DOI: 10.3969/j.issn.1009-6868.2015.05.011
网络出版地址:http://www.cnki.net/kcms/detail/34.1228.TN.20150923.1005.004.html

安全可信智能移动终端研究

Security and Trusted Intelligent Mobile Terminal

中图分类号:TN929.1 文献标志码:A 文章编号:1009-6868 (2015) 05-0039-006

摘要: 从软件方案、基于可信执行环境(TEE)方案和基于典型安全元件(SE)方案3个方面对智能移动终端安全技术进行了探讨。软件层面探讨了一般运行环境中的安全技术,基于TEE的方案探讨了TEE的系统架构、隔离技术和安全执行技术,基于SE的方案探讨了基于本地SE和云端SE的安全增强技术。认为只有将可信硬件平台和可信软件加以结合,才能为智能移动终端提供完整的安全保障。

关键词: 智能移动终端;可信执行环境;可信计算;安全元件

Abstract: This paper discusses the smart mobile terminal security technology from three aspects: software solutions, solution based on trusted execution environment (TEE) and solution based on secure element (SE). A software-level solution involves security technology used in the rich execution environment. A solution based on TEE involves the system architecture of TEE, isolation technology of TEE and the trusted execution technology. A solution based on SE involves security-enhancement technology based on local SE and cloud of SE. A combination of trusted software and trusted hardware platform guarantees security for smart mobile terminals.

Key words: intelligent mobile terminal; trusted execution environment; trusted computing; secure element

张大伟/ZHANG Dawei
郭烜/GUO Xuan
韩臻/HAN Zhen
(北京交通大学,北京 100044)
(Beijing Jiaotong University, Beijing 100044, China)

随着移动互联网的发展,智能移动终端的数量急剧增加,功能也日益完善。2013年全球智能移动终端出货量近10亿部;全球计算平台(含PC和智能移动终端)中移动操作系统(Android和iOS)的占比超过50%[1];2013年中国智能移动终端用户规模为3.2亿,2014年已达10.6亿,较2013年增长231.7%[2]。截至2014年12月底,手机网民规模达5.57亿,较2013年底增加5 672万人。网民中使用手机上网人群占比由2013年的81.0%提升至85.8%[3]。

收稿日期:2015-03-02
网络出版时间:2015-05-01
基金项目:国家自然科学基金(61402035);新世纪优秀人才项目(NCET-11-0565);中央高校基本科研业务费专项资金资助项目(2015JBM041)

随着智能移动终端应用的普及,移动终端中存储的敏感信息越来越多,但丰富的通信和数据交换功能为信息泄露和恶意软件传播提供了通道,各种安全问题日益凸显。

1 智能移动终端的安全需求

智能移动终端已从过去的基本通信工具演变为工作、生活工具。它们已经包括多媒体播放、照相、定位、移动钱包、移动办公、移动医疗等新功能。随着用户敏感数据和关键业务在智能移动终端上的不断积累,智能移动终端也越来越需要被保护。

智能移动终端应用环境下的安全需求如下:

(1)开放环境下的安全需求

不同于传统手机的封闭系统,新型智能移动终端设备通常都是构建在提供开放式操作环境的操作系统之上,如Android、iOS操作系统。使用这些操作系统的一个主要优点是用户可以随时添加应用程序,同时可以几乎不必考虑对设备稳定性的影响。然而,这种开放式的环境也是设备暴露在不断增长的多种形式的攻击之下。

(2)数据安全需求

智能移动终端设备上存储着不断增长的个人信息(如联系人、邮件、照片等)甚至是敏感数据(证书、密码等)。为了防止在设备丢失、被盗或者其他不良情况,必须有足够的安全措施来保护这些隐私信息。

(3)安全连接的需求

通过多种网络技术如3G、4G或者Wi-Fi,以及个人通信手段如蓝牙、近场通信(NFC),越来越多的用户可以使用他们的设备进行P2P通信和访问网络。如何保证连接过程的安全,尤其是终端上的安全接入问题也有待进一步深入研究。

