[19] 中华人民共和国国家知识产权局

[51] Int. Cl[7]
G06F 9/44
G06F 9/46 G06F 17/00

[12] 发明专利申请公开说明书

[21] 申请号 200410056423.1

[43] 公开日 2005 年 2 月 16 日

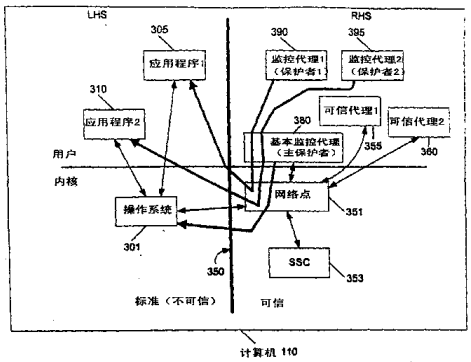
[11] 公开号 CN 1581073A

[22] 申请日 2004.8.6
[21] 申请号 200410056423.1
[30] 优先权
[32] 2003. 8. 7 [33] US [31] 10/638,199
[71] 申请人 微软公司
地址 美国华盛顿州
[72] 发明人 B·M·维尔曼 P·英格兰德
K·D·雷 K·卡普兰
V·库里恩 M·D·马尔
[74] 专利代理机构 上海专利商标事务所
代理人 谢喜堂
权利要求书 6 页 说明书 18 页 附图 4 页

[54] 发明名称 从可信环境到不可信环境的可信性投影

[57] 摘要

在具有运行在不可信环境中的实体和运行在可信环境中的实体的单个机器上，可信环境中的实体的可信性被投影到不可信环境中的实体。 例如，这可以应用到 Microsoft ®的下一代安全计算基础(NGSCB)，其中，常规操作系统(如，Windows 操作系统)主含安全操作系统(如，网络点)。



ISSN1008-4274

1．一种系统，其特征在于，它包括：

一不可信环境；

一可信环境；以及

至少一个基本监控代理，运行在所述可信环境中，监控所述不可信环境。

2．如权利要求1所述的系统，其特征在于，它还包括运行在所述可信环境中的多个监控代理，每一监控代理监控运行在所述不可信环境中的至少一个应用程序、扩充或组件。

3．如权利要求2所述的系统，其特征在于，每一监控代理与一应用程序关联，并且其中，每一监控代理监控其关联的应用程序的攻击和非一致性，由此将所述可信环境的可信性投影到所述不可信环境。

4．如权利要求3所述的系统，其特征在于，每一监控代理包括其所关联的应用程序的一部分。

5．如权利要求3所述的系统，其特征在于，每一监控代理具有可调节的检查级别，以将威胁指向所述关联的应用程序。

6．如权利要求3所述的系统，其特征在于，每一监控代理能够接收安全输入并将所述安全输入传输到所关联的应用程序。

7．如权利要求2所述的系统，其特征在于，所述监控代理的至少一个监控运行在所述可信环境中的应用程序代理。

8．如权利要求2所述的系统，其特征在于，它还包括运行在所述可信环境中的另一监控代理，其中，所述监控代理彼此间进行通信。

9．如权利要求1所述的系统，其特征在于，所述基本监控代理检测所述不可信环境中的不一致性。

10．如权利要求9所述的系统，其特征在于，它还包括检测运行在所述不可信环境中的应用程序中的不一致性的多个监控代理。

11．如权利要求10所述的系统，其特征在于，它还包括检测运行在所述可信环境中的应用程序中的不一致性的另外的监控代理。

12．如权利要求1所述的系统，其特征在于，所述基本监控代理的至少一个批准或否决不可信环境事件。

13. 如权利要求 12 所述的系统，其特征在于，所述至少一个基本监控代理包括用于接收输入的安全输入，所述基本监控代理基于所述接收的输入批准或否决。

14. 如权利要求 1 所述的系统，其特征在于，所述基本监控代理的至少一个在未经由安全输入接收到批准的情况下拒绝允许对所述不可信环境的变化。

15. 如权利要求 1 所述的系统，其特征在于，所述基本监控代理的至少一个拒绝允许对所述不可信环境的变化，除非所述变化由经批准方签署的包描述。

16. 如权利要求 1 所述的系统，其特征在于，它还包括运行在所述可信环境中的一监控代理，所述监控代理监控运行在所述不可信环境中的至少一个应用程序、扩充或组件，其中，所述监控代理使用密封存储来对驻留在所述不可信环境中的操作系统或应用程序保守机密。

17. 如权利要求 16 所述的系统，其特征在于，所述监控代理拒绝向所述操作系统或应用程序揭示所述机密，除非所述操作系统或应用程序具有与所述机密的所有者匹配的摘要。

18. 如权利要求 16 所述的系统，其特征在于，所述监控代理拒绝向所述操作系统或应用程序揭示所述机密，除非所述操作系统或应用程序在可读取所述机密的摘要列表上。

19. 如权利要求 16 所述的系统，其特征在于，所述监控代理使用预定的测试来确定是否合法实体正在请求所述机密。

20. 如权利要求 19 所述的系统，其特征在于，所述预定测试包括检查所述实体的栈并确保所述栈具有合法的栈内容。

21. 如权利要求 1 所述的系统，其特征在于，它还包括运行在所述可信环境中的一监控代理，所述监控代理监控运行在所述不可信环境中的至少一个应用程序、扩充或组件，其中，所述监控代理编辑所述不可信环境的状态以将其变为安全或以其它方式可接受。

22. 如权利要求 21 所述的系统，其特征在于，所述状态包括初始配置或错误报告选项。

23. 如权利要求 1 所述的系统，其特征在于，所述基本监控代理以零值代表不属于所述不可信环境中已知为好的配置或所述可信环境的物理存储器。

24. 如权利要求 1 所述的系统，其特征在于，所述不可信环境包括一基本输入输出系统（BIOS）、固件或加载器。

25. 如权利要求 1 所述的系统，其特征在于，它还包括用于在启动时运行所述基本监控代理的网络点。

26. 如权利要求 1 所述的系统，其特征在于，它还包括所述可信环境中的一计数器，所述计数器用于确定是否应当运行所述基本监控代理。

27. 如权利要求 26 所述的系统，其特征在于，所述计数器对不可信存储器编辑操作的数量进行计数。

28. 一种监控不可信环境的方法，其特征在于，它包括：

提供所述不可信环境；

在所述可信环境中提供至少一个基本监控代理；以及

监控所述不可信环境的攻击和不一致性以将所述可信环境的可信性投影到所述不可信环境。

29. 如权利要求 28 所述的方法，其特征在于，它还包括：

提供运行在所述可信环境中的多个监控代理；以及

监控运行在所述不可信环境中的至少一个应用程序、扩充或组件。

30. 如权利要求 29 所述的方法，其特征在于，它还包括：

将每一监控代理与一应用程序关联；以及

监控每一关联的应用程序的攻击和不一致性。

31. 如权利要求 29 所述的方法，其特征在于，它还包括把应用程序与所述监控代理之一相关联，并将所述应用程序的一部分传输到所述监控代理使所述部分驻留在所述可信环境中。

32. 如权利要求 29 所述的方法，其特征在于，它还包括将应用程序与所述监控代理相关联，并调节所述监控代理中的检查等级，以将威胁指向所述关联的应用程序。

33. 如权利要求 29 所述的方法，其特征在于，它还包括将应用程序与所述监控代理关联，并在所述监控代理上接收安全输入并将所述安全输入传输到所述应用程序。

34. 如权利要求 29 所述的方法，其特征在于，所述监控代理彼此进行通信。

35. 如权利要求 28 所述的方法，其特征在于，它还包括所述基本监控代理批准或否决不可信环境事件。

36. 如权利要求 35 所述的方法，其特征在于，它还包括所述基本监控代理从

安全输入接收输入。

37. 如权利要求 28 所述的方法，其特征在于，它还包括所述基本监控代理在未经由安全输入接收批准的情况下拒绝允许对所述不可信环境的变化。

38. 如权利要求 28 所述的方法，其特征在于，它还包括所述基本监控代理拒绝允许对所述不可信环境的变化，除非所述变化由经批准方签署的包描述。

39. 如权利要求 28 所述的方法，其特征在于，它还包括：

提供运行在所述可信环境中的多个监控代理；以及

所述监控代理之一使用密封存储来对驻留在所述不可信环境中的操作系统或应用程序保守机密。

40. 如权利要求 39 所述的方法，其特征在于，所述监控代理拒绝向所述操作系统或应用程序揭示所述机密，除非所述操作系统或应用程序具有与所述机密的所有者匹配的摘要。

