软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2018,29(9):2874−2895 [doi: 10.13328/j.cnki.jos.005264]

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

云环境中可信虚拟平台的远程证明方案研究*

胡玲碧[1], 谭 良[1,2]

[1](四川师范大学 计算机科学学院,四川 成都 610068)

[2](中国科学院 计算机技术研究所,北京 100190)

通讯作者: 胡玲碧, E-mail: csjkhlb@163.com

摘 要: 云环境中如何证明虚拟平台的可信,是值得研究的问题.由于云环境中虚拟平台包括运行于物理平台上的虚拟机管理器和虚拟机,它们是不同的逻辑运行实体,具有层次性和动态性,因此,现有的可信终端远程证明方案,包括隐私 CA(privacy certification authority,简称 PCA)方案和直接匿名证明(direct anonymous attestation,简称 DAA)方案,都并不能直接用于可信虚拟平台.而 TCG 发布的 Virtualized Trusted Platform Architecture Specification 1.0 版中,可信虚拟平台的远程证明方案仅仅是个框架,并没有具体实施方案.为此,提出了一种自顶向下的可信虚拟平台远程证明实施方案——TVP-PCA.该方案是在虚拟机中设置一个认证代理,在虚拟机管理器中新增一个认证服务,挑战方首先通过顶层的认证代理证明虚拟机环境可信,然后通过底层的认证服务证明运行于物理平台上的虚拟机管理器可信,顶层和底层证明合起来确保了整个虚拟平台的可信,有效解决了顶层证明和底层证明的同一性问题.实验结果表明,该方案不仅能够证明虚拟机的可信,而且还能证明虚拟机管理器和物理平台的可信,因而证明了云环境中的虚拟平台是真正可信的.

关键词: 可信计算;可信虚拟平台;远程证明;可信云环境

中图法分类号: TP316

中文引用格式: 胡玲碧,谭良.云环境中可信虚拟平台的远程证明方案研究.软件学报,2018,29(9):2874−2895. http://www.jos.org.cn/1000-9825/5264.htm

英文引用格式: Hu LB, Tan L. Research on the trusted virtual platform remote attestation method in cloud computing. Ruan Jian Xue Bao/Journal of Software, 2018,29(9):2874−2895 (in Chinese). http://www.jos.org.cn/1000-9825/5264.htm

Research on Trusted Virtual Platform Remote Attestation Method in Cloud Computing

HU Ling-Bi[1], TAN Liang[1,2]

[1](College of Computer Science, Sichuan Normal University, Chengdu 610068, China)

[2](Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China)

Abstract: In cloud computing, how to prove the trust of a virtual platform is a hot problem. A virtual platform includes the virtual machine manager that runs on the physical platform and the virtual machines that are different logical entities with hierarchy and dynamics. Existing trusted computing remote attestation schemes, such as the privacy certification authority (PCA) scheme and the direct anonymous attestation (DAA) scheme, cannot be directly used for trusted virtual platform. Moreover, the remote attestation scheme of trusted virtual platform in virtualized trusted platform architecture specification of TCG is only a framework without concrete implementation plan. To address these issues, this paper proposes a top-down remote attestation project, called TVP-PCA, for trusted virtual platform. This project designs and implements an attestation agent in the top-level virtual machine and an attestation service in the

* 基金项目: 国家自然科学基金(61373162); 四川省科技支撑项目(2014GZ007)

Foundation item: National Natural Science Foundation of China (61373162); Sichuan Science and Technology Support Project (2014GZ007)

收稿时间: 2016-09-04; 修改时间: 2016-12-07; 采用时间: 2017-01-20; jos 在线出版时间: 2017-07-12

CNKI 网络优先出版: 2017-07-12 15:33:16, http://kns.cnki.net/kcms/detail/11.2560.TP.20170712.1533.002.html

underlying virtual machine manager. With this approach, a challenger can first use the top-level agent to prove that the virtual machine is trusted, and then use the underlying service to prove that the virtual machine manager can be trusted, both attestations together ensure the credibility of the entire virtual platform. This paper solves the identity problem of the top-level attestation and the underlying attestation effectively. Experiments show that this project can not only prove the trust of the virtual machine, but also prove the trust of the virtual machine manager and the physical platform, thus establishing that the virtual platform of the cloud computing is trusted.

Key words: trusted computing; trusted virtual platform; remote attestation; trusted cloud computing

虚拟化技术是指将各种计算机资源通过转化后呈现出来的方法,由于它能够最大化利用物理设备为用户节约成本而被广泛利用.云计算[1−5]是虚拟化技术的应用场景之一.在虚拟化技术的支持下,云计算可以提供更为灵活可变、可扩展的服务平台.然而对于这种虚拟平台,用户怎样确定虚拟平台提供的资源和服务是安全的,怎样确保虚拟平台是可信的,这是一个至关重要的问题.而可信计算[6−9]是保障计算平台可信的基础手段,它通过提供数据保护、身份证明以及完整性测量、存储与报告等功能,以提高计算平台整体的可信性.因此,将可信计算技术融入虚拟平台,已成为云安全研究领域的一大热点[10−40].

可信基于证明,只有证明才能在不可信的环境中建立信任关系[16−19].一般可信终端的远程证明是TPM完整性度量报告,即:从物理 TPM 出发,沿着信任链一直认证到应用程序,以确保整个终端是可信的[11,20].TCG 推荐了两种证明方案:其一是 PCA 方案,其二是 DAA 方案.而对于虚拟平台的远程证明,国内外学者的研究正逐步展开[36−40].在云环境中,当分配给云租户的客户虚拟机作为一个通用的可信终端时,其远程证明与 TCG 架构下可信终端远程证明具有较大的相似性,其证明既可以采用 PCA 也可以采用 DAA,如果 vTPM(virtualization trusted platform module)能够按照 TCG 规范正确设计和执行,那么远程证明者就能够相信客户虚拟机的软件配置和完整性度量报告[22,23].然而,vTPM 与 TPM(trusted platform module)并不完全相同,vTPM 是软件,且基于云环境,一般云环境包括 IaaS(infrastructure as a service)层、PaaS(platform as a service)层、SaaS(software as a service)层,相互是彼此独立的,从运行状态角度看,每个层次内的运行节点是动态化的,物理平台和虚拟机管理器任何时候都可能发生变化.显然,可信虚拟平台的远程认证需要将信任链从物理 TPM 扩展到了虚拟机上,如果将现有的认证方法运用于可信虚拟平台,认证只会从 vTPM 出发直到虚拟机上的应用程序.因此,原样将可信终端远程证明已有的研究成果移植到可信虚拟平台是不行的.另外,尽管 TCG 在发布的 Virtualized Trusted Platform Architecture Specification 1.0 版中有可信虚拟平台的远程证明方案,不过该远程证明方案仅是个框架[6],并没有具体实施方案,还存在一些难以遇见的困难.

针对以上不足,本文提出了一种自顶向下的可信虚拟平台远程证明实施方案——TVP-PCA.该方案是在虚拟机中设置一个认证代理,在虚拟机管理器中新增一个认证服务,挑战方首先通过顶层的认证代理证明虚拟机环境可信,然后通过底层的认证服务证明运行于物理平台上的虚拟机管理器可信,顶层和底层证明合起来确保了整个虚拟平台的可信.

1 相关工作

对于可信终端的远程证明,到现在为止取得了较多研究成果,TCG 组织一直是这一工作积极的主导者和推动者.TCG 最初采用的是 EK 证书证明身份平台,由于直接使用背书密钥会暴露平台身份信息,所以在 TCG 规范 1.1 版本中,为避免平台身份信息泄露,提出使用 PCA 方案进行远程证明.在该方案中,可信平台需要向 CA 申请 AIK 证书,导致 CA 中心会负载过大.文献[21]对 PCA 方案进行了优化,提出用证书和令牌标识可信计算平台并直接使用令牌证明平台身份方案,平台申请一次证书即可以利用该证书多次申请身份令牌,然后直接使用令牌进行身份证明.文献[22]认为:一般的终端可信只需满足信任链以及相关软件配置的可信证明,并不能保证终端运行环境的可信.针对此问题,提出一种可信终端运行环境远程证明方案,通过对终端的静态环境和动态环境两方面的证明来保证终端运行环境的可信,该方案本质上属于 PCA 范畴.文献[23]提出将 Privacy CA 应用于无线网络环境终端平台可信性的远程认证,可以正确验证移动节点的可信.文献[24]提出将可信第三方运用于移动互联网下 MTT 的远程认证.2004 年,E.Brickell 等人提出 DAA 方案[25],通过采用零知识证明以及群签名等技术,

使得证明方只需要申请一次证书就可以在不暴露平台的真实身份信息前提下,向不同的挑战方证明自己的平台是一个可信平台.该协议不需要可信第三方的在线参与,同时又保证了可信平台的匿名性.TCG 在 TPM 规范 V1.2 中采用此方案,但此方案一次证明至少需要 3 次零知识证明,因而具有性能较差、效率低、复杂度较大等特点.为了提高 DAA 方法的效率和性能,文献[26,27]提出通过降低签名过程中可信平台模块的计算量来对其进行优化,文献[26]的具体办法是通过缩短签名长度,文献[27]的具体办法是通过用椭圆曲线的点加和标量乘取代椭圆曲线离散对数.文献[28]针对现有 DAA 方案存在计算开销大和无法满足跨域匿名认证需求的不足,提出基于身份的直接匿名认证机制,采用代理签名和直接匿名证明技术实现移动互联网下可信移动平台(TMP)的跨域匿名认证.该机制具有匿名性、无关联性和高性能等性质的同时,能够抵抗平台的伪装攻击、替换攻击和重放攻击等敌手攻击行为.杨力在文献[29]中提出:将 DAA 运用于无线网络可信接入的远程证明,由外地网络代理服务器直接验证平台身份和完整性,借助本地网络代理服务器验证用户身份,既安全高效又能满足无线移动网络需求.之后,他在文献[30]中运用代理签名技术和直接匿名证明方法,实现对移动终端在多可信域之间漫游时的可信计算平台认证,接着还提出通过引入会话密钥协商机制,抵抗重放攻击和平台伪装攻击[31].文献[32]用 DAA 方法实现了对可信终端动态运行环境的可信证据收集代理,该代理能够为其终端动态环境的可信性提供客观、全面的运行时可信证据,并且通过特定的可信性评估模型监控终端的运行状态.文献[33,34]还把可信第三方引入直接匿名认证方法以提高认证效率.特别地,文献[34]提出的 TMP-UAA 模型还能够有效地解决直接匿名认证中存在的 R 攻击及跨信任域问题.文献[35]也是将 DAA 应用于移动平台可信证明,提出了基于 TrustZone 安全技术可信移动平台体系结构,这个框架结构能够兼容 DAA 方案和曲线,还具有较高的计算速度.除此以外,国内外众多研究机构和学者提出了许多不同的远程证明方法,其中,研究成果最多的是基于属性的远程证明[22],这方面的成果评述读者可参考文献[22].

在可信虚拟平台的远程证明方面,TCG 虚拟化平台工作组已发布了 Virtualized Trusted Platform Architecture Specification 1.0 版[6],其中也包含了可信虚拟平台的远程证明框架,但并没有具体实施方案.除此以外,一些学者对此也展开了研究.文献[36]提出了一种在可信云环境中证明虚拟软件服务的框架,通过虚拟客户机中运行一个可信测量模块,测量平台状态值和可信平台中已经有的状态表值做对比,来确定软件服务是否可信.这个框架可以有效地检测用户级应用的可信,但不足的是,没有对 VMM 进行可信证明.文献[37]提出一种对可信云环境中保证客户虚拟机可信的方法,认证虚拟化架构中的可信网络接口、可信辅助存储器、可信执行环境.但是该方法中认为,虚拟机管理操作系统是不重要的,并不对虚拟机管理操作系统做可信验证.文献[38]针对现有虚拟机身份证明方案中身份权威信息公开问题,提出云环境虚拟机匿名身份证明方案,该方案可以实现对身份权威信息的隐藏,避免位置信息暴露,做到真正的结构透明性和位置无关.但是该文献并没有提出对平台配置信息的证明方案.文献[39]提出一种基于 TrustZone 的可信移动终端云服务安全接入方案,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.另外,文献[40]也提出了一种可测量云环境中 SaaS 的可信认证框架,该认证更强调对攻击者的精确定位.然而,这两种架构都忽略对基础设施服务和平台服务没有提出相关的检测方法.

综上,一方面,对于已有的可信终端远程证明研究成果,只能用于有一个逻辑运行实体的可信证明,不能直接用于虚拟化平台,因为虚拟化平台同时有多个逻辑运行实体;另一方面,在可信虚拟平台的远程证明方面,尽管 TCG 在发布了的 Virtualized Trusted Platform Architecture Specification 1.0 版中有可信虚拟平台的远程证明框架,不过该远程证明方案仅是个框架,并没有具体实施方案,还存在一些难以遇见的困难.而其他学者对可信虚拟平台远程证明的研究成果并不完全是 TCG 规范定义的可信证明,均不能完全满足规范 Virtualized Trusted Platform Architecture Specification 1.0 版.

