软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2016,27(6):1366−1383 [doi: 10.13328/j.cnki.jos.005000]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

基于 TrustZone 的可信移动终端云服务安全接入方案*

杨 波[1,2], 冯登国[1,3], 秦 宇[1], 张英骏[1,2]

[1](中国科学院 软件研究所 可信计算与信息保障实验室,北京 100190)

[2](中国科学院大学,北京 100049)

[3](计算机科学国家重点实验室(中国科学院 软件研究所),北京 100190)

通讯作者: 杨波, E-mail: yangbo@tca.iscas.ac.cn

摘 要: 可信云架构为云计算用户提供了安全可信的云服务执行环境,保护了用户私有数据的计算与存储安全.然而在移动云计算高速发展的今天,仍然没有移动终端接入可信云服务的安全解决方案.针对上述问题,提出了一种可信移动终端云服务安全接入方案.方案充分考虑了移动云计算应用背景,利用 ARM TrustZone 硬件隔离技术构建可信移动终端,保护云服务客户端及安全敏感操作在移动终端的安全执行.结合物理不可克隆函数技术,给出了移动终端密钥与敏感数据管理机制.在此基础上,借鉴可信计算技术思想设计了云服务安全接入协议.协议兼容可信云架构,提供云服务端与移动客户端间的端到端认证.分析了方案具备的 6 种安全属性,给出了基于方案的移动云存储应用实例,实现了方案的原型系统.实验结果表明:可信移动终端 TCB 较小,方案具有良好的可扩展性和安全可控性,整体运行效率较高.

关键词: 移动云计算;可信计算;可信移动终端;安全接入;TrustZone;物理不可克隆函数(PUF)

中图法分类号: TP309

中文引用格式: 杨波,冯登国,秦宇,张英骏.基于 TrustZone 的可信移动终端云服务安全接入方案.软件学报,2016,27(6):1366−1383. http://www.jos.org.cn/1000-9825/5000.htm

英文引用格式: Yang B, Feng DG, Qin Y, Zhang YJ. Secure access scheme of cloud services for trusted mobile terminals using TrustZone. Ruan Jian Xue Bao/Journal of Software, 2016,27(6):1366−1383 (in Chinese). http://www.jos.org.cn/1000-9825/5000.htm

Secure Access Scheme of Cloud Services for Trusted Mobile Terminals using TrustZone

YANG Bo[1,2], FENG Deng-Guo[1,3], QIN Yu[1], ZHANG Ying-Jun[1,2]

[1](Trusted Computing and Information Assurance Laboratory, Institute of Software, The Chinese Academy of Sciences, Beijing 100190, China)

[2](University of Chinese Academy of Sciences, Beijing 100049, China)

[3](State Key Laboratory of Computer Science (Institute of Software, The Chinese Academy of Sciences), Beijing 100190, China)

Abstract: Trusted cloud architecture provides isolated execution environment for trusted and secure cloud services, which protects the security of cloud users' data computation and storage. However, with the rapid development of mobile cloud computing, there is currently no secure solution for mobile terminals accessing trusted cloud architecture. To address this issue, this research proposes a secure access scheme of cloud services for trusted mobile terminals. By fully considering the background of mobile cloud computing, an architecture of trusted mobile terminal is constructed using ARM TrustZone hardware-based isolation technology that can prevent the cloud service client and security-sensitive operations on the terminal from malicious attacks. Leveraging physical unclonable function (PUF), the key and

* 基金项目: 国家自然科学基金(91118006, 61202414, 61402455); 国家重点基础研究发展计划(973)(2013CB338003)

Foundation item: National Natural Science Foundation of China (91118006, 61202414, 61402455); National Program on Key Basic Research Project of China (973) (2013CB338003)

收稿时间: 2015-08-15; 修改时间: 2015-10-09; 采用时间: 2015-12-05; jos 在线出版时间: 2016-01-21

CNKI 网络优先出版: 2016-01-22 11:20:04, http://www.cnki.net/kcms/detail/11.2560.TP.20160122.1120.013.html

sensitive data management mechanism is presented. Based on the trusted mobile terminal and by employing trusted computing technology, the secure access protocol is designed. The protocol is compatible with trusted cloud architecture and establishes an end-to-end authenticated channel between mobile cloud client and cloud server. Six security properties of the scheme are analyzed and an instance of mobile cloud storage is provided. Finally a prototype system is implement. The experimental results indicate that the proposed scheme has good expandability and secure controllability. Moreover, the scheme achieves small TCB for mobile terminal and high operating efficiency for cloud users.

Key words: mobile cloud computing; trusted computing; trusted mobile terminal; secure access; TrustZone; PUF

随着云计算技术、移动终端设备、移动通信技术和移动互联网应用的高速发展,移动云计算(mobile cloud computing,简称 MCC)概念正在逐渐影响人们的日常生活.国际移动云计算论坛[1]和 Intel Aepona[2]给出的相关定义指出,移动云计算是移动终端设备通过移动云应用将数据处理和数据存储外包给云中资源丰富的计算平台的一种综合性技术.移动云计算可以有效降低移动设备计算资源、存储资源和电量的开销,提升复杂应用在移动终端的可用性[3].相比于 PC 平台云计算服务,移动云计算呈现给用户的多为软件即服务(software as a service,简称 SaaS)的层次[4],用户使用移动设备瘦客户端软件或 Web 浏览器通过无线网络访问远程云服务.近年来,移动云计算推广涉及的领域包括云办公、云邮件、云存储、云支付、云游戏和云视频等,各公司也为移动云计算推出了相应的支持技术和产品,其中包括苹果的 iCloud、谷歌的 Cloud Console、微软的 OneDrive 和亚马逊的 AppStream 等,它们极大地提升了移动用户体验云服务的便捷性.

根据市场研究公司 Visiongain 发表的移动云计算行业市场报告预测[5],移动云计算市场正进入快速增长期,到 2016 年,移动云计算服务年收入将达到 450 亿美元.

然而,伴随着移动云计算的普及,移动用户的信息安全正面临着日益严峻的威胁.除了传统云计算云端安全问题之外,移动终端的安全问题为移动云计算安全提出了新的挑战,也为潜在的敌手提供了新的突破口.在云计算安全研究中,用户的数据安全和隐私保护被作为云安全技术框架的主要目标[6],但移动终端安全的脆弱性可能破坏整个云计算框架的鲁棒性,造成用户的隐私数据泄露,甚至影响云端主机的安全和稳定.当前,恶意代码对移动终端软件和系统的攻击屡见不鲜,一旦用户的移动设备被攻破,敌手可以轻松窃取用户的账户口令,冒名访问用户的的敏感云服务进而获取其在云端存储的私有数据,导致隐私信息的泄露.2014 年 9 月,黑客攻击了多个苹果 iCloud 账号,导致詹妮弗·劳伦斯等好莱坞明星裸照被泄露于互联网.在云计算环境下,上述安全问题来源于云服务缺乏对移动用户有效的安全认证.在移动终端及应用得不到认证保护的情况下,用户口令极易被盗用,云端无法与移动用户建立足够的信任关系[7],使得传统的安全认证机制很难发挥作用.云安全联盟(cloud security alliance,简称 CSA)在发布的最新版云计算服务安全实践手册《云计算安全指南(v3.0)》[8]中,着重强调了数据安全、应用安全和身份授权对于云安全的重要性,只有这 3 方面同时满足要求,移动云计算用户的信息安全才可能得到保障.

在现有的安全技术中,可信计算[9]和可信虚拟化[10]被广泛应用于构建可信云[11−15].可信云架构可以良好地保护用户数据在云端主机上的机密性和完整性,能够提供用户 PC 端与云端主机之间的安全证明,但目前尚无成熟有效的可信移动终端解决方案能够保证移动设备安全接入可信云服务.传统的传输层安全协议如 TLS (transport layer security)等可以被用于保护用户数据在移动终端与云端主机之间网络传输的安全,借助 PKI (public key infrastructure)技术,可以进一步实现对移动云计算参与实体的认证.然而这些技术的安全性基于移动终端系统和应用程序均为安全的前提假设下,在针对移动设备攻击极为活跃的今天,这些假设是很难被满足的.因此,需要面向移动云计算场景设计一种云服务安全接入方案,在保证移动终端可信的基础上构建移动设备客户端与云服务之间端到端的信任连接.

面对层出不穷的移动系统及应用安全漏洞,在移动终端设备上构建可信执行环境(trusted execution environment,简称 TEE)成为近年来一个备受关注的研究领域.隔离于通用执行环境(rich execution environment,简称 REE),TEE 旨在保护安全敏感的代码执行和相关数据信息免受恶意敌手的攻击和破坏[16].TEE 是建立可信移动终端的基础,也是建立移动云计算安全架构的重要环节.作为 ARM 架构的安全扩展技术,TrustZone[17]提供

了基于硬件的隔离机制,目前已被移动嵌入式设备广泛支持,各大移动设备厂商正积极研发基于TrustZone的安全应用.TrustZone 技术可以被用于构建灵活高效的 TEE,但尚无统一标准的构建方法.此外,可信移动终端的构建需要可靠的信任根和密钥与数据管理机制,TrustZone 技术本身并不能提供这些功能,物理不可克隆函数(physical unclonable function,简称 PUF)[18]可作为 TrustZone 技术的补充,实现密钥与敏感数据管理机制,进而为移动终端提供良好的信任根.本文将利用 TrustZone 技术和 PUF 技术构建可信移动终端.

本文的主要贡献如下:

- 面向移动云计算场景,提出了基于 TrustZone 技术构建可信移动终端的方法,确保云服务客户端程序及相关敏感操作在移动终端运行的安全与可靠;
- 提出了基于 PUF 技术的密钥与敏感数据管理机制,该机制配合 TrustZone 技术为可信移动终端及云服务安全接入提供信任根功能;
- 提出了一种可信移动终端云服务安全接入协议,利用可信计算技术,在云服务端与移动客户端间建立端到端的双向认证通道,保护用户数据和云服务访问请求在接入过程中的认证性、机密性和完整性.该协议兼容可信云架构;
- 给出了一种基于本文方案的移动云存储应用实例设计,证明了方案具有良好的可用性与可扩展性;
- 实现了本文方案的原型系统.实验评估表明,本文方案在整体上具有良好的运行效率.

本文第 1 节讨论本研究领域的相关工作.第 2 节介绍预备知识、涵盖可信计算技术、TrustZone 技术和 PUF 技术.第 3 节提出方案的系统模型与假设.第 4 节从可信移动终端体系结构、密钥与敏感数据管理、云服务安全接入协议和安全性分析 4 个方面详细阐述可信移动终端云服务安全接入方案的设计.第 5 节提出基于本文方案的移动云存储应用实例.第 6 节是原型系统的实现.基于该系统的方案评估在第 7 节给出.第 8 节对本文方案涉及的匿名性问题展开讨论.第 9 节对全文进行总结,并展望未来的研究工作.

