基于 Eucalyptus 的基础设施即服务云框架协议设计

崔 巍* 李益发 斯雪明
(信息工程大学信息工程学院 郑州 450002)

摘 要：云计算中的基础设施即服务(IaaS)免去用户自主管理计算机硬件的麻烦，随时随地按需向用户提供计算和存储资源。Eucalyptus 是一个被学术研究关注的开源 IaaS 实现，然而没有文献描述完整的利用 Eucalyptus 的所有模块来提供安全的基础设施服务。该文针对 IaaS 的安全需求，提出一个可信的 IaaS 框架。框架将基础设施服务提供过程细化为 5 个环节，并基于可信平台模块(TPM)设计相应安全协议实现这些环节。协议的设计过程严格遵守 TPM 的操作规范，并加入可信第三方以制约服务提供商的权力。协议的安全性均通过 Scyther 自动化分析工具的检测，从而保证整个框架满足 IaaS 的安全需求。

关键词：云计算；基础设施即服务(IaaS)；Eucalyptus；可信计算
中图分类号：TP309 **文献标识码**： A **文章编号**：1009-5896(2012)07-1748-07
DOI: 10.3724/SP.J.1146.2011.01150

The Protocol Design of a Eucalyptus-based Infrastructure-as-a-Service (IaaS) Cloud Framework

Cui Wei Li Yi-fa Si Xue-ming
(*School of Information Engineering*, *Information Engineering University*, *Zhengzhou* 450002, *China*)

Abstract: Infrastructure-as-a-Service (IaaS) frees users from the trouble of self-management of computer hardware, and provides users with anytime, anywhere on demand computing and storage resources. Eucalyptus is an open source IaaS framework implementation which is used for research. However, there is no paper describes how to use all the modules of Eucalyptus to supply security infrastructure service. In accordance with the security requirements of IaaS, a trusted IaaS framework is provided. The framework provides infrastructure service in five steps, and designs protocols which based on Trusted Platform Module (TPM) to achieve these steps. During the designing process, the use of TPM is strictly standardized and trusted third party is concerned in order to restrict the power of service operator. All the protocols pass the security examination of automatic analysis tool-Scyther, so the conclusion that the framework meets the requirements of IaaS is generalized.

Key words: Cloud computing; Infrastructure-as-a-Service (IaaS); Eucalyptus; Trusted computing

1 引言

云计算代表信息技术领域向集约化、规模化和专业化道路发展的趋势，是信息技术领域正在发生的深刻变革。但是，随着云计算服务的普及，其安全问题也随之凸现出来，已经成为制约云计算发展的重要因素。因此，推动云计算的大规模应用，必须首先解决云计算面临的各种安全问题。

美国国家标准与技术研究院(NIST)将云计算服务分为 3 个层次：基础设施即服务(IaaS)、平台即服务(PaaS)与软件即服务(SaaS)。其中，IaaS 可以按需向用户提供量化的计算和存储资源。利用 IaaS，一些 IT 厂商不再需要建立和维护自己的机群，而只需花费相对较少的代价来使用 IaaS 提供的资源，从而更加专注于自身核心业务的发展。由此，IaaS 吸引了更多企业的关注，再加之 IaaS 提供的资源属于底层关键资源，因而它的安全性显得尤为重要[1]。

当前一些大型 IT 企业已经开始面向公众提供 IaaS，例如 Amazon 的 EC2。但此类 IaaS 的实现细节均没有被公布。学术上对于 IaaS 安全性的研究集中在一个开源的 IaaS 实现：Eucalyptus[2]。在 IaaS 提供过程中，服务提供商管理虚拟机(VM)的创建与运行，而 VM 的创建依赖于用户上传的特定虚拟机映像(VMI)或者服务提供商预定义的一般 VMI。用户关注 VM 的运行回馈，从而获得服务提供商给予的资源。

文献[3]首次基于 Eucalyptus 讨论用户使用基础

2011-11-07 收到，2012-01-18 改回
国家 863 计划项目(2009AA012201)和现代通信实验室预研项目(9140C1103040902)资助课题
*通信作者：崔巍 mortimercui@gmail.com

设施服务时面临的安全问题。文章引入虚拟机监视器(VMM)为用户 VM 构建一个封闭的运行环境，从而确保用户 VM 不被非法用户访问，并通过可信第三方和可信平台模块(TPM)使得服务提供商能够向用户远程证明 VM 的安全性。这种思想被后续的研究所采用。

