(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 107092495 B
(45)授权公告日 2020.09.22

(21)申请号 201611007938.1
(22)申请日 2011.09.12
(65)同一申请的已公布的文献号
申请公布号 CN 107092495 A
(43)申请公布日 2017.08.25
(30)优先权数据
12/887,866 2010.09.22 US
(62)分案原申请数据
201180045399.X 2011.09.12
(73)专利权人 英特尔公司
地址 美国加利福尼亚
(72)发明人 A·R·威什曼 S·D·盖切
M·内韦德梅韦尔格尼
U·S·沃里尔 A·卡拉尔
D·R·莫兰 K·布兰诺克
(74)专利代理机构 永新专利商标代理有限公司 72002
代理人 刘瑜 王英
(51)Int.Cl.
G06F 8/65(2018.01)
审查员 祝子豪

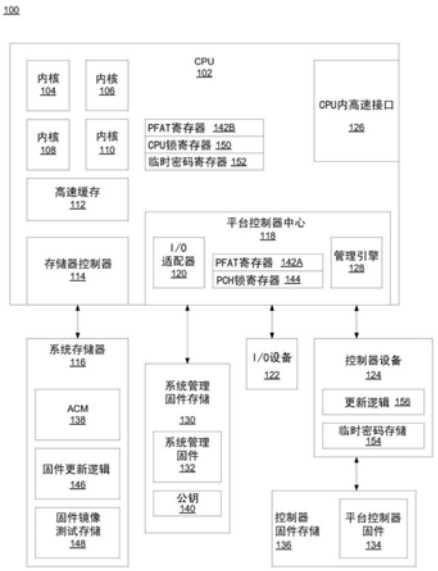
权利要求书4页 说明书9页 附图4页

(54)发明名称

平台固件铠装技术

(57)摘要

公开了一种方法、装置、方法、机器可读介质和系统。在一个实施例中，所述方法包括处理程序。所述处理程序包括在计算机平台的引导中，将位于所述计算机平台中的平台固件更新装置转换为平台固件铠装技术(PFAT)模式。所述计算机平台包括存储平台固件的平台固件存储单元。随后，所述方法响应于所述平台固件更新装置转换为PFAT模式而持续地锁定所述平台固件存储单元。当持续锁定时，仅在平台固件更新装置解锁程序之后，并且仅允许在运行平台中的认证代码模块来向平台固件存储单元进行写入。



CN 107092495 B

1.一种更新计算机平台固件的方法,包括:

在计算机平台中,使用平台部件中心锁定存储第一平台固件的平台固件存储单元;

使用在所述计算机平台的处理器上执行的认证代码模块ACM来执行平台固件更新装置解锁程序;

仅在所述认证代码模块ACM已经执行了平台固件更新装置解锁程序之后并且仅允许由所述认证代码模块ACM向所述平台固件存储单元进行写入。

2.根据权利要求1所述的方法,其中,所述解锁程序还包括:

启动对平台固件铠装技术PFAT保护控制寄存器的特定存储器写入周期,其中,对所述平台固件铠装技术PFAT保护控制寄存器的所述特定存储器写入周期致使耦合到所述平台固件存储单元的控制器接受从所述平台固件更新装置到所述平台固件存储单元的地址空间的写入命令。

3.根据权利要求2所述的方法,还包括:

接收用更新的固件镜像来更新所述第一平台固件的请求;

响应于所述请求,将所述更新的固件镜像加载到所述计算机平台中的系统存储器中;

一旦将所述更新的固件镜像完全地加载到所述系统存储器中,则由所述计算机平台的处理器调用所述认证代码模块ACM;

使用所述认证代码模块ACM来使用公共加密密钥执行对所述更新的固件镜像的认证;

响应于成功认证所述更新的固件镜像,使用所述认证代码模块ACM来执行用于所述平台固件存储单元的解锁程序;以及

使用所述认证代码模块ACM来将已认证的固件镜像从所述系统存储器拷贝到所述平台固件存储单元。

4.根据权利要求3所述的方法,还包括:

当对所述更新的固件镜像的认证失败时,发布错误。

5.根据权利要求3所述的方法,其中,所述第一平台固件包括系统管理固件,并且其中,所述控制器包括平台部件中心。

6.根据权利要求3所述的方法,其中,所述第一平台固件包括控制器管理固件,并且其中,所述控制器包括位于所述计算机平台中的控制器。

7.根据权利要求6所述的方法,还包括:

生成短暂密码;

将所述短暂密码存储在仅能被所述认证代码模块ACM访问的安全单元中;以及

将所述短暂密码至少分配到与所述平台固件存储单元耦合的所述控制器。

8.根据权利要求7所述的方法,还包括:

响应于使用所述公共加密密钥成功地认证所述更新的固件镜像,所述认证代码模块ACM向所述控制器发送请求来请求许可更新所述控制器管理固件,其中,所述请求包括所述短暂密码;

所述控制器将从所述请求接收到的所述短暂密码与在所述计算机平台的最近引导期间接收到的所述短暂密码进行比较;并且

响应于两个所述短暂密码是相同的,所述控制器允许所述认证代码模块ACM执行对所述控制器管理固件的更新。

9.一种更新计算机平台固件的装置,包括:

平台固件存储单元,用于存储第一平台固件;

硬件逻辑单元,用于锁定所述平台固件存储单元,其中,当所述平台固件存储单元被锁定时,仅在认证代码模块ACM已经执行了平台固件更新装置解锁程序之后并且仅允许由所述认证代码模块ACM向所述平台固件存储单元进行写入。

10.根据权利要求9所述的装置,还包括硬件逻辑单元用来:

启动对平台固件铠装技术PFAT保护控制寄存器的特定存储器写入周期,其中,对所述平台固件铠装技术PFAT保护控制寄存器的所述特定存储器写入周期致使耦合到所述平台固件存储单元的控制器接受从所述平台固件更新装置到所述平台固件存储单元的地址空间的写入命令。

11.根据权利要求10所述的装置,还包括硬件逻辑单元用来:

接收用更新的固件镜像来更新所述第一平台固件的请求;

响应于所述请求,将所述更新的固件镜像加载到所述计算机平台中的系统存储器中;

一旦将所述更新的固件镜像完全地加载到系统存储器中,则调用所述认证代码模块ACM;

使用存储的公共加密密钥执行对所述更新的固件镜像的认证;