2 云环境中可信虚拟平台远程证明方案——TVP-PCA

我们提出的可信虚拟平台远程证明方案 TVP-PCA 实施过程如图 1 所示,该方法中共包含 6 个实体:挑战者、TPM、vTPM、虚拟机认证代理、虚拟机管理器认证服务、证书中心(CA).其中,挑战者是资源请求方,要求提供

资源的一方即可信虚拟平台是可信的,通过对可信虚拟平台请求的可信证据对其进行可信验证,从而做出访问决策;TPM 是一个具有存储功能、密码生成和管理的小型片上系统 SoC(system on chip),它作为可信计算平台的信任根,可以为平台提供身份证明、平台度量、安全存储等;vTPM 是具有和 TPM 相同功能的软件产品,在虚拟化技术下,作为虚拟机的信任根,为虚拟机提供平台身份证明信息;虚拟机认证代理主要负责代替 vTPM 向挑战者通信并提供虚拟机的可信证据,挑战者向代理发送远程认证请求,代理收到请求后,转交 vTPM 提供的可信证据给挑战者,完成对虚拟机的远程认证;虚拟机管理器认证服务主要是代替 TPM 向挑战者通信,并提供运行在物理平台之上的虚拟机管理器的可信证据,挑战者向服务发送远程认证请求,服务收到请求后,转交 TPM 提供的可信证据给挑战者,完成对运行与物理平台之上的虚拟机管理器的远程认证;证书中心是负责为提供 EK 证书的 TPM 和 vTPM 颁发 AIK 证书,还有为虚拟机认证代理和虚拟机管理器认证服务等可信软件颁发安全证书.

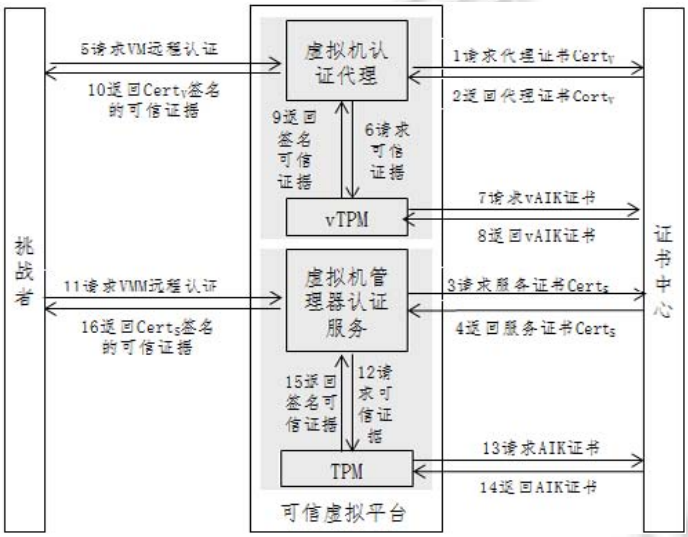


Fig.1 Trusted virtual platform remote attestation method

图 1 可信虚拟平台远程证明方案

由于可信虚拟平台可以认为是两层结构:虚拟机和运行于物理平台上的虚拟机管理器.远程认证时采取自顶向下的方法,即先对顶层的虚拟机进行远程认证,再对运行于物理平台之上的虚拟机管理器进行远程认证.本文提出的 TVP-PCA 方法具体包括 3 个阶段:第 1 阶段是完成虚拟机认证代理和虚拟机管理器认证服务的初始化(步骤 1~步骤 4),虚拟机认证代理从 CA 处获得代理证书 $Cert_V$,虚拟机管理器认证服务从 CA 处得到服务证书 $Cert_S$;第 2 阶段是虚拟机远程证明阶段(步骤 5~步骤 10),当挑战者向虚拟机认证代理提出平台远程证明请求时,认证代理需要返回虚拟机可信证据即签名的 vAIK 证书和 vPCR 值,代理在收到请求之后,从 vTPM 获得 vAIK 证书和 vPCR 值,其中,vAIK 证书是 vTPM 向 CA 中心申请的,代理把收到的可信证据用 $Cert_V$ 签名发给挑战者,挑战者对可信证据进行验证完成虚拟机的可信认证;第 3 阶段是运行于物理平台之上的虚拟机管理器证明阶段(步骤 11~步骤 16),挑战者向虚拟机管理器认证服务提出远程认证请求时,认证服务需要返回可信证据,即认证服务签名的 AIK 证书和 PCR 值,服务在收到请求之后,从 TPM 获得 AIK 证书和 PCR 值,其中,AIK 证书是 TPM 向 CA 中心申请的,服务把收到的可信证据用 $Cert_S$ 签名发给挑战者,挑战者对可信证据进行验证,完成运行于物理平台之上的虚拟机管理器的可信认证.

下面,我们将详细介绍初始化阶段、虚拟机远程证明阶段、运行于物理平台之上的虚拟机管理器证明阶段的证明过程.为了更加专注本文所要解决的问题,我们作如下假设.

- TPM 和 vTPM 是整个虚拟平台的 TCB;
- VMAgent,VMMService 和 vTPM 是稳定的和安全的,它们受 TPM 保护,抗恶意攻击,对云平台或虚拟机

的稳定性和安全性影响在本文中暂不考虑;

- Challenger 是诚实的,即,挑战方确实是想验证虚拟平台及虚拟机的可信性而发出验证请求.

2.1 初始化阶段协议

初始化阶段包括两部分.

- 其一是虚拟机认证代理的初始化,如图 2 所示,主要功能是获取代理证书.

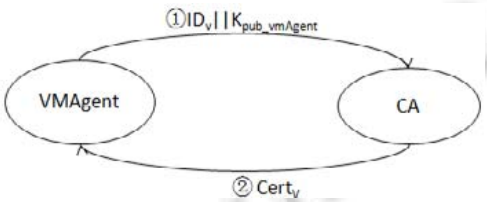


Fig.2 Initialzation of virtual machine remote attestation agent

图 2 虚拟机认证代理初始化

我们用 *VMAgent* 代表虚拟机认证代理,($K_{pub\_vmAgent}$,$K_{pri\_vmAgent}$)表示虚拟机认证代理的密钥对,$URL_{vmAgent}$ 是 *VMAgent* 的域名.具体交互过程如下.

1. $VMAgent \rightarrow CA: ID_v || K_{pub\_vmAgent}$;
2. $CA \rightarrow VMAgent: Cert_V=[ID_v||K_{pub\_vmAgent}||URL_{vmAgent}||T_{00}||Sig_{pri\_CA}(ID_v||K_{pub\_vmAgent}||URL_{vmAgent}||T_{00})]$.

在步骤 1 中,*VMAgent* 向 *CA* 发送证书请求,发送的具体内容:*VMAgent* 标识符 ID_v 以及 *VMAgent* 的公钥 $K_{pub\_vmAgent}$;步骤 2 中,*CA* 给虚拟机认证代理签发证书并公开,证书中的信息有代理的标识符 ID_v,$K_{pub\_vmAgent}$,$URL_{vmAgent}$ 和时间戳 T_{00},虚拟机认证代理初始化过程完成.

- 其二是虚拟机管理器认证服务初始化,如图 3 所示,主要功能是获取认证服务证书.

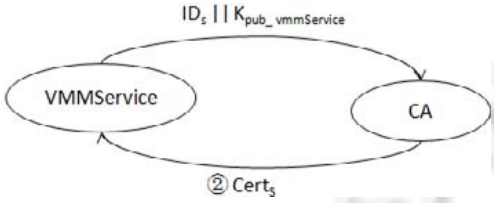


Fig.3 Initialzation of virtual machine manager remote attestation service

图 3 虚拟机管理器认证代理初始化

我们用 *VMMService* 代表虚拟机管理器认证服务,*CA* 表示证书中心,(K_{pub_CA},K_{pri_CA})标识 *CA* 用户签名验证的密钥对,($K_{pub\_vmmService}$,$K_{pri\_vmmService}$)表示虚拟机管理器认证服务的密钥对,$URL_{vmmService}$ 是 *VMMService* 的域名.具体交互过程如下.

1. $VMMService \rightarrow CA: ID_s || K_{pub\_vmmService}$;
2. $CA \rightarrow VMMService: Cert_S=[ID_s||K_{pub\_vmmService}||URL_{vmmService}||T_{10}||Sig_{pri\_CA}(ID_s||K_{pub\_vmmService}||URL_{vmmService}||T_{10})]$.

在步骤 1 中,*VMMService* 向 *CA* 发送证书请求,发送的具体内容:*VMMService* 标识符 ID_s 以及 *VMMService* 的公钥 $K_{pub\_vmmService}$;步骤 2 中,*CA* 给虚拟机管理器认证服务签发证书并公开.$Cert_S$ 中的信息有认证服务的标识符 ID_s,$K_{pub\_vmmService}$,$URL_{vmmService}$ 和时间戳 T_{10}.至此,虚拟机管理器认证服务初始化过程完成.

2.2 顶层虚拟机远程证明阶段协议

虚拟机远程证明阶段可以分为两个部分:其一是虚拟机向挑战者提供可信证明信息;其二是挑战者对可信

证据进行验证,做出访问决策.

第 1 部分虚拟机向挑战者提供可信证明,即,虚拟机认证代理收集证明信息并交给挑战者.我们用 *Challenger* 代表挑战者,*VMAgent* 代表虚拟机认证代理,*vTPM* 代表虚拟机的可信平台模块,$Cert_{vEK}$ 表示 *vEK* 证书,K_{s0} 是由 *Challenger* 选定的对称密钥,用于和 *VMAgent* 之间的保密通信,$(K_{pub_vAIK},K_{pri_vAIK})$表示 *vTPM* 的 *vAIK* 密钥对,$Cert_{vAIK}$ 表示 *vAIK* 证书,T_{01} 是一个时间戳,*vPCRs* 表示 *vTPM* 寄存器中的值,*Challenger* 和 *VMAgent* 双方选择素数 q 以及 q 的一个原根 a.为了方便协议叙述,*Sig* 表示签名算法,*Ver* 表示验证算法,如图 4 所示.

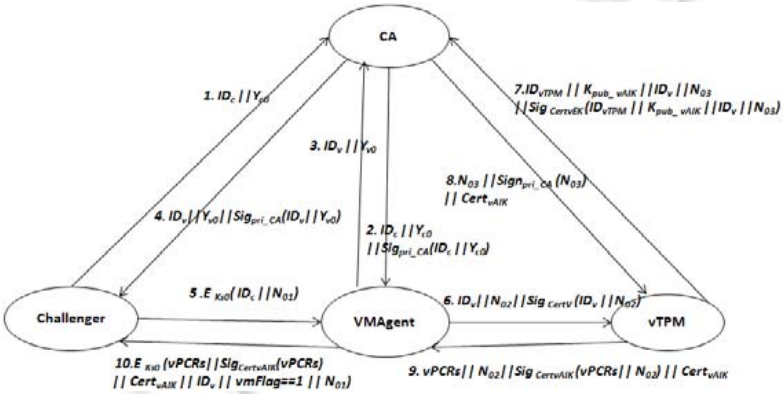


Fig.4 Process of virtual machine remote attestation

图 4 虚拟机远程证明过程

具体过程交互如下.