(4)交易安全需求

使用智能移动终端进行金融交易已经成为移动市场的主流。2014年底,移动金融整体用户规模达到

8.7亿，较年初翻一番[2]，越来越多的用户选择使用移动端金融交易服务。移动支付的实现方式包括远程支付和近场支付。在移动支付中需要保证信息的机密性、完整性和不可抵赖性、交易的真实性以及解决交易中的身份鉴别等问题。

（5）管理策略的安全需求

智能移动终端正不断被企业用于承载关键技术及核心应用，同时携带个人设备（BYOD）策略也被大量引入企业。为了防止企业数据泄露和个人使用环境中的恶意软件对企业数据的窃取，必须提高设备安全并引入有效的移动设备管理措施。

2 一般运行环境的终端安全技术

一般运行环境（REE）主要包括运行于通用嵌入式处理器中的一般操作系统（Rich OS）及其上的客户端应用程序。诸如Android、iOS等一般操作系统赋予了智能移动终端功能的可扩展性和使用的便利性。与此同时，也带来了多种安全威胁。

2.1 传统的设备访问控制

智能移动终端提供了包括密码配置、用户身份鉴别等传统的设备访问控制机制。以Android和iOS系统为例。Android系统提供了身份鉴别、口令设置、重鉴别和鉴别失败处理机制[4]。Android系统提供基本密码配置选项，包括设置图案密码、数字密码、混合密码等多种密码方式。有些机型还为用户配置了基于用户生物特征，如面部识别和指纹识别的身份鉴别机制。iOS的系统管理者可以设定密码强度，可确定用户频繁使用后需要设定新密码的周期[5]。Android和iOS的用户还可以设置用户错误登录的上限，以及超过这个上限后系统是否擦除设备信息。

2.2 设备数据加密机制

智能移动终端操作系统为设备中的数据提供了数据加密机制。

Android 3.0及之后的版本的系统提供了文件系统的加密机制[4]。所有的用户数据均可使用AES-128算法，以密码分组链接（CBC）模式进行加密。文件系统密钥通过使用由用户口令派生出的密钥以AES128算法进行保护。生成加密文件系统密钥的加密密钥时，采用标准的基于口令的密钥派生PBKDF2算法，由用户口令派生出加密密钥。

iOS系统中，所有用户数据强制加密[6]。每台iOS设备都配备了专用的AES-256加密引擎，它内置于闪存与主系统内存之间的直接存储器访问（DMA）路径中，可以实现高效的文件加密。加密解密所使用的密钥主要来自设备的唯一标识（UID）以及设备组标识（GID）。设备的UID及GID全部被固化在芯片内部，除了AES加密引擎，没有其他方式可以直接读取。只能查看使用它们进行加解密后的结果。每台设备的UID是唯一的。使用UID的加密方式将数据与特定的设备捆绑起来，因此，如果将内存芯片从一台设备整体移至另一台设备，文件将不可访问。除了GID及UID，其他加密使用的密钥全部由系统自带的随机数生成器产生。

除了iOS设备内置的硬件加密功能，iOS系统还提供了名为文件数据保护的数据保护方法，进一步保护设备闪存中的数据。每次在数据分区中创建文件时，数据分区都会创建一个新的256位密钥，并将其提供给AES引擎，以对文件进行加密。这些密钥被称作文件密钥。每个文件的文件密钥是不同的，被加密封装于文件的元数据中。

2.3 应用运行时的隔离机制

智能移动终端为在其上运行的应用程序提供了应用隔离机制。Android系统提供了沙盒机制，为每个应用在运行过程中提供了一个沙盒[7]。其具体的实现是，系统为每个应用提供了一个Dalvik虚拟机实例，使其独立地运行于一个进程，并为每个应用创建一个Linux底层的用户名，设置UID。具有相同用户签名的应用通过设置SharedUserID方式来共享数据和权限。iOS沙盒的实质是一个基于TrustBSD策略框架的内核扩展模块访问控制系统[9-10]。应用间不能查看或者修改数据和运行逻辑，并且应用在执行过程中也不可能查看到设备上已安装的其他应用。

