中国科学技术大学

硕士学位论文

可信云计算平台模型的研究及其改进

姓名：王含章

申请学位级别：硕士

专业：信息安全

指导教师：黄刘生

2011-04-15

摘 要

云端数据安全与隐私保护被普遍认为是云计算发展的两大重要内容，然而到目前为止客户没有任何手段能够证明自己上传的数据与应用的完整性与机密性，因此产生了客户与云服务提供商之间的信任问题，这也就成了阻碍云计算发展的最大障碍之一。

将可信计算技术与云计算相结合是解决这一信任问题的方向之一，这个针对基础架构即服务（Infrastructure-as-a-Service, IaaS）模型提出的可信云计算平台（Trusted Cloud Computing Platform, TCCP)解决方案在 2009 年提出。该可信云计算平台模型通过一个外部的可信协调者(Trusted Coordinator, TC)来认证内置可信芯片的服务器，然后管理活跃的可信服务器列表，并参与到虚拟机（Virtual Machine）启动与动态迁移的过程中。而可信节点（内置可信芯片且安全运行的服务器）通过可信虚拟机监视器（Trusted Virtual Machine Monitor, TVMM）来保证运行的虚拟机内部数据不被黑客或有特权的管理人员监视或修改。不过由于对可信第三方的过度依赖，可信云计算平台模型只能停留在理论探讨阶段，没有实用的可行性。本文针对可信云计算平台模型的局限性进行了深入研究，提出了一种改进的基于直接匿名认证（Direct Anonymous Attestation, DAA）与隐私认证中心（Privacy Certification Authority, Privacy CA）策略的 TCCP 模型，较好解决了可信第三方在性能与安全性的瓶颈问题。

本文主要是通过引入 DAA 策略并且根据可信芯片的中立特性设计了一种让一部分内部可信节点（Trusted Nodes, TNs）成为 Privacy CAs(内部小型可信第三方)的协议，其中内部的 Privacy CA 承担了 TC 的 TNs 管理职责与部分认证职责。在云计算平台的每一个区域（Zone）里面都设有多个 Privacy CA，每个 Privacy CA 有自己管理的可信区域，本文采取这样一种隔离措施来提高平台的安全性，若某个 Privacy CA 管理的区域被攻破后能够将损失降低到最小。这些不同的管理区域是通过 Privacy CAs 选出来的内部可信联络者（Internal Trusted Coordinator, ITC）进行联系，ITC 的选举采用安全多方计算（Secure Multi-party Computation, SMC）算法，从而保证了 ITC 的选举不被少数被操纵的无赖 Privacy CAs 所影响。Privacy CAs 与选举出来的 ITC 能够完全替代 TC 的可信节点管理功能，并参与到虚拟机的创建与动态迁移的过程中。这里的难点是如何让 Privacy CA 与 ITC 替代 TC 大部分功能同时不减弱平台的安全性。

在原始可信云计算平台模型中，虚拟可信平台的概念直接从基于虚拟技术的可信平台架构里面引入，对于是否适合云计算平台的问题并没有进行探讨。而本

文对虚拟可信平台的架构进行了研究，并根据 TCCP 模型的需求对可信平台架构进行了相应的改变。

本文采取的云平台架构基于开源的 Eucalyptus, 虚拟可信平台架构基于 vTPM，可信芯片必须符合 TPM 1.2 标准（DAA 策略在 TPM1.2 标准下才能够被使用）。

可信云计算平台模型有可信芯片这种专门的安全硬件来支撑，是解决云计算隐私保护与安全方面问题的最好方法之一，尤其是可信计算技术的研究与应用已经比较成熟。因此针对可信云计算平台模型的设计具有很高的参考价值，本文改进原始可信云计算平台的设计思路也可以为以后实际平台的设计提供帮助，使得人们对数据安全的担忧不再成为云计算发展的障碍之一。

关键词：云计算 虚拟可信平台架构 隐私保护 架构即服务 直接匿名认证

ABSTRACT

Data security and privacy preserving in cloud are widely believed to be two prime fields during the rapid expansion of cloud computing. However, by far, there are no effective methods for users to verify the confidentiality and integrity of the data that they upload to the cloud. Hence,a trust issue between Cloud Service Provider (CSP) and users comes up, and it's becoming one of the biggest barriers to cloud computing development.

Introducing trust computing technology is a possible solution dealing with the trust issue. A Trusted Cloud Computing Platform (TCCP) model based on Infrastructure-As-A-Service (IaaS) is proposed in 2009. In that paper there is a Trusted Coordinator (TC) in External Trusted Entity (ETE) which verifies active servers containing Trusted Platform Module (TPM) and maintains a list of trusted nodes (TNs, active normally running servers containing TPM). The backend of TN runs a Trusted Virtual Machine Monitor (TVMM) to prevent hackers and insiders from inspecting or modifying the memory of user's Virtual Machine (VM). However the model relies on trusted third party too much, and that makes it becoming a bottleneck, hence the TCCP model is of no practical use. In this paper the limitation of the original TCCP model is studied intensively, and an improved TCCP model based on Direct Anonymous Attestation (DAA) and Privacy Certification Authority (Privacy CA) scheme is proposed to solve the over-dependence on the trusted third party issue.

In this papaer based on the neutral feature of TPM we introduce DAA and Privacy CA scheme to deSign a protocol that lets a part of TNs become Privacy CAs (tiny internal trusted third parties) which would take part responsibility of managing and verifying TNs. In a Zone of cloud there is a number of Privacy CAs, and each of them has its own trusted management area, the quarantine measure applied here is used to prevent further losses if one trusted management area is damaged or hacked. And these trusted management areas are connected by an Internal Trusted Coordinator (ITC) which is elected by Privacy CAs. In case of the election result gets manipulated, a Secure Multi-Party Computation (SMC) algorithm is applied here. Privacy CAs and ITC would take the whole responsibility of managing TNs, and get involved in the creation and live migration of VMs. The aporia in this paper is how to provide the same level of security and privacy preserving while modifying the original TCCP model to make it more available and reliable.

In the original TCCP model, the concept of virtual trusted platform is directly introduced from the virtualization of trusted platform, and the issue whether it's suitable for cloud computing platform is not disscussed. In this paper the architecture of virtual trusted platforms is studied, and modified based on the needs of our TCCP model.

The architecture of cloud computing platform here is based on an open source Eucalyptus, the architecture of virtual trusted platform is based on vTPM, and TPM must reach TPM 1.2 specification (DAA scheme can only be applied in TPM 1.2 specification).

The TCCP model is supported by the dedicated safety hardware TPM, hence it's one of the best models dealing with the security and privacy preserving issue in cloud, especially the theory researches and application practices of trusted computing technoligy are quite mature now. Therefore the deSign of TCCP model has very high reference value; the consideration of improving the original TCCP model in this paper might provide ideas for deSigning an actual TCCP to clear trust issue barrier in future.

Key Words: cloud computing, virtual trusted platform architechture, privacy preserving, Infrastructure-as-a-service, direct anonymous attestation

中国科学技术大学学位论文原创性声明

本人声明所呈交的学位论文,是本人在导师指导下进行研究工作所取得的成果。除已特别加以标注和致谢的地方外，论文中不包含任何他人已经发表或撰写过的研究成果。与我一同工作的同志对本研究所做的贡献均已在论文中作了明确的说明。

作者签名：_____ 签字日期：_____

中国科学技术大学学位论文授权使用声明

作为申请学位的条件之一，学位论文著作权拥有者授权中国科学技术大学拥有学位论文的部分使用权，即：学校有权按有关规定向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅，可以将学位论文编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。本人提交的电子文档的内容和纸质论文的内容相一致。

保密的学位论文在解密后也遵守此规定。

□公开 □保密（____年）

作者签名：_____ 导师签名：_____

签字日期：_____ 签字日期：_____

第 1 章　绪论

1.1　引言

云计算作为一个正在快速发展的计算模型，能够大幅减少 IT 的运营成本以及显著提高资源管理效率，并且能快速满足业务要求，已经被人们认为将引发信息时代的下一场浪潮。而这种新的计算模型同时也带来了对于不同层面安全的不确定性（例如网络，主机，程序与数据层面），因此对于云计算安全方面的研究也成了热点之一。

在云安全研究内容里面，对于云端数据与计算完整性与机密性的保护是急需得到解决的课题，因为公司或其他组织与个人需要将资料或程序上传到云端，这样会直接失去对数据与应用的控制权，同时云服务提供商对于云端内部的信息却甚少透露，用户对于云端服务提供商如何处理自己的数据与应用毫不知情。控制权尤其是知情权的丧失会引发用户对于是否采用云计算解决方案的担忧，用户与云计算服务提供商之间的信任问题由此成为阻碍云计算发展的最大障碍之一。解决用户对云服务提供商的这种信任问题正是本文研究的重心。

在 2009 年 6 月由 N. Santos et al 结合可信计算技术提出可信云计算平台[1]的思路是解决信任问题的方向之一，本文深入探讨了原始可信云计算平台模型的局限性以及解决这种局限性的方法，首先研究了虚拟可信平台架构，并根据云计算平台的要求进行了相应改动，然后针对原始可信云计算平台的局限性提出一种将云端内部可信服务器承担可信第三方大部分职责的方案，使得可信云计算平台具有更高的可用性，可靠性与安全性。

本章首先介绍的是与本文研究内容密切相关的云计算安全研究现状，为此也简要描述了云计算的概念，模型描述以及与本文相关的云计算技术。然后对可信技术与可信平台的研究现状做了必要的介绍。接着对原始可信云计算平台模型的原理与局限性做了相关说明。最后列出了本文的的主要工作及内容上的结构安排。

1.2　计算安全技术的研究现状

1.2.1　云计算的定义，模型描述与相关技术

1.2.1.1　云计算的概念与模型描述

“ 云计算 ”的概念被普遍认为基于 5 个特征：多租户架构，巨大的可扩展性，弹性，按需付款与资源的自动供应。它不仅被认为是一个技术名词，也被用来描述一种将计算与应用能力按需外包的商业模型。

云计算可以按照不同的方式进行分类，其中一种很普遍的按照服务类型进行分类的模型被称作 SPI 模型，大写的三个字母分别代表三种不同的服务模型：基础架构即服务（Infrastructure-as-a-Service, IaaS），平台及服务（Platform-as-a-Service, PaaS）和软件即服务（Software-as-a-Service, SaaS）。

本文研究的云计算平台属于 IaaS，IaaS 就是以服务的形式为用户提供基本的基础设施资源，例如虚拟机/存储/网络等资源。用户不需要购买或者维护这些硬件设备或者相关软件，只需要通过网络租赁来搭建自己的平台系统。典型的 IaaS 为 AWS （Amazon Web Service），它包括基于 Xen 虚拟技术为用户提供计算资源的 EC2（Elastic Compute Cloud），为用户提供高可用低成本存储的 S3（Simple Storage Service）以及提供可靠信息传送的 SQS（Simple Queue Service）等等。本文选择 Eucalyptus 作为云计算平台的研究对象，因为它可以看作是 AWS 的开源实现，接口与 Amazon EC2, S3 和 EBS 兼容。

1.2.1.2　云计算的相关技术

云计算的发展离不开技术的支持，将已存在的技术进行改变后能够适应云计算的部分需要，除此之外还需要大量创新的技术。本小节将介绍与本文研究内容有关的技术现状。

❖　虚拟化

虚拟化技术的出现是为了充分利用现有的服务器，网络，存储系统等设备资源，并且快速有效地调配系统资源以适应业务需求。Wikipedia 对虚拟化的定义是“ 虚拟化是一个表现逻辑群组或电脑资源的子集的进程，用户可以用比原本的组态更好的方式来存取这些进程。这些资源的新虚拟部份是不受现有资源的架设方式，地域或物理组态所限制。一般所指的虚拟化资源包括计算能力和资料储存。” [1]。

虚拟化技术的一个常见类型是系统虚拟化，系统虚拟化的思想是在物理机上虚拟出多台虚拟机（Virtual machine, VM）。VM 包含操作系统与应用程序，运行

在一个隔离环境下具有完整硬件功能的逻辑计算机系统中。VMs 共享物理资源，同时运行且不互相影响。系统虚拟化用到服务器上就成了服务器虚拟化，多个虚拟服务器可以被物理服务器托管。

这里面需要提到的是半虚拟化技术（Para-virtualization），它使用虚拟机监视器（Virtual Machine Monitor, VMM）分享存取底层硬件。半虚拟化技术使得操作系统知道自身运行在虚拟环境中，在必要时对虚拟化平台进行调用来执行指令。虽然半虚拟化技术需要操作系统作出部分修改与虚拟化平台完全兼容，但是半虚拟化技术提供了与原始操作系统相近的性能。Citrix 的 Xen 就是半虚拟化的一个典型的例子，而且是开源的，因此 Xen 也被本文选作虚拟化平台的研究对象。不过随着硬件辅助虚拟化技术的发展，客户操作系统无需通过修改就能兼容虚拟化平台。

如前所述，云计算的一大特点是多租户架构，即不同的用户共享一个物理宿主机的软硬件资源，如果不能保证多个用户之间的有效隔离，那么用户的使用体验可能受到影响，数据隐私可能受到侵犯。如果采用虚拟化技术，用户的每个应用或服务单独存在一个虚拟机环境内，这样不同虚拟机之间就有较强的隔离(但还是会有漏洞)。另外虚拟化技术能够抽象软硬件资源，这样不仅使得底层设备的差异性对于上层是透明的，简化了云计算的管理工作与开发人员的编程难度，而且能够保证服务以一定的标准提供给用户，不会因为服务器的规格，网络环境与地理位置的不同造成服务质量的不一致。正是因为虚拟化计算的种种优势，大部分云计算解决方案都采用了虚拟化技术。

❖ 资源调度

资源调度的概念是对各种资源进行合理有效的调节和测量以及分析和使用[2]。若一个计算任务需要进行资源调度，一般来说有两种方式：提高计算任务能够使用的资源或者将计算任务转移到其他机器上。对于采用虚拟化技术的云计算而言，由于计算任务被封装在虚拟机里面，若采用转移计算任务的方式又不想影响用户虚拟机的正常运作，需要采用虚拟机的实时迁移技术（Live Migration）。

实时迁移技术是指虚拟机在运行时，将其运行装态完整快速地从一台宿主机转移到另一台宿主机，期间用户感觉不到任何异同。实时迁移技术需要 VMM 的支持，开始时源宿主机 VMM 将虚拟机内存页面拷贝到目的宿主机 VMM，当拷贝完成后目标虚拟机就开始运行，源虚拟机的运行被终止，迁移过程结束。

1.2.2　云计算的潜在安全风险与应对策略

首先如图 1.1 所示[5]，在传统架构上除了底层控制权需要与服务提供商共享

外，公司或组织的 IT 部门几乎掌管五个技术层面。而过渡到 IaaS，PaaS 和 SaaS，IT 部门的控制权逐渐丧失。围绕图 1.1 讨论下云计算面临的潜在安全风险，本文给出了部分对应措施的建议。

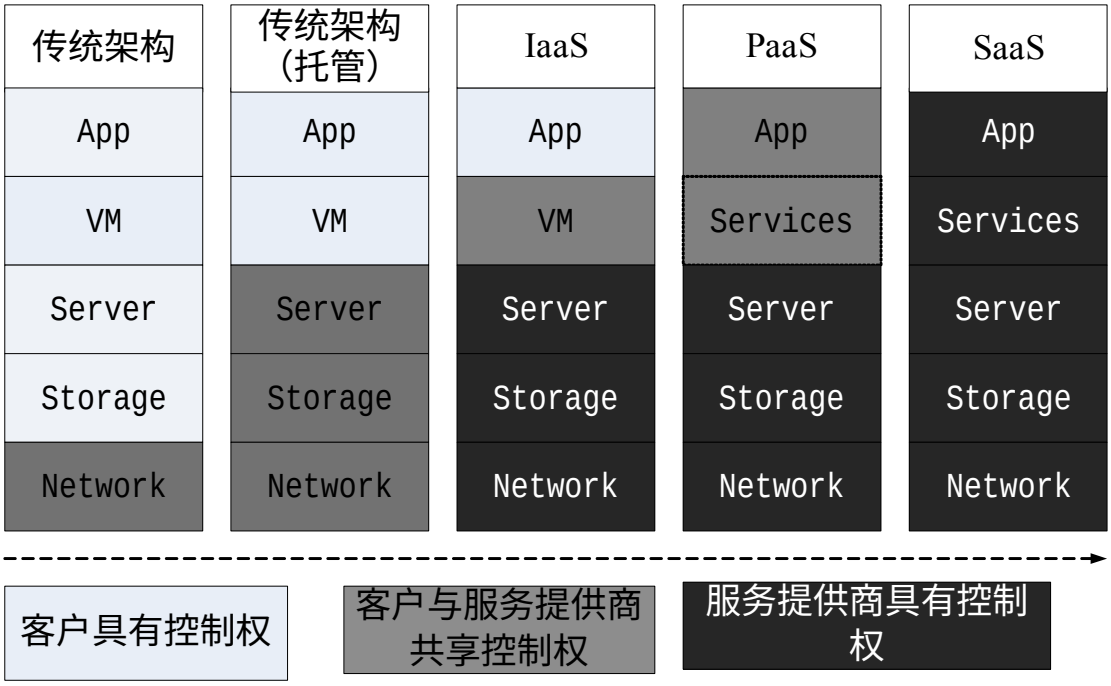


图 1.1　计算给 IT 部门的管理架构带来的影响

❖　架构安全与安全管理

在云计算时代，从 IaaS 模型到 SaaS 模型，云服务提供商（Cloud Service Provider, CSP）的控制权逐渐增加。架构安全的解决方案需要划清 CSP 与用户的安全责任，例如网络到服务器的底层完全由 CSP 控制，那么底层安全应该由 CSP 负责管理以及承担安全责任。对于上层来说，SaaS 与 PaaS 模型中安全防护的责任也应该由 CSP 承担，但是为了防止黑客入侵，CSP 极少公开相关信息，同样的用户对于平台安全性能知之甚少，因此为了自身数据与应用安全的考虑，用户应该和 CSP 协商获得一定知情权以确保 CSP 对于上层的安全防护。而对于 IaaS 来说，由图 1.1 可知用户应该对部署在云端的 VM 内部安全负主要安全责任，然而负责创建与维护 VM 的 VMM 则是由 CSP 管理，CSP 应该确保 VMM 不会影响到 VM 安全。

❖　数据安全

数据安全是云计算安全里面最重要的一方面，对于存储在云端或者在云端内部传输的数据来说，最好的安全措施是采用加密技术。然而部分 CSPs 对与存储的数据不采用加密技术，因为这会妨碍对数据的索引或搜索，更何况大规模采用

加密技术会花费大量的计算与金钱。而对于 SaaS 与 PaaS 模型来说，大量的用户数据处于共享状态下，未经授权的访问会对数据的安全性造成破坏。而且即使存储与传送数据都采用了加密技术，当数据需要被处理时，它还是需要处于解密状态。加密技术的广泛使用同时又带来了密钥管理的难题。

