(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 101395587 B
(45)授权公告日 2011.09.07

(21)申请号 200780007176.8
(22)申请日 2007.02.28
(30)优先权数据
11/365,364 2006.03.01 US
(85)PCT申请进入国家阶段日
2008.08.29
(86)PCT申请的申请数据
PCT/US2007/005398 2007.02.28
(87)PCT申请的公布数据
WO2007/103192 EN 2007.09.13
(73)专利权人 微软公司
地址 美国华盛顿州
(72)发明人 S·A·费尔德
(74)专利代理机构 上海专利商标事务所有限公司 31100
代理人 陈斌
(51)Int.Cl.
G06F 12/14(2006.01)
G06F 12/02(2006.01)

(56)对比文件
US 2002/0099952 A1,2002.07.25, 全文.
US 5483649 A,1996.01.09, 全文.
CN 1564992 A,2005.01.12, 全文.

审查员 张蕾

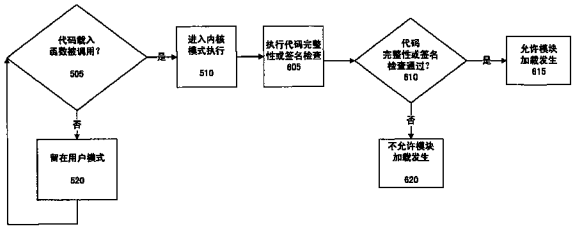
权利要求书 1 页 说明书 8 页 附图 6 页

(54)发明名称

防止可执行程序被修改

(57)摘要

通过将分配和修改现有存储器支持的代码页面的高特权操作系统(OS)函数来防止可执行代码被修改。所加载代码的完整性也可以在加载时刻在OS内核中被可选地检查。在系统中的特权检查在可执行页面被分配或修改时被调用。这一特权只被分配给操作系统内核以及该操作系统中高度可信的身份。



CN 101395587 B

1. 一种用于防止可执行代码被修改的方法，包括：

在操作系统的用户模式下确定可执行代码加载功能是否被调用；以及

响应于可执行代码加载功能被调用，将计算机的操作系统从用户模式切换到内核模式并随后执行安全检查，否则，则停留在用户模式。

2. 如权利要求 1 所述的方法，其特征在于，还包括实施所述可执行代码的页面保护。

3. 如权利要求 2 所述的方法，其特征在于，还包括当所述可执行代码的可执行页面被分配或者当所述可执行页面的属性被改变时调用特权检查。

4. 如权利要求 3 所述的方法，其特征在于，所述特权检查在允许所述可执行页面的分配或所述可执行页面属性的改变之前确定被所述操作系统所允许的分配所述可执行页面和改变所述可执行页面属性的特权是否存在。

5. 如权利要求 4 所述的方法，其特征在于，还包括在所述可执行代码被加载之后，检查其完整性。

6. 如权利要求 5 所述的方法，其特征在于，所述页面级保护使用基于硬件的实施和对可执行页面的跟踪来执行的。

7. 一种防止可执行代码被修改的系统，包括：

用于在操作系统的用户模式下确定可执行代码加载功能是否被调用的装置；以及

用于响应于可执行代码加载功能被调用，将计算机的操作系统从用户模式切换到内核模式并随后执行安全检查，否则，则停留在用户模式的装置。

8. 如权利要求 7 所述的系统，其特征在于，还包括用于实施可执行代码的页面保护的装置。

9. 如权利要求 8 所述的系统，其特征在于，还包括用于当所述可执行代码的可执行页面被分配或者当所述可执行页面的属性被改变时调用特权检查的装置。

10. 如权利要求 9 所述的系统，其特征在于，所述用于调用特权检查的装置在允许所述可执行页面的分配或所述可执行页面属性的改变之前确定被所述操作系统所允许的分配所述可执行页面和改变所述可执行页面属性的特权是否存在。

11. 如权利要求 10 所述的系统，其特征在于，还包括用于在所述可执行代码被加载之后检查其完整性的装置。

12. 如权利要求 11 所述的系统，其特征在于，所述用于实施页面级保护的装置包括用于基于硬件的实施和对可执行页面的跟踪的装置。

13. 一种防止数据页面被修改的方法，包括：

确定可执行数据页面是否被分配或者所述数据页面的属性是否被改变；

响应于可执行数据页面被分配或者所述数据页面的属性被改变，执行特权检查；以及

响应于特权存在，允许分配可执行数据页面或者改变可执行数据页面的属性。

14. 一种防止数据页面被修改的系统，包括：

用于确定可执行数据页面是否被分配或者所述数据页面的属性是否被改变的装置；

用于响应于可执行数据页面被分配或者所述数据页面的属性被改变，执行特权检查的装置；以及

用于响应于特权存在，允许分配可执行数据页面或者改变可执行数据页面的属性的装置。

防止可执行程序被修改

[0001] 背景

[0002] 如今，恶意软件代码（即，恶意软件 (malware)）通过获取对计算机 CPU 的控制进而执行恶意 CPU 指令（代码）引起对计算机系统的损害。现今应对恶意软件的方法并不是完全有效。一个常见方法是用基于签名的病毒检测工具来处理病毒。不幸的是这种方法无法发现同种攻击的下一代变种。由于病毒传播如此之快，此种病毒检测的反应性方法不能有效阻止许多种类的病毒。因此，随着新的且更有侵犯性的代码变得愈发流行，阻止恶意代码的执行变得愈发重要。

[0003] 此外，现今的操作系统允许无特权的用户代码自由统治对可执行页面的分配和修改。因此，如果攻击者能够入侵现有程序（例如，通过缓存器溢出或者其它编程错误），则他们就能自由修改存储器中被入侵的程序，或者使得来自盘或者其它介质的新 CPU 指令被执行。

[0004] 因此，需要新的过程和系统去解决现存技术的缺点。

[0005] 概述

[0006] 提供本概述以便以简化形式介绍概念的精选，这些概念将在以下的详细描述中被进一步描述。本概述并不旨在确定所要求保护的主题的关键特征或必要特征，也不旨在用于帮助确定所要求保护的主题的范围。

[0007] 考虑到上文所述本领域的缺点，提供了防止可执行代码被修改以及防止未经认可的代码被加载的方法。对于若干实施方式而言，一种防止可执行代码被修改的方法包括将把可执行代码载入存储器的函数限制到操作系统的特权环 (privilegedring)。而且，该方法还包括实施可执行代码的页面级的保护。可以当可执行代码的可执行页面被分配或者可执行页面的属性被改变时调用特权检查。例如，特权检查在允许可执行页面的分配或可执行页属性的改变之前确定只分配给 OS 特权环的特权是否存在。作为上述方法的补充或替代，在可执行代码被载入存储器之前或之后检查其完整性。

