

Java 卡 COS 安全漏洞的一种探测与分析方法

徐江珮¹, 乌力吉¹, 杨向军², 王玉忠²

(1 清华大学 微电子学研究所, 北京 100084;

2 中钞信用卡产业发展有限公司 北京智能卡技术研究院, 北京 100032)

摘 要: 区别于已有的 Java 卡漏洞探测方法, 提出一种新的方法并以此来攻击卡片, 找出了市场上主流 Java 卡 COS 存在的一个漏洞. 通过进一步的实验, 分析该漏洞被利用后对卡片产生的安全威胁.

关键词: Java 卡; COS; 安全漏洞

中图分类号: TN402

文献标识码: A

文章编号: 1000-7180(2014)12-0071-04

A Method for Detection and Analysis of Security Flaw of Java Card COS

XU Jiang-pei¹, WU Li-ji¹, YANG Xiang-jun², WANG Yu-zhong

(Institute of Microelectronics, Tsinghua University, Beijing 100084, China)

Abstract: This article proposes a new method to detect the security flaw of Java Card and uses it to attack general Java Cards in the market, which ends up finding a security flaw of Java Card COS. What kind of security threat the flaw causes has been described through further tests.

Key words: Java Card; COS; security flaw

1 引言

国际上正在进行 IC 卡的 EMV 迁移, 国内也在全面推广 IC 卡的使用, 以适应全球磁条卡向芯片卡迁移的发展趋势. 国际上有两大 IC 卡安全管理体系: 一是 EMV Co, 它是管理、维护和完善 EMV 智能(芯片)卡的规格标准, 该标准中既包含芯片的安全, 也包含 COS 的安全, 并提供 EMV LEVEL 1 和 LEVEL 2 认证; 二是 CC 的 EAL 标准, 它是对信息技术产品进行安全性评价的国际通用准则, 分为 7 个等级(EAL1~7), 目前电子护照、电子身份证和银行 IC 卡都要求满足 EAL4+ 的标准. 现阶段, 国内生产的 IC 卡尚未获得国际上的认可, 原因主要是国内自主知识产权的 IC 卡难以满足国际 IC 卡安全测评标准.

研究和提高金融 IC 卡的安全性有助于保障国家的金融安全. 目前 Java 卡应用技术是金融 IC 卡

的主流技术. 金融 IC 卡普遍使用 Java 卡的原因主要有: Java 卡具有一卡多用和复用等特点, 并且 Java 卡应用的开发简单快速, 一次编写可处处运行, 跨平台性和兼容性较好^[1]. Java 卡的安全包括集成电路安全和嵌入式软件安全两部分. 嵌入式软件分为应用和 COS(Chip Operating System). 针对 Java 卡金融应用的安全评估, 国内权威检测机构已有相当的基础和积累, 而对于底层 COS 的检测目前进行得很少, 也是一个难点.

本文基于主动策略的方式, 通过查找漏洞检测点、编写检测用例, 对大量主流 Java 卡进行测试, 探查到 Java 卡 COS 的一个安全漏洞并给出相关的分析方法与过程. 该漏洞的探测方法能为相关权威检测机构对 Java 卡的安全评估提供参考. 本文的最后给出了修复此漏洞的建议, 旨在为国内自主 Java 卡的研发人员提供借鉴以提高自主 Java 卡的安全等级.