1 相关工作

为了从计算机底层硬件解决信息安全问题,可信计算概念被提出并在科研和产业界得到了一定的推广.国际可信计算组织 TCG(Trusted Computing Group)针对 x86 硬件平台推出了可信平台模块(trusted platform module,简称 TPM)的安全解决方案,由 TCG 于 2003 年制定的 TPM1.2 主规范[19]经过多次修订在 2009 年被接收为 ISO 国际标准,2013 年,TCG 正式发布了最新的 TPM2.0 标准[20].在我国,国家密码管理局于 2007 年提出了具有自主知识产权的可信密码模块(trusted cryptography module,简称 TCM)[21]和相关接口规范.作为基础安全技术,可信计算的一个重要应用场景是构建可信虚拟化平台,虚拟可信平台模块 vTPM[22]可以用于保护虚拟机监视器的安全,TrustVisor[23]通过创建虚拟 TPM 的实例为隔离的代码提供可信服务.由于在安全隔离、安全干预和数据保护等方面的优势,基础设施虚拟化技术被云计算架构广泛使用,利用可信虚拟化技术构建可信云计算环境成为近年来的研究热点.文献[12]概述了可信云计算平台(trusted cloud computing platform,简称 TCCP)理念,TCCP 通过将可信平台的功能延伸至云基础设施的方式为用户虚拟机提供封闭的运行环境,可以有效保护用户数据的机密性和完整性,并保证云端主机的安全.武汉大学赵波教授等人[11]总结了可信云计算环境构建的技术方法和面临的挑战.TrustCloud[14]为云计算设计了一种信任构建和安全审计的框架.Cloud Terminal[15]使用可信证明方法将用户敏感应用的数据处理安全外包给云服务提供商,用户本地主机只进行界面显示.CloudProxy[13]借助可信计算技术在云端主机与用户主机间建立了端到端的信任连接,保护用户数据在传输和云端运行过程中的安全.然而,上述可信云计算环境的构建方法都是针对 x86 硬件平台设计的,移动终端设备如何安全有效地接入可信云环境,仍是一个有待解决的问题.

在移动云计算发展迅速的今天,移动云计算安全受到了越来越多人的关注.相关研究[24,25]指出:移动用户的私有数据在移动终端、云端主机和通信信道上的机密性和完整性,是移动云计算安全的关键.可信虚拟化技术和可信云架构可以保护用户数据在云端主机的安全,但目前缺乏配套的、能够在移动终端及移动网络通信中保护用户数据且面向云环境设计的可信解决方案.为移动终端构建云服务安全接入方法,是解决上述问题的一个

重要途径,而可信移动终端的设计与实现是构建该方法的基础.文献[26]给出了基于移动操作系统访问控制策略的可信安全隔离方法,该类方法建立在移动操作系统安全的前提下.然而,成功利用 Android 系统漏洞实施攻击的事件层出不穷,操作系统本身并不能提供高强度的安全.对于可信移动终端的研究,TCG 继承了传统 PC 平台的可信计算思路,从底层硬件出发,为移动终端发布了移动可信模块(mobile trusted module,简称 MTM)规范[27],但由于需要依赖额外的硬件模块,该规范并没有在移动产业界得到推广.

相比于 TCG 的可信计算思路,使用灵活的 TEE 技术构建可信移动终端的方法得到了更为广泛的研究和支持.移动平台的 TEE 标准化概念[28]最早由 GP(GlobalPlatform)给出,TEE 能够结合移动终端设备的特点,为实际应用提供可行的安全执行环境解决方案.由于使用 ARM 架构处理器的移动设备占据市场的主流地位,当前最为流行的构建移动平台 TEE 方法是利用 ARM TrustZone 提供的基于硬件扩展的安全隔离技术.TI(texas instruments)在文献[29]中介绍了运用 TrustZone 建立移动平台 TEE 的潜在方式.文献[30]提出了一种基于 TrustZone 构建移动终端信任链的方法,该方法可以为移动应用提供可信运行时环境.文献[31]在 TrustZone 构建的 TEE 基础上设计了一套移动平台匿名购物系统.在实际应用中,三星在 TrustZone 技术基础上开发和实现的 KNOX[32]正在为其移动设备提供安全服务,苹果的移动终端产品使用了基于 TrustZone 技术设计的 Apple Pay 安全支付方案,苹果和华为的移动设备指纹识别功能也被认为构建在 TrustZone 的基础之上.虽然目前使用 TrustZone 技术构建移动平台 TEE 及其应用的方法有很多,但他们或多或少都存在一些不足,仍然没有一种被广泛认可的移动 TEE 构建及应用方法.

2 预备知识

2.1 可信计算技术

可信计算存在多种不同的定义,广义的可信计算平台能够保护数据存储区域,避免敌手直接物理访问到机密数据存储区,并保证系统的运行环境是安全的、未被篡改,所有的代码能够运行于一个未被篡改的执行环境内.在 TCG 的可信计算概念中,可信指的是对行为的信任,即,平台实现特定目标的计算行为与预期一致.TCG 提出了通过嵌入在硬件平台上的 TPM 安全芯片来提高计算机系统安全性的技术思路,所构建的可信平台能够证明平台自身的安全属性、保证部分关键计算和数据不受干扰、标识计算平台的身份、对外提供自身行为和环境的证据.可信计算涉及的特征技术包含了信任根提供、完整性度量、敏感数据封装以及可信环境的构建和证明等.以可信虚拟化为核心的可信云计算架构采用了可信计算中的相关技术,在本文所提出的方案中,可信移动终端的构建和云服务安全接入协议的设计也借鉴了可信计算技术的部分思想.

2.2 TrustZone安全技术

作为一种硬件安全扩展技术,TrustZone 自 ARM v6 开始引入 ARM 架构规范,支持用户自主开发、设计特定的安全系统,被大量移动嵌入式设备所应用.该技术提供两个执行环境:安全世界 SW(secure world)和普通世界 NW(normal world).两个世界实现了代码隔离,通过安全监视器(secure monitor)控制两个世界的切换.其中:SW 通常用于实现 TEE,执行定制的安全 OS 以及安全敏感的任务;而 NW 用于实现 REE,可以运行通用 OS 和普通应用程序.SW 与 NW 由底层 ARM 芯片提供基于硬件的物理隔离,两个世界具有独立的系统资源,包括寄存器、物理内存和外设,SW 中的代码和资源受到严格的访问控制策略保护.TrustZone 的一大优势是在通用的 ARM 处理器中实现扩展,通过额外增加的扩展,单个处理器能够以时间片的方式安全有效地执行不同世界的代码,无需使用专用安全处理器即可实现类似的安全功能.此外,TrustZone 在同一时刻只允许运行一个世界的代码,每个世界运行时都可以独占 CPU 资源,这使得 SW 中的安全运算性能要远高于一般资源受限的安全处理器.

2.3 物理不可克隆函数(PUF)

PUF[18]是输入/输出关系,由特殊物理系统决定的一种函数,这里的输入/输出也被认为是一对挑战和响应.PUF 具有随机和不可克隆两个重要属性,其不可克隆性来源于物理设备生产过程中随机引入的不确定因素.PUF 的响应具有一定的噪声,模糊提取器[33]能够帮助 PUF 消减噪声干扰.PUF 的物理优势使其可以隐式存储一

段若干字节的秘密数据,该秘密数据不会显式暴露于外界,通过物理特性提取秘密数据的过程无法被其他设备所克隆.相比于普通的非易失性存储器,PUF 提供了更高的物理安全特性,可以防止秘密数据直接从存储器中被恶意读出.PUF 是一项低成本技术,可以利用当前普通的生产处理工艺快速实现.

事实上,TrustZone 技术仅提供隔离执行环境,唯有配备可靠且可用于证明的信任根,才能真正实现 TEE[18].由于对于支持 TrustZone 的移动设备来说可用的内置设备密钥不是通用必选配置,因而移动终端自身并不总是具备提供信任根的能力.为了弥补这一可能的缺陷,PUF 可以作为一个有效的信任根与 TrustZone 配合使用.本文方案将 PUF 提取的唯一性秘密数据作为根密钥种子,以此派生其他安全密钥,进而实现身份证明和数据保护功能.本文方案采用了以静态随机存储器 SRAM 为物理介质的 PUF[34],该 PUF 以 SRAM 的单元地址访问作为挑战,以 SRAM 加电后短时间内的随机值作为响应.

3 系统模型与假设

3.1 系统模型

在本文提出的可信移动终端云服务安全接入方案的系统模型中,共有 4 个参与实体:移动终端制造商$\mathcal{M}$(manufacturer)、移动终端$\mathcal{T}$(mobile terminal)、应用服务提供商$\mathcal{A}$(application service provider)和云服务提供商$\mathcal{C}$(cloud service provider).在实际应用场景中,$\mathcal{T}$由$\mathcal{M}$生成制造,是移动用户直接操作的实体,配备了支持 TrustZone 安全扩展技术的 ARM 处理器芯片.在$\mathcal{T}$出厂前,$\mathcal{M}$为$\mathcal{T}$颁发设备证书,证书将被嵌入设备中进行存储.$\mathcal{A}$是移动云计算应用的提供商,其可以是云服务账户的管理者,如管理 iCloud 的苹果公司,也可以是具体应用的设计与发布者,如提供云存储应用的 DropBox 公司.$\mathcal{A}$在本系统模型中主要负责识别希望接入云服务的$\mathcal{T}$及其用户的身份,并给予合法$\mathcal{T}$访问云服务的授权.移动应用后台的服务实际由$\mathcal{C}$提供,$\mathcal{C}$可能拥有多个数据中心(data center),具有强大的计算与存储能力.在云计算环境下,$\mathcal{A}$和$\mathcal{C}$可以位于同一公司内部,云计算平台由移动应用提供商自主搭建,移动用户可以体验一体化的云服务;$\mathcal{A}$也可以租用第三方云服务提供商$\mathcal{C}$提供的基础设施和计算能力,将应用服务外包给$\mathcal{C}$后,由$\mathcal{C}$向移动用户提供基于软件应用的云服务.图 1 描绘了本文所提方案的系统模型.