[0008] 另选的，相对于只是可执行代码，还利用一种防止数据页面被修改的方法，该方法包括将把数据页面载入存储器的合适限制到计算机的操作系统特权环。

[0009] 此项发明的其它优点和特性，将会在下文中得到描述。

[0010] 附图简述

[0011] 结合以下所附简图，防止可执行代码被修改的方法进一步讨论。

[0012] 图 1 是表示适于连同防止可执行代码被修改使用的示例性计算设备的框图，。

[0013] 图 2 示出了在其中多个计算机化过程可以被实现用于执行阻止可执行代码被修改的示例性联网计算环境。

[0014] 图 3 是示出使用特权检查阻止可执行代码被修改的过程的图示。

[0015] 图 4 是示出操作系统的用户模式和内核模式特征的示例性体系结构的框图。

[0016] 图 5 是示出在内核模式执行中使用安全检查来防止可执行代码被修改的过程的图示。

[0017] 图 6 是示出在图 5 所示的阻止可执行代码修改的过程中使用的示例性安全检查的

图示。

[0018] 详细描述

[0019] 某些特定细节在以下的描述和附图中阐述以提供对本发明各种实施方式的透彻理解。与计算和软件技术相关联的某些周知细节在以下讨论中不做阐述以免对本发明的各种实施方式产生不必要的模糊。此外，本领域普通技术人员将会理解无需以下描述的一个或多个细节也仍能够实践本发明的其它实施方式。最后，虽然对各种方法的描述参考了以下公开中的步骤和次序，但是这类描述是为提供本发明各实施方式的清楚实现，并且这些步骤和次序并不是实践本发明所必须的。

[0020] 示例性计算环境

[0021] 参见图 1，示出的是表示适于连同实现上述过程一起使用的示例性计算设备的框图。例如，用来执行防止可执行代码被修改的过程和方法的计算机可执行指令可驻留在图 1 所示的这一计算环境中和 / 或在次环境中被执行。计算系统环境 220 仅仅是一个合适的计算环境示例，而非暗示对本发明的使用或功能范围的任何限制。此外计算环境 220 也不应该被解释为对示例性操作环境 220 中所示组件的的任一个或组合具有任何依赖或要求。例如，计算机游戏控制台也可包括下文所描述的用于连同上述实现过程一起使用的那些项目。

[0022] 本发明的各方面可以用众多其它通用或专用计算系统环境或配置来操作。适合本发明使用的周知计算系统、环境和 / 或配置的示例包括但不限于个人计算机、服务器计算机、手持式或膝上型设备、多处理器系统、基于微处理器的系统、网络 PC、小型计算机、大型计算机、包括任一上述系统或设备的分布式计算环境等等。

[0023] 本发明的各方面可以在计算机可执行指令的通用上下文中实现，比如计算机执行的程序模块。一般而言，程序模块包括执行特定任务或实现特定抽象数据类型的例程、程序、对象、组件、解释代码、数据结构等等。本发明的各方面也可以在其中任务由通过通信网络链接的远程处理设备来执行的分布式计算环境中实践。在分布式计算环境中，程序模块可以位于包括存储器存储设备的本地和远程计算机存储介质两者中。

[0024] 用于实现本发明各方面的示例性系统包括具有计算机 241 形式的通用计算设备。计算机 241 的组件可以包括但不限于处理单元 259，系统存储器 222 以及用于将包括系统存储器的各种系统组件耦合至处理单元 259 的系统总线 221。系统总线 221 可以是由若干类型总线结构的任一种，包括存储器总线或存储器控制器、外围总线、使用各类总线体系结构中的任一种的局域总线。作为示例而非局限，这类体系结构可包括工业标准体系结构 (ISA) 总线、微通道体系结构 (MCA) 总线、增强型 ISA(EISA) 总线、视频电子技术标准协会 (VESA) 局部总线、以及外围部件互连 (PCI) 总线 (也称为小背板 (Mezzanine) 总线)。

[0025] 计算机 241 通常包括各种计算机可读介质。计算机可读介质可以是可由计算机 241 访问的任一可用介质，包括易失性和非易失性介质、可移动和不可移动介质。作为示例而非局限，计算机可读介质包括计算机存储介质和通信介质。计算机存储介质包括以用于储存诸如计算机可读指令、数据结构、程序模块或其它数据等信息的任一方法或技术实现的易失性和非易失性，可移动和不可移动介质。计算机存储介质包括并不局限于，RAM、ROM、EEPROM、闪存或者其它存储技术、CD-ROM、多功能数码光碟 (DVD) 或者其它的光学存储盘、磁卡带、磁带、磁盘以及其它的磁存储介质、或者可以用来由计算机 241 存储期望信息以及

能够由计算机访问的任何其他介质。通信介质通常具体化为诸如载波和其它传输机制等已调制数据信号中的计算机可读指令、数据结构、程序模块或其他数据，且包括任何信息传递介质。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改变其一个或多个特征的信号。作为示例，而非限制，通信介质包括诸如有线网络或直线连接的有线介质，以及诸如声学、RF、红外线和其它无线介质的无线介质。上述任一的组合也应当包括在计算机可读介质的范围之内。

[0026] 系统存储器 222 包括易失性和 / 或非易失性的形式的计算机存储介质，诸如只读存储器 (ROM)223 和随机存储存取器 (RAM)260。基本输入输出系统 224(BIOS) 一般被存储在 ROM223 中，它包含用于协助计算机 241 内各元件之间传递信息的基本例程。RAM260 一般包含可以被处理组件 259 立即访问和目前正在操作的数据和 / 或程序模块。作为示例而非局限，图 1 描述了操作系统 225、应用程序 226、其它程序模块 227 以及程序数据 228。

[0027] 计算机 241 也可包括其它可移动 / 不可移动、易失性 / 非易失性计算机存储介质。仅作示例，图 1 示出了对不可移动、非易失性磁介质进行读写的硬盘驱动器 238，对可移动、非易失性磁盘 239 进行读写的磁盘驱动器 254，以及对可移动、非易失性光盘 253，如 CD ROM 或其它光介质进行读写的光盘驱动器 240。可以在示例性操作环境中使用的其它可移动 / 不可移动、易失性 / 非易失性计算机存储介质包括但不限于，磁带盒、闪存卡、数字多功能盘、数字录像带、固态 RAM、固态 ROM 等等。硬盘驱动器 238 通常由不可移动存储器接口，如接口 234 连接到系统总线 221，磁盘驱动器 239 和光盘驱动器 240 通常由可移动存储器接口，如接口 235 连接到系统总线 221。

