(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 114096965 B
(45)授权公告日 2024.07.26

(21)申请号 202080050367.8
(22)申请日 2020.07.01
(65)同一申请的已公布的文献号
申请公布号 CN 114096965 A
(43)申请公布日 2022.02.25
(30)优先权数据
16/509,137 2019.07.11 US
(85)PCT国际申请进入国家阶段日
2022.01.07
(86)PCT国际申请的申请数据
PCT/EP2020/068570 2020.07.01
(87)PCT国际申请的公布数据
WO2021/004863 EN 2021.01.14

(73)专利权人 国际商业机器公司
地址 美国纽约
(72)发明人 黄海 林家俊 S·苏内亚
R·A·科勒杰米奥 M·斯坦德
(74)专利代理机构 北京市中咨律师事务所 11247
专利代理师 于静 刘薇

(51)Int.Cl.
G06F 21/57(2013.01)
(56)对比文件
CN 106663038 A,2017.05.10
CN 107003815 A,2017.08.01
审查员 杨怡睿

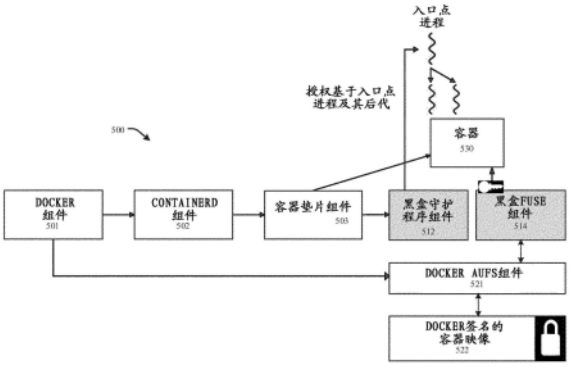
权利要求书2页 说明书19页 附图10页

(54)发明名称

容器的黑盒安全性

(57)摘要

提供了促进托管容器的安全强化系统的技术。在一个示例中,一种系统包括:存储计算机可执行组件的存储器和执行在存储器中存储的计算机可执行组件的处理器。计算机可执行组件包括:引导组件执行可信引导序列的一部分以将系统安全地引导到其中关闭对容器存储器的一种或多种类型的管理性访问的定义的安全状态。计算机可执行组件还包括:核心服务组件,其作为可信引导序列的一部分被启动,并安全地获得用于与容器存储器一起使用的一个或多个解密密钥;运行时解密组件,其使用所述一个或多个解密密钥来执行由与容器存储器相关联的容器访问的一个或多个文件的运行时解密。



CN 114096965 B

1.一种用于管理容器安全的系统,包括:

存储计算机可执行组件的存储器;

处理器,其执行存储在存储器中的计算机可执行组件,其中,计算机可执行组件包括:

引导组件,其执行可信引导序列的至少一部分,以将系统安全地引导到其中关闭对容器存储器的一种或多种类型的管理性访问的定义的安全状态;

核心服务组件,其作为可信引导序列的一部分被启动,并且安全地获得用于与容器存储器一起使用的一个或多个解密密钥;和

运行时解密组件,其使用所述一个或多个解密密钥来执行由与容器存储器相关联的容器的入口点进程或入口点进程的后代访问的一个或多个文件的运行时解密。

2.根据权利要求1所述的系统,其中,被关闭的对容器存储器的一个或多个类型的管理性访问包括以下各项中的一个或多个:

通过来自经修改的内核的引导对容器存储器的管理性访问,所述经修改的内核相对于与可信引导序列相关联的可信内核进行了修改;或者

通过加载除了可信内核的内核模块之外的附加内核模块对容器存储器的管理性访问。

3.根据权利要求1所述的系统,其中,被关闭的对容器存储器的一个或多个类型的管理性访问包括以下各项中的一个或多个:

通过一个或多个虚拟存储器管理设备对容器存储器的管理性访问;或通过一个或多个运行时调试功能对容器存储器的管理性访问。

4.根据权利要求1所述的系统,其中,被关闭的对容器存储器的一个或多个类型的管理性访问包括以下各项中的一个或多个:

通过暂停运行进程以查看与运行进程相关联的存储器对容器存储器的管理性访问;或者

通过内核存储器交换操作对容器存储器的管理性访问。

5.根据权利要求1至4中任一项所述的系统,其中,可信引导序列包括测量核心服务组件并且将所得到的核心服务组件测量存储在可信处理模块TPM平台配置寄存器(PCR)中。

6.根据权利要求1至4中任一项所述的系统,其中,核心服务组件向可信第三方服务设备运用可信处理模块TPM远程认证来安全地获得所述一个或多个解密密钥。

7.根据权利要求1至4中任一项所述的系统,其中,运行时解密组件传递对非加密文件的一个或多个请求,并且其中,运行时解密组件检查请求加密文件的一个或多个进程的进程标识符PID,以确保所述PID属于容器的入口点进程或入口点进程的后代进程。

8.根据权利要求1至4中任一项所述的系统,其中,运行时解密组件采用所述一个或多个解密密钥来解密加密的容器映像,以便实例化容器。

9.一种用于管理容器安全的计算机实现的方法,所述方法包括:

由操作地耦合到处理器的引导组件执行可信引导序列的至少一部分,以将系统安全地引导到其中关闭对容器存储器的一种或多种类型的管理性访问的定义的安全状态;

由引导组件作为可信引导序列的一部分启动操作地耦合到处理器的核心服务组件;

由核心服务部件安全地获得用于与容器存储器一起使用的一个或多个解密密钥;和

由操作地耦合到处理器的运行时解密组件使用所述一个或多个解密密钥来执行由与容器存储器相关联的容器访问的一个或多个文件的运行时解密。

10.根据权利要求9所述的计算机实现的方法,其中,被关闭的对容器存储器的一个或多个类型的管理性访问包括以下各项中的一个或多个:

通过来自经修改的内核的引导对容器存储器的管理性访问,所述经修改的内核相对于与可信引导序列相关联的可信内核进行了修改;或者

通过加载除了可信内核的内核模块之外的附加内核模块对容器存储器的管理性访问。

11.根据权利要求9所述的计算机实现的方法,其中,被关闭的对容器存储器的一个或多个类型的管理性访问包括以下各项中的一个或多个:

通过一个或多个虚拟存储器管理设备对容器存储器的管理性访问;或通过一个或多个运行时调试功能对容器存储器的管理性访问。

12.根据权利要求9所述的计算机实现的方法,其中,被关闭的对容器存储器的一个或多个类型的管理性访问包括以下各项中的一个或多个:

通过暂停运行进程以查看与运行进程相关联的存储器对容器存储器的管理性访问;或者

通过内核存储器交换操作对容器存储器的管理性访问。

13.根据权利要求9至12中任一项所述的计算机实现的方法,其中,可信引导序列包括测量核心服务组件并且将所得到的核心服务组件测量存储在可信处理模块TPM平台配置寄存器PCR中,并且其中,安全地获得所述一个或多个解密密钥向可信第三方服务设备运用可信处理模块TPM远程认证。

14.根据权利要求9至12中任一项所述的计算机实现的方法,其中,由操作地耦合到处理器的运行时解密组件使用所述一个或多个解密密钥来执行由与容器存储器相关联的容器访问的一个或多个文件的运行时解密包括:

由所述运行时解密组件传递对非加密文件的一个或多个请求;和

运行时间解密模块检查请求加密文件的一个或多个进程的进程标识符PID,以确保所述PID属于所述容器的入口点进程或入口点进程的后代进程。

15.根据权利要求9至12中任一项所述的计算机实现的方法,进一步包括由运行时解密组件采用所述一个或多个解密密钥来解密加密的容器映像,以便实例化容器。

16.根据权利要求9至12中任一项所述的计算机实现的方法,其中,所述计算机实现方法通过在容器服务器处的容器的实例化和运行时间期间保护与容器相关联的数据免受对容器存储器的一个或多个类型的管理性访问来增强在容器服务器处的容器的安全性。

17.一种用于管理容器安全的计算机程序产品,所述计算机程序产品包括:

计算机可读存储介质,可由处理电路读取并且存储用于由处理电路执行以执行根据权利要求9至16中任一项所述的方法的指令。

容器的黑盒安全性

技术领域

[0001] 本公开涉及操作系统级虚拟化技术,其中操作系统内核支持多个被称为容器的隔离的用户空间实例。

背景技术

[0002] “容器”技术的使用在云计算环境中越来越受欢迎,这在很大程度上是因为容器具有虚拟机的许多好处,诸如降低的物理基础设施成本和更好的可扩展性和灵活性,而没有操作系统倍增以及与虚拟机相关联的相应更高的资源开销。本说明书使用术语“容器”来描述本文中的技术的一方面,然而,应当理解的是,行业中已知有容器的其他术语。例如,容器有时被称为开放容器倡议(OCI)容器、Kubernetes容器、Windows服务器容器、Hyper-V容器、Intel Clear容器或Kata容器。容器技术通常允许可移植容器在一个或多个虚拟机或其他操作系统上运行。容器是隔离的,因此不能彼此干扰,并且不能在未经许可的情况下访问彼此的资源。在此使用的术语“容器”不限于任何特定类型的容器。

[0003] 本说明书使用术语“容器引擎”来描述本文中的技术的另一方面,然而,应当理解的是,行业中已知有容器引擎的其他术语。容器引擎通常为隔离容器的容器提供运行时环境。“Dockers”是广泛使用的容器引擎的实例。容器引擎通常可以尤其包括容器守护程序,其提供应用编程接口(API)和其他特征功能以供容器使用。容器引擎可进一步包括负责启动、停止、暂停、取消暂停和删除容器的执行逻辑。在此使用的术语“容器引擎”不限于任何特定类型的容器引擎。

[0004] 容器通常以“容器映像”的形式进行传输和存储,可以存储在本地或网络容器注册表中。容器映像可以用任何期望的信息进行标记。在一些情况下,容器映像可由其256位哈希来标识。

[0005] 在一些情况下,容器可以在多个协作容器的“群集”中协作。群集是多个协作容器的组。群集可包括在网络上协作的一组节点中的每个节点处的容器。服务可以在群集上运行,而不是在单个容器上运行。每个群集具有将任务分派给工作者的管理者,并且这些管理者还可以充当工作者。管理者可以选择一个领导者来分配任务并且重新分配失败的工作者的任务。如果先前选择的领导者失败,则除领导者以外的管理者可以随时准备选择新的领导者。使用群集,使用容器的服务可以根据需要进行缩放。

[0006] 容器采用多种安全功能。一般而言,容器技术提供容器隔离,因此容器不能彼此干扰,未经许可不能访问彼此的资源。此外,可以对容器映像或其部分进行加密,以在注册表中存储容器映像时或在传输容器映像时保护容器代码和数据。然而,一旦容器映像被下载到带有加密密钥的主机,所有容器映像内容都以明文解密,容易受到恶意管理员(即,运行容器的操作系统的根用户)的横向攻击和窥探的影响。本公开的多个方面提供防止此类管理性访问的安全强化措施,由此改进容器的安全性。

[0007] 容器通常托管在云计算环境中的服务器上。应当理解,虽然本公开包括云计算实施例的详细描述,但是本文所陈述的教导的实现不限于云计算环境。相反,本公开的实施例

能够结合现在已知的或以后开发的任何其他类型的计算环境来实现。

[0008] 因此,在本领域中需要解决上述问题。

发明内容

[0009] 本文描述的一个或多个实施例中,描述了有助于容器的安全强化的系统、计算机实现的方法、装置和/或计算机程序产品。

[0010] 从第一方面来看,本发明提供了一种用于管理容器安全性的系统,包括:存储计算机可执行组件的存储器;处理器,其执行存储在存储器中的计算机可执行组件,其中,计算机可执行组件包括:引导组件,其执行可信引导序列的至少一部分,以将系统安全地引导到其中关闭对容器存储器的一种或多种类型的管理性访问的定义的安全状态;核心服务组件,其作为可信引导序列的一部分被启动,并且安全地获得用于与容器存储器一起使用的一个或多个解密密钥;以及运行时解密组件,其使用所述一个或多个解密密钥来执行由与容器存储器相关联的容器的入口点进程或入口点进程的后代访问的一个或多个文件的运行时解密。

[0011] 从另一方面来看,本发明提供了一种用于管理容器安全的计算机实现的方法,该方法包括:由操作地耦合到处理器的引导组件执行可信引导序列的至少一部分,以将系统安全地引导到其中关闭对容器存储器的一种或多种类型的管理性访问的定义的安全状态;由引导组件作为可信引导序列的一部分启动操作地耦合到处理器的核心服务组件;由核心服务部件安全地获得用于与容器存储器一起使用的一个或多个解密密钥;以及由操作地耦合到处理器的运行时解密组件使用所述一个或多个解密密钥来执行由与容器存储器相关联的容器访问的一个或多个文件的运行时解密。

[0012] 从另一方面来看,本发明提供了一种用于管理容器安全性的计算机程序产品,该计算机程序产品包括计算机可读存储介质,该计算机可读存储介质可由处理电路读取并且存储用于由该处理电路执行以便执行用于执行本发明的步骤的方法的指令。

[0013] 从另一方面来看,本发明提供一种存储在计算机可读介质上并且可加载到数字计算机的内部存储器中的计算机程序,该计算机程序包括当所述程序在计算机上运行时用于执行本发明的步骤的软件代码部分。

[0014] 从另一方面来看,本发明提供一种促进容器安全的计算机程序产品,所述计算机程序产品包括计算机可读存储介质,所述计算机可读存储介质具有随其体现的程序指令,所述程序指令可由处理组件执行以使处理组件:由处理组件执行可信引导序列的至少一部分以将计算系统安全地引导到其中关闭对容器存储器的一种或多种类型的管理性访问的定义的安全状态;作为可信引导序列的一部分,启动核心服务组件;由核心服务部件安全地获得用于与容器存储器一起使用的一个或多个解密密钥;以及由处理组件使用所述一个或多个解密密钥来执行由与容器存储器相关联的容器访问的一个或多个文件的运行时解密。

[0015] 根据实施例,一种系统可以包括:存储计算机可执行组件的存储器;以及处理器,处理器,其执行存储在存储器中的计算机可执行组件。计算机可执行组件包括引导组件,其执行可信引导序列的至少一部分,以将系统安全地引导到其中关闭对容器存储器的一种或多种类型的管理性访问的定义的安全状态。作为可信引导序列的一部分而启动的核心服务组件可以安全地获得用于与容器存储器一起使用的一个或多个解密密钥。运行时解密组件