2 Java 卡结构

Java 卡分为集成电路和嵌入式软件两部分,如图 1 所示,Java 卡的结构中,集成电路之外的部分都属于嵌入式软件。

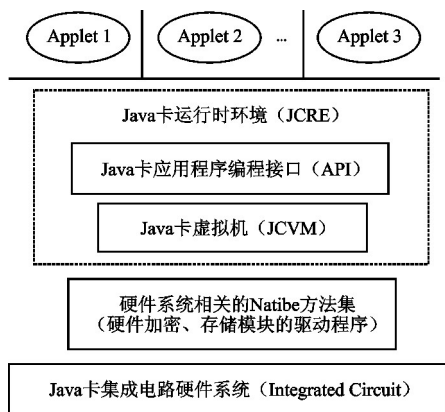


图 1 Java 卡结构图

Java 卡 COS 是处于集成电路和应用软件 (Applets) 之间的部分,它既支撑顶层的应用,又通过 I/O 接口与硬件电路模块进行交互。Java 卡 COS 主要包含 Java 卡虚拟机 (Java Card Virtual Machine, JCVM), 应用程序编程接口 (Application Programming Interface, API), 以及 Java 卡运行时环境 (Java Card Runtime Environment, JCRE)。JCVM 主要执行字节码和实现内存的自动管理机制^[2]; API 定义了类和接口的框架,为顶层应用提供应用程序接口; JCRE 则为 JCVM 和 API 的执行提供一个可靠的运行时环境。Java 卡 COS 在开发过程中参照的是 Sun 公司的 Java 卡规范,当前市面上普遍使用的 Java 卡所遵循的规范版本是 v2.2.1。本文主要基于 Java 卡虚拟机规范 v2.2.1,通过越界访问的手段,找出一个虚拟机在执行对象读取操作时存在的安全漏洞。

3 越界访问的一般方法

文献[3]中提到了一种越界访问 Java 卡内存数据的方法,由此发现某些 Java 卡的一个漏洞为:卡片在访问数组时只检查数组内存的逻辑边界而未检查其物理边界。探测该漏洞的具体做法是在一个恶意编写的 Applet 中,滥用事务机制来制造数组类型混淆,使得一个类型为 short 的数组引用和一个类型为 byte 的数组引用同时引用一个 byte 数组。当以类型为 short 的数组引用去访问 byte 数组时,若卡片未对数组的物理边界进行检查,则可以读取到

长度为该 byte 数组两倍的数据 (1 short = 2 bytes),实现了越界访问的目的。此漏洞已经被国外的研究者所公开,并且基于该漏洞的检测方法,国内检测机构已经利用相关检测工具针对此漏洞对 Java 卡进行检测。

4 篡改 (Converted Applet, CAP) 实现越界访问

Java 卡应用的编程语言是 Java 语言的子集。在编译的最初阶段,Java 卡程序与 Java 程序一样,源代码经过编译后都生成 class 文件。对于 Java 卡程序而言,最终要下载到卡片上,因此 class 文件需经过一个转换工具 converter 转换之后,生成以包为单位的 CAP 文件。CAP 文件是下载到卡片上的可执行文件,数据结构是十六进制的字节码流。在 Java 卡的安全保障措施中有一个校验字节码的环节,但现有的 Java 卡上没有字节码校验器,因此字节码校验只能在卡外进行^[4]。然而 Java 卡开发的过程中并未对所有下载到卡上的 CAP 文件进行字节码校验,因此下载到卡上的 CAP 文件存在被恶意篡改的风险。

本文探查并分析了另一种对 Java 卡内存进行越界访问的方法——篡改 CAP 文件。篡改 CAP 文件类似于篡改 PC 上的可执行程序,以达到修改程序功能的目的^[5]。在篡改完成后,对大量 Java 卡进行了系统性的测试。

4.1 探测方法

首先,在一个 Applet 里依次创建类 ClassA 的实例对象 p 和数组对象 a :

```
 $p = \text{new ClassA}();$ 
```

```
 $a = \text{new byte}[];$ 
```

并在类 ClassA 里声明 n 个 short 类型的实例域。虚拟机规范中,对实例域的引用通过以下结构来表示:

```
Constant_InstanceFieldref_info {
    ul tag
    class_refclass
    ul token
}
```

CAP 文件包含 11 个不同的组件,实例域的信息在 CAP 文件的常量池组件里。基于上述结构,通过查询 Applet 的 CAP 文件,在常量池组件里找到了类 ClassA 中 n 个 short 域的信息, n 个 short 域所对应的 Token 值分别为 $0 \sim n-1$ (在 CAP 文件中

以十六进制表示). 这些 Token 值是虚拟机在内存中读取实例域时使用的相对索引. 在 CAP 文件的常量池组件里, 将这 n 个实例域的 Token 值从 $0 \sim n-1$ 依次篡改改为 $n \sim 2n-1$, 将篡改后的 CAP 文件替换原来自动编译生成的 CAP 文件并下载到卡片上, 再试图去读对象 p , 并输出读取到的对象 p 中 n 个实例域的数据, 测试不同卡片的响应.

