附件1.2：学术型硕士学位论文封面格式（浅黄色A4皮纹纸装订，打印时此行取消）

分类号： (按中国图书分类法) 单位代码： 10636

密 级：（注明密级与保密期限） 学 号：



**硕士学位论文**



**中文论文题目 ：** （小二号仿宋体加黑）

**英文论文题目：** （16pt Time New Roman，Bold）

论文作者： （四号仿宋）

指导教师：

专业名称：

研究方向：

所在学院：

论文提交日期： 年 月 日

论文答辩日期： 年 月 日

附件2：论文独创性及使用授权声明

四川师范大学学位论文独创性声明

本人声明：所呈交学位论文 ，是本人在导师 指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不含任何其他个人或集体已经发表或撰写过的作品或成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本声明的法律结果由本人承担。

本人承诺：已提交的学位论文电子版与论文纸本的内容一致。如因不符而引起的学术声誉上的损失由本人自负。

学位论文作者： 签字日期： 年 月 日

四川师范大学学位论文版权使用授权书

本人同意所撰写学位论文的使用授权遵照学校的管理规定：

学校作为申请学位的条件之一，学位论文著作权拥有者须授权所在大学拥有学位论文的部分使用权，即：1）已获学位的研究生必须按学校规定提交印刷版和电子版学位论文，可以将学位论文的全部或部分内容编入有关数据库供检索；2）为教学、科研和学术交流目的，学校可以将公开的学位论文或解密后的学位论文作为资料在图书馆、资料室等场所或在有关网络上供阅读、浏览。

本人授权万方数据电子出版社将本学位论文收录到《中国学位论文全文数据库》，并通过网络向社会公众提供信息服务。同意按相关规定享受相关权益。

（保密的学位论文在解密后适用本授权书）

学位论文作者签名： 导师签名：

签字日期： 年 月 日 签字日期： 年 月 日

附件3：论文摘要格式

高中语文古诗词鉴赏教学研究

□□□□□□专业

研究生 □□□□ 指导教师 □□□□

**摘要** 古诗词是中华民族文化中的精华。古诗词鉴赏教学可以激发学生的情感，提高学生的审美情趣、陶冶学生的情操，使学生受到传统文化、人文精神、道德修养、文化品味的熏陶和感染，这对于继承民族文化传统，弘扬民族精神、实现语文的美育功能都具有非常重要的意义。

本文分四章来论述高中语文古诗词鉴赏教学，……

本文将从知人论世策略；比较策略；整体感知策略；张扬主体、多元解读策略；营造情景策略五个方面对高中语文古诗词鉴赏教学作探讨。

关键词：高中语文 古典诗词 鉴赏教学

附件4：目次页格式

**目 次**

摘要（小4号宋体） I

ABSTRACT II

插图和附表清单 Ⅲ

目次

1 （第1章）引言（绪论） 1

1.1（第1章第1节）题名 3

2 （第2章）题名 5

2.1（第2章第1节）题名 7

2.2（第2章第2节）题名 10

…………

5 （第5章）结论 71

参考文献 93

附录 96

致谢 98

在校期间的科研成果 102

**1** 绪论

**1.1 研究背景及意义**

根据NIST定义，云计算是可以使用户按照使用量付费并且获得高效的、快捷的网络服务的新型资源共享模式，其主要目的是在提高网络、服务器、硬盘存储、软件等共享资源利用率的同时，使云租户不再关注硬件资源的管理和维护，云租户只需要在硬件资源上投入很少的管理维护工作就可以得到很高的资源回报。云计算的高效的资源处理能力也带动了大数据、人工智能等相关领域的发展。目前在云计算提高商，比如国外的Intel，IBM，微软，以及国内的腾讯，阿里巴巴都拥有非常成熟的云计算技术和应用服务提供技术。云计算的快速发展同时也给云计算带了除传统信息安全、网络安全之外的安全问题，其中，如何向云租户证明云计算底层平台的安全性、虚拟机的安全性是一个非常重要的问题。而可信计算是保障信息系统安全最为重要的技术手段之一，它通过提供数据保护、身份认证、远程证明以及完整性度量等特性提高包括底层物理资源、应用软件等在内的计算平台的可信性和可靠性。因此，将可信计算技术应用在提高云计算环境的安全性是工业界和产业界必须重视的地方。