文献[4]进一步改进文献[3]的工作。同文献[3]一样，它基于 Eucalyptus 模型，引入 TPM、可信第三方和 VMM，设计了 3 个协议，用于实现用户 VM 在云端的可信计算和安全迁移。但这两篇文献的设计均存在一些缺陷：

(1)两篇文献均没有讨论存储控制器在VM部署过程中的作用，而实际上存储控制器作为 Eucalyptus 架构的一个主要组成部分，在确保 VM 机密性与恶意代码检查上有重要作用。

(2)两篇文献在使用 TPM 设计协议时均忽视TPM密钥使用方面的细节，例如使用签注密钥(EK)进行数据签名，同时使用身份证明密钥进行加密，这两个操作可能带来安全上的隐患[5]。

(3)两篇文献对设计的协议均没有进行安全性分析。协议的安全性分析可以增加人们使用该协议的信心，应该属于协议设计的一个重要组成部分。

文献[6]设计了可信云(CertiCloud)来保证 IaaS 层的安全。文章仍然沿用了文献[3]的设计思想，但是着重从用户的角度进行安全考虑。通过可信云可以保证用户 VM 的安全部署，并且能够按照用户的需求，验证 VM 的运行情况。但是，文章在设计过程中没有引入可信第三方，导致对云服务提供商的约束较小，这可能带来安全隐患。并且，云端 TPM 向用户传递平台配置寄存器(PCR)信息后，文章假设用户自行验证 PCR 信息的正确性，而用户实际上还是求助于可信第三方进行该操作，不如在协议过程中考虑可信第三方，这样可以进一步完整描述协议的运行过程，同时提高协议的整体运行效率。再者，文章没有考虑 VM 迁移。VM 迁移对云端的负载平衡意义重大，而 VM 迁移后，用户在验证 VM 运行情况时，会话节点也发生了改变，文章设计的 VerifyMyVM 协议将不能发挥效用。

本文考察 IaaS 提供过程中，用户和服务提供商关心的安全问题，提炼为 3 点：

(1)用户如何确认所使用的云服务是可信的？

(2)云服务提供商如何保证用户VM在云中的建立和运行过程是安全的？又如何向用户证明安全？

(3)云服务提供商如何防止用户在使用云资源时恶意攻击云？

针对这 3 个安全问题，本文基于 Eucalyptus 提出一种可信 IaaS 云框架。为了实现对云服务提供商的约束，在框架的设计中引入可信第三方。而可信第三方的引入在处理云安全问题中被认为是十分必要的[7]。

该框架涉及 Eucalyptus 中 4 个主要部件(云控制器、集群控制器、节点控制器以及存储控制器)的相互协作。基础设施服务的提供过程被细化为 5 个环节，分别为：节点注册，用户上传 VMI，启动 VM, VM 迁移以及 VM 的终止。前 4 个环节通过本文设计的 4 个协议实现，最后一个环节由 VMM 管理。框架设计的核心思想是以可信平台模块(TPM)和可信第三方为信任基，通过协议的执行过程，形成用户与服务提供商之间的信任链条，从而解决双方关心的 3 个安全问题。可信 IaaS 云框架弥补了上述文献的设计中存在的问题。

协议的安全性分析是一个需要细致推理的工作。当前有一些流行的协议分析模型与自动化分析工具。例如协议组合逻辑(PCL)[8]，泛组合模型(UC)[9], ASPIER[10]和 Scyther[11]自动化分析工具。限于篇幅，本文没有提供对可信 IaaS 云框架以及 4 个安全协议在协议分析模型下的安全性证明，该证明将另文发表。但对每个协议均使用 Scyther 进行安全性检测，以保证所设计协议的安全性。

以下第 2 节介绍可信 IaaS 云框架的设计背景，第 3 节介绍其整体设计与技术细节，第 4 节是总结。

2　设计背景

2.1 Eucalyptus

Eucalyptus 基于模块化的思想构建，共有 4 个模块：云控制器(CLC)，集群控制器(CC)，存储控制器(CS)，节点控制器(NC)。这 4 个模块共同协作，完成整个基础设施服务的提供与内部管理。