1. *Challenger*→*CA*:$ID_c||Y_{c0}$,其中,$Y_{c0}=a^{X_{c0}} \bmod q$,$X_{c0}$ 是 *Challenger* 的任选素数,$X_{c0}<q$;
2. *CA*→*VMAgent*:$ID_c||Y_{c0}||Sig_{pri\_CA}(ID_c||Y_{c0})$.*VMAgent* 验证 *CA* 对 *Challenger* 的签名 $Ver_{pub\_CA}(Sig_{pri\_CA}(ID_c||Y_{c0}))$,然后任选素数 X_{v0},且 $X_{v0}<q$,并计算 $K_{s0}=Y_{c0}^{X_{v0}} \bmod q$;
3. *VMAgent*→*CA*:$ID_v||Y_{v0}$,其中,$Y_{v0}=a^{X_{v0}} \bmod q$;
4. *CA*→*Challenger*:$ID_v||Y_{v0}||Sig_{pri\_CA}(ID_v||Y_{v0})$.
   Challenger 验证 *CA* 对 *VMAgent* 的签名 $Ver_{pub\_CA}(Sig_{pri\_CA}(ID_v||Y_{v0}))$,并计算 $K_{s0}=Y_{v0}^{X_{c0}} \bmod q$;
5. *Challenger*→*VMAgent*: $E_{K_{s0}}(ID_c||N_{01})$,*VMAgent* 用 K_{s0} 解密得到 ID_c,N_{01};
6. *VMAgent*→*vTPM*:$ID_v||N_{02}||Sig_{Cert_V}(ID_v||N_{02})$,*vTPM* 验证 *VMAgent* 的签名:
   $$Ver_{pub\_vmAgent}(Sig_{Cert_V}(ID_v||N_{02}));$$
7. *vTPM*→*CA*:$ID_{vTPM}||K_{pub\_vAIK}||ID_v||N_{03}||Sig_{Cert_{vEK}}(ID_{vTPM}||K_{pub\_vAIK}||ID_v||N_{03})$,*CA* 验证 *vEK* 证书对信息的签名 $Ver_{pub\_vEK}(Sig_{Cert_{vEK}}(ID_{vTPM}||K_{pub\_vAIK}||ID_v||N_{03}))$;
8. *CA*→*vTPM*:$N_{03}||Sig_{pri\_CA}(N_{03})||Cert_{vAIK}$,其中:
   $$Cert_{vAIK}=[K_{pub\_vAIK}||ID_{vTPM}||T_{01},Sig_{pri\_CA}(K_{pub\_vAIK}||ID_{vTPM}||T_{01})],$$
   vTPM 验证 *CA* 对 N_{03} 的签名 $Ver_{pub\_CA}(Sig_{pri\_CA}(N_{03}))$,*vTPM* 验证 *CA* 对 *vAIK* 证书的签名:
   $$Ver_{pub\_CA}(Sig_{pri\_CA}(K_{pub\_vAIK}||ID_{vTPM}||T_{01}));$$
9. *vTPM*→*VMAgent*:$vPCRs||N_{02}||Sig_{Cert_{vAIK}}(vPCRs||N_{02})||Cert_{vAIK}$,*VMAgent* 验证 *vAIK* 证书对信息的签名:
   $$Ver_{pub\_vAIK}(Sig_{Cert_{vAIK}}(vPCRs||N_{02}));$$
10. *VMAgent*→*Challenger*: $E_{K_{s0}}(vPCRs||Sig_{Cert_{vAIK}}(vPCRs)||Cert_{vAIK}||ID_v||vmFlag==1||N_{01})$,用 K_{s0} 解密信息,*Challenger* 验证 *vAIK* 证书对 *vPCRs* 的签名 $Ver_{pub\_vAIK}(Sig_{Cert_{vAIK}}(vPCRs))$;

下面,我们对上述每一步进行详细解释.具体如下.

步骤 1. *Challenge* 向 *CA* 发送请求信息,发送的具体内容有:*Challenge* 的标识符 ID_c、公开秘钥 Y_{c0},其中,$Y_{c0}=a^{X_{c0}} \bmod q$,$X_{c0}$ 是 *Challenger* 的任选素数,$X_{c0}<q$;

步骤 2. *CA* 向 *VMAgent* 转发请求信息,发送的具体内容有:ID_c、Y_{c0} 以及 *CA* 的签名 $Sig_{pri\_CA}(ID_c||Y_{c0})$.*VMAgent* 收到信息后,验证 *CA* 对 *Challenger* 的签名 $Ver_{pub\_CA}(Sig_{pri\_CA}(ID_c||Y_{c0}))$,然后任选素数 X_{v0},且 $X_{v0}<q$,并计算:

$$K_{s0}=Y_{c0}^{X_{v0}} \bmod q;$$

步骤 3. *VMAgent* 向 *CA* 发送请求信息,发送的具体内容有:*VMAgent* 标识符 ID_v,Y_{v0},其中,$Y_{v0}=a^{X_{v0}} \bmod q$;

步骤 4. *CA* 向 *Challenger* 转发请求信息,发送的具体内容包括:ID_v、Y_{v0} 以及 *CA* 的签名 $Sig_{pri\_CA}(ID_v||Y_{v0})$. *Challenger* 验证 *CA* 对 *VMAgent* 的签名 $Ver_{pub\_CA}(Sig_{pri\_CA}(ID_v||Y_{v0}))$,并计算 $K_{s0}=Y_{v0}^{X_{c0}} \bmod q$;

经过前面 4 步,*Challenger* 和 *VMAgent* 之间获得一个共享的秘密密钥 K_{s0}.这 4 步实际上是能防止中间人攻击的 Diffle-Hellman 算法.

步骤 5. *Challenger* 用通信秘钥 K_{s0} 加密向 *VMAgent* 请求远程认证,加密发送信息的具体内容为:*Challenger* 标识符 ID_c、现时 N_{01},*VMAgent* 用 K_{s0} 解密获得信息内容;

步骤 6. *VMAgent* 向 *vTPM* 发出可信证据信息请求,发送的具体内容有:*VMAgent* 标识符 ID_v、现时 N_{02} 以及 *VMAgent* 证书 $Cert_V$ 对信息的签名 $Sig_{Cert_V}(ID_v||N_{02})$,*vTPM* 收到信息后验证 *VMAgent* 的签名:

$$Ver_{pub\_vmAgent}(Sig_{Cert_V}(ID_v||N_{02}));$$

步骤 7. *vTPM* 向 *CA* 中心发送 *vAIK* 证书请求,发送的具体内容有:*vTPM* 标识符 ID_{vTPM}、*vAIK* 公钥 K_{pub_vAIK}、*VMAgent* 标识符 ID_v、现时 N_3 及用 *vEK* 证书对信息的签名 $Sig_{Cert_{vEK}}(ID_{vTPM}||K_{pub\_vAIK}||ID_v||N_{03})$,*CA* 验证消息:

$$Ver_{pub\_vEK}(Sig_{Cert_{vEK}}(ID_{vTPM}||K_{pub\_vAIK}||ID_v||N_{03}));$$

步骤 8. *CA* 为 *vTPM* 生成 *vAIK* 证书 $Cert_{vAIK}$ 并公开,$Cert_{vAIK}$ 具体内容有:ID_{vTPM}、K_{pub_vAIK}、时间戳 T_{01} 以及 *CA* 的签名 $Sig_{pri\_CA}(K_{pub\_vAIK}||ID_{vTPM}||T_{01})$,*CA* 中心向 *vTPM* 返回信息,信息的具体内容为:现时 N_{03} 及 *CA* 的签名 $Sig_{pri_CA}(N_{03})$和 *vAIK* 证书 $Cert_{vAIK}$,*vTPM* 验证 *CA* 对 *vAIK* 证书的签名 $Ver_{pub\_CA}(Sig_{pri\_CA}(K_{pub\_vAIK}||ID_{vTPM}||T_{01}))$ 和 *CA* 对现时的签名 $Ver_{pub\_CA}(Sig_{pri\_CA}(N_{03}))$;

步骤 9. *vTPM* 把收到的现时 N_{03} 和 *vTPM* 之前向 *CA* 发送请求时的现时 N_{03} 进行匹配,如果匹配成功,发送可信证据信息给 *VMAgent*,发送的具体内容有:*vPCRs*、现时 N_{02} 及 *vAIK* 对 *vPCRs* 的签名 $Sig_{Cert_{vAIK}}(vPCRs \| N_{02})$,*VMAgent* 验证 *vTPM* 对信息的签名 $Ver_{pub\_vAIK}(Sig_{Cert_{vAIK}}(vPCRs \| N_{02}))$;

步骤 10. *VMAgent* 收到信息后,把收到的现时 N_{02} 和 *VMAgent* 之前向 *vTPM* 发送请求时的现时 N_{02} 进行匹配,如果匹配成功,就用通信秘钥 K_{s0} 加密可信证据信息发送给 *Challenger*,加密的可信证据信息内容有:*vPCRs* 及 $Sig_{Cert_{vAIK}}(vPCRs)$、*VMAgent* 标识符 ID_v、值为 1 的虚拟机标识符 *vmFlag*、*URLvmmService*、现时 N_{01} 等.

第 2 部分是 *Challenger* 接收到 *VMAgent* 的返回值,用 K_{s0} 解密得到这些信息.首先把收到的现时 N_{01} 和 *Challenger* 之前向 *VMAgent* 发送请求时的现时 N_{01} 进行匹配,如果匹配成功,则挑战者验证 *CA* 对 *vAIK* 证书的签名 $Ver_{pub\_CA}(Sig_{pri\_CA}(K_{pub\_vAIK}||ID_{vTPM}||T_{01}))$,再用 *vAIK* 证书对验证 *vPCRs* 的签名 $Ver_{pub\_vAIK}(Sig_{Cert_{vAIK}}(vPCRs \| N_{02}))$,最后根据 *vPCRs* 值做出对虚拟机可信决策.

对于此阶段,有两个非常重要的关键点必须说明.

(1) *Challenger* 如何确定对方是虚拟平台.当 *Challenger* 发送出的请求被 *VMAgent* 接收,返回的信息中如果虚拟机标识符 *vmFlag* 的值为 1,则代表对方为虚拟机平台;反之,则为物理平台.如果对方是虚拟机平台,要确保真正可信需要进一步证明该虚拟机所在的虚拟机管理器和物理平台是可信的;

(2) 如何确定虚拟机管理器所在平台的远程认证服务地址.当 *vmFlag* 的值为 1 时,表明 *Challenger* 当前验证的是虚拟机,按照设计的流程,*Challenger* 应该继续验证运行于物理平台之上的虚拟机管理器,因此,*Challenger* 从公开的虚拟机管理器认证服务证书 $Cert_S$ 中获得地址 $URL_{vmmService}$,通过该地址,完成对运行于物理平台之上的虚拟机管理器进行远程认证.

2.3 底层运行于物理平台之上的虚拟机管理器证明阶段协议

运行于物理平台之上的虚拟机管理器证明阶段也可以分为两个部分:其一是运行于物理平台之上的虚拟机管理器向挑战者提供可信证明信息;其二是挑战者对可信证据进行验证,做出访问决策.

第 1 步运行于物理平台之上的虚拟机管理器向挑战者提供可信证明,即,虚拟机管理器认证服务收集信息并交给挑战者.我们用 *VMMService* 代表虚拟机管理器认证服务,K_{S1} 是由挑战者选定的对称密钥,用于和 *VMMService* 之间的保密通信,$(K_{pub_vmmService},K_{pri_vmmService})$表示虚拟机管理器认证服务的密钥对,*TPM* 代表可信平台模块,(K_{pub_EK},K_{pri_EK})表示 *TPM* 的 *EK* 密钥对,$Cert_{EK}$ 表示 *EK* 证书,$(K_{pub_AIK},K_{pri_AIK})$表示 *TPM* 的 *AIK* 密钥对,$Cert_{AIK}$ 表示 *AIK* 证书,T_{11} 是一个时间戳,*PCRs* 表示 *TPM* 寄存器中的值.*Challenger* 和 *VMMService* 双方选择素数 q_1 以及 q_1 的一个原根 a_1.如图 5 所示:

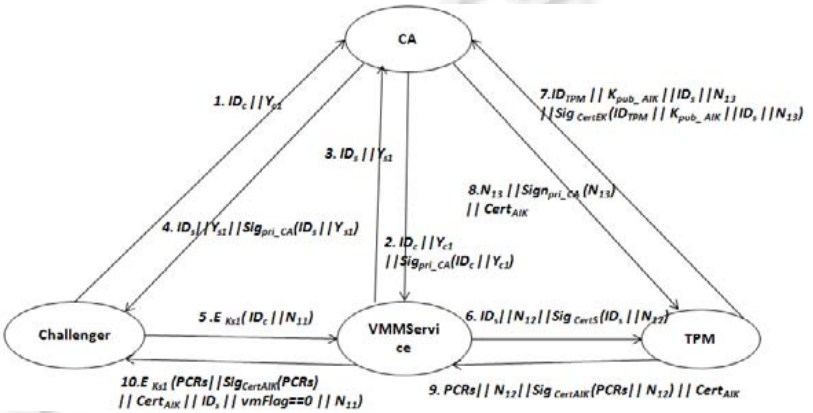


Fig.5 Process of virtual machine manager remote attestation

图 5 虚拟机管理器远程证明过程

具体过程交互如下.