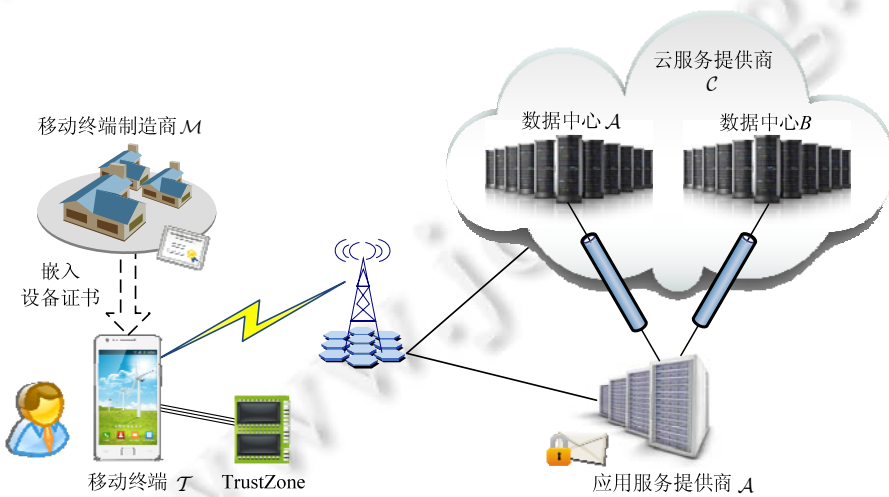


Fig.1 System model of secure access scheme

图 1 安全接入方案的系统模型

3.2 安全假设与敌手模型

在系统模型中,我们假设$\mathcal{A}$与$\mathcal{C}$之间的数据通信建立在诸如 TLS/SSL 的安全传输层协议基础之上,数据的传输受到机密性、完整性和认证性的保护,$\mathcal{C}$的数据中心可以正确获取$\mathcal{A}$发送的相关秘密数据.此外,$\mathcal{T}$在下载由$\mathcal{A}$

提供的移动应用时,也被认为可以从应用中直接提取$\mathcal{A}$的公钥.如何防止$\mathcal{T}$从非法的移动应用市场下载敌手冒充$\mathcal{A}$提供的恶意应用已经超出了本文的研究范围,在此不再赘述.

安全接入方案的运行依赖于应用提供商$\mathcal{A}$对于移动终端制造商$\mathcal{M}$的前提信任,在这里,$\mathcal{A}$信任$\mathcal{M}$只为支持 TrustZone 技术且符合一定安全标准的移动终端$\mathcal{T}$颁发设备证书.这一假设在现实中是合理的,为了维护品牌信誉,$\mathcal{M}$的行为将受到市场的监督和政府的监管.本文方案以移动终端安全为出发点,不考虑$\mathcal{A}$和$\mathcal{C}$自身内部的恶意行为以及可被敌手利用的安全漏洞.

根据上述安全假设,本文提出的安全接入方案可以防御具有以下能力的敌手:

- 敌手攻击方案中的安全接入协议,试图盗用或伪装移动终端和用户的合法身份访问云服务,试图窃取、伪造或篡改实体间的通信数据;
- 敌手对移动终端实施基于软件代码的攻击,破坏运行在 TrustZone NW(REE)中的通用移动 OS 或应用程序,敌手可以访问本方案相关程序(如云服务应用客户端)在 NW 中的接口;
- 敌手对移动终端具有一定的物理接入能力,可以重启设备并直接读取设备中非易失性存储器的数据.

本文方案信任提供 TrustZone 技术的终端硬件安全性,不考虑针对 TrustZone 的恶意物理攻击和针对 PUF 的侧信道攻击.

4 可信移动终端云服务安全接入方案设计

本节首先给出面向云环境的可信移动终端体系结构设计,再介绍密钥与敏感数据管理机制,然后结合体系结构和管理机制详细阐述云服务安全接入协议,最后对本文所提出的方案进行安全性分析.

4.1 可信移动终端体系结构

借助 TrustZone 和 PUF 技术,我们面向云计算场景设计了可信移动终端体系结构.在现有移动终端硬件架构的基础上,我们的可信终端方案以基于软件的设计和实现为主,以低成本性、高灵活性和易扩展性为设计目标.图 2 展示了本文提出的可信移动终端体系结构以及各组件间的交互细节.

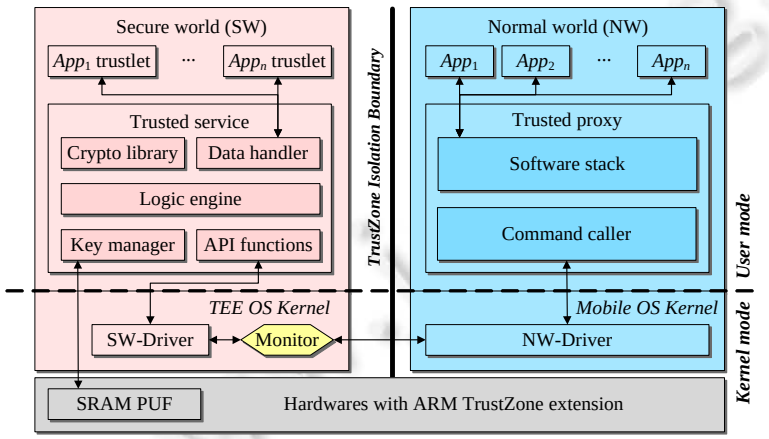


Fig.2 Architecture of trusted mobile terminal for cloud computing

图 2 面向云计算环境的可信移动终端体系结构

利用文献[31]给出的方法,可以在 TrustZone 的 SW 中安全有效地构建 TEE.SW 中实现的 TEE 物理隔离于 NW 中实现的通用移动系统环境,SW 中运行有定制的 TEE OS,用于执行安全敏感的程序代码,NW 中运行有通用的移动 OS,该 OS 可以是 Android 或 iOS 系统,能够执行常规的移动应用程序.下面将详细介绍各组件的功能.

(1) 可信代理(trusted proxy)

可信代理在 NW 中直接与移动应用程序交互.该组件接收移动应用程序的可信服务请求,根据请求类型组装调用 SW 中可信服务组件的命令,为 SW 中的实质性安全运算做准备.该组件包含如下两个子组件:

- 软件栈(software stack):为移动应用程序提供高层可信服务接口,负责解析应用程序请求数据,并返回服务响应结果;
- 命令调用器(command caller):组装可信服务调用命令,与 SW 中的可信服务组件交互,通过规范的 TEE 客户端命令接口(GP TEE client API)[35]实现命令的发送,借助 NW 底层驱动请求 NW 向 SW 的切换并等待数据返回.

(2) 可信服务(trusted service)

可信服务组件是可信移动终端的核心组件,不仅实现了信任根呈现、密钥与敏感数据管理和可信环境证明等可信计算相关功能,还实现了安全计入协议在移动终端的执行逻辑.该组件的代码执行受到 TrustZone 隔离机制的保护,由以下 5 个子组件组成:

- 应用程序接口函数(API functions):接收来自 NW 中可信代理发送的可信服务请求,解析命令数据,将运算指令传递给逻辑引擎并等待结果返回可信代理;
- 密钥管理器(key manager):利用从 SRAM PUF 中提取的根密钥种子生产多种密码学密钥,将密钥提供给数据处理器使用;
- 数据处理器(data handler):为了防止敌手伪造安全参数(通常是用户名和口令),数据处理器只接收来自 SW 中移动应用程序可信单元的参数输入,并将参数传递给逻辑引擎;此外,该子组件还负责敏感数据的封装与解封,封装后的数据可以存储在移动设备的通用非易失性存储器中;
- 密码算法库(crypto library):为密钥管理器、数据处理器和逻辑引擎提供密码学算法支持,其实现有对称和非对称加解密与签名验证算法及多种消息摘要算法;
- 逻辑引擎(logic engine):从其他子组件获取必要的参数输入,依据所设计的安全接入协议逻辑,执行移动终端安全敏感的可信服务运算操作,输出执行结果;此外,该子组件实现了对 SW 中应用程序进程的加载度量和启动管控.

(3) 移动应用程序(App)和移动应用程序可信单元(App trustlet)

当移动用户希望访问$\mathcal{C}$处的云服务时,无论是采用浏览器方式还是客户端方式,都需要启动相应的移动应用.针对可信移动终端,应用服务提供商$\mathcal{A}$提供的移动应用包含两个部分:运行在 NW 中的移动应用程序和运行在 SW 中的移动应用程序可信单元.App 只向用户提供图形用户界面(GUI)和基本的非安全敏感功能,App trustlet 负责收集和预处理接入云服务所需要的敏感数据信息,并将其呈递给可信服务以用于安全接入协议的运算.当 App 需要通过执行安全接入协议访问云服务时,其通过调用可信代理的软件栈进行可信服务请求,在 TrustZone 使用系统中断完成 NW 向 SW 的切换后,可信服务将加载启动 App trustlet,其代码完整性由可信服务的逻辑引擎负责度量.基于我们之前的研究工作[36],使用白名单机制在 SW 中一旦发现 App trustlet 被敌手篡改,可以禁止其启动.当 App trustlet 被正确启动后,用户可以在安全模式下将云服务用户名和口令等敏感数据信息输入 App trustlet,进而交由可信服务进行处理.App 通过 TrustZone 提供的域间通信机制[37]实现与 App trustlet 的业务通信,此处的移动应用程序设计符合当前 TrustZone 的常规应用模式.

(4) 内核中的组件

在 SW 的 TEE 操作系统内核中,有驱动组件 SW-Driver;在 NW 的移动操作系统内核中,有驱动组件 NW-Driver.以上两个驱动组件用于处理在 TrustZone 两个世界切换时的请求与响应命令,其中包含了两个世界的通信数据.作为 TrustZone 定义的安全监视器的实现,监视器(monitor)位于 SW 的系统内核中,其控制底层硬件完成 TrustZone 世界切换的具体动作.除了这些特殊的组件外,NW 中的 OS 内核实现有各种通用的硬件驱动,这其中包含有网络通信驱动,可信移动终端与云服务的数据通信均依赖于该驱动.

(5) 硬件中的组件

可信移动终端设备的硬件支持 ARM TrustZone 扩展技术,受到该技术的保护,位于硬件中的 SRAM PUF 物

理组件仅能被 SW 访问,PUF 的软件算法由可信服务的密钥管理器实现.

4.2 密钥与敏感数据管理

在详细阐述本文提出的云服务安全接入协议之前,首先介绍由 SRAM PUF 提取根密钥种子以及使用该种子派生多种不同用途密钥的方法;此外,使用派生密钥保护敏感数据的机制也将在本小节给出.

4.2.1 根密钥种子的提取

本文使用文献[18]提出的 SRAM PUF 技术提取根密钥种子 s,s 是移动终端制造商$\mathcal{M}$在设备生产过程中随机选取的一段具有唯一性的比特串,$\mathcal{M}$利用移动终端$\mathcal{T}$中 SRAM 特定区域的物理特性将 s 存储在其中.s 仅在$\mathcal{T}$每次正常加电启动时被从 SRAM PUF 组件中重现出来,并被 SW 中的密钥管理器安全缓存.s 的机密性受到 TrustZone 的严格保护.