2.4 基于权限的访问控制

智能移动终端为在其上运行的应用程序提供基于权限的访问控制机制。在Android系统中，每个应用程序都会有一个嵌入式的权限列表，只有用户授予了该项权限，应用才能使用该项功能[8]。iOS系统中，GPS定位功能、接受来自互联网的通知提醒功能、拨打电话、发送短信或电子邮件这4项功能需要授权使用[5]。

2.5 应用逆向工程的防止策略

智能移动终端的应用程序通常会使用各种手段来防止逆向。在Android系统中，通常的做法是应用程序的混淆和加壳技术。此外，还有使用动态链接（SO）库和采用Android类动态加载技术的方法[11]防止逆向。iOS系统中，通过使用统一资源定位（URL）编码加密、方法体方法名高级混淆和程序结构混排加密等方式防止逆向[5]。

2.6 系统安全更新

类似于桌面操作系统，智能移动终端系统具有不定期系统安全更新机制。通过不断的系统安全补丁或者发布带有新的安全机制的系统升级减少攻击的发生。

3 可信执行环境技术的终端安全技术

尽管在REE中采取了诸多安全措施来保障应用和数据的安全，众多

的攻击案例和系统漏洞表明，这些仍然无法保证敏感数据的安全性。因此开放移动终端组织（OMTP）首先提出了可信执行环境（TEE）概念。2010年7月，全球平台组织（GP）第一个提出了TEE标准[12]。

3.1 TEE概述

TEE是运行于一般操作系统之外的独立运行环境。TEE向一般操作系统提供安全服务并且与Rich OS隔离。Rich OS及其上的应用程序无法访问它的硬件和软件安全资源。TEE的架构如图1所示[13]。

图1 中，TEE向被称作可信应用程序（TA）的安全软件提供安全可执行环境。它同时加强了对这些可信应用程序中数据和资源的机密性、完整性和访问权限的保护。为了保证TEE的可信根，TEE在安全引导过程中进行认证并且与Rich OS分离。在TEE内部，每一个可信应用都是独立的。可信应用程序不能未经授权的访问另一个可信应用程序的安全资源。可信应用程序可以由不同的应用提供商提供。TEE中，通过TEE内部接口（TEE internal API）控制可信应用对安全资源和服务的访问。这些资源和服务包括密钥注入和管理、加密、安全存储、安全时钟、可信用户界面（UI）和可信键盘等。TEE将执行一个度量程序，其中包括功能性测试和安全性评估。

TEE提供了介于典型操作系统和典型安全元件（SE）之间的安全层。如果我们认为Rich OS是一个易于被攻击的环境，SE是一个能够抵抗攻击但是应用受限的环境，那么TEE就扮演着介于两者之间的角色。Rich OS、TEE、SE所处的位置和比较如图2所示[13]。

在一般情况下，TEE提供了一个比Rich OS更高安全等级的运行环境，但它的安全等级比SE所提供的要低。TEE提供的安全性足以满足大多数的应用。此外，TEE提供比SE