图 1 为 Eucalyptus 的整体架构。CLC 为用户提供进入云的接口，使得用户可以操作 VM。CLC 根据用户的要求总体协调，并向相应的 CC 发送任务。CC 管理集群内部的 NC，收集它们的运行状态，接受 CLC 的任务并分配至相应的 NC。CS 提供存储以及进入 VMI 和用户数据的机制。本文在协议设计过程中，忽略 CLC 与 CC, NC 和 CS 的交互过程，因为它们之间的交互属于云内部的协作，对于用户是透明的。从用户的角度可以单纯地看作与 NC 和 CS 进行交互。

云计算的最大特点是 IT 资源服务化。而这又依赖于虚拟化的设计思想。用户向 IaaS 层请求计算与存储资源时，通常是由云在某些节点上建立 VM 并分配相应的计算与存储资源。VM 的启动与运行是遵照 VMI 实施的。用户可以通过向 CS 提交 VMI

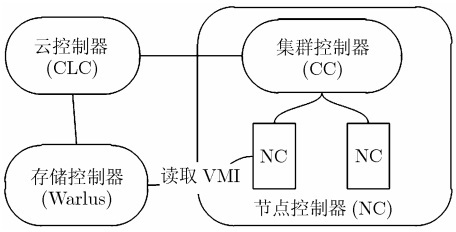

图 1 Eucalyptus 架构

的方式指定云端建立的 VM 运行自己设计的程序，也可以使用云端提供的 VMI。对于后一种情况，NC 直接获取 CS 中存储的 VMI 即可，不涉及到用户向 CS 传输 VMI 的过程。因此，后一种情况是前一种的简化，文章仅考虑前一种情形，后一种可类似地实现。CS 将 VMI 压缩、加密后分散存储起来。当 NC 向 CS 请求 VMI 时，需要出示相应的凭据，如果认证通过，VMI 就被解密和解压缩，并发送至 NC。Eucalyptus 的管理过程与 Amazon 设计的 EC2 是相同的。

在 NC 端，我们假设 VM 的管理是由安全的虚拟机监视器(SVMM)实施的。SVMM 可以为 VM 提供一个封闭的运行环境，阻止恶意用户访问和修改 VM 的资源，同时防止 VM 中的恶意代码攻击 NC。VM 启动后，未必固定在原始节点运行，随着 Eucalyptus 运行状态的变化，有时可能会将用户 VM 从一个节点迁移至另一个节点，以保持整体负载平衡。

2.2 可信平台模块

可信平台模块(TPM)是由可信计算组织(TCG)[12]提出的一种绑定在计算机主板上的硬件设施。TPM 包含一个处理器和两块存储区。处理器硬件实现了一些基本的密码算法，使得 TPM 可以进行诸如加解密、签名、随机数生成、Hash 等运算。两块存储区，一块不可改写，并在 TPM 出厂时存储两个密钥对。一个密钥为签注密钥(EK)，用于唯一标志该 TPM，EK 的公钥部分由该厂商签名并公布，保证 TPM 的合法性。另一个密钥为身份证明密钥(AIK)，出于安全和隐私考虑，EK 通常不用于签名操作，而 AIK 则用于签名 TPM 内部产生的数据，包括 PCR，其他密钥和 TPM 状态信息，从而提供 TPM 的身份证实，并且 AIK 从不用于加密数据或签名 TPM 外部提供的数据[5]。认证机构(CA)会向公众证明 AIK 与从属于 TPM 的 EK 的绑定关系，从而防止身份伪造。另一个可改写的存储区包含一组平台配置寄存器(PCR)。在计算机启动时，TPM 会依次计算基本输入输出系统(BIOS)、启动加载项、VMM 以及计算机加载的所有应用软件的 Hash，并维护一个测量列表(ML)。ML 以明文形式存放在 PCR 中。TPM 隐蔽 PCR 的地址，使得恶意程序无法更改 PCR 的值。ML 是用于远程可信证明的重要数据，通过它，计算机可以向第三方证明当前运行的软件是合法的，没有病毒、木马或恶意程序的加载。

总体来说，TPM 是计算机系统的监视者，它可以随时测量计算机系统的运行信息，并向第三方汇报信息，而系统却无法访问它。本文假设云端的协议参与方均安装了 TPM 芯片，用户端不必安装 TPM 芯片，但是拥有 CA 颁发的一对公私钥。协议参与方均从 CA 处获知其他参与方的公钥与其身份的正确绑定关系。

