可信计算研究的初步探疑*)

周明辉　梅　宏
(北京大学信息科学技术学院软件研究所　北京 100871)

摘　要　当计算发展到一定成熟阶段，可信需求显得迫切而关键。本文分析了当前学术界和工业界在可信计算方面的认知和作为，并以此为基础研究了可信计算的内涵，对在 Internet 环境下提供软件可信性保障所需解决的问题进行了探讨，为我们下一步将要开展的工作奠定基础。

关键词　可信计算，可靠，可用，安全

Research and Development of Trusted Computing

ZHOU Ming-Hui　MEI Hong
(Department of Computer Science and Technology, Peking University, Beijing 100871)

Abstract　Trustworthiness is key concern that has been addressed only after a technology is developed and deployed. This paper analyzes the recognitions and research actions of academe and corporations currently on trusted computing, based on which investigates the connotation about it, and discusses the problems needed to solve in order to provide trustworthiness for applications on Internet platform.

Keywords　Trusted computing, Reliability, Availability, Security

1　问题背景

Internet 及支撑它的不同信息技术网络构成了人类有史以来最为复杂的结构[1]。网络为人们带来了一个可以掌握在指尖的知识世界，在其逐步的迅速的发展过程中越来越深刻地影响着人们的日常生活。社会各个领域的应用越来越多地基于 Internet 实现和开展，涉及金融、电信、宇航甚至军事，这些应用领域向 Internet 拓展的同时也对基于 Internet 的计算的可信性提出了很高的要求。

这样，一方面，应用领域的拓展对系统的安全和可靠提出了迫切的需求；而另一方面，应用系统因为 Internet 天生的开放和动态特性导致在安全和可靠方面的欠缺。如何切合应用需要，在开放动态的 Internet 环境下，实现可信的信息资源共享和协作，即如何为基于 Internet 的应用系统提供可信保障，成为当前工业界和学术界密切关注和积极从事的重要课题。

2　可信计算的引入

可信计算是新近发展的一门学科，着力于解决计算世界当前所面临的普遍的安全威胁和不可信危机。在面向 Internet 的计算的初步发展阶段，可信技术并不是业界研究的主旨，只有现在——当计算已经发展到了一个比较成熟的阶段，可信需求才显得迫切而关键。美国国家科学基金会在文[2]中就此事有所论述。

可信计算是一个广泛的概念，它是对若干传统计算技术的综合。这些传统计算技术虽然研究历史悠久，并且在各自的领域也有卓有成效的研究成果，但基本上彼此独立、分而治之，因此对于整个系统来说很难有一致的执行效果。例如安全和隐私就是一对矛盾体，其中对安全的强调可能会破坏隐私，反之亦然。而今天我们不可能撇开其中任何一项而只考虑另外一项。如何对它们进行权衡和折衷从而达到系统的一致状态，就是可信计算需要研究的问题。当然，首先必须给出可信计算的内涵：它涵盖哪些计算技术？只有定义清楚可信计算所应考虑的可信属性，然后对这些可信属性分别加以透彻的研究和分析，最后才能进行评估和权衡。而所有这些都只是可信计算研究的基础内容。

不同的研究机构或公司对可信计算所涵盖的技术有不同的定义，下面分别对当前学术界和企业界在可信计算方面的认知和作为作出简要分析。

3　研究现状

3.1　美国国家科学基金会

美国国家科学基金会(National Science Foundation，简称为 NSF)有一个专门的可信计算计划(Trusted Computing Project，NSF01-160)，该计划试图为管理网络世界的安全和隐私建立合理的科学和技术基础[2]。他们认为，一个可信的信息系统所应考虑的方面包括[3]：security，privacy，safety，reliability。

该项研究以构建安全和可靠系统为目的，这对于高度互连的信息技术使能的社会来说是非常必要的。该计划支持在安全和可靠信息系统的所有方面有创新性的研究。所涉及的研究领域具体包括：