可使用一个或多个解密密钥来执行由与容器存储器相关联的容器访问的一个或多个文件的运行时解密。

[0016] 根据一个实施例,一种计算机实现的方法可以包括由操作地耦合到处理器的引导组件执行可信引导序列的至少一部分以将计算系统安全地引导到其中关闭对容器存储器的一种或多种类型的管理性访问的定义的安全状态。引导组件可以作为可信引导序列的一部分启动操作性地耦合到处理器的核心服务组件。所述计算机实现的方法还可包括由核心服务组件安全地获得用于与容器存储器一起使用的一个或多个解密密钥,并且由操作地耦合到处理器的运行时解密组件使用所述一个或多个解密密钥来执行由与容器存储器相关联的容器访问的一个或多个文件的运行时解密。

[0017] 根据另一个实施例,一种促进容器安全性的计算机程序产品可以包括计算机可读存储介质,该计算机可读存储介质具有随其体现的程序指令。程序指令可由处理组件执行并且使处理组件:执行可信引导序列的至少一部分以将计算系统安全地引导其中关闭对容器存储器的一种或多种类型的管理性访问的定义的安全状态;以及作为可信引导序列的一部分,启动核心服务。程序指令还可执行以:由核心服务安全地获得用于与容器存储器一起使用的一个或多个解密密钥,以及由处理组件使用所述一个或多个解密密钥来执行由与容器存储器相关联的容器访问的一个或多个文件的运行时解密。

[0018] 根据另一实施例,一种计算机实现的方法可包括:由操作耦合到处理器的系统通过操作耦合到处理器的docker组件实例化一个或多个容器;以及针对所述一个或多个容器中的至少一个,由所述系统管理所述一个或多个容器中的至少一个进行的一个或多个文件访问,其中,运行时解密组件传递访问非加密文件的一个或多个请求,并且其中,运行时解密组件检查请求访问加密文件的一个或多个进程的进程标识符(PID),以确保所述PID属于所述一个或多个容器中的至少一个的入口点进程或入口点进程的后代进程。

[0019] 根据又一实施例,一种计算机实现的方法可以包括由操作地耦合到处理器的引导组件来执行可信引导序列的至少一部分以安全地引导计算系统,其中,可信引导序列包括:重置一个或多个可信处理模块(TPM)平台配置寄存器(PCR);以及运行一系列引导组件,其中该系列中的每个引导组件测量该系列中的下一引导组件并存储对应的PCR值。该系列引导组件中的至少一个引导组件可包括核心服务组件。所述计算机实现的方法还可以包括:由操作地耦合到处理器的核心服务组件向可信第三方服务设备进行TPM远程认证;由核心服务组件从可信第三方服务设备安全地获得用于容器的一个或多个解密密钥;以及将用于容器的一个或多个解密密钥存储在与容器相关联的容器存储器中。

附图说明

[0020] 现在将参考如在以下附图中所展示的优选实施例仅通过实例的方式来描述本发明:

[0021] 图1示出了根据本文描述的一个或多个实施例的云计算环境。

[0022] 图2示出了根据本文描述的一个或多个实施方式的抽象模型层。

[0023] 图3示出了根据本文描述的一个或多个实施例的示例非限制性计算环境的框图。

[0024] 图4A和4B示出了根据本文描述的一个或多个实施例的检索、存储和运行容器的示例非限制性系统的框图。

[0025] 图5示出了根据本文所述的一个或多个实施例的采用修改的docker技术来保护容器的示例非限制性系统的框图。

[0026] 图6示出了根据本文所述的一个或多个实施例的示例非限制性引导组件的高级框图。

[0027] 图7示出根据本文描述的一个或多个实施例的示例非限制性核心服务组件的高级框图。

[0028] 图8示出了根据本文描述的一个或多个实施例的示例非限制性运行时解密组件的流程图。

[0029] 图9示出根据本文描述的一个或多个实施例的促进托管容器的系统的安全强化的示例非限制性计算机实现的方法的流程图。

[0030] 图10示出了根据本文描述的一个或多个实施例的促进托管容器的系统的安全强化的示例非限制性计算机实现的方法的流程图。

[0031] 图11示出根据本文描述的一个或多个实施例的促进托管容器的系统的安全强化的示例非限制性计算机实现的方法的流程图。

具体实施方式

[0032] 以下详细说明仅是说明性的并且不旨在限制实施例和/或实施例的应用或使用。

[0033] 现在参考附图描述一个或多个实施例,其中在全文中用相同的附图标记指代相同的元件。在以下描述中,出于解释的目的,阐述了许多具体细节以提供对一个或多个实施例的更透彻理解。然而,明显的是,在各种情况下,可以在没有这些具体细节的情况下实践一个或多个实施例。

[0034] 云计算是服务交付的模型,用于使得能够方便地、按需地网络访问可配置计算资源(例如,网络、网络带宽、服务器、处理、存储器、存储、应用、虚拟机和服务)的共享池,所述可配置计算资源可以以最小的管理努力或与所述服务的提供者的交互来快速供应和释放。该云模型可以包括至少五个特性、至少三个服务模型和至少四个部署模型。

[0035] 特性如下:

[0036] 按需自助服务:云消费者可以单方面地根据需要自动地提供计算能力,诸如服务器时间和网络存储,而不需要与服务的提供者的人类交互。

[0037] 广泛的网络接入:能力可通过网络获得并且通过标准机制接入,该标准机制促进异构瘦客户机平台或厚客户机平台(例如,移动电话、膝上型计算机和PDA)的使用。

[0038] 资源池:提供者的计算资源被池化以使用多租户模型来服务于多个消费者,其中不同的物理和虚拟资源根据需要动态地指派和重新指派。存在位置独立性的感觉,因为消费者通常不具有对所提供的资源的确切位置的控制或知识,但是能够以更高的抽象级别(例如,国家、州、或数据中心)指定位置。

[0039] 快速弹性:能够快速和弹性地提供能力,在一些情况下自动地快速缩小和快速释放以快速放大。对于消费者而言,可用于供应的能力通常显得不受限制并且可以在任何时间以任何数量购买。

[0040] 测量的服务:云系统通过在适合于服务类型(例如,存储、处理、带宽和活动用户账户)的某个抽象级别处利用计量能力来自动控制和优化资源使用。可以监视、控制和报告资

源使用,为所利用的服务的提供者和消费者提供透明度。

[0041] 服务模型如下:

[0042] 软件即服务(SaaS):提供给消费者的能力是使用在云基础设施上运行的提供者的应用。可通过诸如web浏览器(例如,基于web的电子邮件)之类的瘦客户端接口从不同客户端设备访问应用。消费者不管理或控制包括网络、服务器、操作系统、存储或甚至单独的应用能力的底层云基础设施,可能的例外是有限的用户特定应用配置设置。

[0043] 平台即服务(PaaS):提供给消费者的能力是将消费者创建的或获取的使用由提供商支持的编程语言和工具创建的应用部署到云基础设施上。消费者不管理或控制包括网络、服务器、操作系统或存储的底层云基础设施,但是对所部署的应用和可能的应用托管环境配置具有控制。

[0044] 基础设施即服务(IaaS):提供给消费者的能力是提供处理、存储、网络和消费者能够部署和运行任意软件的其他基本计算资源,所述软件可以包括操作系统和应用。消费者不管理或控制底层云基础设施,而是具有对操作系统、存储、所部署的应用的控制以及对所选联网组件(例如,主机防火墙)的可能受限的控制。

[0045] 部署模型如下:

[0046] 私有云:云基础架构仅为组织运作。它可以由组织或第三方管理,并且可以存在于场所内或场所外。

[0047] 社区云:云基础架构被若干组织共享并支持共享了关注(例如,任务、安全要求、策略、和合规性考虑)的特定社区。它可以由组织或第三方管理,并且可以存在于场所内或场所外。

[0048] 公共云:使云基础架构对公众或大型行业组可用,并且由出售云服务的组织拥有。混合云:云基础架构是两个或更多个云(私有、社区或公共)的组合,这些云保持唯一实体但通过使数据和应用能够移植的标准化或专有技术(例如,云突发以用于云之间的负载平衡)绑定在一起。

[0049] 云计算环境是面向服务的,集中于无状态、低耦合、模块化和语义互操作性。云计算的核心是包括互连节点网络的基础设施。

[0050] 现在参见图1,描述了说明性云计算环境50。如图所示,云计算环境50包括云消费者使用的本地计算设备可以与其通信的一个或多个云计算节点10,本地计算设备诸如例如个人数字助理(PDA)或蜂窝电话54A、台式计算机54B、膝上型计算机54C和/或汽车计算机系统54N。云计算节点10可彼此通信。它们可以物理地或虚拟地分组(未示出)在一个或多个网络中,诸如如上所述的私有云、社区云、公共云或混合云、或其组合。这允许云计算环境50提供基础设施、平台和/或软件作为云消费者不需要为其维护本地计算设备上的资源的服务。应当理解,图1中所示的计算装置54A-N的类型仅旨在是说明性的,并且计算节点10和云计算环境50可通过任何类型的网络和/或网络可寻址连接(例如,使用网络浏览器)与任何类型的计算机化设备通信。

[0051] 现在参见图2,示出了由云计算环境50(图1)提供的一组功能抽象层。应提前理解,图2中所示的组件、层和功能仅旨在是说明性的,并且本发明的实施例不限于此。如图所示,提供以下层和对应功能:

[0052] 硬件和软件层60包括硬件和软件组件。硬件组件的示例包括:大型机61;基于RISC

(精简指令集计算机)架构的服务器62;服务器63;刀片服务器64;存储设备65;和网络和联网组件66。在一些实施例中,软件组件包括网络应用服务器软件67和数据库软件68。

[0053] 虚拟化层70提供抽象层,从该抽象层可以提供虚拟实体的以下示例:虚拟服务器71;虚拟存储器72;虚拟网络73,包括虚拟专用网络;虚拟应用和操作系统74;和虚拟客户端75。

[0054] 在一个示例中,管理层80可以提供下面描述的功能81-84。功能81-84可包括例如资源供应,其提供计算资源和用于在云计算环境内执行任务的其他资源的动态采购。计量和定价可以提供成本跟踪,因为在云计算环境内利用资源,并且为这些资源的消费开账单或发票。在一个示例中,这些资源可以包括应用软件许可证。安全性为云消费者和任务提供身份验证,以及为数据和其他资源提供保护。用户门户可以为消费者和系统管理员提供对云计算环境的访问。服务水平管理可以提供云计算资源分配和管理,使得满足所需的服务水平。服务水平协议(SLA)规划和履行可以为云计算资源提供预安排和采购,根据该SLA预期该云计算资源的未来要求。此外,功能85可以提供管理在工作负载层90中执行的容器的容器引擎。

[0055] 工作负载层90提供可以利用云计算环境的功能的示例。可以从该层提供的工作负荷和功能的示例包括:地图和导航91;软件开发和生命周期管理92;虚拟课堂教育交付93;数据分析处理94;交易处理95;和容器96。工作负载层90可以可选地托管可以由函数85提供的容器引擎管理的许多容器96。容器96中的每个容器可包括容器代码和数据。

[0056] 图3示出了根据本文描述的一个或多个实施例的示例非限制性计算环境的框图。计算环境的各个方面是硬件组件,而其他方面是软件组件。参考图3,用于实现本公开的各个方面的合适的操作环境300可以包括计算机310。计算机310可包括例如云计算节点,诸如图1中所示的云计算节点10中的一个。因此,计算机310可参与图1中所示的云计算环境50。如将理解的,在其他实施例中,计算机310可实现本公开的各方面而不参与图1和图2所示的云计算环境。

[0057] 计算机310可以包括处理单元311、系统存储器312和系统总线315。系统总线315将包括但不限于系统存储器312的系统组件耦合到处理单元311。处理单元311可以是各种可用处理器中的任何处理器。双微处理器和其他多处理器架构也可以被用作处理单元311。系统总线315可以是若干类型的总线结构中的任何一种,包括存储器总线或存储器控制器、外围总线或外部总线、和/或局部总线,其使用任何各种可用总线架构,包括但不限于工业标准架构(ISA)、微通道架构(MSA)、扩展ISA(EISA)、智能驱动电子器件(IDE)、VESA局部总线(VLB)、外围组件互连(PCI)、卡总线、通用串行总线(USB)、高级图形端口(AGP)、火线(IEEE1394)、以及小型计算机系统接口(SCSI)。

[0058] 系统存储器312还可以包括易失性存储器313和非易失性存储器314。包含用于诸如在启动期间在计算机310内的元件之间传输信息的基本例程的基本输入/输出系统(BIOS)被存储在非易失性存储器314中。作为示例而非限制,非易失性存储器314可包括只读存储器(ROM)、可编程ROM(PROM)、电可编程ROM(EPROM)、电可擦除可编程ROM(EEPROM)、闪存、或非易失性随机存取存储器(RAM)(例如,铁电RAM(FeRAM))。易失性存储器313还可以包括充当外部高速缓冲存储器的随机存取存储器(RAM)。作为说明而非限制,RAM可以以许多形式获得,诸如静态RAM(SRAM)、动态RAM(DRAM)、同步DRAM(SDRAM)、双数据速率SDRAM

(DDRSDRAM)、增强型SDRAM(ESDRAM)、Synchlink DRAM(SLDRAM)、直接Rambus RAM(DRRAM)、直接Rambus动态RAM(DRDRAM)和Rambus动态RAM。

[0059] 计算机310还可包括可移动/不可移动、易失性/非易失性计算机存储介质。例如,图3示出了磁盘存储器316。磁盘存储器316还可包括但不限于诸如磁盘驱动器、软盘驱动器、磁带驱动器、Jaz驱动器、Zip驱动器、LS-100驱动器、闪存卡或记忆棒之类的设备。磁盘存储器316还可包括单独的或与其他存储介质组合的存储介质,包括但不限于光盘驱动器,诸如致密盘ROM设备(CD-ROM)、CD可记录驱动器(CD-R驱动器)、CD可重写驱动器(CD-RW驱动器)或数字通用盘ROM驱动器(DVD-ROM)。为了便于磁盘存储器316与系统总线315的连接,通常使用可移动或不可移动接口,诸如接口315。

[0060] 在一些实施例中,加密的容器映像317可以由计算机310从容器注册表344检索并且存储在磁盘存储装置316中。因为加密的容器映像317被转移到计算机310,并且以加密格式存储在磁盘存储器316中,所以计算机310的恶意管理员访问加密的容器映像317的内容的风险很小。还应该注意的是,加密的容器映像317可以存储在系统存储器312而不是存储在磁盘存储器316中,或者除了存储在磁盘存储器316中还存储在系统存储器312。

[0061] 图3还示出了充当一方面的用户、应用和容器与另一方面的计算机310资源之间的中介的软件。这样的软件可以包括例如操作系统330。可存储在磁盘存储器316上的操作系统330用于控制和分配计算机310的资源。在一些实施例中,操作系统330可以是虚拟机(VM),并且计算机310可以托管多个VM。应当理解,本公开可用不同操作系统或操作系统的组合来实现。