响应于成功认证所述更新的固件镜像,执行用于所述平台固件存储单元的解锁程序;以及

将已认证的更新的固件镜像从所述系统存储器拷贝到所述平台固件存储单元。

12.根据权利要求11所述的装置,还包括硬件逻辑单元用来:

当对所述更新的固件镜像的认证失败时,发布错误。

13.根据权利要求11所述的装置,其中,所述第一平台固件包括系统管理固件,并且其中,所述控制器包括平台部件中心。

14.根据权利要求11所述的装置,其中,所述第一平台固件包括控制器管理固件,并且其中,所述控制器包括位于所述计算机平台中的控制器。

15.根据权利要求14所述的装置,还包括这样的硬件逻辑单元:

在所述计算机平台的引导中,生成短暂密码,将所述短暂密码存储在仅能被所述认证代码模块ACM访问的单元中,以及将所述短暂密码至少分配到与所述平台固件存储单元耦合的所述控制器。

16.根据权利要求15所述的装置,还包括:

响应于使用所述公共加密密钥成功地认证所述更新的固件镜像,所述认证代码模块ACM向所述控制器发送请求来请求许可更新所述控制器管理固件,其中,所述请求包括所述短暂密码;

在所述控制器内用于将从所述请求接收到的所述短暂密码与在所述计算机平台的最近引导期间接收到的所述短暂密码进行比较的硬件逻辑单元;以及

在所述控制器内用于响应于两个所述短暂密码是相同的而允许所述认证代码模块ACM执行对所述控制器管理固件的更新的硬件逻辑单元。

17.一种其上存储有指令的机器可读介质,如果机器执行所述指令则致使所述机器执行包括以下步骤的方法:

在计算机平台的引导中，将位于所述计算机平台中的平台固件更新装置转换为平台固件铠装技术PFAT模式，其中，所述计算机平台包括存储第一平台固件的平台固件存储单元；

响应于所述平台固件更新装置转换为所述平台固件铠装技术PFAT模式，持续地锁定所述平台固件存储单元，其中，当持续锁定时，不允许任何实体向所述平台固件存储单元进行写入，除非在平台固件更新装置解锁程序之后，由认证代码模块ACM进行写入。

18.根据权利要求17所述的机器可读介质，其中，所述解锁程序还包括：

所述平台固件更新装置启动对平台固件铠装技术PFAT保护控制寄存器的特定存储器写入周期，其中，对所述平台固件铠装技术PFAT保护控制寄存器的所述特定存储器写入周期致使耦合到所述平台固件存储单元的控制器接受从所述平台固件更新装置到所述平台固件存储单元的地址空间的写入命令。

19.根据权利要求18所述的机器可读介质，其中，被执行的方法还包括：

将公共加密密钥提供给所述平台固件存储单元；

接收用更新的固件镜像来更新所述第一平台固件的非安全请求；

响应于所述非安全请求，将所述更新的固件镜像加载到所述计算机平台中的系统存储器中；

一旦将所述固件镜像完全地加载到系统存储器中，则调用所述认证代码模块ACM；

使用所述公共加密密钥执行对所述更新的固件镜像的认证；

响应于成功认证所述更新的固件镜像，执行用于所述平台固件存储单元的解锁程序；以及

将已认证的固件镜像从所述系统存储器拷贝到所述平台固件存储单元。

20.根据权利要求19所述的机器可读介质，其中，所述被执行的方法还包括：

当对所述固件镜像的认证失败时，发布错误。

21.根据权利要求19所述的机器可读介质，其中，所述第一平台固件包括系统管理固件，并且其中，所述控制器包括平台部件中心。

22.根据权利要求19所述的机器可读介质，其中，所述第一平台固件包括控制器管理固件，并且其中，所述控制器包括位于所述计算机平台中的控制器。

23.根据权利要求22所述的机器可读介质，其中，所述被执行的方法还包括：

在所述计算机平台的每一次引导中，

生成短暂密码；

将所述短暂密码存储在仅能被所述认证代码模块ACM访问的安全单元中；以及

将所述短暂密码至少分配到与所述平台固件存储单元耦合的所述控制器。

24.根据权利要求23所述的机器可读介质，其中，所述被执行的方法还包括：

响应于使用所述公共加密密钥成功地认证所述更新的固件镜像，所述认证代码模块ACM向所述控制器发送安全请求来请求许可更新所述控制器管理固件，其中，所述安全请求包括所述短暂密码；

所述控制器将从所述安全请求接收到的所述短暂密码与在所述计算机平台的最近引导期间接收到的所述短暂密码进行比较；并且

响应于两个所述短暂密码是相同的，所述控制器允许所述认证代码模块ACM执行对所述控制器管理固件的更新。

25.一种更新计算机平台固件的系统,包括:

平台部件中心;

用于存储平台固件的平台固件存储单元;

用于存储认证代码模块ACM的系统存储器;

在计算机平台的引导中,将位于所述计算机平台中的平台固件更新装置转换为平台固件铠装技术PFAT模式的逻辑单元;以及

响应于所述平台固件更新装置转换为所述平台固件铠装技术PFAT模式,所述平台部件中心持续地锁定所述平台固件存储单元,其中,当持续锁定时,不允许任何实体向所述平台固件存储单元进行写入,除非在平台固件更新装置解锁程序之后,由所述认证代码模块ACM进行写入。

26.一种其上存储有指令的机器可读介质,如果机器执行所述指令则致使所述机器执行根据权利要求1-8中的任一项所述的方法。

27.一种更新计算机平台固件的设备,包括:

存储指令的存储器;以及

耦合到所述存储器的处理器,所述指令在被所述处理器执行时执行根据权利要求1-8中的任一项所述的方法。

28.一种更新计算机平台固件的装置,包括用于执行根据权利要求1-8中的任一项所述的方法的模块。

平台固件铠装技术

技术领域

[0001] 本发明涉及安全地更新计算机平台固件。

背景技术

[0002] 在计算机系统上操作的第一组指令源自平台固件。平台固件可以包括与基本输入/输出系统、可扩展固件、嵌入式控制器和微控制器固件相关联的固件，以及任意其他驻留在计算机平台内的存储单元处的这种固件。平台固件在贯穿计算机系统的生命期中通常不是静态的。如同有操作系统和软件应用的更新一样，也有平台固件的更新。因为固件指令对许多计算机系统的成功操作是非常重要的，所以当更新固件时，以安全的方式进行更新是至关重要的。