· 构件技术：研究构件的规范和设计、开发、测试及验证方法，为满足指定的可信性属性而提供可以计量的保证。

· 组装方法：研究按照指定的可信性属性将构件组装成

*)本课题得到国家"973"重点基础研究发展规划项目(2002CB312003)和"863"高技术研究发展计划项目(2001AA113060)资助。**周明辉**　博士后，主要研究方向为分布计算和软件构件技术。**梅　宏**　教授，博士生导师，主要研究方向为软件工程，软件复用和软件构件技术，分布对象技术等。

子系统或系统的方法。

·研究在系统适应(adapt)或者演化(evolve)时维护其可信性的方法。

·研究增强人们理解关键系统行为和控制的方法。

·评估对可信系统设计的不同属性(例如安全和性能)进行权衡的方法。

·研究对可信系统和构件的可信属性进行建模、分析和预测的技术。

2002年,可信计算计划收到了大约130件项目建议书,其中20项得到了批准。但从这些项目的研究内容可以看出,许多项目的侧重点还在于安全,着重研究入侵检测和访问控制等,对综合问题考虑得比较少。NSF也指出[2]:"虽然今年(2002年)立项的许多项目的确关注了未来新技术,但是大部分还是着眼于相对近期的目标。"

3.2 德国达姆施塔特大学

该大学的计算机科学和工程系的数据库和分布式系统组(Databases and Distributed Systems Group)中有一项"可信计划"(TRUSTED project)[4],致力于解决与可信分布式系统相关的许多问题。

他们认为,可信性是比任何其它特征都具综合性的一个性质,因为一个可信系统必须具有下述属性[4]:end to end security, availability, reliability, timeliness, consistency, predictability,scalability。

也就是说,可信性是涵盖上述所有特性的特性。并且,这些特性在下述环境下都应该能够得到满足:一个可信系统的构件可能是分布的、由开放网络连接的、无处不在的以及自然连接的。这些属性的综合是今天的软件开发的最终挑战。

"可信计划"是想以一种集成方式来解决可信分布式系统的许多相关问题的一种尝试。第一阶段的主要工作是将数据库和分布式系统组从事的若干项目与博士生课题相结合,研究电子商务的使能技术,主要包括端到端的安全,消息中间件,大型分布式异构系统的体系结构描述,以及基于Jini的无处不在的计算和自然网络。该计划的长期目标是希望能与企业合作建立一个可信计算中心,将其在可信计算领域的研究经验应用到后勤行业,大型网络基础设施,以及电信部门的关键应用等。

"可信计划"目前正在开发一个安全漏洞的数据库,并将对其应用数据挖掘技术以寻找出某些典型模式。它所从事的另外一个领域是中间件的安全服务,特别是CORBA安全服务的实现及其属性的分析等。

3.3 微软

微软作为全球软件业的业界领袖,在其产品风靡全世界的同时,也饱受非议之苦。而所有的责难基本上都来自于对其产品的安全漏洞的不满。为了解决广大用户的信任危机,当然也为了引领一个新的潮流,微软对外界宣称此后十年的战略重点是"可信赖的计算(Trustworthy Computing)",主要目标在于"让计算机能够更加安全与稳定,让人们对于计算机的信任就像使用电力与电话一样安心"[5]。

微软的CTO Craig Mundie在2001年的可信计算论坛(Trusted Computing PressPass)上声明,围绕计算的可信性问题,需要创建一种新的分类学,并随之提出"Trustworthy Computing"[1]以切合需要。它主要包含五个基本概念:availability,suitability,integrity,privacy,reputation。2002年12月,微软在其一份白皮书中[7],又将可信计算的四个目的总结为:security,privacy,reliability,business integrity。

纵观微软短短一年多间在可信计算领域的发展历程,可以看出它在概念上也是不断更新和变化的。这充分说明,对于一个新的领域,大家都在探索之中。

微软公司为实现"可信赖的计算"的目标,提出了短期、中期及长期的工作重点。从短期方向上来看,将着重产品的安全性、对于包括Windows、VisualStudio .NET、Office等产品进行完整的安全编码检测、与IBM及VeriSign共同宣布WS-Security项目——即从Web Service的基础环境下建立安全性、开发如软件更新服务及安全分析器等免费的安全管理工具、建立领先的安全反馈流程;中期的工作重点包括建立自我修复的管理机制与自动化服务策略;长期工作重点则着重于基础研究及面对策略方面的挑战。