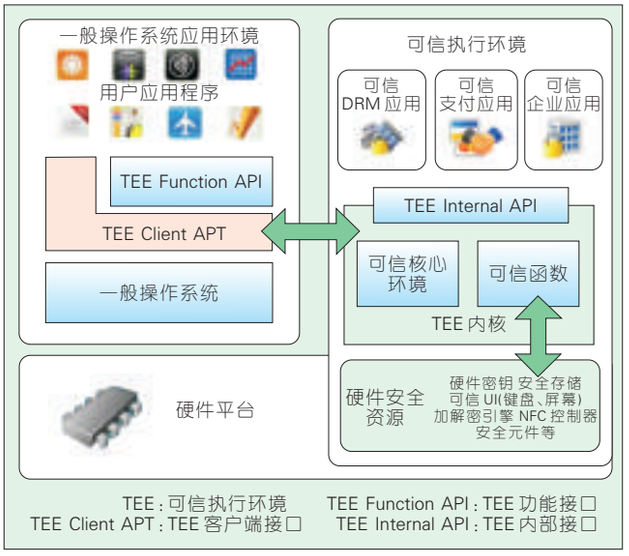


图1▶ TEE架构

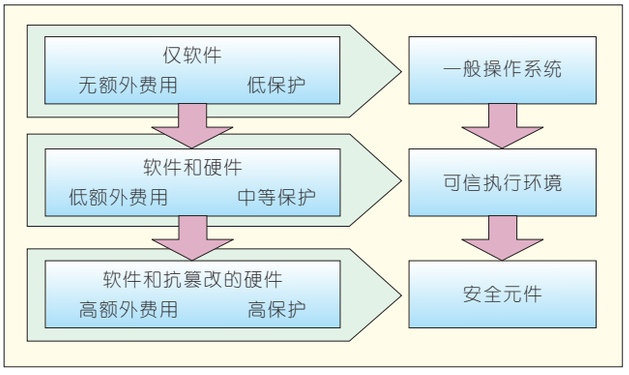


图2▶ Rich OS、TEE、SE所处的位置和比较

更强大的处理能力和更大的可访问的内存空间。由于TEE比SE支持更多的用户接口和外围连接，它允许在其上开发有一定用户体验的安全程序。此外，因为TEE与Rich OS是隔离的，它能够抵御在Rich OS中发生的软件攻击。

3.2 TEE系统架构

3.2.1 TEE硬件架构

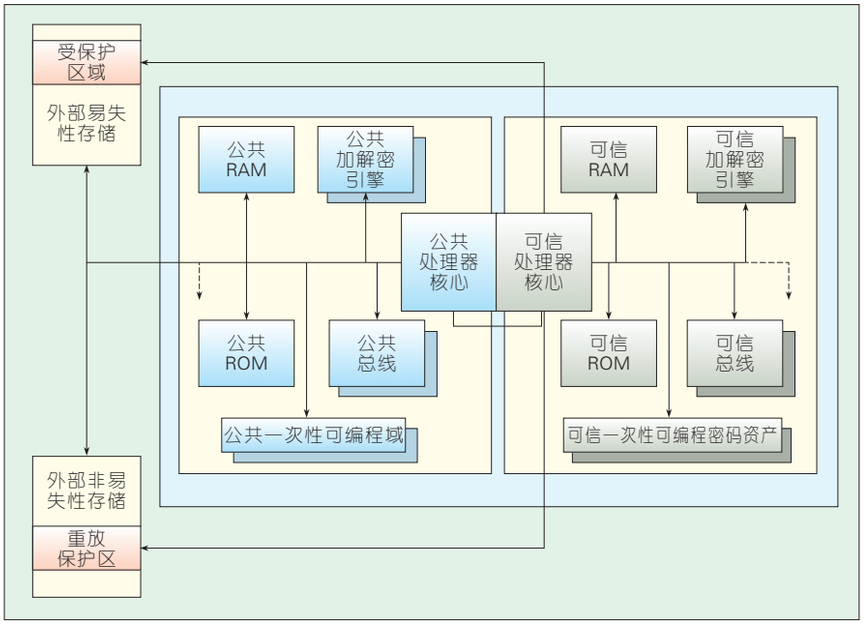
芯片级别的TEE硬件它连接着如处理器、RAM和Flash等组件[14]。REE和TEE都会使用一些专有硬件，如处理器、RAM、ROM和加解密引擎。处理器之外的实体被称作资源。一些能够被REE访问的资源也能够被TEE访问，反之，REE不能访问未经TEE授权的TEE资源。可信资源仅能由其他可信资源访问，从而保证了与一般操作系统隔离形成封闭系统。一个封装在片上系统（SoC）上的资源组合结构如图3所示[14]。