可信平台模块TPM(Trusted Platform Module, TPM)是可信计算的核心和关键技术，是可信计算机系统的信任根，通过构建从平台底层硬件到平台上层应用程序的信任链，并结合可信远程证明向平台外部实体提供可信证明。TCG（Trusted Computer Group, TCG）定义的可信计算平台的核心功能：度量、存储和报告等均依赖于TPM；可信计算平台的3个信任根：可信测量根RTM(Root of Trust for Measurement, RTM )、可信存储根RTS(Root of Trust for Storage, RTS )和可信报告根RTR(Root of Trust for Report, RTR)等均与TPM有直接的关系，其中可信测量根RTM由CRTM(Core Root of Trust for Measurement, CRTM)和TPM中的一组PCR存储器组成，可信存储根RTS由TPM和存储根密钥SRK组成，可信报告根RTR由TPM和EK组成。TCG的TPM工作组负责TPM规范的起草、修订和发布，其规范已从TPM1.0、TPM1.2发展到现在的TPM2.0。TPM2.0规范于2014年发布，在密码算法支持、密钥、授权、签名、虚拟化等方面均有些新的特点，尤其在虚拟化方面的更新推动了可信计算技术与云计算技术的结合，共同保障云计算安全，向云租户提供一个安全可靠的云服务。利用可信计算技术构建可信虚拟平台（TVP）是研究的重要方向。TVP可以利用可信计算技术对云计算平台启动过程中，不仅可以对云计算物理资源组件进行信任链可信度量，比如BIOS、操作系统内核OS、虚拟机管理器等，也可以对虚拟机启动的组件进行可信度量，比如虚拟机镜像系统、虚拟机文件、虚拟机虚拟化过程所需的OS、虚拟BIOS等。TVP的出现能够帮助云计算提供商向云租户更好的提供安全可靠的云服务。

根据中国信息通信研究院2017年7月发布的《云计算关键行业应用报告》，2016 年以 IaaS、PaaS 和 SaaS 为代表的典型云服务市场规模达到 654.8 亿美元，增速 25.4%，预计 2020年将达到 1435.3 亿美元，年复合增长率达 21.7%。我国云计算市场总体保持高速增长趋势。2016 年我国云计算整体市场规模达 514.9 亿元，整体增速 35.9%，高于全球平均水平。工业和信息化部发布的《云计算发展三年行动计划（2017－2019年）》提到我国云计算的发展目标“到2019年，我国云计算产业规模达到4300亿元”，该行动计划为我国云计算安全技术创新和产业发展指明了方向，提供了政策保障和法律依托。并且根据McAfee发布的“2017年全球云计算安全报告”显示，在过去15个月，80%的IT预算用于云应用和解决方案。云计算已经成为通过部署应用来销售服务的企业的首要事项。很多企业正在快速向云计算转移，以跟上合作伙伴和供应商的网络。无论顾客选择任何购买渠道和方式，都能提供卓越的客户体验，这也推动了云计算的部署。但是，只有23%的企业完全信任公共云可以保护他们的数据安全。由此可见目前的大部分云租户是不能完全信任云提供商提供的云计算服务，主要是由于云租户在使用云提供商提供的虚拟机时，并不能确认云计算平台上的物理主机是与云提供商按照各自操作系统官方文件进行启动的，以及租户请求的虚拟机是按照预期的配置和要求进行启动的。因为云计算环境下的虚拟机存在着包括传统信息系统安全以及新型网络安全等威胁，比如：虚拟机恶意代码攻击、虚拟机逃逸等，这些都会导致虚拟机在重新启动时的组件被篡改，在云租户对虚拟机进行重新启动时，可能无法判断虚拟机遭受操作系统、数据是否被篡改。而可信虚拟平台的构建可以利用TPM中的可信度量、可信报告等技术向用户发送关于云计算平台的可信度量结果，并且证明自身的安全性。