附图说明

[0003] 本发明通过示例的方式示出并且不受附图限制，在附图中类似的标记指示相似的元件，在其中：

[0004] 图1示出了能够安全地更新平台固件的计算机系统的实施例。

[0005] 图2是锁定平台部件中心使其不允许向系统管理固件存储空间写入的过程的实施例的流程图。

[0006] 图3A示出了开始安全平台固件更新程序的过程的实施例的流程图。

[0007] 图3B示出了继续安全平台固件更新程序的过程的实施例的流程图。

[0008] 图4是安全地更新在计算机系统中的控制器固件的过程的实施例的流程图。

具体实施方式

[0009] 公开了能够安全地更新平台固件的装置、系统、方法和计算机可读介质的实施例。

[0010] 在计算机系统中的固件提供了用于计算机系统中的一个或多个设备的多种类型的初始化、管理和操作指令。固件的更新通常充满安全漏洞。如果恶意实体能够将一块受到危害的固件提供给计算机系统，则破坏可能是严重的，这是因为在计算机系统的操作期间，固件通常是低于标准的病毒级别保护的级别。因此，在平台固件的更新期间，安全是极其重要的。

[0011] 在许多实施例中，在计算机系统中启动了平台固件铠装技术(platform firmware armoring technology,PFAT)模式。PFAT模式限制了大多数实体更新平台固件的能力。确切地说，PFAT模式能要求在安全认证代码(AC)模式下由认证代码模块(ACM)执行系统管理固件更新。在标准的操作模式中，平台部件中心(PCH)可以停止任何写入到达存储。为了允许固件更新，逻辑可以调用ACM，并且ACM可以接着通过对CPU寄存器执行特定写周期以解锁固件存储来用于进行写入。在许多实施例中，ACM是仅有的可以执行特定写周期的实体，并且一旦解锁固件存储，ACM则是仅有的可以执行实际固件更新的实体。此外，ACM可以测量更新后的固件镜像以使用提供给CPU的公钥来验证其真实性。

[0012] 图1示出了能够安全更新平台固件的计算机系统的实施例。

[0013] 示出了计算机系统100。计算机系统可以是台式机、服务器、工作站、膝上型电脑、手持型、电视机顶盒、媒体中心、游戏控制器、集成系统(例如在车中),或其他类型的计算机系统。在若干实施例中,计算机系统100包括一个或多个中央处理单元(CPU),也称为“处理器”。虽然在许多实施例中潜在地存在许多CPU,但是为了清楚,在图1示出的实施例中仅示出了CPU 102。CPU 102可以是Intel®公司的CPU或另一品牌的CPU。在不同的实施例中,CPU 102包括一个或多个内核。示出的CPU 102包括四个内核(内核104、106、108和110)。

[0014] 在许多实施例中,每一个内核包括内部功能块,例如一个或多个执行单元、收回(retirement)单元、一组通用寄存器和专用寄存器等。在单线程内核中,每一个内核可被称为硬件线程。当内核是多线程或超线程时,则在每一个内核内操作的每一个线程也可以被称为硬件线程。因此,运行在计算机系统100中的执行的任意单线程均可被称作硬件线程。例如,在图1中,如果每一个内核是单线程的,则在系统中存在四个硬件线程(四个内核)。另一方面,如果每一个内核是多线程的并且具有同时维护两个线程状态的能力,则在系统中存在八个硬件线程(四个内核,每一个内核有两个线程)。

[0015] CPU 102还可以包括一个或多个高速缓存,例如高速缓存112。在许多未示出的实施例中,可以实施除了高速缓存112以外的另外的高速缓存,使得在存储器和每一个内核中的执行单元之间存在多个级别的高速缓存。在不同的实施例中,可以用不同的方式分配高速缓存112。此外,在不同的实施例中,高速缓存112可以具有许多不同大小。例如,高速缓存112可以是8兆字节(MB)高速缓存、16MB高速缓存等。此外,在不同的实施例中,高速缓存可以直接映射高速缓存、全相联高速缓存、多路组相联高速缓存,或具有另一类型映射的高速缓存。在许多实施例中,高速缓存112可以包括一个在所有内核间共享的大的部分,或者高速缓存112可以被划分为若干个独立的功能片(例如,对于每一个内核划分一个片)。高速缓存112还可以包括在所有内核间共享的一个部分和为用于每个内核的独立功能片的若干个其他部分。

[0016] 在许多实施例中,CPU 102包括提供与系统存储器116进行通信的接口的集成系统存储器控制器114。在另一未示出的实施例中,存储器控制器114可以位于计算机系统100中的分立的其他单元。

[0017] 系统存储器116可以包括动态随机存取存储器(DRAM),例如双倍数据速率(DDR)类型的DRAM;非易失性存储器,例如闪存、相变存储器(PCM);或其它类型的存储器技术。系统存储器116可以是存储待由CPU102操作的数据和指令的通用存储器。此外,在计算机系统100内可以有其他潜在的设备,所述设备具有读取和写入系统存储器的能力,例如能直接存储器存取(DMA)的I/O(输入/输出)设备。

[0018] 将CPU 102与系统存储器116耦合的链路(即,总线、互连等)可以包括能够传输数据、地址、控制和时钟信息的一个或多个光纤、金属或其他电线(即,线路)。

[0019] 平台控制器中心118(例如,I/O控制器中心)包括使得能够在CPU 102和外部I/O设备之间进行通信的I/O接口。该中心可以包括一个或多个I/O适配器,例如I/O适配器120。I/O适配器将在CPU 102内使用的主机通信协议转换为与特定I/O设备(例如I/O设备122)兼容的协议。给定的I/O适配器可以转换的一些协议包括外围部件互连(PCI)、通用串行总线(USB)、IDE、SCSI和1394“火线”等。此外,可以有一个或多个无线协议I/O适配器。无线协议

的示例有蓝牙、基于IEEE 802.11的无线协议，和蜂窝协议等。

[0020] 在许多实施例中，控制器设备124驻留在计算机系统100内。控制器设备124可以并入多个功能。例如，RAID存储控制器设备可以存在于计算机系统100内。RAID控制器可以管理硬盘驱动器阵列或固态硬盘(SSD)阵列。控制器设备的其他示例可以是分立式带外管理引擎、嵌入式微控制器，或其它类型的控制器。

[0021] CPU内高速接口126可以提供耦合到一个或多个另外的CPU的链路的接口，并且允许进行CPU内通信。例如，CPU内高速接口可以是快速路径互连或其他类似的接口。