4.2.2 密钥派生

在移动终端 SW 中,可信服务的密钥管理器具有密钥生成函数 KDF(key derivation function),该函数是一种确定性的映射:$\tilde{\mathcal{S}}\times\tilde{\mathcal{P}}\to\tilde{\mathcal{K}}$,其中,$\tilde{\mathcal{S}}$ 是密钥种子空间,$\tilde{\mathcal{P}}$ 是声明密钥用途的字符串参数集合,$\tilde{\mathcal{K}}$ 是生成密钥的空间.使用 KDF 和根密钥种子 s,可以生成唯一标识移动终端身份的设备密钥公私钥对 (dpk_T, dsk_T),生成方式为

$$(dpk_T, dsk_T) \leftarrow KDF_s(\text{“}identity\text{”}).$$

类似地,可以生成存储根密钥 srk,生成方式为 $srk\leftarrow KDF_s(\text{“}storage\_root\text{”})$.$srk$ 用于进一步生成存储保护实际敏感数据的存储密钥,该套存储密钥层次结构增强了密钥使用的隔离性和安全性.值得强调的一点是:这里生成的所有存储密钥和设备密钥的私钥从不离开 SW,也不在移动设备的非易失性存储器上存储,如果需要使用,它们将被使用 KDF 以同样的方式重构,这样可以减小密钥丢失的风险.

4.2.3 敏感数据管理

我们使用由 srk 派生的多种存储密钥对应用服务提供商$\mathcal{A}$的公钥 apk 和云服务会话所需的密钥包(ID,k^{enc},k^{mac},n_i)进行封装存储保护,这些密钥数据的具体含义及用途将在第 4.3 节中详细介绍.封装操作在 SW 可信服务的数据处理器中进行,数据处理器实现了数据封装函数 $Data\_Seal()$,封装后的数据块可以存储在设备的公共非易失性存储器中.在本文中,我们用 $MAC_k(m)$表示使用密钥 k 对数据 m 计算消息认证码;$Enc_k(m)$表示使用密钥 k 对数据 m 进行加密, 根据 k 的类型可以相应表示对称与非对称加密;$Sign_k(m)$表示签名操作;$\|$表示数据的连接操作.以下给出具体的封装方法:

- 对于公钥 apk,封装时只需保护其完整性,防止移动应用被恶意篡改后造成的公钥破坏,步骤为

$$mk_{apk}\leftarrow KDF_{srk}(\text{“}storage\_key\text{”},\text{“}MAC\text{”},apk),$$
$$blob_{apk}\leftarrow Data\_Seal(\text{“}MAC\text{”},mk_{apk},apk);$$

其中,$blob_{apk} := apk \,\|\, MAC_{mk_{apk}}(apk)$;

- 对于密钥包(ID,k^{enc},k^{mac},n_i),封装时需要保护其机密性和完整性,防止敌手的窃取或篡改,步骤为

$$(sk_{ID},mk_{ID})\leftarrow KDF_{srk}(\text{“}storage\_key\text{”},\text{“}Enc+MAC\text{”},ID),$$
$$Blob_{ID}\leftarrow Data\_Seal(\text{“}Enc+MAC\text{”},sk_{ID},mk_{ID},(ID,k^{enc},k^{mac},n_i));$$

其中,sk_{ID} 为对称密钥,$blob_{ID} := Enc_{sk_{ID}}(ID,k^{enc},k^{mac},n_i)\,\|\,MAC_{mk_{ID}}(Enc_{sk_{ID}}(ID,k^{enc},k^{mac},n_i))$.

利用密钥管理器中重构出的存储密钥,数据处理器可以调用 $Data_Unseal()$函数从相应的数据块中恢复并验证敏感数据.

4.3 云服务安全接入协议

云服务安全接入协议的交互参与实体有$\mathcal{T}$,$\mathcal{A}$和$\mathcal{C}$,在正常情况下,协议执行的概述如图 3 所示.在某段时间内,当$\mathcal{T}$首次访问位于$\mathcal{C}$处的云服务时,其首先向$\mathcal{A}$发送接入服务的授权请求;经过计费和合法性认证后,$\mathcal{A}$生成会话密钥包,将其颁发给$\mathcal{T}$;同时,$\mathcal{A}$通过安全信道将已认证的用户数据发送给$\mathcal{C}$,在这里,我们对$\mathcal{C}$中的用户管理主机和服务运算主机不作区分;当获得授权的会话密钥包后,$\mathcal{T}$使用相关密钥和认证信息向$\mathcal{C}$发送服务接入请求;$\mathcal{C}$对请求进行验证后,将验证结果返回$\mathcal{T}$;完成接入认证后,$\mathcal{T}$与$\mathcal{C}$展开正常的云服务交互.我们给出的安全接入协议可以

与可信云架构进行对接,实现$\mathcal{T}$与$\mathcal{C}$间安全执行环境的双向认证,本文假设$\mathcal{C}$采用文献[13]提出的可信云架构,$\mathcal{C}$在返回$\mathcal{T}$服务接入请求验证结果时,将附带云端主机云服务程序的完整性度量值.

在安全接入协议执行前,需要完成一些准备工作.$\mathcal{T}$出厂之前,$\mathcal{M}$选取良好的根密钥种子 s 存入$\mathcal{T}$的 SRAM PUF 中,并引导$\mathcal{T}$在 SW 中生成唯一标识身份的设备密钥对 (dpk_T, dsk_T).$\mathcal{M}$为其公钥 dpk_T 颁发设备证书 $Cert_T$,该证书能够证明$\mathcal{T}$已通过$\mathcal{M}$的安全与合法性测试和认证.此外,$Cert_T$ 中还可包含设备$\mathcal{T}$的若干配置信息,例如芯片类型、芯片版本号和 TrustZone 是否可用等.

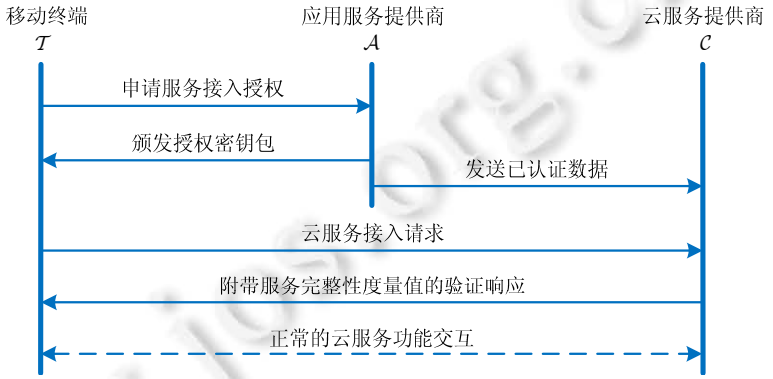


Fig.3 An overview of secure access protocol under normal condition

图 3 安全接入协议在正常情况下的执行概述

以可信移动终端为核心的安全接入协议由授权申请、接入请求、验证响应和授权撤销 4 个部分组成,其中,授权申请只在 3 种情况下被执行:(1) 用户首次使用$\mathcal{T}$请求接入云服务;(2) 上一次授权申请已过期;(3) 因网络错误或恶意攻击造成授权被撤销.在成功执行授权申请后,用户使用$\mathcal{T}$可以在一定时间内多次请求接入云服务,协议的接入请求和验证响应可以随之多次被执行.

4.3.1 授权申请

在本部分协议中,用户使用$\mathcal{T}$向$\mathcal{A}$发送云服务接入授权的申请,$\mathcal{A}$验证申请中的相关参数,在确定$\mathcal{T}$及其用户的合法性后,生成用于$\mathcal{T}$与$\mathcal{C}$未来进行会话的密钥包,将其分别发送给双方,具体步骤如下:

(1) 用户在 NW 中操作 App 请求访问云服务,TrustZone 切换至 SW,$\mathcal{T}$调用 *KDF* 生成消息完整性保护密钥 mk_{auth},该密钥用于$\mathcal{A}$向$\mathcal{T}$发送会话密钥包时保护数据通信的完整性,密钥生成方法为

$$mk_{auth} \leftarrow KDF_s(\text{“}session\_key\text{”}, \text{“}MAC\text{”}, r),$$

其中,r 是密钥管理器生成的随机数,用于生成不同的 mk_{auth}.

(2) $\mathcal{T}$在 SW 中加载启动 App trustlet 时,对加载的代码进行完整性度量,利用哈希函数得到度量值$\mu(app)$.基于我们的白名单机制[37],可以发现 App trustlet 的篡改,如果被篡改,协议将终止执行.

(3) 用户向 SW 的 App trustlet 输入登陆云服务的用户名 *user* 和口令 *pswd*,App trustlet 计算口令哈希值 $H(pswd)$后,连同用户名一起发送给可信服务的数据处理器,可信服务将 App trustlet 关闭;

(4) SW 中可信服务的逻辑引擎调用授权申请 API:*Apply*(),生成授权申请消息 *m_apply*:

$$m\_apply \leftarrow Apply(Cert_T, dsk_T, blob_{apk}, mk_{auth}, \mu(app), user, H(pswd)).$$

该 API 具体执行操作如下:

1) 调用密钥管理器重构设备密钥私钥 dsk_T;

2) 调用密钥管理器重构 *apk* 的存储保护密钥,调用数据处理器解封 $blob_{apk}$,得到正确的 *apk*;

3) 调用签名函数生成签名: $\omega := Sign_{dsk_T}(mk_{auth}, \mu(app), user, H(pswd))$;

4) 调用加密函数生成最终的通信消息: $m\_apply := Enc_{apk}(Cert_T, mk_{auth}, \mu(app), user, H(pswd), \omega)$.

这里的 *apk* 来源于$\mathcal{A}$,实际上,$\mathcal{A}$为发行的每一种应用生成一对公私钥对(*apk*,*ask*),在应用程序安装时,*apk* 可以被$\mathcal{T}$提取用于与$\mathcal{A}$的认证通信;此外,*apk* 可以唯一标识一个应用程序.