信任链技术是可信计算的关键技术，针对可信计算技术与云计算技术结合的可信虚拟平台的信任链构建更是十分有必要的。利用虚拟可信平台模块（vTPM）在云计算环境中构建安全可靠的可信虚拟平台，并且利用可信计算中的关键技术——信任链技术对整个云计算平台进行信任链构建，建立从云计算平台底层物理服务器到提供服务的可信虚拟机的信任链，可以给目前的云计算安全问题提供一个新的解决思路，为构建云服务提供更好的安全保障。

为此，本文对可信虚拟平台及其至上的信任链进行研究，并利用已有的形式化分析方法安全逻辑方法和改进的无干扰理论分别对信任链进行形式化分析。首先提出了带有可信衔接点的可信虚拟平台TVP-QT，该可信衔接点由三部分组件构成，虚拟机构建模块，虚拟可信模块构建模块，VM和vTPM绑定模块，该可信衔接点充分连接了云计算平台和虚拟机，保障信任链构建的合理性和完整性。其次，本文提出了一种具有瀑布特征的信任链模型—TVP-QT，该模型以硬件TPM为起点，在底层虚拟化平台和顶层用户虚拟机信任链之间加入可信衔接点。当信任链从底层虚拟化平台传递到可信衔接点时，由可信衔接点负责对用户虚拟机的vTPM进行度量，之后将控制权交给vTPM，由vTPM负责对用户虚拟机启动的组件及应用进行度量。该模型中可信衔接点具有承上启下的瀑布特征，能满足虚拟化环境的层次性和动态性特征，保证了整个可信虚拟平台的可信性。然后，本文利用目前的安全逻辑形式方法和改进后的无干扰理论方法——无干扰理论+方法对TVP-QT进行形式化分析，不仅在实验上证明了TVP-QT的合理性和可靠性，也在理论上证明了TVP-QT的合理性。

**1.2 国内外研究现状**

可信计算技术与虚拟化技术的结合的TVP一直以来都受到国内外学者的广泛关注。目前已经有涌现出很多优秀的研究成果。Intel的StefanBerger等人最先提出vTPM的概念，随后产生了很多关于TVP及其信任链构建的研究成功，其中，开源的VMM——Xen是最早支持vTPM的，已经有许多学术成果将Xen作为实验平台；KVM对支持vTPM主要是依赖QEMU；2015年，VMware宣布在vSphere 6.0中支持vTPM；Oracle的Virtual Box也通过IBM的PCIXCC支持vTPM。这些知名公司的技术促进了可信虚拟平台的发展。为更好的对国内外研究现状进行阐释，本文将从可信虚拟平台、可信虚拟平台信任链、针对信任链的形式化分析方法三部分进行描述。

1.2.1 **可信虚拟平台**

对于TVP的研究，早在TVP概念出现之前，就出现了利用可信计算技术解决虚拟系统平台安全的方案，为TVP的发展提供了一些理论和构建基础，这些平台包括Terra [17]，Perseus [18]等，这些平台的主要思想是把底层计算平台分为两部分，可信区域和不可信区域，其中可信区域上运行着高安全性需求的虚拟机。

TVP 的概念首先由 Stefan Berger [12] 等人提出，随后文献[13-16]等学者针对如何构建具体应用场景的 TVP 功能应用以及抽象和统一的 TVP 概念取得了很多较好的研究成果，并且达成了一些基本的共识。目前，研究此方面的学者绝大多数都认为，在物理上，TVP作为一个可以支持虚拟化技术的可信主机，并且与一般的可信计算平台的主要区别有两方面，一是拥有在物理硬件可信平台模块（Trusted Platform Module, TPM）构建起来的虚拟可信信任根；而是可以并发的为在可信虚拟平台之上的多个用户虚拟机（Virtual Machine, VM）提供可信虚拟信任环境。这种TVP的运行架构如图1所示。从功能上看，TVP架构主要分为4个层次。第一层为硬件信任根TVP，作为整个架构的最底层，是整个平台信任的物理保证。第二层主要包括虚拟机监视器（Virtual Machine Monitor, VMM），及构建与VMM之上的管理域（主要是其内核及相关域管理工具），它们通常被认为是 TVP 的可信计算基（Trusted Computing Base, TCB）。第三层是虚拟信任根（Virtual Root of Trust, vRT），由于实现方案不同（如图1中a、b所示），其加载过程可能是传统信任链的一部分，或直接利用动态加载机制如动态度量信任根（Dynamic Root of Trusted Measurement, DRTM）机制启动，这使得它或者成为 TCB 的一部分，或者作为应用进程单独存在。最上层是用户虚拟机，是与用户应用密切相关的部分。