[0022] 虽然没有示出，但是在许多实施例中，计算机系统100包括能够提供使一个或多个客户操作系统(OS)运行在虚拟机(VM)环境中的虚拟化环境的硬件和软件逻辑。虚拟机监视器(VMM)或管理程序可以在系统内的逻辑中实现，以隔离每一个VM的操作环境(即，使每一个VM和OS以及运行其中的应用与系统中存在的其它VM隔离的，并且不会感知系统中存在的其他VM)。

[0023] 在许多实施例中，在系统100中存在管理引擎128。管理引擎可以包括多个特征，所述特征包括涉及远程管理、安全管理和电源管理的管理逻辑。在许多实施例中，管理引擎128使用带外(OOB)通信通道，其在计算机系统100中运行的操作系统(OS)的级别之下操作。OOB通道通常将能够维持与远程系统的通信而不论OS的状态。在许多实施例中，虽然计算机系统100处于低功率状态或完全关闭，但是OOB通道也能继续通信。在一些实施例中，管理引擎128包括Intel®主动管理技术硬件逻辑。在其他的实施例中，使用了另一形式的硬件逻辑。

[0024] 在许多实施例中，固件存储在计算机系统100中。存储在计算机系统100中的任意单元的固件可被称为“平台固件”。更具体地，可以存在多种类型的固件。例如，系统管理固件存储单元130可以存储系统管理固件132。系统管理固件130可以包括可扩展固件、基本输入/输出系统(BIOS)，和/或可被用来例如在引导过程期间提供用于计算机系统100的关键指令的其他类型的固件。在计算机系统中的另一固件可以包括用于控制器设备124的平台控制器固件134。可将这个固件存储在平台控制器固件存储136中，其与控制器设备124相耦合。平台控制器固件132可以提供关于管理控制器设备124的特征的指令。

[0025] 在许多实施例中，在系统中的每个固件存储单元(例如，系统管理固件存储单元130、平台控制器固件存储单元136等)包括例如NAND闪存、NOR闪存、相变存储器的一种非易失性类型的存储器，或另一形式的非易失性存储器。

[0026] 虽然为了清楚的目的而没有示出，但是CPU可以具有另外的接口，例如处理图形和网络业务的高速I/O接口。在许多实施例中，这些高速I/O接口可以包括一个或多个PCI-Express接口。

[0027] 计算机系统100存储代码来安全地启动认证代码模块(ACM)138，该认证代码模块138是CPU 102调用并签名(即，模块的安全测量)的软件模块。可以使用CPU 102的生产厂商管理的私钥利用非对称加密对ACM138进行签名。当CPU 102调用ACM 138时，首先使用提供的公钥的散列进行认证，所述公钥存储在CPU 102、芯片组相关电路(例如分立的PCH)，或在计算机系统中的其他硬件中。使用公钥对由内部私钥加密的信息进行解密。一般来说，公钥将是不变的。通过使用这些公知的安全程序来测量ACM 138，能够证实模块为可信任执行环境。出于安全的目的，ACM可以运行在认证代码(AC)模式中。当在AC模式中时，所有的系统中

断和事件都不可中断ACM，并且保护ACM不受其他系统和DMA代理的影响。当以AC模式运行时，仅有一个硬件线程是活动的，因此集合所有其他硬件线程并使其进入静止/睡眠状态。

[0028] 在计算机系统100的首次引导期间或在此之前，可以提供固件更新公钥140并将其存储在系统管理固件存储130中。可以使用固件更新公钥140来认证固件镜像信息。在一些实施例中，排他地使用公钥140用于认证固件镜像。虽然使用了术语“公钥”并且一些实施例存储了实际字母数字、未加密密钥，但是在其他的实施例中，“公钥”可以指公钥的散列。可将公钥存储在处理器中的只读存储器(ROM)中或其他存储单元。可以存储散列而非完整的公钥来潜在地节省存储空间。将被写入到计算机系统中的固件镜像可以包括公钥。然后可以散列化在镜像中的公钥并将其与已经存储在系统ROM中的公钥的散列进行比较。在某些实施例中，出于另外的安全目的可以将散列加密。

[0029] 在许多实施例中，计算机系统100能够进入平台固件铠装技术(PFAT)模式。PFAT模式致使平台固件的锁定。当在PFAT模式中时，仅允许运行在AC模式中的ACM执行对计算机系统100中的固件存储单元的更新(即，写入)。并且甚至要求ACM执行特定的过程以允许对平台固件存储进行写入。一旦在PFAT模式中，计算机系统不能退出该模式。在许多实施例中，在每次计算机系统引导时进入的PFAT模式致使任何固件所在的存储器地址空间锁定。例如，因为系统管理固件存储130通过平台控制器中心118与CPU 102相耦合，所以任何针对系统管理固件存储130的写入会路由通过平台控制器中心118。当系统在PFAT模式中时，平台控制器中心118将拒绝任何向保留用于系统管理固件存储130的存储器单元的尝试性写入(没有首先使用安全的基于ACM的解锁程序)。安全的基于ACM的解锁程序将在下面讨论。

[0030] 在许多实施例中，在引导过程期间在固件内执行的代码将设置启动PFAT模式的一次写入寄存器。例如，位于平台中的PFAT管理寄存器可以具有PFAT启动位来启动PFAT模式。在一些实施例中，PFAT管理寄存器位于平台控制器中心118(PFAT寄存器142A)中。在其他的实施例中，PFAT管理寄存器位于CPU非内核的其他地方(PFAT寄存器142B)。这种寄存器可以具有与管理PFAT模式相关的若干个位。例如，PFAT寄存器可以包括PFAT模式启动位，当设置所述启动位时启动PFAT模式。另外，PFAT寄存器还可以包括PFAT模式锁定位，当设置所述锁定位时不允许进一步更新PFAT模式启动位(至少到系统完全重置前)。锁定位消除了恶意实体在计算机系统100正常操作期间使PFAT模式失效的能力。

[0031] 如讨论的，一旦在PFAT模式中，则不允许定期的向表示系统管理固件存储130的存储器地址空间写入。在许多实施例中，当计算机系统100进入PFAT模式时，平台控制器中心118设置内部PCH锁寄存器144，其指示向系统管理固件存储130的写入是不允许的。当设置PCH锁寄存器114时，平台控制器中心118将丢弃任何尝试向固件存储130写入的事务。