(5) $\mathcal{T}$从 SW 切换至 NW,向$\mathcal{A}$发送 *m_apply*,$\mathcal{A}$使用自己掌握的私钥 *ask* 解密消息,使用权威发布的$\mathcal{M}$公钥证设备证书 $Cert_{\mathcal{T}}$,并从中获得$\mathcal{T}$的设备密钥公钥 $dpk_{\mathcal{T}}$,验证相关数据的签名,提取授权申请消息的有效数据元组:

$$(mk_{auth},\mu(app),user,H(pswd)).$$

(6) $\mathcal{A}$根据自己发布的应用程序代码验证$\mathcal{T}$中 App trustlet 的完整性度量值$\mu(app)$,使用 *user* 和 $H(pswd)$验证用户账户的合法性,可以对账户的余额进行检查:若相关验证失败,返回$\mathcal{T}$申请失败的消息及原因;若所有验证通过,$\mathcal{A}$为$\mathcal{T}$和$\mathcal{C}$生成会话密钥包元组(ID,k^{enc},k^{mac},n_0),其中:*ID* 唯一标识该密钥包;k^{enc}用于保护会话的机密性;k^{mac}用于保护会话的完整性;n_0 是随机选取的 nonce 值,用于防止重放攻击.$\mathcal{T}$与$\mathcal{C}$每经过一次正确的服务访问连接,各自将该值加 1,第 i+1 次连接,使用的即为 n_i;

(7) $\mathcal{A}$为$\mathcal{T}$颁发授权的会话密钥包,$\mathcal{A}$将会话密码包签名后与相关信息一起使用 $dpk_{\mathcal{T}}$ 加密生成σ:

$$\sigma := Enc_{dpk_{\mathcal{T}}}(apk,(ID,k^{enc},k^{mac},n_0),Sign_{ask}(ID,k^{enc},k^{mac},n_0)),$$

其中,*apk* 用于标识该加密数据对应的应用程序.最终,$\mathcal{A}$以如下方式生成授权响应消息:

$$m\_reply := \sigma \| MAC_{mk_{auth}}(\sigma).$$

(8) $\mathcal{A}$将 *m_reply* 发送给$\mathcal{T}$后,通过安全信道将(ID,k^{enc},k^{mac},n_0)连同 *user* 和$\mu(app)$一起发送给$\mathcal{C}$,$\mathcal{C}$如果在数据库中通过 *user* 查找到同一用户之前的会话密钥包,则将旧的密钥包删除.会话密钥包的有效期可以根据云服务的安全敏感度设置不同的长度,可以是 1 天、7 天或 30 天.该有效期记录在$\mathcal{C}$处,如果超过有效期,会话密钥包将自动失效,失效的会话密钥包由$\mathcal{C}$定期清理删除;

(9) $\mathcal{T}$接收 *m_reply* 后切换至 SW,可信服务对 *m_reply* 进行解析,通过验证消息完整性和签名正确性后,将提取的会话密钥包(ID,k^{enc},k^{mac},n_0)封装为 $blob_{ID}$,存储于移动设备中.

4.3.2 接入请求

在本部分协议中,$\mathcal{T}$利用会话密钥包向$\mathcal{C}$发送云服务接入请求,具体步骤如下:

(1) $\mathcal{T}$重新加载启动 App trustlet,对加载的代码再一次进行完整性度量,利用哈希函数得到度量值$\mu'(app)$,可以使用白名单机制再次检查 App trustlet 是否被篡改;

(2) SW 中可信服务的逻辑引擎调用云服务接入请求 API:*Request*(),生成 *m_request*:

$$m\_request \leftarrow Request(blob_{ID},\mu'(app)).$$

该 API 具体执行操作如下:

1) 调用密钥管理器重构会话密钥包的存储保护密钥,调用数据处理器解封 $blob_{ID}$,得到正确会话密钥包元组(ID,k^{enc},k^{mac},n_i);

2) 生成云服务接入请求通信消息 *m_request*:

$$m\_request := ID \| Enc_{k^{enc}}(\text{“}request\text{”}, n_i, \mu'(app)) \| MAC_{k^{mac}}(ID, Enc_{k^{enc}}(\text{“}request\text{”}, n_i, \mu'(app))),$$

其中,*ID* 用于告知$\mathcal{C}$使用哪个会话密钥包解密和验证消息;*request* 是标识$\mathcal{T}$请求$\mathcal{C}$处云服务执行的命令参数,这里表示申请接入云服务的命令.

(3) $\mathcal{T}$切换至 NW,将 *m_request* 发送给$\mathcal{C}$.

4.3.3 验证响应

在本部分协议中,$\mathcal{C}$利用会话密钥包解析来自$\mathcal{T}$的接入请求,并将验证结果与云服务执行程序的完整性度量值返回给$\mathcal{T}$,具体步骤如下:

(1) $\mathcal{C}$接收到 *m_request* 后,根据其中的 *ID* 查找到数据库中对应的会话密钥包,检查会话密钥包是否仍然有效、过期或已被撤销,对于未能查找到密钥包或密钥包失效的情况,$\mathcal{C}$发送标记为验证失败的响应给$\mathcal{T}$,$\mathcal{T}$将重新执行授权申请协议.

(2) $\mathcal{C}$使用合法密钥包解析 *m_request* 后,将其中的 n_i 与数据库中会话密钥包记录的当前 nonce 值比对,如果

不相同,返回验证失败的响应给$\mathcal{T}$,$\mathcal{T}$将重新执行授权申请协议.

(3) $\mathcal{C}$将 *m_request* 中的$\mu'(app)$与数据库中关联相应会话密钥包的原始$\mu(app)$比对,如果不相同,说明$\mathcal{T}$中的App trustlet 很有可能已被篡改,$\mathcal{C}$将拒绝$\mathcal{T}$的接入请求,返回验证失败的响应.

(4) 通过上述 3 步验证后,$\mathcal{C}$使用可信云架构中的安全方法生成虚拟机中具体运行$\mathcal{T}$所请求云服务的程序的完整性度量值$\mu(csp)$,在一些特定的应用场景中,该度量值可能来源于对整个虚拟机镜像的哈希度量,对于$\mu(csp)$的证明方法可参见可信云架构的具体协议.

(5) $\mathcal{C}$生成验证响应的通信消息 *m_response*:

$$m\_response := ID \| Enc_{k^{enc}}(\text{“}response\text{”},\text{“}passed\text{”}\ n_i, apk, \mu(csp)),$$
$$\| MAC_{k^{mac}}(ID, Enc_{k^{enc}}(\text{“}response\text{”},\text{“}passed\text{”}\ n_i, apk, \mu(csp))),$$

其中,“*passed*”表示验证通过,公钥 *apk* 标识了消息针对的移动云应用.

(6) $\mathcal{C}$将 *m_response* 发送给$\mathcal{T}$后,更新 nonce 值:$n_{i+1}=n_i+1$,并设置此后接入服务次数的 nonce 限定值 $limit\_n=n_i+j$,j 代表此次验证后允许$\mathcal{T}$接入云服务发送操作命令的次数,若在第 x 次接入时,$\mathcal{C}$的 $n_{i+x}>limit\_n$,$\mathcal{T}$需要重新执行接入请求协议,使$\mathcal{C}$设置更新 *limit_n*;

(7) $\mathcal{T}$收到 *m_response* 后切换至 SW 对其进行解析和验证,同时更新自己的会话密钥包 nonce 值:$n_{i+1}=n_i+1$,配合可信云架构的安全方法可以通过$\mu(csp)$验证云端服务程序的完整性.待验证通过,App trustlet 根据用户请求云服务的具体功能向可信服务传送命令参数,可信服务使用会话密钥包组装云服务操作命令,与$\mathcal{C}$进行交互通信以完成具体的云服务功能.本文第 5 节将给出一种基于本协议的移动云存储应用实例.

4.3.4 授权撤销

当$\mathcal{T}$的会话密钥包尚在有效期内但遇到以下 4 种情况时,$\mathcal{T}$原有的合法授权将被撤销:

- $\mathcal{C}$发现$\mathcal{T}$当前发送的某个通信消息中的 n_i 与云端存储的当前 n_i 不一致;
- $\mathcal{A}$发现发布的移动应用存在漏洞;
- $\mathcal{A}$发布移动应用更新;
- $\mathcal{A}$或$\mathcal{C}$根据网络监控与分析推断会话密钥包信息可能已被泄露.

在撤销协议被触发后,$\mathcal{C}$自身或由$\mathcal{A}$通知$\mathcal{C}$将$\mathcal{C}$存储的相应会话密钥包信息删除,$\mathcal{T}$只有重新执行授权申请协议才可能再次正常访问云服务.

4.4 安全性分析

在第 3.2 节中给出的安全假设下,本文提出的可信移动终端云服务安全接入方案具有数据机密性与完整性、用户身份不可伪造性、授权不可伪造性、应用程序可证明性、授权可撤销性和设备丢失保护性 6 种安全属性.假设密码学原语(如加密解密、签名验证和消息认证码等操作)的实现是安全的,以下给出对本方案的非形式化安全分析.

(1) 数据机密性与完整性

数据机密性与完整性一方面指的是可信移动终端对用户数据和接收到的敏感数据提供本地的机密性和完整性保护,另一方面指的是$\mathcal{T}$和$\mathcal{A}$之间以及$\mathcal{T}$和$\mathcal{C}$之间数据通信受到机密性和完整性的保护.

- 首先,用户的核心数据,即用户名和口令,只被允许输入到$\mathcal{T}$的 SW 中,这由 App trustlet 的安全性和 TrustZone 的隔离机制提供保护,而 App trustlet 的安全加载受到 SW 中可信服务的监控,因此,用户口令难以以明文的形式流入$\mathcal{T}$的 NW 中,这减小了其受到恶意代码攻击的可能;类似地,在接入云服务后,用户的私有信息均通过 App trustlet 输入,经过 SW 中的密码学处理后受保护地发送给云端.可信移动终端对于提取到的移动应用公钥和接收到的会话密钥包进行封装存储,封装过程在 SW 中完成,封装使用的密钥不会泄露到 SW 外,因此上述数据在可信移动终端通用非易失性存储器上的存储是安全的;
- 其次,即使敌手攻破了移动终端的 NW,也无法通过窥探 SW 可信服务的接口获取有价值的数据.这是因为流入 NW 的敏感数据都是经过 SW 加密或封装的;

- 最后,$\mathcal{T}$和$\mathcal{A}$之间以及$\mathcal{T}$和$\mathcal{C}$之间的数据通信受到方案的安全接入协议保护,其中:$\mathcal{T}$向$\mathcal{A}$发送的授权申请消息 *m_apply* 机密性受到应用公钥 *apk* 的保护,完整性间接由设备证书 $Cert_{\mathcal{T}}$ 及证书中的相关信息提供保护;$\mathcal{A}$向$\mathcal{T}$返回的的授权响应消息 *m_reply* 机密性受到设备密钥公钥 $dpk_{\mathcal{T}}$ 保护,完整性受到密钥 mk_{auth} 保护;授权申请过程中,敌手无法通过转发 *m_apply* 实现重放攻击,这源于他没有$\mathcal{T}$的设备密钥私钥,进而也无法从获得的 *m_reply* 中解密出合法的会话密钥包;$\mathcal{T}$与$\mathcal{C}$之间的所有通信消息(包括 *m_request* 和 *m_response*)的机密性和完整均受到密钥 k^{enc} 和 k^{mac} 的保护,消息新鲜性由 n_i 给予保护.