图1TVP基本运行架构

根据文献[12]的vRT等概念，HP、IBM等研究机构分别提出并构建了相应的TVP[13][14]，其TVP架构可根据不同应用需求建立用户可定制的TVP，在很大程度上推动了TVP的发展。随后，Krautheim[15]、王丽娜[16]等学者基于云计算环境建立了TVP，使其可以保护云计算环境下的虚拟机运行，以及保护虚拟机运行时上层服务软件的完整性、安全性。之后，常德显[19]等根据TVP的功能层次给出了包括虚拟机和虚拟可信根的TVP定义，并细分为VMM、Dom0、TPM、vRT等组件。Zhang Lei[20]等提出一种具有可信域层次的TVP，通过可信云平台和可信虚拟机进行分离的TVP构建机制，并实现了对可信云平台以及可信虚拟机的安全保障。文献[21-26]也建立了类似以上的可信虚拟平台。可以说TVP在保证云计算环境安全、构建可信云平台上起到了重要的作用。总结起来，目前针对TVP模型的研究尽管取得了很多成果并达成了基本共识，即TVP模型都包含基本组件vRT、vTPM等，但绝大多数已有的研究成果把TVP的VMM和管理域都作为TCB，一起作为虚拟机的vRT，这显然过粗且逻辑上不完全合理的，因为管理域包含OS及大量的应用程序，显然不能采用链式度量所有的应用程序并存储其PCR值。

上述的TVP基本运行架构以及信任链传递模型存在过粗且逻辑上不完全合理的问题，与具体云环境中虚拟化平台也不完全相符合。如图1所示，为了便于叙述，本文将图1中从TPM到第三层的信任链称为可信虚拟平台信任链，将第四层的信任链称为虚拟机信任链。具体问题表现在：

（1）现有的TVP模型把整个第三层都作为TVP的TCB并作为虚拟机的vRT，显然是不精细的且逻辑上也不完全合理的。第三层包括VMM以及DOM管理域，信任链为CRTM→BIOS→BootLoader→VMM→DOM OS→Apps，DOM管理域包含OS及大量的应用程序，显然不能采用链式度量所有的应用程序并存储其PCR（Platform Configuration Module）值。

（2）虚拟平台信任链与虚拟机信任链是两条不同的信任链，即在整个TVP以及客户虚拟机启动过程中存在两条完全分隔的信任链，一条是可信虚拟平台在启动时的信任链，另一条是客户虚拟机在启动时的信任链，这两条信任链在度量层次和度量时间上均是分离的。如何向虚拟机用户展示一条从TPM到虚拟机应用的完整信任链呢？显然，这两条信任链存在如何衔接的问题。