[0028] 上文所述以及图 1 所示的驱动器以及相关联的计算机存储介质给计算机 241 提供了对计算机可读指令、数据结构、程序模块以及其它数据的存储。例如，在图 1 中，示出硬盘驱动 238 存储操作系统 258、应用程序 257、其它的程序模块 256 和程序数据 255。注意，这些组件可以与操作系统 225、应用程序 226、其它程序模块 227 和程序数据 228 相同或不同。操作系统 258、应用程序 257、其它程序模块 256 和程序数据 255 在这里被标注了不同的标号是为了说明至少它们是不同的副本。用户可以通过输入设备，如键盘 251 和定位设备 252（通常指鼠标、跟踪球或触摸垫）向计算机 241 输入命令和信息。其它输入设备（未示出）可包括话筒、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 259 的用户输入接口 236 连接至处理单元 259，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 242 或其它类型的显示设备通过像视频接口 232 之类的接口与系统总线 221 相连。除了监视器外，计算机可以有通过外围接口 233 相连的其它外围输出设备，这些设备可以是扬声器 244 和打印机 243。

[0029] 计算机 241 可以使用至诸如计算机 246 的一个或多个远程计算机的逻辑连接在网络化环境中操作。远程计算机 246 可以是个人计算机、服务器、路由器、网络 PC、对等设备和其它常见网络节点，且通常包括上文相对于计算机 241 描述的许多或全部元件，虽然在图 1 中仅仅显示出了存储器存储设备 247。图 1 中所示的逻辑连接包括局域网 (LAN)245 和广域网 (WAN)249，但是还可以包括其它网络。这种联网环境在办公室、企业范围计算机网络、内联网和因特网中是常见的。

[0030] 当用于局域网环境中，计算机 241 通过网络接口或者适配器 237 与局域网 245 相

连。当用于广域网环境中，一般而言计算机 241 需要有调制解调器 250 或者其它的方式从而建立如英特网那样的广域网 249 的连接。调制解调器 250 可以是内置或外置的，它通过用户输入接口 236 或其它适当的机制连接至系统总线 221。在网络化环境中，相对于计算机 241 所描述的程序模块或其部分可被储存在远程存储器存储设备中。作为示例，而非限制，在图 1 中，远程应用程序 248 是存于存储器介质 247 中。这里所列举的网络连接只是示例，还可以有其它计算机连接方式被使用。

[0031] 应该理解，这里所讨论的各种技术可以结合硬件或软件或者两者的适当组合来实现。因此，本发明的方法和装置或其某些方面或部分可以采用包含在有形介质中的程序代码（即，指令）的形式，有形介质诸如软盘、CD-ROM、硬盘驱动器、或者任何其它机器可读存储介质，其中，当程序代码加载至诸如计算机等的机器并由其执行时，机器成为实现本发明的装置。在可编程计算机上运行程序代码的情况下，计算设备通常包括处理器、该处理器可读的存储介质（包括易失性和非易失性存储器和 / 或存储元件）、至少一个输入设备以及至少一个输出设备。一个或多个程序可以例如，通过使用 API、可重用控件等实现和利用结合本发明描述的过程。这样的程序优选地用高级过程语言或面向对象编程语言来实现，以与计算机系统通信。然而，程序也可以在需要时用汇编或机器语言实现。无论如何，语言可以是编译的和解释的语言，且与硬件实现相结合。

[0032] 虽然示例性实施方式可以涉及在一个或多个独立计算机系统的环境中利用本发明的各个方面，但本发明并不局限于此，而是可以结合像网络或者分布式计算环境之类的任何计算环境来实现。而且，本发明的各方面可以在多个处理芯片或设备中实现或者跨多个处理芯片或设备实现，且存储可以类似地跨多个设备来实现。这样的设备可以包括个人计算机、网络服务器、手持式设备、超级计算机或者集成到如汽车和飞机等其他系统中的计算机。

[0033] 考虑到其它的计算环境可能根据图 1 所示的结构框图构造各种各样的计算环境，本文提供的系统和方法并不被解释为局限于某种特定的计算机体系结构。相反地，本发明不应局限于任何单个实施方式，而是应该根据所附权利要求在更广的范围内解释。

[0034] 接下来参见图 2，示出的是在其中可以实现许多计算化过程以执行上文所描述过程的示例性联网计算环境。s。本领域普通技术人员能够认识到网络可以连接任何计算机或者其它客户机或服务设备，或者可位于分布式计算环境中。在这点上，具有任意数目处理器、存储器、存储单元以及任意数目同时运行的应用和过程的计算机系统或环境被认为是适于连同所提供的系统和方法一起使用的。

[0035] 通过计算设备和系统的交互，分布式计算提供计算机资源和服务的共享。这些资源和服务包括信息的交换、文件的高速缓存存储和盘存储。分布式计算利用网络连接性来保证客户可以利用总资源来实现整个系统的优化。从这点上来讲，各种设备都可以具有涉及本文所描述过程的应用、对象或资源。

[0036] 图 2 是示例性联网或分布式计算环境的示意图。此环境包括计算机设备 271、272、276 和 277，对象 273、274、275 以及数据库 278。这些实体 271、272、273、274、275、276、277 和 278 的每一个可以包括或使用程序、方法、数据存储、可编程逻辑等等。实体 271、272、273、274、275、276、277 和 278 可涉及相同或不同的设备的各部分，这些设备诸如 PDA、音频 / 视频设备、MP3 播放器、个人计算机等。每个实体 271、272、273、274、275、276、277 和 278 可以

通过通信网络同另一实体 271、272、273、274、275、276、277 和 278 进行通信。在这一点上，任何实体可以负责数据库 278 或其它存储设备的维护和更新。

[0037] 这一网络 270 自己可以包括为图 2 系统提供服务的其它计算实体，且自己可以表示多个互连网络。根据本发明的一方面，每个实体 271、272、273、274、275、276、277 和 278 可以包含离散功能程序模块，从而可以利用 API 或者其它对象、软件、固件和 / 或硬件向一个或多个其他实体 271、272、273、274、275、276、277 和 278 服务请求。

[0038] 还应理解诸如 275 的对象可以主存在计算设备 276 上。因此，虽然图中所描述的物理环境将连接设备示出为计算机，但这仅是示例，并且物理环境可以被替换地描绘或描述为包括如 PDA、电视、MP3 播放器等的各种数字设备以及诸如接口、COM 对象等的软件对象。

[0039] 存在有支持分布式计算环境的各种系统、组件和网络配置。例如，计算系统可以通过有线或无线系统，通过局域网或广域网连接在一起。如今，许多网络耦合至因特网，而后者给广泛分布的计算机提供基础结构并且包含了许多不同的网络。任何这些基础结构，无论其是否耦合至因特网，都可连同所提供的方法和系统一起使用。