(2) 用户身份不可伪造性

用户身份不可伪造性指的是:在没有获得某用户 *U* 的用户名和口令的情况下,敌手无法伪造 *U* 的合法身份并冒名接入 *U* 的云服务账户.由上一个安全属性可知:敌手无法通过攻击本方案获取 *U* 的口令,也无法直接截获由$\mathcal{A}$颁发给 *U* 的会话密钥包;在没有 *U* 的口令情况下,即使敌手使用一个合法的移动设备向$\mathcal{A}$发送授权申请,他也无法通过验证并获得 *U* 与$\mathcal{C}$的会话密钥包.缺少 *U* 的会话密钥包,敌手无法接入 *U* 的云服务,也无法访问其云端的私有数据.

(3) 授权不可伪造性

授权不可伪造性指的是:敌手无法伪造$\mathcal{A}$的身份颁发合法的会话密钥包,进而利用密钥包访问用户 *U* 的云端私有数据.根据假设,$\mathcal{T}$可以安全获取$\mathcal{A}$为移动应用生成的公钥 *apk*,$\mathcal{T}$使用 *apk* 加密相关数据后生成授权申请消息 *m_apply*,因为缺乏私钥 *ask*,敌手无法解密 *m_apply*,也就无法生成具有$\mathcal{A}$签名的且由 mk_{auth} 保护完整性的授权响应消息 *m_reply*,这使得敌手无法向$\mathcal{T}$发送伪造的会话密钥包.类似地,$\mathcal{A}$与$\mathcal{C}$的通信受到安全信道的保护,敌手无法将伪造的会话密钥包成功地发送给$\mathcal{C}$.在无法伪造会话密码包的情况下,敌手无法介入$\mathcal{T}$与$\mathcal{C}$的数据通信.

(4) 应用程序可证明性

应用程序可证明性指的是:移动终端中,执行敏感数据操作的 App trustlet 和云服务端提供服务的程序代码可被度量和报告,实现完整性的证明.可证明性的概念来源于可信计算技术,在移动终端,可信服务对 App trustlet 进行度量得到$\mu(app)$,在授权申请时,移动终端对$\mu(app)$签名后发送给$\mathcal{C}$,$\mathcal{C}$通过验证签名和比对$\mu(app)$可以确定移动终端$\mathcal{T}$运行的 App trustlet 没有被篡改.反过来,本方案接入可信云架构时,可以通过比对$\mu(csp)$来判断云服务主机上运行的云服务程序是否被篡改.只有完成双向的完整性验证,用户才能放心的通过$\mathcal{T}$接入$\mathcal{C}$,$\mathcal{C}$才能安全接收$\mathcal{T}$的消息,减少恶意代码入侵的几率.

(5) 授权可撤销性

授权可撤销性指的是方案支持特殊情况下对合法授权的紧急撤销,这样可以尽可能地减少用户的损失.根据安全接入协议的撤销部分功能,$\mathcal{A}$与$\mathcal{C}$一旦发现网络通信可能被攻击、移动应用存在漏洞、用户版本老旧或会话密钥包可能被泄露,将触发撤销协议.本方案的撤销协议既可以作为一种主动防御措施,又可以作为一种紧急补救措施,提升方案整体的可控性和安全性.

(6) 设备丢失保护性

设备丢失保护性指的是:当用户发现自己的移动终端设备丢失且之前正确接入过云服务时,可以通过一定的主动行为阻止丢失设备再次接入自己的云服务账户,保护自己的隐私数据.在本方案中,用户只要使用另一台合法的移动终端设备和自己的用户名与口令再次尝试接入一次自己的云服务即可.当授权申请被重新执行后,新生成的会话密码包连同 *user* 被发送到$\mathcal{C}$,$\mathcal{C}$如果在数据库中通过 *user* 查找到之前旧的会话密钥包,则将其删除以替换新的密钥包,敌手无法使用丢失设备继续访问失主的云服务账户.

5 移动云存储应用实例

基于本文提出的可信移动终端云服务安全接入方案,我们设计了一个移动云存储应用实例 MCFile.MCFile 的安全目标主要包括:(1) 保护移动终端用户向云服务器发送和接收文件的机密性和完整性;(2) 云服务端能够对移动终端用户的接入请求和文件访问采取强制的安全认证和访问控制策略.仿照当前市场上的云存储应用,MCFile 实现的云存储服务操作命令可以有:创建文件(create)、删除文件(delete)、写文件(write)、读文件(read)、

添加权限(addright)和取消权限(removeright).

假设有两个用户,其用户名分别为 *user*1 和 *user*2,*user*1 在云端存储有名为 $\mathscr{F}_{user1}$ 的文件,那么在 *user*1 使用移动终端$\mathcal{T}$成功执行完云服务安全接入协议的验证响应后,可以在 SW 中生成如下云服务请求命令读取云端文件 $\mathscr{F}_{user1}$:

$$ID \| Enc_{k^{enc}}(\text{“}read\text{”}, \mathscr{F}_{user1}, n_{i+1}) \| MAC_{k^{mac}}(ID, Enc_{k^{enc}}(\text{“}read\text{”}, \mathscr{F}_{user1}, n_{i+1})).$$

$\mathcal{C}$收到命令后,根据 *ID* 查找相关联的会话密钥包和用户名 *user*1,在解析请求命令并验证命令的合法性后,$\mathcal{C}$检查 *user*1 对于文件 $\mathscr{F}_{user1}$ 拥有的权限,如果判断为可读,则返回以下服务响应:

$$ID \| Enc_{k^{enc}}(\text{“}Res\_read\text{”},\text{“true”}, n_{i+1}, apk, File(\mathscr{F}_{user1})) \| MAC_{k^{mac}}(ID, Enc_{k^{enc}}(\text{“}Res\_read\text{”},\text{“true”}, n_{i+1}, apk, File(\mathscr{F}_{user1}))).$$

其中 $File(\mathscr{F}_{user1})$表示 $\mathscr{F}_{user1}$ 指明的文件实体,使用数据分块技术可以实现大体积文件的网络加密传输.

如果 *user*1 希望将文件 $\mathscr{F}_{user1}$ 分享给 *user*2 阅读,其可以将 $\mathscr{F}_{user1}$ 的可读权限添加给 *user*2,相应的云服务请求命令如下:

$$ID \| Enc_{k^{enc}}(\text{“}addright\text{”},user2,\text{“}read\text{”}, \mathscr{F}_{user1}, n_{i+1}) \| MAC_{k^{mac}}(ID, Enc_{k^{enc}}(\text{“}addright\text{”},user2,\text{“}read\text{”}, \mathscr{F}_{user1}, n_{i+1})).$$

$\mathcal{C}$收到命令经过一些列验证后,在文件 $\mathscr{F}_{user1}$ 的权限列表中增加 *user*2 可读的条目,但此时 $\mathscr{F}_{user1}$ 的拥有者仍是 *user*1,*user*1 可以发送命令取消 *user*2 对 $\mathscr{F}_{user1}$ 的可读权限.此外,当在上述请求命令中使用符号*代替 *user*2 时,*user*1 将 $\mathscr{F}_{user1}$ 可读的权限赋予给所有合法用户,即实现了文件的公开分享.MCFile 其他云存储服务功能的实现方式可以以此类推,在此不再过多叙述.

6 实 现

我们实现了本文方案的原型系统,第 7 节将给出基于该原型系统的方案评估,以下先从硬件平台和软件实现两个方面详细介绍原型系统的实现方法.

6.1 硬件平台

我们分别模拟实现了移动终端$\mathcal{T}$、应用服务提供商$\mathcal{A}$和云服务提供商$\mathcal{C}$.对于移动终端设备的仿真实现,我们采用了嵌入式开发板 Zynq-7000 AP Soc Evaluation Kit[38].该开发板支持 TrustZone 安全扩展,配备有 ARM Cortex-A9 MPCore 处理器、1GB DDR3 内存以及包括 256KB SRAM 在内的 OCM(on-chip memory)模块.由于该开发板加电后,SRAM 会立刻被引导只读存储器 BootROM 初始化,我们无法直接读取 SRAM 的初始数据,因此,我们使用一块 SRAM 芯片作为我们的 SRAM PUF,该芯片型号为 IS61LV6416-10TL[39].SRAM 芯片的初始数据由通用异步收发器 UART(universal asynchronous receiver/transmitter)传送到 Zynq 开发板上,我们使用 Verilog 硬件描述语言以 FPGA 的方式实现了 UART.在 Zynq 开发板上,UART 接收器将 SRAM 初始数据存储在 RAM 缓存中,开发板处理器可以通过总线从 RAM 缓存中读取 SRAM 初始数据.在安全世界 SW 中,我们使用了 Open Virtualization SirerrTEE 作为 TEE OS,该 OS 与 GP 的 TEE 规范[28]相兼容.在普通世界 NW 中,我们使用内核版本为 3.8.6 的 Linux 系统作为 REE OS,该系统是 SirerrTEE 项目中提供的具有 NW-Driver 和 GP TEE Client API 的 Linux.

对于应用服务提供商的仿真实现,我们使用了一台戴尔 OptiPlex 990 台式计算机,该计算机配备 3.3GHz Intel i3-2120 双核处理器和 4GB 内存,运行有内核版本为 Linux 2.6.32 的 Ubuntu10.04 操作系统.对于云服务提供商的仿真实现,我们使用了一台联想 ThinkCentre M8500t 台式计算机,配备有 3.4GHz Intel i7-4770 四核处理器和 8GB 内存,操作系统与前者相同.

6.2 软件实现

在仿真移动终端的嵌入式开发板上,以 C 语言为主的软件实现主要分两个部分:在 SW 中,我们实现了可信服务,其中的密钥管理器包含有 PUF 模糊提取器,具备使用 SRAM PUF 的能力,模糊提取器使用开源的 BCH 代码[40]实现;对于密码算法库,我们利用 OpenSSL-0.9.8y 实现了所需的密码学算法.在具体实现过程中:我们使用密钥长度为 2048 bit 的 RSA 作为非对称加解密与签名验证算法;128 bit 安全强度的 AES 作为对称加解密算法;

SHA256 作为 MAC 和代码完整性度量算法,MAC 实现时将 128 bit MAC 密钥填充为 SHA256 的前 128 bit 进行计算.在 NW 中,我们在通用 OS 上实现了相应的可信代理.此外,具备最基本功能的 App 和 App trustlet 也被实现,App trustlet 可以模拟用户输入用户名和口令.