[0062] 在引导操作系统330中可以涉及诸如引导组件370的一个或多个引导组件。引导组件370可以与可信处理模块(TPM)324交互以安全地引导操作系统330。一般而言,TPM 324将可信引导组件的哈希安全地存储在平台配置寄存器(PCR)325中。每个引导组件执行可信引导序列中的下一个引导组件的哈希,并使用TPM 324来针对PCR 325中的对应哈希检查所测量的哈希。可信引导序列中的下一个引导组件仅在其哈希正确时被加载,由此确保该引导组件是可信的,并且导致计算机310引导到可验证的安全状态。

[0063] 引导组件370可对计算机310的安全状态提出要求。例如,引导组件370可限制或阻止某些操作系统330功能。如本文将进一步描述的,引导组件370可以例如关闭对容器存储器(例如,为容器(诸如容器374)保留的系统存储器312的一部分)的一种或多种类型的管理性访问。

[0064] 此外,引导组件370可以确保可信核心服务组件371作为可信引导序列的一部分被加载。核心服务组件371可负责安全地获得解密密钥以解密加密的容器映像317或其部分。在一些实施例中,核心服务组件371可以利用上文介绍的TPM 324来安全地获得解密密钥。核心服务组件371可以向可信第三方服务设备345运用TPM 324远程认证,以获得解密密钥。

[0065] 系统应用331可以利用操作系统330通过例如存储在系统存储器312或盘存储316上的程序模块332和程序数据333对资源的管理。类似地,容器引擎372可在操作系统330上运行。容器引擎372可以例如装载、启动和/或停止诸如374、375和376的容器。容器引擎372可以隔离容器374、375和376,因此容器374、375和376不会相互干扰或访问彼此的数据。如本文进一步描述的,容器引擎372可任选地利用运行时解密组件(RDC)373提供进一步的容器安全性,该运行时解密组件可管理容器374的加密文件访问。运行时解密组件373可以利

用由可信核心服务组件371安全地获得的解密密钥。

[0066] 本公开的实施例可至少部分地通过以下各项的组合来防止管理员对容器374代码和数据的不必要窥探:(1)引导组件370强制实施计算机310的定义状态,在该状态下,对某些类型的管理性存储器访问被关闭,(2)核心服务组件371安全地获得用于容器374文件访问的解密密钥,以及(3)运行时解密组件373代表容器374利用解密密钥。在一些实施例中,引导组件370、核心服务组件371和/或运行时解密组件373可以是可被添加到计算机310并且具有单独的存储器和/或处理器组件的硬件组件。在其他实施例中,引导组件370、核心服务组件371和/或运行时解密组件373可以使计算机代码能够用指令来执行一个或多个不同操作以促进可信引导序列的执行。由于这些措施,计算机310的管理员不能访问系统存储器312中的未加密容器374代码和数据,并且计算机310的管理员不能获得解密密钥来解密容器374代码和数据。

[0067] 在一些实例中,用户可通过输入设备342将命令或信息输入到计算机310中。在其他实施例中,任何实体可将命令或信息输入到计算机310中。这些实体可以包括但不限于机器人、人工智能设备、计算机等。

[0068] 输入设备342包括但不限于诸如鼠标、跟踪球、指示笔、触摸板、键盘、麦克风、操纵杆、游戏板、圆盘式卫星天线、扫描仪、TV调谐器卡、数码相机、数码摄像机、网络相机等定点设备。这些和其他输入设备经由接口端口322通过系统总线315连接到处理单元311。接口端口322包括例如串行端口、并行端口、游戏端口和通用串行总线(USB)。输出设备341使用与输入设备342相同类型的端口中的一些端口。由此,例如,USB端口可以用于向计算机310提供输入,以及从计算机310向输出设备341输出信息。提供输出适配器321以说明除了其他输出设备341之外,还存在一些需要特殊适配器的输出设备341,例如监视器、扬声器和打印机。作为说明而非限制,输出适配器321包括提供输出设备341和系统总线315之间的连接装置的视频和声卡。应注意,其他设备和/或设备的系统提供输入和输出功能二者。

[0069] 计算机310可以使用到一个或多个远程计算机(诸如容器注册表344和可信第三方服务设备345)的逻辑连接在联网环境中操作。容器注册表344可以包括从其获得加密的容器映像317的服务器。可信第三方服务设备345可以包括例如由核心服务组件371结合TPM 324从其获取用于加密的容器映像317的解密密钥的服务器。计算机310可与之通信的其他远程计算机包括计算机、服务器、路由器、网络PC、工作站、基于微处理器的装置、对等设备或其他公共网络节点等,并且通常还可包括相对于计算机310描述的许多或所有元件。

[0070] 远程计算机(诸如344和345)可以通过网络接口343逻辑地连接到计算机310,然后经由通信连接323物理连接。网络接口343包括有线和/或无线通信网络,诸如局域网(LAN)、广域网(WAN)、蜂窝网络等。LAN技术包括光纤分布式数据接口(FDDI)、铜线分布式数据接口(CDDI)、以太网、令牌环等。WAN技术包括但不限于点对点链路、电路交换网络(如综合业务数字网(ISDN))及其变型、分组交换网络和数字用户线路(DSL)。通信连接323是指用于将网络接口343连接到系统总线315的硬件/软件。尽管为了清楚起见在计算机310内部示出了通信连接323,但它也可以在计算机310的外部。仅出于示例性目的,用于连接到网络接口343的硬件/软件还可包括内部和外部技术,诸如调制解调器,包括常规电话级调制解调器、电缆调制解调器和DSL调制解调器、ISDN适配器和以太网卡。

[0071] 图4A和4B示出了根据本文描述的一个或多个实施例的检索、存储和运行容器的示

例非限制性系统的框图。为了简洁起见,省略对本文中描述的其他实施例中采用的相似元件的重复描述。

[0072] 图4A示出了弱安全主机410,其与图4B中示出的安全强化主机450形成鲜明对比。在图4A和图4B两者中,相应的主机410、450可以从注册表400检索加密的容器映像401。注册表400包括多个加密的容器映像401、402、403和404。

[0073] 主机410和注册表400可提供例如图1所示的云计算节点10中的云计算节点。主机410可以是被配备成托管不具有(例如)安全功能(诸如图3中所示的引导组件370、核心服务组件371和运行时解密组件373)的容器的服务器。照此,主机410可安全地从注册表400检索加密的容器映像401。密钥获取组件412还可以获取解密密钥以对加密的容器映像401解密。

[0074] 一旦加密的容器映像401被解密并作为解密的容器映像401A被加载到系统存储器414中,解密的容器映像401A中的代码和数据可能不再被有效地保护。开发者放入加密的容器映像401中的代码、数据和配置被暴露给主机410的管理员,以显而易见的方式或通过少量工作便能查看。实际上,在一些实例中,解密的容器映像401A可以作为明文存储在主机410处。

[0075] 此外,在主机410中,可以用各种管理性访问功能416中的任何功能来访问系统存储器414内的解密的容器映像401A。例如,在例如不使用可信引导序列来引导主机410的情况下,对系统存储器414内的容器存储器的管理性访问,可以通过从修改后的内核的引导来完成。还可以通过除了可信内核的内核模块之外还加载另外的内核模块,来实现对系统存储器414内的容器存储器的管理性访问。

[0076] 在一些实例中,对系统存储器414内的容器存储器的管理性访问还可通过一个或多个虚拟存储器管理设备(例如使用/dev/mem、/dev/kmem和/proc/kcore虚拟存储器管理设备)来实现。对系统存储器414内的容器存储器的管理性访问也可以通过一个或多个运行时调试功能来实现。运行时调试功能通常是允许进程附接到另一进程的功能,并且可以例如允许根暂停运行进程并窥视它们的存储器。运行时调试功能的示例包括ptrace和扩展的Berkeley分组过滤器(eBPF)。

[0077] 在一些实例中,对系统存储器414内的容器存储器的管理性访问还可通过暂停运行进程以查看与运行进程相关联的存储器来实现。对系统存储器414内的容器存储器的管理性访问还可通过内核存储器交换操作来实现,其中内核将系统存储器414的部分写入到磁盘以释放系统存储器中的空间。可以通过将多余数据加载到系统存储器414中来故意触发内核存储器交换。

[0078] 总之,虽然主机410可以从注册表400安全地检索加密的容器映像401,但是在加密的容器映像401在主机410处被解密之后,加密的容器映像401可能易受恶意管理员的攻击。偶尔加密的容器映像(诸如401)可能包括各种各样的敏感信息(诸如专有代码和算法、患者/健康数据、机器学习(ML)模型等)中的任何敏感信息。如下所述,主机450实施一组安全强化功能,其改进在主机450处的加密的容器映像401的安全性以保护这样的敏感信息。

[0079] 在一些实施例中,主机450可以提供改进的安全性。如同主机410,主机450可以提供例如图1中示出的云计算节点10中的云计算节点,并且主机450可以包括被装备到主机容器的服务器。然而,与主机410不同的是,主机450可以实施例如图3中所示的引导组件370、核心服务组件371和运行时解密组件373之类的安全功能。照此,主机450可以安全地从注册

表400检索加密的容器映像401,获取用以解密加密的容器映像401的解密密钥,解密并运行加密的容器映像401。

[0080] 引导组件370可执行一个或多个操作以在主机450处执行可信引导序列的一部分,以将主机450安全地引导到定义的安全状态,在该安全状态下,对系统存储器454内的容器存储器的一种或多种类型的管理性访问被关闭。在图4B中,管理性访问功能416被关闭,从解密的容器映像401A指向管理性访问功能416的箭头被阻止。在实施例中,通过使用可信引导来防止内核代码、内核配置和核心服务被篡改,可以保护系统存储器454免受所有用户(包括主机450的管理员或根用户)的影响。此外,可以关闭允许具有特定权限集合的特定用户(例如,默认有权访问特定操作系统中的不同命令和文件的根用户)窥视到系统存储器454中的某些内核选项和虚拟设备。可替代地,可以使用安全硬件技术来保护系统存储器454。

[0081] 核心服务组件371可以作为可信引导序列的一部分被启动,并且可以实现图4B中所示的安全密钥获取452。安全密钥获取452可以安全地获得与加密的容器映像401一起使用的一个或多个解密密钥,而不允许主机450的特定用户访问解密密钥。例如,可以将所检索的解密密钥,存储在被保护免受管理性访问功能416的影响的系统存储器454中。

[0082] 在示例性实施例中,可以使用TPM 324和PCR 325来启动核心服务组件371,作为可信引导序列的一部分。当需要解密密钥时,核心服务组件371可以使用TPM 324来执行TPM远程认证,以从可信第三方服务设备345获取解密密钥。如果TPM远程认证成功,那么可信第三方服务设备345可以向安全密钥获取452提供一个或多个解密密钥。所获得的解密密钥可以保存在系统存储器454中,其中,由引导组件370实现的可信引导序列确保根用户不能从系统存储器454获取解密密钥。

[0083] 当容器在主机450处被启动时,例如,运行时解密组件373可使用安全密钥获取452所获取的解密密钥来解密容器映像,例如,来解密加密的容器映像401,从而将经解密的容器映像401A加载到系统存储器454中。运行时解密组件373还可执行对解密的容器镜像401A所访问的文件的运行时解密。

[0084] 在一些实施例中,运行时解密组件373可以至少部分以文件系统中介的形式来实现。当容器在主机450处被启动时,文件系统中介可以被放置在容器的根文件系统上,使得所有文件访问都通过文件系统中介。对于非加密文件,文件访问可以简单地通过文件系统中介。当访问加密文件时,可以检查请求进程的进程标识符(PID)。如果PID属于容器的入口点进程或入口点进程的后代进程,则可以解密文件。如果请求进程的PID不属于入口点进程或入口点进程的后代,则可以阻止解密。

[0085] 总之,关于图4B,主机450可以通过可单独或组合采用的若干安全强化措施来保护容器的解密的代码和数据。主机450被引导至安全状态以确保主机450内核是可信的,并且关闭否则就会使得能够对系统存储器454进行访问的各种管理性访问功能416中的任何功能。主机450可以包括安全密钥获取452,例如核心服务组件371,以便安全地检索并在诸如系统存储器454中的位置存储解密密钥,该位置的解密密钥不能被主机450的用户访问。主机450然后可以在容器运行时安全地解密容器数据。

[0086] 图5示出了根据本文所述的一个或多个实施例的采用修改的docker技术来保护容器的示例非限制性系统的框图。为了简洁起见,省略对本文中描述的其他实施例中采用的

相似元件的重复描述。

[0087] 图5所示的实施例包括用于容器运行时操作的经修改的基于docker的系统和方法500。docker组件501是docker守护程序,其提供容器引擎。为了运行包括在docker签名的容器映像522中的容器,docker组件501可以调用containerd组件502,其提供容器守护程序。如本文所使用的,术语“containerd组件”表示容器守护程序组件。docker组件501还启动docker高级多层统一文件系统(AUFS)组件521。在一些实施例中,可以用例如扩展文件属性或特殊文件名来注释docker签名的容器映像522,以指示docker签名的容器映像522受本文所公开的安全强化措施的约束。

[0088] containerd组件502启动容器垫片组件503,其为单个容器530提供运行时环境。在rootfs被容器存储驱动器挂载之后,containerd组件502可以对黑盒守护程序组件512进行远程过程调用(RPC)。

[0089] 在该示例中,黑盒守护程序组件512实现诸如结合图3描述的371的核心服务组件。黑盒守护程序组件512最初可以作为例如由PCR 325中的PCR#10保护的可信引导序列的一部分而启动。在启动时,黑盒守护程序组件512可以执行TPM远程认证,之后,黑盒守护程序组件512可以例如从可信第三方服务设备345检索密封的解密密钥的列表。黑盒守护程序组件512可以解封存储器中的所检索的解密密钥。黑盒守护程序组件512可以在完成时调用TPM_PCR_Extend PCR#10。

[0090] 在容器运行时,响应于来自containerd组件502的RPC调用,并且如果docker签名的容器映像被签名,则黑盒守护程序组件512例如可以发起入口点进程,并按需开始代表容器530对文件进行解密,同时将明文或以其他方式解密的文件仅存储在被保护免受诸如根用户(或默认有权访问操作系统中各种命令或文件的其他用户)之类的用户影响的存储器中。

[0091] 此外,黑盒守护程序组件512可以启动用户空间中的黑盒文件系统(FUSE)组件514,以利用用于容器530的解密密钥启动FUSE进程。黑盒FUSE组件514提供了图5的实施例中的运行时解密组件373。黑盒FUSE组件514作为中间文件系统层位于容器530的rootfs(即docker AUFS组件521)之上。如果请求进程是容器530的入口点进程(其在一些情况下也可以是init进程)或容器530的入口点(或init)进程的后代进程,则黑盒FUSE组件514对加密的文件进行解密。黑盒FUSE组件514还可以为请求进程检索任何未加密文件。