3.2.2 TEE软件架构

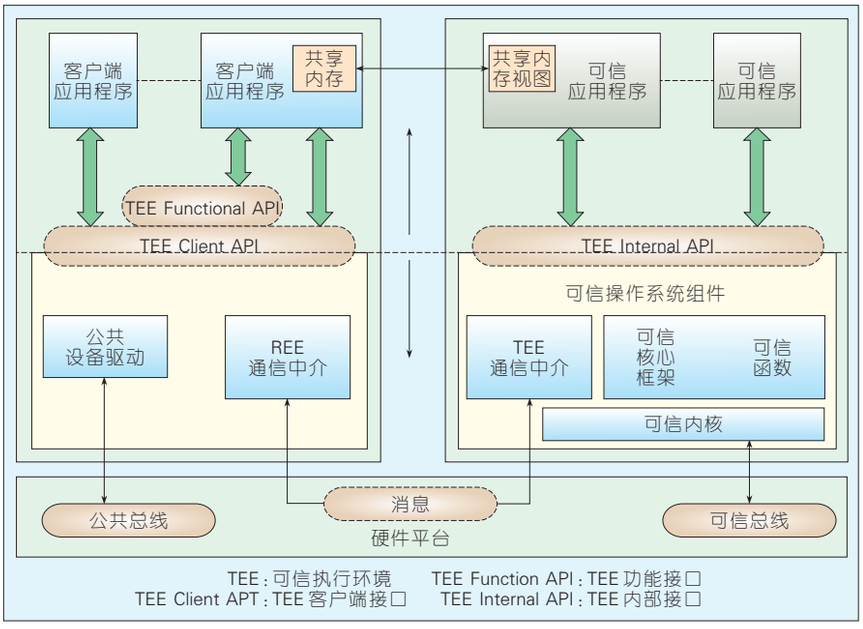
TEE系统软件[14]架构如图4所示。TEE软件架构的目标是为可信应用程序提供隔离的和可信服务，并且这些服务可以间接的被客户端应用程序（CA）使用。

TEE软件架构包括4部分：REE调用接口、可信操作系统组件、可信应用程序和共享内存。

REE调用借口包括两类API接口，TEE功能接口（TEE Function API），TEE客户端接口（TEE Client API）和一类通信中介。TEE Function API向CA提供一套操作系统友好



▲图3 REE和TEE的硬件架构



▲图4 TEE系统软件架构

API。允许程序员以类似于编写操作系统应用的方式调用TEE服务，如进行密码运算和存储。TEE Client API是一个低级的通信接口。它被设计用于使运行于Rich OS中的应用程序访问和交换运行于TEE中的可信应用程序中的数据。REE通信中介提供了CA和TA之间的消息支持。

在TEE内部有两类不同的软件结构：可信操作系统组件和可信应用程序。可信操作系统组件由可信核心框架、可信函数和TEE通信中介组成。可信核心框架向可信应用程序提供了操作系统功能，可信函数向开发者提供功能性调用。TEE通信中介与REE通信中介一同工作，为CA和TA之间的信息交互提供安全保障。可信应用程序是调用可信操作系统组件的API的内部应用程序。当一个客户端应用程序与一个可信应用程序开启一个会话进行交互时，客户端应用程序与可信应用程序的一个实例进行连接。每个可信应用程序的实例都与其他所有的可信应用程序的实例在物理内存空间上隔离。共享内存能够被TEE和REE共同访问，它提供了允许CA和TA之间大量数据有效快速交互的能力。

3.3 使用TrustZone技术的TEE实现

得益于ARM优秀的设计和商业模式，ARM架构非常适用于智能移动终端市场。因此，目前智能移动终端所使用的CPU内核以ARM架构居多。ARM TrustZone技术是ARM提出的系统范围的安全方法。TrustZone技术包括在ARM处理器架构和系统架构上添加的处理器的安全扩展、附加总线等技术。使用ARM TrustZone技术构建TEE是绝大多数智能移动终端的实现方式。