❖ 隐私与审计

云计算的隐私问题在访问控制方面，上传到云端的敏感数据能否被非授权访问，用户访问时的个人身份信息会被保留多久，CSP 是否会向政府或有关机构提供用户的个人身份信息等等问题都值得关注。当 CSP 失去对用户个人身份信息的控制时，用户会直接或间接承担个人身份信息丢失的后果，例如身份盗用，丧失对云端数据的控制权或个人生活被干扰等。

在数据隐私方面，敏感数据是否会被 CSP 利用，敏感数据被删除时 CSP 是否会保留备份，敏感数据能否保证不被非法篡改等等问题也值得关注。

如何处理穿越国界的数据流是隐私保护不得不面临的问题之一，云端内部有大量的数据中心，数据可能存储在不同的国家，CSP 的数据中心是否在所在国的司法管辖之下。若这些国家进行调查时，CSP 会不会提交用户的数据。当数据需要穿越国界时有没有得到用户的同意，尤其是不同国家会有冲突的法律规定，某些数据的传输会给用户造成负面影响。

正因为用户对于数据安全与隐私保护问题的担心，所以 CSP 需要处理大量来自用户的要求其证明安全性与可靠性的要求。有时政府或企业也会对云端内部违法行为和利益纠纷进行调查，CSP 如何配合提供相关数据也是难题之一。除了在技术层面解决上述问题以外，这里还需要法律的配合。

1.3　可信云计算技术的研究现状

用户对于上传到云端的数据与应用的安全与隐私保护问题的担忧造成了与 CSP 之间的信任问题，如果黑客可以通过漏洞甚至 CSP 内部恶意员工可以利用特权获得用户敏感数据的话，那么使用审计手段也无法完全解决这种信任问题。

在 2009 年提出的 TCCP[1]模型开启了解决信任问题的思路，那就是结合可信计算从技术层面防止用户的数据被非法窥探或篡改。要介绍 TCCP 模型，首先要介绍可信计算平台的相关内容。

1.3.1　可信计算平台的研究现状与相关技术

本文讨论的可信计算平台（Trusted Computing Platform, TCP）是由可信计算

组织（Trusted Computing Group, TCG）提出的。如果一个平台总是按预期的方式来执行指定的功能，那么 TCG 认为它就是可信的，其使用的方法是通过系统启动系列来判断。根据信任传递的思想，TCG 采用了链式的设计方案，系统是从信任根（即可信度量根，CRTM）开始启动，CRTM 是 Bios 里面的一部分，是启动时第一个获取控制权的模块，其完整性与可信性由物理方式得以保护。然后由信任根通过完整性度量方法建立一条信任链，逐步将信任关系从信任根扩大到整个系统。

在有众多参与者的大型系统和分布式环境中，TCP 能够保护至少一个参与方的利益，比如可以在客户端保护服务器端的利益，在服务器端保护客户端的利益，甚至在多方计算时保护所有参与者的利益与隐私。这也是 TCP 最有价值的功能之一。

❖ TPM 芯片

TPM 模块芯片是嵌入主板上的一个价格低廉的独立芯片，用来参与计算以提高计算的安全性。它的主要功能是对计算平台的环境进行度量，保护并将度量结果报告给验证者。系统启动时，信任根 CRTM 记录了控制权传递给整个 BIOS 之前哪一部分将被用来启动系统，接下来启动的每一个部件都记录了下一个将控制系统的部件，TPM 正是用于存储这些记录值并根据外部请求作出响应。

由于 TPM 存储的有限性，TPM 采用存储信息摘要的方式解决这一问题，其中负责存储的部件叫做平台配置寄存器（Platform Configuration Register, PCR）。各种信息摘要存储在 PCR 里面，并通过扩展（extend）操作进行更改：

$$PCR \leftarrow SHA1(PCR||Input) \tag{1.1}$$

其中 Input 为 160bit 的哈希值。

TPM 其他的功能还包括提供对数据与签名的安全存储，存储对称与非对称密钥并附带密码算法。另外标识 TPM 身份的是一个唯一的背书密钥（Endorsement Key, EK），EK 由生产厂商出厂时确定，由于它的唯一性会带来隐私问题，一般 EK 只用于对其他产生密钥的 TPM 证书的解密，这只能由 TPM 所有者与 CA 协同完成。

在虚拟化环境中，为了使虚拟机得到可信计算技术的支持，TPM 设备的虚拟化是需要的。如何使虚拟化的 TPM（virtualized TPM, vTPM）保持 TPM 物理设备同样的安全性质，如何让信任链从 TPM 扩展到 vTPM 以及当 VM 迁移时 vTPM 如何跟着迁移都是 TPM 虚拟化需要解决的问题。其中由 IBM 开发的 vTPM[6]**错误！未找到引用源。** 是 TPM 虚拟化的典型例子之一。

❖ 完整性度量

系统启动时 TPM 参与完整性度量最开始的过程如图 1.2 所示，CRTM 记录了 BIOS 状态，BIOS 记录了参与启动的硬件和引导加载程序的信任状态，引导加载程序记录了操作系统的信任状态[7]。在这个过程中，当信任传递到下一个执行模块之前，当前执行模块将对下一个模块的完整性进行度量：首先计算下一个模块的 hash 值（ExecutableCode），然后调用 TPM 的 extend 操作，由（1.1）式得：

$$PCR \leftarrow SHA1(PCR|| SHA1(ExecutableCode))$$

信任链建立完成后，PCR 寄存器会得到一个累计的度量结果。一般来说最终形式的信任链如下所示：

CRTM →BIOS →Boot Loader →OS→ OS Components →App。

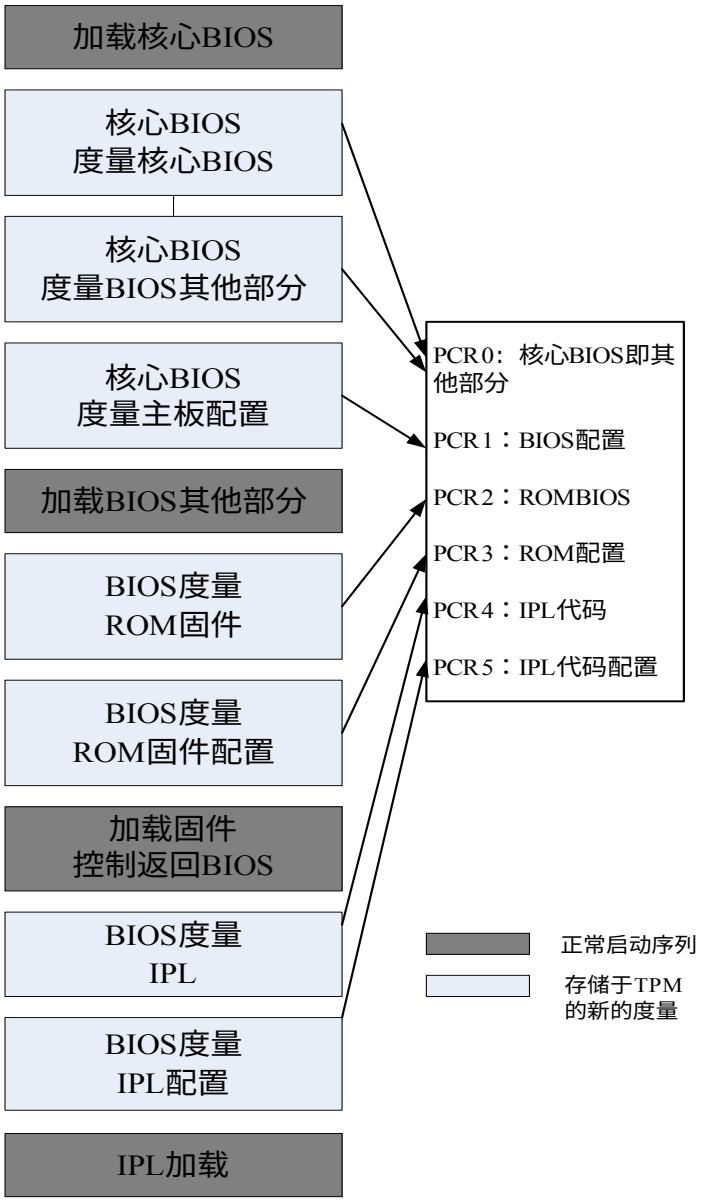


图 1.2　信任链的建立与完整性度量的过程

❖ 远程证明

当使用者通过互联网访问远程 TCP 时，用户如何知道 TCP 处于可信状态，这里需要采用远程证明（Remote attestation）的方式，其主要步骤如下：

1. 验证者向远程 TCP 发出证明请求，请求包含了一个 nonce ；
2. TPM 使用身份密钥对平台完整性度量结果和 nonce 进行数字签名；
3. 验证者收到反馈后，根据 nonce 值确认来自目的 TPM，然后根据平台完整性度量结果判断平台配置是否可信。

然而平台上运行的软件由于版本，类别和配置等种种不同会导致度量结果极为复杂，验证者一般无法仅仅通过完整性度量结果判断平台的可信性，目前针对这个问题有基于属性证明[26][29]（Property-Based Attestation, PBA）的解决思路和框架。

基于属性证明的思想是：验证者并不关心度量结果 hash 值（包括软件配置信息），关心的是平台是否保持了完整性。通过将度量结果 hash 值与属性证书绑定以后，证明者可以通过展示属性证书来证明自己平台的完整性。A. R. Sadeghi and C. Stüble[26]提出了一种可信第三方证明度量结果 hash 值与属性值相关联并颁布属性证书的方案，不过需要对现有的 TPM 标准进行修改以支持属性证书的概念。而 U. Kühn[28]提出的方案直接基于支持现有 TCG 技术的软硬件环境，使用 bootloader 利用可信第三方颁布的属性证书将度量结果 hash 值翻译成安全属性。此外不依赖可信第三方的属性证明方案也被提出了[29]。

1.3.2 可信云计算平台的研究现状

在云计算安全所面临的风险中，用户对于数据的安全与隐私问题最为关注。而 CSP 为了应对黑客攻击的威胁，云端内部的信息甚少透露，用户丧失数据控制权引发的焦虑再加上缺少必要的知情权，与 CSP 之间产生的信任问题成为了云计算发展的巨大障碍。

在 IaaS，PaaS 和 SaaS 模型中，唯有 IaaS 模型让用户完全掌控 VM 内部的数据和计算，因此在 IaaS 模型中更容易实现对用户数据安全与隐私的保护。图 1.3 显示了 TCCP 模型的架构，它包括提供并管理云服务的不可信云端管理方（untrusted Cloud Manager, CM），运行 TVMM 以防止内部入侵者窥探或篡改 VM 内部数据的可信节点（Trusted Nodes, TNs）以及由外部可信实体（External Trusted Entity, ETE）维护的用来管理可信节点列表的可信协调者（Trusted Coordinator, TC）。

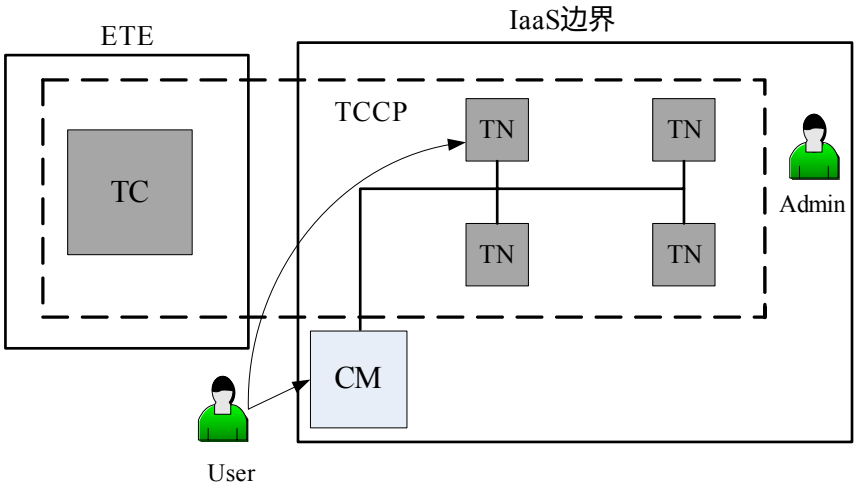


图 1.3　TCCP 模型架构

在原始 TCCP 模型的解决方案下，含有 TPM 的可信服务器上线时首先会通过 EK 与完整性度量结果（Measurement List, ML）向 TC 验证自己的身份。如果验证通过，TC 会将其加入 TN 列表内。当云端一个有高安全需求的 VM 需要启动或实时迁移时，TC 要参与协议之中以确保目的服务器在 TN 列表中，这样 VM 只会运行在可信区间内。

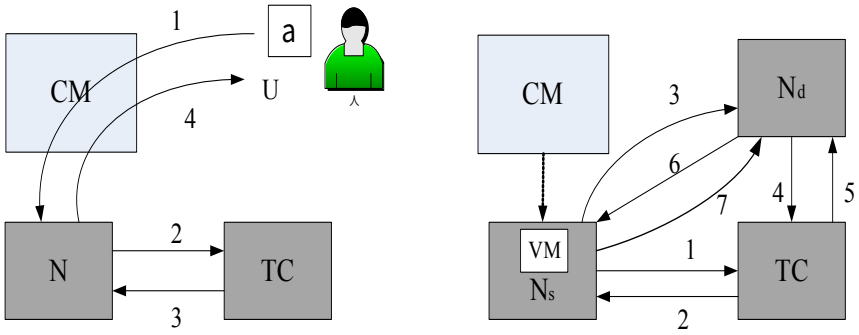


图 1.4　VM 创建与动态迁移协议中的消息传递过程

图 1.4 分别显示了 TCCP 模型里面高安全需求 VM 创建与动态迁移协议里面的消息传递过程[1]，这里省略了协议的详细细节。图 1.4 清楚地显示 TC 在两个协议之中参与了多次计算以确保用户 VM 运行在 TN 上，一旦 TC 瘫痪，TCCP 就不能正常运作。而且云端内部有大规模的可信服务器（TCCP 模型并未对 TC 架构进行详细讨论）要与 TC 进行交互，这也使得 TC 成为整个 TCCP 模型中性能与安全的瓶颈。

TCCP 模型另一个问题是直接利用了可信平台 Terra[10]里面的研究结果，即 Terra 能够创建一个 VMM 强制产生一种黑盒环境，这样运行中的用户虚拟机里面的数据不能被服务器的管理员窥探或篡改。如果每台可信服务器能够保证高安全需求 VM 的数据安全与隐私，那么只需要确保云计算环境中只允许 VM 在可

信区间中创建或迁移。但是虚拟可信平台在云计算环境中可能需要作出哪些改变的问题并没有被探讨。

2009 年这篇关于原始 TCCP 模型设计的论文发表以来，后续论文只是对云计算安全的展望时引用了原始 TCCP 模型的研究成果，至于原始 TCCP 模型的局限性探讨与设计细节上并没有深入。目前 IBM，EMC[23]和 CSA[24][25]等都在研究可信云计算，可以找到的有价值信息却极少。

1.4　章节安排

本章首先对云计算面临的安全风险以及可能采取的应对措施进了必要的介绍，为此还简要阐述了云计算的概念，模型描述与相关的技术。接着对解决用户与可信服务提供商之间的信任问题而提出的原始可信云计算平台模型的架构及其局限性进行了说明，为此本文还简要阐述了与原始可信云计算平台模型相关的可信计算技术。本文接着会对可信云计算平台模型进行更深入的研究，后续章节的安排如下：

第二章具体介绍与本文提出的可信云计算平台模型设计相关的背景知识，包括虚拟可信平台 vTPM 与云计算平台 Eucalyptus 的架构设计，与远程证明相关的 DAA 策略以及与选举算法有关的安全多方计算技术。

第三章提出了一种改进后的可信云计算平台模型，详细阐述了改进模型的架构设计与内部的协议设计，并进行了性能与安全的分析得出改进模型比原始可信云计算平台模型有着更高的安全性，可用性以及可靠性。

第四章是对全文进行了总结，列出了改进原始可信云计算平台采用的思路与解决方案，最后分析了改进模型自身的不足以确定之后的研究方向。

第 2 章　可信云计算平台设计的相关背景知识

2.1　虚拟可信平台与 Xen 架构

结合可信计算技术提出的虚拟可信平台技术是为了解决硬件虚拟化快速发展带来的安全问题。目前的虚拟可信平台包括 NGSCB(Next-Generation Secure Computing Base)[11]，Terra[10]，Perseus[13]等。它们的主要思想都是将计算平台分为可信区域与不可信区域。可信区域上运行高安全需求的 VM，甚至只能运行专门的安全操作系统。不可信区域提供与普通的虚拟化平台一样的环境。NGSCB 使用一种定制的 VMM，在可信区域只能运行可信操作系统 Nexus[12]。Terra 是原始 TCCP 模型采用的模型，它实现了一种可信虚拟机监视器（Trusted Virtual Machine Monitor, TVMM）提供两种环境，一种是白盒 VM（Open-Box VM）环境，另一种是黑盒 VM(Open-Box VM)环境。其中 TVMM 通过保护运行在黑盒环境中的虚拟机内存与存储，能够防止服务器管理员或平台所有者的恶意入侵，同时 TVMM 可以让黑盒环境中 VM 的应用向验证者远程证明自己的身份。

本文采用的虚拟可信平台是基于 vTPM[6]架构，并汲取 Terra 的思想（即黑盒环境）。vTPM 是 TPM 虚拟化解决方案之一，它使得在虚拟化环境里每一个虚拟机都能获得完整的可信计算功能，因此将 vTPM 改造成需要的虚拟可信平台难度不大。本文不直接使用 Terra 而选择采用改造 vTPM 的理由有 2 个方面：

1. Terra 的 TVMM 是基于 VMware GSX Server，而 vTPM 是基于 Xen 的。后者是开源软件而前者不是，因此使用 Xen 更适合研究 TCCP 模型。而且使用支持半虚拟化的 Xen 能够获得更少的性能损失。

2. Terra 只支持 TPM 1.1b 标准，本文需采用的 DAA 策略直到 1.2 标准才被 TPM 支持。而 vTPM 不存在这个问题。

2.1.1　Xen 架构

为了更好地了解 vTPM，首先简要介绍一下 Xen 的架构。如图 2.1 所示，在服务器物理硬件层上是软件层 Xen hypervisor，它负责抽象底层的硬件资源并为上层的 VM 提供虚拟资源，例如 vCPU，vMemory，共享内存资源等。运行在 Xen hypervisor 上的就是 VMs。最先被创建的 VM 也被称为 Domain0，由于 Xen hypervisor 中不包含任何与硬件对话的驱动，也没有与管理员对话的接口，这些驱动就由 Domain0 来提供了。Domain0 由 xend 进程负责管理客户 VM，包括创