4.2 实验结果

在类 ClassA 里声明了 n 个实例域, 被解析后的 Token 值分别为 $0 \sim n-1$, 则在正常情况下, 用 n 以及 n 以上的 Token 值去索引这 n 个实例域应是无效的. 但是在对象 p 的实例域 Token 值被进行上述篡改之后, 一些 Java 卡在运行时并未出现异常, 并且可以顺利读到实例域数据. 由此找到某些 Java 卡存在的一个漏洞为: 虚拟机在访问对象实例域时未校验 Token 值的有效性.

某些 Java 卡的具体测试结果是: 对象 p 的 n 个实例域仍可被顺利读取, 而读取到的是与对象 p 相邻的对象 a 的信息. 在测试用例中, p 与 a 都是在类里声明的全局变量, 因此它们被存储在内存的永久存储区 EEPROM 中. 通过篡改对象 p 的实例域 Token 读到相邻对象 a , 首先读到的是 a 的对象头, 然后是 a 的域. 由此可以判断出在 Java 卡内存中这两个对象的存储结构以及内存的读取顺序如图 2 所示.

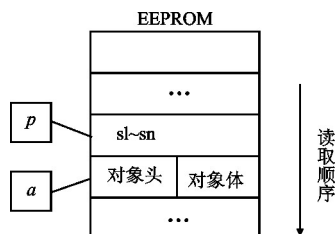


图2 EEPROM存储对象的结构及读取顺序

任何对象的对象头对外是不可视的, 读到了对象头信息就可以分析出不同 Java 卡存储对象头采用的数据结构了. 在测试用例中, 先后创建不同类型以及不同长度的数组对象, 经过多次测试比较, 得出 EEPROM 中存储的对象头的长度以及对象头中每个字节所代表的含义. 以某 Java 卡的实验结果为例, 对象头由 6 字节构成, 前两个字节标识对象的长度, 第三字节标识对象类型(临时对象/永久对象), 第四字节标识上下文信息, 最后两个字节标识数组的类型(byte、short、boolean).

5 安全威胁

为分析该漏洞对 Java 卡产生的安全威胁, 通过另外的实验来作进一步的分析. Java 卡 COS 有一个安全保护的机制——防火墙的隔离机制. 如图 3 所示, 防火墙阻拦的是跨上下文访问的行为.

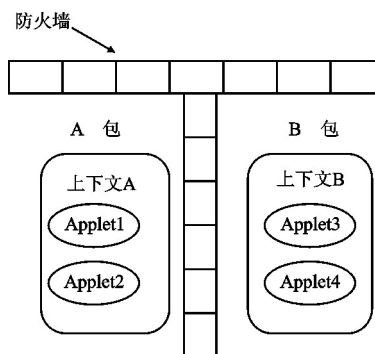


图3 Java卡的防火墙

A 包里的两个应用 Applet 1、2 属于同一个上下文, Applet 1 和 2 之间可以相互访问; 同样, B 包里的两个应用 Applet 3、4 属于同一个上下文, 因此 Applet 3 和 4 之间也可以相互访问. 但是防火墙会阻拦 A 包的 Applets 访问 B 包的数据. 利用上述漏洞进行了这样一个实验, 首先在 A 包的 Applet 1 里创建一个对象 obj_a , 在 B 包的 Applet 3 里创建一个对象 obj_b , 将这两个包连续下载到卡片上. 篡改 Applet 1 的可执行文件, 读到对象 obj_a 的对象头信息, 将对象头里标识对象长度的字节篡改为一个比较大的值, 再读取对象 obj_a , 由于读取的内存数据较大, 读到的数据中包含了 B 包的数据, 并且找到了 B 包中的对象 obj_b . 篡改对象 obj_b 的对象头中标识上下文的字节, 将其上下文改成与 A 包的上下文一致. 进行一系列恶意篡改之后, 通过 A 包里的 Applet 1 就可以直接访问 B 包里的对象 obj_b 了, 此时就相当于防火墙失效了.