41. 如权利要求 39 所述的方法，其特征在于，所述监控代理拒绝向所述操作系统或应用程序揭示所述机密，除非所述操作系统或应用程序在可读取所述机密的摘要列表上。

42. 如权利要求 39 所述的方法，其特征在于，它还包括使用预定测试来确定是否合法实体正在请求所述机密。

43. 如权利要求 42 所述的方法，其特征在于，所述预定测试包括检查所述实体的栈并确保所述栈具有合法的栈内容。

44. 如权利要求 28 所述的方法，其特征在于，它还包括：

提供运行在所述可信环境中的多个监控代理；以及

所述监控代理之一编辑所述不可信环境的状态以将其变为安全或以其它方式可接受。

45. 如权利要求 44 所述的方法，其特征在于，所述状态包括初始配置或错误报告选项。

46. 如权利要求 28 所述的方法，其特征在于，它还包括所述基本监控代理以零值代表不属于所述不可信环境的已知为好的配置或所述可信环境的物理存储器。

47. 如权利要求 28 所述的方法，其特征在于，所述不可信环境包括一基本输入/输出系统（BIOS）、固件或加载器。

48. 如权利要求 28 所述的方法，其特征在于，它还包括在启动时通过网络点

运行所述基本监控代理。

49. 如权利要求 28 所述的方法，其特征在于，它还包括确定响应于计数器是否应当运行所述基本监控代理。

50. 如权利要求 49 所述的方法，其特征在于，所述计数器对不可信存储器编辑操作的数量进行计数。

51. 如权利要求 28 所述的方法，其特征在于，它还包括提供检测运行在所述不可信环境中的应用程序中的不一致性的多个监控代理。

52. 如权利要求 51 所述的方法，其特征在于，它还包括提供检测运行在所述可信环境中的应用程序中的不一致性的另外的监控代理。

53. 一种系统，其特征在于，它包括：

一可信环境，具有操作系统、固件和基本输入/输出系统（BIOS）的至少一个；

一不可信环境；以及

运行在所述可信环境中并与运行在所述不可信环境中的操作系统、固件、BIOS 和应用程序之一关联的至少一个监控代理。

54. 如权利要求 53 所述的系统，其特征在于，所述至少一个监控代理包括多个监控代理，每一监控代理具有相关联的权力。

55. 如权利要求 53 所述的系统，其特征在于，所述可信环境在第一处理器体系结构上运行，并且所述不可信环境在第二处理器体系结构上运行，它还包括运行在所述第一处理器上的一基本监控代理。

56. 如权利要求 53 所述的系统，其特征在于，所述可信环境和不可信环境运行在同一处理器上，它还包括运行在所述可信环境中的一基本监控代理。

57. 如权利要求 53 所述的系统，其特征在于，所述可信环境运行在第一处理器上，所述不可信环境运行在第二处理器上，所述第一和第二处理器能够在可信模式或不可信模式中运行。

58. 如权利要求 53 所述的系统，其特征在于，它还包括驻留在所述可信环境中的一网络点和一基本监控代理，所述基本监控代理被绑定、连接或编译到所述网络点。

59. 如权利要求 53 所述的系统，其特征在于，它还包括驻留在所述可信环境中的一网络点和一基本监控代理，其中，所述基本监控代理是运行在所述网络点上的用户模式处理器。

60. 如权利要求 53 所述的系统，其特征在于，它还包括的一基本监控代理，该基本监控代理驻留在所述可信环境中，并根据且使用且在一构建环境中开发，该构建环境与所述不可信环境的操作系统相同或相关。

61. 如权利要求 53 所述的系统，其特征在于，它还包括驻留在所述可信环境中的一基本监控代理，所述基本监控代理是用于安全评估的可信计算基础的一部分。

62. 如权利要求 53 所述的系统，其特征在于，它还包括一基本监控代理，所述基本监控代理的第一部分驻留在所述可信环境中，并且所述基本监控代理的第二部分驻留在物理远程机器上，所述第一和第二部分由安全链路连接。

63. 一种系统，其特征在于，它包括：

一运行在可信环境中的代理；

一运行在所述可信环境中并提供到所述代理的投影的监控代理；以及

一运行在所述可信环境中监控所述监控代理的基本监控代理。

64. 如权利要求 63 所述的系统，其特征在于，所述代理是虚拟机器代理，并且所述监控代理是虚拟机器监控代理。

65. 如权利要求 64 所述的系统，其特征在于，所述基本监控代理是虚拟机器基本监控代理，并投影所述虚拟机器代理中的操作系统映象。

66. 如权利要求 64 所述的系统，其特征在于，它还包括与所述虚拟机器监控代理关联的应用程序，其中，所述基本监控代理是虚拟机器基本监控代理并提供到所述应用程序的投影。

从可信环境到不可信环境的可信性投影

技术领域

本发明一般涉及计算机安全领域，尤其涉及单个计算装置上多执行环境（如操作系统）的使用，并提供支持这一操作系统或环境的可信性的技术。

背景技术

第一台计算机仅能够每次运行单个程序。然而，在现代，期望计算机能够一次运行软件的若干不同片段。例如，典型的多任务操作系统能够在单个机器上一次运行若干个应用程序。鉴于这一情况以及共享、开放网络（即，因特网）的发展，安全和私密性变为计算机工业所面对的两个重要且困难的问题。由于个人计算机对家庭、工作和学校来说变得更主要，消费者和商业客户等越来越意识到私密性和安全问题。提高软件和硬件的能力来保护数字信息的完整性和计算机用户的私密性对软件开发者和硬件制造商来说已经成为一个关键性焦点。华盛顿州雷蒙德市的微软公司引入了下一代安全计算基础（NGSCB）个人计算机平台，它在操作系统中提供了安全和私密性。

在计算机 110 内的常规 NGSCB 中，如图 2 所示，“右侧”（RHS）安全系统与传统的“左侧”（LHS）系统和中央处理单元（CPU）一起工作。设计 RHS 以在保持操作系统的开放性的同时防止恶意软件。使用 NGSCB，应用程序在高度防软件篡改和干扰的保护的存储器空间中运行。通常，在计算机 110 内有一个芯片同时由 LHS 和 RHS 使用。LHS 和 RHS 是计算机 110 的逻辑而物理实施的分部或分区。

LHS 包括传统应用程序 205、210，如 Microsoft® Word®和 Microsoft® Excel®，以及常规操作系统 201，如 Microsoft® Windows®操作系统。尽管示出了两个应用程序，通常可以实现任意数量的应用程序。

RHS 包括可信代理 255、260，以及“网络点（nexus）”251。网络点是“高保证”操作系统，提供对于其行为的某一级别的保证，并可包括 RHS 上所有内核模式代码。例如，通过提供保证不向网络点之外的世界泄漏信息的屏蔽存储器

（curtained memory），并通过仅许可某些已鉴定应用程序在该网络点下执行并访问屏蔽存储器，可以使用网络点来使用不应泄露的机密信息（如，加密密钥等等）工作。网络点 251 不应当以允许在主操作系统 201 上发生的事件泄漏网络点 251 的行为的任一方式与主操作系统交互。网络点 251 可许可所有应用程序运行，或者机器所有者可配置一机器策略，其中网络点 251 仅许可某些代理运行。换言之，网络点 251 运行机器所有者告诉它运行的任一代理。机器所有者也可以告诉网络点不运行什么。

网络点 251 隔离可信代理 255、260，管理至和自可信代理 255、260 的通信，并使用加密地封装储存的数据（如储存在硬盘驱动器上）。更具体地，网络点 251 在可信空间中以内核模式执行，并向可信代理 255、260 提供基本服务，如用于与可信代理和其它应用程序的进行通信的进程机制的建立，以及特殊信任服务，如硬件/软件平台或执行环境的证明和机密的封装和拆封。证明是代码片段数字地签名或向数据片段证明并进一步向接收者确保数据是由不可锻（unforgeable）、使用密码标识的软件栈构造的能力。

可信代理是在可信空间中以用户模式运行的程序、程序的部分或服务。可信代理 255、260 调用网络点 251，用于安全相关服务和关键性一般服务，如存储器管理。可信代理能够使用密封的存储来储存机密，并使用网络点的证明服务来验证其本身。每一可信代理或实体控制其自己的可信域，并且它们不需要相互依赖。

RHS 还包括使用公钥基础结构（PKI）密钥对的安全支持组件（SSC）253 以及加密功能来提供安全状态。

NGSCB 提供诸如“证明”、“密封存储”以及“强进程隔绝”等特征。证明令其它计算机知道一个计算机的确是它所声称的计算机，并且正在运行它所声称正在运行的软件。由于 NGSCB 软件和硬件对用户和其它计算机、程序和服务是加密地可核实的，系统能够在加入其它计算机和进程或共享信息之前核实它们是可信的。由此，证明允许用户向外部请求者展现操作环境的所选择的特征。

密封存储允许用户加密信息，使该信息仅能由可信应用程序访问。这可包括的确是首先创建该信息的应用程序，或者被拥有该数据的应用程序信任的任何应用程序。因此，密封存储允许程序储存机密，该机密无法由不可信的程序，如病毒或特洛伊木马检索。

强进程隔绝通过开拓安全区域（RHS）提供了可信空间。运行在 RHS 上的操

作被保护并从 LHS 隔离，这使它们显著地更安全来免于被攻击。

NGSCB 也提供安全输入和输出。采用 NGSCB，一旦按键到达 RHS，在它们被软件读取并解密之前先对它们进行加密。这意味着无法使用恶意软件来记录、偷取或修改按键。安全输出也类似。出现在屏幕上的信息可以向用户呈现，使得没有其他人能够截取并读取该信息。同时采用两者允许用户高度信任地知道他的计算机中的软件正在做它所应该做的工作。

