片中的卡片发行商 ID 或卡片发行商产品 ID 是否属于 ALC 中的卡片发行商 ID 组或卡片发行商产品 ID 组。这两个数据的比较保证了该卡是卡片发行商发行的。签名算法 ID 和 HASH 算法 ID 的比较保证了卡内有相应的算法能够识别卡片发行商生成的加密 ALU。这样一系列数据的比较保证了下载的 ALU 和 ALC 是特定于正在进行应用下载操作的这张智能卡的，实现了应用和卡片的双向身份认证。

3.2　基于 FAT12 文件系统的存储管理

文件系统是卡操作系统的关键组成部分，直接影响到操作系统的性能。“一卡多用”智能卡文件系统设计时符合 ISO 7816 标准，由主文件（Master File）、专用文件（Dedicate File）和基本文件（Elementary File）组成。文件系统采用 FAT12 对存储器进行管理，数据区被划分成同等大小的数据块，FAT 记录了为每个文件分配的数据块。系统保存并同时维护两个 FAT，互为备份，防止文件系统因 FAT 数据异常而崩溃。

应用代码是以主文件 MF 下的一个 EF 文件来存放的，应用配置数据则存放在主文件 MF 下的一个 DF 文件里面。DF 是连续存储的空间，存储了一个应用所

有要用到的数据,在 DF 文件头里面记录了存放应用代码 EF 文件的地址。一个具体应用在卡内是以一个 DF 文件及其对应的 EF 文件的形式存在的。MF 下可创建多个 DF(多应用)。

实现卡内应用的动态下载、更新和删除,必须记录卡内当前已经存在的应用。系统通过注册表文件来实现应用的物理存储空间的管理。注册表是在 MF 下的一个特殊的 EF 文件,它由注册表记录组成,其结构如下表。通过检索注册表中的应用 ID 来识别已安装的应用,通过检索注册表中已经分配的存储器的信息为新应用的安装分配一段连续的存储空间。设置注册表文件的机制,为多应用在卡上的安全管理提供了依据。只有卡上的公共操作系统在达到了相应的安全条件后才能操作此文件。

3.3 AAM 保证应用在卡内运行的安全

(1) 应用防火墙机制。从第 3 节的叙述可知应用不能直接读写存储器,而是通过 AAM 操作的。当任意一个应用被激活时,AAM 以相同的存储管理机制对待它们,形成相同的内存映象图。该应用看不到其它应用,整个硬件资源好像都分配给了它。为了保证当前应用不影响其它应用的操作环境,AAM 利用该应用的注册表记录中的 DF 文件的起始和结束物理地址等信息,实现了一个应用防火墙。

(2) 命令处理机制。传统 COS 中所有命令的执行方式是相同的:系统调用命令时,就把执行权限交给了命令程序,此时,命令程序对卡上的所有的硬软件资源拥有完全的控制权。而在"一卡多用"的 AAM 中定义了一个虚拟处理器,所有的命令都由 AAM 解释执行。命令被分为核心命令和用户命令两类。核心命令独立于具体的应用,在满足相应的安全状态条件后对所有的硬软件资源拥有完全的控制权,如应用管理命令(选择应用、装载应用和退出应用等)身份验证命令、通信传输命令等。用户命令是与具体的应用相联系的所有命令的集合,在执行时只能由处于激活状态的应用解释执行。

3.4 应用提供商利用 AAM 提供的 API 函数实现安全业务数据服务

已授权的应用拥有者对用户数据和交易的安全负全责,因为它可以读写分配给它的整个 DF 空间。AAM 提供了一系列的 API 函数,应用开发者可以直接利用它们实现一套完整的安全机制,也可以自己增加安全算法。这些 API 函数包括:

加解密算法类,如 DES 加解密函数、3DES 加解密函数、DES 和 3DES 签名生成函数等;

身份认证类,如内部认证函数、外部认证函数、Hash 摘要生成函数、取随机数函数等;

安全机制类,如防拔机制函数、锁卡函数、开锁函数等。

利用这些 API 函数,可实现卡和终端的相互认证,持卡人的身份认证、安全报文传输,数据的加解密等功能。应用提供商应根据各自部门和行业特点制定应用安全标准,确保卡上应用数据的安全,避免给本部门(行业)和持卡人带来经济损失。发行应用代码前要对其安全性进行严格审查和测试。

4 结束语

动态"一卡多用"与传统智能卡相比,在技术上有动态性强、平台移植性好和开发语言灵活等优点,在实际运用中有减少重复发卡、避免无序发卡、提高投资效率和方便用户持卡等优势。对"一卡多用"相关技术的研究对加快我国社会信息化的进程具有重大意义。

参考文献

1 吴东辉、周 捷、陈章龙, Java 卡的设计,微型电脑应用,2003,19(12):20~23。

2 MAOSCO Ltd. Guide to Generating Application Load Units. http://www.multos.com, 2004

3 史肖燕、熊璋、蒲菊华, 智能卡操作系统— BHCOS 的设计和实现, 计算机工程, 2003,29(2):207~209。

4 Mike Hendry 著,杨义先等译, 智能卡安全与应用, 人民邮电出版社, 2002。

5 徐中华、刘玉珍、张焕国, 一种新的"一卡多用"智能卡模型,计算机工程, 2003(4):43~45。