6 抗攻击措施

实验证明, 用篡改 CAP 文件的方式找出的这一漏洞确实会对卡片的安全造成威胁, 因此可将其视为安全漏洞. 窃取/篡改卡片中的数据是攻击者攻击金融 IC 卡的主要意图^[6], 导致数据被越界访问的漏洞会对金融 IC 卡的安全造成重大威胁. 在 Java 卡开发者对 COS 进行开发时, 可以在虚拟机的执行程序里将常量池组件里实例域的 Token 值与同一对象中实例域的个数进行对比, 如果 Token 值的范围

超出了正常值就应该在运行时抛出异常,避免越界访问行为的发生。

7 结束语

本文针对 Java 卡 COS 进行了漏洞分析,首先通过篡改 CAP 文件找到卡片在执行时存在的一个漏洞,其次利用该漏洞做了进一步实验来攻击卡片,导致卡片的防火墙受到安全威胁,以此证明此漏洞为安全漏洞,最后提出了一种抗攻击措施。

参考文献:

- [1] 王晓亮,方勇,彭彦. Java 智能卡安全运行时环境的设计[J]. 网络安全技术与应用,2009,9(2):39-41.
- [2] Zeqiri Nderin, Luma Artan. Intelligent communication system, automation control, java card and security[C] // Mathematics and Computers in Science and Engineering. North Holland, 2008:117-121.
- [3] Wojciech Mostowski, Erik Poll. Malicious code on java card smartcards: attacks and countermeasures[J]. Smart Card Research and Advanced Applications, Lec-

ture Notes in Computer Science,2008(5189):1-16.

- [4] 吴俊军,马鑫龙. 一种混合模式的 Java 卡内字节码校验算法[J]. 微电子学与计算机,2009,26(4):152-155.
- [5] Yuchuan Wu, Yaqin Sun. Analysis and research of securing from attack for Java Card[C] // Proceedings of the International Conference on E-Business and E-Government, ICEE, Shanghai, 2010:1296-1298.
- [6] 周捷,吴东辉,陈章龙. Java 卡平台安全性的设计与实现[J]. 计算机工程与应用,2004,40(21):114-117.

作者简介:

徐江珮 女,(1990-),硕士研究生. 研究方向为 Java 卡的安全. E-mail:xjp11@mails.tsinghua.edu.cn.

乌力吉(通讯作者) 男,(1965-),副教授,博士生导师. 研究方向为集成电路与系统、商用信息安全.

杨向军 男,(1974-),高级工程师. 研究方向为 Java 卡嵌入式软件的开发与应用.

王玉忠 男,(1975-),高级工程师. 研究方向为 Java 卡嵌入式软件的开发与应用.

(上接第 70 页)

- [6] 赵晓华,李卫民,辛明军. 基于 TCPN 的 LBS 动态服务组合建模与验证[J]. 小型微型计算机系统,2013(1):96-99.
- [7] 王珏,向朝参. 语义 Web 服务发现研究现状与发展[J]. 计算机应用研究,2013,30(1):7-12.
- [8] 李建新,柯钢,杨怀德. 一种基于 OWL-S 领域本体的 Web 服务语义标注方法研究[J]. 现代计算机,2013(4):27-30.
- [9] 曾诚,韩光辉,李兵,等. 一种改进的基于路径的语义相似度计算算法[J]. 微电子学与计算机,2011,28

(8):135-141.

作者简介:

朱雪峰 男,(1989-),硕士研究生. 研究方向计算机网络. E-mail:zhuxuefeng01@yeah.net.

谢 谦 男,(1973-),博士研究生,副研究员. 研究方向计算机网络和操作系统.

戴桂兰 女,(1972-),博士研究生,副研究员. 研究方向为计算机网络和软件工程.