此外,微软公司也将从设计(Design)、缺省(Default)及布署(Deployment)三大方面来搭建安全的计算机环境[6]:

设计:目前设计的工作重点包括Windows与Visual Studio .NET程序代码安全检测、对包括Windows、Internet Security and Accelaration(ISA)及.NET的9000多位程序开发人员进行安全培训、以及加强合作伙伴在Windows、ISA及.NET Framework安全性方面的审核。未来的重点则将强调Windows .NETServer与Windows XP的升级机制。

缺省:为了提供安全的计算机环境,微软公司在Microsoft Windows XP、Office XP及未来的Windows .NET Server上均设置了安全的缺省值,Office XP将自动屏蔽带有.exe的可执行文件和脚本文件;Windows XP则加入网络连结防火墙功能;即将推出的Windows .NET Server则将内置的Internet Information Service 6.0缺省为关闭,客户如果在有需求时,可以根据向导来启动。

布署:重点工作包括强调获得安全(Got Secure)与保持安全(Stay Secure)的战略性技术保障计划、提供免费资源、免费工具和动员微软公司全球范围的客户支持机构与内部开发团队来帮助客户以简单易行的方式营造并确保系统安全。

可以看到,微软对可信计算的解决方案仍然秉承其一贯风格,商业味道颇为浓厚。

3.4 TCPA(Trusted Computing Platform Alliance)

TCPA[11,12]是由Compaq,HP,IBM,Intel和Microsoft于1999年10月发起的一个组织。该组织的目标是致力于促成新一代具有安全、信任能力的硬件运算平台。

TCPA描述了一个计算平台的所有构件,从硬件到操作系统,再到用户应用,如何互相协作以保证平台可信。TCPA为可信计算平台制定了一个规范,规范主要定义内嵌在平台中的小部分硬件如何为软件进程奠定可信基础。也就是说,TCPA提供了硬件保全监控(就是无法由用户关闭)的芯片功能,让操作系统使用。目前的实际成品,初步会开发出一块叫做Fritz的芯片,放在主机板上。最后的目标,是放在CPU里,让你想拔都拔不掉。

图1所示为TCPA所定义的整个可信计算平台的模型。其中TPM是硬件部分,为平台实现了一组受保护的安全服务,访问TPM及其数据被严格控制,并且TPM的所有操作都要求认证。CRTM是在启动过程中运行的第一个软件,一般被设置在TPM中(但规范并不要求)。TSS是对TPM能力的扩增,这些服务是可信系统的一部分,但无需是安全的也并不受限于只能由平台所有者使用。

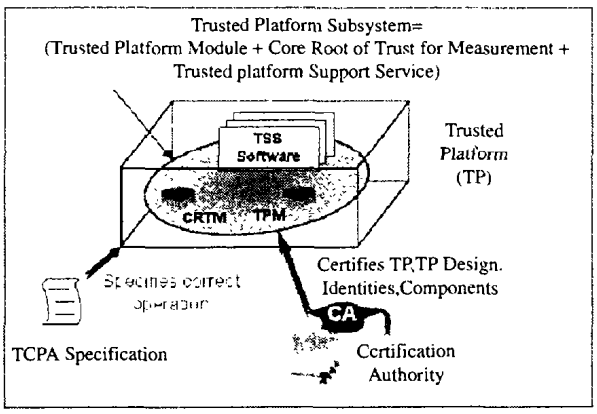

图 1　TCPA 的可信计算平台模型

TCPA 规范声称如果系统的所有层次都像所描述的那样实现，则整个平台就能构建一条可信的安全链。

但是，对于 TCPA 的这种实现所谓可信平台的解决方案，业界有许多反对意见。文[8]中 Ross Anderson 认为"所以称之为'所信任的计算机'，意思是可以有着伤害我的安全之虞的计算机"；文[9]中 Richard Stallman 认为"许多人认为他们的计算机应该服从他们自己，而不是别人。但是 TCPA 正计划使得你的计算机服从他们，而不是你。"