尽管对 RHS 有充分的可信资源可用，然而 LHS 仍是不可信的。本发明解决这一问题以及当前可信计算系统的其它不足。

发明内容

本发明提供了将可信环境中的实体的可信性投影到不可信环境中的实体的机制。

描述了提供不可信环境和可信环境的系统和方法。基本监控代理（agent）在可信环境中运行。基本监控代理监控不可信环境。

依照一个实施例，监控代理与应用程序关联，并且监控代理监控其关联的应用程序的可能指示攻击的事件或行为。监控代理的可信特性允许可靠地检测并报告这些事件/行为，借此将可信环境的可信性投影到不可信环境。基本监控代理能够批准、禁止或者修改由监控代理报告或发现的不可信环境事件。例如，报告覆盖了诸如硬件报告将 GDT（全局描述符表）移动到进而将其向基本监控代理报告的网络点的尝试的情况。例如，发现可以是基本监控管理（对 OS）或监控管理（对某一应用程序）通过扫描不可信应用程序的存储器发现问题的情况。

对于另一实施例，基本监控代理响应从安全输入接收的输入。例如，基本监控代理可以在没有通过安全输入接收批准的情况下拒绝允许对不可信环境的变化。作为另一示例，基本监控代理可以拒绝对不可信环境的变化，除非该变化由经批准方签署的包来描述。

对于再一实施例，监控代理使用密封存储来对驻留在不可信环境中的操作系统或应用程序保守机密。监控代理可拒绝向操作系统或应用程序揭示该机密，除非该操作系统或应用程序具有与该机密的所有者匹配的摘要。作为替代，监控代理可拒绝向操作系统或应用程序揭示机密，除非该操作系统或应用程序在可读取该机密的摘要列表上。

依照其它特点，监控代理使用测试来确定合法实体是否请求机密。一个这样的测试包括检查该实体的栈并确保该栈具有合法栈内容。此外，监控代理可编辑不可信环境的状态来令其变得安全或可接受。状态可包括初始配置或错误报告选项。

结合附图阅读以下说明性实施例的详细描述可以更清楚本发明的另外的特点和优点。

附图说明

当结合附图阅读以上概述以及以下较佳实施例的详细描述，可以获得较好的理解。为说明本发明的目的，附图中示出了本发明的示例性构造；然而，本发明不局限于所揭示的特定方法和手段。附图中：

图 1 所示是可以实现本发明的各方面的示例性计算环境的结构图；

图 2 所示是具有可信和不可信环境的现有 NGSCB 系统的结构图；

图 3 所示是依照本发明的示例性投影系统的结构图；以及

图 4 所示是依照本发明的示例性投影方法的流程图。

具体实施方式

概述

在具有运行在不可信环境中的实体和运行在可信环境中的实体的单个机器中，本发明提供了一种将可信环境中的实体的可信性投影到不可信环境中的实体的机制。本发明针对当第一执行环境（如，操作系统）主含第二执行环境时所使用的机制。本发明应用到诸如 Microsoft®的下一代安全计算基础（NGSCB），其中，常规操作系统（如，Windows®操作系统）主含安全操作系统（如，网络点）。描述了允许第二环境将其可信性投影到第一环境的各种机制。

示例计算环境

图 1 说明了适合在其中实现本发明的各方面的计算系统环境 100 的一个示例。计算系统环境 100 仅为合适的计算环境的一个示例，并非对本发明的使用或功能的范围提出任何限制。也不应将计算环境 100 解释为对示例性操作环境 100 中说明的任一组件或其组合具有依赖或需求。

本发明可以使用众多其它通用或专用计算系统环境或配置来操作。适合使用

本发明的众所周知的计算系统、环境和/或配置包括但不限于：个人计算机、服务器计算机、手持式或膝上设备、多处理器系统、基于微处理器的系统、机顶盒、移动电话、可编程消费者电子设备、网络 PC、小型机、大型机、包括任一上述系统或设备的分布式计算环境等等。

本发明可以在计算机可执行指令的一般上下文中描述，计算机可执行指令如程序模块，由计算机执行。一般而言，程序模块包括例程、程序、对象、组件、数据结构等等，执行特定的任务或实现特定的抽象数据类型。本发明也可以在分布式计算环境中实践，其中，任务由通过通信网络或其它数据传输媒质连接的远程处理设备来执行。在分布式计算环境中，程序模块和其它数据可以位于本地和远程计算机存储媒质中，包括存储器存储设备。

参考图 1，用于实现本发明的示例性系统包括以计算机 110 形式的通用计算装置。计算机 110 的组件可包括但不限于，处理单元 120、系统存储器 130 以及将各类系统组件包括系统存储器耦合至处理单元 120 的系统总线 121。系统总线 121 可以是若干种总线结构类型的任一种，包括存储器总线或存储器控制器、外围总线以及使用各类总线结构的本地总线。作为示例而非局限，这类结构包括工业标准体系结构（ISA）总线、微通道体系结构（MCA）总线、增强 ISA（EISA）总线、视频电子标准协会（VESA）本地总线以及外围部件互连（PCI）总线（也称为 Mezzanine 总线）。

计算机 110 通常包括各种计算机可读媒质。计算机可读媒质可以是可由计算机 110 访问的任一可用媒质，包括易失和非易失媒质、可移动和不可移动媒质。作为示例而非局限，计算机可读媒质包括计算机存储媒质和通信媒质。计算机存储媒质包括以用于储存信息的任一方法或技术实现的易失和非易失，可移动和不可移动媒质，信息如计算机可读指令、数据结构、程序模块或其它数据。计算机存储媒质包括但不限于，RAM、ROM、EEPROM、闪存或其它存储器技术、CD-ROM、数字多功能盘（DVD）或其它光盘存储、磁带盒、磁带、磁盘存储或其它磁存储设备、或可以用来储存所期望的信息并可由计算机 110 访问的任一其它媒质。通信媒质通常在诸如载波或其它传输机制的已调制数据信号中包含计算机可读指令、数据结构、程序模块或其它数据，并包括任一信息传送媒质。术语“已调制数据信号”指以对信号中的信息进行编码的方式设置或改变其一个或多个特征的信号。作为示例而非局限，通信媒质包括有线媒质，如有线网络或直接连线连接，以及无线媒质，

如声学、RF、红外和其它无线媒质。上述任一的组合也应当包括在计算机可读媒质的范围之内。

系统存储器 130 包括以易失和/或非易失存储器形式的计算机存储媒质，如只读存储器（ROM）131 和随机存取存储器（RAM）132。基本输入/输出系统 133（BIOS）包括如在启动时帮助在计算机 110 内的元件之间传输信息的基本例程，通常储存在 ROM 131 中。RAM 132 通常包含处理单元 120 立即可访问或者当前正在操作的数据和/或程序模块。作为示例而非局限，图 1 示出了操作系统 134、应用程序 135、其它程序模块 136 和程序数据 137。

计算机 110 也可包括其它可移动/不可移动、易失/非易失计算机存储媒质。仅作示例，图 1 说明了对不可移动、非易失磁媒质进行读写的硬盘驱动器 141、对可移动、非易失磁盘 152 进行读写的磁盘驱动器 151 以及对可移动、非易失光盘 156，如 CD ROM 或其它光媒质进行读写的光盘驱动器 155。可以在示例性操作环境中使用的其它可移动/不可移动、易失/非易失计算机存储媒质包括但不限于，磁带盒、闪存卡、数字多功能盘、数字视频带、固态 RAM、固态 ROM 等等。硬盘驱动器 141 通常通过不可移动存储器接口，如接口 140 连接到系统总线 121，磁盘驱动器 151 和光盘驱动器 155 通常通过可移动存储器接口，如接口 150 连接到系统总线 121。还考虑本发明也可以在嵌入式微处理器中实现，其中，CPU 和所有存储器都在单个包中的单个电路小片上。

图 1 讨论并说明的驱动器及其关联的计算机存储媒质为计算机 110 提供了计算机可读指令、数据结构、程序模块和其它数据的存储。例如，在图 1 中，说明硬盘驱动器 141 储存操作系统 144、应用程序 145、其它程序模块 146 和程序数据 147。注意，这些组件可以与操作系统 134、应用程序 135、其它程序模块 136 和程序数据 137 相同，也可以与它们不同。这里对操作系统 144、应用程序 145、其它程序模块 146 和程序数据 147 给予不同的标号来说明至少它们是不同的副本。用户可以通过输入设备和指向设备 161（通常指鼠标、轨迹球或触摸板）向计算机 110 输入命令和信息。其它输入设备（未示出）可包括麦克风、操纵杆、游戏垫、圆盘式卫星天线、扫描仪等等。这些和其它输入设备通常通过耦合至系统总线的用户输入接口 160 连接至处理单元 120，但是也可以通过其它接口和总线结构连接，如并行端口、游戏端口或通用串行总线（USB）。监视器 191 或其它类型的显示设备也通过接口，如视频接口 190 连接至系统总线 121。除监视器之外，计算机也包括其它外