[0032] 此外，在许多实施例中，运行在OS顶层的固件更新逻辑146存在于系统存储器116中。固件更新逻辑146可以接收更新在计算机系统100中存在的特定固件的请求。这个更新请求可以来自网络上的系统管理员，它可以来自于系统内部(例如，来自修复OS的修改)，或来自其他单元。在许多实施例中，更新固件的请求伴随着实际更新的镜像。固件镜像要么可以直接伴随请求，要么在稍后于接收并接受请求时到达。固件镜像可以是固件的完全重写(例如，新的版本)或可以是整个固件的较小部分的更新(例如，重写整个固件的小部分)。在某个时间点，固件更新逻辑146接收实际的更新镜像。固件更新镜像存储在系统存储器116中的固件镜像测试存储148中。一旦固件更新逻辑146将整个镜像存储到测试存储中，则调

用ACM 138并且系统进入AC模式。在从运行固件更新逻辑146的标准模式转换到运行在AC模式中的ACM 138期间，固件更新逻辑146将指向系统存储器116中的固件镜像测试存储148单元的指针传递给ACM 138。这种传递可以使用在CPU 102中的寄存器来存储指针，或者指针可以存储在系统存储器116中的已知传递单元。

[0033] 一旦在AC模式中并且准备好存储固件镜像的单元，ACM 138使用公钥140来初始化固件镜像的认证程序。认证程序使用不对称加密来解密并且测量接收到的固件镜像。在许多实施例中，可以使用存储在计算机系统中的不可变的公钥来验证从供应商处接收到的固件镜像是可信的。在许多实施例中，公钥对任何实体是可取得的，在其他的实施例中，一组有限的实体具有访问特权来读取存储公钥的存储单元。例如，在一些实施例中，在AC模式中的ACM 138是仅有的能够访问存储在CPU 102中的公钥140的实体。

[0034] ACM 138执行公钥测量以认证存储在固件镜像测试存储148中的固件镜像。如果能够成功认证固件镜像(即，镜像的安全性没有受到危害)，则ACM 138对CPU 102中的锁寄存器150执行特定的写周期。在许多实施例中，在CPU 102中的锁寄存器150是特定的MSR(模型相关寄存器)，其仅能被AC模式中的ACM写入。因此，这个寄存器对于标准模式的OS或另一类似的实体来说是禁止的。对锁寄存器150的这个特定写周期致使CPU 102生成向平台控制器中心118发送的安全的基于ACM的解锁命令。当这个安全的解锁命令到达时，在平台控制器中心118内的逻辑将清除PCH锁寄存器144。当清除了PCH锁寄存器114时，平台控制器中心118将允许写入系统管理固件存储130地址空间。

[0035] 一旦允许写入固件存储，ACM 138就将认证的固件镜像从固件镜像测试存储148单元拷贝到系统管理固件存储130。一旦拷贝结束，ACM 138可以这样发出随后的锁命令：通过向CPU锁寄存器150发出另一特定写周期，该写周期接着致使生成向平台控制器中心118发送的安全的基于ACM的锁命令。当平台部件中心118接收到锁命令时，在中心内的逻辑设置PCH锁寄存器144，并且再一次阻止系统管理固件存储130接收任何写入。

[0036] 在图1中的前述示例涉及系统管理固件132的更新。在另一示例中，更新可以涉及平台控制器固件134。

[0037] 在计算机系统100每一次引导期间，可以生成短暂的(即，临时的)密码。生成这个短暂密码的逻辑可以在CPU 102中、在系统管理固件132中或在计算机系统100的其他单元中。短暂密码通常由操作系统在计算机系统的正常操作之前的早期引导序列中生成。生成的短暂密码内部地存储在CPU 102中的临时密码寄存器152中。还将这个同样的密码分配给在计算机系统中需要在正常操作期间与CPU进行安全握手的任何控制器或其他设备。例如控制器设备124的控制器可以具有内部临时密码存储单元154，用以存储在引导时间密码分配期间接收到的密码。通常在安全引导过程期间创建、分配并存储这个短暂密码来获得更高的安全性。安全引导过程比在正常操作期间更少可能受到危害，因此在这个时间期间分配短暂密码将使得密码更少可能被盗取。

[0038] 随后，在计算机系统100完全操作之后(例如，正在运行OS)，固件更新逻辑146可以接收更新平台控制器固件134的请求。当接收到这个请求时，进行关于将更新镜像拷贝到固件镜像测试存储单元148的相同过程。固件更新逻辑146调用ACM 138并且将系统的控制传递给ACM 138。ACM138接着认证镜像，并且如果成功地认证了镜像，则尝试执行固件更新。但是，不像系统管理固件132，ACM 138不具有对平台控制器固件134的完全控制。

[0039] 控制器设备124可以具有其自身的安全程序和不同于CPU 102的独立的操作需求。本质上,控制器设备124期望确切地知道对控制器固件存储136空间的写入请求事实上来自于ACM 138。为了创建更安全的验证过程,ACM 138可以向控制器设备124发送请求来执行对控制器固件存储136中的地址空间的固件更新。这个基于ACM的请求可以和短暂密码一起到达。

[0040] 为了验证接收到的ACM 138请求的真实性,控制器设备124可以将在请求中的接收到的短暂密码与最初在计算机系统100的当前引导期间到达的所存储的短暂密码进行比较。在许多实施例中,在控制器设备124中的更新逻辑156执行这个比较。如果两个密码是相同的,则控制器设备124可以向来自于ACM 138的写入开放控制器固件存储136地址空间。否则,将保持锁定控制器固件存储136,并且控制器设备124可以生成指示安全问题的某种错误消息。在许多实施例中,为了保持在CPU 102中的临时密码寄存器152中的短暂密码的安全,仅在AC模式下由ACM 138访问该寄存器。还有,在许多实施例中,一旦ACM 138完成了将已认证的固件镜像写入控制器固件存储136,ACM 138就可以接着向控制器设备124发送后继的通信,声明更新已结束,因而控制器设备124知道不允许向控制器固件存储空间136进行进一步写入。

[0041] 在许多实施例中,ACM可以实施回滚保护来用于固件更新。回滚保护限制平台固件的更新仅为固件更新镜像的较新版本。在一些实施例中,固件镜像可以具有包括固件版本的头部,并且当前驻留的固件具有类似的头部。ACM可以读取两个头部,并且如果新的固件镜像是比当前驻留在平台中的固件更新版本的固件,则仅允许将新的固件镜像写入到固件存储单元。