2.3 协议安全性分析

安全协议的设计是一项十分精细的工作，而且往往容易产生难以察觉的错误。例如 Needham-Schroeder 协议在使用了 17 年后才被发现存在漏洞。为了系统分析协议的安全性，诞生了许多协议自动化分析工具与形式化分析模型。Scyther 是当前使用较多的一种协议自动化分析工具。它能根据用户描述的协议安全目标与设定的搜索界，自动进行协议安全性证明与攻击路径搜索。一旦搜索到攻击路径，还会形成可视的攻击路径图，对于查找当前协议的漏洞并有针对性地改进协议很有帮助。

3 整体设计与实现细节

以下介绍可信 IaaS 云框架的整体设计与技术细节。在设计时，我们屏蔽 CLC, CC 管理 NC 和 CS 的过程，认为用户直接与 NC 和 CS 进行交互。假设存在一个可信第三方，称作可信中心(TC)，它与云服务提供商没有利益上的交互，因此不会合谋。TC 的任务是协助云向用户证明服务的安全性。

图 2 为可信 IaaS 云框架的整体设计。TC 监管整个云，用户在与云进行交互时，会先确认云是否得到了 TC 的信任。因此，NC 和 CS 必须先在 TC 处进行注册，这是云提供服务的基础。TC 认为一个可信的 NC 或 CS 必须处在自己监管的安全域内，没有恶意程序的运行，同时运行相关的合法程序，例如 NC 必须运行 SVMM，而 CS 必须运行安全的 VMI 存储管理软件。

随后用户可能会向 CS 上传自己的 VMI。上传完毕后，CS 将 VMI 压缩解密存储起来。此时，CS 可以根据需要对用户的 VMI 进行检查，以便确认用户的 VMI 没有包含恶意程序。这一步和 SVMM 对 VM 的控制是云防止用户恶意攻击的主要措施。这里面还可能涉及到用户 VMI 的机密问题，例如 VMI

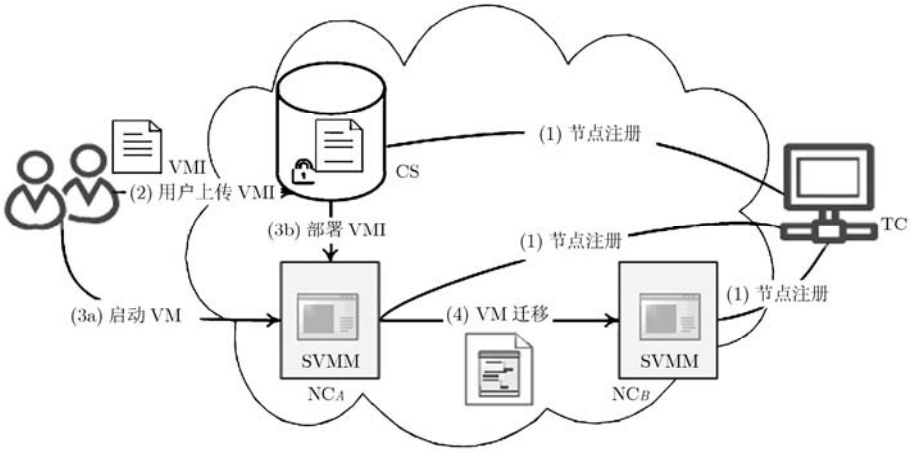

图 2 可信 IaaS 云整体设计

里面可能包含具有知识产权的程序，而用户不希望第三者得到该 VMI 的拷贝。而这涉及到一个相对复杂的云存储安全问题，解决该问题不属于本文探讨的内容。我们这里假设，TC 在注册 CS 时，通过检查 TPM 返回的 CS 的系统信息，可以向用户保证这类问题不会发生。同时，云内部建立足够好的监管机制，可以防止通过访问 CS 物理介质的方式取用数据。

当用户希望获取基础设施服务时，他与 NC 进行交互，建立 VM 并交予 NC 进行管理。在这个过程中，用户首先会验证 NC 的可信性，然后 NC 会利用用户传来的凭据，与 CS 进行交互，取用 VMI，并建立 VM。若前两个环节成功实施，用户可以保证 VM 运行在可信环境中，同时取用了正确的 VMI。随后，NC 管理 VM 的运行和停止。当云内部需要调整负载时，NC 可能将 VM 交予另一台 NC 进行管理，这涉及到 VM 的迁移。最终，当 VM 执行完毕后 SVMM 停止该程序从而完成整个服务提供过程。由此可见，IaaS 服务的提供可以细化为 5 个主要环节：(1)云节点向 TC 进行注册；(2)用户向 CS 上传 VMI；(3)启动 VM；(4)VM 的迁移；(5)VM 的停止。其中 VM 的安全停止由 SVMM 管理，其余 4 个环节通过 4 个安全协议实现。