1.2.2 **可信虚拟平台信任链模型**

对于TVP信任链模型的研究，主要包括三个方面。其一是通过对TCG链式信任链模型的扩展，实现TVP下可信度量以及信任传递。Scarlata[27]等提出在构建TVP时，通过可信测量构建从CRTM可信根到每个客户虚拟机的信任链，就可以证明每个客户虚拟机是可信的，显然这种信任链模型是不完善的，无法适应比较复杂的TVP环境。John[28]对信任链扩展上提出了“Transitive Trust Chain”信任链模型，并且简要的指出了信任链传递过程为TPM →VMM→TVEM manager →TVEM→VM OS(应用程序，但是此种信任链模型没有详细的描述特权域操作系统以及虚拟机操作系统的可信度量。Shen[29]等根据TCG动态度量方法提出了一种基于Xen的可信虚拟机在 DRTM下的信任链构建，其具体的构建过程为：CPU→可信代码→Xen VMM→Dom0（→vTPM Manager→Domain Builder）→Guest 0S →Guest Application，此种信任链模型也存在John[28]中的问题。等等。其二是通过研究可信云平台和可信虚拟机两部分的信任链，构建TVP下的信任链模型。常德显[19]等提出TVP信任链包括按照TVP的功能层次从硬件TPM层→TCB层→vRT层→用户虚拟机层的信任链模型，此信任链模型对vRT及层次间的连接定义比较模糊。Zhang Lei[20]等提出一种基于无干扰的可信域层次信任链模型，并且指出分别度量物理主机和VM的方式，即首先度量从物理的TPM到物理主机的应用程序，然后度量VM的vTPM和应用程序，显然此种信任链模型无法有效的对TVP下构建完整的链式信任链模型，不能向用户虚拟机呈现一条完整的信任链模型，文献[22][23]也存在此类问题。其三是树形或者星形的信任链模型。其三是树形或者星形的信任链模型。一部分学者认为TCG的链式信任链可信度量方式在虚拟化环境下是难以有效构建的。朱智强[30]提出了一种安全可扩展的星型信任度量结构，在信任度量时只需要信任根（RT）对管理域节点进行度量即可，但是此信任链模型的关键节点RT需要对所有的管理节点进行度量，RT的负担重，无法高效的完成TVP下的可信度量以及信任传递。曲文涛等[31]提出了一种解决RT负担的改进方案，带链式结构的星型信任链模型，设计了MDn节点分担了RT的部分度量负担，但是此种信任链模型也存在负担重的MDn节点。总结起来，目前针对TVP的信任链模型的共同问题是信任链模型过粗且逻辑上不合理的问题，与具体云环境中虚拟化平台也不完全相符合，且目前研究内容中的可信虚拟平台信任链与虚拟机信任链是两条不同的信任链，这两条信任链在度量层次和度量时间上均是分离的，不能向虚拟机用户展示一条从TPM到虚拟机应用的完整信任链。

1.2.3 **无干扰理论**

针对于确保平台信任可验证的信任链形式化建模与分析的方法，目前的研究大部分是基于传统的可信计算平台。陈书义[21]等人利用一阶逻辑对可信计算平台启动过程进行建模以分析其信任传递过程，并提出长度受限的信任链模型。张兴[22]等人基于无干扰模型对信任链进行了建模分析，从系统信息流控制角度验证满足传递无干扰安全策略的信息流才能构建有效的信任链。上述方法主要针对普通可信计算平台，并不能直接适用于云计算环境下信任链形式化分析。虽然Zhang[30]等人利用无干扰理论对可信云计算环境信任链进行了形式化分析和验证，但是此种信任链分析方法是建立在不连续的可信云计算信任链模型上，不能够对可信云计算环境进行正确的形式化验证。常德显[29]等人基于扩展安全系统逻辑的分析方法也存在Zhang中的问题。