[0042] 图2是锁定平台部件中心使其不允许向系统管理固件存储空间写入的过程的实施例的流程图。可以由处理逻辑来执行该过程,所述处理逻辑可以包括硬件电路、软件代码、固件代码,或任意这三种类型处理逻辑的结合。贯穿整个本文档使用术语“处理逻辑”的任何事物可以有效地用上述逻辑的任意组合以及甚至其他形式的逻辑来实现。

[0043] 由处理逻辑在处理器引导期间进行的过程开始于:设置PFAT模式启动位(处理块200)来激活计算机系统中的固件更新保护装置。PFAT模式启动位可以位于计算机系统中的PCH设备中的固件管理寄存器中。一旦设置了PFAT模式启动位来启动PFAT,处理逻辑接着设置PFAT模式启动锁定位(处理块202)。PFAT模式启动锁定位不允许进一步修改PFAT模式启动位。在许多实施例中,PFAT模式启动锁定位是在每次计算机系统引导时写入一次位。在这些实施例中,在计算机系统完全重置期间,可以再次修改PFAT模式锁定位。因此,启动PFAT模式并接着锁定PFAT模式的过程将在每次引导期间发生。

[0044] 虽然在图2中未示出其他的实施例,但是PFAT模式启动位是写入一次寄存器,其在被写入后的计算机系统的整个剩余生命期中不能被修改。在这些实施例中,将在每次计算机系统引导期间写入一次PFAT模式启动位,并且此后(直到重新引导计算机系统)将指示计算机系统是否已启动PFAT模式。这个过程块如块200所示。还有,在这些实施例中,PFAT模式启动锁定位不必像PFAT模式启动位那样在引导过程期间首次写入之后将永远是不可再次写入的。因此,过程块202在这些替换实施例中不是必需的。

[0045] 一旦结束了这个过程,计算机平台将继续其引导程序。

[0046] 图3A示出了开始安全平台固件更新程序的过程的实施例的流程图。

[0047] 处理逻辑开始于接收更新平台固件的请求(处理块300)。这个请求可以涉及在整个计算机系统中的任意类型的可更新固件。因此,在一些实施例中,这个请求可以涉及系统管理固件;在其他的实施例中,这个请求可以涉及平台控制器固件;并且在另外的实施例中,甚至可以是存储在计算机系统/平台中的其他类型和单元的固件。

[0048] 通过处理逻辑将与请求相关联的固件镜像拷贝到系统存储器中的单元(处理块302)而继续所述过程。取决于更新的类型,镜像可以是整个固件或仅是一部分。在许多实施例中,镜像是加密的。加密的镜像可能已使用利用了公钥的非对称加密算法进行了加密。在建立系统时或建立系统稍后时间将公钥提供给计算机系统。可以将公钥安全地存储在不可写的存储器单元,例如系统管理固件存储的安全部分。返回到图3A,一旦已将固件镜像拷贝到系统存储器,处理逻辑使用存储在计算机系统中的安全单元中的可用公钥来认证镜像(处理块304)。

[0049] 可以由在系统中调用的并且在AC模式中运行的ACM中的逻辑来处理块304和在图3A中的块。参考图1在上文讨论了调用ACM。在许多实施例中,认证过程包括验证存储在系统中的不可变公钥与和镜像一起包括的公钥相匹配。可以使用在计算机系统中的安全过程来测量镜像。确定镜像真实性的过程可以采用数种形式,但是期望的结果是确认镜像没有受到危害。处理逻辑检查镜像是否已被验证为真实的(处理块306)。另外,虽然没有示出,但是潜在地以加密形式发送固件镜像,其可能需要ACM可访问的额外的解密密钥来允许在认证之前进行镜像的解密。

[0050] 如果镜像的认证失败,则处理逻辑向CPU发送错误消息(处理块308)。例如,处理逻辑可以启动中断来将失败/问题通知给镜像的认证。另一方面,如果成功认证镜像(即,镜像被验证为真实的),则处理逻辑可以进入到图3B中示出的过程。

[0051] 图3B示出了继续安全平台固件更新程序的过程的实施例的流程图。

[0052] 此时在处理流程中,处理逻辑(例如,在ACM中的逻辑)解锁平台固件存储单元来允许进行写入(处理块310)。在平台固件存储单元是系统管理固件存储单元(如参考图1在上文所述的)的实施例中,在验证了镜像的真实性(在图3A中的块306中)时,ACM内的逻辑继续进行解锁PCH以允许向系统管理固件存储单元的写入。在平台固件驻留在控制器存储中的其他实施例中,需要另一过程来继续向固件存储进行写入(在图4中描述的基于固件的控制器的过程)。

[0053] 返回到图3B,一旦解锁平台固件存储单元,处理逻辑将固件镜像拷贝到平台固件存储单元(处理块312)。最后,在将已认证的镜像拷贝到平台固件存储单元之后,处理逻辑接着锁定固件存储单元,使得不再允许写入固件存储单元(处理块314),并且结束过程。

[0054] 如上文进一步讨论的,平台固件存储解锁过程取决于固件存储的所在单元。在平台固件存储是系统管理固件存储单元的情况下,解锁过程可以要求在AC模式下运行的ACM(图1中的138)向CPU锁寄存器(图1中的150)进行写入。在许多实施例中,CPU锁寄存器是CPU(图1中的102)非内核中的MSR。仅被在AC模式中的ACM启动的特定写入MSR(WRMSR)命令向CPU锁寄存器写入。这个过程接着生成针对存在于PCH(图1中的118)中的PCH锁寄存器(图1中的114)的特定写入。针对PCH锁寄存器的写入命令可以致使PCH向系统管理固件存储(图1中的130)开放写入能力,这将允许ACM将新的系统管理固件镜像写入正确的存储单元。在结束写入镜像时,接着可以启动类似的WRMSR命令给同一CPU锁寄存器来重新锁定系统管理固

件存储。

[0055] 图4是安全地更新在计算机系统中的控制器固件的过程的实施例的流程图。在许多实施例中，在图4中示出的过程流包括用于ACM/CPU的处理逻辑以及用于正在讨论的控制器的处理逻辑。因此，为了使过程清楚到逻辑处理的每一个块，图4的左侧(粗短划线的左边)可以表示与ACM/CPU相关的逻辑，而且图4的右侧(粗短划线的右边)可以表示与控制器相关的逻辑。