3.1 节点注册协议

节点注册是其他环节的基础。本协议完成节点与 TC 的双向认证，并检查对方的 PCR，确保对方执行的程序是可信的。协议使用 NC 示例，CS 的注册过程与此完全一致。

图 3 所示为节点注册协议。(1)NC 首先通过向 TC 传递随机串与自己的名称发起注册请求。(2)TC 收到请求后，从 TPM 获得自己 PCR 信息的拷贝。PCR 信息与随机串一起用 TPM 的 AIK 进行签名，构成 PCR 验证信息。这可以保证 PCR 验证信息的正确来源与时效性。然后 TC 将 TPM 发来的信息连同 NC 传递的随机串一起用 NC 的 EK 进行加密并传递给 NC，后者又直接传递给自己的 TPM。(3)NC 的 TPM 可以解密 TC 传来的消息，之后使用相同的方式构建 PCR 验证信息，并一起传递给 NC。NC 首先验证 TC 的 PCR 消息是否正确。这里我们假设 TC 做为一个可信机构已经通过某种方式(例如互联网)公布自己正确的 PCR 信息，NC 可以将该信息与公布的信息进行比对，以确认 TC 的可信性。验证完毕后，NC 将自己的 PCR 验证消息连同前两步获得的随机串一起用 TC 的 EK 加密传递过去，后者传递给自己的 TPM。(4)TC 的 TPM 解密消息并将得到的部分传递给 TC, TC 首先验证 NC 是否执行合法代码，验证成功后，将 NC 的 PCR 验证信息里面的随机串加密传给 NC，后者传递给自己的 TPM。(5)NC 的 TPM 解密后，通过随机串确保消息的时效性，随后返回给 NC 协议执行成功的消息。至此协议执行完毕。随后 TC 记录节点名称，EK 公钥部分，AIK 公钥部分，PCR 信息，以便在执行后面的协议时，帮助用户确认节点是否可信。NC 如果重启，将重复执行上面的步骤。这里 TPM 有一个检查机制，它将记录每次启动时获得的 PCR 是否向 TC 上传过。如果节点注册协议没有成功完成，TPM 在后面的协议中会拒绝为 NC 提供签名、加密等操作，导致 NC 无法进行后续的协议。这就保证了 NC 每次启动后都经过 TC 的注册。节点注册协议执行成功后，可以将节点的 TPM 与节点看成一个整体。在以后的协议描述中也将省略二者之间的通信过程。