建，关闭和迁移 VMs 等等。同时它具有访问物理 I/O 的权限，并同客户 VM 进行交互。客户 VM 即 DomainU 被创建时是由管理员使用配置程序与 Domain0 直接对话，客户 OS 一般是由 Linux OS，Windows OS 修改得来。

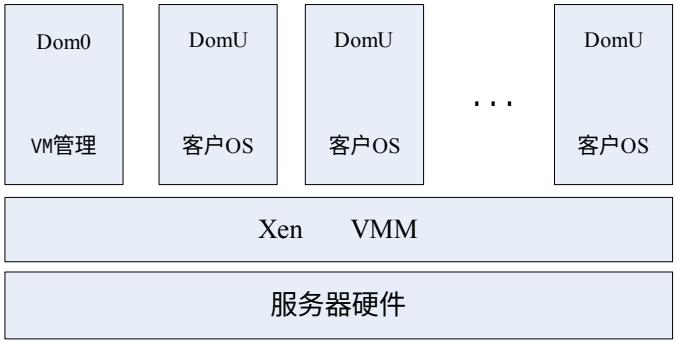


图 2.1　Xen 的架构

◆　攻击模型

目前大多数 IaaS 平台都采用 Xen 半虚拟化技术，从 Xen 的架构可以看出管理员控制着 Domain0，管理员可能通过在 Domain0 的特权获得对客户 VM 内存的访问权。目前就存在 Xenaccess 技术[14]可以使管理员在 Domain0 内运行一个用户级进程直接访问到客户 VM 的内存。正如 1.2.2 节所讨论的，下层 VMM 的安全可靠性决定了上层客户 VM 的安全性。因此 TCCP 模型需要确保 VM 不会被创建或迁移到不安全的服务器上，并且管理员无法访问客户 VM 的内存。

2.1.2　vTPM 架构

图 2.2 显示了基于 Xen 的 vTPM 架构[6]，vTPM 的完整功能是由 vTPM 管理者与 vTPM 实例协同完成，每一个 vTPM 实例都支持 TPM1.2 标准，并有独立的 EK，需要可信计算能力的 VM 必须与一个独立的 vTPM 实例相关联。通过服务器端 TPM 驱动给每条 TPM 命令包上添加 4bit 的 vTPM 实例标识的手段使得用户 VM 与 vTPM 一一对应，这样客户 VM 无法通过伪造命令包来与 vTPM 进行通信。在客户 VM 被休眠或被迁移时，与它关联的 vTPM 也一起被休眠或被迁移，这是为了能够保留已存在的完整性度量结果，避免了运行在一个新的环境下需要重新完整性度量的复杂性。因此这需要 vTPM 在转移时不能被修改或复制。vTPM 原型系统还设计了一种特权 vTPM 实例，它能够产生或管理子 vTPM 实例（子 vTPM 实例能够继承父 vTPM 的特殊权利），提供加解密，生成对称密钥，负责迁移 vTPM 的非对称密钥等功能，目的是为了迁移 vTPM 实例时能够加密 vTPM 实例的状态。

vTPM 实例的迁移与本文研究密切相关，因此简要介绍一下 vTPM 迁移的协

议：

1. 确定了用户 VM 迁移的目的平台后，为了迁移源 vTPM 实例的状态，首先在目的平台创建一个空 vTPM 实例，然后空 vTPM 产生一个 nonce 加密传给源平台。nonce 被确认通过之后，所有 vTPM 状态的转移都与 nonce 绑定，这避免源 vTPM 迁移到其他平台。

2. 源平台请求源 vTPM 实例产生一个对称密钥，这个对称密钥被父 vTPM 实例的存储密钥加密发送给目的平台。同时存储密钥也会发送给目的 vTPM 实例的父 vTPM 实例，这样目的平台就能解密得到这个对称密钥。

3. 源 vTPM 实例的状态例如密钥，证书，计数器等被对称密钥加密收集，当状态信息收集完毕，vTPM 实例被锁定不能进行任何更改了。同时加密后的状态会被 hash 得到一个迁移摘要。

4. 通过对称密钥的解密，目的 vTPM 实例获得了迁移 vTPM 实例的全部状态，重新进行 hash 计算后与迁移摘要核对以防止黑客攻击。

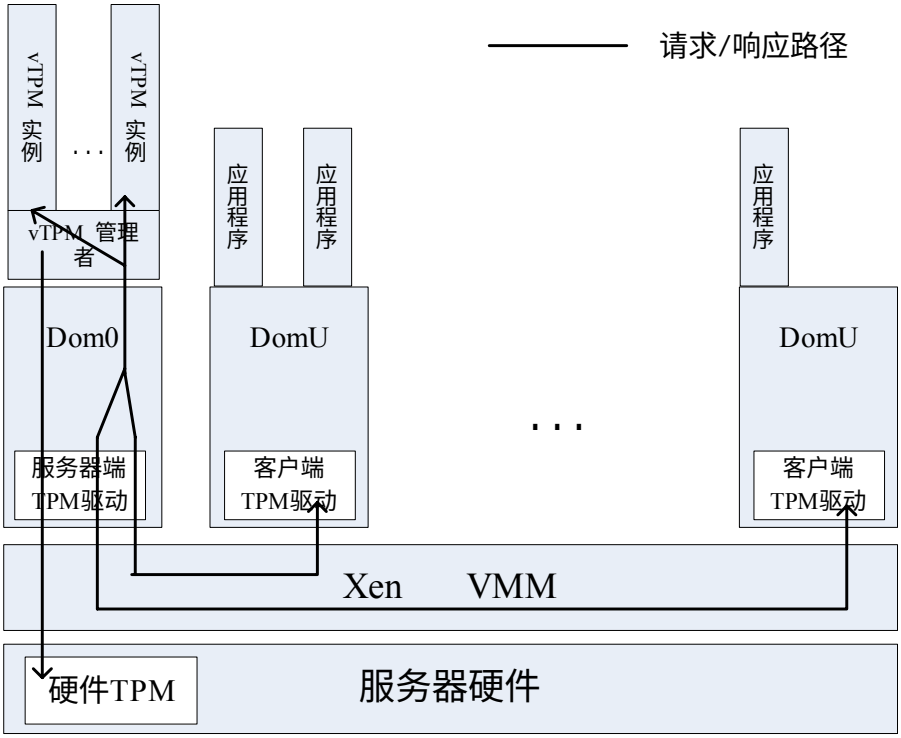


图 2.2　vTPM 架构

2.2　远程证明

在 1.3.1 节中简要介绍的远程证明协议中，TCP 使用身份密钥（EK 私钥）对平台完整性度量结果进行签名，如果验证者拥有 TCP 的 EK 公钥，他就能得到平台完整性度量结果。但是同时也带来一个隐私问题，由于 TPM 只拥有唯一

一个 EK，这样 TCP 也向验证者泄露了 TPM 的身份。为此 TCG 在 1.1 标准里面添加了基于可信第三方即 Privacy CA 策略的 AIK（Attestation Identity Key）方案。在 AIK 方案中，假设 Privacy CA 知道所有有效 TPM 的 EK 公钥。远程证明协议如下：

1. 当 TCP 需要向验证者证明自己时，首先生成一个不对称密钥对即 AIK，然后使用 EK 私钥对 AIK 公钥进行签名，并发给 Privacy CA；

2. 如果 Privacy CA 能够利用列表里面的 EK 公钥成功获得 AIK 公钥，它会就 AIK 公钥发布证书。然后使用 Privacy CA 的私钥证书进行签名，并发给 TCP；

3. TCP 会将平台完整性度量结果使用 AIK 私钥加密，与 Privacy CA 发布的证书一同发给验证者；

4. 验证者首先使用 Privacy CA 公钥解密获得 AIK 公钥的证书，然后使用 AIK 公钥解密得到 TCP 的完整性度量结果。并根据完整性度量结果判断 TCP 的完整性。

AIK 策略完全隐藏了 TPM 的身份，并且根据 TPM 与验证者的不同交流生成不同的短期 AIK，连 AIK 可能属于哪个 TPM 的信息都隐藏了。

在这个策略中无赖 TPM 有两种被发现的方式[15]：

1) 如果 TPM 被攻陷，里面的 EK 私钥被发布出来，Privacy CA 会计算对应的 EK 公钥，并将其从列表中移除；

2) 如果 Privacy CA 收到来自同一个 EK 签名的请求数超过一定的阈值，它会回绝这些请求，并将这个 EK 从列表删除。

不过 AIK 策略的缺陷与原始 TCCP 模型相同，Privacy CA 不得不参与每一项远程证明过程并需要一直保持高可用性。

2.2.1　DAA 策略

为了解决 AIK 策略的缺陷，直接匿名证明（Direct Anonymous Attestation，DAA）策略由 Ernie Brickell et al[15]提出，并且之后被纳入 TPM 1.2 标准。

如图 2.3 所示，DAA 策略里面一般有 3 个参与实体：

1) DAA 发证者，它通过 Join 协议发行 DAA 证书给证明者的第三方机构，所有 DAA 策略的参与者都必须信任它。对于 DAA 发证者发布的公钥，一般会由证书中心的私钥证明它的真实性。

2) 证明者，它根据 DAA 证书使用 Sign 协议生成的 DAA 签名被验证者检验，在这里证明者是 TPM；

3) 验证者，它验证证明者生成的签名。

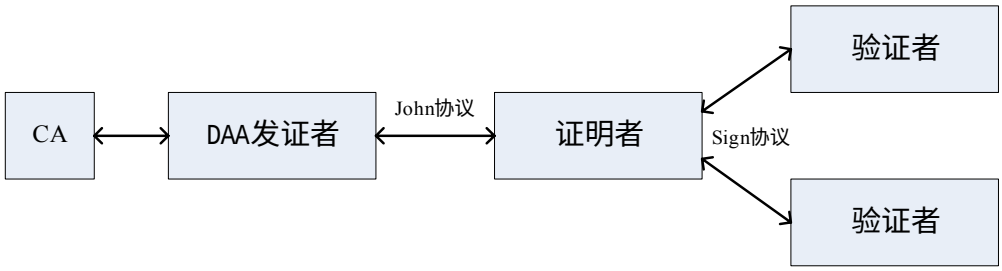


图 2.3　DAA 策略模型

DAA 策略属于一种特殊的群签名策略，它的主要作用是证明者能够匿名不向验证者出示 DAA 证书的同时，让验证者相信证明者拥有由 DAA Issuer 发布的 DAA 证书，这里需要使用零知识证明。

为了向验证者证明拥有 DAA 证书，DAA 策略使用了 Camenisch-Lysyanskaya 签名策略[16]和基于离散对数的证明。DAA 证书的无法伪造性是基于强 RSA 假定，其隐私性与匿名性由决策性 Diffie-Hellman 假定保证。DAA 策略还使用 Fiat-Shamir 启发式[17]将非交互式零知识证明变成签名。

2.2.1.1　预备知识

◆　预定义

首先对符号与参数进行预定义，这些参数在以后的协议中需要用到。

符号：

$\{0,1\}^{\lambda}$：长度为 λ 的二进制字符串集合，表示 $[0,2^{\lambda}-1]$ 的整数；

$LSB_u(x)$：整数 x 对应的二进制串低 u 位二进制串代表的整数；

$CAR_u(x)$：证书 x 对应的二进制串去掉低 u 位得到的二进制串所代表的整数；

$PK\{(\alpha,\beta,\gamma): y=g^{\alpha}h^{\beta} \wedge \tilde{y}=\tilde{g}^{\alpha}\tilde{h}^{\gamma} \wedge (u\le\alpha\le v)\}$：通过 $y=g^{\alpha}h^{\beta}$，$\tilde{y}=\tilde{g}^{\alpha}\tilde{h}^{\gamma}$ 且 $u\le\alpha\le v$ 零知识证明 α，β，γ，这里 y，g，h，$\tilde{y}$，$\tilde{g}$ 和 $\tilde{h}$ 为群 $G=\langle g\rangle=\langle h\rangle$ 和 $\tilde{G}=\langle\tilde{g}\rangle\langle\tilde{h}\rangle$ 的元素。除了需要证明的希腊字母代表的知识外，其余参数对于验证者都是可知的；

$SPK\{(\alpha): y=g^{\alpha}\}(m)$：表示通过 Fiat-Shamir 启发式[17][19]将 PK 零证明变成签名，m 代表被签名的信息；

安全参数：

$\lambda_n(2048)$：RSA 密钥长度；

$\lambda_f(104)$：证书编码信息 f_i 的长度；

$\lambda_e(368)$：证书里面的指数 e 长度；

$\lambda'_e(120)$：选取指数 e 的区间长度；

$\lambda_v(2536)$　：证书里面的随机值 v 的长度；

$l_{\varnothing}(80)$：控制静态零知识属性的安全参数；

$\lambda_{\mathrm{H}}(160)$：Fiat-Shamir 启发式使用的 hash 函数的输出长度；

$\lambda_r(80)$：减少安全证明需要的安全参数；

$\lambda_s(1024)$：为了简化 TPM 运算大指数被分解后的长度；

$\lambda_{\Gamma}(1632)$：模数 Γ 的长度；

$\lambda_p(208)$：用于检验无赖 TPM 的 $\mathbf{Z}_{\Gamma}^*$ 的子群的阶的长度；

其中 H 为抗碰撞 hash 函数 $\mathrm{H}:\{0,1\}^*\to\{0,1\}^{l_{\mathrm{H}}}$，这里面有 $\mathrm{H}_{\Gamma}(\cdot):\{0,1\}^*\to\{0,1\}^{l_{\Gamma}+l_{\varnothing}}$，$\mathrm{H}(\cdot):\{0,1\}^*\to\{0,1\}^{\lambda_{\mathrm{H}}}$。另外需要

$\lambda_p=2\lambda_f$，

$l_e>l_{\varnothing}+l_{\mathrm{H}}+\max\{l_f+4,l'_e+2\}$，

$l_v>l_n+l_{\mathrm{H}}+\max\{l_f+l_r+3,l_{\varnothing}+2\}$。

◆ 密码学假定

DAA 策略的安全性保证是基于强 RSA 假定与决策性 Diffie-Hellman 假定。

假定 1（强 RSA 假定）：对于给定一个随机 RSA 模数 n 以及从 $\mathbf{Z}_n^*$ 中随机选取一个元素 u，找出 e（$e>1$）和 v 使得 $v^e\equiv u(\mathrm{mod}\,n)$ 的问题是难于求解的。

假定 2（DDH 假定）：对于阶为 q 生成元为 g 的循环群 G，随机选取 $a,b,c\in\mathbf{Z}_q$，分布 (g^a,g^b,g^{ab}) 与 (g^a,g^b,g^c) 是计算上不可区分的。

◆ 零知识证明

零知识证明是指证明者除了向验证者提供声明本身之外不提供其他有价值消息，仍能让验证者相信它的声明是正确的。若证明者的声明为假，只有极小概率让验证者相信声明为真。

使用图灵机的形式化定义[18]如下，假设 P，V 和 S 为图灵机，L 是 $\{0,1\}^*$ 上的语言，交互式证明系统的计算模型使用 (P,V) 进行描述，其中 P 为证明者，V 为验证者，P 的计算能力是无限的。(P,V) 被称为 L 的零知识如果对于任意概率多项式时间验证者 $\tilde{\mathrm{V}}$ 存在期望的概率多项式时间模拟器 S 使得

$\forall x \in L, z \in \{0,1\}^*, View_{\tilde{V}}[P(x) \leftrightarrow \tilde{V}(x,z)] = S(x,z)$ 。

直观的意思是交互式证明系统(P,V)是零知识的，如果对于任意验证者$\tilde{V}$存在一个有效率的对于任意输入能重现P与$\tilde{V}$对话的模拟器S。这里辅助串z扮演先验知识的角色，暗示了$\tilde{V}$不能利用任何先验知识从与P的对话中挖掘出有价值信息，因为我们需要的是如果S也被给予先验知识，那么它仍然能重现P与$\tilde{V}$对话。

Fiat-Shamir 启发式[17]使用基于随机预言机的 hash 函数能够将一般需 3 次消息交换的知识证明转变成一个非交互式的零知识证明，以$PK\{(\alpha): A = g^{\alpha}\}$为例，Schnorr 协议[20]需要证明者与验证者之间 3 次消息交换：

1) 首先证明者选择$t \in \mathbf{Z}_q$，并计算$T = g^x$，发给验证者；
2) 验证者选择$c \in \mathbf{Z}_q$，发给证明者；
3) 证明者计算$s = xc + t(\mathrm{mod}\, q)$，发给验证者；
4) 验证者接受证明者的声明当且仅当$g^s = A^c T$。

如果使用了 Fiat-Shamir 启发式，首先声明者计算$T = g^t$，然后计算$c = \mathrm{H}(T)$，最后计算$s = xc + t$并发送(T,s)给验证者，验证者接受证明者的声明当且仅当$g^s = A^c T$。

◆ Camenisch-Lysyanskaya 签名策略[16]

DAA 策略的基石是 Camenisch-Lysyanskaya（CL）签名策略，它允许有效协议进行签名知识证明，并且根据基于知识证明的离散对数有效恢复签名的秘密信息。CL 签名在强 RSA 假定下能够对抗适应性选择消息攻击。

最简单的针对单条消息签名的 CL 协议如下（针对消息块加密的正式 CL 协议请参见原文[16]）：

密钥生成：对于输入1^k，选择特殊 RSA 模数n使得$n = pq$，$p = 2p' + 1$，$q = 2q' + 1$且$\lambda_n = 2k$。一致地随机选取$a,b,c \in QR_n$，输出$PK = (n,a,b,c)$，$SK = p$。

消息空间：消息空间包括所有长度为λ_m的二进制串。

签名算法：对于输入m，选择一个长度为$\lambda_e = \lambda_m + 2$的随机素数$e > 2^{\lambda_m + 1}$和一个长度为$\lambda_s = \lambda_n + \lambda_m + l$（这里$\lambda$是一个安全参数）。计算$v$使得

$$v^e \equiv a^m b^s c(\mathrm{mod}\, n)$$

验证算法：验证(e,s,v)为消息m的签名，只需查看$v^e \equiv a^m b^s c(\mathrm{mod}\, n)$且$2^{\lambda_e} > e > 2^{\lambda_e - 1}$。

2.2.1.2　DAA 策略细节分析

DAA 策略的思想是证明者 TPM 选择秘密信息f，通过与 DAA 发证者的安全通信获得秘密消息f的 CL 签名，然后通过对 CL 签名的知识证明让验证者确

信它获得了匿名证明。

整个策略由 Join 协议与 Sign 协议共同完成，由于效率方面的考虑秘密信息 f 被分成了 2 个 λ_f 位的消息 f_0 和 f_1。而且由于 TPM 计算能力的有限，证明者的工作由 TPM 和主机（host）共同完成，这需要 host 也是诚实的，因为 host 总能揭示 TPM 的身份。