围输出设备，如扬声器 197 和打印机 196，通过输出外围接口 195 连接。

计算机 110 可以在使用到一个或多个远程计算机，如远程计算机 180 的逻辑连接的网络化环境中操作。远程计算机 180 可以是个人计算机、服务器、路由器、网络 PC、对等设备或其它公用网络节点，并通常包括许多或所有上述与计算机 110 相关的元件，尽管在图 1 中仅示出了存储器存储设备 181。图 1 描述的逻辑连接包括局域网（LAN）171 和广域网（WAN）173，但是也可以包括其它网络。这类网络环境常见于办公室、企业范围计算机网络、内联网以及因特网。

当在局域网网络环境中使用时，计算机 110 通过网络接口或适配器 170 连接至 LAN 171。当在 WAN 网络环境中使用时，计算机 110 通常包括调制解调器 172 或其它装置，用于通过 WAN 173，如因特网建立通信。调制解调器 172 可以是内置或外置的，通过用户输入接口 160 或其它合适的机制连接至系统总线 121。在网络化环境中，描述的与计算机 110 相关的程序模块或其部分可储存在远程存储器存储设备中。作为示例而非局限，图 1 示出了远程应用程序 185 驻留在存储器设备 181 中。可以理解，示出的网络连接是示例性的，也可以使用在计算机之间建立通信链路的其它装置。

示例实施例

如上所述，本领域中已知计算机可配置成提供两种不同的环境：可信和不可信。其可信性尚未被核实的普通代码（即，其行动尚未被核实的代码，或者不能从可能达到恶意目的中排除的代码）运行在不可信环境中。普通应用软件，如游戏、字处理、电子表格等等，以及普通操作系统、设备驱动器和调试器一般落入不可信分类中。以某一方式核实其可信性的代码可运行在可信环境中。计算机的存储器的某一部分（即，“隔离”或“屏蔽”存储器）被设计成仅可由可信环境访问。

对于以下讨论，如果代理已依照被设计成保持其完整性或将其完整性的任意破坏变得明显的安全过程例示，则它是“可信”的。例如，可以通过核实代理的身份和其运行的环境的可信过程（证明）来例示代理，可以向其分配不可由其它可信或不可信代理访问的安全存储器位置（屏蔽存储器），并且它能够密封机密。这一可信代理可以被唯一且可靠地标识。

在可信环境中，限制了允许哪些代码运行。例如，有较少的可信 API（与典型 LHS 中大量丰富的 API 形成对比），运行在可信环境中的代理仅可以通过限制的

正式进程间通信（IPC）机制来相互通信，并且代理可具有对用于向用户呈现文本和图像的更限制且原始 API 和服务组的访问。这些限制降低了复杂性并因此降低了可信环境和在其中操作的可信代理的攻击表面。另一方面，不可信环境类似于通常由“开放”计算系统（如，个人计算机、手持式计算机等等）上的操作系统创建的环境－即，在这一不可信环境中，几乎任何代码都被许可执行，并且运行在标准环境中的代码对大量且丰富的编程服务和接口组具有完全的访问。不可信环境和可信环境可以被进一步划分成子环境。例如，不可信环境可以被划分成不可信用户模式（其中，普通应用程序执行）和不可信内核模式（其中，普通操作系统执行）。类似地，可信环境可以被划分成可信用户模式（其中，特殊、可信的应用程序执行）和可信内核模式（其中，对可信应用程序创建可信环境的可信操作系统执行）。

当可信和不可信环境在同一计算机系统中共存时，可信环境可采取步骤来确保其可信性不被不可信环境中发生的任何事攻击，或被可信环境中的任何用户模式代码攻击。本发明的实施例提供了为不可信端的利益投影或使用可信端的可信性的机制。

图 3 是依照本发明的投影系统的一个实施例的结构图，图 4 是依照本发明的投影方法的一个实施例的流程图。运行在计算机 110 上的系统的 LHS 类似于参考图 2 所描述的。两个应用程序 305、310 与操作系统 301 一起运行。RHS 的部分也类似于参考图 2 所描述的。两个可信代理 355、360 与网络点 351 和 SSC 353 一起运行。考虑在 LHS 上可以运行任意数量的应用程序，并且在 RHS 上可以运行任意数量的可信代理。

图 3 示出了在其中操作系统 301 和网络点 351 在单个计算机 110 上执行的系统。操作系统 301 和网络点 351 之间的逻辑分隔 350 准许出现某些通信，同时保护网络点 351 免受操作系统 301 中起源的事件的影响。

在图 3 所示的实施例中，操作系统 301 是主操作系统，网络点 351 是由 OS 301 主含的访客。即，OS 301 为网络点 351 提供某些服务和资源，如，存储器和处理器时间。对于一个实施例，逻辑分隔 350 允许网络点 351 依赖于操作系统 301 的某些资源，而仍允许网络点 351 保护其自身免受操作系统 301 中引起的并可能导致网络点 351 以与其行为说明相反的方式表现的行动（无论恶意还是无意）的影响。例如，网络点 351 及其关联的可信资源，如，SSC 353，可管理逻辑分隔。然而，可以理解，本发明不局限在特定形式的网络点 351 上。考虑那样的机制，它允许构造

分隔 350 来允许交互和保护的这一平衡。

应当注意，图 3 示出了操作系统 301 是“主”，而网络点 351 是“客”。一般而言，这一表征指在这些实施例中，操作系统 301 提供由操作系统 301 和网络点 351 使用的某些操作系统基础结构（如，设备驱动器、调度等等）。网络点 351 是“客”指的是它可以依赖于操作系统 301 的某些基础结构资源而不是自己提供它们。然而，应当注意，令操作系统为“主”或“客”的参数是灵活的。应当理解，这里描述的技术可以应用到运行在同一机器（或甚至在连接的机器上的同一组）上的任意两个或多个操作系统的交互。运行在单个机器上的两个或多个操作系统是可能需要在单个机器上彼此交互的“环境”的示例，尽管可以理解，本发明不局限于传统的操作系统。

投影是可信代理（RHS 上）的一些能力和特性能够被扩充到 LHS 代码的机制。依照一个示例，投影允许 NGSCB 个人计算机平台的能力应用到现有代码。例如，不是将诸如 Microsoft® Excel®的应用程序移植到 RHS，依照本发明的投影允许对应用程序（这里也称为受保护者（mortal））构造监控代理（这里也称为保护者（angel）），进而准许现有应用程序作为可信代理带着许多相同的有用特性运行。投影可以应用到需要某一级别的可信操作的 LHS 操作系统（如，Microsoft® Windows®）和任何操作 LHS 应用程序（如，Microsoft® Office®）。投影也可以应用到 LHS 设备驱动器。由此，如后文所描述的，投影允许可信代理保护、确保、证明并扩充 LHS 操作系统、服务和程序。

图 3 示出了对应于应用程序 305 的监控代理 390 以及对应于应用程序 310 的监控代理 395（图 4 的步骤 400）。每一监控代理或保护者保护其关联的应用程序。

对于一个实施例，关注的 LHS 实体的创建者（如，应用程序）也创建保护该 LHS 实体的保护者。这允许创建者向保护者提供它所监控的应用程序的深层知识。这一保护者可以对它所监控的应用程序中的异态更敏感，由此更有效地保护并确认它。例如，由操作系统开发者创建的基本监控代理可结合关于操作系统存储器管理的详细知识，允许它快速地标识可疑的存储器操作。

对于另一实施例，如果保护者在其关联的应用程序中检测到反常或可疑活动，它可以采取纠正或预防行动。例如，保护者可检测其关联的应用程序改变存储器中应用程序创建者认为是不变的关键字变量的尝试，并截取对该变量的写。这一写行动可能至少指示应用程序代码的破坏，如果不是被恶意，如病毒代码完全破坏的话。

简言之，保护者担当监控代理的角色，监视其关联的应用程序中的负面或可疑活动，并采取纠正或预防行动。可以限制其行动来防止保护者以免其关联的应用程序遭受破坏。例如，保护者可以挂连到特定的实体、程序或应用程序，或一组这类实体、程序和/或应用程序。

基本监控代理（这里也称为主保护者）380 与基本操作系统（即，LHS OS 301）关联（块 410）。对于一个实施例，基本监控代理 380 由 LHS 操作系统的创建者写入。这允许基本监控代理 380 结合关于 LHS 操作系统的详细知识，令它对关联的操作系统的恶意行为更敏感。

例如，主保护者能够知道虚拟地址数据库、进程数据库以及 PFN（页帧号）数据库的格式，并基于此，检测欺诈设备驱动器通过映射它们不应具有的 PFN 作出到进程的非法映射的情况。由此，主保护者能够检测不由存储器管理者作出（如，由欺诈设备驱动器作出）的映射，并能够检测不应在那里的跨进程映射。

在这一情况下，主保护者能够与改变的受保护者 OS 协作。例如，OS 和主保护者可同意 PFN 数据库应当总是一致，只要不持有特定的锁，并且应当可通过校验和（checksum）来表示该一致性。因此，以周期的间隔，主保护者这能够检查该锁，并且如果发现它未锁定（它是存储器变量，因此很容易测试），则可以进行并对 PFN 数据库求校验和。如果主保护者发现校验和不匹配，则它知道 PRN 数据库被篡改。