3.3.1 TrustZone的隔离技术

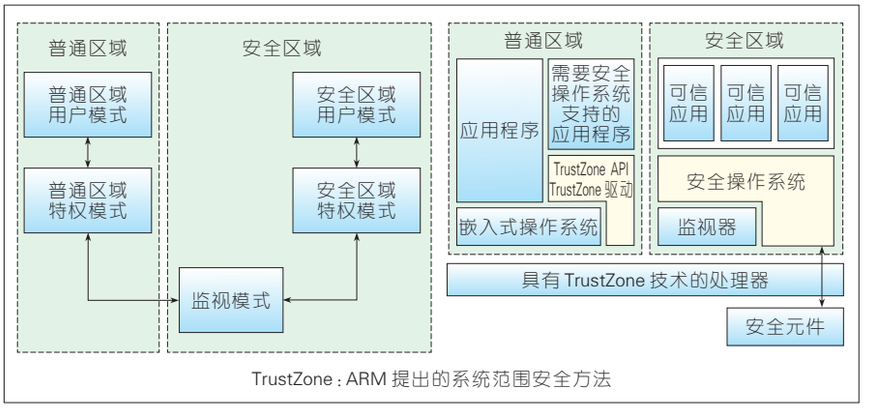
TrustZone技术的关键是隔离[15]。它将每一个物理处理器核心划分为两个虚拟核心，一个是非安全核心（Non-secure），另一个是安全核心（Secure）。同时提供了名为Monitor模式的机制来进行安全上下文切换。TrustZone技术隔离所有SoC硬件和软件资源，使它们分别位于两个区域（用于安全子系统的安全区域以及用于存储其他所有内容的普通区域）中。支持TrustZone总线构造中的硬件逻辑可确保普通区域组件无法访问安全区域资源，从而在这两个区域之间构建强大边界。将敏感资源放入安全区域的设计，以及在安全的处理器内核中可靠运行软件可确保资产能够抵御众多潜在攻击，包括通常难以防护的攻击如使用键盘或触摸屏输入密码等。TrustZone技术的硬

件和软件架构如图5所示[15]。

3.3.2 TrustZone技术中的安全启动

安全可信系统周期中一个重要的阶段是系统启动过程。有许多攻击尝试在设备断电后破解软件。如果系统从未经检验真实性的存储上引导镜像，这个系统就是不可信的。

TrustZone技术为可信域内的所有软件和和普通区域可能的软件生成一条可信链。这条可信链的可信根是难以被篡改的。使用TrustZone技术的处理器在安全区域中启动。使用TrustZone技术的处理器的启动过程如图6所示[15]。