在 Join 协议中首先 TPM 创建出秘密信息 f 和一个盲因子 v'。然后发送给 DAA 发证者一个关于 (f_0, f_1) 的承诺 $U := blind(f_0, f_1, v')$，$N_I := \zeta_I^f$（$\zeta_I := (hash(1 \| bsn_I))^{(\Gamma-1)/\rho} (\mathrm{mod}\,\Gamma)$ 是根据 DAA 发证者的长期的基名（basename）生成的）。DAA 发证者检查 N_I 是否由无赖 TPM 计算，若确认是有效 TPM，DAA 发证者要求 TPM 发送关于 U 和 N_I 正确性的知识证明。若 DAA 发证者确认 U 和 N_I 的正确性，会计算

$A := (\frac{Z}{US^{v''}})^{1/e} (\mathrm{mod}\, n)$。，将 (A, e, v'') 和关于 A 计算正确性的证明发给 TPM。关于秘密信息 f 的签名就是 $(A, e, v := v' + v'')$，这里只有 v 必须保密。

在 Sign 协议里 TPM 就可以通过证明它有关于秘密信息 f 的 CL 签名 (A, e, v) 的方式达到远程证明的目的。

这里有几处细节需要深入理解。

❖ 无赖 TPM 检测

DAA 策略只能由有效 TPM 参与，为了检测无赖 TPM，每一个 TPM 都有一个化名 N_v，这里 N_v 由 ζ 和秘密信息 f 决定（$N_v := \zeta^{f_0 + f_1 2^{\lambda_f}}$）。一般的 ζ 可以由 TPM 自由选定（$\zeta \in_R \langle \gamma \rangle$）或者根据验证者的基名生成（$\zeta := (H_\Gamma(1 \| bsn_v))^{(\Gamma-1)/\rho}$），若 ζ 是自由选定的，验证者无法进行无赖 TPM 检测，所以在 DAA 策略里 ζ 是根据验证者的基名生成（若 ζ 是个定值会带来隐私问题，因此验证者应该定期更换基名）。验证者检测无赖 TPM 有两种方式：1）若无赖 TPM 被发现，且 TPM 中 f_0 与 f_1 被提取并发布出来，验证者会将 (f_0, f_1) 放入黑名单（记为 $(\hat{f}_0, \hat{f}_1)$）。验证者可以通过比较 N_v 与 $\zeta^{\hat{f}_0 + \hat{f}_1 2^{\lambda_f}}$ 判断无赖 TPM；2）N_v 出现的次数过多。另外 DAA 发证者可以通过比较 N_I 与 $\zeta_I^{\hat{f}_0 + \hat{f}_1 2^{\lambda_f}}$ 判断无赖 TPM。

若证书 (A, e, v) 与相关的 (f_0, f_1) 被发布出来，验证者通过查看 $A^e R_0^{f_0} R_1^{f_1} S^v \stackrel{?}{\equiv} Z (\mathrm{mod}\, n)$，若成立验证者将 (f_0, f_1) 放入黑名单。

❖ 活动连接

由于在 DAA 策略里 ζ 不采用随机选取的方案，对于同一个验证者来说，TPM 产生的 N_v 是固定的，因此验证者虽然不能了解 TPM 的身份，但是能够将同一 TPM 的不同活动连接到一起。这样会降低 DAA 策略带来的隐私保护性能。

为了解决这个问题，Jan Camenisch[22]修改了 DAA 策略使得修改后 DAA 策略的隐私保护性能与 Privacy CA 策略相同，同时又能检测无赖 TPM。对这个协议稍作修改后在本文研究中可以用于不同的用途（本文同时使用原始 DAA 策略）。在这里 Jan Camenisch 使用了 Privacy CA 的概念，而此时 Privacy CA 的作用与 Privacy CA 策略里面不同（本文提出的 Privacy CA 又与此处的概念不同），它验证证明者从 DAA 发证者获得的证明证书，在此基础上发布匿名的一次性证书（又称频率证书，Privacy CA 的输入规定了证明者允许获得证书的频率），此证书与 DAA 证书通过同一个值 k_t 绑定（目的是使一次性证书与证明者绑定，无法被迁移用作其他证明者的一次性证书）。当证明者需要向验证者证明身份时（这里使用的基 ζ 是随机的，目的是使得验证者连接证明者不同活动的动作不可行），会使验证者确信它获得了有效的证明证书，Privacy CA 发布的一次性证书以及证明证书与一次性证书通过一个共同的值 k_t 绑定。

因此在 Join 协议里面，DAA 发证者还需要对 k_t 签名，对于证明者保存的证书 (A,e,v) 变为应满足 $A^e R_0^{f_0} R_1^{f_1} R_2^{k_t} S^v \equiv Z(\mathrm{mod}\, n)$，这里 R_2 是 DAA 发证者公钥里面的新参数。

在发布一次性证书的协议中，Privacy CA 首先需要验证证明者的证明证书，因此这个协议可以看作 DAA 的 Join 协议，Sign 协议与 verify 算法的结合。

❖ 密钥生成涉及的零知识证明

和 CL 签名策略一样，DAA 策略的开始步骤也是密钥生成（DAA 发证者的密钥生成）。

DAA 发证者密钥生成协议结束后会得到公钥 $(n,g',g,h,S,Z,R_0,R_1,\gamma,\Gamma,\rho)$ 和私钥 $p'q'$，在这过程中使用了 Fiat-Shamir 启发式生成了非交互零知识证明让证明者确认密钥参数中 R_0，R_1，S，Z，g 和 h 的正确生成，目的是保证签名的保密性与匿名性。其证明生成过程如下：

1) DAA 发证者随机选择

$\widetilde{x}_{(g,1)},...,\widetilde{x}_{(g,\lambda_{\mathrm{H}})} \in_R [1, p'q']$，$\widetilde{x}_{(h,1)},...,\widetilde{x}_{(h,\lambda_{\mathrm{H}})} \in_R [1, p'q']$，$\widetilde{x}_{(s,1)},...,\widetilde{x}_{(s,\lambda_{\mathrm{H}})} \in_R [1, p'q']$，$\widetilde{x}_{(z,1)},...,\widetilde{x}_{(z,\lambda_{\mathrm{H}})} \in_R [1, p'q']$，$\widetilde{x}_{(0,1)},...,\widetilde{x}_{(0,\lambda_{\mathrm{H}})} \in_R [1, p'q']$，$\widetilde{x}_{(1,1)},...,\widetilde{x}_{(1,\lambda_{\mathrm{H}})} \in_R [1, p'q']$，并计算

$\widetilde{g}_{(g,i):} = g'^{\widetilde{x}_{(g,i)}}(\mathrm{mod}\, n)$，$\widetilde{h}_{(h,i):} = g'^{\widetilde{x}_{(h,i)}}(\mathrm{mod}\, n)$，$\widetilde{S}_{(s,i):} = h^{\widetilde{x}_{(s,i)}}(\mathrm{mod}\, n)$，$\widetilde{Z}_{(z,i):} = h^{\widetilde{x}_{(z,i)}}(\mathrm{mod}\, n)$，$\widetilde{R}_{(0,i):} = S^{\widetilde{x}_{(0,i)}}(\mathrm{mod}\, n)$，$\widetilde{R}_{(1,i):} = S^{\widetilde{x}_{(1,i)}}(\mathrm{mod}\, n)$，这里 $i = 1,...,\lambda_{\mathrm{H}}$；

2) DAA 发证者计算

$c := H(n,g',g,h,S,Z,R_0,R_1,\widetilde{g}_{(g,1)},...,\widetilde{g}_{(g,\lambda_{\mathrm{H}})},\widetilde{g}_{(h,1)},...,\widetilde{g}_{(h,\lambda_{\mathrm{H}})},$

$\widetilde{S}_{(s,1)},...\widetilde{S}_{(s,\lambda_{\mathrm{H}})},\widetilde{Z}_{(z,1)},...,\widetilde{Z}_{(z,\lambda_{\mathrm{H}})},\widetilde{R}_{(0,1)},...,\widetilde{R}_{(0,\lambda_{\mathrm{H}})},\widetilde{R}_{(1,1)},...,\widetilde{R}_{(1,\lambda_{\mathrm{H}})})$；

3) DAA 发证者计算

$\hat{x}_{(g,i)} := \tilde{x}_{(g,i)} - c_i x_g \pmod{p'q'}$，$\hat{x}_{(h,i)} := \tilde{x}_{(h,i)} - c_i x_h \pmod{p'q'}$，

$\hat{x}_{(s,i)} := \tilde{x}_{(s,i)} - c_i x_s \pmod{p'q'}$，$\hat{x}_{(z,i)} := \tilde{x}_{(z,i)} - c_i x_z \pmod{p'q'}$，

$\hat{x}_{(0,i)} := \hat{x}_{(0,i)} - c_i x_0 \pmod{p'q'}$，$\hat{x}_{(1,i)} := \hat{x}_{(1,i)} - c_i x_1 \pmod{p'q'}$；

4) DAA 发证者发布证明

$proof := (c, \hat{x}_{(g,1)},..., \hat{x}_{(g,\lambda_H)}, \hat{x}_{(h,1)},..., \hat{x}_{(h,\lambda_H)}, \hat{x}_{(s,1)},...,$

$\hat{x}_{(s,\lambda_H)}, \hat{x}_{(z,1)},..., \hat{x}_{(z,\lambda_H)}, \hat{x}_{(0,1)},..., \hat{x}_{(0,\lambda_H)}, \hat{x}_{(1,1)},..., \hat{x}_{(1,\lambda_H)})$。

证明者收到$(n, g', g, h, S, Z, R_0, R_1, \gamma, \Gamma, \rho)$与 *proof* 后，验证参数$R_0$，$R_1$，$S$，$Z$，$g$和$h$的过程如下：

1) 证明者计算

$\hat{g}_{(g,i):} = g^{c_i} g'^{\hat{x}_{(g,i)}} \pmod{n}$，$\hat{h}_{(h,i):} = h^{c_i} g'^{\hat{x}_{(h,i)}} \pmod{n}$，$S_{(s,i):} = S^{c_i} h^{\hat{x}_{(s,i)}} \pmod{n}$，

，$\hat{R}_{(0,i):} = R_0^{c_i} S^{\hat{x}_{(0,i)}} \pmod{n}$，$\hat{R}_{(1,i):} = R_1^{c_i} S^{\hat{x}_{(1,i)}} \pmod{n}$，

这里$i = 1,..., \lambda_H$，c_i为c的第i bit；

2) 验证

$$c :\stackrel{?}{=} H(n, g', g, h, S, Z, R_0, R_1, \hat{g}_{(g,1)},..., \hat{g}_{(g,\lambda_H)}, \hat{h}_{(h,1)},..., \hat{h}_{(h,\lambda_H)},$$

$$\hat{S}_{(s,1)},...\hat{S}_{(s,\lambda_H)}, \hat{Z}_{(z,1)},..., \hat{Z}_{(z,\lambda_H)}, \hat{R}_{(0,1)},..., \hat{R}_{(0,\lambda_H)}, \hat{R}_{(1,1)},..., \hat{R}_{(1,\lambda_H)}) ,$$

如果验证通过就能证明$g, h \in \langle g' \rangle$，$S, Z \in \langle h \rangle$且$R_0, R_1 \in \langle S \rangle$。

❖ Join 协议涉及的零知识证明

Sign 协议涉及到的零知识证明签名有两处，开始 TPM 需要向 DAA 发证者证明U和N_I的正确性以及秘密信息f_i在合适的区间内，即

$$SPK\{(f_0, f_1, v') : U \equiv \pm R_0^{f_0} R_1^{f_1} S^{v'} \pmod{n} \wedge$$

$$N_I \equiv \zeta_I^{f_0 + f_1 2^{l_f}} \pmod{\Gamma} \wedge f_0, f_1 \in \{0,1\}^{l_f + l_\varnothing + l_H + 2} \wedge v' \in \{0,1\}^{l_n + l_\varnothing + l_H + 2}\}(n_t \| n_i) :$$

1) TPM 选择随机整数$r_{f_0}, r_{f_1} \in_R \{0,1\}^{l_f + l_\varnothing + l_H}$和$r_{v'} \in_R \{0,1\}^{l_n + 2l_\varnothing + l_H}$，计算$\tilde{U} := R_0^{r_{f_0}} R_1^{r_{f_1}} S^{r_{v'}} \pmod{n}$和$\tilde{N}_I := \zeta_I^{r_{f_0} + r_{f_1} 2^{\lambda_f}} \pmod{\Gamma}$，并发送给 host；

2) DAA 发布者选择随机二进制串$n_i \in \{0,1\}^{\lambda_H}$，并发送给 host；

3) host 计算$c_h := HN(n \| R_0 \| R_1 \| S \| U \| N_I \| \tilde{U} \| \tilde{N}_I \| n_i)$，并发送给 TPM；

4) TPM 随机选择$n_t \in \{0,1\}^{l_\varnothing}$，并计算$c := H(c_h \| n_t) \in [0, 2^{\lambda_H} - 1]$；

5) TPM 计算$s_{f_0} := r_{f_0} + c \cdot f_0$，$s_{f_1} := r_{f_1} + c \cdot f_1$和$s_{v'} := r_{v'} + c \cdot v'$，发送$(c, n_t, s_{f_0}, s_{f_1}, s_{v'})$给 host；

6) host 将$(c, n_t, s_{f_0}, s_{f_1}, s_{v'})$发送给 DAA 发证者；

7) DAA 验证签名通过计算$\hat{U} := U^{-c} R_0^{s_{f_0}} R_1^{s_{f_1}} S^{s_{v'}} \pmod{n}$和$\hat{N}_I = N_I^{-c} \zeta_I^{s_{f_0} + 2^{l_f} s_{f_1}} \pmod{n}$，并判断

$c \stackrel{?}{=} H(H(n \| R_0 \| R_1 \| S \| U \| N_I \| \hat{U} \| \hat{N}_I \| n_i) \| n_t)$ ，$s_{f_0}, s_{f_1} \stackrel{?}{\in} \{0,1\}^{1_f+1_\varnothing+1_H+1}$ 和 $s_{v'} \stackrel{?}{\in} \{0,1\}^{1_f+2l_\varnothing+1_H+1}$。

另一个是DAA发证者向TPM证明A的计算正确性(同时Join协议要求DAA发证者证明A在$\langle h \rangle$内)，即

$SPK\{(d): A \equiv (\frac{Z}{US^{v''}})^d (\bmod n)\}(n_h)$：

1) host 选择随机整数 $n_h \in \{0,1\}^{1_\varnothing}$ 并发送给 DAA 发证者；

2) DAA 发证者随机选择 $r_e \in_R [0, p'q']$，并计算 $\tilde{A} = (\frac{Z}{US^{v''}})^{re} (\bmod n)$，$c' := H(n \| Z \| S \| U \| v'' \| A \| \tilde{A} \| n_h)$ 和 $s_e := r_e - c'/e (\bmod p'q')$，并将 c'，s_e 和 (A, e, v'') 发送给 host；

3) host 检验 e 是否为素数，且在 $[2^{\lambda_e-1}, 2^{\lambda_e-1} + 2^{\lambda'_e-1}]$ 内，并计算 $\hat{A} := A^{c'} (\frac{Z}{US^{v''}})^{s_e} (\bmod n)$，并检验 $c' \stackrel{?}{=} H(n \| Z \| S \| U \| v'' \| A \| \hat{A} \| n_h)$。

❖ Rudolph 攻击模型

Rudolph 攻击[21]的原理是在 Join 协议中 DAA 发证者对于不同的证明者产生不同的短期公钥 PK_I，当验证者与 DAA 发证者合谋时，在 Sign 协议里面验证者可以从证明者那里获取 PK_I，验证者会将此 PK_I 发往 DAA 发证者，DAA 发证者根据 (EK, PK_I) 的映射表能够获知 TPM 的身份，作为合谋者的验证者就能够确定证明者的身份了。

2.3 安全多方计算背景知识

本文提出的改进 TCCP 模型设计中引入了安全多方计算协议（Secure Multi-party Computation），下面简略介绍一下 SMC 的背景知识。

安全多方计算[27]的概念与零知识证明相似，参与者都希望能够获得需要的结论却不想暴露自己掌握的信息。在无可信第三方的情况下，多个参与者根据他们各自掌握的信息 $(x_1, x_2, x_3, ...)$ 联合计算一个函数 $f(x_1, x_2, x_3, ...)$ 以获得想要的结果，在这个过程中不希望自己的信息 x_i 被其他参与者掌握。

根据攻击者计算能力的不同区分 SMC 协议的安全程度，在攻击者计算能力无限的情况下 SMC 协议是安全的，那么此 SMC 协议为无条件安全；在攻击者计算能力为多项式级别的情况下 SMC 协议是安全的，那么此 SMC 协议为条件安全。对于无条件安全模型下与条件安全模型，恶意参与者数目分别占据 1/3 和 1/5 以上，不存在安全的 SMC 方案。

2.4　Eucalyptus 架构

IaaS 服务提供商像 Amazon EC2 允许用户按需支配整个 VM，其中 Eucalyptus 是 IaaS 平台的开源实现。如图 2.3 所示[3]，为了反映资源拓扑 Eucalyptus 平台使用了层次性架构，因为 CSP 为了降低数据中心的成本，数据中心会建立在有利于使用节能技术的各种区域，因此 CSP 使用“区域”（Zone）来表示数据中心。

整个 Eucalyptus 架构包括 4 种角色：Node Controller（NC），Cluster Controller（CC），Cloud Controller（CLC）和用户。这里 NC 角色其实就是 VMM，负责启动，休眠，关闭和迁移 VM，它有一个称为 *describeResource* 的操作负责汇报现在物理资源状况。CC 是能同时连接内网和外网的服务器，它负责收集 Zone 里面的所有 NC 汇报的资源状态，根据 NCs 的资源状态处理 VM 创建与迁移请求。而架构里面唯一的 CLC 是用户入口点以及全局决策分析组件，它负责处理用户与管理员的请求，做出高层 VM 调度决策，处理服务水平协议以及维护持久的系统与用户元数据。