[0056] 在计算机平台的引导序列期间，基于CPU的处理逻辑开始于创建短暂密码(处理块400)。短暂密码可由当前的系统管理固件或在计算机系统内的其他逻辑随机地生成。随后，这个随机密码将由处理逻辑本地存储在CPU中仅有ACM可访问的存储单元中，或存储在可被CPU访问的另一安全存储单元中(处理块402)。例如，在CPU中的临时密码寄存器(图1中的152)可以存储这个生成的短暂密码，并且这个寄存器只能被在AC模式中的ACM所访问。

[0057] 基于CPU的处理逻辑接着向控制器发送短暂密码(处理块404)。控制器处理逻辑仍然在系统引导序列期间接收短暂密码(处理块406)。在此时，控制器处理逻辑假定短暂密码是有效的，这是因为系统已经进入正常操作并且应该还没有机会受到危害。处理逻辑接着将接收到的短暂密码进行本地存储(处理块408)(例如，可将短暂密码存储在图1中的临时密码存储154中)。在此时，在控制器中的处理逻辑等待直到接收到固件更新请求。

[0058] 在此期间，返回到图3A，实体可以请求更新平台固件(图3A中的300)，并且基于CPU/ACM的处理逻辑将通过与图3A相关联的块来认证新的控制器专用平台固件镜像。一旦已认证镜像，ACM处理逻辑将处理在图3A中的下一个块，并向控制器发送请求来更新控制器固件，其包括在请求中的短暂密码(处理块410)。控制器逻辑接收具有由CPU存储的当前短暂密码的固件更新请求(处理块412)。接下来，控制器处理逻辑将刚接收到的当前短暂密码与在引导过程期间接收到的可信短暂密码进行比较(处理块414)。

[0059] 如果匹配(处理块416)，则控制器处理逻辑接着允许从ACM向控制器的固件存储进行写入，并且向ACM发送接受更新响应(处理块418)。基于ACM的处理逻辑通过继续在图3B中的处理块进行控制器更新来继续其更新处理。否则，如果不匹配，则控制器处理逻辑向ACM发送拒绝响应(同时继续不允许向控制器的固件存储进行写入)。基于ACM的逻辑接收拒绝响应并启动错误序列(处理块422)。错误序列可以是多样的，如生成中断、设置错误标记、致使系统关闭，或通过管理引擎(图1中的128)向信息技术管理员发送涉及错误的带外信息等其他响应。

[0060] 也可以将本发明的实施例的元件提供为用于存储机器可执行指令的机器可读介质。机器可读介质可以包括但不仅限于，闪存、光盘、只读光盘存储器(CD-ROM)、数字通用/视频盘(DVD)ROM、随机访问存储器(RAM)、可擦除可编程只读存储器(EPROM)、电可擦除可编程只读存储器(EEPROM)、磁卡或光学卡、传播媒介或其他类型的适用于存储电子指令的机器可读媒介。

[0061] 在以上说明书和权利要求书中，可以使用术语“包括”和“包含”以及它们的派生词，并且意图将它们用作彼此的同义词。此外，在以下说明书和权利要求书中，可以使用术语“耦合”和“连接”以及它们的派生词。应该理解，并不意图将这些术语用作彼此的同义词。相反，在特定的实施例中，“连接”可以被用来指示两个或更多个元件彼此之间是直接物理接触或电接触的。“耦合”可以意味着两个或更多个元件是直接物理接触或电接触的。然而，

“耦合”还可以意味着两个或更多的元件彼此之间不是直接接触的，但彼此之间仍然可以协作或交互。

[0062] 在以上的说明中，使用某个术语来描述本发明的实施例。例如，术语“逻辑”表示用于执行一个或多个功能的硬件、固件、软件(或其任意组合)。例如，“硬件”的例子包括但不限于：集成电路、有限状态机，或甚至是组合逻辑。集成电路可以采用以下形式：诸如微处理器的处理器、专用集成电路、数字信号处理器、微控制器等等。

[0063] 应当理解的是，在说明书中通篇提到“一个实施例”或者“实施例”意指结合该实施例描述的具体特征、结构或特性被包括在本发明的至少一个实施例中。因此，应当强调并理解的是，在说明书中各处两次或多次提到“实施例”或者“一个实施例”或者“替换实施例”并非必须全都指同一实施例。此外，所述具体的特征、结构或特性可以在本发明的一个或多个实施例中以任何适当的方式组合。

[0064] 类似地，应当理解的是，在前文对本发明的实施例的描述中，为了使得本公开流畅以帮助理解各个发明性方案中的一个或多个的目的，而有时将各种特征一起组合到单个实施例、图或其描述中。但是，不能将这种公开方法解释为反应了所要求保护的主题需要的特征比每个权利要求中明确表述的更多的意图。而是，根据所附权利要求所反应的，发明性的方案所依赖的特征少于单个在前文公开的实施例中的全部特征。因此，将随附于详细描述的权利要求明确地并入该详细描述中。