[0092] 图6示出了根据本文所述的一个或多个实施例的示例非限制性引导组件的高级框图。为了简洁起见,省略对本文中描述的其他实施例中采用的相似元件的重复描述。

[0093] 引导组件600的各方面可构成体现在机器内(例如,体现在与一个或多个机器相关联的一个或多个计算机可读介质中)的机器可执行组件。当由一个或多个机器(例如计算机、计算设备、虚拟机等)执行时,这样的组件可以使机器执行所述的操作。一方面,存储器608可以存储计算机可执行组件和指令。此外,处理器604可以促进与引导组件600相关联的指令(例如,计算机可执行组件和指令)的操作。

[0094] 如图6所示,在一些实施例中,启动组件600可以包括处理器604、TPM606和存储器608。在其他实施例中,引导组件600k可以不是具有单独的处理器604,而是可以访问核心服务组件700的处理器704(图7)、运行时解密组件800的处理器804(图8)和/或处理单元311(图3)中的一个或多个处理器和/或与其共享处理器。相应地,在所示实施例中,引导组件

600是硬件,但是在一些实施例中,引导组件600可以计算机代码的形式实现,或者引导组件600的一个或多个操作可以通过计算机代码来实现。

[0095] TPM 606可以例如在PCR 325的PCR中包括定义的安全状态610。存储器608可以包括例如下一组件612且可选地包括定义的安全状态610。在非限制性示例中,图3的元件在图6中实现。引导组件600提供结合到计算机310中的引导组件370的示例。同样地,处理器604可提供处理器311,存储器608可提供系统存储器312或包括在计算机310中的其他存储器,并且TPM 606可提供图3的TPM 324。

[0096] 在某些实现方式中,如由引导组件600下面的箭头所指示的,引导组件600可以被用来执行被标注为系统的可信引导序列的操作序列的一部分。引导组件600可以执行可信引导序列的至少一部分,并且引导组件600然后可以使下一组件612作为可信引导序列的一部分来操作或执行。可信引导序列的结果是,这种操作序列可以根据定义的安全状态610安全地将系统引导到定义的安全状态,其中关闭对容器存储器的一种或多种类型的管理性访问。

[0097] 在非限制性示例中,引导组件600可以检查下一组件612以确保下一组件612满足所定义的安全状态610。定义的安全状态610可包括其中关闭对容器存储器的一种或多种类型的管理性访问的状态。即,在根据定义的安全状态610完全地引导计算机310之后,可以关闭通常提供对特定系统区段(例如,系统存储器)的管理性访问的某些功能。

[0098] 例如,在一实施例中,下一组件612可包括操作系统内核。定义的安全状态610可以包括内核的哈希,其中,与某些类型的管理性存储器访问相关联的管理性访问功能被关闭。引导组件600可执行下一组件612或其部分的哈希,并且引导组件600可检查所测量的哈希与存储在所定义的安全状态610中的哈希相匹配。引导组件600可以在哈希匹配的情况下加载下一组件612,或者在哈希不匹配的情况下停止可信引导序列。

[0099] TPM 606可例如在PCR 325的PCR中安全地存储下一组件612的哈希。引导组件600可测量下一组件612的哈希并将该哈希值传递给TPM 606。当哈希值匹配时,TPM 606可以通知引导组件600可以安全地加载下一组件612,于是引导组件600可以加载下一组件612。当哈希不匹配时,TPM 606可通知引导组件600,于是引导组件600可以停止可信引导序列。

[0100] 在一些实例中,诸如结合图7所描述的核心服务部件可以包括定义的安全状态610的至少一个方面。在这样的情况下,引导组件600可以保证被关闭的管理性访问功能以及引导系统中核心服务组件的存在。引导组件600可执行核心服务组件、操作系统内核或任何其他下一组件612或其组合的哈希。

[0101] 在其他实施例中,可以实现指定定义的安全状态610的其他方案。例如,定义的安全状态610可包括将在下一组件612中关闭的管理性访问功能的列表。引导组件600可以从存储器608读取定义的安全状态610,并且引导组件600可以检查下一组件612以确保所列出的管理性访问功能被关闭。如果所列出的管理性访问功能关闭,则引导组件600可以加载下一组件612。如果所列出的管理性访问功能未关闭,则引导组件600可以停止可信引导序列。

[0102] 图7示出根据本文描述的一个或多个实施例的示例非限制性核心服务组件的高级框图。为了简洁起见,省略对本文描述的其他实施例中采用的相似元件的重复描述。

[0103] 核心服务组件700的各方面可构成体现在机器内(例如,体现在与一个或多个机器相关联的一个或多个计算机可读介质中)的机器可执行组件。当由一个或多个机器(例如计

算机、计算设备、虚拟机等)执行时,这样的组件可以使机器执行所述的操作。一方面,存储器708可以存储计算机可执行组件和指令。此外,处理器704可以促进与核心服务组件700相关联的指令(例如,计算机可执行组件和指令)的操作。

[0104] 如图7所示,在一些实施例中,核心服务组件700可包括处理器704、TPM 706、存储器708和通信组件712。在其他实施例中,核心服务组件700k可以不是具有单独的处理器704,而是可以访问引导组件600的处理器604(图6)、运行时解密组件800的处理器804(图8)和/或处理单元311(图3)中的一个或多个和/或与其共享处理器。相应地,在所示实施例中,引导组件700是硬件,但是在一些实施例中,引导组件700可以计算机代码的形式实现,或者引导组件700的一个或多个操作可以通过计算机代码来实现。

[0105] TPM 706可以包括密封的秘密710。存储器708可以包括例如解密密钥720。核心服务组件700可以可选地作为上文参考引导组件600描述的可信引导序列的一部分被启动。核心服务部件700可以安全地获得一个或多个解密密钥720以与容器存储器一起使用。在非限制性示例中,图3的元件在图7中实现。核心服务部件700是图3的核心服务部件371的实施例的示例。

[0106] 在一些示例中,核心服务部件700可以在计算机310启动时获得解密密钥720的包以用于与在计算机310处托管的多个容器一起使用。核心服务组件700能例如识别存储在计算机310处的多个容器映像,并且核心服务组件700可采用通信组件712和TPM 706来向可信第三方服务设备345执行TPM 706远程认证。TPM 706可以例如向可信第三方服务设备345提供密封的秘密710。可信第三方服务设备345可以使用密封的秘密710来认证计算机310。可信第三方服务设备345然后可以向核心服务组件700返回例如一组所请求的解密密钥。核心服务组件700可将所返回的解密密钥作为解密密钥720存储在存储器708中。然后,所述解密密钥可以视情况而定用于解密相应容器的容器数据。

[0107] 在一些示例中,核心服务组件700可在容器运行时获得各个容器的解密密钥720。如本文所述,当在计算机310处加载容器时,核心服务组件700可以采用通信组件712和TPM 706来向可信第三方服务设备345执行TPM706远程认证。可信第三方服务设备345然后可以例如向核心服务组件700返回所请求的解密密钥。核心服务组件700可将所返回的解密密钥作为解密密钥720的解密密钥存储在存储器708中。然后,可以使用解密密钥来对容器数据进行解密。因为解密密钥720被加载到存储器708中,并且存储器708被引导组件600保护以免于管理员查看,所以解密密钥720保持安全,在没有授权的情况下,不可被访问以用于解密容器数据。

[0108] 图8示出了根据本文描述的一个或多个实施例的示例非限制性运行时解密组件的高级框图。为了简洁起见,省略对在此描述的其他实施例中采用的相似元件的重复描述。

[0109] 运行时解密组件800的各方面可构成体现在机器内(例如,体现在与一个或多个机器相关联的一个或多个计算机可读介质中)的机器可执行组件。当由一个或多个机器(例如计算机、计算设备、虚拟机等)执行时,这样的组件可以使机器执行所述的操作。一方面,存储器808可以存储计算机可执行组件和指令。此外,处理器804可以促进与运行时解密组件800相关联的指令(例如,计算机可执行组件和指令)的操作。

[0110] 如图8所示,运行时解密组件800可以包括处理器804、检查PID组件802和存储器808。如图8所示,在一些实施例中,运行时解密组件800可以包括处理器804、检查PID组件

802和存储器808。相应地，在所示实施例中，运行时解密组件800是硬件，但在一些实施例中，运行时解密组件800可被实现为计算机代码，或运行时解密组件800的一个或多个操作可经由计算机代码来实现。在其他实施例中，运行时解密组件800可以不具有单独的处理器804，而是可以访问引导组件600的处理器604(图6)、核心服务组件的处理器704(图7)和/或处理单元311(图3)中的一个或多个和/或与其共享处理器。

[0111] 存储器808可以包括例如解密密钥820。运行时解密组件800可以使用解密密钥820(例如，由核心服务组件700检索的解密密钥)来执行由与系统存储器312内的容器存储器相关联的容器374访问的一个或多个文件的运行时解密。在非限制性示例中，图3的元件在图8中实现。运行时解密组件800可以提供结合到计算机310中的运行时解密组件373的示例。同样地，处理器804可以提供处理器311，存储器808可以提供系统存储器312或包括在计算机310中的其他存储器。

[0112] 在一些示例中，运行时解密组件800可以从容器374接收文件请求。文件请求可包括例如加密文件请求831和非加密文件请求835。加密文件请求831是对加密文件的请求，非加密文件请求835是对非加密文件的请求。运行时解密组件800可以不同地处理请求831、835，然而运行时解密组件800可以最终例如以对应于加密文件请求831的返回832的形式以及对应于非加密文件请求836的返回836的形式，返回所请求的文件。

[0113] 对于加密文件请求831，运行时解密组件800可以采用检查PID组件802来检查请求进程的进程标识符(PID)。检查PID组件802可以确保请求进程的PID属于容器374的入口点进程，或入口点进程的后代进程。如果PID确实属于入口点进程或后代，则运行时解密组件800可以继续从文件系统850检索所请求的文件，使用解密密钥820解密所请求的文件，并返回832所请求的文件。如果PID未被授权，则运行时解密组件800可以抛出错误。

[0114] 对于非加密文件请求835，在实施例中，运行时解密组件800可以将请求835传递到文件系统850。文件系统850可将所请求的非加密文件返回到运行时解密组件800，运行时解密组件800可在返回836处将所请求的文件返回到请求容器。

[0115] 虽然图6、图7和图8分别描绘了单独的组件，即引导组件600、核心服务组件700和运行时解密组件800，但是应当理解，可以在公共组件中实现两个或更多个组件。进一步，应当理解，引导组件600、核心服务组件700和运行时解密组件800的设计可以包括其他组件选择、组件放置等以促进增强的容器安全性。此外，已经关于若干组件之间的交互描述了上述系统和/或设备。应当理解，这样的系统和组件可以包括其中指定的那些组件或子组件、所指定的组件或子组件中的一些、和/或附加的组件。子组件还可以被实现为通信地耦合到其他组件的组件，而不是被包括在父组件内。此外，可以将一个或多个组件和/或子组件组合成提供聚合功能的单个组件。这些组件还可以与为简要起见未在此具体描述、但本领域的技术人员已知的一个或多个其他组件交互。

[0116] 图9示出根据本文描述的一个或多个实施例的促进托管容器的系统的安全强化的示例非限制性计算机实现方法900的流程图。参照图3中所示的计算机310来描述图9的方法。图9的计算机实现的方法，通过在容器服务器(诸如计算机310)处的容器374的实例化和运行时期间保护与容器374相关联的数据免受对容器存储器的一种或多种类型的管理性访问，增强了容器服务器(诸如计算机310)处的容器(诸如374)的安全性。

[0117] 在902，操作地耦合到处理器311的引导组件370执行可信引导序列的至少一部分

以将计算机310安全地引导到关闭对容器存储器(系统存储器312的容器存储器部分)的一种或多种类型的管理性访问的定义的安全状态。被关闭的对容器存储器的管理性访问的示例类型包括但不限于:通过来自经修改的内核的引导对容器存储器的管理性访问,所述经修改的内核相对于与所述可信引导序列相关联的可信内核进行了修改;通过加载除了可信内核的内核模块之外的附加内核模块对容器存储器的管理性访问;通过一个或多个虚拟存储器管理设备对容器存储器的管理性访问;通过一个或多个运行时调试功能来对容器存储器的管理性访问;通过暂停运行进程以查看与运行进程相关联的存储器对容器存储器的管理性访问;和通过内核存储器交换操作对容器存储器的管理性访问。在实施例中,可信引导序列可包括对存储在核心服务组件371处或可由核心服务组件371访问的计算机代码执行哈希,以及在PCR 325的PCR中存储所得到的核心服务组件371哈希值。

[0118] 在904,引导组件370作为可信引导序列的一部分而启动操作性地耦合到处理器311的核心服务组件371。在906,在实施例中,核心服务组件371安全地获得用以与系统存储器312的容器存储器部分一起使用的一个或多个解密密钥。在实施例中,安全地获得一个或多个解密密钥可以向可信第三方服务设备345运用TPM 324远程认证。

[0119] 在908,例如由运行时解密组件373确定容器374所请求的文件是否是加密的。如果是,则在910,运行时解密组件373可检查请求加密文件的一个或多个进程的进程标识符(PID),以确保PID属于容器374的入口点进程,或入口点进程的后代进程。在912,运行时解密组件373可以用由核心服务组件371获得的一个或多个解密密钥来执行对容器374访问的一个或多个文件的运行时解密,该容器374是与系统存储器312的容器存储器部分相关联的。在一些示例中,运行时解密组件373初始可以采用该一个或多个解密密钥对用于实例化容器374的加密的容器映像进行解密。

[0120] 响应于运行时解密组件373在904确定容器374所请求的文件未被加密,运行时解密组件373可在914传递对非加密文件的一个或多个请求。

[0121] 图10示出了根据本文描述的一个或多个实施例的促进托管容器的系统的安全强化的另一示例非限制性计算机实现方法1000的流程图。图10的方法是参考图5中示出的基于docker的实施例来描述的。在1002,操作地耦合到处理器(图5中未示出)的docker组件501可以实例化一个或多个容器,诸如容器530。在实施例中,由docker组件501实例化一个或多个容器530可以包括:实例化操作地耦合到处理器的至少一个第一容器守护程序组件502,由第一容器守护程序组件502实例化运行时解密组件(诸如黑盒FUSE组件514),以及由黑盒FUSE组件514实例化一个或多个容器530中的至少一个。在一些实例中,黑盒守护程序组件512可以实例化容器530。