此外，主保护者能够知道内核调试器的控制变量，并强迫该控制变量禁用内核调试器。

另外的示例包括进程加载：监控加载器、高速缓存管理器、页错误处理器等等，来确保正确的位被正确地加载到用户模式进程（或系统中加载的任一其它模块）中，或被正确地签名，可能在主保护者所知的表内保持的散列表中列出。主保护者能够预期加载器、页错误处理器等等何时需要将代码/数据映射到/出进程中（分页等等）。RHS 能够对锁定的进程保持 LHS 物理页（甚至对 LHS OS），除非 OS 正在执行已知为好的功能。RHS 对 LHS 进程控制页表。由此，主保护者书写者能够在主保护者中融入多种机制来限制坏行为。

另一示例包括进程硬化（hardening）。对一个进程修改另一进程有已知并批准的机制。主保护者能够确保所有共享的存储器映射以及数据复制入/出不同的进程空间是受限制的。另一示例涉及只读内核，其中，内核和设备驱动器的所有“文

本”页（代码页）被锁定。

主保护者 380 也支持每进程（限制访问）到保护者的投影。例如，这意味着那些保护者，它们类似于那些代理，其中该系统将运行用户要求它运行的任何保护者（与用户政策一致），并且不是如下文定义的证明矢量的一部分（即，主保护者实际上是机器的配置的部分）能够施加严重损坏、侵入左侧私密性、探查它们不应当应用的受保护者的应用程序。因此，需要保护者十分坚固地绑定在特定的应用程序（受保护者）上。这最好通过准许保护者仅影响该保护者启动的受保护者来完成，或通过允许保护者仅应用到与保护者的清单中声明的摘要匹配的受保护者来完成，摘要检查由主保护者仅在受保护者应用程序调用保护者的摘要来启动之后完成。这一特性是期望的，因为它令允许任一应用程序销售商对其应用程序书写保护者并允许任一用户使用它变得安全且实用，而没有对其它所有东西作出严重损坏或破坏私密性的风险。

由此，主保护者既是监视 LHS 的代理，又是向其它保护者提供服务的代理。由于主保护者具有 LHS 进程结构的最详细知识，它可能是决定哪一保护者能够被绑定到哪一 LHS 进程的主保护者。限制指的是保护者（不是网络点证明矢量的一部分）仅能接触它所启动或调用它的进程，来保护它们。这防止保护者随便地在 LHS 进程上起作用。这一划分（主保护者获得 OS 级能力并类似于网络点被确认，保护者获得限制的应用级能力，并能够类似于任一其它保护者自由运行）是期望的。

对于一个实施例，保护者可在其证明矢量中包括主保护者（并通过扩充 LHS 基本 OS）。证明矢量是建立实体的安全相关配置的安全相关组件的摘要的列表。例如，代理的摘要可包括机器或主板其本身、网络点和代理其本身，以及其它信息。这一数字栈是代理是什么以及该代理运行在什么环境中的强健、可靠指示符。它允许另一实体信任它是否正在处理“真正的代理”。证明矢量化了栈（因此该代理的摘要不是网络节点矢量的部分，但是网络节点的摘要是该代理的摘要的一部分）。因此，当某一事物的证明矢量包括在另一事物中时，这指出它们都被绑定到可识别的安全配置中。证明的一个特性是它很强地标识了系统的安全相关配置。

考虑另一方式，证明矢量是定义 RHS 的软件身份的摘要值的列表。较佳地，RHS 上加载的软件在被加载前被编入摘要，并且进程其本身被较好地隔离使它无法改变。这是归纳进程：硬件对网络点的摘要进行签名（向网络点的摘要证明），网络点进而证明代理。以这一方式，外部方能够对已知列表确认这些摘要，来确定

该外部方是否批准该软件在系统上运行。由于保护者和主保护者运行在 RHS 上，它们具有较佳定义的代码身份。鉴于此原因，这些代码身份能够在描述 LHS 代码运行的环境的证明矢量中列出。由于保护者无法完全控制 LHS 代码的执行，这一代码身份陈述不如 RHS 代理的代码身份陈述有力，但是它的确意味着给定的 LHS 代码片段在的确具有强代码身份的保护者、主保护者和网络点的限制下运行。

主保护者的实施例可向保护者展现某一组 API 来提供对保护者的一些功能和/或特点的支持。例如，对任一存储器操作，主保护者是所需要的中间体。保护者可能期望检查虚拟地址 VA = 100 上的覆盖应用程序代码。然而，它可能不知道那映射到什么物理地址。网络点不知道这一结构。因此，作为替代，主保护者（知道 LHS OS 如何工作）使用基本网络点服务（只有主保护者可以调用）来读取相关的 LHS 内核存储器。主保护者使用来自 LHS OS 存储器的数据来对 LHS 应用程序存储器计算正确的映射。然后告知保护者哪一覆盖应用程序地址对应于保护者地址，保护者然后可以检查这些内容并继续处理。简言之，对于进程绑定保护者（即，仅应用到授权的进程而非在 LHS 状态上随便漫游的保护者），需要主保护者解释 LHS 数据结构。

另外的示例性功能包括提供仅允许 LHS 应用程序和 RHS 保护者看见数据的保护的 IPC 通道。LHS 内核通常能够看见通过 LHS 和 RHS 之间的 IPC 通道的所有页，但是如果这些页仅能够在主保护者的注意视线下访问，则提供了高保证，仅有关的进程（由给定保护者控制的进程）能够看见该通道中的数据。另一示例性功能给予保护者控制哪些模块（如，DLL）以及这些模块的哪些版本能够被加载到给定进程的进程空间的能力。

作为可信的实体，主保护者 380 具有对与 LHS 关联的存储器的访问，并在 LHS 上发生事件的任一时刻被通知。主保护者 380 采用它用来检测非一致性的知识体来预编程，以确定是否应当为了安全或保护采取行动。例如，主保护者 380 能够俘获某些 LHS 事件组。这些可以是被 LHS 允许的事件以及不被网络点或它所管理的可信环境排除的事件。例如，主保护者 380 能够检测指示可能的攻击或安全问题的 LHS 上的不正常映射（网络点 351 在其它情况允许的）。主保护者 380 也能够执行一致性检查。

对于图 3 所示的实施例，每一保护者被界限或者被主保护者 380 和网络点 351 监管（块 420）。主保护者 380 实施保护者及其关联的 LHS 代码之间的绑定，限

制了保护者例如在 LHS 上影响私密性和安全性的能力。

需要约束保护者的行为仅影响它们应当挂连的进程，因为依照用户的策略，网络点 351 和主保护者 380 将运行用户指示它们运行的任一保护者。主保护者具有等同于网络点的能力，并被细查至约同一级别。对于保护者，如同对任一其它代理一样，网络点运行用户吩咐它们运行的一切。因此，尽管网络点和主保护者是被约束的，普通保护者（类似于代理）不受拘束（尽管用户能够设置策略，例如，告诉网络点运行或不运行由特定估算器签名的代理或保护者）。

需要保护者是受限制的。例如，具有说明“对第一程序的保护者”的签名块的保护者不允许使用 LHS 基本 OS 存储器，或使用其它程序的存储器。允许这一使用将破坏许多用户权限，并令保护者变得危险而非有益。因此，主保护者确保保护者仅获得对它们应当能够访问的 LHS 程序的访问。

可信代理较佳地没有多于 LHS 程序的能力。具体而言，可信代理不能观察 LHS OS 或控制或编辑 LHS OS 配置状态。作为替代，保护者较佳地仅允许检查或修改它们所应用的受保护者的存储器。此外，在一些实施例中，主保护者可以不允许保护者改变受保护者的代码，限制保护者读取其受保护者的用户模式地址空间中的一切，并允许它写受保护者的非共享读写存储器空间。然而，一些机制需要允许受保护者对保护者的调用不是返回到调用点，而是返回到计算的返回点。这允许保护者强迫某些事件在受保护者中的已知的正确地址上启动——种对抗基于改变返回地址的破坏的栈的弹性攻击（trampoline attacck）的强健的方法。

保护者仅能监控其关联的实体或实体组（块 430），并且不比任一其它代理更可信。保护者不能监控或观察非关联的实体。具体而言，保护者具有以下一个或多个特性：

a. 保护者仅能够监控它所挂连的一个或多个进程（即，受保护者）的用户模式存储器（不是由 RHS 代码正常提供的能力－见上文）。

b. 只有主保护者能够看到它所挂连的 LHS OS 的内核模式存储器。

c. 保护者仅能应用到调用或要求它的那些 LHS 进程，或仅应用到它启动的 LHS 进程。

d. 保护者可由声明性实施来限制。例如，网络点和/或主保护者可以约束保护者仅投影到包含与保护者清单中声明的可执行码匹配的可执行码的进程。由此，例如，在没有人改变保护者清单的情况下，“黑客工具”的保护者不能偶尔或恶意投