1. *Challenger*→*CA*:$ID_c||Y_{c1}$,其中,$Y_{c1}=a_1^{X_{c1}} \bmod q_1$,$X_{c1}$ 是 *Challenger* 的任选素数,$X_{c1}<q_1$;

2. *CA*→*VMMService*:$ID_c||Y_{c1}||Sig_{pri\_CA}(ID_c||Y_{c1})$.*VMMService* 验证 *CA* 对 *Challenger* 的签名 $Ver_{pub\_CA}(Sig_{pri\_CA}(ID_c||Y_{c1}))$,然后任选素数 X_{s1},且 $X_{s1}<q_1$,并计算 $K_{S1}=Y_{c1}^{X_{s1}} \bmod q_1$;

3. *VMMService*→*CA*:$ID_s||Y_{s1}$,其中,$Y_{s1}=a_1^{X_{s1}} \bmod q_1$;

4. *CA*→*Challenger*:$ID_s||Y_{s1}||Sig_{pri\_CA}(ID_s||Y_{s1})$.*Challenger* 验证 *CA* 对 *VMMService* 的签名 $Ver_{pub\_CA}(Sig_{pri\_CA}(ID_s||Y_{s1}))$,并计算 $K_{s1}=Y_{s1}^{X_{c1}} \bmod q_1$;

5. *Challenger*→*VMMService*: $E_{K_{s1}}(ID_c||N_{11})$,*VMMService* 用 K_{s1} 解密得到 ID_c,N_{11};

6. *VMMService*→*TPM*:$ID_s||N_{12}||Sig_{Cert_S}(ID_s||N_{12})$,*TPM* 验证 *VMMService* 的签名:

$$Ver_{pub\_vmmService}(Sig_{Cert_S}(ID_s||N_{12}));$$

7. *TPM*→*CA*:$ID_{TPM}||K_{pub\_AIK}||ID_s||N_{13}||Sig_{Cert_{EK}}(ID_{TPM}||K_{pub\_AIK}||ID_s||N_{13})$,*CA* 验证 *EK* 证书对信息的签名:

$$Ver_{pub\_EK}(Sig_{Cert_{EK}}(ID_{TPM}||K_{pub\_AIK}||ID_s||N_{13}));$$

8. *CA*→*TPM*:$N_{13}||Sig_{pri\_CA}(N_{13})||Cert_{AIK}$,其中,$Cert_{AIK}=[K_{pub\_AIK}||ID_{TPM}||T_{11},Sig_{pri\_CA}(K_{pub\_AIK}||ID_{TPM}||T_{11})]$,*TPM* 验证 *CA* 对 N_{13} 的签名 $Ver_{pub\_CA}(Sig_{pri\_CA}(N_{13}))$,*TPM* 验证 *CA* 对 *AIK* 证书的签名:

$$Ver_{pub\_CA}(Sig_{pri\_CA}(K_{pub\_AIK}||ID_{TPM}||T_{11}));$$

9. *TPM*→*VMMService*:$PCRs||N_{12}||Sig_{Cert_{AIK}}(PCRs||N_{12})||Cert_{AIK}$,*VMMService* 验证 *AIK* 证书对信息的签名:

$$Ver_{pub\_AIK}(Sig_{Cert_{AIK}}(PCRs \| N_{12}));$$

10. *VMMService*→*Challenger*: $E_{K_{s1}}(PCRs \| Sig_{Cert\,AIK}(PCRs) \| Cert_{AIK} \| ID_S \| vmFlag==0 \| N_{11})$,用 K_{s1} 解密信息,*Challenger* 验证签名 $Ver_{pub\_AIK} \| (Sig_{Cert\,AIK}(PCRs))$.

下面,我们对上述每一步进行详细解释.具体如下.

步骤 1. *Challenger* 向 *CA* 发送请求信息,发送的具体内容有:*Challenger* 的标识符 ID_c,Y_{c1},其中,$Y_{c1}=a_1^{X_{c1}} \bmod q_1$,$X_{c1}$ 是 *Challenger* 的任选素数,$X_{c1}<q_1$;

步骤 2. *CA* 向 *VMMService* 发送 *Challenge* 标识符 ID_c、公开秘钥 Y_{c1},发送的具体内容有:ID_c,Y_{c1} 以及 *CA* 的签名 $Sig_{pri\_CA}(ID_c||Y_{c1})$,*VMMService* 收到信息后,验证 *CA* 对 *Challenger* 的签名 $Ver_{pub\_CA}(Sig_{pri\_CA}(ID_c||Y_{c1}))$,然后任选素数 X_{s1},且 $X_{s1}<q_1$,并计算 $K_{s1}=Y_{c1}^{X_{s1}} \bmod q_1$;

步骤 3. *VMMService* 向 *CA* 发送请求信息,发送的具体内容有:*VMMService* 标识符 ID_s,Y_{s1},其中:

$$Y_{s1}=a_1^{X_{s1}} \bmod q_1;$$

步骤 4. *CA* 向 *Challenger* 返回 *VMMService* 标识符 ID_s,Y_{s1} 以及 *CA* 的签名 $Sig_{pri\_CA}(ID_s||Y_{s1})$,*Challenger* 验证 *CA* 对 *VMMService* 的签名 $Ver_{pub\_CA}(Sig_{pri\_CA}(ID_s||Y_{s1}))$,并计算 $K_{s1}=Y_{s1}^{X_{c1}} \bmod q_1$;

经过前面 4 步,*Challenger* 和 *VMMService* 之间获得一个共享的秘密密钥 K_{s1}.这 4 步实际上是能防止中间人攻击的 Diffle-Hellman 算法.

步骤 5. *Challenger* 用秘钥 K_{s1} 加密向 *VMMService* 发送的远程认证请求,加密的具体内容为:*Challenger* 标识符 ID_c、现时 N_{11},*VMMService* 用 K_{s1} 解密获得信息内容;

步骤 6. *VMMService* 向 *TPM* 发出可信证据信息请求,发送的具体内容有:*VMMService* 标识符 ID_s、现时 N_{12} 及 *VMMService* 对信息的签名 $Sig_{Cert_S}(ID_s||N_{12})$,*TPM* 收到信息后,验证 *VMMService* 的签名:

$$Ver_{pub\_vmmService}(Sig_{Cert_S}(ID_s||N_{12}));$$

步骤 7. *TPM* 向 *CA* 中心发送 *AIK* 证书请求,发送的具体内容有:*TPM* 标识符 ID_{TPM}、*AIK* 公钥 K_{pub_AIK}、*VMMService* 标识符 ID_s、现时 N_{13} 及用 *EK* 证书对信息的签名 $Sig_{Cert_{EK}}(ID_{TPM}||K_{pub\_AIK}||ID_s||N_{13})$,*CA* 验证 EK 证书签名的消息 $Ver_{pub\_EK}(Sig_{Cert_{EK}}(ID_{TPM}||K_{pub\_AIK}||ID_s||N_{13}))$;

步骤 8. *CA* 为 *TPM* 生成 *AIK* 证书,证书的具体内容有:ID_{TPM}、K_{pub_AIK}、时间戳 T_1 以及 *CA* 的签名 $Sig_{pri\_CA}(K_{pub\_AIK}||ID_{TPM}||T_{11})$,*CA* 中心向 *TPM* 返回信息,信息的具体内容为:现时 N_{13}、*CA* 的签名 $Sig_{pri_CA}(N_{13})$和 *AIK* 证书 $Cert_{AIK}$,*TPM* 验证 *CA* 对 *AIK* 证书的签名 $Ver_{pub_CA}(Sig_{pri_CA}(K_{pub_AIK}||ID_{TPM}||T_{11}))$和 *CA* 对现时的签名 $Ver_{pub\_CA}(Sig_{pri\_CA}(N_{13}))$;

步骤 9. *TPM* 把收到的现时 N_{13} 和 *TPM* 之前向 *CA* 发送的现时 N_{13} 进行匹配,如果匹配成功,发送可信证据给 *VMMService*,发送的具体内容有:*PCRs*、现时 N_{12} 和 *AIK* 证书及 *AIK* 证书对 *PCRs* 的签名 $Sig_{Cert_{AIK}}(PCRs||N_{12})$,*VMMService* 验证 *TPM* 对信息的签名 $Ver_{pub\_AIK}(Sig_{Cert\,AIK}(PCRs \| N_{12}))$;

步骤 10. *VMMService* 收到信息后,把收到的现时 N_{12} 和 *VMMService* 之前向 *TPM* 发送的现时 N_{12} 进行匹配,如果匹配成功,就用通信秘钥 K_{s1} 加密可信证据信息发送给 *Challenger*,加密的可信证据信息内容有:*PCRs* 及 $Sig_{Cert\,AIK}(PCRs)$、*AIK* 证书、*VMMService* 标识符 ID_s、值为 0 的虚拟机标识符 *vmFlag*、现时 N_{11} 等,*Challenger* 用 K_{s1} 解密得到信息.

第 2 部分是 *Challenger* 接收到 *VMMService* 的返回值,首先,把收到的现时 N_{11} 和 *Challenger* 之前向 *VMMService* 发送的现时 N_{11} 进行匹配,如果匹配成功,则 *Challenger* 验证 *CA* 对 *AIK* 证书的签名 $Ver_{pub\_CA}(Sig_{pri\_CA}(K_{pub\_AIK}||ID_{TPM}||T_{11}))$,再验证 *AIK* 证书对 *PCRs* 的签名 $Ver_{pub\_AIK} \| (Sig_{Cert\,AIK}(PCRs))$.此时,挑战者对 *PCRs* 进行验证,从而可以进行验证运行于物理平台之上的虚拟机管理器是否可信.

至此,Challenger 已经完成对虚拟机和运行于物理平台之上的虚拟机管理器的认证.

3 TVP-PCA 方案的可信判定

可信虚拟平台的远程证明方案的核心是判断问题.判定依赖于具体的策略,该策略必须与虚拟终端实际环

境相吻合.因此,必须详细分析虚拟机的信任链、虚拟机管理器的信任链,为远程证明的判定提供基础保证.为了方便文章叙述,我们先形式化定义软件组件、信任根集等基本概念.

定义 1. 可信虚拟平台的软件组件定义为 $C=\{C_1,C_2\}$,其中,C_1 表示顶层虚拟机的软件组件集,C_2 表示底层运行于物理硬件上的 *VMM* 及管理域的软件组件集.

我们用 C_{1i} 表示底层虚拟机的软件组件:$C_{1i}=\{(c_{1i\text{-}code},c_{1i\text{-}d\text{-}code},c_{1i\text{-}cofile}),(c_{1i\text{-}f},c_{1i\text{-}v},c_{1i\text{-}o})\}$.其中,

- $(c_{1i\text{-}code},c_{1i\text{-}d\text{-}code},c_{1i\text{-}cofile})$表示 C_{1i} 的代码特征:$c_{1i\text{-}code}$ 代表 C_{1i} 的源代码;$c_{1i\text{-}d\text{-}code}$ 代表 C_{1i} 的依赖,如其依赖的静态库、动态库或第三方代码库;$c_{1i\text{-}cofile}$ 代表 C_{1i} 的策略配置文件;
- $(c_{1i\text{-}f},c_{1i\text{-}v},c_{1i\text{-}o})$表示 C_{1i} 的功能特征:$c_{1i\text{-}f}$ 表示 C_{1i} 的功能;$c_{1i\text{-}v}$ 表示测 C_{1i} 的版本号;$c_{1i\text{-}o}$ 表示 C_{1i} 的其他相关属性,比如软件名、开发者、发布时间等.

用 C_{2i} 表示 *VMM* 及管理域的软件组件:$C_{2i}=\{(c_{2i\text{-}code},c_{2i\text{-}d\text{-}code},c_{2i\text{-}cofile}),(c_{2i\text{-}f},c_{2i\text{-}v},c_{2i\text{-}o})\}$.其中,

- $(c_{2i\text{-}code},c_{2i\text{-}d\text{-}code},c_{2i\text{-}cofile})$表示 *VMM* 及管理域软件 C_{2i} 的代码特征:$c_{2i\text{-}code}$ 代表 C_{2i} 的源代码;$c_{2i\text{-}d\text{-}code}$ 代表 C_{2i} 的依赖,如其依赖的静态库、动态库或第三方代码库;$c_{2i\text{-}cofile}$ 代表 C_{2i} 的策略配置文件;
- $(c_{2i\text{-}f},c_{2i\text{-}v},c_{2i\text{-}o})$表示 C_{2i} 的功能特征:$c_{2i\text{-}f}$ 表示 C_{2i} 的功能,$c_{2i\text{-}v}$ 表示测组件 C_{2i} 的版本号,$c_{2i\text{-}o}$ 表示软件组件 C_{2i} 的其他相关属性.

依据定义 1,顶层虚拟机的软件组件集为 $C_1=\{c_{11},c_{12},\ldots,c_{1n},\ldots\}$,*VMM* 及管理域的软件组件集为

$$C_2=\{c_{11},c_{22},\ldots,c_{2n},\ldots\}.$$

定义 2. 信任根集(trustedroot set)定义为 $TRS=\{TRS_1,TRS_2\}$,其中,TRS_1 是顶层虚拟机的信任根集,TRS_2 是底层运行于物理硬件上的 *VMM* 及管理域的信任根集.信任根集为一个由特殊属性组成的集合,包含所有其可信性无需进行证明的属性,所以,$TRS_1=\{vTPM\}$,$TRS_2=\{TPM\}$.