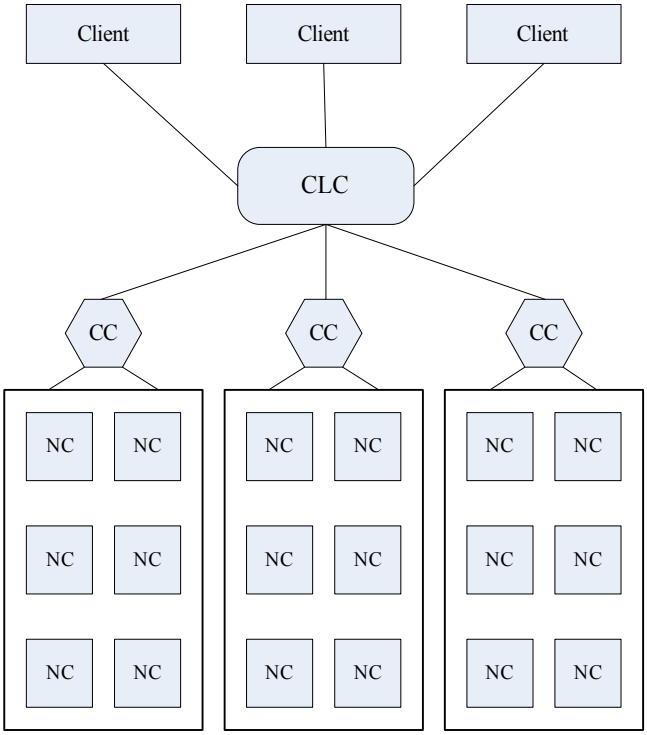


图 2.3　Eucalyptus 使用了层次结构设计

2.5　本章小结

本章介绍了新型 TCCP 模型设计所需的背景知识，首先是讨论了虚拟可信平

台架构，在比较不同的平台架构以及相关背景知识后选择了 vTPM 架构作为本文研究的虚拟可信平台架构的基础。为了解决原始 TCCP 的性能瓶颈问题，本文引入了 DAA 策略，为此本章深入讨论了与 DAA 策略相关的零知识证明与 CL 签名的知识（在后面具体的协议设计中本文将不再对使用零知识证明的细节做更多说明），并对 DAA 策略设计里面需要注意的方面做了深入探讨。接着简要描述了安全多方计算的背景知识，最后本文讨论了 Eucalyptus 的平台架构，并将其作为本文云平台的架构的基础。

第 3 章　一种基于 DAA 与 Privacy CA 策略的 TCCP 模型

本章提出了一种基于 DAA 与 Privacy CA 策略的改进 TCCP 模型旨在解决原始 TCCP 中的性能瓶颈问题同时能很好地保护用户 VM 的数据不被 CSP 内部恶意特权员工窥探或篡改。

3.1　TCCP 模型总体架构设计

本文在 1.3.2 节讨论过原始 TCCP 平台的局限性，即 TCCP 模型的性能与安全性瓶颈为 ETE 内部的 TC。本文解决瓶颈问题的思路是利用可信芯片的中立特性，将 TC 管理 TNs 的功能移到云端内部的 TN 上，即云端有一个内部 TC（Internal TC，ITC）承担了 ETE 内部 TC 的管理职能，这里 ITC 的角色显然需要从 TNs 之间选举产生。而由于单个服务器平台性能与资源的有限性再加上云端一个 Zone 内部服务器数量也具有大规模性，因此单个 TN 无法胜任 ITC 角色的工作量。本文将 ITC 角色的工作量分散给多个 TN,每个 TN 能够承担合适的工作量，本文称这样的 TN 为 Privacy CA。Privacy CA 会管理 Zone 内一部分 TNs 列表，负责与之相关的 VM 创建与迁移的工作并收集 TNs 的可用资源信息以便确定 VM 创建与迁移的目的 TN。

由于承担 Privacy CA 角色的 TNs 之间有不同的硬件配置，Privacy CA 的性能也会不一致，一般地假定 Privacy CA 能够管理的 TNs 最大的数量为 $n_{\max}$（由于平台运行时可用资源是动态变化的，$n_{\max}$ 也处于变动中），至少能管理的数量为 $n_{\min}$（Privacy CA 管理 TNs 数量的下限），目前管理的 TNs 的数量为 n_{TN}，如何确定 $N_{\max}$ 与 $n_{\min}$ 的值本文不作讨论。

设置多个 Privacy CA 的方案在安全和性能优化方面考虑还能提供一种隔离机制，在性能优化方面：1）VM 迁移目的地优先考虑在同一 Privacy CA 管理区域内（除非 Privacy CA 发现自己的区域内 TNs 都没有足够的资源供 VM 迁移），一般情况下不需要在整个 Zone 的区域内迁移；2）多个 Privacy CA 能够并行行使 TC 的职责，显著提高 TCCP 模型的效率；3）单个 Privacy CA 无法正常运行时，影响不会带到整个 Zone，其他管理区域仍然能够正常运作。在安全方面：一旦 Privacy CA 被攻破，破坏力影响只限在其管理的区域。

虽然 Privacy CAs 承担了 ITC 的职责，ITC 的角色在本文仍然是需要的，因为需要有角色将同一 Zone 内的 Privacy CA 管理区域联系起来。ITC 就被委派负责联络各个 Privacy CA，收集不同管理区域还能够接纳的 TNs 数量

$n_{available} := n_{\max} - n_{TN}$，还能够分配给 VM 的剩余资源概况以及负责指定 VM 创建的目的区域与整个 Zone 范围内跨区域的 VM 迁移。ITC 由 Privacy CAs 联合选举出来，本文采用了 SMC 算法以保证无赖 Privacy CAs（被攻陷然后被操纵的 Privacy CAs）不会操纵选举结果，该结论必须在无条件安全模型下无赖 Privacy CAs 数目小于总数 1/3，条件安全模型下无赖 Privacy CAs 数目小于总数 1/5 的时候才成立。

在云端内部 Privacy CAs 与 ITC 完全替代 ETE 内 TC 的 TNs 管理功能后，TC 的负责 TN 可信性认证的角色在本文中仍然得以保留。因为 TC 一般为可信芯片制造商，只有 TC 能够判断 TPM 的 EK 是否有效，在本文中 TC 即为 DAA 发证者。另外 TC 负责给 TN 指定 Privacy CA 角色，之后 Privacy CAs 会在云端内部选举出 ITC 并向 TC 证明选举结果的正当性。

若 ITC 选举结果被承认，ITC 会取得与 TC 的联系并向 TC 汇报不同 Privacy CA 的 $n_{available} := n_{\max} - n_{TN}$，TC 会根据 $n_{available}$ 确定是否给 TN 指定 Privacy CA 角色。以图 3.1 所示为例，最左边的状态表明的是 Privacy CA 至少能管理的 TNs 数量 $n_{\min}$，在 Zone 中已有两个 Privacy CA 正在运行，他们管理的 TNs 数量已经达到 $n_{\max}$，ITC 会将这一情况汇报给 TC，当新的 TN 向 TC 注册证实自己含有有效的 TPM 后，TC 会指定它承担 Privacy CA 角色。

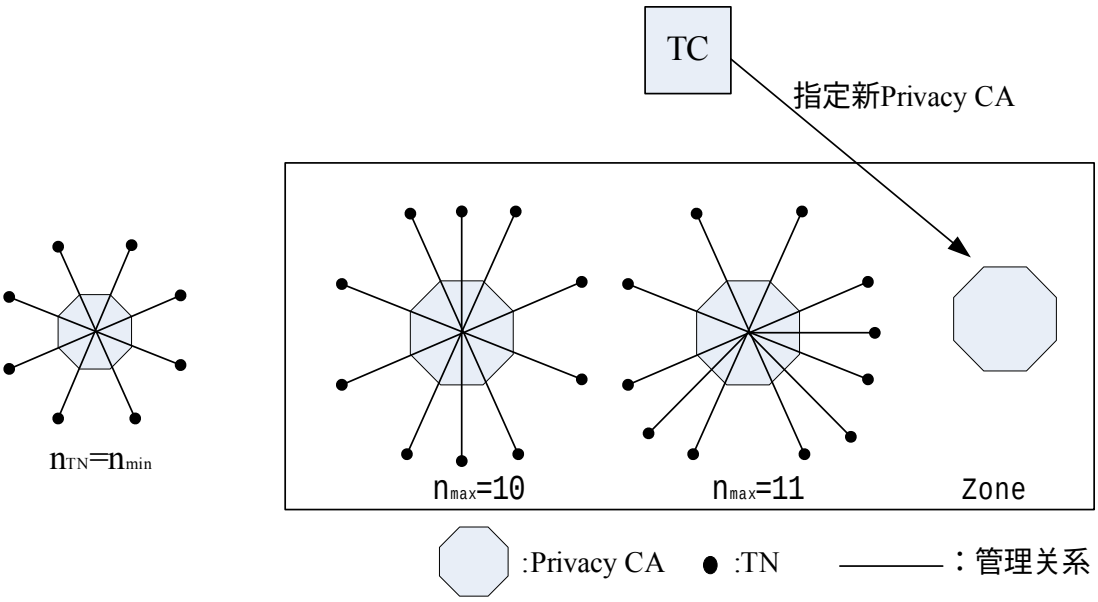


图 3.1　TC 指派 Privacy CA 的依据

然而在整个 Zone 区域刚刚启动到 ITC 被选举出来之间，ITC 无法通过反馈的方式让 TC 决定 Privacy CA 角色的指派。本文采用的方案是在这段特殊时期内 TC 指派 Privacy CA 管理的 TNs 数量为定值 $n_{\min}$。等 ITC 被选举出来，整个 Zone 内的可信管理关系稳定之后，再扩展 Privacy CA 能够管理的 TNs 数量。

图 3.2 显示的是 TCCP 模型的总体架构（Zone 内部的网络拓朴被简化，每一个节点都能通过 CC 与其他节点通信），针对云平台多区域的特性本文采用的 TC 架构采用了层次性设计，上层为 TC 管理者（TC Manager， TCM），它的功能有 1）认证 TN 中 TPM 的有效性和平台完整性；2）管理 TCs，将 TC 与 Zone 一一对应；3）利用自己产生的私钥对 TC 发布的公钥进行认证以防止 Rudolph 攻击。3）负责 ETE 与云平台 CLC 联络。下层的每个 TC 负责一个 Zone 内的 DAA 证书发布以及向 CSP 汇报无赖 TPM 的状况，图 3.2 中 TC1 对应 Zone1，TC2 对应 Zone2，在云平台 Zone 中，无可信芯片的普通服务器以白色标识，普通 TN 以浅灰色标识，具有 Privacy CA 角色的 TN 以中灰色标识，而 ITC 以深灰色表示，虚线框内为 Privacy CA 管理的区域。从图中可知，ITC 也在某一 Privacy CA 管理的区域内，这是因为 TN 仅仅承担 ITC 与 Privacy CA 的职责太浪费资源，在本文中被选中成为 ITC 与 Privacy CA 角色的 TN 依然需要承担普通 TN 的工作量，负责为用户提供黑盒可信环境以运行高安全需求的 VM。出于安全与隐私保护方面考虑 TN 只能承担 ITC 角色与 Privacy 角色中的一种。

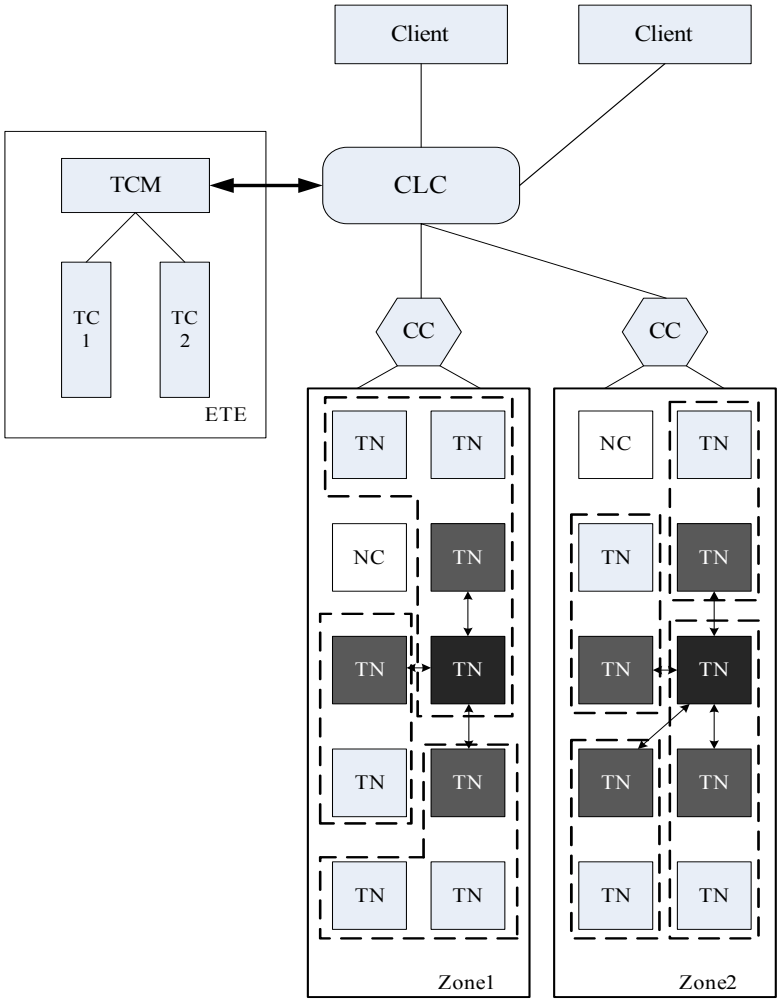


图 3.2　TCCP 模型总体架构

3.2　虚拟可信平台架构设计

原始 TCCP 模型直接引用虚拟可信平台的研究成果，虚拟可信平台需要具有类似 Terra 的功能，即能够在可信服务器中提供黑盒环境给高安全需求的 VM 使用。本文基于 vTPM 架构并根据云计算平台的特征提出了自己的虚拟可信平台架构设计。

图 3.3 显示的是本文基于 vTPM 的虚拟可信平台架构，和 Terra 平台类似，它除了提供白盒不可信环境（open-box untrusted environment，即运行标准的 VM），黑盒可信环境（closed-box trusted environment，由虚拟 TPM 支持运行高安全要求的 VM，其内存活动无法被 CSP 内部员工或黑客窥探）之外，还能提供白盒可信环境（open-box trusted environment，由虚拟 TPM 支持运行标准的 VM）。添加白盒可信环境的理由是部分用户对于 VM 的数据保密性要求不是很

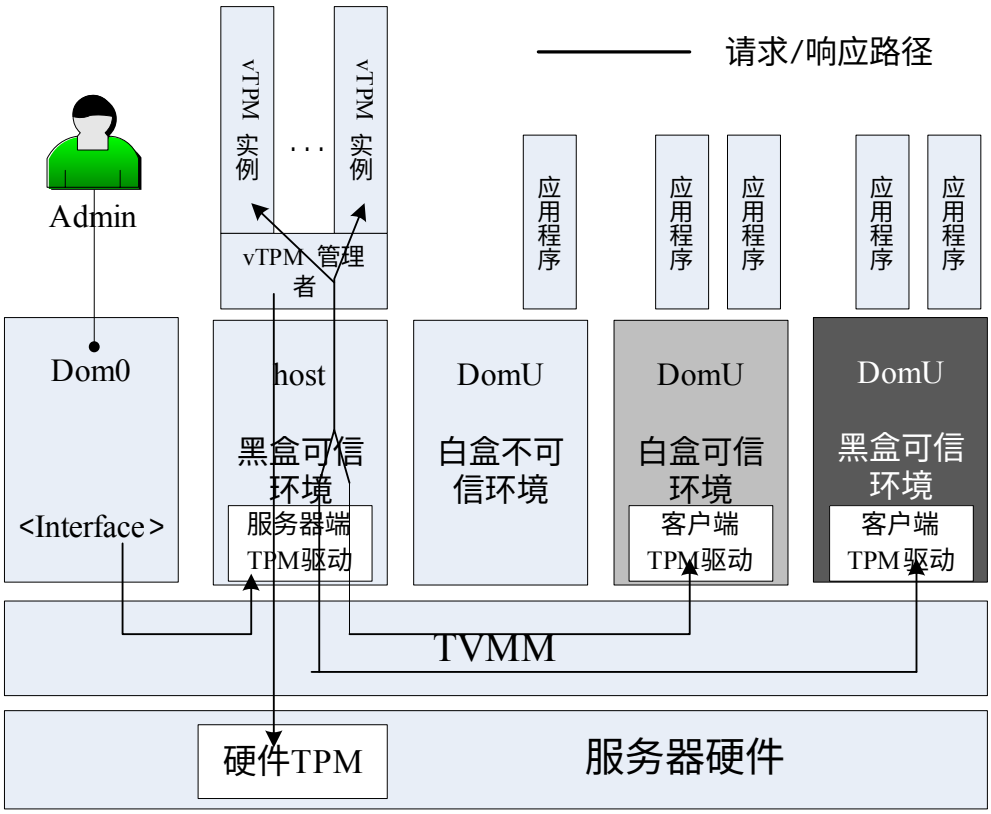


图 3.3 虚拟可信平台架构

高，但需要 VM 能够提供远程证明的功能确保其处在可信完整性状态下（更简单地，用户需要 VM 具有可信计算能力但不要求 VM 的安全性很高）。这时白盒可信环境能够很好满足用户要求，而且更节省资源。

管理 VM（即 Domain0）由管理员控制，在 2.1.1 节针对 Xen 架构的攻击模型提到过的 Xenaccess 只需在 Domain0 内运行用户级进程就能够窥探到用户 VM

的内存活动。然而 vTPM 架构设计将 vTPM manager 和 vTPM 实例都设置在 Domain0 内，并且管理权也交给管理员，这显然没有考虑管理员为假象敌的情况。若管理员是恶意的，vTPM manager 和 vTPM 实例很容易被破坏。此外采用 DAA 策略后，设计平台的协议需要 host 和 TPM 共同完成。而 host 总能揭露出 TPM 的身份标识，因此 DAA 策略需要 host 总是诚实的。在可信虚拟平台中若 host 的工作由 Domain0 完成，而 Domain0 由恶意管理员控制，TPM 的身份也会暴露。因此本文设立了一个专门的 THVM（Trusted Host VM，可信 Host 虚拟机），THVM 在平台初始化后最先被创建（优先于 Domain0），由 TVMM 支持在黑盒可信环境中运行专门的可信 OS。

除了之前在 Domain0 内管理 vTPM 的功能移到了 THVM 内，THVM 还负责 DAA 策略里面 host 的计算。这样的设计不仅避免 vTPM 的管理工作被干扰，也保证了 host 的诚实性（远程验证者可以通过完整性度量来判断）。此外 Privacy CA 和 ITC 角色的工作也在 THVM 中执行。本文在 Domain0 里面为管理员提供了一个接口，管理员可以通过接口收集必要信息或协调 VM 的运行与调度，但不能进行任何越权操作。

在 2.1.2 节提到 vTPM 架构设计还提出了一种一直保持运行的特权 vTPM 根实例，它能够提供诸如对称密钥生成，加解密，生成子 vTPM 实例以及处理 vTPM 非对称密钥迁移的功能，目的是为了迁移 vTPM 实例时能够加密 vTPM 的状态。本文在处理 VM 迁移时也需要 vTPM 实例跟着迁移，因此 vTPM 根实例的角色也被引入。在本文中 THVM 可以很好地充当这个角色，和 vTPM 根实例一样，它在平台初始化后就被创建并一直运行直到平台重启或关闭。THVM 在 TPM 驱动的支持下能够完成一系列的密码学操作，再加上 vTPM 实例都运行在 THVM 中使得 THVM 成为一个天然的根特权 vTPM 实例。