4　可以研究的问题

4.1　可信计算概念

事实上，可信计算是若干传统计算的综合，遵循传统计算概念的定义，我们可以将可信计算定义为：用以描述系统按照可信模式执行计算的处理过程。其中的关键概念首先是可信模式。什么是可信模式？以现今各研究机构和公司对可信模式的理解为基础，我们将其概括为可靠、可用和安全等。必须说明的是，随着我们研究工作的深入，对此可能会有扩展和修订。

· 可靠性，系统在一个完整的时间间隔之内正常服务的能力。可以描述为一个时间函数 $R(t)$，被定义为系统在某个时间段 $[t_0,t]$ 正确执行的概率（假设系统在 t_0 时刻正确执行）[13]。一般也可以用下述表达式描述：$MTBF/(1+MTTR)$，其中 $MTBF$ 为系统平均失效间隔，$MTTR$ 为系统平均恢复时间。系统无故障时间越长，可靠性越高。

· 可用性，系统在某个时刻可用的概率。也可以描述为一个时间函数 $A(t)$，被定义为系统在时刻 t 正常操作且功能可用的概率[13]。一般也可以用下述表达式描述：

$$MTBF/(MTBF+MTTR)$$

可用性主要依赖于系统迅速可用的能力，即其快速恢复的能力。

· 安全性，对于一个计算系统来说，安全意味着系统上的信息受到保护，不会被未经授权地泄露[15]。美国计算机安全专家 Bruce Schneier 这样认为[17]："安全不是产品；安全是一条链子，包括了硬件、软件、网络、使用者以及相互交互的复杂系统。在这条链子上，任何一个环节出现问题，安全将不存在……这条链子的强度由链子最弱一环的强度决定。"这是目前为止对安全表述最清晰的定义。

与人们业已形成的"安全"概念最为接近的就是信息的完整性和安全性。所谓完整性，就是说这个体系应该按照我们设想的方式，不犯任何错误地工作，要达到这种高完整性必须尽可能地清除各种各样的 Bug 或者是缺陷。而安全性，也即是说，对于用户的数据和系统，应该比较容易方便地"接触到"。这里指的当然是经过授权许可的用户，反过来，还需要防止一些人未经许可地获得数据，防止一些恶意的人对系统进行攻击[16]。

为构件或者软件系统保障上述特性，可以从许多方面着手（照 Bruce Schneier 说的那条链子，我们可以选择任何一个环节，或者全盘考虑？）。例如加强构件或系统本身的体系结构和代码设计，使得它们天生强壮；例如冗余机制，采用资源复制的方式提高系统可靠性和可用性；例如访问控制策略，通过限制访问者权限而增强系统安全性。但是，如何以一种统一的方式考虑软件可信性保障问题，是目前有待研究的课题，下文将对此进行分析和阐述。

4.2　待研究的问题

可信计算中每个关键概念的研究历史都比较久远，但基本上都是分而治之，因此很难有一致执行。例如，对安全的考虑可能会影响隐私，对可靠的考虑可能会破坏可用。为此，可信计算的重点在于如何综合及权衡可信属性各个方面，以达到系统的可信需求。

为研究 Internet 环境下的软件可信性保障技术，我们需要解决下述问题。

首先面临的问题是，可信计算的内涵是什么？因为可信计算是个比较新的概念，但是它涵盖的内容又是老的内容，如何对可信计算做出有效的定义这就是一个很值得深究的问题。可信计算涉及的计算领域非常广，但是它不应该仅仅是传统计算概念的简单累加。虽然不管是可靠性、完整性、安全性，还是隐私性等，对绝大多数人来说一点也不新鲜，但是对单个问题的处理，本身已经很困难很复杂，更何况需要综合在一起加以解决。而且，对于一个问题的研究，首先会将其细分后再分别进行研究，如果对每个层面都进行深入研究的话，这个问题将会变得更加深不可测，因为有时候要使各个方面的研究或者计算成果达成一致是非常困难的，这时候可能需要在思路上进行创新[16]。

其次，我们应该如何提供可信保障？事实上 Bruce Schneier 的"链条"思想（参见 4.1 节"安全"定义）可以应用到任何一个计算领域。对于安全是如此，对于可靠、可用等也是如此。一个系统可用，包括了硬件、操作系统、中间件、应用软件、网络、使用者等各个层次上的可用。我们可以选择链条中的哪一个环节，或者有基础加以全盘考虑？