[0040] 网络基础结构可以启用网络拓扑结构的主机，诸如客户 / 服务器、对等体、或混合体系结构。“客户”是一类或组中的成员，该类或组使用与其不相关的另一类或组的服务。在计算中，客户是进程，即一般说来是一组需要其它程序提供服务的指令或任务。客户进程利用所请求的服务而无需“知道”任何关于其他程序和服务本身的工作细节。在客户 / 服务器体系结构中，尤其是在联网系统中，客户是指能够访问由例如服务器的另一计算机提供的共享网络资源的计算机。在图 2 的例子中，任何实体 271、272、273、274、275、276、277 和 278 可以依据情况被视为客户、服务器或两者。

[0041] 典型而非必须地，服务器是可以通过诸如因特网的远程或局域网进行访问的远程计算机系统。客户进程可以在第一台计算机系统上活动，而服务器进程可以在第二台计算机系统上活动，它们经通信媒介彼此通信，从而提供分布式功能并允许多个客户利用服务器的信息汇集能力。任何软件对象都以跨多个计算设备或对象分布。

[0042] 客户和服务器利用由协议层提供的功能来彼此通信。例如，超文本传输协议 (HTTP) 是连同万维网 (WWW) 或“Web”一并使用的常见协议。通常，像因特网协议 (IP) 地址或诸如统一资源定位器 (URL) 等其他引用的计算机网络地址可用于彼此标识服务器或客户计算机。网络地址可被称为 URL 地址。可以经通信介质来提供通信，例如客户和服务器可以经由用于大容量通信的 TCP/IP 连接来彼此耦合。

[0043] 考虑到各种计算环境可根据图 2 所提供的通用架构来构建，以及在诸如图 2 的网络环境中的计算方式今后可能发生的改变，本文提供的系统和方法不能被解释为以任何方式限于某一特定的计算体系结构。相反，本发明不应限于任何单个实施方式，而是应该根据所附权利要求的宽度和范围来解释。

[0044] 防止可执行代码被修改的硬件和操作系统

[0045] 接下来参见图 3，示出的是示出使用特权检查来防止可执行代码被修改的过程框图。可以向操作系统的存储器管理器添加支持程序，用以检查当可执行存储器页面被分配或其属性被修改时，调用者是否具有特权。现代的 x64 和AMD®的 CPU 允许基于硬件的实施和可执行页面的跟踪。例如，首先从 Windows XP® ServicePack2 开始，Windows® 系统 32 位版本利用由AMD®定义的非执行页面保护 (NX) 处理器特性或者利用由Intel®定义的

执行禁用位 (Execute Disable Bit) 特性。为了使用这些处理器特性，处理器必须运行在物理地址扩展 (PAE) 模式下。Windows XP®的 64 位版本在 64 位扩展上使用 NX 处理器特性以及在 IPF 处理器上使用访问权限页表项 (PTE) 域的某些值。

[0046] 执行禁用位能力是 32 位Intel®架构的增强。具有执行禁用位能力的 IA-32 处理器可以保护数据页免于恶意软件使用来执行代码。处理器通过以下任一种模式提供页面保护：

[0047] • 传统保护模式，如果物理地址扩展 (PAE) 被启用

[0048] • IA-32e 模式，当Intel® 64 位扩展存储器技术(Intel® EM64T) 被启用时

[0049] 注意进入 IA-32 模式需要启用 PAE。虽然执行禁用位能力不引入新指令，但是它的确需要操作系统运行在启用 PAE 的环境中并且为存储器建立页面粒度保护策略。

[0050] 通过在 EAX 中输入 80000001H，软件可以使用 CPUID 指令来检测是否有执行禁用位能力。其存在通过 EDX 内的返回值指示。如果 EDX 的第 20 位被置位，则执行禁用位能力可用。如果 CPUID 扩展功能 80000001H 报告执行禁用位能力可用且 PAE 被启用，则软件可以通过将 IA32_EFERMSR(地址 C0000080H) 的 NXE 位设置为 1 来启用执行禁用位能力。如果由 CPUID 扩展功能 80000001H 返回的 EDX 寄存器的第 20 位或第 29 位为 1，则 IA32_EFER 可用。

[0051] 当物理地址扩展被启用时 (无论是在 IA-32e 的模式下还是在传统保护模式下)，执行禁用位能力通过将 IA32EFER 的第 11 位设置为 1 来启用。如果 CPUID 扩展功能 80000001H 包括执行禁用位能力不可用，则 IA32_EFER 的第 11 位将被保留。对 IA32_EFER.NXE 的写入将会产生 #GP 异常。微软 Windows®存储器管理器也会在所分配存储器页面上跟踪页面属性。

[0052] 再次参见图 3，如果可执行存储器页面被分配或它们的属性被改变 305，则执行特权检查 310 来确定 315 调用者是否具有分配存储器页面或改变其属性的正确特权。如果存在正确特权，则存储器页面的分配或属性修改得到 OS 的允许。若为否，则存储器页面的分配或属性修改不为 OS325 所允许。

[0053] 在代码能够运行之前，所有的对象代码必须被转换成可执行代码。所有的对象代码被收集到一起并且加入关于每个例程如何调用它需要调用的其他例程和系统函数的信息。在很多的软件环境中，所有的对象代码被链接在一起成为单个“可执行映像”，它是包含所有例程的大片机器代码并且被存储在盘上。在运行时，这一大型可执行映像被加载到主存储器中并在随后被执行。在防止可执行代码被修改的另一方面，诸如载入可执行映像的函数由用户模式移至内核模式。

[0054] 接下来参见图 4，示出的是用于说明操作系统的用户模式和内核模式特征的示例性体系结构的框图。例如，内核模式 405 其中执行 Microsoft NT®的核，并且在内核 405 中各组件可以直接访问硬件以及执行对包括存储器、设备和进程的计算机资源的服务。因此，只要在用户模式 410 下执行的程序想要执行 I/O、分配或解除分配虚拟存储器，开启一线程或进程，或与全局资源交互时，它必须调用 420 存留于内核模式 405 中的一个或多个服务 445。

[0055] 调用本机应用编程接口 (API) 的 KERNEL32(内核 32)425 函数直接包括其所有的 I/O(例如，CreatFile()(创建文件)、ReadFile()(读文件)、WriteFile()(写文件))、同

步（例如，WaitForSingleObject()（等待单个对象）、SetEvent()（设置事件））和存储器管理（例如 VirtualAlloc()（虚拟分配），VirtualProtect()（虚拟保护））函数。实际上，KERNEL32 的 425 输出的大部分例程直接使用本机 API。在图 4 示出了从执行 Win32 调用 (CreatFile()) 的 Win32 应用 430 经由 KERNEL32425、NTDLL435 进入内核模式 405 的控制流程，在内核模式 405 中控制被转换为 NtCreateFile(Nt 创建文件）440 系统服务。