针对于无干扰理论的研究，目前大部分的研究是基于信息流的无干扰模型从动作和运行结果的角度建立系统安全策略模型。张兴、赵佳[33,34]等在Rushby[35]的无干扰理论的基础上将系统安全域集实体化为进程集，给出了进程运行的可信条件，推导出系统运行可信定理，保证了终端的安全，但是其模型中的没有针对动作的详细定义，不适合验证可信云环境信任链。刘鹏威[36]等提出了基于非传递的无干扰理念的二元多级安全模型，在Rushby无干扰理论的基础上重新定义了清除函数，将传递的元干扰理论过渡到非传递的无干扰理论，并依据BLP和Biba模型保护了信息的机密性和完整性，然而同样存在赵佳中的问题。陈菊[37]等从进程数据和代码完整性检测出发，利用无干扰理论保证进程之间的操作合法，试图在不安全的操作系统中建立安全的应用支撑。徐甫[38]等扩展了非传递无干扰理论，并试图通过重新定义静态干扰和动态干扰，使其支持进程自身代码的修改，但是其静态干扰和动态干扰的定义过于抽象，难以和实际的终端系统相对应，因此并不能在实际上完成其所描述的支持自身代码修改的功能。秦晰[39]等提出了一种容忍非信任组件的可信终端模型，该模型利用可信组件对非信任组件的输出进行封装，保证了非信任组件在终端上的存在不会造成严重的安全威胁，实现了域间隔离和无干扰，保证了结果的可预测性和可控性，但是并没有针对安全域进行详细的描述。上述对无干扰理论的研究，均没有考虑到云计算运行中时的安全域、动作所属主体以及动作对安全域和系统状态的影响进行详细的说明，比如针对于安全域，在运行中各个安全域可能是独立也可能是通过组件绑定在一起。

**1.3 本文主要工作**

1. 具有瀑布特征的可信虚拟平台

由本文对可信虚拟平台的研究综述来看，目前的可信虚拟平台存在一些需要改进的地方。本文针对现有的可信虚拟平台，在可信虚拟平台和用户虚拟机之间加入可信衔接点，设计一种在逻辑上合理，vRT精细的可信虚拟平台。此可信虚拟平台的主要架构如下图所示：

在原有的TVP中的第二层和第三层加入可信衔接点。可信连接点位于Dom0，是Dom0的一组应用程序，包括VM的创建组件VM Builder、 vTPM实例的创建模块vTPM Builder以及VM-vTPM映射组件VM-vTPM Binding，且作为vRT的一部分，在信任链上按照VM BuildervTPM Builder VM-vTPM Binding的顺序依次进行度量。可信衔接点可对TVP-QT的第一、第二层与第四、第五层进行有效衔接，保证TVP-QT信任链构建的连贯性，起到承上启下的作用，具有瀑布特征。所谓“瀑布特征”，借鉴软件开发过程中的“瀑布模型”，软件开发中的“瀑布模型”，其特点是每一个开发流程必须从上一项开发流程接收输入，并且为下一个流程产生输出；如果重点任何一个环节没有输入或者产生输出，则严重影响开发的流程，甚至终端。本文提出的具有可信衔接点的TVP-QT模型在任何一个环节都会与前后组件进行衔接，如果中间因任何不安全问题出现任何错误或者中断，则TVP-QT及其信任链构建都会终止。

其中VM Builder表示与创建用户虚拟机相关的配置文件以及组件等，如创建vm组件以及vm配置文件等；vTPM Builder表示与创建和管理vTPM实例的相关组件，并负责提供给vm运行时的vTPM标识以及端口的提供。而VM-vTPM Binding则表示对vm和vTPM实例间的绑定关系，在TVP-QT涉及到的vTPM架构中，由vTPM Builder提供给vTPM实例的标识，每个vm必须与独一的vTPM实例绑定。

从此架构图可以看出，可信衔接点的存在可以有效的对用户虚拟机和可信虚拟平台进行衔接，并且具有承上启下的作用，使得整个可信虚拟平台在逻辑上都非常合理，并且加强了用户虚拟机和可信虚拟平台之间的联系。

1. 具有瀑布特征的信任链模型

由本课题针对可信虚拟平台之上的信任链模型，对本课题设计的带有可信衔接点的可信虚拟平台上进行构建，并对可信衔接点部分的组件进行细分，使得整个可信虚拟平台在启动过程中能够对虚拟平台TCB层度量之后能够有效的把控制权交给用户虚拟机的vRT，实现向用户展示一条完整的信任链模型的目标。

本课题构建的信任链模型大体架构如图：

具体的信任链度量过程描述如下：

针对可信虚拟平台构建的信任链模型总体过程如下：

第一层硬件TPM（CRTM）-->第二层TCB（BIOSOSLoaderVMMDom0 Kernel）-->第三层可信衔接点（VM BuildervTPM BuilderVM-vTPM Binding）-->第四层vTPM（vTPM实例）-->第五层可信虚拟机（VBIOSVOSLoaderVMOSAPP）;并利用通过TVP平台对信任链进行实例系统的分析和构建。本课题拟基于Xen VMM对信任链进行构建和实现。