[0122] 在1004,针对该一个或多个容器530中的至少一个,诸如黑盒FUSE组件514的运行时解密组件可以管理该一个或多个容器530中的至少一个的一个或多个文件访问。在1008,黑盒FUSE组件514可以确定所请求的文件是否是加密的。如果所请求的文件是加密的,则黑盒FUSE组件514包括文件系统中介,文件系统中介可以在1010检查请求访问加密文件的一个或多个进程的PID以确保PID属于容器530的入口点进程或入口点进程的后代进程。如果PID被授权,则在1012,黑盒FUSE组件514可以检索、解密和返回所请求的文件。例如,如参考图5所描述的,黑盒守护程序组件512可以在用户空间(FUSE)组件514中启动黑盒文件系统,以利用用于容器530的解密密钥启动FUSE进程。黑盒FUSE组件514提供图5的实施例中的运

行时解密组件373。黑盒FUSE组件514作为中间文件系统层位于容器530(即docker AUFS组件521)的rootfs之上。如果请求进程是容器530的入口点进程(其在一些情况下也可以是init进程)或容器530的入口点(或init)进程的后代进程,则黑盒FUSE组件514对加密文件进行解密。

[0123] 黑盒FUSE组件514还可以为请求过程检索任何未加密文件。如果所请求的文件未被加密,则在1014,黑盒FUSE组件514可以向docker AUFS组件521传递一个或多个访问未加密文件的请求。

[0124] 图11示出根据本文描述的一个或多个实施例的促进托管容器的系统的安全强化的另一示例非限制性计算机实现的方法1100的流程图。参考图3描述图11的方法。

[0125] 在1102,操作地耦合到处理器311的引导组件370可以执行可信引导序列的至少一部分以安全地引导计算机310。可信引导序列可以包括例如在1104重置一个或多个PCR 325,以及在1106运行一系列引导组件,其中该系列中的每个引导组件执行在该系列中的下一个引导组件处存储的或可由其访问的计算机代码的哈希,并存储对应的PCR哈希值和/或使用TPM 324来将该下一个引导组件哈希值与存储的PCR哈希值进行比较。可信引导序列将计算机310安全地引导到其中关闭对系统存储器312的容器存储器部分的一种或多种类型的管理性访问的定义的安全状态。如本文中所述,该系列引导组件中的至少一个引导组件可以包括核心服务组件371,或者核心服务组件371可以包括在可信引导序列结束时加载的可信内核中。

[0126] 在1108,操作地耦合到处理器311的核心服务组件371可以向可信第三方服务设备345执行TPM 324远程认证。在1110,核心服务组件371可从可信第三方服务设备345安全地获得容器374的一个或多个解密密钥,核心服务组件371可将容器374的一个或多个解密密钥存储在与容器374相关联的容器存储器中。

[0127] 在1112,操作地耦合到处理器311的运行时解密组件373可至少部分地通过检查请求访问加密文件的一个或多个进程的PID以确保PID属于容器374的入口点进程或入口点进程的后代进程来管理容器374的一个或多个文件访问,其中,用在1110获得的一个或多个解密密钥来解密加密文件。

[0128] 为了说明的简要,将计算机实现的方法描绘和描述为一系列动作。应该理解和认识到,本主题创新不受所例示的动作和/或动作的顺序的限制,例如动作可以按不同的顺序发生和/或同时发生,且可与在此未呈现和描述的其他动作一起发生。此外,并非所有示出的动作都是实现根据所公开的主题的计算机实现的方法所必需的。此外,本领域技术人员将理解和领会,计算机实现的方法可替代地通过状态图或事件而表示为一系列相互关联的状态。此外,还应当理解,本说明书通篇公开的计算机实现的方法能够被存储在制品上以便于传送和将这样的计算机实现的方法传输到计算机。本文中所使用的术语制品,旨在涵盖可从任何计算机可读设备或存储介质访问的计算机程序。

[0129] 此外,因为数据分组的配置和/或组件之间的通信是从电气和机械组件和电路的组合建立的,所以人类无法复制或执行主题数据分组配置和/或处理组件和/或分配组件之间的主题通信。例如,人类无法测量引导过程中的下一组件的哈希,或解密加密文件等。此外,人类无法对可能包括对应于在各种容器安全过程期间生成的信息的比特序列的数据进行分组,或者不能发送可能包括对应于在本文所述的安全过程期间生成的信息的比特序列

的数据等。

[0130] 本发明可以是在任何可能的技术细节集成度上的系统、方法、装置和/或计算机程序产品。计算机程序产品可包括其上具有用于使处理器执行本发明的各方面的计算机可读程序指令的计算机可读存储介质。计算机可读存储介质可以为可保留和存储供指令执行设备使用的指令的有形设备。计算机可读存储介质可以是例如但不限于电子存储设备、磁存储设备、光存储设备、电磁存储设备、半导体存储设备、或者上述的任意合适的组合。计算机可读存储介质的更具体示例的非穷尽列表还可以包括以下各项:便携式计算机盘、硬盘、随机存取存储器(RAM)、只读存储器(ROM)、可擦式可编程只读存储器(EPROM或闪存)、静态随机存取存储器(SRAM)、便携式紧凑盘只读存储器(CD-ROM)、数字通用盘(DVD)、记忆棒、软盘、诸如穿孔卡或具有记录在其上的指令的槽中的凸出结构之类的机械编码设备、以及上述各项的任何合适的组合。如本文所使用的计算机可读存储介质不应被解释为暂时性信号本身,例如无线电波或其他自由传播的电磁波、通过波导或其他传输介质传播的电磁波(例如,穿过光纤电缆的光脉冲)或通过电线发射的电信号。

[0131] 本文所述的计算机可读程序指令可以经由网络(例如,互联网、局域网、广域网和/或无线网络)从计算机可读存储介质下载到相应的计算/处理设备,或者下载到外部计算机或外部存储设备。网络可以包括铜传输电缆、光传输纤维、无线传输、路由器、防火墙、交换机、网关计算机和/或边缘服务器。每个计算/处理设备中的网络适配器卡或网络接口接收来自网络的计算机可读程序指令,并转发计算机可读程序指令以存储在相应计算/处理设备内的计算机可读存储介质中。用于执行本发明的操作的计算机可读程序指令可以是汇编指令、指令集架构(ISA)指令、机器指令、机器相关指令、微代码、固件指令、状态设置数据、集成电路的配置数据、或以一种或多种程序设计语言的任何组合编写的源代码或目标代码,这些程序设计语言包括面向对象的程序设计语言(诸如Smalltalk、C++等)和过程程序设计语言(诸如“C”程序设计语言或类似程序设计语言)。计算机可读程序指令可以完全地在用户计算机上执行、部分在用户计算机上执行、作为独立软件包执行、部分在用户计算机上部分在远程计算机上执行或者完全在远程计算机或服务器上执行。在后一种情况下,远程计算机可通过任何类型的网络(包括局域网(LAN)或广域网(WAN))连接至用户计算机,或者可连接至外部计算机(例如,使用互联网服务提供商通过互联网)。在一些实施例中,包括例如可编程逻辑电路、现场可编程门阵列(FPGA)或可编程逻辑阵列(PLA)的电子电路可以通过利用计算机可读程序指令的状态信息来使电子电路个性化来执行计算机可读程序指令,以执行本发明的各方面。

[0132] 参照根据本发明实施例的方法、装置(系统)和计算机程序产品的流程图和/或框图描述了本发明。应当理解,流程图和/或框图的每个方框以及流程图和/或框图中各方框的组合,都可以由计算机可读程序指令实现。这些计算机可读程序指令可被提供给通用计算机、专用计算机或其他可编程数据处理装置的处理器以产生机器,使得经由计算机或其他可编程数据处理装置的处理器执行的指令创建用于实现在流程图和/或框图的或多个框中指定的功能/动作的装置。也可以把这些计算机可读程序指令存储在计算机可读存储介质中,这些指令使得计算机、可编程数据处理装置、和/或其他设备以特定方式工作,从而,其中存储有指令的计算机可读存储介质包括包含实现流程图和/或框图中的或多个方框中规定的功能/动作的方面的指令的制造品。也可以把计算机可读程序指令加载到计算机、其

他可编程数据处理装置、或其他设备上，使得在计算机、其他可编程装置或其他设备上执行一系列操作动作，以产生计算机实现的处理，使得在计算机、其他可编程装置或其他设备上执行的指令实现在流程图和/或框图的或多个框中指定的功能/动作。

[0133] 附图中的流程图和框图示出了根据本发明的不同实施例的系统、方法和计算机程序产品的可能实现方式的架构、功能和操作。对此，流程图或框图中的每个框可表示指令的模块、段或部分，其包括用于实现指定的逻辑功能的一个或多个可执行指令。在一些备选实现中，框中标注的功能可以不按照图中标注的顺序发生。例如，取决于所涉及的功能，连续示出的两个框实际上可以基本上同时执行，或者这些框有时可以以相反的顺序执行。也要注意的是，框图和/或流程图中的每个方框、以及框图和/或流程图中的方框的组合，可以用执行规定的功能或动作或执行专用硬件与计算机指令的组合的专用的基于硬件的系统来实现。

[0134] 虽然上文已经在运行在计算机和/或计算机上的计算机程序产品的计算机可执行指令的一般上下文中描述了主题，但本领域技术人员将认识到，本公开还可以其他程序模块或与其他程序模块组合地实现。通常，程序模块包括执行特定任务和/或实现特定抽象数据类型的例程、程序、组件、数据结构等。此外，本领域的技术人员将认识到，本发明的计算机实现的方法可以用其他计算机系统配置来实践，包括单处理器或多处理器计算机系统、小型计算设备、大型计算机、以及计算机、手持式计算设备(例如，PDA、电话)、基于微处理器或可编程的消费者或工业电子产品等。所展示的各方面还可以在分布式计算环境中实现，其中，任务由通过通信网络链接的远程处理设备来执行。然而，本发明的一些(如果不是全部的话)方面可在独立计算机上实践。在分布式计算环境中，程序模块可以位于本地和远程存储器存储设备两者中。

[0135] 如在本申请中所使用的术语“组件”、“系统”、“平台”、“接口”等，可以指和/或可以包括计算机相关实体或与具有一个或多个特定功能的操作机器相关的实体。本文公开的实体可以是硬件、硬件和软件的组合、软件或执行中的软件。例如，组件可以是但不限于在处理器上运行的进程、处理器、对象、可执行文件、执行线程、程序和/或计算机。作为示例，在服务器上运行的应用和服务器两者都可以是组件。一个或多个组件可以驻留在进程和/或执行的线程内，并且组件可以位于一个计算机上和/或分布在两个或更多个计算机之间。在另一实例中，相应组件可从具有存储于其上的不同数据结构的不同计算机可读介质执行。组件可以经由本地和/或远程进程通信，诸如根据具有一个或多个数据分组的信号(例如，来自与本地系统、分布式系统中的另一组件进行交互的一个组件的数据，和/或经由该信号跨诸如互联网之类的网络与其他系统进行交互的一个组件的数据)。作为另一示例，组件可以是具有由电气或电子电路操作的机械部件提供的特定功能的装置，该电气或电子电路由处理器执行的软件或固件应用操作。在这样的情况下，处理器可以在装置的内部或外部，并且可以执行软件或固件应用的至少一部分。作为又一示例，组件可以是通过没有机械部件的电子组件来提供特定功能的装置，其中电子组件可以包括处理器或用于执行至少部分地赋予电子组件的功能的软件或固件的其他装置。在一方面中，组件可经由例如云计算系统内的虚拟机来仿真电子组件。

[0136] 此外，术语“或”旨在意指包括性的“或”而不是排他性的“或”。也就是说，除非另外指明，或者上下文清楚，“X采用A或B”旨在意指任何自然的包含性排列。即，如果X采用A；X采

用B;或X采用A和B两者,则在任何前述情况下满足“X采用A或B”。此外,如主题说明书和附图中所使用的冠词“一个”和“一种”通常应被解释为意指“一个或多个”,除非另外说明或上下文指明为单数形式。如本文所使用的术语“实例”和/或“示例性”,用于表示用作实例、例子或例证。为了避免疑问,本文公开的主题不受此类实例的限制。此外,本文中描述为“实例”和/或“示例性”的任何方面或设计,不一定被解释为优于或优于其他方面或设计,也不意味着排除本领域普通技术人员已知的等效的示例性结构和技术。

[0137] 如在本说明书中所采用的术语“处理器”,可以指基本上任何计算处理单元或设备,包括但不限于单核处理器;具有软件多线程执行能力的单处理器;多核处理器;具有软件多线程执行能力的多核处理器;具有硬件多线程技术的多核处理器;并行平台;和具有分布式共享存储器的并行平台。另外,处理器可指代经设计以执行本文中所描述的功能的集成电路、专用集成电路(ASIC)、数字信号处理器(DSP)、现场可编程门阵列(FPGA)、可编程逻辑控制器(PLC)、复杂可编程逻辑装置(CPLD)、离散门或晶体管逻辑、离散硬件组件或其任何组合。进一步,处理器可以利用纳米级架构,诸如但不限于基于分子和量子点的晶体管、开关和门,以优化空间使用或增强用户设备的性能。处理器还可以被实现为计算处理单元的组合。在本公开中,诸如与组件的操作和功能相关的“储存”、“存储”、“数据储存”、“数据存储”、“数据库”和基本上任何其他信息存储组件的术语用于指“存储器组件”、“体现在“存储器”中的实体、或包括存储器的组件。应当理解,本文所述的存储器和/或存储器部件可以是易失性存储器或非易失性存储器,或者可以包括易失性存储器和非易失性存储器两者。作为示例而非限制,非易失性存储器可包括只读存储器(ROM)、可编程ROM(PROM)、电可编程ROM(EPROM)、电可擦除ROM(EEPROM)、闪存、或非易失性随机存取存储器(RAM)(例如,铁电RAM(FeRAM))。易失性存储器可包括例如可充当外部高速缓冲存储器的RAM。作为示例而非限制,RAM可以以许多形式获得,诸如同步RAM(SRAM)、动态RAM(DRAM)、同步DRAM(SDRAM)、双数据速率SDRAM(DDRSDRAM)、增强SDRAM(ESDRAM)、SynchlinkDRAM(SLDRAM)、直接RambusRAM(DRRAM)、直接Rambus动态RAM(DRDRAM)和Rambus动态RAM(RDRAM)。另外,本文所披露的系统或计算机实施的方法的存储器组件旨在包括(但不限于包括)这些和任何其他合适类型的存储器。