对于远程证明来说，其需要的信任链是从 CRTM 一直到 THVM。由于 THVM 是平台中最先启动的 VM（优先于 Domain0），只要 THVM 运行正常，TN 就能为 VM 提供可信计算能力与黑盒可信环境，即使管理员在 Domain0 运行恶意程序也并不会影响 THVM 提供正常的服务，因此只要 CRTM 到 THVM 的信任链对应的平台完整性度量结果正常，验证者就能确信 TN 平台的完整可信性。在这里本文并没有采用基于属性证明的方式，由于 THVM 上运行的是专门的可信 OS，本文假定完整性度量结果 hash 值能够让验证者较容易判断平台的完整可信性。

3.3　可信管理关系的建立过程

本节根据 Zone 内部可信管理关系（Trusted Management Relationship,TMR）

的建立过程详细描述 TCCP 内部机制的设计细节。

TMR 表示的是 Privacy CA 与其管理区域内部 TNs 的管理关系以及 ITC 与它负责联络的 Privacy CAs 之间的管理关系的总和。Zone 内 TMR 的建立过程从 Privacy CA 建立自己的管理区域开始到 ITC 被 Privacy CAs 联合选举出来之后结束。ITC 选举的结束标志着稳定的 TMR 已经建立起来。

TMR 建立的前提条件是 TN 能够证明自己平台的完整可信性，并且 Privacy CA 要能向将被其管理的 TN 证明自己的角色是由 TC 指定的。本文的思路是针对普通 TN 与承担 Privacy CA 角色的 TN 提供两种不同的 DAA 证书，TNs 可以通过不同的 DAA 证书证明自己的平台完整可信性与角色身份。为此本文采用并修改了原始 DAA 策略[15]与结合 Privacy CA 的 DAA 策略[22]。

3.3.1　DAA 证书发布过程

为了获得 DAA 证书，TN 上线后首先必须联络 TCM，这里假定 TNs 都以一个秘密渠道知道 TCM 认证密钥的公钥（比如以可信服务器出厂时保存在 TPM 内部的形式），本文使用 PK_{TCM} 表示（对应的私钥使用 pK_{TCM} 表示）。

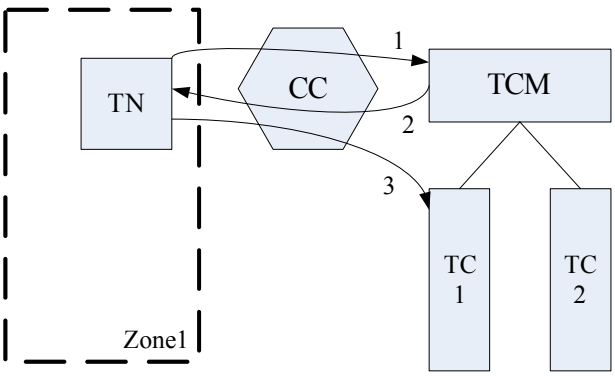


图 3.4　TCM 检验 TN 平台完整性与 TPM 有效性

图 3.4 显示了 Zone1 内 TN 联络 TCM 的 2 步过程：

1. TN 向 TCM 发送 $\{\{n_N, ML_{TN}\}_{pK_{EK}}, H(PK_{EK})\}_{PK_{TCM}}$，这里 $n_N \in \{0,1\}^{\varnothing}$ 为 TN 生成的 nonce，pK_{EK} 为 TN 中 TPM 的私钥，ML_{TN} 为 TN 的平台完整性度量结果 hash 值，$H(PK_{EK})$ 为 EK 公钥的 hash 值。$\{\{n_N, ML_{TN}\}_{pK_{EK}}\}_{PK_{TCM}}$ 表示 $\{n_N, ML_{TN}\}$ 由 EK 私钥加密后再由 TCM 的公钥加密；

2. TCM 收到后使用 pK_{TCM} 解密 $\{\{n_N, ML_{TN}\}_{pK_{EK}}, H(PK_{EK})\}_{PK_{TCM}}$ 得到 $\{n_N, ML_{TN}\}_{pK_{EK}}$ 和 $H(PK_{EK})$，然后对有效 EK 的公钥列表中的公钥进行 hash 运行比对 $H(PK_{EK})$，使用合适的 PK_{EK} 对消息进行解密得到 n_N 和 ML_{TN}，TCM 根据 ML_{TN} 判断 TN 平台的完整性。平台完整性验证通过后,TCM 根据 TN 所在的 Zone 选择 TC1 作为 TN

的 DAA 发证者并获得 TC1 的完整性度量结果 hash 值 ML_{TC1}，TC1 会根据 TN 角色分配算法指定目前注册的 TN 的角色，本文设定 $M \in \{0,1\}$ 负责角色的指派，即 $M=0$ 意味着该 TN 将被指派承担 Privacy CA 角色的职责，$M=1$ 意味着该 TN 将被指派承担普通 TN 角色的职责，M 的不同也意味着分发的 PK_{TC1} 的不同（分给普通 TN 的 $PK_{TC1} := (n, g', g, h, S, Z, R_0, R_1, R_2)$，分给 Privacy CA 的 $PK_{TC1} := (n, g', g, h, S, Z, R_0, R_1)$，$PK_{TC1}$ 的生成详见文献[15][22]）。TCM 根据 TC1 反馈的 M 值选择对应的 PK_{TC1}，并使用自己的 pK_{TCM} 对其进行认证以防止 Rudolph 攻击，发送给 TN 的消息为 $\{\{ML_{TC1}, PK_{TC1}, M, bsn_{TC1}, n_{TC1}\}_{PK_{EK}}, n_N\}_{pK_{TCM}}$，这里 $n_{TC1} \in \{0,1\}^{l_\varnothing}$ 由 TC1 随机选择的 nonce 用来防止重放攻击，bsn_{TC1} 为 TCM 保存的 TC1 长期基名。

在发表的论文中步骤 1）TN 向 TCM 发送的消息为 $\{n_N, PK_{EK}, ML_{TN}\}_{PK_{TCM}}$，这主要是出于 TCM 解密的效率考虑，直接给出 PK_{EK} 的话 TCM 只需直接查看有效 TPM 的 PK_{EK} 列表，不过假如 TCM 的 PK_{TCM} 与 pK_{TCM} 都被窃取，敌方就有可能获得 PK_{EK} 与 ML_{TN}。出于安全方面考虑还是使用 pK_{EK} 对 ML_{TN} 进行加密，本文为了提高效率 TN 同时向 TCM 发送 $H(PK_{EK})$，这使得 TCM 不需要逐一试用有效 TPM 列表中的 PK_{EK} 对消息进行试解密，只需要在之前对列表中 PK_{EK} 进行 hash 运算比对 TN 的 $H(PK_{EK})$，找到合适的 PK_{EK} 再试解密 $\{n_N, ML_{TN}\}_{pK_{EK}}$。

接下来就开始了 TN 与 TC1 的联络（主要为 DAA 证书的颁布过程），这里我们假设 TN 与 TC1 能够以保密的方式进行交流：

3. TN 收到消息后使用 PK_{TCM} 解密得到 $\{ML_{TC1}, PK_{TC1}, M, n_{TC1}\}_{PK_{EK}}$ 和 n_N，TN 比对 n_N 确定返回的消息来自正确的 TCM，然后使用 pK_{EK} 解密得到 ML_{TC1}，PK_{TC1}，M，bsn_{TC1} 和 n_{TC1}，根据 ML_{TC1} 判断 TC1 的平台完整性，然后由 M 的值获知自己的角色，计算 U 和 N_{TC1} 并发送给 TC1；

1) TN 内 THVM 计算 $\zeta_{TC1} := (H_\Gamma(1 \| bsn_{TC1}))^{(\Gamma-1)/\rho} (\mathrm{mod}\,\Gamma)$，并选择 cnt（即平台运行 Join 协议的次数，不过 cnt 可以保持不变）发送给 TPM；

2) TPM 检查 $\zeta_{TC1}^{\rho} \equiv 1 (\mathrm{mod}\,\Gamma)$，并计算

$f : H(H(DAAseed \| H(PK_{TCM})) \| cnt \| 0) \| H(DAAseed \| H(PK_{TCM})) \| cnt \| 1)$，

并计算 $f_0 := LSB_{\lambda_f}(f)$，$f_1 := CAR_{\lambda_f}(f)$，盲因子 $v' \in_R \{0,1\}^{l_n + l_\varnothing}$，

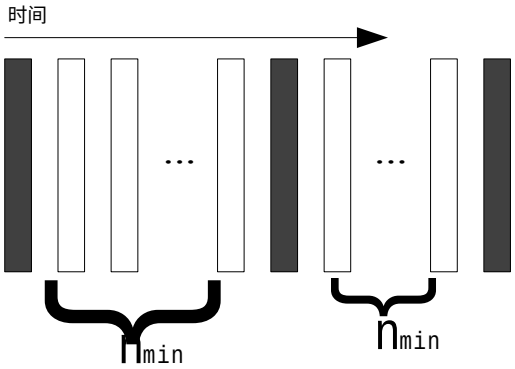
$U := R_0^{f_0} R_1^{f_1} S^{v'} (\mathrm{mod}\, n)$，$N_{TC1} := \zeta_{TC1}^{f_0 + f_1 2^{\lambda_f}} (\mathrm{mod}\,\Gamma)$，将 U 和 N_{TC1} 发送给 TC1。

4. TC1 将无赖 TPM 列表中的 (f_0, f_1) 作为输入检查 $N_{TC1} \stackrel{?}{\neq} (\zeta_{TC1}^{f_0 + f_1 2^{\lambda_f}})$，若 TC1 发现 N_{TC1} 为无赖 TPM 生成，立即终止协议。

5. TPM 向 DAA 发证者证明 U 和 N_I 的正确性以及秘密信息 f_i 在合适的区间内（详细过程见 2.2.1.2 节 Join 协议涉及的零知识证明）：

$$SPK\{(f_0, f_1, v') : U \equiv \pm R_0^{f_0} R_1^{f_1} S^{v'} (\mathrm{mod}\, n) \wedge$$

$$N_I \equiv \zeta_I^{f_0 + f_1 2^{l_f}} (\mathrm{mod}\, n) \wedge f_0, f_1 \in \{0,1\}^{l_f + l_\varnothing + l_H + 2} \wedge v' \in \{0,1\}^{l_n + l_\varnothing + l_H + 2}\}(n_t \| n_i)$$



。

图 3.5　TMR 建立过程中 TN 角色的指派流程

表 3.1　Privacy CA 与普通 TN 获取 DAA 证书的后续步骤

	不同步骤	
相同步骤	Privacy CA　（m = 0）	普通 TN（m = 1）
6.　TC 选择 $\hat{v} \in_R \{0,1\}^{\lambda_v - 1}$，一个素数 $e \in_R [2^{\lambda_e - 1}, 2^{\lambda_e - 1} + 2^{\lambda'_e - 1}]$，并计算 $v'' := \hat{v} + 2^{\lambda_v - 1}$ ；	$A := (\frac{Z}{US^{v''}})^{1/e} (\mathrm{mod}\, n)$	TC1 选择一个唯一的 k_t，并计算 $A := (\frac{Z}{US^{v''} R_2^{k_t}})^{1/e} (\mathrm{mod}\, n)$
7.	$SPK\{(d) : A \equiv (\frac{Z}{US^{v''}})^d (\mathrm{mod}\, n)\}(n_h)$，在证明过程中，Privacy CA 生成非对称密钥对 (PK_{PN}, pK_{PN})，并生成 Privacy CA 的公钥 $PK_P := (n, g', g, h, S, Z, R_0, R_1, R_2, R_3)$，将 PK_P，PK_{PN} 和 bsn_P 发送给 TC1 用于与普通 TNs 的联络。TC1 生成一个唯一 k_s，并将用于以后 Privacy CAs 之间联络的会话密钥 K_{PCA} 一起发送给 THVM；	$SPK\{(d) : A \equiv (\frac{Z}{US^{v''} R_2^{k_t}})^d (\mathrm{mod} n)\}(n_h)$，TC1 发送给 THVM 管理该 TN 的 Privacy CA 的 PK_P，bsn_P，PK_{PN} 和 k_s ；
8.　THVM 检查 e 时都为素数，且在 $[2^{\lambda_e - 1}, 2^{\lambda_e - 1} + 2^{\lambda'_e - 1}]$ 内，然后将 v'' 发送给 TPM，TPM 保存 (f_0, f_1, v)。	THVM 保存 $(A, e, PK_{PN}, pK_{PN}, PK_P, K_{PCA}, k_t, k_s)$。	THVM 保存 $(A, e, PK_{PN}, PK_P, bsn_p, k_s)$。

给 Privacy CA 与普通 TN 角色颁布 DAA 证书的协议在以上前 5 步都是一致的，后面步骤的相同与不同点见表 3.1。

在描述表 3.1 之前，TC1 在 TMR 建立时如何进行 TN 的角色指派需要进一步说明。本文采用的指派方案按 TN 注册的时间先后安排，假定 TN a 为 Zone 内首个向 TC1 申请 DAA 证书的 TN，a 将会被指派为 Privacy CA，其后的 TN 都被指派为普通 TN 且归 a 管理直到普通 TNs 的注册数量等于 $n_{\min}$，TC1 才会指派下一个注册的 TN 为 Privacy CA（在 TMR 稳定的时候 TC1 是根据 ITC 反馈 Zone 内的 $n_{available}$ 情况指派 TN 的角色），图 3.5 反映了 TMR 建立过程中的角色指派流程。

因此当 Zone 内首个 TN a 向 TC1 申请 DAA 证书时获得的是 Privacy CA 的 DAA 证书，如表 3.1 所示，在步骤 7 中 a 确信 A 被正确计算后，它会向 TC 发送 PK_{PN} 用于与它管理的 TNs 的保密联络，bsn_P 用于在以后的协议中 ζ_P 计算和 PK_P 用于发布 Privacy CA 匿名证书。TC1 会为 a 生成一个唯一的 k_s 用于确认 Privacy CA 与其下属的 TNs 的管理关系以及用于以后 ITC 选举结果的确认，并将用于 Privacy CAs 之间联络的会话密钥 K_{PCA} 一起发送给 a。

若同一个 Zone 内第二个向 TC1 注册的 TN b 申请 DAA 证书时获得的是普通 TN 的 DAA 证书，在步骤 7 中 b 确信 A 被正确计算且得知区域内 Privacy CA 为 a 后，TC1 会向 b 发送用于联络 a 的 PK_{PN}，Privacy CA 用于发布证书的 PK_P，Privacy CA 的长期基名 bsn_P，用于确认管理关系的 k_s。

3.3.2　Privacy CA 匿名证书发布过程

然后 b 为了成为 a 管理区域内的成员，必须向 a 证明拥有有效的 DAA 证书，本文采用并修改了文献[22]的一次性证书（在本文不再是一次性的证书，我们称为 Privacy CA 匿名证书）颁布协议用于不同的用途，协议完成后 b 能够证明自己拥有有效的 DAA 证书，并且获得与之绑定的 Privacy CA 匿名证书，其用途是若 a 定期检查管理区域 TN 的状态，只需验证 TN 对 Privacy CA 匿名证书的签名即可，因此可以看出 a 同时拥有证书颁布者与验证者两种角色。

Privacy CA 匿名证书颁布协议如下（以 TN b 向其上级 Privacy CA a 注册为例）：

1. b 向 a 发送 $\{k_s, n_b\}_{PK_{PN}}$，这里 n_b 为一个 nonce；
2. a 收到后使用 pK_{PN} 解密，查看 k_s 是否与自己的一致，若一致则证明 b 确实被 TC1 指定加入 a 的管理区域，此时运行原始 DAA 协议的 Sign 协议与 Verify 算法，a 向 b 证明自己是可信的 Privacy CA，然后 a 需要检查 b 的平台完整性以及 b 是否具有有效的 DAA 证书，若 b 通过检查 a 将向 b 颁布 Privacy CA 证书，因此 a

发送 $\{n_b, start, n_a\}_{pK_{PN}}$，这里 n_a 为一个 nonce；

3.

1) b 的 THVM 选择随机整数 $w, r \in [1, \lfloor n/4 \rfloor]$，计算 $\zeta_P := (H_\Gamma(1 \| bsn_P))^{(\Gamma\text{-}1)/\rho} (\mathrm{mod}\,\Gamma)$，$T_1 := Ah^w (\mathrm{mod}\, n)$，$T_2 := g^w h^e (g')^r (\mathrm{mod}\, n)$,并将 ζ_P 发送给 TPM；
2) TPM 计算 $N_P := \zeta_P^{f_0 + f_1 2^{\lambda_f}} (\mathrm{mod}\,\Gamma)$，发送 N_P 给 THVM；
3) b 的 THVM 选择足够大的随机串 $ran \in_R \{0,1\}^{\lambda_H}$，计算 $k_0 := LSB_{\lambda_f}(H(bsn_P \| ran))$, $k_1 := CAR_{\lambda_f}(H(bsn_P \| ran))$，并选择随机整数 $v' \in_R [0, \lfloor n/4 \rfloor]$，计算 $U := R_0^{k_0} R_1^{k1} R_2^{k_t} S^{v'} (\mathrm{mod}\, n)$。
4) THVM 将 U 与证实 U 计算正确的零知识证明签名

$$SPK\{(f_0, f_1, v, e, w, r, k_t, k_0, k_1, v') : U \equiv \pm R_0^{f_0} R_1^{f_1} R_2^{k_t} S^{v'} (\mathrm{mod}\, n) \wedge$$
$$Z \equiv T_1^e R_0^{f_0} R_1^{f_1} R_2^{k_t} S^v h^{-ew} (\mathrm{mod}\, n) \wedge 1 \equiv T_2^{-e} g^{ew} h^{ee} g'^{er} (\mathrm{mod}\, n) \wedge$$
$$N_P \equiv \zeta_P^{f_0 + f_1 2^{l_f}} (\mathrm{mod}\,\Gamma) \wedge T_2 \equiv g^w h^e g'^r (\mathrm{mod}\, n) \wedge$$
$$f_0, f_1, k_t, k_0, k_1 \in \{0,1\}^{l_f + l_\varnothing + l_H + 2} \wedge (e - 2^{l_e}) \in \{0,1\}^{l'_e + l_\varnothing + l_H + 2}\}$$