[0056] 接下来参见图 5，示出的是用于说明使用在内核模式执行下的安全检查来防止可执行代码被修改的过程图示。在诸如将可执行映像从用户模式载入内核模式 405 的移动函数中，如果调用了加载可执行代码映像 505 的函数，则做出到内核模式 405 的切换 510，从而做出适当的安全检查。否则，应用的执行将会留在用户模式 410。普通的用户模式代码不会有足够的权限来分配可执行代码页。大部分代码不能自我修改，所以这种方式可以跨例如基于微软Windows®系统而被很广泛地实施。

[0057] 接下来参见图 6，示出的是用于说明在图 5 所示的防止可执行代码被修改的过程中使用的示例性安全检查的图示。如果应用触发了对可执行代码映像 505 的装载，则由 OS 做出到内核模式 405 的切换 510，使得合适安全检查得以做出，诸如代码完整性、签名检查或者内核之内的其他安全协议检查 605。例如，代码完整性检测可是加密校验和，即一种分派给文件并在今后用来“测试”文件以检验该文件中的数据尚未被恶意修改的数值（被称为检验和）。加密检验和通过一系列复杂的数学操作（已知为加密算法）将文件中的数据转换为被称为散列值的一串固定数字，后者则在随后用作检验和。如果不知道是用何种加密算法来产生散列值，那么未经授权的人是几乎不可能在改变数据文件的同时又能保证相应的检验和不改变。加密校验和也可被称为消息鉴别码、完整性检查值、修改检测码、或者消息完整性码。

[0058] 签名是对象中数据的加密算术摘要。因此如果验证期间对象中的数据与签名时的数据相匹配，则认为该签名匹配或者有效。无效签名基于将对象签名时被创建的加密算术摘要与签名验证期间的加密算术摘要相比较来确定。签名验证过程比较两个摘要值。如果两个值不相同，则对象内容自从签名完成后已经改变，因此可以认为该签名无效。再次参考图 6，如果代码完整性或者签名检查通过 610，那么就允许模块加载发生 615。否则，不允许模块加载发生 620。需要注意的是本文描述的过程无需仅仅针对可执行页面，它也可以扩展到例如只读数据页面和加载模块的任何其他方面。

[0059] 额外安全策略检查的一个示例涉及限制模块类型或者被加载模块的原型。例如，微软视窗系统服务可以被配置为只允许微软可执行代码以本机格式被加载。

[0060] 本文描述的各种系统、方法和技术可以使用硬件或软件或其适当组合来实现。因此本发明的方法和装载或其某些方面或部分可以采取包含在诸如软盘、CD-ROM、硬盘驱动器、或者任何其它机器可读存储介质等有形介质中的程序代码（即，指令）的形式，其中当程序代码被加载到诸如计算机之类的机器内并由其执行时，该机器成为用于实现本发明的装载。在可编程的计算机上运行程序代码的情况下，计算机通常包括处理器、该处理器可读的存储介质（包括易失性和非易失性的存储器和 / 或存储元件）、至少一个输入设备以及至少一个输出设备。一个或多个程序优选地以高级过程或面向对象的编程语言实现以与计算机系统通信。然而，程序也可以在需要时用汇编或机器语言实现。无论如何，语言可以是编译的和解释的语言，且与硬件实现相结合。

[0061] 本发明的方法和装置也可以具体化为通过某种传输介质传输的程序代码的形式，诸如经电线或电缆、通过光纤或者经由任何其它传输形式，其中当程序代码由诸如 EPROM、门阵列、可编程逻辑设备 (PLD)、诸如以下附图所示的客户计算机、视盘记录器等机器来接收、加载并执行时，此机器成为用于实践本发明的装置。当在通用处理器上运行时，程序代码结合处理器来提供执行本发明索引功能的唯一装置。

[0062] 虽然已结合各个附图的优选实施方式对本发明进行了描述，可以理解，可以使用其它类似的实施方式，或可以对所描述的实施方式进行修改或添加来实现本发明的相同功能而不背离本发明。此外，应当强调，可构想包括手持式设备操作系统和其它应用专用的硬件或软件接口系统的各种计算机平台，尤其在无线联网设备的数量持续增长时。因此本发明不应限于任何单个实施方式，而是应该根据所附权利要求的宽度和范围来解释。

[0063] 最后，本文描述的公开实施方式可以适用于其它处理器体系结构、基于计算机的系统或者系统虚拟化使用，而这些实施方式由本文公开明确预期，。因此本发明不应限于本文描述的特定实施方式而相反已被更宽泛地解释。