[0138] 以上已经描述的内容仅包括系统和计算机实施的方法的示例。当然,为了描述本公开的目的,不可能描述组件的每个可想象的组合或计算机实现的方法,但是本领域普通技术人员可以认识到,本公开的许多进一步的组合和置换是可能的。此外,就详细说明、权利要求、附录以及附图中使用术语“包括”、“具有”、“拥有”等的程度而言,这些术语旨在以类似于术语“包含”在权利要求中作为过渡词采用时所作解释的方式是包括性的。已经出于说明的目的呈现了不同实施例的描述,但并不旨在是详尽的或限于所公开的实施例。在不脱离所描述的实施例的范围的情况下,许多修改和变化对于本领域普通技术人员来说是显而易见的。这里使用的术语被选择来最好地解释实施例的原理、实际应用或对在市场中找到的技术的技术改进,或者使得本领域普通技术人员能够理解这里公开的实施例。

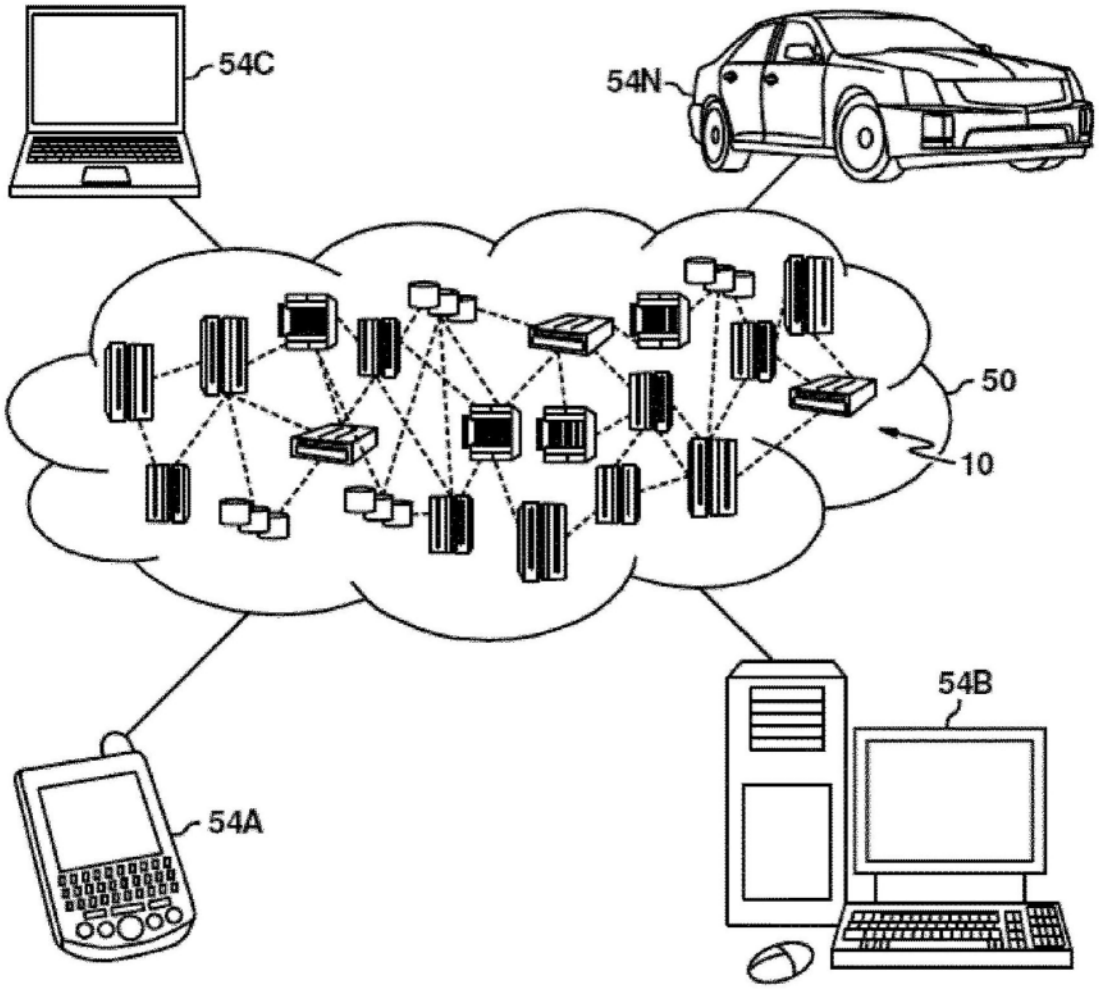


图1

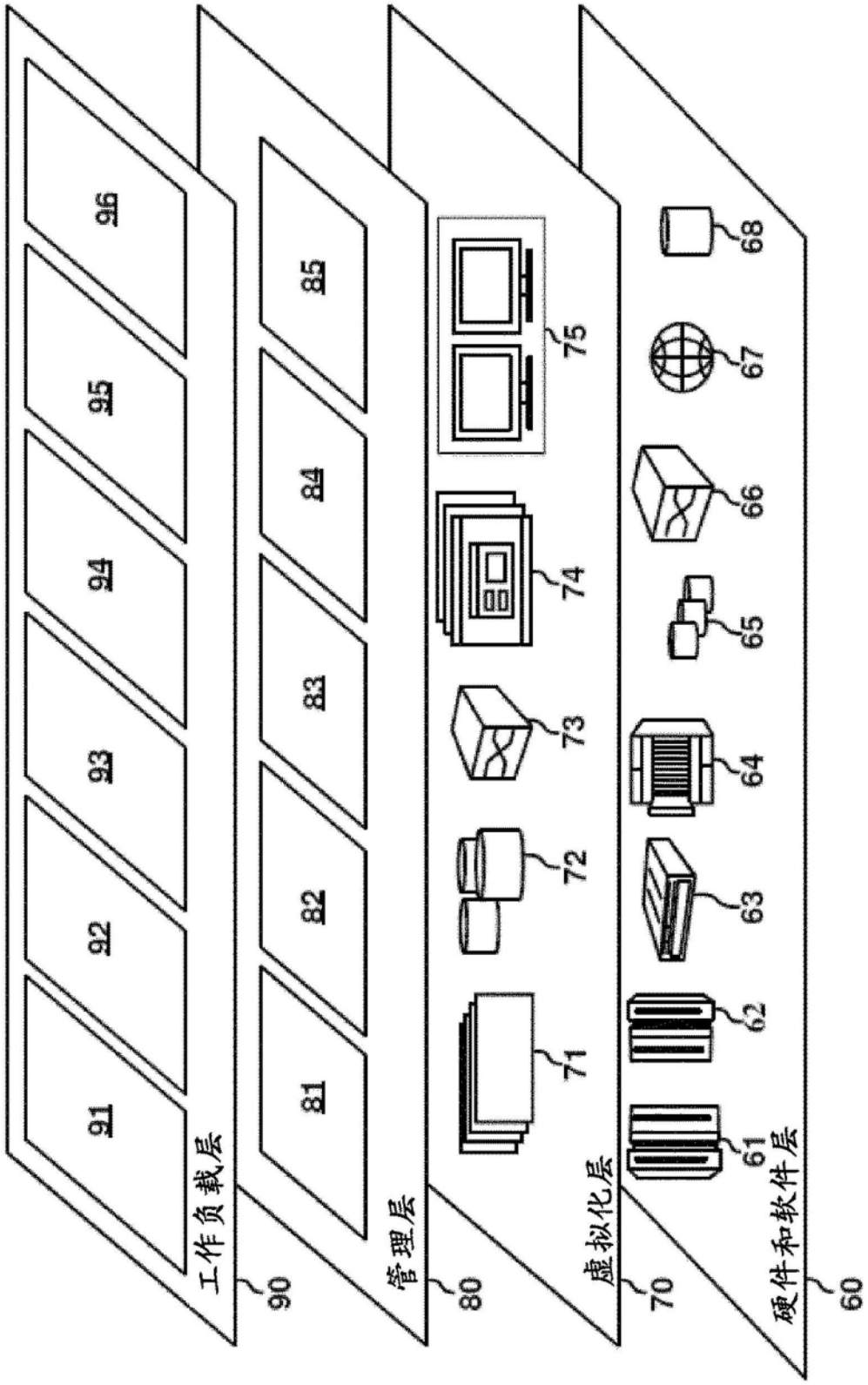

工作负载层
90
91
92
93
94
95
96
管理层
80
81
82
83
84
85
虚拟化层
70
71
72
73
74
75
硬件和软件层
60
61
62
63
64
65
66
67
68

图2

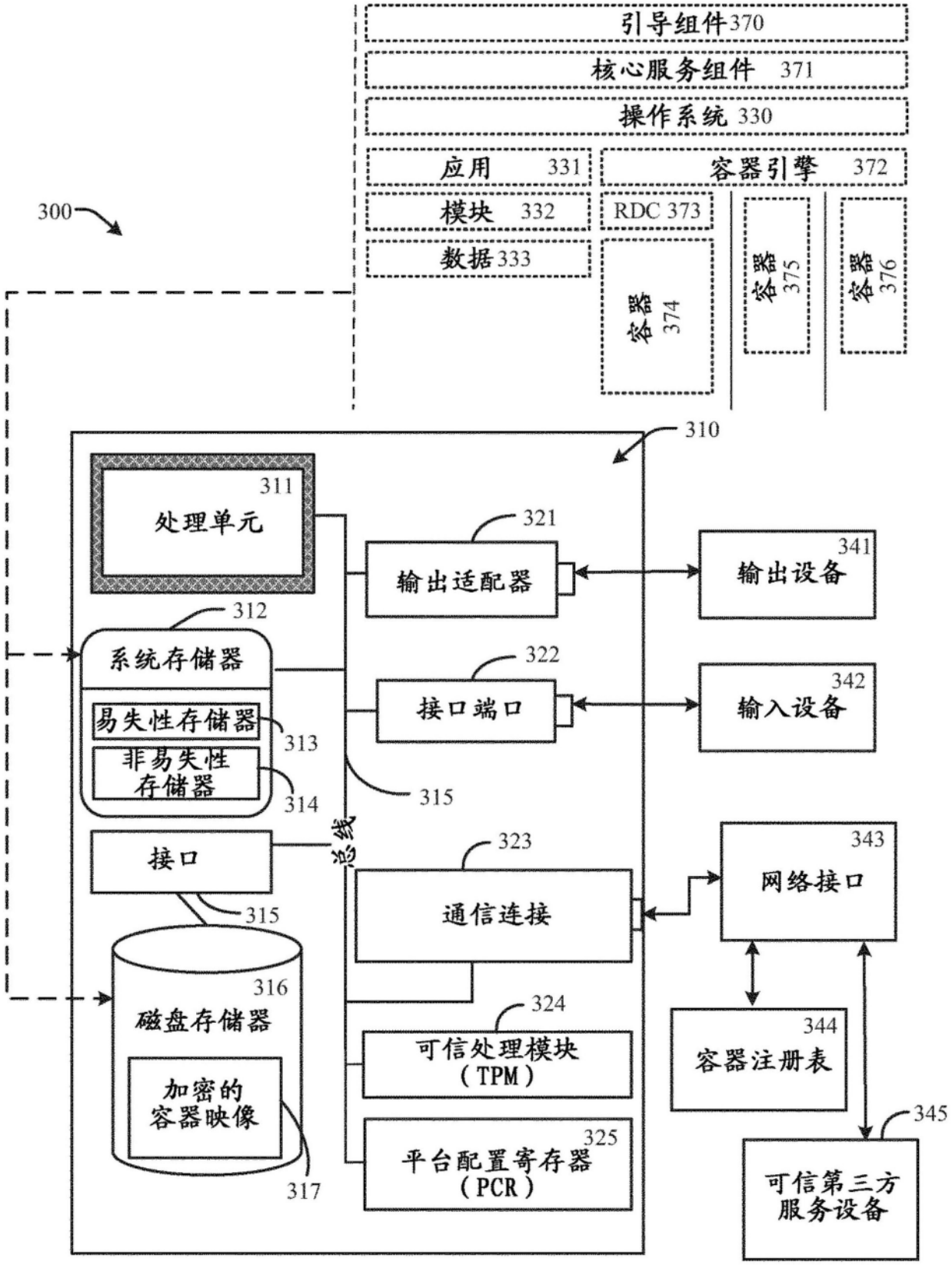

引导组件370
核心服务组件 371
操作系统 330
应用 331
模块 332
数据333
容器引擎 372
RDC 373
容器 374
容器 375
容器 376
300
310
311
处理单元
321
输出适配器
341
输出设备
312
系统存储器
322
接口端口
342
输入设备
易失性存储器
313
非易失性存储器
314
315
总线
接口
323
通信连接
343
网络接口
315
316
磁盘存储器
加密的容器映像
317
324
可信处理模块（TPM）
344
容器注册表
345
325
平台配置寄存器（PCR）
可信第三方服务设备

图3

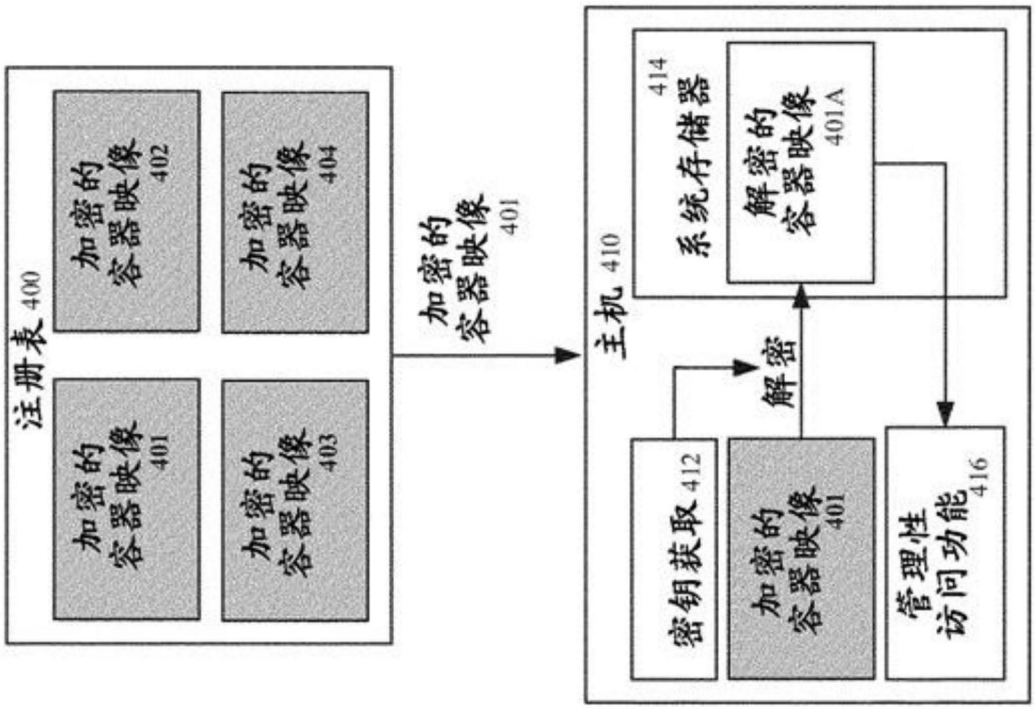

注册表 400
加密的容器映像 402
加密的容器映像 404
加密的容器映像 401
加密的容器映像 403
加密的容器映像 401
主机 410
系统存储器 414
解密的容器映像 401A
密钥获取 412
解密
加密的容器映像 401
管理性访问功能 416