▲图5 TrustZone硬件(左)和软件(右)架构

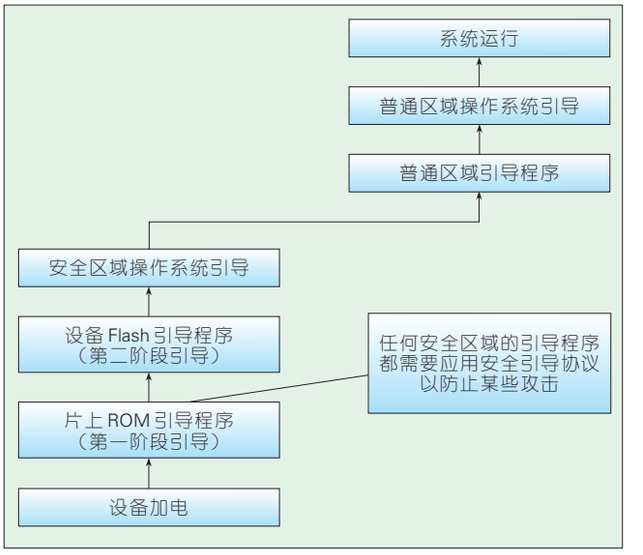


图6▶ TrustZone技术的典型启动序列

4 基于安全元件的终端安全技术

4.1 安全元件概述

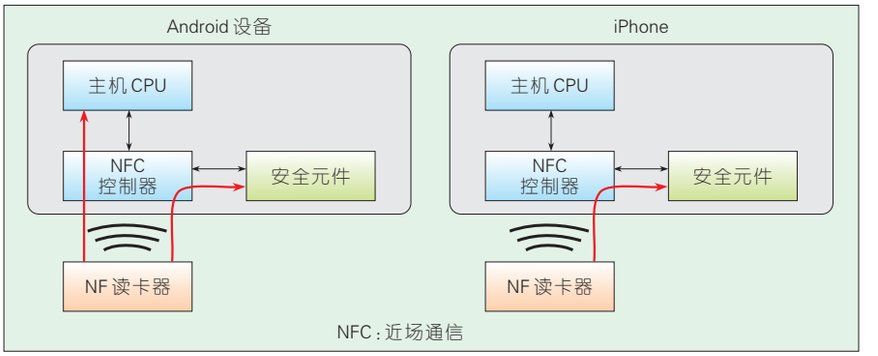
GP组织将安全元件(SE)定义为由硬件、软件和能够嵌入智能卡级应用的协议组成的防篡改组合。它可以通过一组由可信方预设的安全规则和要求来保护应用程序和其机密信息。SE的典型实现包括SIM/UICC卡、嵌入式SE和可移动存储卡。SE提供比TEE更高的安全级别，但与此同时它的花费也最高。在近场移动支付中，通常使用SE模拟非接触卡。SE与终端进行通信，发送查询响应，生成动态密码等。最新的安全方案是使用基于主机的卡模拟方式(HCE)，这种方式将安全存储和运行环境转移到云端，而不是存储在本地的SE中。

4.2 基于本地SE解决方案

在SE的实现方式中，通常把嵌入在智能设备中的SE和嵌入在运营商SIM卡中的SE称为本地SE。以谷歌钱包(Google Wallet)和ApplePay为例，来探讨在移动支付中基于本地SE的解决方案。

谷歌钱包1.0版本和ApplePay都是基于本地SE实现的。他们的工作方式如图7所示。

在谷歌钱包1.0中，Android设备的NFC控制器工作在卡模拟模式。在SE中存储的移动支付应用程序模拟非接触卡片，使用标准的应用协议数据单元(APDU)指令与终端进行通信。ApplePay中使用本地设备SE执行卡模拟和安全存储。在许多方面它与谷歌钱包1.0类似，但也有重要的差异。ApplePay在SE中不存储真实的主账号(PAN)，这与谷歌钱包1.0完全相反。ApplePay存储的是符



▲图7 Google钱包1.0和ApplePay工作方式

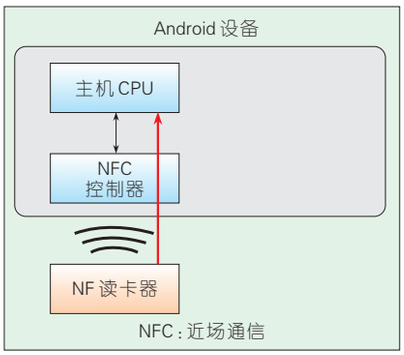
合EMVco令牌化标准[16]的令牌。在交易过程中，这个令牌被发送给终端。在授权流程中，网络识别令牌，去令牌化，生成真正的PAN，将PAN交给发卡行以授权。

4.3 基于云的SE解决方案

基于本地SE的解决方式本质上是安全的，这是它的一个非常大的优势。然而，它也有一个很大的缺点。SE的拥有者决定了市场准入。其他所有人都需要通过复杂的商业模式、合作方式和依赖关系进入市场。这让整个过程变得复杂而昂贵。此外，SE本身的存储容量和处理速度有限也是这种方式的一个缺点。

一种可行的解决方案是使用基于主机的卡模拟方式。Android 4.4及其后版本的操作系统提供了HCE模式的API[17]。HCE模式的运行方式如图8所示。

当消费者把手机放置在NFC终端上时，NFC控制器将所有的数据直接发送到直接运行应用程序的主机CPU中。然后Android应用程序（移动钱包）和特定的支付程序开始进行处理，进行卡仿真，进行请求和响应。由于主机CPU本身是不安全的，因此任何真实的支付凭证不会存储在手机钱包中。以谷歌钱包3.0为例，谷歌将所有这些真实的数据托管到云服务中，在那里进行安全存储和安全处理。从本质上来说，这是一种基于云的SE方式。实现了从基于本地的SE到基于云的SE的转变。



▲图8 HCE模式运行方式

这种方法也有它的缺点，如安全性和交易过程中需要网络连接。同时还需要使用如支付卡令牌化的技术来保证它的安全性。然而另一方面，这种方式可以使商业模式、合作关系变得简单，而且没有对本地SE的接入限制。这使得应用提供商可以轻松的提供服务。