定义 3. 完整性测量组件集定义为 $I=\{I_1,I_2\}$,其中,

- I_1 是顶层虚拟机的完整性测量组件集,$I_1=\{i_{11},i_{12},\ldots,i_{1n},\ldots\}$,其中,$i_{1n}\in C_1$;
- I_2 是底层运行于物理硬件上的 *VMM* 及管理域的完整性测量组件集,$I_2=\{i_{21},i_{22},\ldots,i_{2n},\ldots\}$,其中,$i_{2n}\in C_2$.

定义 4. $Measure(i_n,i_{n+1},PCR_{n+1},SML_{n+1})$表示在完整性测量过程中,$i_n$ 对 i_{n+1} 的完整性进行测量,其增量哈希值存储于 PCR_{n+1},相对应的存储测量值日志为 SML_{n+1},n 是正整数且 $0\leqslant n\leqslant 23$.当顶层虚拟机进行完整性测量时,将哈希值存储在 *vPCR* 中,对应的测量值日志存在 *vSML* 中,当底层运行于物理硬件上的 *VMM* 及管理域进行完整性测量时,将哈希值存储在 *PCR* 中,对应的测量值日志存在 *SML* 中.

定义 5. 完整性验证函数 $Verify(i_n,i_{n+1},PCR_{n+1},LPCR_{n+1})$表示将 i_n 对 i_{n+1} 的完整性测量值 PCR_{n+1} 与可信策略库中对应的标准 $LPCR_{n+1}$ 进行比较.当顶层虚拟机进行完整性验证时,将 *PCR* 中的值和 *LPCR* 中的值对比:如果完全匹配,返回 TRUE;否则,返回 FALSE.类似的,当底层运行于物理硬件上的 *VMM* 及管理域进行完整性验证时,将 *vPCR* 中的值和 *vLPCR* 中的值对比:如果完全匹配,返回 TRUE;否则,返回 FALSE.

定义 6. 完整性传递函数 $Integrity(i_n,i_{n+1})$表示完整性能够从 i_n 有效地传递至 i_{n+1},而不遭受破坏与损失.当顶层虚拟机进行完整性传递时,将从 i_{1n} 有效地传递至 i_{1n+1},而不遭受破坏与损失;当底层运行于物理硬件上的 *VMM* 及管理域进行完整性验证时,将从 i_{2n} 有效地传递至 i_{2n+1},而不遭受破坏与损失.

3.1 顶层证明的可信判定

顶层证明是证明运行于虚拟机管理器之上的虚拟机的可信,实质是指虚拟机满足基于完整性测量的虚拟机信任链传递.为了保证 *VMAgent* 可信,我们扩展虚拟机的信任传递过程为:*INIT*→*vBIOS*→*VMOS Loader*→*VMOS*→*VMAgent*→*Application*.如图 6 所示,该信任链将 *VMAgent* 作为可信平台链式度量的重要一环.这种方式是可行的,*VMAgent* 的度量可由 *VMOS* 主导完成,并由 *VMOS* 将 *VMAgent* 程序作为第 1 个应用程序首先启动.

在图 6 中,实现信任传递的相关参数主要包括:

(1) 系统配置:将完整性测量组件的哈希值存入 *vTPM* 的 24 个 *vPCR* 中;

(2) 虚拟机存储测量值日志(VM storage measurement log,简称 vSML),其中包含了存储在 *vTPM* 中的所有测量值的事件结构以及被测量的软件组件的$(c_{1i\text{-}f},c_{1i\text{-}v},c_{1i\text{-}o})$.

由图 6 可知,虚拟机完整性测量组件集 I_1={*INIT*,*vBIOS*,*VMOSLoader*,*VMOS Kernel*,*VMAgent*,*Applications*},显然,$I_1 \subset C_1$.

根据以上分析,我们得出顶层虚拟机的判定算法,如图 7 所示.图 7 算法首先判断信任根集是不是空,然后判断信任根集里 TRS_1 信任根是不是唯一.如果这两个条件均成立,则调用 *Measure* 函数对 *INIT*→*vBIOS*→*VMOS Loader*→*VMOS*→*VMAgent*→*Application* 进行完整性测量,并调用完整性验证函数 *Verify* 进行验证,如果成立,则调用 *Integrity* 进行完整性可信传递:如果最后完整性验证都通过,则输出 TRUE;否则,输出 FALSE.

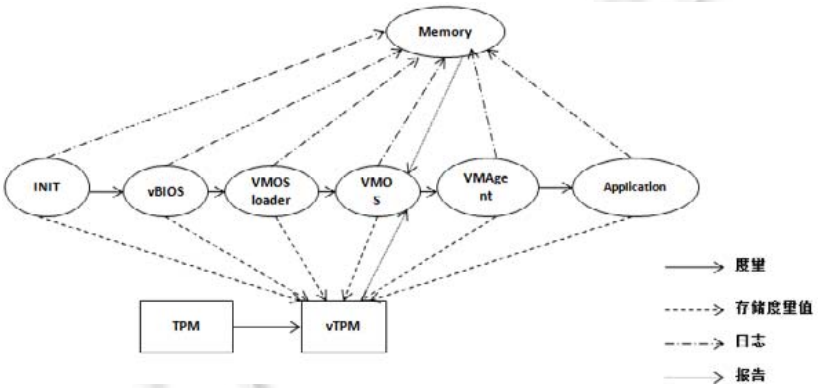


Fig.6 Measurement of attestation agent by *vTPM*

图 6 基于 *vTPM* 的证明代理的可信度量

算法 1. 顶层证明的可信判定算法 *VM_Trusted*(I_1).

输入:I_1;

输出:TRUE 或 FALSE.

```
1.IF (TRS==NULL) OR (|TRS|==0) THEN return FALSE END IF;   //|I1|表示集合 I1 中元素的个数
2. IF ((TRS1 in TRS) AND (|TRS1|==1) AND (INIT in TRS1)) THEN   //判断信任根是否存在且唯一
3.    {
        i11=INIT;
4.       FOR n=1 to |I1n| DO
5.             Measure(i1n,i1n+1,vPCRn+1,vSMLn+1);
6.             IF (Verify(i1n,i1n+1,vPCRn+1,LPCRn+1)==TRUE) THEN
7.                 {Integrity(i1n,i1n+1);
RETURN TRUE;}
8.             ELSE
9.                 RETURN FALSE;
10.          END IF
11.       END FOR
12.   }
13.END IF
```

Fig.7 Decidability algorithm of the top-level attestation

图 7 顶层证明的可信判定算法

3.2 底层证明的可信判定

底层证明是证明虚拟机管理器的可信,实质是指运行在物理设备之上的虚拟机管理器满足基于完整性测量的虚拟机管理器的信任链传递.在底层证明过程中,虚拟机证明代理 *VMMService* 运行在虚拟机管理器之上,起着非常重要的作用.为了保证 *VMMService* 可信,我们扩展虚拟机的信任传递过程为 *CRTM*→*BIOS*→*OS Loader*→*OS*→*VMMService*→*Application*.如图 8 所示,将 *VMMService* 作为可信平台链式度量的重要一环.这种方式是可行的,*VMMService* 的度量可由 *OS* 主导完成,并由 *OS* 将 *VMMService* 程序作为第 1 个应用程序首先启动.

在图 8 中,实现信任传递的相关参数主要包括:

(1) 系统配置:将完整性测量组件的哈希值存入 *TPM* 的 24 个 *PCR* 中;

(2) 存储测量值日志(storage measurement log,简称 SML),其中包含了存储在 *TPM* 中的所有测量值的事件结构以及被测量的软件组件的($c_{2i\text{-}f}$,$c_{2i\text{-}v}$,$c_{2i\text{-}o}$).

由图 8 可知,虚拟机管理器的完整性测量组件集 I_2={*CRTM*,*BIOS*,*OS Loader*,*OS Kernel*,*VMMService*,*Applications*}.显然,$I_2 \subset C_2$.

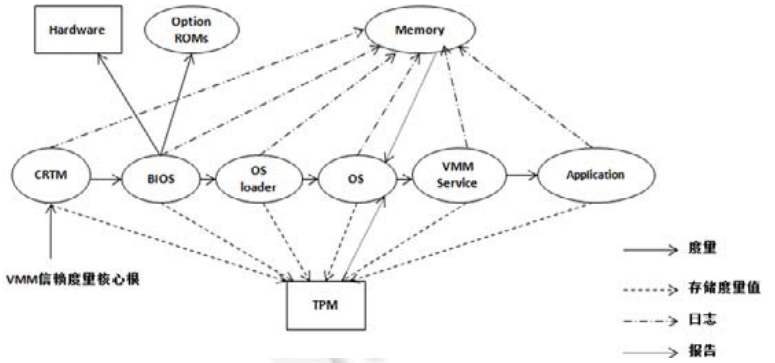


Fig.8 Measurement of attestation service by *TPM*

图 8 基于 *TPM* 的证明服务的可信度量

根据以上分析,我们得出底层虚拟机管理器的判定算法,如图 9 所示.图 9 算法首先判断信任根集是不是空,然后判断信任根集里 TRS_2 信任根是不是唯一.如果这两个条件均成立,则调用 *measure* 函数对 *CRTM*→*BIOS*→*OS Loader*→*OS*→*VMMService*→*Application* 进行完整性测量,并调用完整性验证函数 *Verify* 进行验证,如果成立,则调用 *Integrity* 进行完整性可信传递:如果最后完整性验证都通过,则输出 TRUE;否则,输出 FALSE.

算法 2. 底层证明的可信判定算法 *VMM_Trusted*(I_2).
输入:I_2;
输出:TRUE 或 FALSE.
1.IF (*TRS*==*NULL*) OR (|*TRS*|==0) THEN return FALSE END IF; //|I_2|表示集合 I_2 中元素的个数
2. IF ((TRS_2 in *TRS*) AND (|TRS_2|==1) AND (*CRTM* in TRS_2)) THEN //判断信任根是否存在且唯一
3. {
I_{21}=*CRTM*;
4. FOR *n*=1 to |I_{2n}| DO
5. *Measure*(i_{2n},i_{2n+1},$vPCR_{n+1}$,$vSML_{n+1}$);
6. IF (*Verify*(i_{2n},i_{2n+1},$vPCR_{n+1}$,$LPCR_{n+1}$)==TRUE) THEN
7. {*Integrity*(i_{2n},i_{2n+1});
RETURN TRUE;}
8. ELSE
9. RETURN FALSE;
10. END IF
11. END FOR
12. }
13.END IF

Fig.9 Decidability algorithm of the underlying attestation

图 9 底层证明的可信判定算法

3.3 同一性可信判定

按照本文第 2 节设计的证明方案,要证明整个虚拟平台的可信,需要首先证明顶层虚拟机的可信,然后证明底层物理平台上的虚拟机管理器的可信.在两个相对独立的逻辑运行实体上分阶段证明容易造成同一性问题,即,第 1 阶段的证明和第 2 阶段的证明是否是同一虚拟平台或同一物理平台.TCG 虚拟化平台工作组在规范《Virtualized Trusted Platform Architecture Specification》1.0 版中也明确指出这一问题.

定义 7. 可信虚拟平台远程证明的同一性问题,是指顶层可信虚拟机的远程证明和底层可信虚拟机管理器的远程证明是否属于同一虚拟平台或同一物理平台.

同一性问题在可信虚拟平台远程证明中确实可能存在,因为按照 TVP-PCA 方案,当挑战者完成与顶层虚拟机交互证明后,可能会获得下一阶段的伪冒 *VMMService* 域名 $URL_{vmmService}$.在此,我们可以假设如下 3 种情况.

情况 1:有两个虚拟平台 1 和 2,均具有物理 *TPM*,虚拟机均含有 *vTPM*.当挑战者按照 TVP-PCA 方案在验证虚拟平台 1 时,虚拟平台 1 的 *VMAgent* 故意返回虚拟平台 2 的 *VMMService* 域名 $URL_{vmmService}$.挑战者完成两阶段交互证明.

对于情况 1,我们认为是不必要考虑的.因为虚拟平台 1 本身就是可信的,没有必要借助虚拟平台 2 来证明自己可信.

情况 2:有两个虚拟平台 1 和 2,虚拟平台 1 无物理 *TPM*,但其虚拟机均含有 *vTPM*;虚拟平台 2 具有物理 *TPM*,其虚拟机也含有 *vTPM*.当挑战者按照 TVP-PCA 方案在验证虚拟平台 1 时,虚拟平台 1 的 *VMAgent* 返回的是虚拟平台 2 的 *VMMService* 域名 $URL_{vmmService}$.挑战者也能顺利完成两阶段交互证明.具体如图 10 所示.