图4A

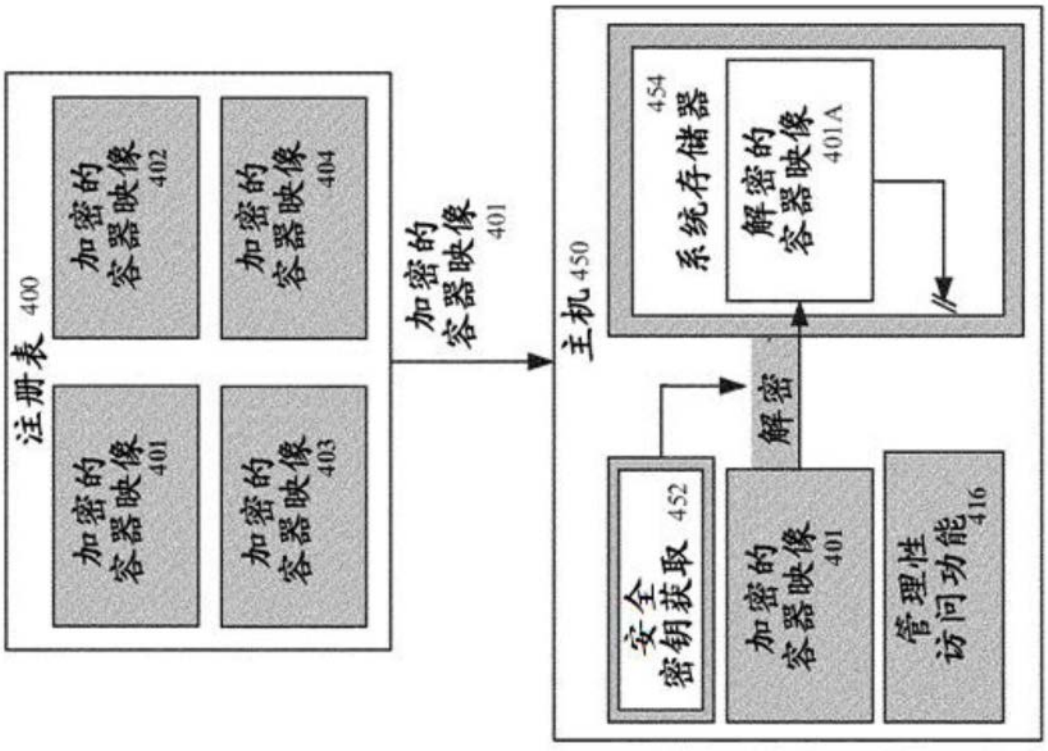

注册表 400
加密的容器映像 402
加密的容器映像 404
加密的容器映像 401
加密的容器映像 403
加密的容器映像 401
主机 450
系统存储器 454
解密的容器映像 401A
安全密钥获取 452
解密
加密的容器映像 401
管理性访问功能 416

图4B

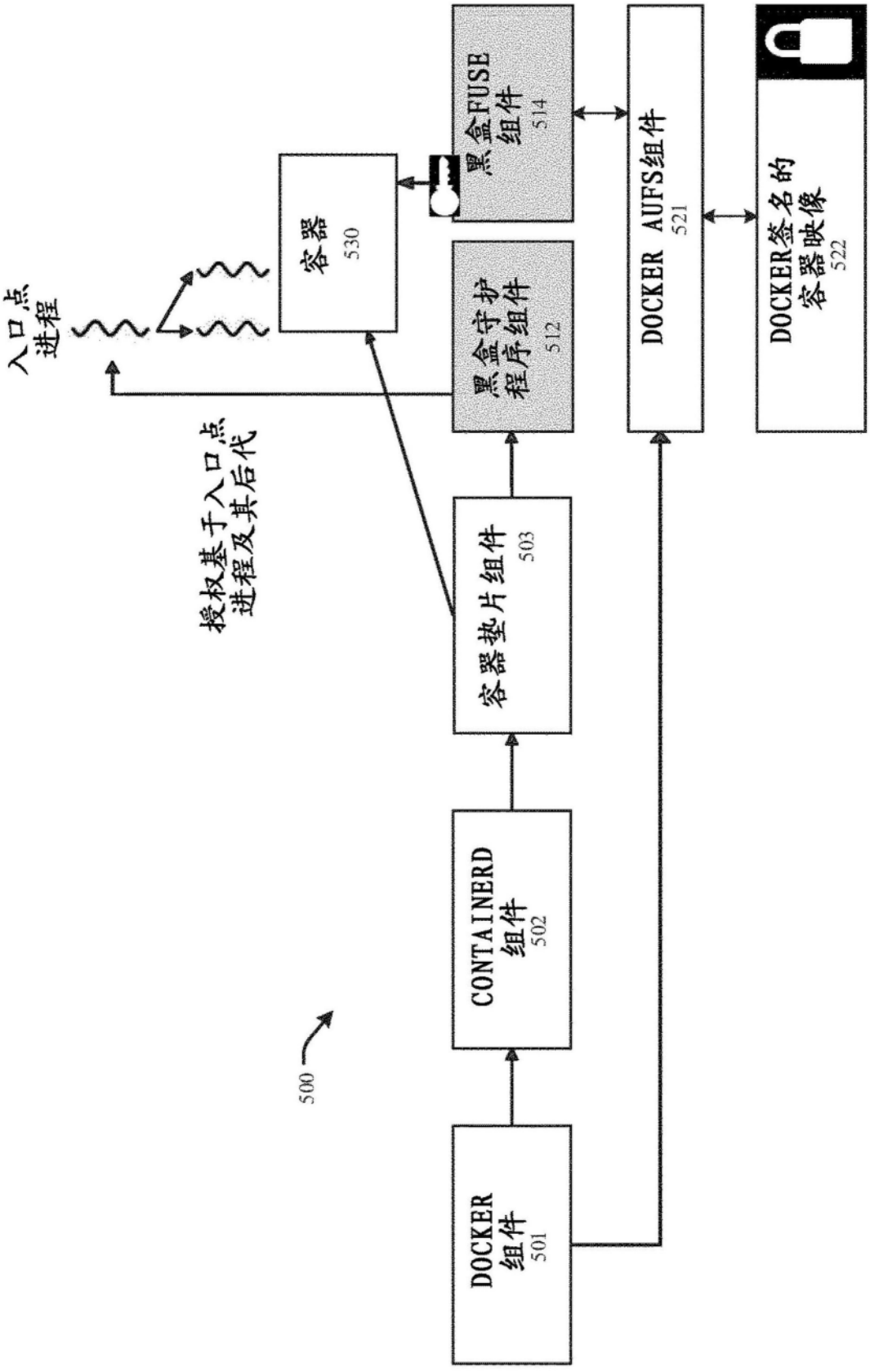

500
DOCKER 组件 501
CONTAINERD 组件 502
容器垫片组件 503
黑盒守护程序组件 512
黑盒FUSE组件 514
DOCKER AUFS组件 521
DOCKER签名的容器映像 522
容器 530
入口点进程
授权基于入口点进程及其后代

图5

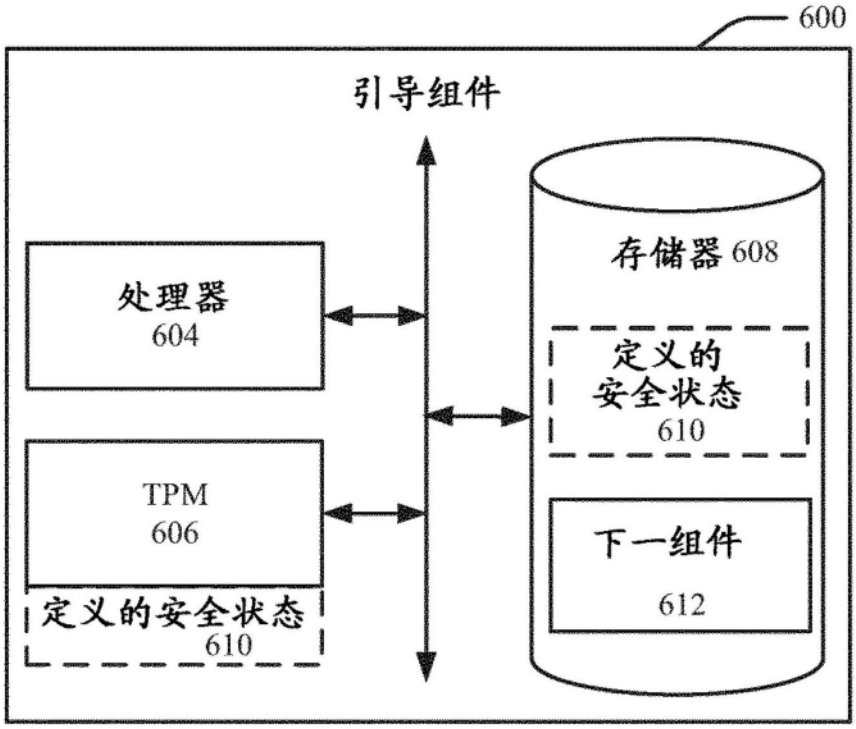

600
引导组件
处理器
604
TPM
606
定义的安全状态
610
存储器 608
定义的
安全状态
610
下一组件
612

图6

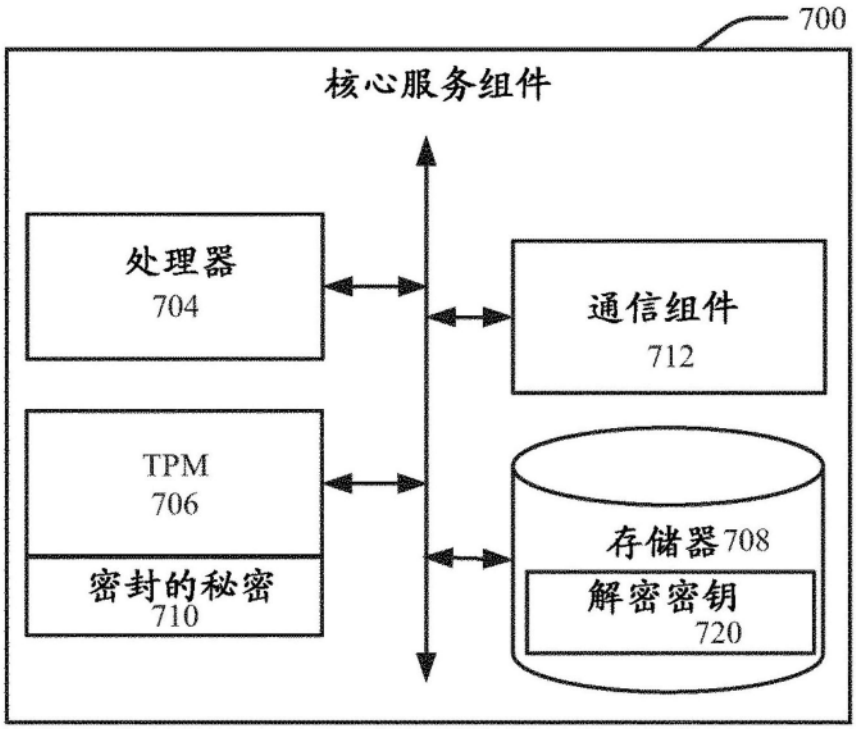

700
核心服务组件
处理器
704
TPM
706
密封的秘密
710
通信组件
712
存储器708
解密密钥
720

图7

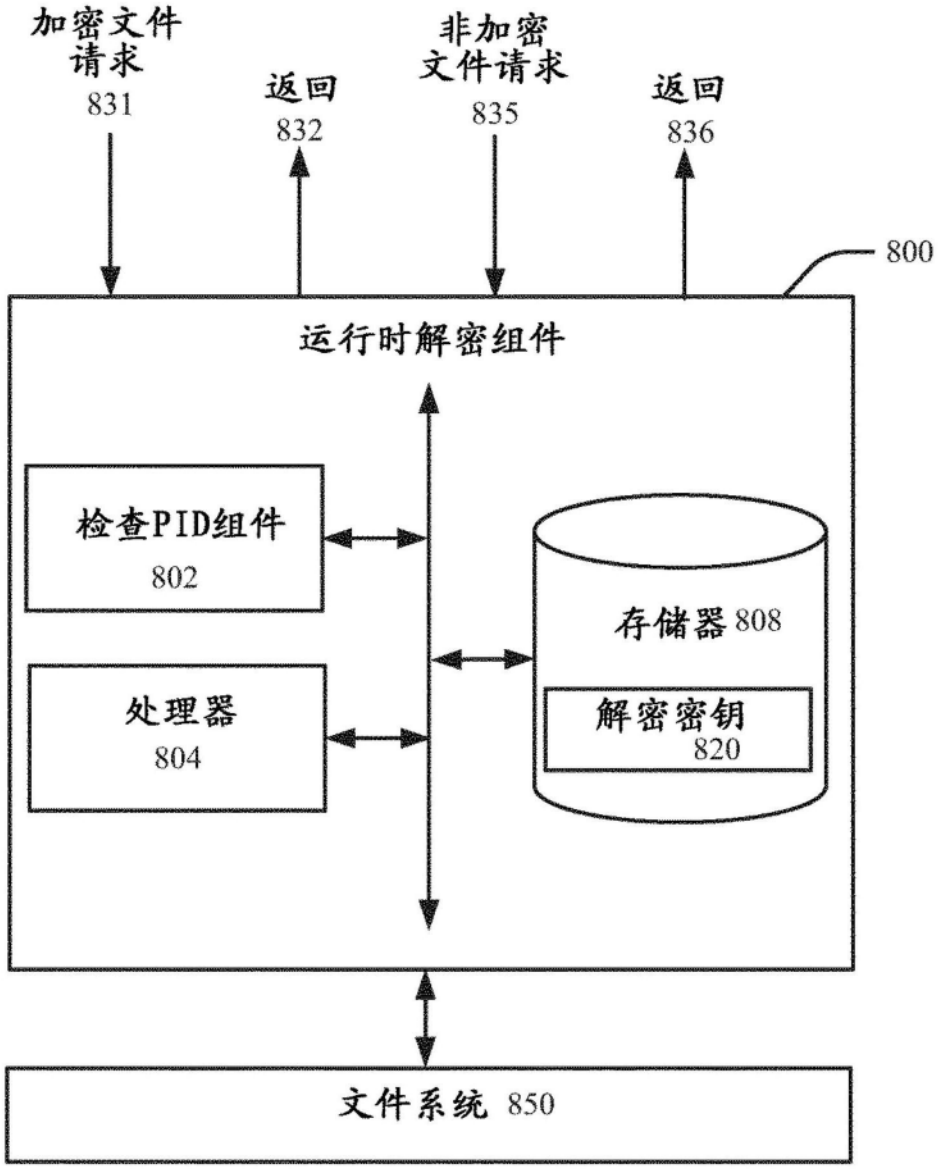

加密文件
请求
831
返回
832
非加密
文件请求
835
返回
836
800
运行时解密组件
检查PID组件
802
处理器
804
存储器 808
解密密钥
820
文件系统 850