5 结束语

本文从软件方案、基于TEE的方案和基于SE的方案3个层面对智能移动终端安全技术进行了探讨。软件层面上，在一般运行环境中，主要使用传统的设备访问控制、数据加密、应用运行时隔离机制、基于权限的访问控制、逆向工程的防止、系统安全更新等措施保护智能移动终端的安全。在基于TEE的方案中，使用特殊的软硬件体系结构、安全隔离和安全启动机制等来保护智能移动终端的安全。在基于SE的方案中，使用了基于本地SE和基于云端SE的安全隔离和可信执行技术来保证智能移动终端的安全。本文认为智能移动终端的安全解决方案是多层面立体式的解决方案。任何一个层面都有安全性或使用方便性的不足。只有将软件和可信硬件平台加以结合，才能为智能移动终端提供完整的安全保障。

参考文献

[1] 工业和信息化部电信研究院. 移动互联网白皮书（2014年）[EB/OL]. [2015-03-01]. http://www.miit.gov.cn/n11293472/n11293832/n15214847/n15218338/index.html
[2] TalkingData. 2014移动互联网报告 [EB/OL]. [2015-03-01]. https://www.talkingdata.com/index/#/datareport/-1/zh_cn
[3] CNNIC. 第35次中国互联网络发展状况统计报告 [EB/OL]. [2015-03-01]. http://www.cnnic.net.cn/hlwfzyj/hlwxzbg/
[4] Android System and kernel security [EB/OL]. [2015-03-01]. http://source.android.com/devices/tech/security/overview/kernel-security.html
[5] iOS Security Overview [EB/OL]. [2015-03-01]. https://developer.apple.com/library/ios/documentation/Security/Conceptual/Security_Overview/Introduction/Introduction.html
[6] iOS Security February 2014 [EB/OL]. [2015-03-01]. http://www.apple.com/ipad/business/docs/iOS_Security_Feb14.pdf
[7] JOSHUAJDRAKE P, OLIVAFORA P, LANLIER Z, et al. Android Hacker's Handbook [M]. John Wiley & Sons, Inc., 2014
[8] ENCK W, ONGTANG M, MCDANIEL P. Understanding android security [J]. IEEE security & privacy, 2009,250(1): 50-57
[9] 凌宁, 张文, 牛少彰, 等. 基于iOS系统的安全性研究 [J]. 中国电子商情·通信市场, 2013,12(4):91-95
[10] iOS Technology Overview [EB/OL]. [2015-03-01]. https://developer.apple.com/library/ios/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/Introduction/Introduction.html
[11] 梅瑞, 武学礼, 文伟平, 等. 基于Android平台的代码保护技术研究 [J]. 信息网络安全, 2013,23(7):10-15
[12] Trusted execution environment [EB/OL]. [2015-03-01]. http://en.wikipedia.org/wiki/Trusted_execution_environment
[13] The Trusted Execution Environment White Paper [EB/OL]. [2015-03-01]. www.globalplatform.org
[14] TEE System Architecture Version 1.0 [EB/OL]. [2015-03-01]. http://www.globalplatform.org/specificationsdevice.asp
[15] ARM Security Technology [EB/OL]. [2015-03-01]. http://infocenter.arm.com/help/topic/com.arm.doc.prd29-genc-009492c/PRD29-GENC-009492C_trustzone_security_whitepaper.pdf
[16] EMV Payment Tokenisation Specification-Technical Framework [EB/OL]. [2015-03-01]. http://www.emvco.com/download_agreement.aspx?id=945
[17] Host-based Card Emulation [EB/OL]. [2015-03-01]. http://developer.android.com

作者简介

张大伟，北京交通大学计算机与信息技术学院讲师；主要从事可信计算、智能卡安全方向的教学、科研工作。

郭烜，北京交通大学计算机与信息技术学院在读硕士研究生；从事移动安全方向的研究。

韩臻，北京交通大学计算机与信息技术学院教授、博士生导师，中国计算机学会信息保密专委会副主任委员，教育部高等学校信息安全专业教学指导委员会副主任委员；从事信息安全体系结构和可信计算技术方面的研究和教学工作。