影到 LHS 应用程序。这一清单变化对政策工具是显而易见的。

主保护者 380 可以实施上述限制（块 440 和 450）。为此目的可以给予主保护者对 LHS 的广泛访问，在这一情况下，它服从于与网络点类似的细查级别（即，高强度细查）。例如，主保护者具有盖过 LHS OS 的能力，并因此具有盖过在 LHS 上运行的任何事物的能力。以另一方式，主保护者能够读取任一 LHS 存储器，但是没有特别的 RHS 能力，如对 RHS 内核存储器的访问，或查看其它代理进程的能力，或对网络点或其它 RHS 代理的限制、添加、修改等等。保护者仅能读取它所应用的程序的地址空间（即，保护者具有仅应用到它们所应用的受保护者的特殊能力）。主保护者也可以读取所有的 LHS 存储器（步骤 440），而提供进程特定服务，使保护者仅能查看它们所监控并保护的程序的地址空间。

保护者可至少以以下方式“投影”其被保护者（步骤 430 和 450）：

a. 它可能在与被保护者行为的协作下可以锁定各种存储器元件或将其标记为只读，来保护对被保护者的某些变化（如，病毒攻击）。

b. 它可以在其可信空间中对被保护者执行某些关键操作。

c. 它可以坚持被保护者特定保护，如限制可以作出何种配置变化，或如果由使用安全输入机制的授权人批准则允许作出这类变化。

d. 它可以以期望的间隔扫描被保护者的存储器和状态，查找一致性错误、破坏等等，并在进一步破坏或无意识/非授权行动出现之前警告用户或暂停被保护者。

e. 它可以仅在需要时向被保护者释放密封/加密数据，以将在任一时刻可能被攻击的这一数据量最小化。

1. 它可以使用密封存储来对 LHS（或 LHS 应用程序）保存密封的机密，并拒绝将这些机密给予不具有匹配机密所有者或被列出为机密所有者允许的摘要的任一 LHS（或 LHS 应用程序）。

f. 给定正确的 API，它可以改变被保护者的执行状态；即，它能够将线程定向到已知的执行点，重定向目标应用程序中的控制流程，或对目标应用程序执行分支计算和执行。它也可以编辑配置状态、启动状态等等，以将事物强制到对被保护者的安全/校正操作的可接受模式。

g. 保护者可以调用主保护者并要求主保护者为了被保护者的利益执行预防、保护、发现或反应。

h. 保护者可以从应用程序中提取输出数据（例如，通过调用或通过存储器检

查），确认那些数据（如，求校验和等等）然后使用安全输出硬件呈现这一数据。

实体或应用程序的部分功能可以移动到保护者中。类似地，LHS 内核的部分功能可以移动到主保护者中。应用程序创建者可以在保护者中实现某些应用程序功能。尽管这将增加 RHS 的负担，但是它允许在可信环境中执行传输的功能。类似地，LHS OS 301 的一部分可以移动到主保护者 380 中。

可以以若干种方式加载或调用保护者。LHS 程序，如应用程序 305 可以调用其保护者 390。例如，以这一方式，在启动应用程序时，对应的保护者被加载。作为替代，可以从 RHS 调用保护者，然后该保护者调用对应的 LHS 进程或应用程序。保护者使用主保护者来接通 LHS 并请求启动该应用程序。主保护者然后将保护者绑定到该应用程序。对于一个实施例，网络点和主保护者向应用程序提供的 API 使它仅能看见它所创建的进程或者其子进程。

作为另一替代，可由清单调用 LHS 程序，然后转到启动保护者的 RHS，从而调用回到 LHS 来开始对应的 LHS 进程或应用程序。通常，通过命名包含它的文件来启动 LHS 程序（例如，API 为“run c:\somedir\sometherdir\someprogram.exe”）。对于 RHS 代码（代理或保护者），通过命名清单来启动，并且清单命名二进制。这是位置不相关的。同样，例如，清单通常被签名或证明，因此它们更难被欺骗。由此，示例性机制可以是向启动 LHS 应用程序和相关保护者并将其绑定在一起的 RHS（网络点）呈现组合的左/右清单。此外，保护者可以用来从 LHS 或 RHS 启动应用程序。

在本发明的一个实施例中，主保护者可以确认 LHS 进程的最初加载的代码映象匹配与该保护者关联的声明的目标代码映象。声明的目标代码映象可通过保护者的清单来提供。这防止声明为对特定应用程序的保护者替代地启动另一应用程序，从而提供了对攻击的额外安全。

依照本发明的一些实施例，防止保护者编辑它所关联的 LHS 应用程序或进程的代码映象。保护者能够读/写数据，但是仅能够读代码。

可以采用这些和类似的政策来防止保护者在不受 LHS 监控或限制的情况下运行，并且防止欺诈保护者使用 LHS 程序和应用来欺骗。

除上述启动机制以外，有其它方式来确保正确的保护者挂连到正确的 LHS（或 RHS）应用程序并保持挂连在其上。在运行应用程序作出对其保护者的调用之前可能被攻击者改变，或者 LHS 病毒可能截取或置换其调用来定位到一些其它保护者。

本发明的实施例可以通过经类似于主保护者或网络点的可信权限处理从应用程序到其保护者的调用来解决这一问题。例如，主保护者可以编制调用 LHS 应用程序的摘要，并将该摘要与同 RHS 保护者关联的“批准”摘要列表相比较。如果它们不匹配，或者由于 LHS 应用程序被置换，或者由于该调用被修改来定位不同的保护者，则该调用失败，系统能够警告用户，和/或采取任意其它数量的行动。

可以使用系统策略来指定哪一保护者挂连到哪一 LHS 应用程序。使用强健的策略机制提供难以欺骗、难以误初始化机制来设置这类依赖性。

在一些实施例中，保护者较佳地具有可调节或各种可编程检查级别，以定位对关联的应用程序的威胁。可以调节保护者对觉察的威胁或攻击的敏感度。

除向 LHS OS 或应用程序提供投影（如，防御、保护、忠告）之外，保护者也可以应用到在可信计算环境中运行的代理。在这一情况下，目标代理（通常为多疑（paranoid）实体）信任所挂连的保护者。这允许外部观察进程拦截目标代理中的各种程序错误和不正当利用。保护者能够实施安全不变量，与扫描安全错误（例如，如常规防病毒技术中）和使用网络点提供的硬性进程分隔和保护相反。

对于一个实施例，代理是虚拟机器，呈现某一真实机器的“有效相同复制品”，其中起动了 OS 映象。可信环境可允许代理访问虚拟机器的进程存储器。访问代理可监控进程存储器来保护虚拟机器免遭来自它所包含的映象的攻击。可信环境可允许保护者投影虚拟机器中的 OS 映象，并允许保护者投影虚拟机器中的应用程序。可以考虑，通常应用到 LHS 应用程序的同一机制也可以替代地应用到虚拟机器环境。

对于本发明的一个实施例，网络点向主保护者提供了 用于存储器检查和改变（至少）的 API。用于俘获改变控制结构的尝试并对其反应的 API 支持方便了投影。例如，在 x86 体系结构中，可通过 API 提供对诸如 GDT、LDT、IDT、调试寄存器、TR 等的控制结构的投影。GDT 指的是全局描述符表，LDT 指的是局部描述符表。锁定 GDTR（全局描述符表寄存器）停止了依赖于歪曲虚拟地址的意义的攻击，以允许跳至攻击者通常无法跳到的地方。IDT 是中断分派表，控制中断的路由。IDT 的位置由 IDTR（中断分派表寄存器）指示。锁定 IDTR 通过停止攻击者在其中使用 IDT 和公布出的中断的攻击以强迫将代码分支到攻击者不能达到的分支，从而令投影变得更有效。

需要可信环境（即，RHS）与开放环境（即，LHS）以某一方式连接。连接允

许可信环境检查状态并且被通知开放环境中的事件。这里的教导为包括但不限于以下结构的结构工作：

1．RHS 和 LHS 位于同一机器上，RHS 能够直接检查 LHS 存储器（而 LHS 未获许可不能检查 RHS 存储器）。

2．RHS 和 LHS 在不同的处理器上，可能具有不同的存储器，但是总线、网络、端口或其它互连允许 RHS 查看 LHS 存储器。例如，ARM 服务处理器能够运行完全可信的栈，并且可信栈能够检查 x86 MP 系统的主存储器。例如，可以有具有 x86 主处理器的机器，并且将 ARM 或 PowerPc 作为服务处理器，并使用本发明的机制来允许服务处理器监视主处理器上的软件。

3．如果 RHS 能够接收 LHS 事件的通知（如，映射的变化），但是不改变或防止它们，或不能查看 LHS 存储器，则投影的某一部分（如，微弱部分）仍是可能的。

4．RHS 能够随意检查 LHS 存储器，能够控制（即，防止或改变）到 LHS 存储器映射和地址翻译结构的 LHS 编辑，控制中断分派矢量指向何处（但是不需要控制中断控制器，尽管如果提供了这一控制，在其中有调节）。考虑到，确定 RHS 需要能够完全控制的状态/事件列表来支持强投影是要对每一处理器结构完成的任务，本领域的技术人员可以理解，该列表对不同的体系结构不同。