在两个台式计算机上,我们使用 C 语言分别实现了安全接入方案中实体$\mathcal{A}$和实体$\mathcal{C}$所需执行的安全接入协议的程序代码,配套算法同样采用 OpenSSL-0.9.8y 实现.对于网络通信与交互,两台计算机均以服务器的形式实现,提供 C 语言实现的 socket 连接,支持并发的连接请求与数据处理.

7 评 估

7.1 代码量与可信计算基

在方案的原型系统中,实现各组件的 C 程序代码近似行数(lines of code,简称 LoC)见表 1.一台设备的可信计算基(trusted computing base,简称 TCB)是实现设备安全所需的软件、硬件和固件集合,其规模越小越不容易产生可被敌手利用的漏洞,安全性也相对更容易被保证.本文方案中,可信移动终端的 TCB 仅包含移动设备硬件和运行在 SW 中的软件,根据文献[41]中所述,目前被投入移动商业市场的某型 SW 安全 OS 具有 6000 LoC,如果采用该型 OS,加上我们实现的可信服务和 App trustlet,本文方案的 TCB 软件部分仅有约 9100 LoC.这一规模是相对较小的,系统安全的可控性相对较高.

Table 1 Code sizes and TCB of implemented components

表 1 实体组件代码量与 TCB

实体	组件	代码量(LoC)	TCB
移动终端$\mathcal{T}$	Trusted service	2 300	✔
	Trusted proxy	1 500	✘
	App trustlet	800	✔
	App	500	✘
应用服务提供商$\mathcal{A}$	授权程序	5 600	–
云服务提供商$\mathcal{C}$	接入认证程序	6 300	–

7.2 性能评估

7.2.1 移动终端方案性能评估

使用原型系统,我们对移动终端$\mathcal{T}$在执行本文方案时所需要的相关操作进行了实验,操作包括封装与解封敏感数据以及生成与解析授权申请、接入请求和云服务请求过程中的通信交互消息,其中,云服务请求与响应消息不考虑具体的云服务命令.各操作时间开销统计结果取 100 次运行的平均值,实验结果见表 2.

Table 2 Time overheads of the operations on mobile terminal

表 2 移动终端相关操作执行的时间开销

操作	时间消耗(ms)
封装 *apk*	0.030
解封 *apk*	0.020
封装会话密钥包	0.081
解封会话密钥包	0.093
生成 *m_apply*	129.906
解析 *m_reply*	128.738
生成 *m_request*	0.117
解析 *m_response*	0.110
生成云服务接入请求	0.152
解析云服务验证响应	0.150

从实验结果中可以看出:主要采用对称加解密和消息摘要算法的数据封装与解封操作、接入请求生成与与验证响应解析和云服务请求生成与云服务响应解析的时间开销均不超过 0.16ms,这些操作在移动终端执行本

方案时被较为频繁地使用.相对应的,由于使用了非对称加解密和签名验证算法,授权申请的生成和授权响应的解析操作耗时在 130ms 左右.这一时间开销造成的移动用户等待时延是完全可以接受的,而且该操作仅在用户首次使用移动设备等 3 种情况下执行(见第 4.3 节),使用频率相对较低.本实验基于资源受限的嵌入式开发板完成,如果使用目前市场主流的移动终端设备,运算性能会有较大幅度的提高,方案的时间开销会进一步下降.因此,根据实验结果可知:本文方案在移动终端具有良好的运行性能,各操作引起的等待时延几乎不会给用户带来困扰.

7.2.2 服务端方案性能评估

使用原型系统,我们对应用服务提供商$\mathcal{A}$和云服务提供商$\mathcal{C}$在执行本文方案时所需要的相关操作进行了实验.$\mathcal{A}$在方案中主要负责接收和解析移动终端发送的授权申请消息,并对其进行验证后生成授权响应消息.实验时,我们将这一过程作为一次响应.首先,我们实验了$\mathcal{A}$单线程完成一次响应的时间开销,耗时取独立 100 次运行的平均值,实验结果为单线程单次响应耗时 13.225ms.随后,我们实验了$\mathcal{A}$接收大量授权申请时在使用线程池并发执行的情况下完成单条响应所需要的时间,实验结果如图 4 所示.从图中可以看出:随着并发请求数量从 100 个上升到 500 个,单条请求的响应时间从约 400ms 增长到约 2 300ms.这一较大幅度的增长源于$\mathcal{A}$在一次响应过程中需要执行非对称加密、解密和签名、验证各一次,这些运算较为消耗系统资源.我们的实验基于通用台式计算机完成,考虑到实际应用中$\mathcal{A}$通常由若干专业服务器集群实现,经过优化的并发响应速度将会有很大的提升空间.此外,移动用户执行授权申请的频率不高,相比于移动网络延时,2 000ms 左右的服务器响应延时也是可以被接受的.

$\mathcal{C}$在方案中主要负责接收和解析移动终端发送的接入请求消息,并对其进行验证后生成验证响应消息.实验时,我们将这一过程作为一次响应.同样地,我们首先实验了$\mathcal{C}$单线程完成一次响应的执行情况,实验结果为单线程单次响应耗时 0.016ms.随后,我们实验了$\mathcal{C}$在并发执行的情况下完成单条响应的时间,实验结果如图 5 所示.从图中可以看出:随着并发请求数量的上升,单条请求的响应时间从约 0.030ms 增长到约 0.100ms,绝对数值与增长幅度均不大,这源于$\mathcal{C}$在一次响应过程中需要的运算操作并不消耗大量的系统资源.除了发送验证响应消息给移动终端外,$\mathcal{C}$还将与移动终端大量交互以完成具体的云服务功能响应,这些响应处理方式与验证响应基本一致.在不考虑具体云服务操作情况下,响应时间开销将与图 5 中的实验结果处于同一水平.$\mathcal{C}$负责处理的请求响应在本方案中将会被频繁执行,其占据方案实际运行交互量的较大比重,实验中反映出的低响应延时表明了方案在云服务提供商处具有良好的性能.

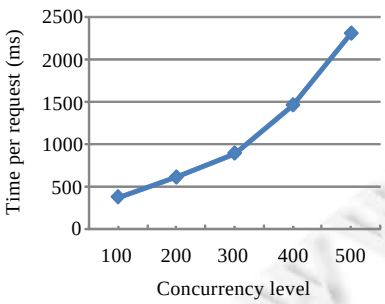


Fig.4 Authorization response latency in $\mathcal{A}$

图 4 $\mathcal{A}$对授权申请的响应延时

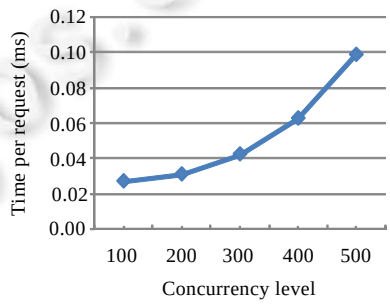


Fig.5 Authentication response latency in $\mathcal{C}$

图 5 $\mathcal{C}$对接入请求的响应延时

综合服务器端的性能表现,本文方案对于移动用户来说整体上具有较高的运行效率,所有操作等待和服务响应时延均在可接受范围内.

8 讨 论

本文提出的可信移动终端云服务安全接入方案可以有效防止敌手访问、窃取和篡改合法用户在云端服务存储的隐私数据,然而方案并没有考虑用户对于应用服务提供商$\mathcal{A}$和云服务提供商$\mathcal{C}$的匿名性:首先$\mathcal{A}$可以利用唯一的设备证书识别用户移动终端的身份;其次,$\mathcal{A}$和$\mathcal{C}$可以利用用户名 *user* 识别用户的身份.在识别身份的情况下,$\mathcal{A}$和$\mathcal{C}$可以链接用户或者移动终端对于云服务操作的所有行为,这间接导致了用户隐私的泄露.在某些场景中,例如移动云购物,用户可能在访问云服务购物时希望具有一定的匿名性,不愿意自己的购物和支付行为被服务提供者链接以造成购物习惯等隐私信息的泄露.事实上,现有的匿名认证系统可以解决这一问题,既能完成对用户和终端的合法性认证,又可以保护用户的匿名性.

在众多匿名认证系统中,直接匿名证明(direct anonymous attestation,简称 DAA)协议由 TCG 最先正式发布用于基于 TPM 的匿名证明,现已被接收为 ISO 标准.该协议可以被适当修改后应用于本文方案,以实现服务提供商对用户的匿名认证.在之前的工作中.我们基于 DAA 协议专为移动终端设计了 DAA-TZ 方案[42],方案使用了 TrustZone 的良好特性,可以提供安全高效的匿名认证服务,方案系统已被完整实现并进行了测试.因此,结合 DAA-TZ,设计和实现具有匿名属性的可信终端云服务安全接入方案已成为可能.

9 总 结

本文针对移动云计算场景分析了移动终端接入云服务的相关安全问题,提出了一种可信移动终端云服务安全接入方案.该方案首先利用 TrustZone 安全扩展技术构建可信移动终端体系结构,可信移动终端使用 SRAM PUF 获得根密钥种子,实现了密钥与敏感数据的安全管理机制;其次,借鉴可信计算技术思想,在可信移动终端的基础上设计了云服务安全接入协议,协议兼容可信云计算架构.本文分析了方案具有的安全属性,给出了基于方案设计的移动云存储应用实例,实现了方案的原型系统.分析和实验结果表明:本文提出的安全接入方案能够有效实现移动终端在接入云服务过程中的安全认证,保护移动用户在云服务端的私有数据安全;方案具有较好的可扩展性和较小的移动终端 TCB,其整体运行效率较高,移动用户等待时延在可接受范围之内.在未来的工作中,我们将对方案中提出的安全接入协议进行形式化分析,给出更为详细的安全证明.

致谢 在此,感谢中国科学院软件研究所赵世军博士、首都师范大学张倩颖博士对本文工作的支持和帮助,感谢实验室杨糠同学在本文写作过程中给予的建议和鼓励.

References:

[1] Dinh HT, Lee C, Niyato D, Wang P. A survey of mobile cloud computing: Architecture, applications, and approaches. Wireless Communications & Mobile Computing, 2013,13(18):1587−1611. [doi: 10.1002/wcm.1203]

[2] Alzahrani A, Alalwan N, Sarrab M. Mobile cloud computing: advantage, disadvantage and open challenge. In: Proc. of the 7th Euro American Conf. on Telematics and Information Systems. ACM Press, 2014. [doi: 10.1145/2590651.2590670]

[3] Preston AC. Mobile cloud computing: Devices, trends, issues, and the enabling technologies. IBM Developer Works, 2011.