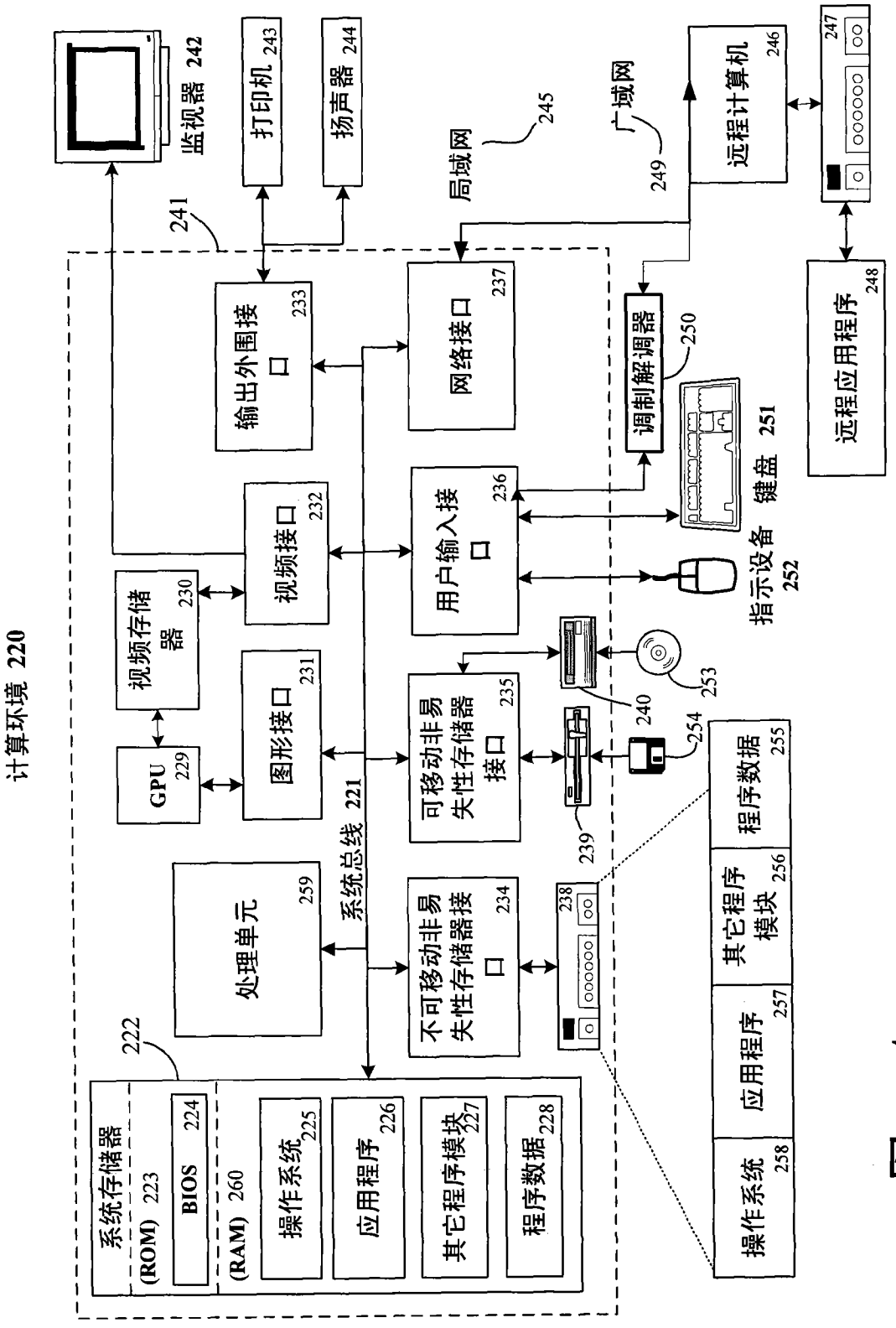

计算环境 220
监视器 242
241
打印机 243
扬声器 244
局域网 245
广域网 249
远程计算机 246
247
输出外围接口 233
网络接口 237
调制解调器 250
远程应用程序 248
视频存储器 230
视频接口 232
用户输入接口 236
键盘 251
指示设备 252
GPU 229
图形接口 231
可移动非易失性存储器接口 235
240
253
254
处理单元 259
系统总线 221
不可移动非易失性存储器接口 234
238
239
程序数据 255
其它程序模块 256
应用程序 257
操作系统 258
222
系统存储器
(ROM) 223
BIOS 224
(RAM) 260
操作系统 225
应用程序 226
其它程序模块 227
程序数据 228