使用 Scyther 进行安全性分析时，根据这一环节的安全需求，我们设定协议的安全目标是 PCR 信

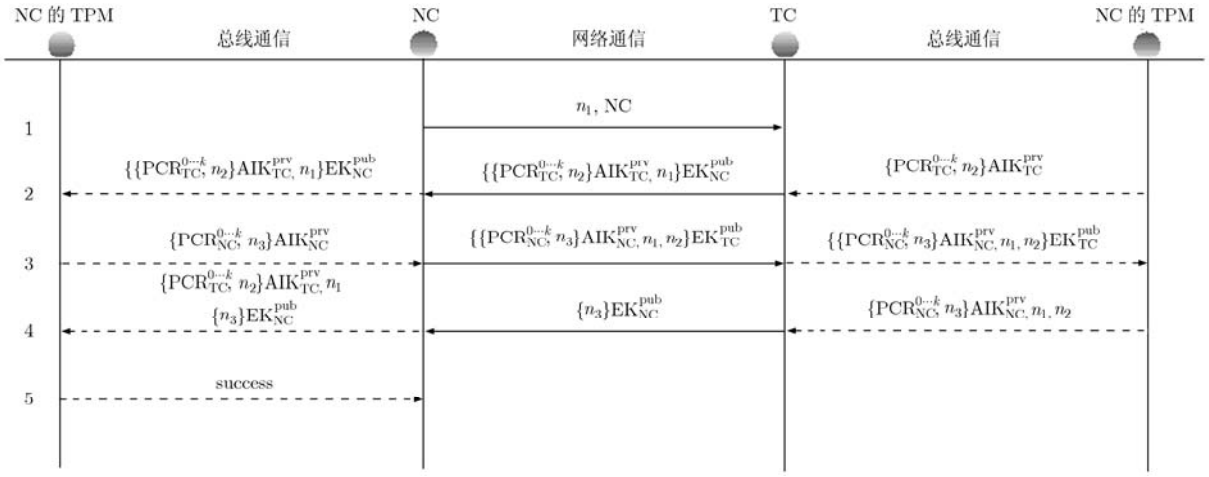

图 3 节点注册协议

息的机密性以及 NC 与 TC 的双向认证和同步执行。协议通过 Scyther 的安全性分析。

3.2 用户上传 VMI 协议

图 4 所示为用户上传 VMI 的过程。执行该协议前，CS 首先要通过节点注册协议在 TC 那里成功注册。用户和 CS 在本协议中，利用 TC 提供的信息，首先完成双向认证和密钥协商，然后用户利用协商的密钥上传其 VMI。上传完毕后，CS 对 VMI 进行保存和集中存放。

协议步骤说明：(1)首先用户将随机串与 VMI 加密密钥 VK 一起签名，添加上期望提供服务的 CS 的名称，一起用 TC 的 EK 加密，传递给 CS。(2)CS 将用户传来的消息、自己的名称和产生的随机串的签名一起用 TC 的 EK 加密传递给 TC。(3)TC 解密后，确认消息来源方与用户期望的 CS 是一致的，并检索 CS 是否已注册。然后将解密得到的两个随机串与 VK 一起用 CS 的 EK 加密并用自己的 AIK 签名，发至 CS。(4)CS 此时可以获得 VK，它将用户的随机串用 VK 加密返回用户。用户通过解密确认是此次会话的消息，从而完成协议。随后，用户使用 VK 加密传输 VMI 至 CS。VK 也成为日后取用 VMI 的唯一凭据。

使用 Scyther 进行安全性分析时，根据这一环节的安全需求，我们设定协议的安全目标是 U, CS 与 TC 的双向认证和同步执行，同时 VK 满足机密性。协议通过 Scyther 的安全性分析。

3.3 启动 VM 协议

当用户希望获得基础设施服务时会运行启动 VM 协议。用户先前验证 CS 的可信性，并与 CS 成功完成上传 VMI 协议，本协议中的 index 是取用 VMI 的凭据，也就是上传 VMI 协议中的 VK。现在用户需要在 TC 的帮助下验证 NC 的可信性，并完成取用 VMI 并在 NC 上启动 VM 的过程。