1. 基于扩展无干扰理论的可信虚拟平台信任链分析方法

目前大部分的研究是基于信息流的无干扰模型从动作和运行结果的角度建立系统安全策略模型，本课题按照云计算环境运行特征，拟对原有无干扰理论中的安全域、动作等定义进行扩充，并将动作主体和动作对安全域以及系统状态的影响等扩展到无干扰理论中；最后应用此扩展的无干扰理论来分析可信云环境信任链传递模型，用形式化的方法证明当符合非传递无干扰安全策略时，云环境安全域之间的信息流受到安全策略限制，隔离了域之间的干扰，满足此条件时用完整性度量方法所建立的云环境信任链才是可信的、有效的。

图4 扩展无干扰理论

**1.4 论文组织结构**

本文共分为六章，每部分的安排如下：

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

**2** 相关技术

**2.1 虚拟化技术**

2.1.1 **虚拟化技术分类**

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

2.1.2 **虚拟机与虚拟机监视器**

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

2.1.3 **Xen与KVM**

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

2.1.4 **本章小结**

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

**3** 具有瀑布特征的可信虚拟平台TVP-QT

**3.1 TVP-QT系统结构**

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

**3.2 TVP-QT组件定义**

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

**3.3 TVP-QT工作程序**

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

**3.3 基于Xen的TVP-QT实现**

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

**3.4 基于Xen的TVP-QT实验**

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

3.4.1 **实验环境**

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

3.4.2 **功能实验**

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

（1）

3.4.3 **性能实验**

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

**3.5 本章小结**

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

**4** 具有瀑布特征的TVP-QT信任链模型

**4.1 TVP-QT信任链分析**

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

4.1.1 **Linux启动分析**

×××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

4.1.2 **Linux启动可信度量因素及内容分析**

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

4.1.3 **TVP-QT信任链分析**

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

**4.2 TVP-QT信任链设计**

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

**4.3 基于Xen的TVP-QT信任链构建**

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

**4.4 基于Xen的TVP-QT信任链实验**

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

4.4.1 **实验环境**

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

4.4.2 **功能实验**

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

4.4.3 **性能实验**

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

**4.5 本章小结**

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

**5** TVP-QT信任链形式化分析

**5.1 基于扩展安全逻辑(LS2)的TVP-QT分析**

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

5.1.1 **扩展LS2概述**

×××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

5.1.2 **主机m信任链的本地验证及远程证明**

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

5.1.2.1 本地程序执行

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

5.1.2.2 本地可信属性描述

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

5.1.2.3 信任链远程验证

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

1. 远程验证程序执行

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

1. 任链属性的远程验证

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

5.1.3 **可信衔接点TJP的本地验证及远程证明**

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

5.1.3.1 本地程序执行

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

5.1.3.2 本地可信属性描述

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

5.1.3.3 信任链远程验证

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

1. 远程验证程序执行

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

1. 任链属性的远程验证

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

5.1.3 **虚拟机vm的本地验证及远程证明**

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

5.1.3.1 本地程序执行

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

5.1.3.2 本地可信属性描述

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

5.1.3.3 信任链远程验证

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

1. 远程验证程序执行

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

1. 任链属性的远程验证

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

**5.2 基于扩展无干扰理论的TVP-QT分析**

5.2.1 **扩展无干扰理论基本假定及定义**

×××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

5.2.2 **主机m信任链的本地验证及远程证明**

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

5.2.2 **TVP-QT信任链传递形式化描述**

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

5.2.3 **扩展无干扰信任传递判定定理**

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

5.2.4 **基于扩展无干扰的TVP-QT验证**

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

**5.3 本章小结**

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

**6** 总结与展望

**6.1 工作总结**

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

**6.2 研究展望**

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

参考文献

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

致谢

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××

在校期间的科研成果

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××