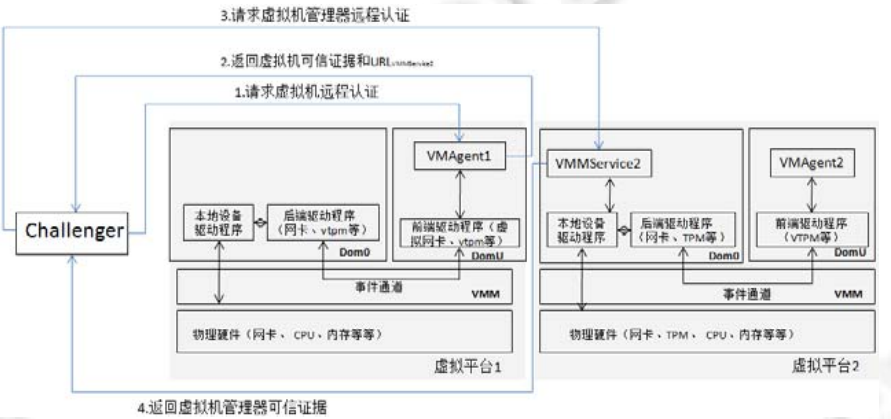


Fig.10 Process of the untrusted virtual platform remote attestation (1)

图 10 不可信虚拟平台远程证明过程(1)

为了防止这两种情况的发生,我们提出信任链判定法.所谓信任链判定法就是通过在底层信任链和顶层信任链之间建立度量联系来判定同一性问题.根据图 6 和图 8,我们定义两条信任链的度量联系如下.

定义 8. 度量联系是指底层信任链的最后一个 *PCR* 值扩展到顶层信任链,即满足等式:

$$vPCR[0]==Hash(PCR[last]||INIT),$$

其中,*vPCR*[0]是顶层虚拟机信任链的第 1 个度量值,*PCR*[*last*]是底层信任链的最后一个度量值,*INIT* 是顶层虚拟机的第 1 个被度量的组件.

根据以上分析,我们得出同一性问题的信任链判定算法,如图 11 所示.

值得注意的是,信任链判定算法只是一个必要条件,即:*Trusted_Chain*(*PCR*[*last*],*INIT*)返回 FALSE,则顶层证明和底层证明一定存在同一性问题;反之则不成立.下列的情况 3 可以说明这一点.

情况 3:有两个虚拟平台 1 和 2,虚拟平台 1 无物理 *TPM*,但其虚拟机均含有 *vTPM*;虚拟平台 2 具有物理 *TPM*,其虚拟机含有 *vTPM*.为了欺骗挑战者,虚拟平台 1 事先作为挑战者对虚拟平台 2 进行验证,获得虚拟平台 2 的 *PCR* 和域名,并按度量联系重构顶层虚拟机信任链.之后,虚拟平台 1 重复情况 2 的过程.挑战者也能顺利完成两阶段交互证明,并成功通过信任链判定算法的检测.具体如图 12 所示.

算法 3. 信任链判定算法 *Trusted_Chain*(*PCR*[*last*],*INIT*).
输入:*PCR*[*last*],*INIT*;
输出:TRUE 或 FALSE.
1.IF (*PCR*[*last*]==*NULL*) OR (*PCR*[*last*]==0) THEN return FALSE END IF; //判断 PCR[last]是否空
2. IF ((*INIT* in I_1) AND ((*INIT*==I_1[0]) THEN //判段 *INIT* 是否存在并使顶层虚拟机度量的第 1 个组件
3. {
5. *Measure*(*PCR*[*last*],*INIT*,*vPCR*[0],$vSML_0$);
6. IF (*Verify*(*PCR*[*last*],*INIT*,*vPCR*[0],$LPCR_0$)==TRUE) THEN
7. {*Integrity*(*PCR*[*last*],*INIT*);
RETURN TRUE;}
8. ELSE
9. RETURN FALSE;
10. ENDIF
11. }
12. ENDIF

Fig.11 Decidability algorithm of the trusted chain

图 11 信任链判定算法

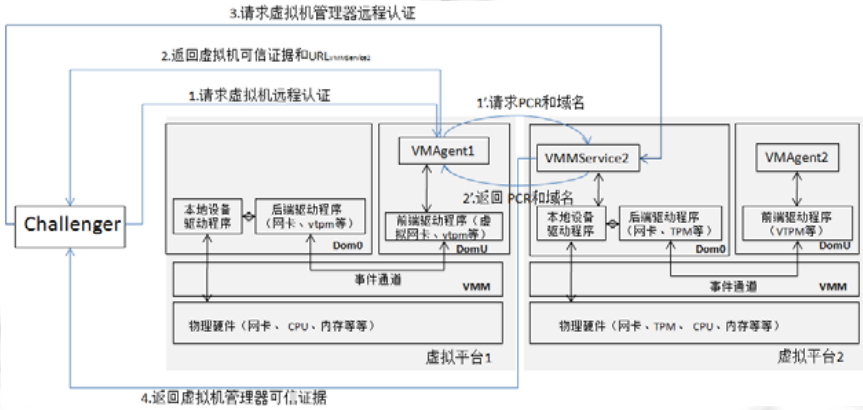


Fig.12 Process of the untrusted virtual platform remote attestation (2)

图 12 不可信虚拟平台远程证明过程(2)

为了防止这种情况的发生,我们提出 MAC 地址判定法.所谓 MAC 地址判定法,就是通过在底层交互证明和顶层交互证明中传递数据包的 MAC 地址是否相同来判定同一性问题.从图 12 可以看出:*Challenger* 在进行顶层证明时的数据来源于虚拟平台 1,在进行底层证明时的数据来源于虚拟平台 2,两阶段接收的数据包的 MAC 地址一定是不同的.了便于叙述,我们定义 MAC 解析函数.

定义 9. *MAC_Parser* 是 MAC 地址解析函数,其函数原型为 *mac_address MAC_Parser*(*send*,*recieve*),其中,函数名是 *MAC_Parser*,*mac_address* 是返回类型.该函数需要两个输入参数:其一,*send* 代表发送方;其二,*receive* 代表接受方.

根据以上分析,我们得出同一性问题的 MAC 地址判定算法,如图 13 所示.

算法 4. *MAC* 地址判定算法 *MAC_CHECK*(*Challenger*,*vmAgent*,*VMMService*).
输入:*Challenger*,*vmAgent*,*VMMService*;
输出:TRUE 或 FALSE.
1. IF (*Challenger*==*NULL*) AND (*vmAgent*==*NULL*) AND (*VMMService*==*NULL*) THEN
return FALSE END IF; //判断 *Challenger*,*vmAgent*,*VMMService* 是否为空
2. IF (*MAC_Parser*(*VMAgent*,*Challenger*)==*MAC_Parser*(*VMMService*,*Challenger*))==TRUE) THEN
3. reture TRUE;
4. ELSE
5. reture FALSE;
6. END IF

Fig.13 Decidability algorithm of MAC address

图 13 MAC 地址判定算法

值得注意的是,我们之所以没有采用两阶段接收数据包的 IP 地址来辨析情况 3,主要包括两方面原因:其一是虚拟平台的网络地址模式比较多,容易发生混淆;其二是通过 IP 地址无法判断两个阶段接收的数据是否来自同一虚拟平台.

3.4 TVP-PCA方案的可信判定算法

根据顶层证明的可信判定、底层证明的可信判定和同一性可信判定,我们得出 TVP-PCA 方案总的可信判定算法.如图 14 所示.

算法 5. *TVP-PCA* 可信判定算法 *TVP-PCA_Whole_Trusted_CHECK*(I_1,I_2,*PCR*[*last*],*INIT*,*Challenger*,*vmAgent*,*VMMService*).
输入:I_1,I_2,*PCR*[*last*],*INIT*,*Challenger*,*vmAgent*,*VMMService*;
输出:TRUE 或 FALSE.

```
1.  IF (VM_Trusted(I1)==FALSE) THEN return FALSE;   //判定顶层证明是否可信
2.  ELSE IF (VMM_Trusted(I2)==FALSE) THEN return FALSE;   //判定底层证明是否可信
3.  ELSE IF(Trusted_Chain(PCR[last],INIT)==FALSE) THEN return FALSE;   //判定同一性问题
4.  ELSE IF(MAC_CHECK(Challenger,vmAgent,VMMService)==FALSE) THEN return FALSE;
5.  ELSE reture TRUE;
```

Fig.14 Decidability algorithm of TVP-PCA

图 14 TVP-PCA 可信判定算法

值得注意的是:其实无论是情况 2 还是情况 3,我们都可以直接采用 MAC 地址判定算法就能判定同一性问题.但由于信任链判判定算法的效率比 MAC 地址判别法的效率高,所以在算法 5 中,我们仍然先采用信任链判判定算法对同一性问题进行判定.如果 *Challenger* 在两阶段接收到的数据不满足信任链判判定算法,直接返回 FALSE,就没有必要再采用 MAC 地址判定算法进行判定了,可以提高算法 5 的效率.

4 TVP-PCA 的特点和安全性分析

4.1 TVP-PCA的特点分析

(1) 一致性.对于虚拟平台的远程认证,TCG 虚拟化平台工作组已发布了《Virtualized Trusted Platform Architecture Specification》1.0 版中的认证思想是:通过两次认证,包括顶层虚拟机认证阶段和底层虚拟机管理器和物理平台认证阶段.本文提出的 TVP-PCA 方案与 TCG 提出的方法具有一致性.

(2) 灵活性.如果挑战者请求虚拟机上的资源或服务,需要先对虚拟机进行认证,再对虚拟机管理器和物理平台进行认证;如果挑战者请求虚拟机管理器上的资源或服务,则可以只对虚拟机管理器和物理平台进行认证.

4.2 TVP-PCA的安全性分析

我们提出的方案 TVP-PCA 具有机密性、抗中间人攻击和抗重放攻击,因而具有较高的安全性.对于本文提出的方法进行安全性分析,具有以下 4 点.

(1) 具有机密性.

在顶层虚拟机远程认证阶段,*Challenger* 和 *VMAgent* 之间的通信,无论是 *Challenger* 向 *VMAgent* 发送远程证明请求信息,还是 *VMAgent* 返回可信证据,都用对称秘密密钥 K_{s0} 加密,如果信息被截取,攻击者会由于没有对称秘密密钥 K_{s0} 而无法获得信息的具体内容,由此确保 *Challenger* 和 *VMAgent* 的通信安全.在底层远程认证阶段,*Challenger* 和 *VMMService* 之间的通信,无论是 *Challenger* 向 *VMMService* 发送远程证明请求信息,还是 *VMMService* 返回可信证据,都用对称秘密密钥 K_{s1} 加密,如果信息被截取,攻击者会由于没有对称秘密密钥 K_{s1} 而无法获得信息的具体内容,由此确保 *Challenger* 和 *VMMService* 的通信安全.

(2) 具有抗中间人攻击.

中间人攻击是一种通过修改或者伪装发送消息而达到攻击目的的手段.首先,在顶层远程认证阶段对称加密密钥 K_{s0} 的产生,我们采用的是抗中间人攻击的 DH 协议,*Challenger* 生成对称加密密钥 K_{s0} 所需要的 Y_{c0} 以及 *VMAgent* 生成对称加密密钥 K_{s0} 所需要的 Y_{v0} 均是通过 *CA* 签名交互,中间人无法伪造和替换;其次,在底层远程认证阶段对称加密密钥 K_{s1} 的产生,我们同样采用的是抗中间人攻击的 DH 协议,*Challenger* 生成对称加密密钥

K_{s1}所需要的 Y_{c1}以及 *VMMService* 生成对称加密密钥 K_{s1}所需要的 Y_{s1}均是通过 *CA* 签名交互,中间人同样无法伪造和替换;第三,*Challenger*,*VMAgent*,*vTPM*,*VMMService*,*TPM* 和 *CA* 等之间通信都是经过证书签名或加密的,证书具有身份认证功能,中间人无法伪造.

(3) 具有抗重放攻击.

在顶层认证阶段,*Challenger* 向 *VMAgent* 发送信息中包含现时 N_{01},*VMAgent* 返回的信息中也必须包含 N_{01},所以 *Challenger* 可以通过对比 N_{01}来确保不是重放信息;同样,*VMAgent* 向 *vTPM* 发送信息中包含现时 N_{02},*vTPM* 返回的信息中也包含 N_{02},*VMAgent* 通过对比 N_{02}来确保不是重放信息;*vTPM* 向 *CA* 发送信息中包含现时 N_{03},*CA* 返回的信息中也包含 N_{03},*vTPM* 通过对比 N_{03}来确保不是重放信息.所以在顶层认证阶段是可以抗重放攻击的.在底层远程认证阶段,*Challenger* 向 *VMMService* 发送信息中包含现时 N_{11},*VMMService* 返回的信息中也必须包含 N_{11},所以 *Challenger* 可以通过对比 N_{11}来确保不是重放信息;同样,*VMMService* 向 *TPM* 发送信息中包含现时 N_{12},*vTPM* 返回的信息中也包含 N_{12},*VMMService* 通过对比 N_{12}来确保不是重放信息;*TPM* 向 *CA* 发送信息中包含现时 N_{13},*CA* 返回的信息中也包含 N_{13},*TPM* 通过对比 N_{13}来确保不是重放信息.所以在底层认证阶段是可以抗重放攻击的.综上,可以确保整个 TVP-PCA 方案的可以抗重放攻击的.