图 5 所示为启动 VM 协议。(1)用户生成会话密

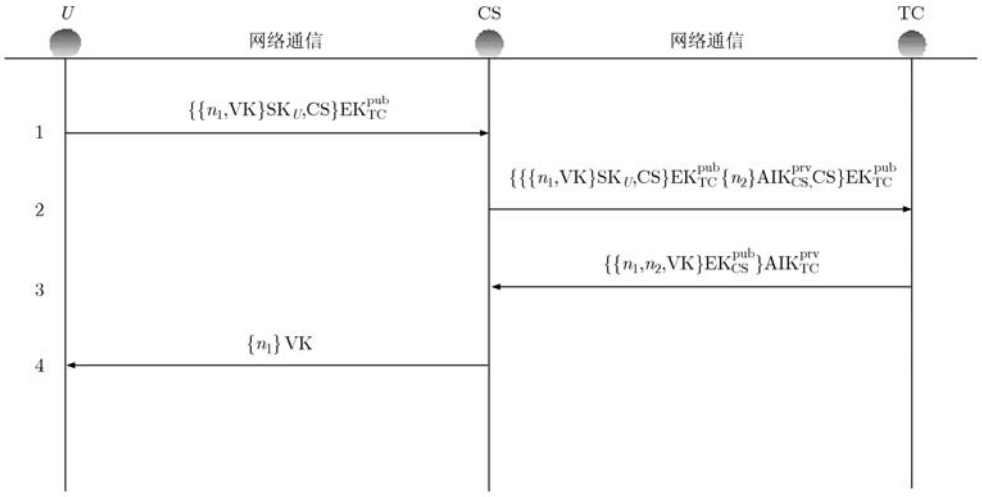

图 4 用户上传 VMI 协议

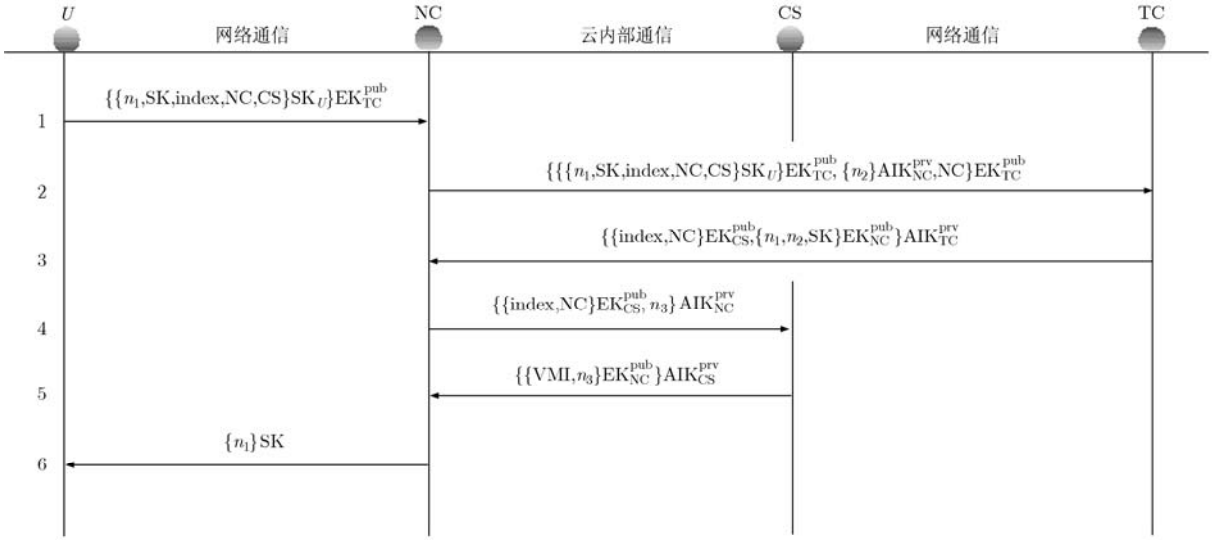

图 5 启动 VM 协议

钥 SK，并与随机串，index, NC 的名称，CS 的名称一起签名并用 TC 的 EK 加密，发至 NC。(2)NC 将自己产生的随机数的签名、自己的名称与用户发来的信息一起用 TC 的 EK 加密，发至 TC。(3)TC 收到信息后，首先检查 NC 与 CS 是否已成功注册，然后将 index 和 NC 的名称单独使用 CS 的 EK 加密，以便这部分消息只有 CS 可以看到，另外将两个随机串和 SK 用 NC 的 EK 加密，并将这两个密文一起签名，发至 NC。(4)NC 将 TC 预留给 CS 的部分与新的随机串一起签名并发至 CS。(5)CS 通过消息可以得到凭据，并确认部署 VMI 的节点名称。然后 CS 根据 index 从 VMI 库中取出正确的 VMI，并与 NC 发来的随机串一起用 NC 的 EK 加密并签名，然后发至 NC。(6)NC 从消息中获得 VMI，分配资源并启动 VM，然后 NC 将用户生成的随机串用会话密钥 SK 加密返回至用户。用户随后可以使用 SK 与 NC 进行保密通信，协调 VM 的运行过程。

使用 Scyther 进行安全性分析时，根据这一环节的安全需求，我们设定协议的安全目标是 U, NC, CS 与 TC 的双向认证和同步执行，同时 SK, index, VMI 满足机密性。协议通过了 Scyther 的安全性分析。

3.4 VM 迁移协议

为了保证云的负载平衡，VM 在运行过程中，可能会从一个节点转移至另一个节点。这时原始节点 NA 需要在 TC 的帮助下，验证待转移节点 NB 是否可信。随后 NA 使用启动 VM 协议中得到的与用户进行保密通信的密钥 SK 加密传输 VM 运行状态。