图8

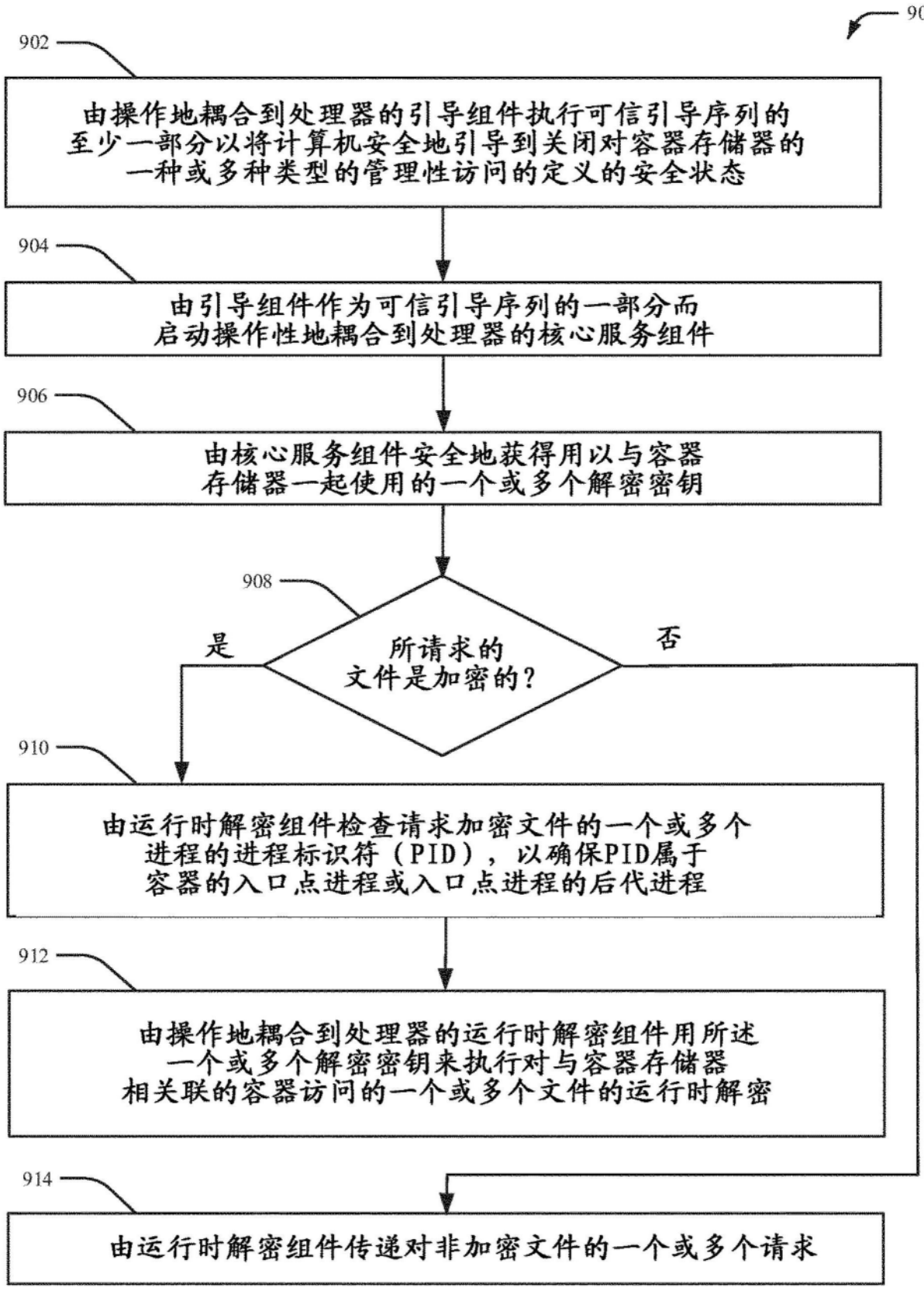

900
902
由操作地耦合到处理器的引导组件执行可信引导序列的至少一部分以将计算机安全地引导到关闭对容器存储器的一种或多种类型的管理性访问的定义的安全状态
904
由引导组件作为可信引导序列的一部分而启动操作性地耦合到处理器的核心服务组件
906
由核心服务组件安全地获得用以与容器存储器一起使用的一个或多个解密密钥
908
所请求的文件是加密的?
是
否
910
由运行时解密组件检查请求加密文件的一个或多个进程的进程标识符（PID），以确保PID属于容器的入口点进程或入口点进程的后代进程
912
由操作地耦合到处理器的运行时解密组件用所述一个或多个解密密钥来执行对与容器存储器相关联的容器访问的一个或多个文件的运行时解密
914
由运行时解密组件传递对非加密文件的一个或多个请求

图9

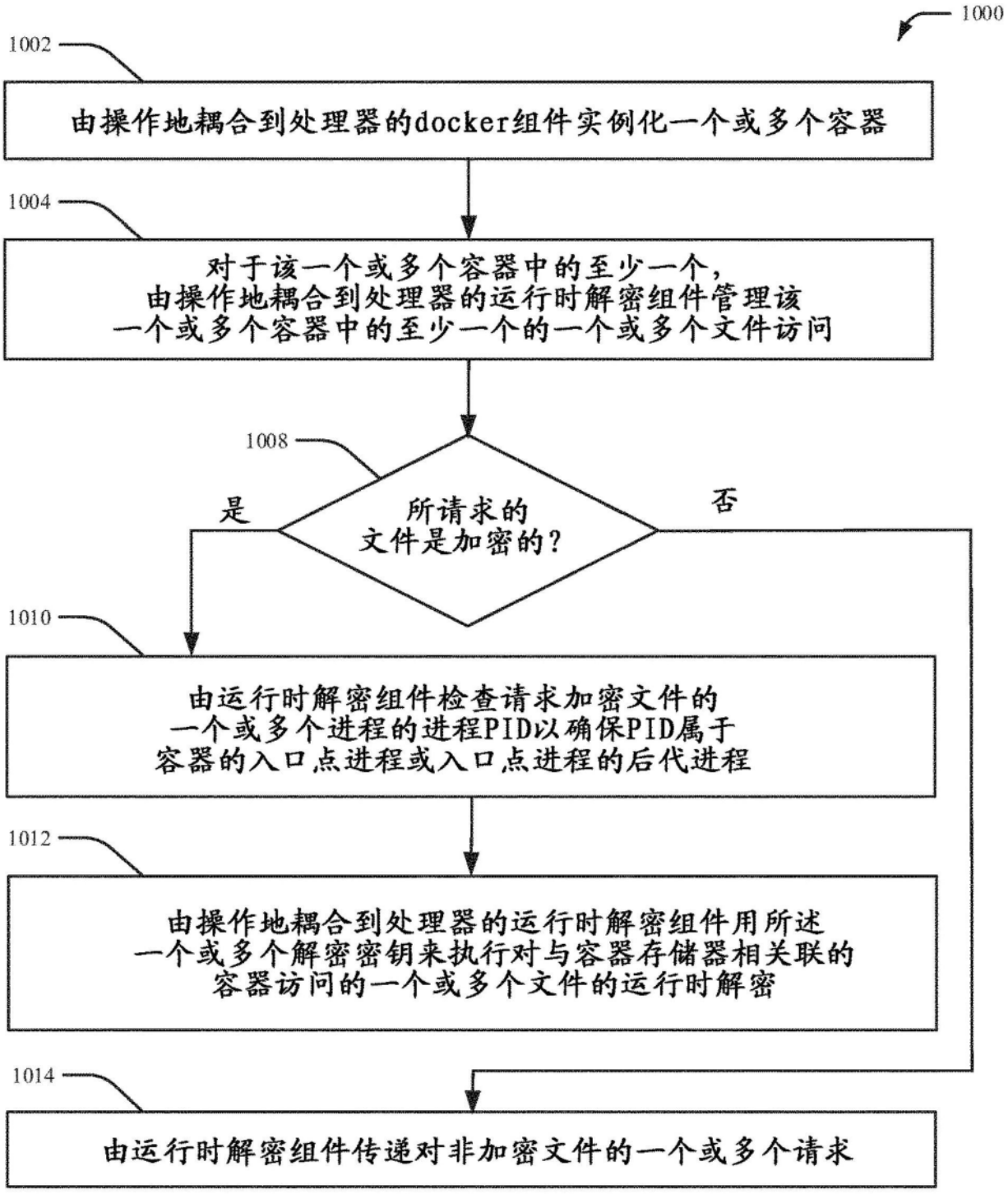

1000
1002
由操作地耦合到处理器的docker组件实例化一个或多个容器
1004
对于该一个或多个容器中的至少一个，
由操作地耦合到处理器的运行时解密组件管理该
一个或多个容器中的至少一个的一个或多个文件访问
1008
所请求的
文件是加密的？
是
否
1010
由运行时解密组件检查请求加密文件的
一个或多个进程的进程PID以确保PID属于
容器的入口点进程或入口点进程的后代进程
1012
由操作地耦合到处理器的运行时解密组件用所述
一个或多个解密密钥来执行对与容器存储器相关联的
容器访问的一个或多个文件的运行时解密
1014
由运行时解密组件传递对非加密文件的一个或多个请求

图10

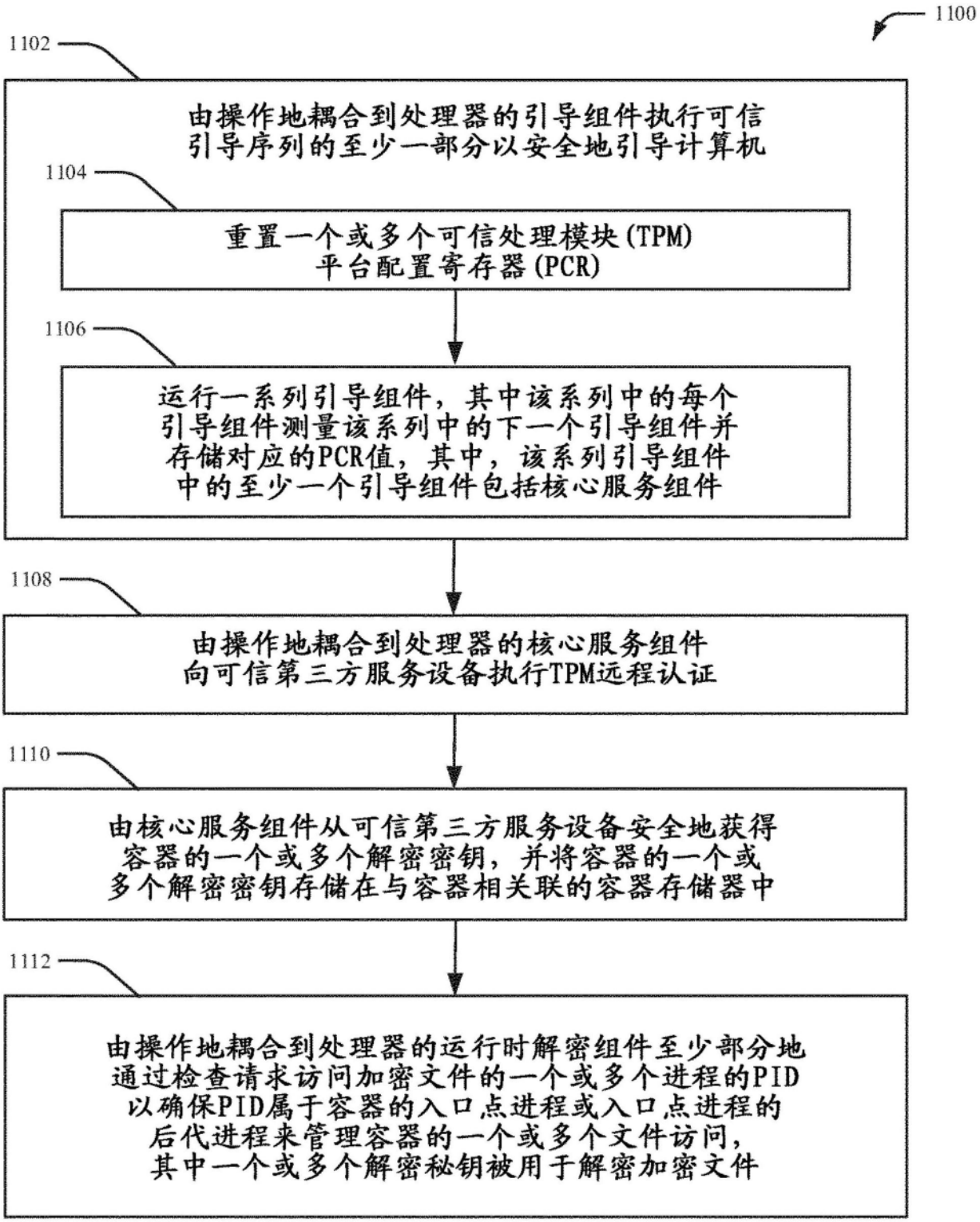

1100
1102
由操作地耦合到处理器的引导组件执行可信引导序列的至少一部分以安全地引导计算机
1104
重置一个或多个可信处理模块(TPM)平台配置寄存器(PCR)
1106
运行一系列引导组件，其中该系列中的每个引导组件测量该系列中的下一个引导组件并存储对应的PCR值，其中，该系列引导组件中的至少一个引导组件包括核心服务组件
1108
由操作地耦合到处理器的核心服务组件向可信第三方服务设备执行TPM远程认证
1110
由核心服务组件从可信第三方服务设备安全地获得容器的一个或多个解密密钥，并将容器的一个或多个解密密钥存储在与容器相关联的容器存储器中
1112
由操作地耦合到处理器的运行时解密组件至少部分地通过检查请求访问加密文件的一个或多个进程的PID以确保PID属于容器的入口点进程或入口点进程的后代进程来管理容器的一个或多个文件访问，其中一个或多个解密秘钥被用于解密加密文件

图11