5．在一个实施例中，x86 TR 寄存器的变化和硬件调试寄存器的设置也可由 RHS 控制。

在当前技术的硬件中，不保证可信环境的运行，因为它可以依赖于公用中断硬件、LHS 中断分派表等等。

在以上列出的硬件中，能够控制 IDT（在 x86 上或等效的其它地方）允许 RHS 确保其所选择的某一中断将始终运行调用 RHS 的代码。

然而，LHS 攻击者或错误可以破坏中断控制器、关闭中断等等。可以考虑，使用 ATC（地址翻译控制）来确保 RHS 经常获得运行。如果 RHS 使用 ATC，它能够修改 ATC 来增加计数器。只要 RHS 调度主保护者，将计数器设置到某一值。如果计数器达到零，则 ATC 知道主保护者“很长时间”未运行，并调用强制运行主保护者的网络点条目点。这一技术不保证主保护者在特定时刻运行，但是的确保证它在若干 LHS 存储器编辑操作之后运行。由此，活动的 LHS 最终必须让主保护者运行。

如果 RHS 能够锁定 IDT，并且系统具有 NMI（不可屏蔽中断）的可靠源，则 RHS 能够强迫 NMI 处理器调用权限。

在示例性实施例中，硬件具有定时器，在许多时间单位（tick）之后强迫对 RHS 的中断。

本发明提供了允许一个计算环境的可信性投影到第二计算环境的机制。运行在单个机器上的两个或多个操作系统是需要在单个机器上彼此交互的“环境”的示例，尽管可以理解，本发明不限于传统的操作系统。此外，在一般情况下，可以使用至少一些这里描述的技术来将可信性从任一类型的可执行实体（如，任一软件片段）投影到任一其它类型的实体。

在两个实体在单个机器上并排存在并且需要彼此交互的情况下，交互可以采取各种形式。例如，两个实体可能需要相互来回传递数据。在实体为操作系统的情况下（或某些其它类型的执行环境，如在虚拟机器上执行脚本的脚本引擎），实体可能需要以某些其它方式彼此交互一如，共享存储器。共享处理器上的时间、共享资源以及处理中断。本发明提供了两个实体能够参与这些类型的彼此交互的技术，同时允许一个实体将其可信性投影到另一实体。

以上描述的实施例着眼于作为被监控的资源的存储器，但是本发明不限于此。如果安全监控程序对不同于存储器的资源可用，基本监控代理（如，主保护者）可以采用这一监控程序作为可信委托方来扩充其可信范围。例如，如果安全 NIC 可用，基本监控代理可使用它来排除发送具有某些头部的包。一般而言，这一可信委托方仅需要理解测量不变量，如匹配<regext>的头部，并可靠地提醒监控代理关于不变量的变化。

注意，仅为解释目的提供上述示例，并且不应构成对本发明的限制。尽管参考各种实施例描述本发明，应当理解，这里使用的词语是描述和说明的词语，而非限制的词语。此外，尽管这里参考特定装置、材料和实施例描述本发明，本发明不局限于这里所解释的细节；相反，本发明延及所有如处于所附权利要求范围之内的功能上等效的结构、方法和使用，。从本说明书的教导受益的本领域的技术人员可以对其作出各种修改，并且可以在不脱离本发明的各方面的范围和精神的情况下作出变化。

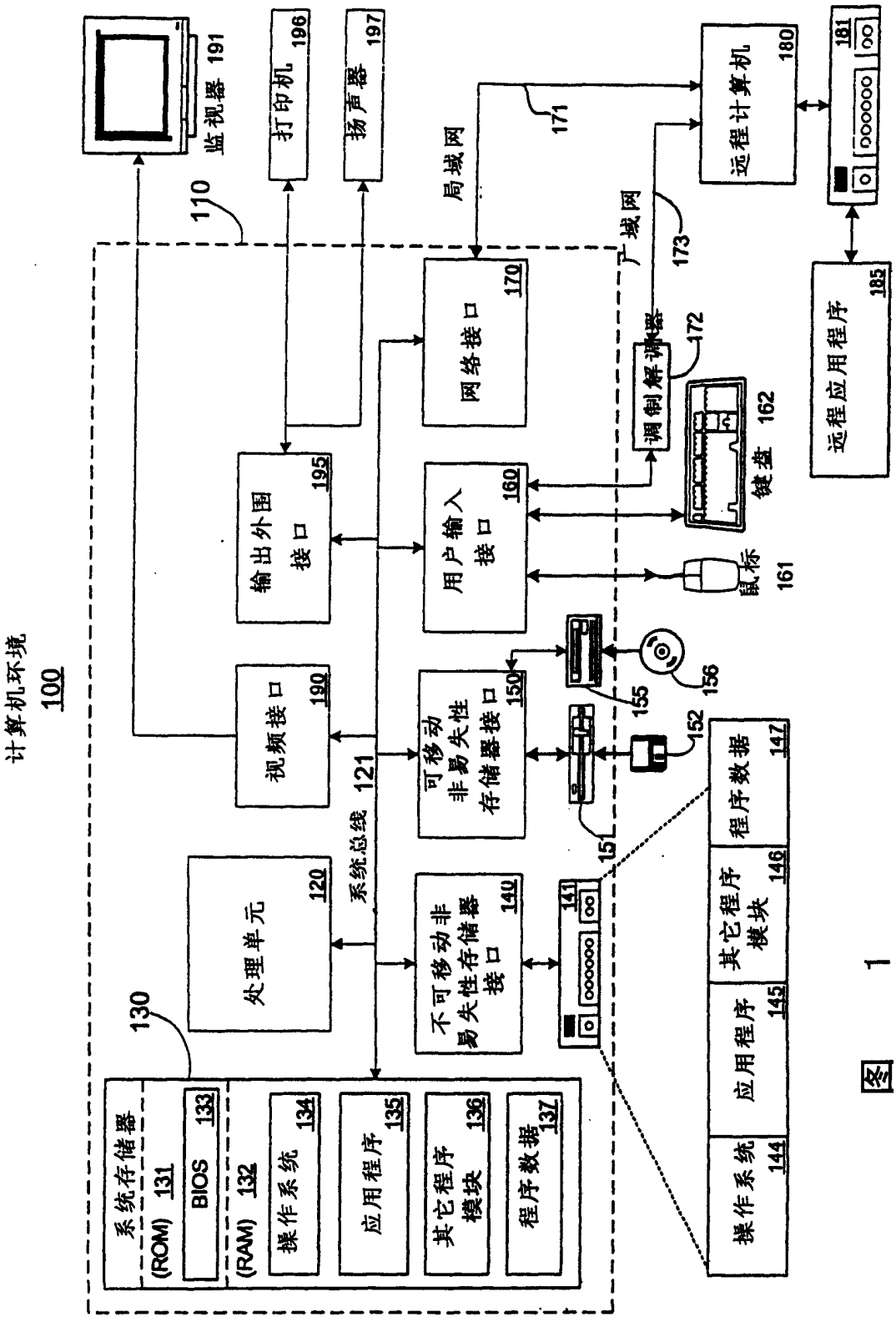

计算机环境
100
110
监视器 191
打印机 196
扬声器 197
局域网
171
广域网
173
远程计算机 180
181
远程应用程序 185
网络接口 170
调制解调器 172
键盘 162
鼠标 161
输出外围接口 195
用户输入接口 160
视频接口 190
可移动非易失性存储器接口 150
不可移动非易失性存储器接口 140
处理单元 120
系统总线 121
系统存储器 130
(ROM) 131
BIOS 133
(RAM) 132
操作系统 134
应用程序 135
其它程序模块 136
程序数据 137
操作系统 144
应用程序 145
其它程序模块 146
程序数据 147
151
152
155
156
141