图 6 所示为 VM 迁移过程。(1)NA 将 SK 签名，并与期望转移的节点名称 NB 一起用 TC 的 EK 加密发至 NB。(2)NB 生成随机串并签名，然后与自己的名称和 NA 发来的消息一起用 TC 的 EK 加密发至 TC。(3)TC 首先验证 NB 是否已成功注册。然后将 SK, NA 的随机串，NA 的名称一起用 NB 的 EK 加密并签名发至 NB。(4)NB 此时可以得到 SK，它将自己的名称用 SK 加密返回给 NA 表明已成功获得 SK。协议执行完毕后，NA 会通知用户当前管理 VM 的节点变为 NB。用户可以继续使用 SK 与 NB 进行保密通信。

使用 Scyther 进行安全性分析时，根据这一环节的安全需求，我们设定协议的安全目标是 NA，NB 与 TC 的双向认证和同步执行，同时 SK 满足机密性。协议通过了 Scyther 的安全性分析。

4　结束语

本文针对 IaaS 的安全需求，归纳出 3 个安全问题，并将基础设施服务的提供过程细化为 5 个环节，然后，基于 Eucalyptus 提出一种可信 IaaS 云框架，并且设计 4 个安全协议用以实现前 4 个环节，最后一个环节由 SVMM 实施。协议的安全性通过 Scyther 自动化分析工具的检测。在设计过程中，我们严格按照 TPM 的操作规范，并加入了可信第三方以制约服务提供商。在后 4 个环节中，每一个均以前一个环节为实施基础，并以前一个环节的安全目标为后一个环节的安全假设。经过 5 个环节的协作，服务提供商可以向用户证明所提供服务的安全性，同时，通过 CS 的 VMI 检查机制与 SVMM 为 VM 提供的封闭式运行环境可以防止用户 VM 的恶

图 6 VM 迁移协议

意攻击，完整解决了所提出的 3 个安全问题。限于篇幅，本文没有提供对可信 IaaS 云架构以及 4 个安全协议安全性的理论分析与证明。这些内容将另文发表。

参 考 文 献

[1] Luis V, Luis R, and Daniel M. Locking the sky: a survey on IaaS cloud security [J]. *Computing*, 2011, 91(1): 93–118.

[2] Daniel D, Rich W, Chris G, *et al*.. The eucalyptus open-source cloud-computing system [C]. Proceedings of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, Shanghai, China, 2009: 124–131.

[3] Nuno S, Krishna G, and Rodrigo R. Towards trusted cloud computing [C]. Proceeding of the 2009 Conference on Hot Topics in Cloud Computing, San Diego, California, 2009: 22–27.

[4] Khan I, Rehman H, and Anwar Z. Design and deployment of a trusted eucalyptus cloud [C]. 2011 IEEE International Conference on Cloud Computing (CLOUD), Washington DC, 2011: 380–387.

[5] ISO-IEC. 11889-1-2009 Information technology - trusted platform module-Part 2: design principles [S]. ISO, 2009.

[6] Berthonlon B, Varrette S, and Bouvry P. Certicloud: a novel TPM-based approach to ensure cloud IaaS security [C]. 2011 IEEE International Conference on Cloud Computing (CLOUD), Washington DC, 2011: 121–130.

[7] Dhinesh B, Venkata K, Mohammed Z, *et al*.. An analysis of security related issues in cloud computing [J]. *Communications in Computer and Information Science*, 2011, 168(2): 180–190.

[8] Anupam D, Ante D, John M, *et al*.. Protocol Composition Logic (PCL) [J]. *Electronic Notes in Theoretical Computer Science*, 2007, 172: 311–358.

[9] Canetti R. Universally composable security: a new paradigm for cryptographic protocols [C]. Proceeding of Foundations of Computer Science, NY, USA, 2001: 136–145.

[10] Chaki S and Datta A. ASPIER: an automated framework for verifying security protocol implementations [C]. Computer Security Foundations Symposium, Port Jefferson NY, 2009: 172–185.

[11] Cremers C. The scyther tool: verification, falsification, and analysis of security protocols [J]. *Lecture Notes in Computer Science*, 2008, 5123: 414–418.

[12] TCG. Trusted Computing Group. http://www.trustedcomputinggroup.org/, 2012.

崔 巍： 男，1986 年生，硕士生，研究方向为信息安全.

李益发： 男，1964 年生，副教授，硕士生导师，研究方向为信息安全.

斯雪明： 男，1966 年生，副研究员，研究方向为信息安全.