假设我们要将对软件可信性的保障定位在某个环节，我们应该定位在软件本身或是软件的支撑平台？如果研究软构件个体的可信技术，则包括在构件的设计、开发、测试和验证方法上都要对可信属性有所考虑和评估，这就要求在程序设计和开发方法，从代码级上有所体现。另一方面，提高软件可信性也可以体现在支撑平台上。支撑平台，一般考虑中间件，在构件开发时可以提供一个保证其可信性属性的开发框架，以目前中间件提供的开发框架为基础，增添一些对可信属性进行评估和度量的工具（则涉及研究对可信系统和构件的可信属性进行建模、分析和预测的技术），或者允许应用系统开发者设置可信属性，开发框架为之体现出来，但这里的问题就体现在如何描述可信属性，以及可信属性如何在开发出来的应用系统中得以体现。

必须重点说明的是，有效地设计一个可信系统是非常富有挑战性的，它涉及大量的 QoS 决策内容，包括性能、可靠性、可用性、以及资源使用等。软件系统的体系结构设计就主

要取决于对系统的各项属性进行权衡的结果。一般地，这些设计决策都假设由设计者，而不是某些基于 QoS 的服务来决定。如果我们能够寻找出一种良好的权衡和评估方法，使得中间件框架能为应用提供这些决策帮助，将是一件非常有意义的事情。

此外，软件系统(或构件)在其运行过程中，如何在环境的变化下维护其可信属性?一般来说，支撑平台必须能够有机制和手段保证软件系统的可信度。例如，如果发生了网络攻击或者人为事故，使得软件系统某时刻某个服务不可用，支撑平台必须能够迅速作出反应(例如启动该服务的一个副本)以继续维持软件系统的可信属性。另外，如果软件系统发生了适应变化或者演化，支撑平台也应该能够维护它们的可信性。

结束语 本文以可信计算为研究焦点，描述了它在学术界和工业界的发展历程，并以此为基础探讨了在 Internet 环境下研究软件可信性保障技术所需要解决的问题，为下一步工作的展开奠定基础。

参 考 文 献

1 Mundie C. Remarks on Trusted Computing Forum 2001, www.microsoft.com/presspass/exec/craig, Nov. 6, 2001
2 NSF:CISE-Trusted Computing, www.nsf.gov, March 2003
3 NSF Program Announcement/Solicitation: Trusted Computing, www.nsf.gov/pubs/2001/nsf01160/nsf01160.html # TOC, Dec. 2001
4 Department of Computer Science of the Technical University Darmstadt. Trusted Systems, www.dvs1.informatik.tudarmstadt.de/DVS1/research/index.html
5 Gates B. Trustworthy Computing, www.microsoft.com, Jan. 2002
6 Microsoft White Paper, Trustworthy Computing, www.microsoft.com, Oct. 2002
7 Microsoft White Paper. Building a Secure Platform for Trustworthy Computing, www.microsoft.com, Dec. 2002
8 Aderson R. TCPA/Palladium Frequently Asked Questions, www.cl.cam.ac.uk/~rja14/tcpa-faq.html, July 2002
9 Stallman R. Can you trusted your computer, newsforge.com search.pl? topic=19, Oct. 2002
10 Workshop on Trusted Computing Paradigm'02
11 Gates M. Motivation for TCPA and Descriptions of Some Components, mgates.oxonet.com/tcpa/report.html, May 2002
12 Pearson S. Trusted Computing Platforms, the Next Security Solution, www.informit.com, Aug. 2002
13 Pradham D K. Fault-Tolerant Computer System Design. Prentice Hall PTR, June 1996
14 Wu Jie. Distributed System Design. CRC Press LLC, 1999
15 Introduction to security, ou800doc.caldera.com/SEC-admin/CTOC-IntroScur.html, Caldera International, Inc., 2001
16 高信度计算的未来, www.ciweekly.com/article/20021111/20021111222259-1.xml, 2002.11
17 Schneier B. Secrets & Lies: Digital Security in a Networked World, John Wiley & Sons, Aug. 2000

(上接第 4 页)