发给 a，该签名除了能够证明 U 计算正确，还能够证明 b 获得了 DAA 证书且 DAA 证书与 U 包含了同一个值 k_t。

5) a 检查 $\zeta_P \stackrel{?}{\equiv} (H_\Gamma(1 \| bsn_P))^{(\Gamma\text{-}1)/\rho} (\mathrm{mod}\,\Gamma)$，并将无赖 TPM 列表 (f_0, f_1) 作为输入，检查 $N_P \stackrel{?}{\neq} (\zeta_P^{f_0 + f_1 2^{\lambda_f}}) (\mathrm{mod}\,\Gamma)$ 以及 N_P 是否出现得太频繁；
6) a 确定 $\exp\text{-}date \in \{0,1\}^{\lambda_f}$，这里 $\exp\text{-}date$ 表示 Privacy CA 证书的到期期限，由于 Privacy CA 证书不再是一次性证书，因此证书的寿命长于原协议中的一次性证书寿命。a 选择 $\hat{v} \in_R \{0,1\}^{\lambda_v - 1}$，一个素数 $e \in_R [2^{\lambda_e - 1}, 2^{\lambda_e - 1} + 2^{\lambda'_e - 1}]$，计算 $v'' := \hat{v} + 2^{\lambda_v - 1}$ 以及 $A := (\frac{Z}{UR_3^{\exp\text{-}date} S^{v''}})^{1/e} (\mathrm{mod}\, n)$，将 (A, e, v'') 与 $\exp\text{-}date$ 发送给 b 的 THVM。
7) 为了让 b 确信 A 计算正确，a 运行零知识证明签名 $SPK\{(d) : A \equiv \pm (\frac{Z}{UR_3^{\exp\text{-}date} S^{v''}})^d (\mathrm{mod}\, n)\}$。
8) b 的 THVM 检查 e 是否为素数且处在 $[2^{\lambda_e - 1}, 2^{\lambda_e - 1} + 2^{\lambda'_e - 1}]$ 内，$A^e R_0^{k_0} R_1^{k_1} R_2^{k_t} R_3^{\exp\text{-}date} S^v \equiv Z (\mathrm{mod}\, n)$，并计算 $v := v'' + v'$ 并存储 (A, e, v)，ran 与 $\exp\text{-}date$。

b 向 a 注册过程到此结束，b 现在拥有了通过 k_t 与 DAA 证书绑定的 Privacy CA 证书，在 a 的管理过程中若想要周期性地查看 b 的可信完整性状态，他们只需要运行文献[22]的 Sign 协议（需将原协议中 $verifier - name$ 改为 bsn_P）。值得注意的是出于安全考虑 Privacy CA 证书有使用期限，若 b 的 Privacy CA 证书到期，b 会运行上述协议重新获得 Privacy CA 证书。

3.3.3　ITC 选举过程

Zone 内 Privacy CA 管理区域都已经建立起来后就需要 ITC 将不同的管理区域联系起来。在 3.4.1 节中 TC 为 Privacy CA 颁布 DAA 证书时同时为同一个 Zone 内 Privacy CAs 生成了一个会话密钥 K_{PCA} 以便联络。在发表的论文中 Privacy CAs 在进行选举之前会互相验证自己的 DAA 证书，这需要运行 $(N-1)^N$ 次原始 DAA 策略的 Sign 协议与 Verify 算法，这里 N 表示 Privacy CAs 的数量。显然这是一个低效率的做法，既然 Privacy CAs 在申请证书时已经证明自己平台的可信完整性了，在获得 DAA 证书到选举 ITC 这段时间 Privacy CAs 就被攻破的可能性很小，而且采用 SMC 方案能够在一定程度上避免恶意 Privacy CAs 操纵选举结果（在无条件安全模型下无赖 Privacy CAs 数目小于总数 1/3，条件安全模型下无赖 Privacy CAs 数目小于总数 1/5 的情况下）。因此本文不进行平台可信完整性的验证直接运行 SMC 选举算法（本文采用了最简单的 SMC 算法，更优算法不在本文讨论范围以内）：

1）每一个 Privacy CA i 随机选取一个随机正整数 $n_i \in Z+$，并发给其他 Privacy CAs（举例假定 Privacy CA a: 4; t: 2; k: 7; g: 3; p: 16）；

2）每一个 Privacy CA 计算 $h := \sum_{i=1}^{N} n_i \pmod N$（此例 $(4+2+7+3+16) \bmod 5 = 2$），h 对应给出第 $(h+1)$ 小 n_i 的 Privacy CA（此例 a 给出了第 3 小的值）；

3）Privacy CAs 比对计算结果，若都确认了 h 对应唯一一个 Privacy CA，由 Privacy CA 基于管理区域内 TNs 的可用资源使用一种随机算法（随机算法的目的是增加敌手找到 ITC 的难度，具体算法不在本文给出）选择一个普通 TN 作为 ITC（假设 Privacy CA a 选择 b 作为 ITC）；

4）b 首先使用 Privacy CA 匿名证书的 Sign 协议向 a 证明自己的平台可信性，然后以其他 Privacy CAs 的 bsn_P 作为 $verifier-name$ 向 a 申请一次性证书（此处使用文献[22]的原始协议），并通过一次性证书的 Sign 协议分别向其他 Privacy CAs 证明自己的可信性；

5）为了使 TC 确信 ITC 选举结果，Privacy CAs 使用随机算法将自己唯一的分成 k_{s0} 与 k_{s1}（这里 $k_s = k_{s0} + k_{s1}$），这里假设第 0 小的数表示的是步骤 1）中给出的最大的数（此例中第 0 小的数为 16），Privacy CA 在步骤 1）中给出了第 i 小的数，它将分成的 k_{s0} 发送给步骤 1）中给出第 $(i-1) \bmod n$ 小的数的 Privacy CA，k_{s1} 发送给步骤 1）中给出第 $(i+1) \bmod n$ 小的数的 Privacy CA；

6）若 Privacy CA 在步骤 1）中给出了第 i 小的数，那么它收到的一定是步骤 1）中给出第 $(i+1) \bmod n$ 小的数的 Privacy CA 发送的 k_{s0} 以及步骤 1）中给出第 $(i-1) \bmod n$ 小的数的 Privacy CA 发送的 k_{s1}，Privacy CA 将这两个数相加

发送给 ITC；

7）ITC 将每个 Privacy CA 发送的值相加得到 SUM_{k_s}，并以 bsn_{TC} 作为 $verifier-name$ 向所属的管理区域的 Privacy CA 申请一次性证书，使用该一次性证书向 TC 证明自己的可信性，并将 SUM_{K_s} 发送给 TC；

8）TC 若确信 ITC 拥有一次性证书以及与其绑定的 DAA 证书后，将之前为 Privacy CAs 计算的 k_s 相加，检查 $SUM_{k_s} \stackrel{?}{=} \sum k_t$，若检查通过，TC 确认 ITC 选举结果。

到此稳定的 TMR 已经建立起来。

3.4　可信管理关系稳定之后新的可信节点注册过程

为了 TMR 建立的方便，之前每个 Privacy CA 管理的 TNs 数量都是 $n_{\min}$（除了最后一个被指派的 Privacy CA 管理的 TNs 数量可能少于 $n_{\min}$），ITC 被选举出来之后就要开始履行自己的职责，它负责收集 Privacy CAs 管理区域的 $n_{available}$（不同 Privacy CA 性能的不一致会导致 $n_{available}$ 的不同）以及管理区域内能够分配的资源概况。

当新的 TN 上线后向 TC 注册时，TC 首先检查 Zone 内的各个区域的 $n_{available}$，若存在管理区域还存在容纳能力，TC 会给 TN 指派普通 TN 角色。其目的管理区域的选择按照一种基于管理区域容纳能力的随机选择算法确定：

1) TC 以某种方式将管理区域映射到在区间 $[1, \sum n_{available}]$ 中（举例假定 Privacy CA a: $n_{available} = 4$; k: $n_{available} = 7$; p: $n_{available} = 16$，即 Zone 内 Privacy CA 管理区域能够容纳的总量为 $\sum n_{available} = 27$，让 [1,4] 对应 a 的管理区域，[5,11] 表示 k 的管理区域，[12,27] 表示 p 的管理区域）；

2) TC 选择在 $[1, \sum n_{available}]$ 随机选择一个整数 c，根据 c 确定 TN 的管理区域（假设随机整数 c 为 8，落在 [5,11]，表明新 TN 将加入 k 的管理区域），同时会给 TN 该管理区域对应的 k_s 用来申请 Privacy CA 匿名证书。

该算法的目的是通过增加 TMR 的随机性以提高入侵的难度（至少能提高信息收集的难度）。

若 $\sum n_{available} = 0$，新 TN 会被指派为 Privacy CA，然后通过原始 DAA 策略的 Sign 与 Verify 算法向 ITC 证明自己的角色身份（只有 Privacy CA 能够获得原始 DAA 证书）。

3.5　虚拟机管理

TCCP 模型提出的初衷是通过结合可信计算技术保护用户 VM 内存不被窥探或篡改以解决用户与 CSPs 之间的信任问题。假设 TCCP 模型能够有效保护用户 VM，现在最大的问题是用户在根本不知道云端内部信息的情况下怎么确认自己的 VM 运行在黑盒可信环境中。在这里本文引入 iTurtle[30]的概念，本文的 iTurtle 为 USB 设备插在用户电脑上使用，它可以看作 TN 的缩小简化版，具有 TPM，运行在 iTurtle 上的软件足够简单但是能够检验 TN 的完整性度量结果判断平台的可信完整性并且具有自我检验的能力。若 iTurtle 检测到用户 VM 运行在可信黑盒环境中，USB 会以绿灯提示用户。

3.5.1　高安全需求虚拟机的创建过程

高安全需求 VM 的创建过程（若目的 TN 为普通 TN）如下：

1）若用户需要 VM 运行在黑盒可信环境中，会向云端 CLC 提出申请，若申请通过 CLC 首先确定 VM 创建的目的 Zone 并允许 iTurtle 访问 TCM；

2）这里采用 AIK 认证方案，iTurtle 发送 $\{\{PK_{AIK}, n_i, ML_{iTurtle}, bsn_{iTurtle}\}_{pK_{EK}}, H(PK_{EK})\}_{PK_{TCM}}$，这里 n_i 为一个 nonce，若 iTurtle 内含有效 TPM，TCM 将解密得到 PK_{AIK}，n_i，$bsn_{iTurtle}$ 和 $ML_{iTurtle}$，通过检验 $ML_{iTurtle}$ 能够判断 iTurtle 的平台完整性。检验通过后 TCM 通过查找目的 Zone 对应的 TC 内 ITC 的信息找到 ITC 并发送 ITC 的 PK_{AIK}，n_i 和 $bsn_{iTurtle}$；

3）ITC 通过类似于 3.4 节提到的随机选择算法选择目的 Privacy CA 管理区域，这里选择的依据是 Privacy CA 管理区域可以分配资源的概况，管理区域内能够分配的资源越多，VM 创建在该管理区域内的可能性就越大。ITC 将 iTurtle 的 PK_{AIK}，n_i 和 $bsn_{iTurtle}$ 发送给选中的 Privacy CA；

4）Privacy CA 也通过类似于 3.4 节提到的随机选择算法选择目的 TN，这里选择的依据就是单个 TN 可以分配的资源详情，TN 能够分配的资源原多，VM 创建在该管理区域内的可能性就越大。最终 Privacy CA 将 iTurtle 的 PK_{AIK}，n_i 和 $bsn_{iTurtle}$ 发送给选中的 TN；

5）TN 以 $bsn_{iTurtle}$ 作为 $verifier-name$ 向管理区域内的 Privacy CA 申请一次性证书，使用一次性证书的 Sign 协议以及 ML_{TN} 向 iTurtle 确认自己平台的可信完整性。通过 n_i 让 iTurtle 确认 TN 为选中的目的地 TN；

6）用户使用 $\{\alpha, H(\alpha)\}_{pK_{AIK}}$ 向目的 TN 发送虚拟镜像，这里 α 为用户的虚拟镜像，$H(\alpha)$ 为虚拟镜像的 hash 值；

7）TN 使用 PK_{AIK} 解密得到 α 和 $H(\alpha)$，通过对 α 进行 hash 运算与 $H(\alpha)$ 比对判断 α 的完整性，检验通过后 TN 使用虚拟镜像 α 创建虚拟机。

在上述方案中 iTurtle 的 PK_{AIK}，n_i 和 $bsn_{iTurtle}$ 以链式的形式进行传递，即 TC，

ITC，Privacy CA 都有可能知道 PK_{AIK}，n_i 和 $bsn_{iTurtle}$，这有可能造成安全隐患。不过该算法的目的只是找一个 TN 作为 VM 的创建目的地，若存在 TN 向 iTurtle 反馈，iTurtle 还需检验该 TN 平台的可信完整性状态。而且在 VM 运行的时刻，iTurtle 可以不断确认该 TN 平台的可信完整性状态，以及 VM 的完整性状态。

在创建 VM 时，iTurtle 还是需要联系 TCM，这是因为只有 TCM 能够确认 TPM 的有效性，CSPs 不可能允许含有无赖 TPM 的 iTurtle 验证云端内部的 TN 可信完整性状态。

若目的 TN 被选中为 Privacy CA 时，步骤 5）就是采用原始 DAA 策略里面的 Sign 协议与 Verify 算法证明平台的可信完整状态，不过 iTurtle 也会了解目的 TN 是一个 Privacy CA（因为执行的协议不同），这也带来了一个安全隐患。

3.5.2　高安全需求虚拟机的动态迁移过程

高安全需求 VM 动态迁移过程主要分 Zone 内迁移和跨 Zone 迁移，而 Zone 内迁移又分为 Privacy CA 管理区域内迁移和跨 Privacy CA 管理区域迁移。在这里将分别进行说明。

❖ Privacy CA 管理区域内迁移

若 TN 需要维护或高安全需求 VM 有迁移请求，如果 Privacy CA 管理区域内有足够分配的资源，一般地 VM 就在 Privacy CA 管理区域内迁移。

其迁移过程（若源 TN 与目的 TN 都为普通 TN）如下：

1) 有 VM 需要迁移的 TN 生成一个 nonce n_{sTN}，bsn_{sTN}，ML_{sTN} 发送给 Privacy CA，并运行 Privacy CA 匿名证书的 Sign 协议向 Privacy CA 证明自己平台的可信完整性；

2) Privacy CA 也通过类似于 3.4 节提到的随机选择算法选择目的 TN，将源 TN 的 n_{sTN} 和 bsn_{sTN} 发送给选中的 TN；

3) 目的 TN 以源 TN 的 bsn_{sTN} 作为 $verifier-name$ 向管理区域内的 Privacy CA 申请一次性证书，使用一次性证书的 Sign 协议以及自己的 ML_{dTN} 向源 TN 证明自己平台的可信完整性，通过 n_{sTN} 让源 TN 确认选择结果；

4) 源 TN 与目的 TN 开始动态迁移 VM 以及与之绑定的 vTPM 实例（vTPM 实例迁移的详细过程见 2.1.2 节）。

这里若源 TN 为普通 TN 而目的 TN 为 Privacy CA，处理方式与上一节相同；而若源 TN 为 Privacy CA 目的 TN 为普通 TN 的话，在 Privacy CA 选中目的 TN 后，会先要求目的 TN 运行 Privacy CA 匿名证书里面的 Sign 协议以及发送完整性度量结果证明自己的平台可信性，之后 Privacy CA 运行原始 DAA 策略的 Sign

与 Verify 算法以及发送完整性度量结果证明 Privacy CA 的平台完整可信性。这里与普通迁移过程不同的是目的 TN 证明自己在先，这是因为 Privacy CA 经常性地检查下属 TN 的平台完整性，可以把这一次当作普通检查，若检查不通过也不会过早透露自己的完整性度量结果，有效地保护了 Privacy CA 的隐私。

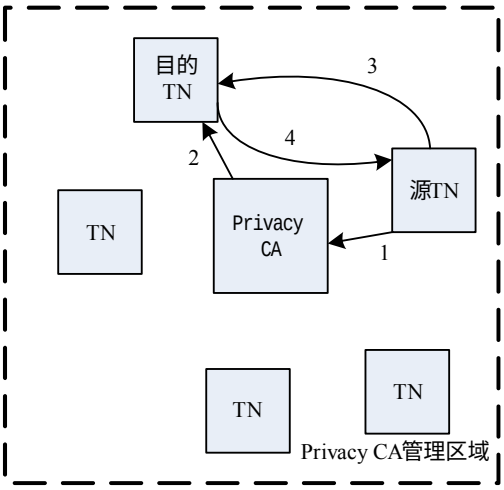


图 3.6　一般情况下 Privacy CA 管理区域内 VM 迁移过程

❖ 跨 Privacy CA 管理区域迁移

若 VM 需迁移时 Privacy CA 发现其管理区域内没有足够可分配的资源，Privacy CA 会联系 ITC，ITC 会根据不同管理区域的可分配资源概况选择 VM 迁往的管理区域，再由选中的管理区域的负责者 Privacy CA 选择目的地 TN。

其迁移过程如下（若源 TN 与目的 TN 都为普通 TN）：

1) 有 VM 需要迁移的 TN 生成一个 nonce n_{sTN}，bsn_{sTN}，ML_{sTN} 发送给 Privacy CA，并运行 Privacy CA 匿名证书的 Sign 协议向 Privacy CA 证明自己平台的可信完整性；

2) Privacy CA 向上级 ITC 反映，并发送 n_{sTN} 和 bsn_{sTN} ；

3）ITC 通过类似于 3.4 节提到的随机选择算法选择目的 Privacy CA 管理区域，将源 TN 的 n_{sTN} 和 bsn_{sTN} 发送给选中的 Privacy CA；

4）Privacy CA 也通过类似于 3.4 节提到的随机选择算法选择目的 TN，将源 TN 的 n_{sTN} 和 bsn_{sTN} 发送给给选中的 TN；

以后的过程与 Privacy CA 管理区域内迁移过程的第 2)步之后相同。

这里若源 TN 与目的 TN 为 Privacy CA，验证平台可信完整性都采用原始 DAA 策略的 Sign 协议与 Verify 算法。

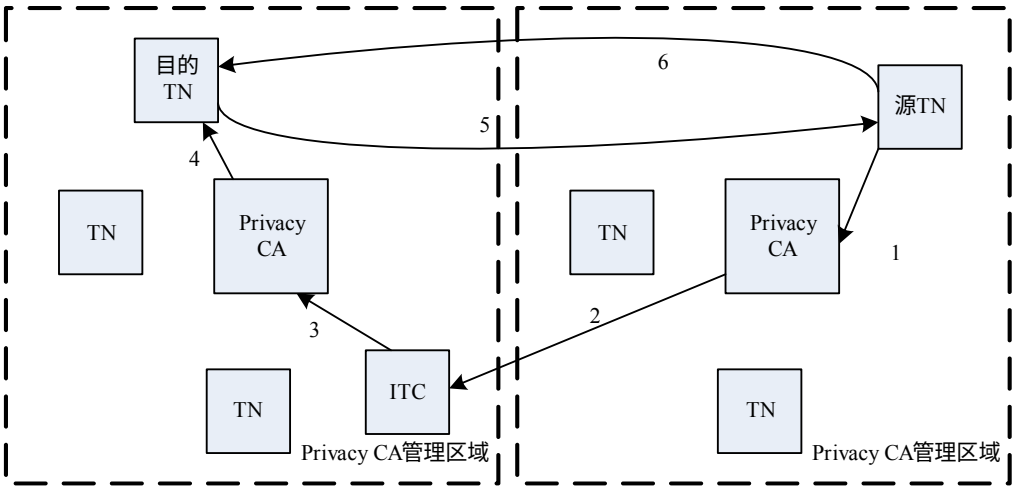


图 3.7　一般情况下跨 Privacy CA 管理区域 VM 迁移过程

❖ 跨 Zone 迁移

除非用户有特殊的要求（比如保存在此 Zone 的数据违反当地法律），一般 VM 不会跨 Zone 迁移。如果用户有这方面的需求，一般会指定迁徙的目的 Zone。跨 Zone 迁移与 Zone 内迁移的区别是两个 Zone 对应的 TC 会参与其中，因为目的 Zone 的 ITC 只有对应的 TC 知道，通过 TCM 沟通两个 TC，使得 Zone 内两个 ITC 取得联系，除此之外 VM 迁移的过程与 Zone 内迁移过程相同。