(4) 具有同一性.

本方案最大的特点之一,就是解决了可信虚拟平台远程认证的同一性问题.我们通过信任链判定算法和 *MAC* 地址判定算法确定顶层远程证明和底层远程证明是否属于同一虚拟平台或同一物理平台.

5 基于 XEN 环境的 TVP-PCA 实验原型分析

该实验一共使用 3 台计算机.一台为普通 PC 机,CPU 是 Inter(R) Core(TM)2 Duo CPU E7500@2.93GHz,内存 4G,虚拟机管理器是 Xen 4.1.6.1,Dom0 操作系统为 Ubuntu 12.04.1LTS(内核 3.2.0.29),DomU 操作系统为 Ubuntu 12.04.1LTS(内核 3.2.0.29).在 Dom0 上运行虚拟机管理器认证服务 *VMMService*,在 DomU 上运行虚拟机认证代理 *VMAgent*,*TPM* 芯片用 *TPM* 模拟器代替,具体为 tpm_emulator-0.7.4.另外还有一台普通 PC 机做为挑战者 *Challenger*,CPU 是 Inter(R) Core(TM)2 Duo CPU E7500@2.93GHz,内存 2G,操作系统为 Microsoft Windows 7 SP1.最后还有一台为服务器,做为 *CA* 证书中心,CPU 是 Inter(R) Core(TM)2 Duo CPU E7500@2.93GHz,内存 2G,操作系统为 Windows Server 2008.具体如图 15 所示.挑战者需要通过 *VMAgent* 提供的可信证据来证明 DomU 的可信,通过 *VMMService* 提供的证据是证明 *VMM* 和 Physical platform 的可信,从而确定整个虚拟平台的可信.在图 15 中:*Challenger* 是一个用 java 实现的代理进程,可跨平台运行,满足挑战者的多样性;*VMAgent* 是一个基于 vmlinuz 的 DomU 内的守护进程,配置的监听 2020 端口;*VMMService* 是一个基于 vmlinuz 的 Dom0 内的守护进程,配置的监听 2021 端口;*vTPM* 是一个模拟实现,采用的是 TPM-emulator 0.7.4.所有其他实体与 *vTPM* 交互均通过 *vTPM* manager,这一点我们沿用 Xen 的方式.*CA* 是用 openssl 模拟实现的,配置的监听 2022 端口.鉴于篇幅,我们将另外撰文介绍 *Challenger*,*VMAgent* 和 *VMMService* 的设计与实现.以下的实验结果均基于我们实现的 *Challenger*,*VMAgent* 和 *VMMService* 原型.

5.1 实验结果

VMAgent 和 *VMMService* 先按照本文 3.1 中介绍的方法进行初始化,*Challenger* 远程认证虚拟平台时,先按照第 3.2 节中介绍的方法通过 *VMAgent* 远程认证 DomU,再按照第 3.3 节中介绍的方法通过 *VMMService* 远程认证 *VMM* 和 *PhysicalPlatform*.特别说明的是,在顶层证明阶段 *Challenger* 和 *VMAgent* 双方选择的乘法群 Z_q^*的素数 q 以及在底层证明阶段 *Challenger* 和 *VMMService* 双方选择的乘法群 Z_q^*的素数 q 的大小很关键,太小不安全,太大又影响性能.根据参考文献[41,42],q 的范围可为 $150\leqslant|q|\leqslant180$,$|q|$表示 q 的十进制位数.本文在编程实现时,q均取为 150 位的十进制数且 $q-1$ 必须有一个大素数因子.X_{c0}和 X_{c1}比 q 略小,而且 q,X_{c0}和 X_{c1}用 Miller-Rabin 算法测试进行了素性测试[43].如图 16 为虚拟机认证代理 *VMAgent* 初始化运行结果图,图 17 为虚拟机管理器认证服务 *VMMService* 初始化运行结果图.

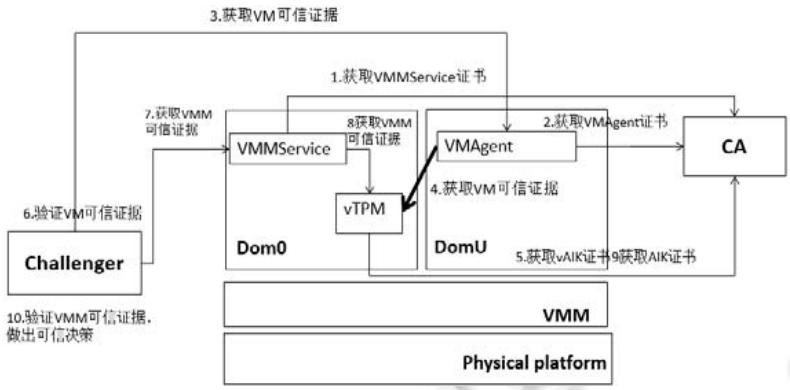


Fig.15 Architecture of experiment of trusted virtual platform remote attestation

图 15 可信虚拟平台远程认证实验架构图

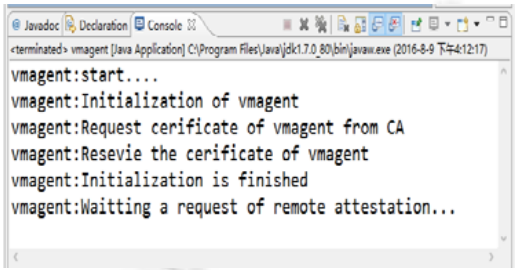


Fig.16 Initialization of *VMAgent*

图 16 虚拟机认证代理初始化图

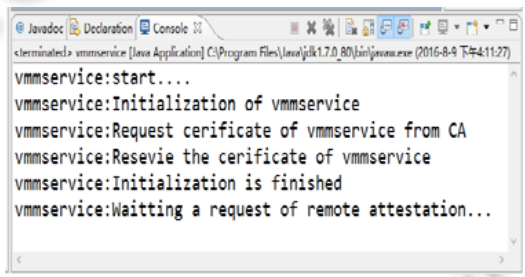


Fig.17 Initialization of *VMMservice*

图 17 虚拟机管理器认证服务初始化

顶层远程认证阶段,*Challenger* 和 *VMAgent* 通过交互来认证虚拟机的可信,图 18 为 *Challenger* 运行结果图,图 19 为 *VMAgent* 运行结果图.

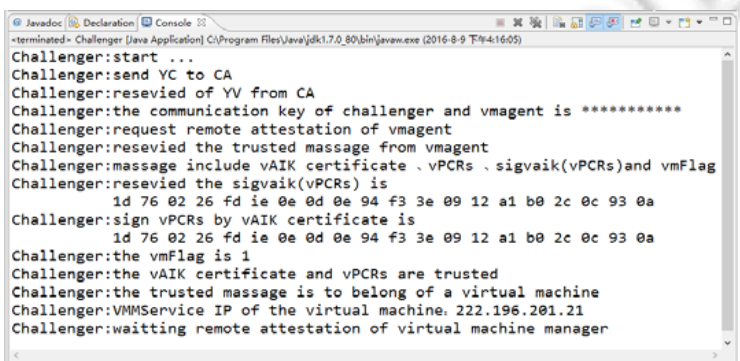


Fig.18 Run result of *Challenger* for remote attestation to *VM*

图 18 挑战者远程认证虚拟机的运行结果图

Fig.19 Run result of *VMAgent*

图 19 虚拟机认证代理运行结果图

底层远程认证阶段,*Challenger* 和 *VMMService* 通过交互来认证运行于物理平台的虚拟机管理器的可信,图 20 为挑战者运行结果图,图 21 为虚拟机管理器认证服务运行结果图.

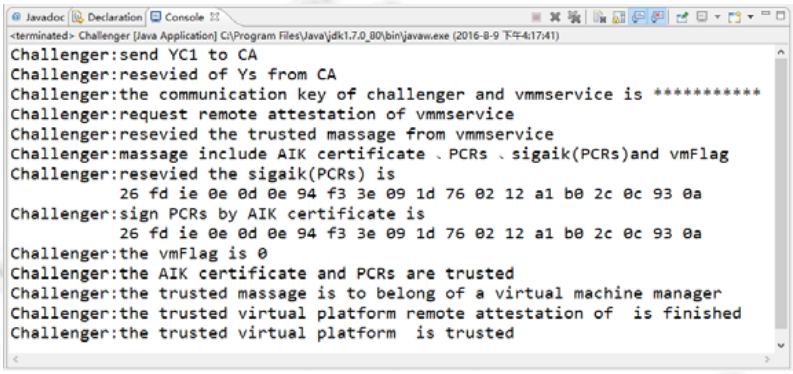


Fig.20 Run result of *Challenger* remote attestation the virtual machine manager

图 20 挑战者远程认证虚拟机管理器的运行结果图

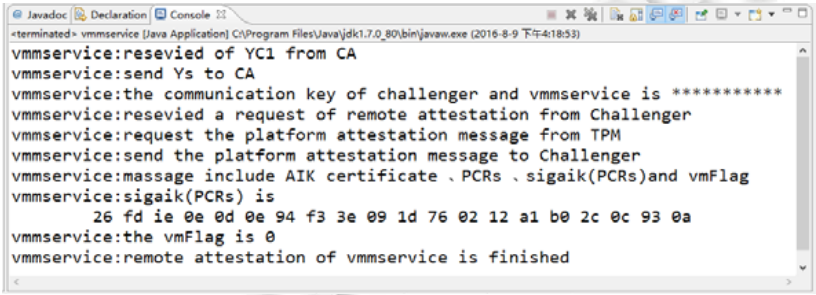


Fig.21 Run result of *VMMservice*

图 21 虚拟机管理器认证服务运行结果图

Challenger 通过和 *VMAgent*,*VMMService* 的远程证明过程,已经获得了虚拟机和运行在物理平台之上的虚拟机管理器的可信证据.最后,*Challenger* 通过可信判定算法判定虚拟平台是否可信.图 22 为 *Challenger* 的判定结果图.

Fig.22 Run result of truest decidability

图 22 可信判定结果图

5.2 TVP-PCA方法的性能分析

初始化过程 *VMAgent* 和 *VMMService* 分别与 *CA* 交互申请证书共需 0.6s.虚拟机远程认证阶段共需耗时 1.6s,其中包括实体间信息传递和每次交互信息的现时验证共 1.58s,*vTPM* 读取 *vPCR* 值执行 *vPCRExtend* 和 *vPCRRead* 共 0.02s,虚拟机从虚拟机管理器证书中获取 *VMMService* 地址和 *Challenger* 与 *VMAgent* 通信时加解密所费时间极少对系统可以忽略不计.虚拟机管理器远程认证阶段和虚拟机远程证明阶段实体间操作过程相同,所以耗时也为 1.6s.最后,挑战者进行可信判定耗时 0.1s.因此,用该方法实现虚拟平台远程认证共需时间为 0.6+1.6×2+0.1=3.9s.以下为虚拟平台分别向 4 个挑战者进行远程认证的耗时图表,每次在认证虚拟平台时让 *VMAgent* 和 *VMMService* 重新初始化,以测试初始化过程的效率;而在真正的使用过程中,只需在启动后的第 1 次远程认证中进行初始化.

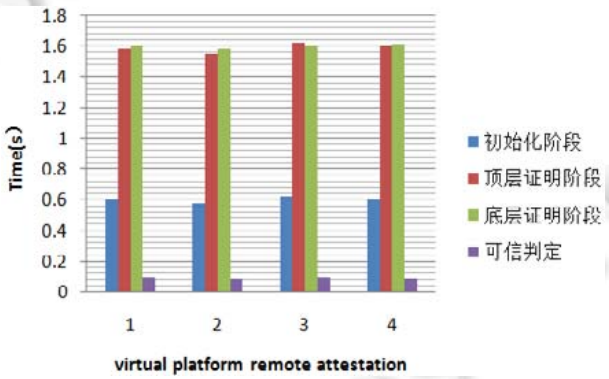


Fig.23 Time of remote attestation

图 23 远程认证时间

6 总 结

本文总结了在可信云环境中远程证明方法现有的成果,针对现有的可信终端远程证明方案,包括隐私 CA (privacy certification authority,简称 PCA)方案和直接匿名证明(direct anonymous attestation,简称 DAA)方案,都并不能直接用于可信虚拟平台,而且 TCG 发布的《Virtualized Trusted Platform Architecture Specification》1.0 版中,可信虚拟平台的远程证明方案仅仅是个框架.本文提出了一种自顶向下的可信虚拟平台远程证明实施方案——TVP-PCA,该方案是在虚拟机中设置一个认证代理,在虚拟机管理器中新增一个认证服务,挑战方首先通过顶层的认证代理证明虚拟机环境可信,然后通过底层的认证服务证明运行于物理平台上的虚拟机管理器可信,顶层和底层证明合起来,确保了整个虚拟平台的可信.文中有效解决了顶层证明和底层证明的同一性问题.实验表明:本方案不仅能证明虚拟机的可信,而且还能证明虚拟机管理器和物理平台的可信,因而证明了云环境中的虚拟机是真正可信的.