4 评述

UML 2.0 被采纳之后，大多数人认为这是一个更加平衡的语言，特别是在行为图的集成能力，对构件、实时和嵌入式系统、业务过程的描述能力方面有很大的增强[15]。但是在文[6,7,11,16]中提到的对 UML2.0 的担忧有些仍然存在，表现在：

1)没有给出 UML 一个简洁、精炼的核心。相反，通过 InfrastructureLibrary 定义了一个语言家族的核心。通过对其应用复用、特化、重定义等办法来定义诸如 UML 和 MOF 等其它元模型。而多数用户使用的 UML 本身仅仅体现在上层结构中。上层结构的规模仍然庞大繁杂，没有做到使用 UML 语言的 20%来详述 80%的常见软件问题[7]。

2)四层元模型的体系结构被弱化。UML 和 MOF 更多地站在了同一个层次上，都是 InfrastructureLibrary 的实例。对于已经熟悉了 OMG 四层元模型体系结构的用户，需要重新来理解新的、基于语言家族的体系结构了。

3)UML 2.0 由相对独立的四个规范组成，并且大量公司参与了制订过程，使其不可避免地出现了"第二系统病症"问题[17]。与 UML1.X 相比，UML2.0 的建模元素从 170 多个增加到了 360 多个，图从 9 种增加到了 13 种，规范文档从 700 多页增加到了总计 1000 多页。虽然，UML 2.0 试图通过包内相容点(compliance point)的定义使得用户和工具商可以递进地、有选择地实现语言特性，但是，毕竟掌握和学习 UML2.0 变成了一件很不轻松的事情。

4)工具商在初期会因 UML 2.0 过于庞大复杂的规模，不得不按照规范中的包内相容点来有选择地实现 UML。但是之后呢，出于市场竞争的需要，工具商可能又会发展到对整体规范的完全支持，就像 SQL 语言发展的老路一样。这样的结果，对用户从可用性、经济的角度来说造成的影响还有待于时间的检验。

5)新增的一阶扩展机制允许用户定义新的模型元素，甚至是语言。自由地定义各种不同于 UML 的新的建模语言究竟是拯救了 UML 还是重新开始一场语言之争，其真实效力如何还有待于实践的检验。

参 考 文 献

1 Booch G, Rumbaugh J, Jacobson I. The Unified Modeling Language User Guide. Addison Wesley Longman, Inc. 1999
2 Cook S. The Architecture of UML. OMG presentations, 2002
3 Salicki S, Jourdanl J, de Miguel M. Practical Experiences in the Application of MDA. UML 2002
4 Weigert T. UML 2.0 RFI Response Overview. Motorola, Dec. 1999
5 Kobryn C, Weigert T. OMG UML 2.0 RFPs ad/00-09-05. Object Management Group, Sep. 2000
6 Kobryn C. UML 2001: A Standardization Odyssey. Communications of the ACM, Oct. 1999,42(10)
7 Kobryn C. Will UML 2.0 Be Agile or Awkward? Communications of the ACM, Jan. 2002,45(1)
8 UML 2.0 Infrastructure, 3rd Revision. OMG document ad/03-03-01. Object Management Group, 2003
9 UML 2.0 Superstructure, 3rd Revision. OMG document ad/03-04-01. Object Management Group, 2003
10 Unified Modeling Language Specification, Version 1.5. OMG document formal/03-03-01, Object Management Group, 2003
11 Kobryn C. Designing a More Agile UML 2.0. Telelogic Research Center, 2002
12 Björkander M, Kobryn C. Architecting Systems with UML 2.0. IEEE SOFTWARE, July/August 2003
13 Kobryn C, Björkander M. UML 2.0 Roadmap: What Should Systems Engineers Expect? UML for Systems Engineering Workshop, Chicago, USA. Sep. 2002
14 Douglass B P. UML 2.0: Incremental Improvements for Scalability and Architecture. I-Logix Co. 2003
15 Vaughan J. Are you ready for some UML 2.0? Application Development Trends, Sep. 2002
16 邵维忠，杨芙清．面向对象系统设计．清华大学出版社，2003
17 Brooks F. The Mythical Man-Month, Anniversary Edition. Addison-Wesley, 1995