图 1

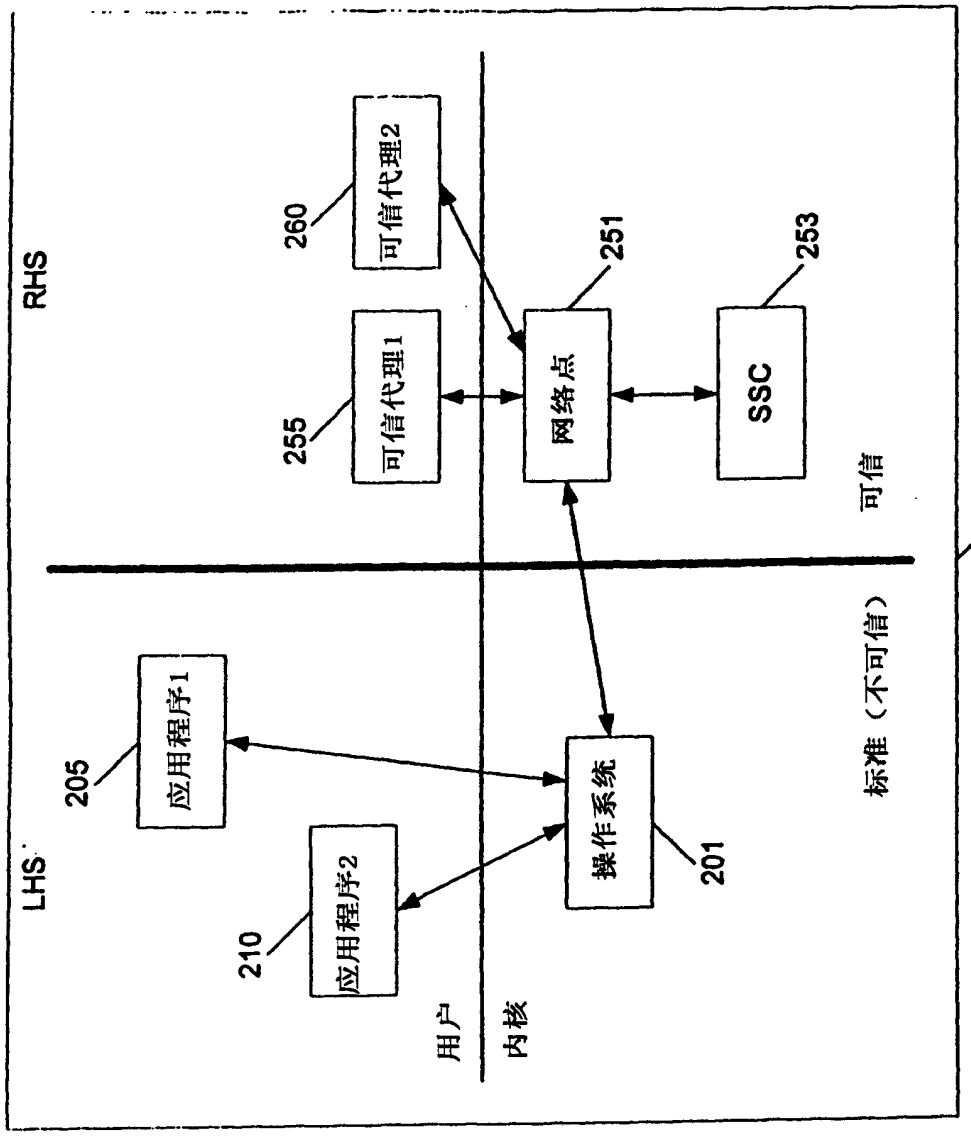

RHS
LHS
260
可信代理2
255
可信代理1
网络点
251
SSC
253
205
应用程序1
210
应用程序2
操作系统
201
用户
内核
可信
标准（不可信）
计算机 110

图 2
（现有技术）

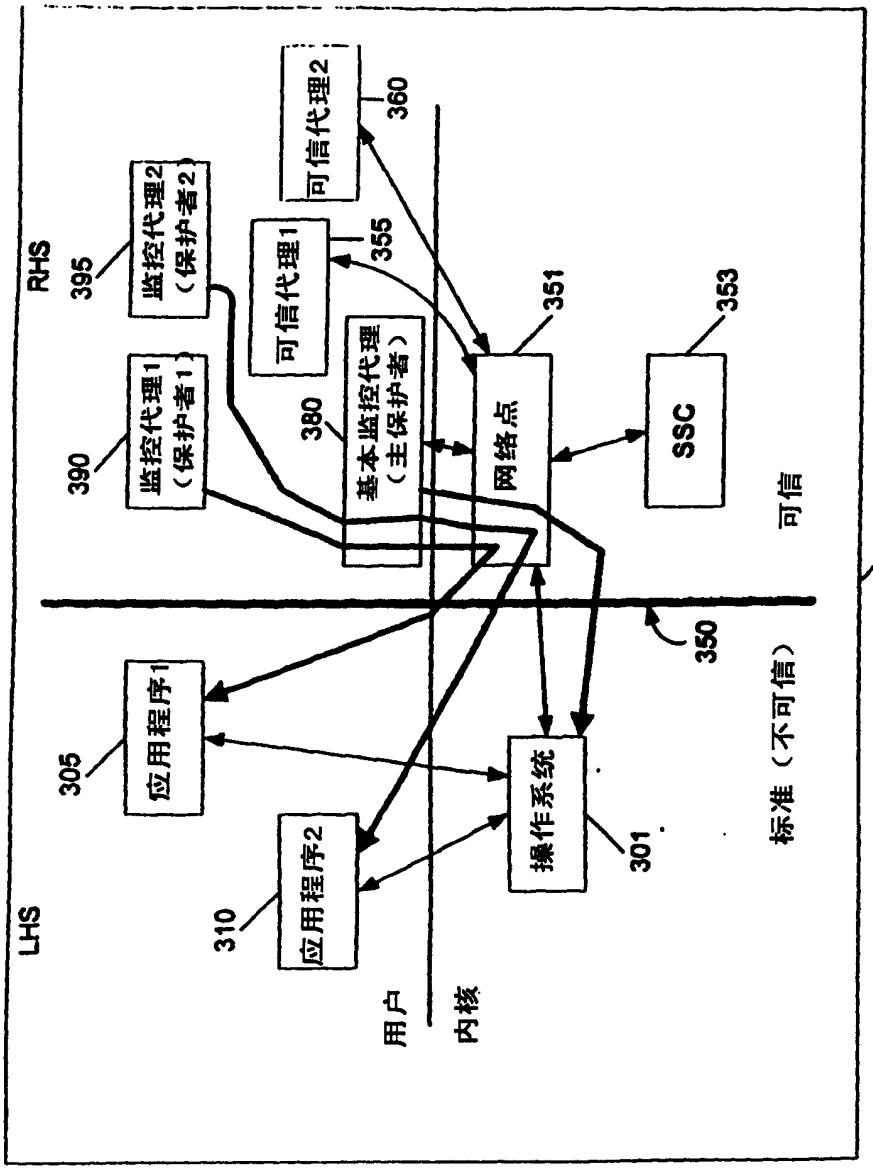

LHS
RHS
计算机 110
可信
标准（不可信）
用户
内核
应用程序1
305
应用程序2
310
操作系统
301
350
网络点
351
SSC
353
基本监控代理
（主保护者）
380
可信代理1
355
可信代理2
360
监控代理1
（保护者1）
390
监控代理2
（保护者2）
395

图 3

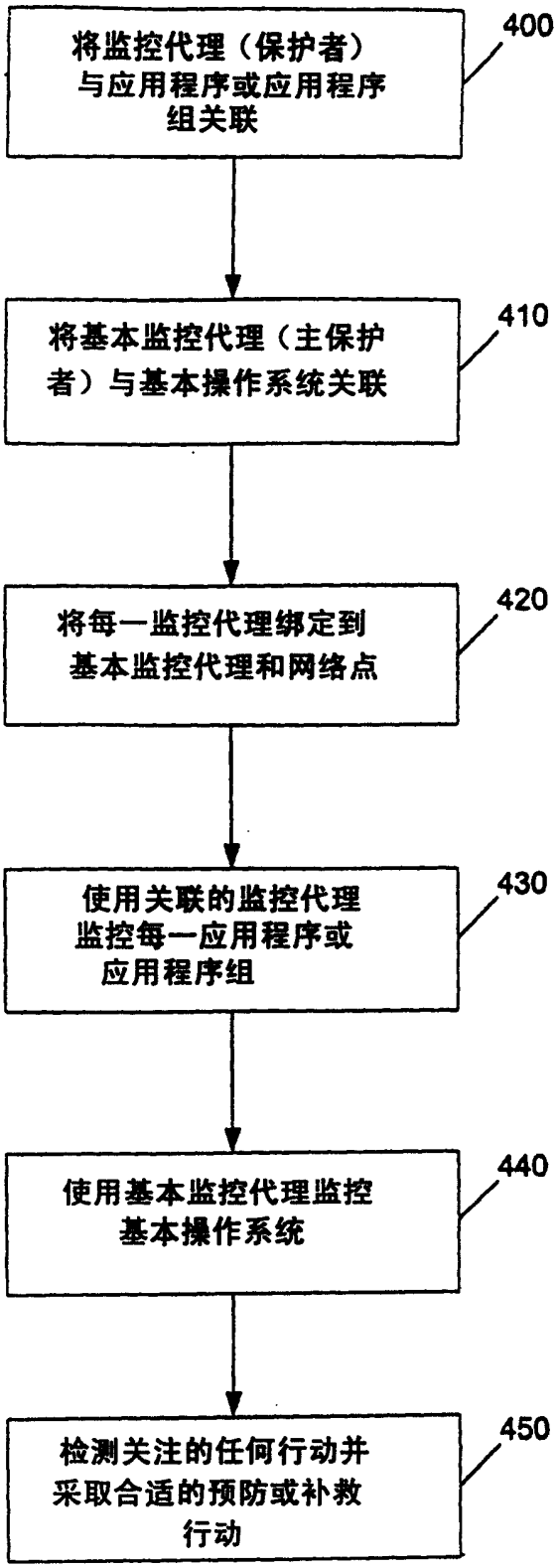

将监控代理（保护者）与应用程序或应用程序组关联
400
将基本监控代理（主保护者）与基本操作系统关联
410
将每一监控代理绑定到基本监控代理和网络点
420
使用关联的监控代理监控每一应用程序或应用程序组
430
使用基本监控代理监控基本操作系统
440
检测关注的任何行动并采取合适的预防或补救行动
450

图 4