References:

[1] Luo L, Wu WJ, Zhang F. Energy modeling based on cloud data center. Ruan Jian Xue Bao/Journal of Software, 2014,25(7): 1371−1387 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4604.htm [doi: 10.13328/j.cnki.jos.004604]

[2] 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]

[3] Bera S, Misra S, Rodrigues JJPC. Cloud computing applications for smart grid: A survey. IEEE Trans. on Parallel and Distributed Systems, 2015,26(5).

[4] Li J, Li JW, Chen XF, Jia CF, Lou WJ. Identity-Based encryption with outsourced revocation in cloud computing. IEEE Trans. on Computers, 2015,64(2):425−437.

[5] Zhang H, Jiang HB, Li B, Liu FM, Vasilakos AV, Liu JC. A framework for truthful online auctions in cloud computing with heterogeneous user demands. Proc. of the IEEE INFOCOM, 2016,12(11):805−818.

[6] TCG. 2012. http://www.trustedcomputinggroup.org/

[7] Shen CX, Zhang HG, Feng DG, *et al*. Survey of information security. Sciencein China Series: E, 2007,37(2):129−150 (in Chinese with English abstract).

[8] 沈昌祥,张焕国,王怀民,王戟,赵波,严飞,余发江,张立强,徐明迪.可信计算的研究与发展.中国科学:信息科学,2010,40(2): 139−166.

[9] 张焕国,严飞,傅建明,徐明迪,杨飚,何凡,詹静.可信计算平台测评理论与关键技术研究.中国科学:信息科学,2010,40(2):167−188.

[10] He RY, Wu SJ, Jiang L. A user-specific trusted virtual environment for cloud computing. Information Technology Journal, 2013, 12(10):1905−1913.

[11] Feng DG, Qin Y, Feng W, Shao JX. The theory and practice in the evolution of trusted computing. Chinese Science Bulletin, 2014, 59(32):4173−4189.

[12] Yu DG, Tan CX, Wang J, Wang HH, Yang J. Provable data possession of resource-constrained mobile devices in cloud computing. Journal of Networks, 2011,6(7):1033−1040.

[13] Thilakanathan D, Calvo RA, Chen SP, Nepal S, Liu DX, Zic J. Secure multiparty data sharing in the cloud using hardware-based TPM devices. In: Proc. of the 2014 IEEE 7th Int'l Conf. on Cloud Computing (CLOUD). Anchorage, 2014.

[14] Teemu K, Sami L, Hilkka K. Opportunities in using a secure element to increase confidence in cloud security monitoring. In: Proc. of the 8th Int'l Conf. on Cloud Computing. New York: IEEE, 2015.

[15] Park SJ, Yoon JN, Kang C, Kim KH, Han TS. TGVisor: A tiny hypervisor-based trusted geolocation framework for mobile cloud clients. In: Proc. of the 3rd IEEE Int'l Conf. on Mobile Cloud Computing, Services, and Engineering. San Francisco: IEEE, 2015.

[16] Tan HL, Hu W, Jha S. A remote attestation protocol with trusted platform modules (TPMs) in wireless sensor networks. Security and Communication Networks, 2015,8(13):2171−2188.

[17] Fu DL, Peng XG. TPM-Based remote attestation for wireless sensor networks. Tsinghua Science and Technology, 2016,21(3): 312−321.

[18] Khiabani H, Idris NB, Manan JA. Unified trust establishment by leveraging remote attestation—Modeling and analysis. Information Management and Computer Security, 2013,21(5).

[19] Chang XL, Liu JQ, Xing B, Yuan ZL. Lightweight, scalable and os-transparent remote attestation of runtime program. Applied Mechanics and Materials, 2012,198(199):506−511.

[20] He RY, Wu SJ, Jiang L. A user-specific trusted virtual environment for cloud computing. Information Technology Journal, 2013, 12(10):1905−1913.

[21] Zhang QY, Feng DG, Zhao SJ. Research of platform identity attestation based on trusted chip. Journal of Communication, 2014,35(8):95−106 (in Chinese with English abstract).

[22] Tan L, Chen J. Remote attestation project of the running environment of the trusted terminal. Ruan Jian Xue Bao/Journal of Software, 2014,25(6):1273−1290 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4414.htm [doi: 10.13328/j. cnki.jos.004414]

[23] Yang L, Ma JF, Zhu JM. Trusted and anonymous authentication scheme for wireless networks. Journal of Communication, 2009,30(9):29−35 (in Chinese with English abstract).

[24] Zhou YW, Yang B, Zhang WZ. Provable secure trusted and anonymous roaming protocol for mobile internet. Chinese Journal of Computers, 2015,38(4):733−748 (in Chinese with English abstract).

[25] Brickell E, Camenisch J, Chen L. Direct anonymous attestation. In: Proc. of the 11th ACM Conf. on Computer and Communications Security. 2004. 132−145.

[26] Chen XF, Feng DG. Direct anonymous attestation based on bilinear maps. Ruan Jian Xue Bao/Journal of Software, 2010,21(8): 2070−2078 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/3579.htm [doi: 10.3724/SP.J.1001.2010. 03579]

[27] Tan L, Meng WM, Zhou MT. An improved direct anonymous attestation scheme. Journal of computer research and development, 2014,51(2):334−343 (in Chinese with English abstract).

[28] 周彦伟,杨波,吴振强,何聚厚,李骏.基于身份的跨域直接匿名认证机制.中国科学:信息科学,2014,44(9):1102−1120.

[29] Yang L, Ma JF, Pei QQ, Ma Z. Direct anonymous authentication scheme for wireless networks under trusted computing. Journal of Communications, 2010,31(8):98−104 (in Chinese with English abstract).

[30] Yang L, Ma JF, Jiang Q. Direct anonymous attestation scheme in cross trusted domain for wireless mobile networks. Ruan Jian Xue Bao/Journal of Software, 2012,23(5):1260−1271 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4052.htm [doi: 10.3724/SP.J.1001.2012.04052]

[31] Yang L, Zhang JW, Ma JF, Liu ZH. Improved direct anonymous attestation scheme for mobile computing platforms. Journal of Communications, 2013,34(6):69−75 (in Chinese with English abstract).

[32] Tan L, Chen J. Trusted agent for collecting trustworthiness evidence in terminal dynamical running environment. Ruan Jian Xue Bao/Journal of Software, 2012,23(8):2084−2103 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4115.htm [doi: 10.3724/SP.J.1001.2012.04115]

[33] Mo JQ, Hu ZW, Lin YH. Improved scheme of DAA authentication based on proof mechanism of a committed number lying in a specific interval. Computer Science, 2012,39(8):111−114 (in Chinese with English abstract).

[34] Yue XH, Zhou FC, Lin MQ, Li FX. Anonymous attestation scheme with user-controlled linkability for trusted mobile platform. Chinese Journal of Computers, 2013,36(7):1434−1447 (in Chinese with English abstract).

[35] Yang B, Feng DG, Qin Y, Zhang QY, Xi C, Zheng CW. Research on direct anonymous attestation scheme based on trusted mobile platform. Journal of Computer Research and Development, 2014,51(7):1436−1445 (in Chinese with English abstract).

[36] Liu Q, Weng C, Li M, Luo Y. An in-VM measuring framework for increasing virtual machine security in clouds. IEEE Computer and Reliability Societies, 2010,8(6):56−62.

[37] Li C, Raghunathan A, Jha NK. A trusted virtual machine in an untrusted management environment. IEEE Computer Society, 2012, 5(4):472−483.

[38] Zhang Y, Feng DG, Yu AM. Virtual machine anonymous attestation in cloud computing. Ruan Jian Xue Bao/Journal of Software, 2013,24(12): 2897−2908 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4389.htm [doi: 10.3724/SP.J.1001. 2013.04389]

[39] Yang B, Feng DG, Qin Y, Zhang YJ. Secure access acheme of cloud services for trusted mobile terminals unsing trustzone. Ruan Jian Xue Bao/Journal of Software, 2016,27(6):1366−1383 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/ 5000.htm [doi: 10.13328/j.cnki.jos.005000]

[40] Du J, Dean DJ, Tan Y, Gu X, Yu T. Scalable distributed service integrity attestation for software-as-a-service clouds. ACM, 2014, 25(3):730−739.

[41] Cavallar S, Dodson B, Lenstra AK. Factorization of 512-bit RSA modulus. In: Proc. of the Eurocrypt 2000. LNCS 1807. 2000. 1−18.

[42] Wang YZ. Security analysis of discrete logarithm based cryptographic schemes and secure implementation technology [Ph.D. Thesis]. Chongqing: Chongqing University (in Chinese with English abstract).

[43] Stallings W. Cryptography and Network Security: Principles and Practice. 6th ed., Beijing: Publishing House of Electronics Industry, 2011 (in Chinese).

附中文参考文献:

[1] 罗亮,吴文峻,张飞.面向云计算数据中心的能耗建模方法.软件学报,2014,25(7):1371−1387. http://www.jos.org.cn/1000-9825/4604.htm [doi: 10.13328/j.cnki.jos.004604]

[2] 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]

[7] 沈昌祥,张焕国,冯登国,等.信息安全综述.中国科学:E 辑,2007,37(2):129−150.

[21] 张倩颖,冯登国,赵世军.基于可信芯片的平台身份证明方案研究.通信学报,2014,35(8):95−106.

[22] 谭良,陈菊.一种可信终端运行环境远程证明方案.软件学报,2014,25(6):1273−1290. http://www.jos.org.cn/1000-9825/4414.htm [doi: 10.13328/j.cnki.jos.004414]

[23] 杨力,马建峰,朱建明.可信的匿名无线认证协议.通信学报,2009,30(9):29−35.

[24] 周彦伟,杨波,张文政.可证安全的移动互联网可信匿名漫游协议.计算机学报,2015,38(4):733−748.

[26] 陈小峰,冯登国.一种基于双线性映射的直接匿名证明方案.软件学报,2010,21(8):2070−2078. http://www.jos.org.cn/1000-9825/3579.htm [doi: doi: 10.3724/SP.J.1001.2010.03579]

[27] 谭良,孟伟明,周明天.一种优化的直接匿名证言协议方案.计算机研究与发展,2014,51(2):334−343.

[29] 杨力,马建峰,裴庆祺,马卓.直接匿名的无线网络可信接入认证方案.通信学报,2010,31(8):98−104.

[30] 杨力,马建峰,姜奇.无线移动网络跨可信域的直接匿名证明方案.软件学报,2012,23(5):1260−1271. http://www.jos.org.cn/1000-9825/4052.htm [doi: 10.3724/SP.J.1001.2012.04052]

[31] 杨力,张俊伟,马建峰,刘志宏.改进的移动计算平台直接匿名证明方案.通信学报,2013,34(6):69−75.

[32] 谭良,陈菊.可信终端动态运行环境的可信证据收集代理.软件学报,2012,23(8):2084−2103. http://www.jos.org.cn/1000-9825/4115.htm [doi: 10.3724/SP.J.1001.2012.04115]

[33] 莫家庆,胡忠望,林瑜华.基于特定区间承诺值证明机制改进的 DAA 认证方案.计算机科学,2012,39(8):111−114.

[34] 岳笑含,周福才,林慕清,李福祥.面向可信移动平台具有用户可控关联性的匿名证明方案.计算机学报,2013,36(7):1434−1447.

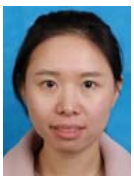
[35] 杨波,冯登国,秦宇,张倩颖,奚璨,郑昌文.基于可信移动平台的直接匿名证明方案研究.计算机研究与发展,2014,51(7):1436−1445.

[38] 张严,冯登国,于爱民.云计算环境虚拟机匿名身份证明方案.软件学报,2013,24(12):2897−2908. http://www.jos.org.cn/1000-9825/4389.htm [doi: 10.3724/SP.J.1001.2013.04389]

[39] 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 [doi: 10.13328/j.cnki.jos.005000]

[42] 王玉柱.离散对数密码系统安全性分析与安全实现技术研究[博士学位论文].重庆:重庆大学.

[43] Stallings W.密码编码学与网络安全.第 6 版,北京:电子工业出版社,2011.



胡玲碧(1993－),女,四川威远人,学士,主要研究领域为可信计算.

谭良(1972－),男,博士,教授,主要研究领域为可信计算,网络安全,云计算,大数据处理.