图 1

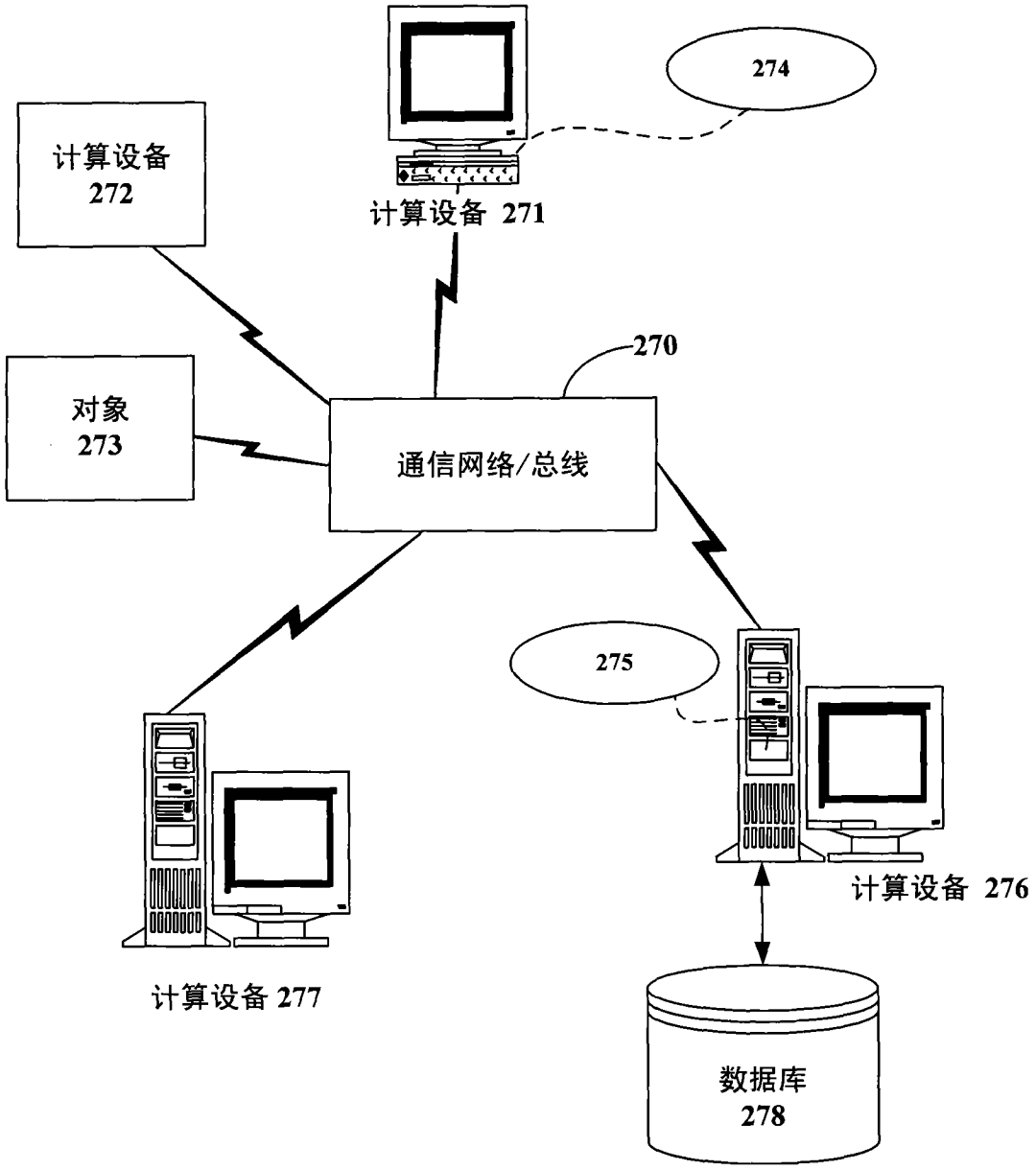

274
计算设备
272
计算设备 271
270
对象
273
通信网络/总线
275
计算设备 276
计算设备 277
数据库
278

图 2

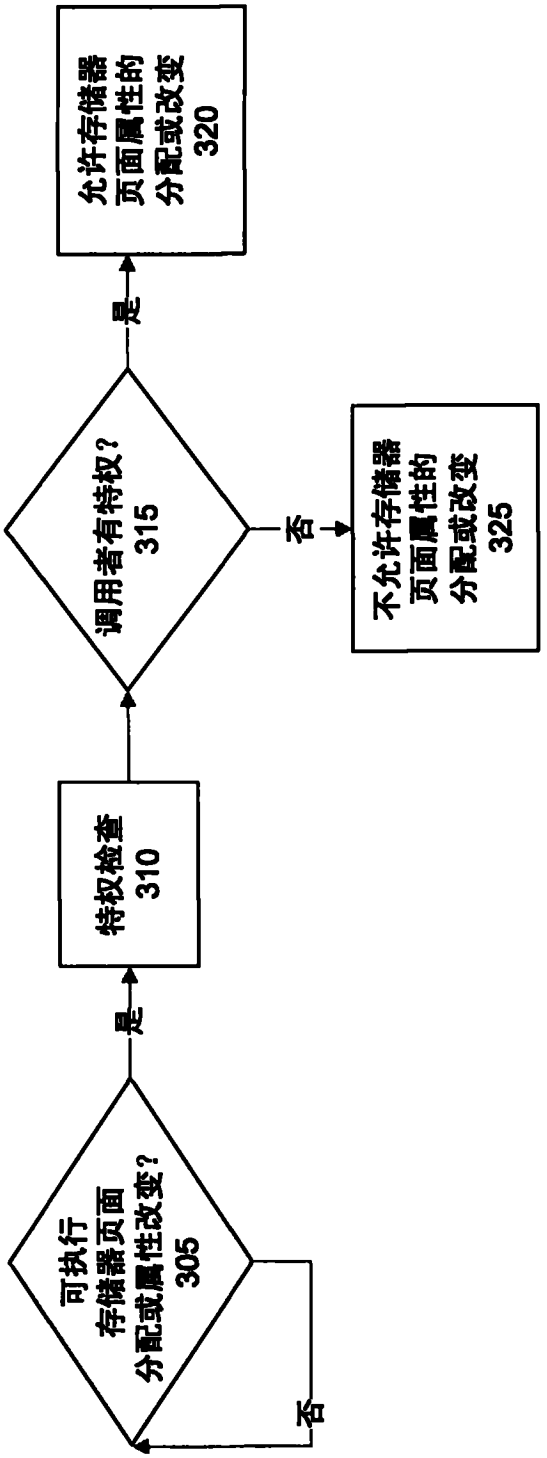

可执行
存储器页面
分配或属性改变?
305
否
是
特权检查
310
调用者有特权?
315
否
不允许存储器
页面属性的
分配或改变
325
是
允许存储器
页面属性的
分配或改变
320

图 3

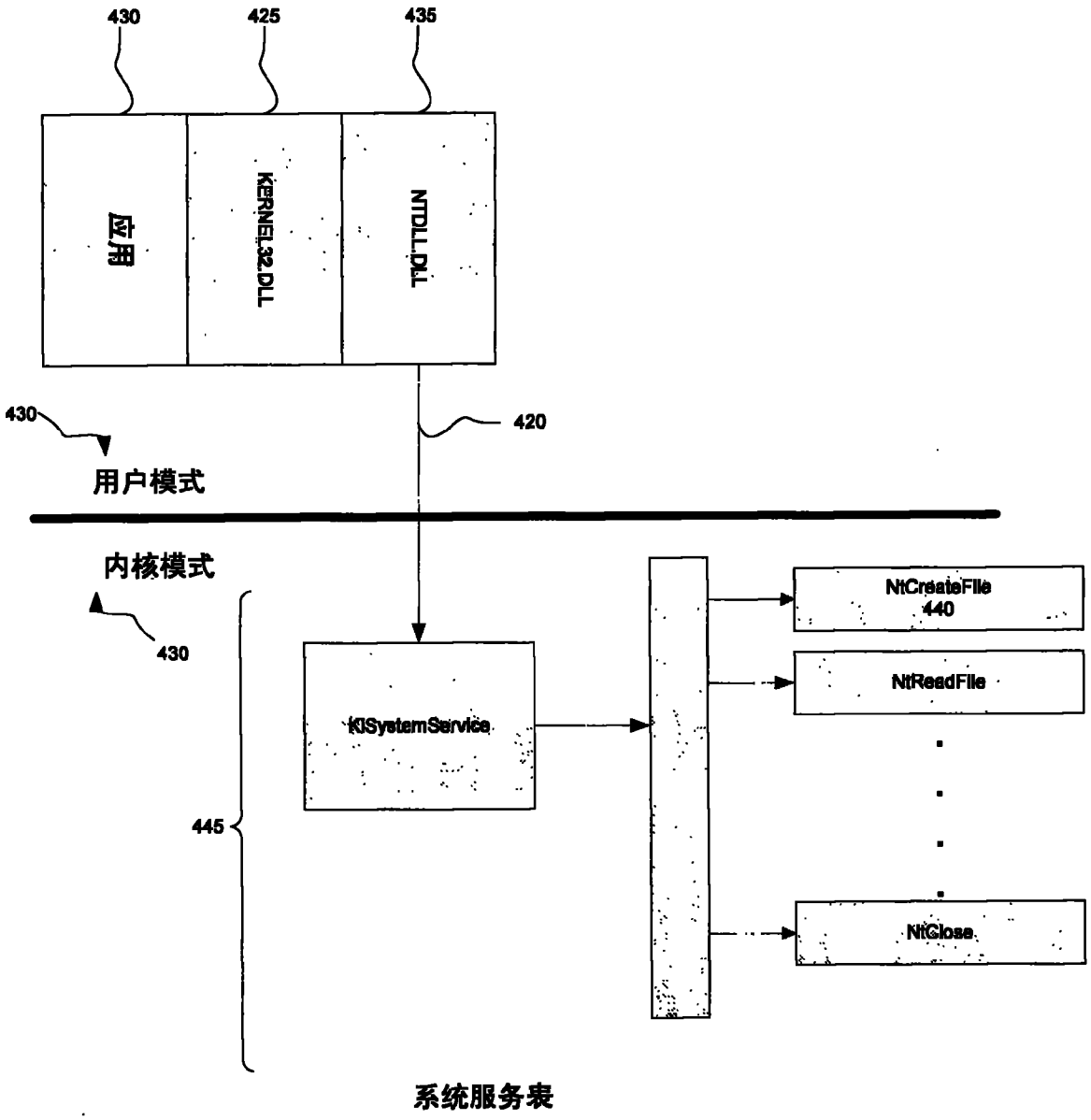

430
425
435
应用
KERNEL32.DLL
NTDLL.DLL
430
420
用户模式
内核模式
430
KiSystemService
445
NtCreateFile
440
NtReadFile
NtClose
系统服务表