3.6　TCCP 改进模型的性能与安全分析

本文提出的 TCCP 改进模型为理论模型，将其变成实际运作的平台需要大量的资源与工作。首先由于 VMM 必须运行在裸机上，通过虚拟化环境进行 TCCP 模型试验是不可行的，即使拥有足够的服务器运行 VMM，vTPM 与 Eucalyptus 内部代码还必须经过修改才能符合 TCCP 的要求。

因此本文仅对 TCCP 模型的安全性与性能进行定性分析。

3.6.1　TCCP 改进模型的性能分析

原始 TCCP 模型的性能瓶颈在于其过度依赖 TC，TC 不得不在 VM 创建与迁移过程中扮演主要角色，如果 TC 瘫痪整个模型也就跟着瘫痪。而本文提出的 TCCP 模型使用 DAA 与 Privacy CA 策略将 TC 管理 TNs 的功能移到了云端内部，这时由 Privacy CA 与 ITC 协同管理 TNs。在 TMR 稳定之后，可以看出 VM 在 Zone 内迁移的过程中已经不需要 TC 的参与了。而且本文采用了多个并行的

Privacy CA 管理区域的设置方案且 VM 迁移时优先采用同一个 Privacy CA 管理区域内的 TN 为目的地，这样能够很大程度上提高 TNs 管理与 VM 迁移的效率。然而该 TCCP 模型不是整个 Zone 上线就能运行的，它必须花费一定的时间建立稳定的 TMR，建立 TMR 时又需要采用 DAA 与 Privacy CA 策略，由文献[15]对原始 DAA 策略的性能分析可以得知 DAA 协议的总体时间开销很大，因此稳定的 TMR 建立起来的总体时间开销也会很大。不过由于 Zone 内 TNs 不会一次性全部关闭，因此只要 Zone 内有活跃的 TNs 就存在稳定的 TMR，TMR 的建立过程应该只会在新的 Zone 上线时发生。

TCCP 改进模型在性能上的提升主要是在可靠性上，若 TC 瘫痪并不会影响 VM 的迁移（其实也可以不影响 VM 的创建，不过从安全角度考虑 TCM 必须首先确认用户的 iTurtle 含有有效的 TPM）;若 ITC 瘫痪或被攻破丧失了平台完整可靠性，Privacy CAs 管理区域内的 VM 迁移并不会受到影响；若 Privacy CA 瘫痪或被攻破丧失了平台完整可靠性，其管理区域下的 TNs 会重新向 TC 注册申请新的角色，即使 Privacy CA 管理区域内的所有 TNs 瘫痪或被攻破丧失了平台完整可靠性，也并不影响到其他区域。

3.6.2　TCCP 改进模型的安全分析

在大幅提高原始 TCCP 模型性能的情况下，TCCP 模型的安全性也要得到保证。在原始 TCCP 模型中，TC 负责认证 TNs 并且参与到 VM 的创建与迁移过程中，若 TC 被攻破则 TN 的身份与其相关联的活动一览无余。本文中 TC 只参与认证 TNs 而并不参与云端 VM 的管理过程，因此无法获悉云端内部运作情况（这也是 CSPs 期望的）。

本文借鉴的两种 DAA 策略[15][22]都被证明在基于随机预言机模型的强 RSA 假定下与决策性 Diffie-Hellman 假定下是安全的，本文对 DAA 策略的修改并不会影响其安全证明的正确性。

本文设置多个 Privacy CAs 的隔离机制不仅能提高 TCCP 模型的性能，也能提高 TCCP 模型的安全程度。若 Privacy CA 被攻破，其影响力只限于其管理的区域。

Privacy CAs 选举 ITC 时本文采用了 SMC 算法，能够在一定程度上避免恶意 Privacy CAs 操纵选举结果（在无条件安全模型下无赖 Privacy CAs 数目小于总数 1/3，条件安全模型下无赖 Privacy CAs 数目小于总数 1/5 的情况下）。

在虚拟可信系统架构的设计方面主要是添加了一个 TN 启动后强制优先运行的 THVM，该 THVM 运行在黑盒可信环境中且内部运行的是安全 OS。THVM 取代 Domain0 管理 vTPM 实例降低了管理员干涉的机会，此外在运行 DAA 策略

时 THVM 负责的是诚实 host 的计算以及在所属 TN 具有 ITC 或 Privacy CA 角色时负责角色需要的计算，相关数据与应用在 THVM 内部得到了很好的隐私保护，这也使得 TMR 对于 CSPs 与黑客是不可见的。

通过引入 iTurtle 概念，让用户了解自己 VM 所处的平台状态成为可能，这是解决用户与 CSPs 信任问题的关键之一。在用户创建 VM 时，为了安全方面考虑，iTurtle 还是需要首先联系 TCM，这是因为只有 TCM 能够确认 TPM 的有效性，CSPs 不可能允许含有无赖 TPM 的 iTurtle 验证云端内部的 TN 可信完整性状态。

TCCP 改进模型在安全方面的主要不足在于两个方面：1）由于采用两种不同的 DAA 策略，验证者在参与过程中能够了解证明者的角色；2）由于采用了原始 DAA 策略，因此 TCCP 模型具有同样的不足，即为了检测无赖 TPM，在 Sign 协议中ζ保持不变，这使得验证者能够将同一证明者的不同活动联系到一起。由于 DAA 策略与 Privacy CA 策略只在 TNs 之间运行，这也使得上述两点不足造成的危害减轻了。

3.7 本章小结

本章提出了一种基于 DAA 与 Privacy CA 策略的改进 TCCP 模型，首先详细说明了 TCCP 模型的总体架构设计，然后说明了针对 TCCP 模型总体架构的特征与云计算平台的需求而提出的虚拟可信平台架构设计。接着本文详细说明了 TCCP 模型可信管理关系的建立过程，以及可信管理关系稳定后新 TN 加入，VM 创建与迁移的过程，这里面都使用了 DAA 与 Privacy CA 策略。最后我们对提出的 TCCP 改进模型进行了性能与安全性的定性分析。

第 4 章　总结与展望

本文深入研究了与可信云计算平台模型相关的内容包括云计算的关键技术，云计算面临的潜在安全风险和可信计算的研究现状等内容，仔细分析了原始 TCCP 的局限性，进一步地，提出了一种基于 DAA 与 Privacy CA 策略的改进 TCCP 模型，相比较原始 TCCP 模型有着更高的实用性，可靠性和安全性。

由于用户丧失了对上传到云端的数据与应用的控制权，而且云端服务提供商较少披露云端内部的情况又使得用户缺乏必要的知情权，由此引发的信任问题正成为云计算快速发展道路中的最大障碍。2009 年发表的第一篇结合可信计算技术提出一种可信云计算平台概念的论文为信任问题的解决提供了很好的思路，在此之后陆续有厂商（例如 IBM 和 EMC）都宣布进行可信云计算的研究，然而对外透露的消息却很少。本文针对原始 TCCP 模型的局限性，以 vTPM 架构和 Eucalytus 架构为载体，做了如下的研究：

1. 对原始 TCCP 模型进行了深入研究，包括架构设计和 VM 管理协议的设计，发现 TCCP 模型的性能瓶颈在 TC 上，要提高 TCCP 模型的实用性与可靠性要先从 TC 入手；
2. 根据 TPM 的中立特性，发现可以使用云端内部运行正常且含有有效 TPM 的服务器（即 TN）代替 TC 管理 TNs 列表；
3. 考虑到单个服务器性能的有限性，使用多个 TN 代替 TC 管理 TNs 列表，这就是 Privacy CA 的雏形。每个 Privacy CA 管理 Zone 内部分 TNs 列表，不仅能提高模型的性能，也能提供一种隔离机制保证模型的安全可靠性；
4. 考虑到 Privacy CA 如何向 TN 证明自己的角色，TN 如何知道管理自己的 Privacy CA，在 VM 创建与迁移时如何不在 TC 的协助下找到合适的目的 TN 等问题，本文引入了两种 DAA 策略，经过修改与混合使用来达到解决上述问题的目的；
5. 在 ITC 选举过程中通过引入安全多方计算的概念提高了 TCCP 模型的抗攻击性；
6. 让 VM 迁移时优先考虑所属管理区域的目的 TN，提高了 TCCP 模型的效率；
7. 考虑到用户如何知道自己的 VM 处在可信黑盒环境中的问题，引入了 iTurtle 概念并作出了本文的解释。

云计算技术还在不断发展，可信云计算平台的研究更是处于初步阶段，在目

前看来本文提出的 TCCP 模型主要有下列地方需要改进，包括：

1. 在远程证明时，本文直接使用完整性度量结果 hash 值来判断平台的完整性，在实际情况中一般是不可行的，对此建议使用属性证明的方法替代（本文没有采用属性证明的方法主要是因为这会增加 TCCP 模型的复杂性，从而增大了理解本文改进 TCCP 模型核心思想的难度）；
2. ITC 的 SMC 选举算法过于简单，只能用于说明本文的解决思路，更合适的 SMC 算法需要进一步研究，同理本文采用的随机性选择算法也过于简单，需要更好的算法取代；
3. 由于不同的角色采用不同的 DAA 策略，证明平台完整可信性时验证者能够发现证明者的角色身份，虽然说参与方都是可信节点。这个安全隐患会造成多大的危害目前还没有结论。
4. 本文将 iTurtle 看作 TN 的简化版并以 USB 的形式插在用户电脑使用，问题是用户电脑里可能运行了各种个性化的软件，平台软件栈相当复杂，iTurtle 的信任链应该起始于何处以及 iTurtle 如何自我检测防止用户电脑的恶意软件入侵都是亟待解决的问题。另外在本文中轻量级的 iTurtle 可能承担不了 DAA 策略需要的大计算量。

参考文献

[1] N. Santos, K. P. Gummadi, and R. Rodrigue, 'Towards Trusted Cloud Computing", In Proc.of the 1st USENIX Workshop on Hot Topics in Cloud Computing, Berkeley, CA, USA, 2009.

[2] http://zh.wikipedia.org/wiki/虚拟化

[3] D.Nurmi, R.Wolski, c.Grzegorczyk, G.Obertelli, S.Soman, Youseff, and DZagorodnov. "Eucalyptus: A Technical Report on an Elastic Utility Computing Architecture Linking Your Programs to Useful Systems", TechnicaIReport2008-10,UCSB.

[4] http://baike.baidu.com/view/3959716.htm

[5] T.Mather, S.Kumaraswamy, S.Latif, "Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance", O ’ Reilly Media, Sebastopol , 30, 2009.

[6] S. Berger, R. C'aceres, K. A. Goldman, R. Perez, R. Sailer, and L. vanDoorn. "vTPM: virtualizing the trusted platform module", In *Proc.of USENIX-SS'06*, Berkeley, CA, USA, 2006.

[7] D. Challener, K. Yoder, R. Catherman, D. Safford, L. V. Doorn. "A Practical Guide to Trusted Computing", IBM Press , 16-17, 2007

[8] J. Poritz, M. Schunter, E. Van Herreweghen, and M. Waidner. "Property attestation - scalable and privacy-friendly assessment of peer computers", IBM research report RZ3548, 2004.

[9] P. England. "Practical techniques for operating system attestation", In *Proc. TRUST 2008*, volume 4968 of LNCS. Springer-Verlag, 1-13, 2008.

[10] T. Garfinkel, B. Pfaff, J. Chow, M. Rosenblum, and D. Boneh. 'Terra: a virtual machine-based platform for trusted computing", In *Proc. Of SOSP'03*, 2003.

[11] M. Peinado, Y. Chen, P. England, and J. Manferdelli. "NGSCB: A Trusted Open System", In *Proc. of 9th AClSP*, Sydney, Australia, 2004.

[12] A. Shieh, D. Williams, E.G. Sirer and F.B.Schneider. "Nexus: a new operating system for trustworthy computing", In *Proc. of 12th ACM symposium on Operating systems principles (SOSP 2005)* , New York, 2005.

[13] B. Pfitzmann, J. Riordan, C. Stuble, M. Waidner and A. Weber. "The PERSEUS system architecture", Technical Report RZ 3335 (#93381), IBM Research Division, Zurich Laboratory, 2001.

[14] B. D. Payne, M. Carbone, and W.Lee. "Secure and Flexible Monitoring of Virtual Machines", In *Proc. of ACSAC'07*, 2007.

[15] E. Brickell, J. Camenisch, and L. Chen. "Direct anonymous attestation", In *11th ACM Conference on Computer and Communications Security*, ACMPress, 2004.

[16] J. Camenisch and A. Lysyanskaya. "A Signature scheme with effcient protocols", In *Security in communication networks*, vol. 2576 of LNCS, 268-289, 2002.

[17] A. Fiat and A. Shamir, “How to prove yourself: Practical solutions to identification and Signature problems”, In *Proc. of Crypto’86*, 186–194, 1986.

[18] http://en.wikipedia.org/wiki/Zero-knowledge_proof

[19] D. Pointcheval and J. Stern, "Security proofs for Signatures", Advances in *cryptology-Eurocrypt’96* , 1996.

[20] C. P. Schnorr, "Efficient Signature generation for smart cards", *Journal of Cryptology*, 4(3):239-252, 1991.

[21] C. Rudolph, "Covert identity information in direct anonymous attestation (DAA)", *In Proc. of the 22nd IFIP TC-11 International Information Security Conference (SEC 2007)*, 2007.

[22] J. Camenisch, "Better privacy for trusted computing platforms", In *9th European Symposium On Research in Computer Security (ESORICS 2004)*, 2004.

[23] http://www.daoliproject.org/

[24] http://www.trusted-cloud.com/

[25] https://cloudsecurityalliance.org/trustedcloud.html

[26] A. R. Sadeghi and C. Stüble, "Property-based attestation for computing platforms: Caring about properties, not mechanisms", In *New Security Paradigms Workshop 2004*, 2004.

[27] A. C. Yao, "Protocols for Secure Computations (Extended Abstract)" In *Proc. of the 23rd FOCS* , 160-164, 1982.

[28] U. Kühn, M. Selhorst and C. Stble, "Realizing property-based attestation and sealing with commonly available hard- and software", In *Proc. of STC 2007*, 1~16, 2007.

[29] L. Chen , H. Löhr , M. Manulis , A. Sadeghi, "Property-Based Attestation without a Trusted Third Party", In *Proc. of the 11th international conference on Information Security*, Taipei, 2008.

[30] J. M. McCune, A. Perrig, A. Seshadri, and L. van Doorn, "Turtles all the way down: Research challenges in user-based attestation", In *Usenix Workshop on HotSec'07*, 2007

致 谢

在论文即将完成的时候，不由得想起本科做毕业设计的那段时光，为解决方案，实验方式，数据处理等等方面的设计而愁苦的画面还历历在目，时光飞梭，转眼间又到了研究生毕业的时候了。庆幸的是三年来在中国科学技术大学的求学历程不仅让我对信息安全学科的知识掌握程度更深了一步，在研究问题方面也学到了很多有价值的思考方式和实验方法，让我的心智更加完善，能够面对更多的挫折与磨难，我想自己现在才真正准备好了走入社会贡献自己的一份力量。我的这一切成长自然离不开老师们的谆谆教诲以及同学们的坦诚帮助。

首先我想要感谢中国科学技术大学为了我提供了这么优秀的深造平台，不仅为我们提供了很好的实验环境，在生活上也尽量为我们提供方便。

在学业上的个人成长主要感谢我的导师黄刘生教授，黄刘生教授影响我最深的是他严谨中立的研究态度，宽阔的思维，以及和蔼可亲的人格魅力。在研究生学习生涯中，黄教授多次和我谈论了信息安全学科的研究方法，信息安全研究方向的选择以及就业还是继续深造的选择，在交谈中黄教授不仅是个优秀的老师指引我的学习，更像一位知心朋友设身处地分析和解决我的困惑，比如在研究方向选择上面黄教授本来期望我从事物联网的可信安全研究，不过由于我个人倾向于研究云计算的可信安全，黄教授也表示了理解和支持。这三年来黄教授教给我的严谨治学的研究态度，不惧挫折开拓创新的研究精神以及为人处世的人格信条是我人生最大的收益之一。

我还要感谢负责信息安全实验室的杨威老师对我的指导，由于黄教授日理万机奔波于苏州与合肥两地实验室，更多时候我的问题都向杨威老师请教，在我研究领域的很多细节问题上杨威老师都给予了足够的指导，还记得发表英文论文时杨威老师仔细地为我查找出了多处语法错误。在我研究过程中遇到挫折而沮丧时杨威老师开朗乐观的个性也感染了我，让我恢复了深入研究下去的信心，在此我再一次对杨威老师的细致教导表示感谢。

这三年来信息安全实验室给了我很多温暖，不管是学习还是生活上，同学们都无私地互相帮助互相关怀，感谢各位学长学姐对我的指导，这里特别感谢余振山师兄，董凡师兄，朱有文师兄，叶云师兄，纪雯师姐和胡智慧师姐。

感谢何立宝同学，王剑锋师弟，韩典琪师弟，张冀雨师妹和张琛师妹，和你们在一个实验室相处感觉很愉快。感谢室友王金锭同学，杨俊同学与我共度三年时光。

最后我要深深地感谢父母对我的养育与栽培之恩，如果没有他们对我的爱，

没有 24 年来无私的付出以及对我各方面的完善教育，我不可能走到今天。在我遇到挫折时父母总是我的坚强后盾，我也为我至今没能很好的回报他们感到羞愧，在此我给我的父母鞠一个躬表达我深深的感激与歉意。

谨以此文献给我爱的人和爱我的人。

在读期间发表的论文与取得的研究成果

已发表论文：

[1] **Wang Han-zhang**, Huang Liu-sheng, “An improved trusted cloud computing platform model based on DAA and privacy CA scheme”, 2010 International Conference on Computer Application and System Modeling , ICCASM 2010, (**EI No. 20104913453626**)

参加的科研项目情况

参加的科研项目

1. 参加国家自然科学基金重大研究计划项目（90818005）：
   可信的安全多方计算环境构建方法及其关键技术研究
2. 参加国家自然科学基金项目（60703071）：
   可信计算中基于隐私数据的信息共享技术研究
3. 参加江苏省自然科学基金项目（BK2007060）：
   网络计算中的隐私保护问题及其应用研究
4. 参与安徽省自然科学基金项目（070411043）：
   安全多方计算基础协议及其应用研究

可信云计算平台模型的研究及其改进

作者：王含章
学位授予单位：中国科学技术大学