[4] Fernando N, Seng WL, Rahayu W. Mobile cloud computing: A survey. Future Generation Computer Systems, 2013,29(1):84−106. [doi: 10.1016/j.future.2012.05.023]

[5] Visiongain. Mobile cloud computing industry outlook report. 2011−2016. Report, 2011. 1−153. https://www.visiongain.com/Report/737/Mobile-Cloud-Computing-Industry-Outlook-Report-2011-2016

[6] Feng DG, Zhang M, Zhang Y, Xu Z. Study on cloud computing security. Ruan Jian Xue Bao/Journal of Software, 2011,22(1): 71−83 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/3958.htm [doi: 10.3724/ SP.J.1001.2011.03958]

[7] Wang YD, Yang JH, Xu C, Ling X, Yang Y. Survey on access control technologies for cloud computing. Ruan Jian Xue Bao/ Journal of Software, 2015,26(5):1129−1150 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4820.htm [doi: 10.13328/j.cnki.jos.004820]

[8] Chris H, Paul S. Security guidance for critical areas of focus in cloud computing V3.0. Technique Guidance, Cloud Security Alliance, 2011.

[9] Proudler G, Chen LQ, Dalton C. Trusted Computing Platforms. Berlin, Heidelberg: Springer-Verlag, 2014. 1–393. [doi: 10.1007/978-3-319-08744-3]

[10] Nicolae P. Trusted computing and secure virtualization in cloud computing [MS. Thesis]. Swedish Institute of Computer Science, 2012.

[11] Zhao B, Yan F, Zhang LQ, Wang J. Build trusted cloud computing environment. Communications of CCF (China Computer Federation), 2012,8(7):28−34 (in Chinese with English abstract).

[12] Santos N, Gummadi KP, Rodrigues R. Towards trusted cloud computing. 2009. https://www.usenix.org/legacy/event/hotcloud09/tech/full_papers/santos.pdf

[13] John M, Tom R, Fred S. The CloudProxy Tao for trusted vomputing. Technical Report, No.UCB/EECS-2013-135, University of California at Berkeley, 2013. http://www.eecs.berkeley.edu/Pubs/TechRpts/2013/EECS-2013-135.html

[14] Ko RKL, Jagadpramana P, Mowbray M, Pearson S, Kirchberg M, Liang Q, Lee BS. TrustCloud: A framework for accountability and trust in cloud computing. In: Proc. of the 2011 IEEE World Congress on Services. IEEE Computer Society, 2011. 584–588. [doi: 10.1109/SERVICES.2011.91]

[15] Martignon, L, Poosankam P, Zaharia M, Han J, Mccamant S, Song D, Paxson V, Perrig A, Shenker S, Stoica I. Cloud terminal: Secure access to sensitive applications from untrusted systems. In: Proc. of the 2012 USENIX Conf. on Annual Technical Conf. 2012. 14–14.

[16] Trusted Computing Group. TPM MOBILE with trusted execution environment for comprehensive mobile device security. White Paper, Trusted Computing Group, Incorporated, 2012. http://www.trustedcomputinggroup.org

[17] ARM. ARM security technology: Building a secure system using TrustZone technology. ARM Limited, 2009. http://infocenter.arm.com/help/topic/com.arm.doc.prd29-genc-009492c/PRD29-GENC-009492C_trustzone_security_whitepaper.pdf

[18] Zhao SJ, Zhang QY, Hu GY, Qin Y, Feng DG. Providing root of trust for ARM TrustZone using on-chip SRAM. In: Proc. of the 4th Int'l Workshop on Trustworthy Embedded Devices. ACM Press, 2014. [doi: 10.1145/2666141.2666145]

[19] Trusted Computing Group. TPM main specification, version1.2, revision 116. Trusted Computing Group, Incorporated, 2011. http://www.trustedcomputinggroup.org

[20] Trusted Computing Group. Trusted platform module library, family 2.0, revision 01.16. Trusted Computing Group, Incorporated, 2014. http://www.trustedcomputinggroup.org

[21] Wu QX, Yang XW, Zou H, Yu FJ, Ning XK, Wang Z. Technic specification of cryptography supporting platform for trusted computing. China State Password Administration Committee, 2007 (in Chinese). http://www.oscca.gov.cn

[22] Berger S, Caceres R, Goldman KA, Perez R, Sailer R, Doorn L. vTPM: Virtualizing the trusted platform module. In: Proc. of the 15th USENIX Security, 2006. 305–320.

[23] Mccune JM, Li Y, Qu N, Zhou Z, Datta A, Gligor V, Perrig A. TrustVisor: Efficient TCB reduction and attestation. In: Proc. of the IEEE Symp. on Security and Privacy (S&P), IEEE, 2010. 143–158. [doi: 10.1109/SP.2010.17]

[24] Klein A, Mannweiler C, Schneider J. Access schemes for mobile cloud computing. In: Proc. of the 11th Int'l Conf. on Mobile Data Management. IEEE, 2010. 387–392. [doi: 10.1109/MDM.2010.79]

[25] Khana AN, Kiaha MLM, Khanb SU, Madanic SA. Towards secure mobile cloud computing: A survey. Future Generation Computer Systems, 2013,29(5):1278−1299. [doi: 10.1016/j.future.2012.08.003]

[26] Wu C, Zhou YJ, Patel K, Liang ZK, Jiang XX. AirBag: Boosting smartphone resistance to malware infection. In: Proc. of the 2014 Network and Distributed System Security Symp. (NDSS 2014). Internet Society, 2014.

[27] Trusted Computing Group. TCG mobile trusted module specification, version1.0, revision 7.02. Trusted Computing Group, Incorporated, 2010. http://www.trustedcomputinggroup.org

[28] Samuel AB, Don F, Virginie G, Franz H, Janne H, Milas F. The trusted execution environment: Delivering enhanced security at a lower cost to the mobile market. White Paper, GlobalPlatform, 2011.

[29] Atul V. Get into the zone: Building secure systems with ARM TrustZone technology. White Paper, Texas Instruments, 2013.

[30] Santos N, Raj H, Saroiu S, Wolman A. Using ARM TrustZone to build a trusted language runtime for mobile applications. In: Proc. of the ASPLOS 2014. ACM Sigplan Notices, 2014,49(1):67−80. [doi: 10.1145/2541940.2541949]

[31] Yang B, Feng DG, Qin Y. A lightweight anonymous mobile shopping scheme based on DAA for trusted mobile platform. In: Proc. of the IEEE 13th Int'l Conf. on Trust, Security and Privacy in Computing and Communications (TrustCom 2014). IEEE, 2014. 9–17. [doi: 10.1109/TrustCom.2014.6]

[32] Samsung. An overview of Samsung KNOX. White Paper, Samsung Electronics Co., Ltd, 2013.

[33] Dodis Y, Reyzin L. Fuzzy extractors: How to generate strong keys from biometrics and other Noisy data. In: Proc. of the Advances in Cryptology (EUROCRYPT 2004). Berlin, Heidelberg: Springer-Verlag, 2004. 523−540. [doi: 10.1007/978-3-540-24676-3_31]

[34] Guajardo J, Kumar SS. FPGA intrinsic PUFs and their use for IP protection. In: Proc. of the 9th Int'l Workshop on Cryptographic Hardware and Embedded Systems. Berlin, Heidelberg: Springer-Verlag, 2007. 63−80. [doi: 10.1007/978-3-540-74735-2_5]

[35] GlobalPlatform Device Technology. TEE client API specification version 1.0. GlobalPlatform, 2010. http://globalplatform.org

[36] Zhang YJ, Feng DG, Qin Y, Yang B. A TrustZone based trusted code execution with strong security requirements. Journal of Computer Research and Development, 2015,52(10):2224−2238 (in Chinese with English abstract).

[37] Jang J, Kong S, Kim M, Kim D, Kang BB. SeCReT: Secure channel between rich execution environment and trusted execution environment. In: Proc. of the 2015 Network and Distributed System Security Symp. (NDSS 2015). Internet Society, 2015. [doi: 10.14722/ndss.2015.23189]

[38] Xilinx. Zynq-7000 all programmable SoC ZC702 evaluation kit. http://www.xilinx.com/products/boards-and-kits/ek-z7-zc702-g.html

[39] Integrated Silicon Solution, Inc. IS61LV6416-10TL. http://www.alldatasheet.com/datasheet-pdf/pdf/505020/ISSI/IS61LV6416-10TL.html

[40] Morelos-Zaragoza R. Encoder/decoder for binary BCH codes in C (Version 3.1). http://www.rajivchakravorty.com/source-code/uncertainty/multimedia-sim/html/bch_8c-source.html

[41] Li WH, Li HB, Chen HB, Xia YB. AdAttester: Secure online mobile advertisement attestation using TrustZone. In: Proc. of the 13th Annual Int'l Conf. on Mobile Systems, Applications, and Services (MobiSys 2015). ACM Press, 2015. 75−88. [doi: 10.1145/2742647.2742676]

[42] Yang B, Yang K, Qin Y, Zhang ZF, Feng DG. DAA-TZ: An efficient DAA scheme for mobile devices using ARM TrustZone. In: Proc. of the 8th Int'l Conf. on Trust and Trustworthy Computing (TRUST 2015). LNCS 9229, Springer Int'l Publishing, 2015. 209−227. [doi: 10.1007/978-3-319-22846-4_13]

附中文参考文献:

[6] 冯登国,张敏,张妍,徐震.云计算安全研究.软件学报,2011,22(1):71−83. http://www.jos.org.cn/1000-9825/3958.htm [doi: 10.3724/SP.J.1001.2011.03958]

[7] 王于丁,杨家海,徐聪,凌晓,杨洋.云计算访问控制技术研究综述.软件学报,2015,26(5):1129−1150. http://www.jos.org.cn/1000-9825/4820.htm [doi: 10.13328/j.cnki.jos.004820]

[11] 赵波,严飞,张立强,王鹃.可信云计算环境的构建.中国计算机学会通讯,2012,8(7):28−34.

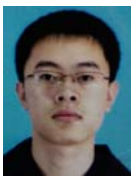
[21] 吴秋新,杨贤伟,邹浩,余发江,宁晓魁,王梓.可信密码支撑平台技术规范.国家密码管理局,2007. http://www.oscca.gov.cn

[36] 张英骏,冯登国,秦宇,杨波.基于 TrustZone 的强安全需求环境下可信代码执行方案.计算机研究与发展,2015,52(10):2224−2238.

杨波(1988－),男,河南沁阳人,博士生,主要研究领域为可信计算,移动平台安全,匿名系统.

秦宇(1979－),男,博士,高级工程师,主要研究领域为信息安全,可信计算.

冯登国(1965－),男,博士,研究员,博士生导师,CCF 高级会员,主要研究领域为网络与系统安全.



张英骏(1990－),男,博士生,主要研究领域为网络与系统安全,可信计算.