图 4

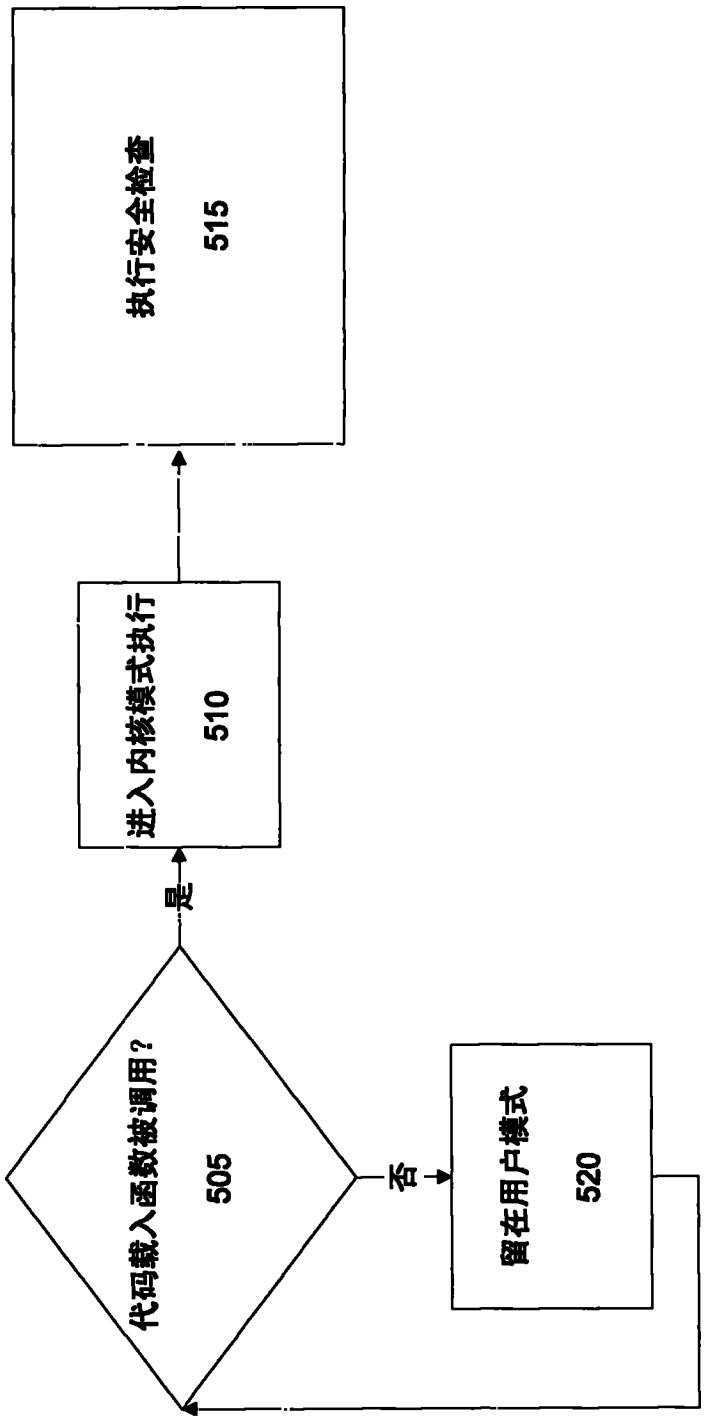


图 5

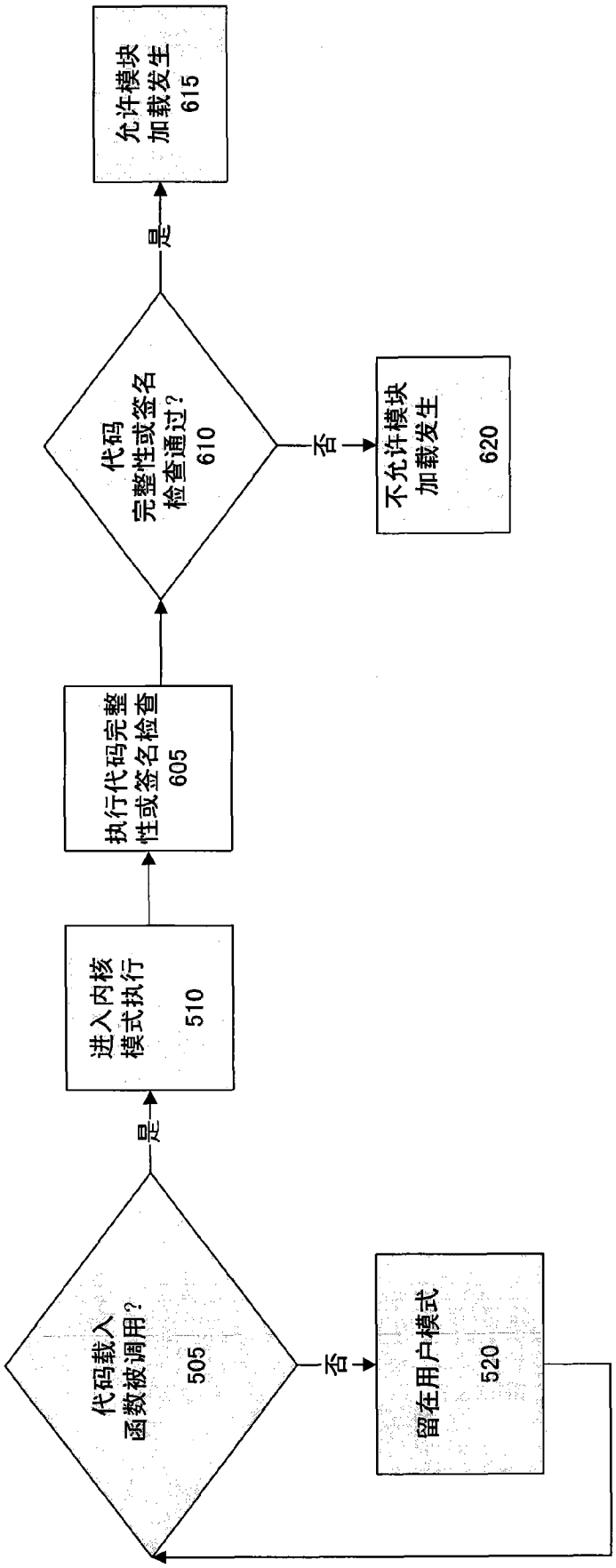

代码载入函数被调用？ 505
是
否
进入内核模式执行 510
留在用户模式 520
执行代码完整性或签名检查 605
代码完整性或签名检查通过？ 610
是
否
允许模块加载发生 615
不允许模块加载发生 620

图 6