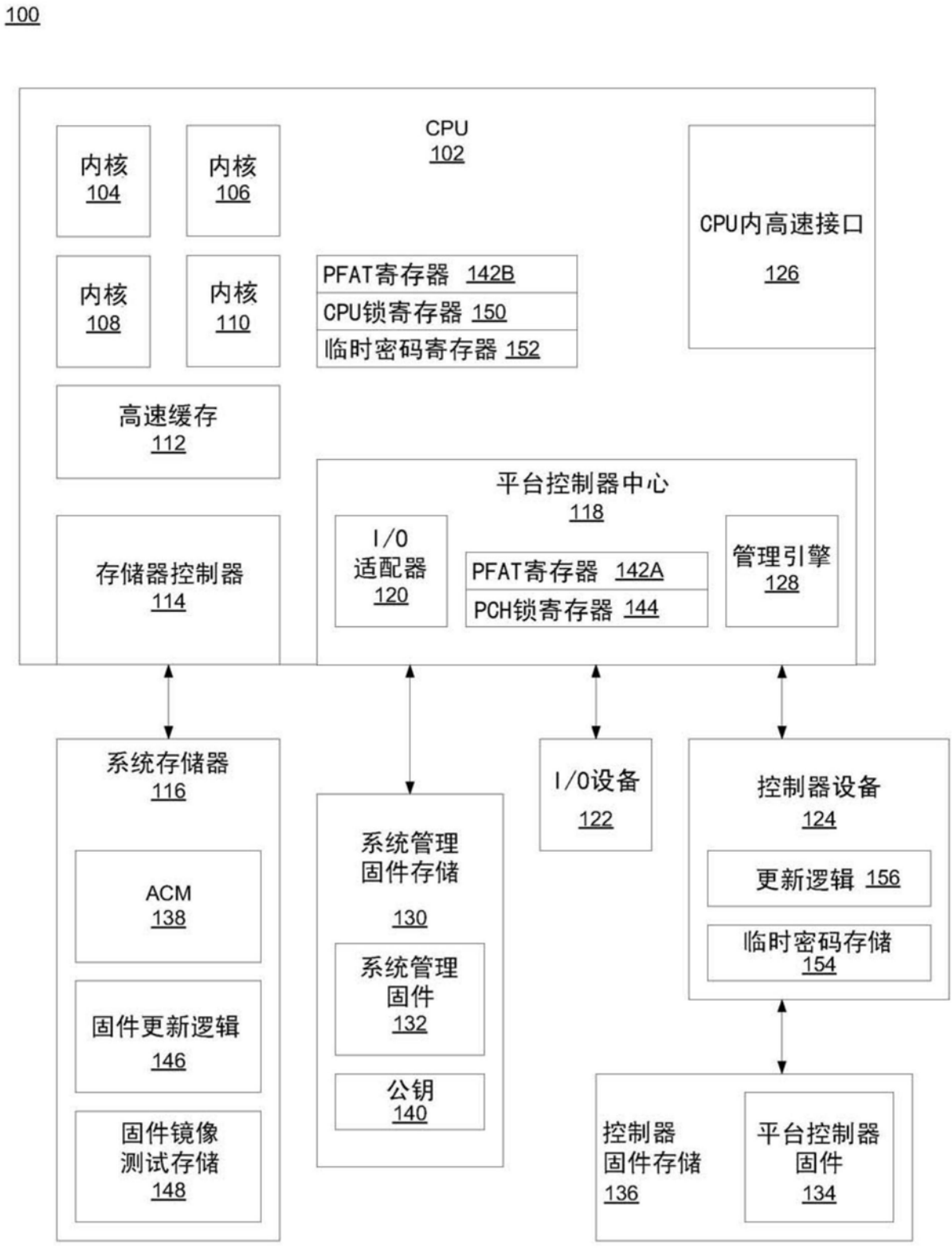

100
CPU
102
内核
104
内核
106
内核
108
内核
110
PFAT寄存器 142B
CPU锁寄存器 150
临时密码寄存器 152
CPU内高速接口
126
高速缓存
112
平台控制器中心
118
I/O
适配器
120
PFAT寄存器 142A
PCH锁寄存器 144
管理引擎
128
存储器控制器
114
系统存储器
116
ACM
138
固件更新逻辑
146
固件镜像
测试存储
148
系统管理
固件存储
130
系统管理
固件
132
公钥
140
I/O设备
122
控制器设备
124
更新逻辑 156
临时密码存储
154
控制器
固件存储
136
平台控制器
固件
134

图1


开始
设置PFAT模式启动位
200
设置PFAT模式锁定位
202
结束

图2

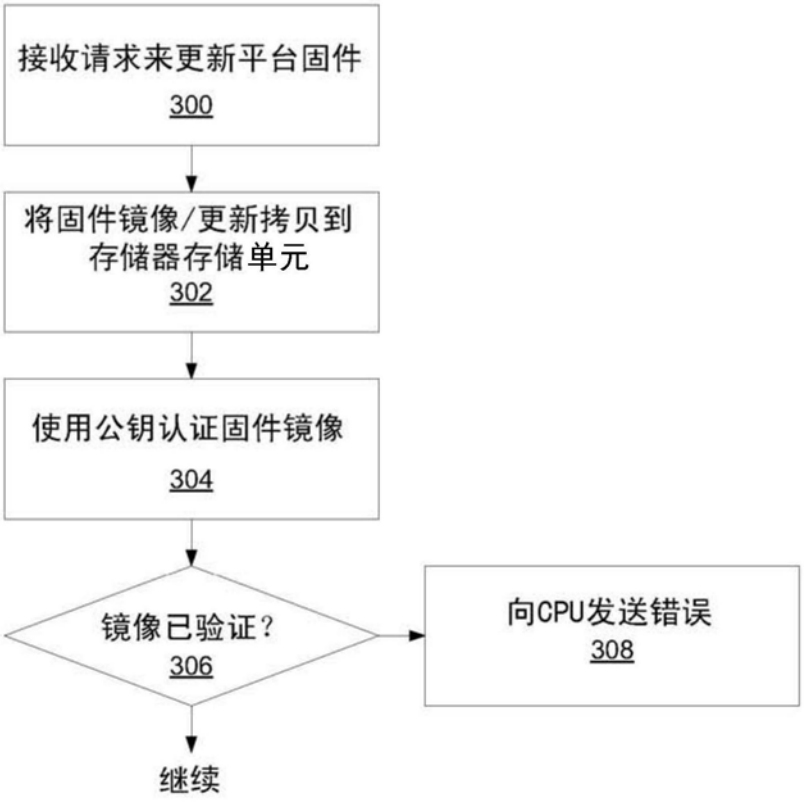

接收请求来更新平台固件
300
将固件镜像/更新拷贝到
存储器存储单元
302
使用公钥认证固件镜像
304
镜像已验证?
306
向CPU发送错误
308
继续

图3A

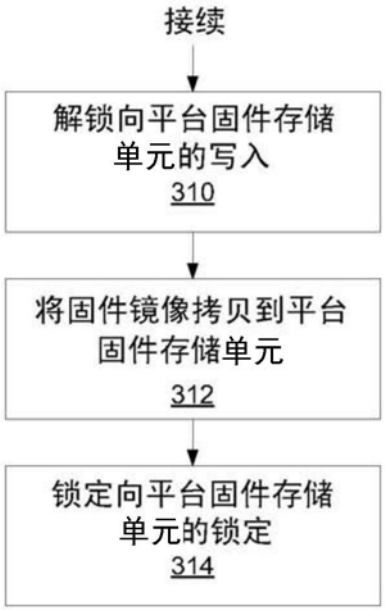

接续
解锁向平台固件存储单元的写入
310
将固件镜像拷贝到平台固件存储单元
312
锁定向平台固件存储单元的锁定
314

图3B

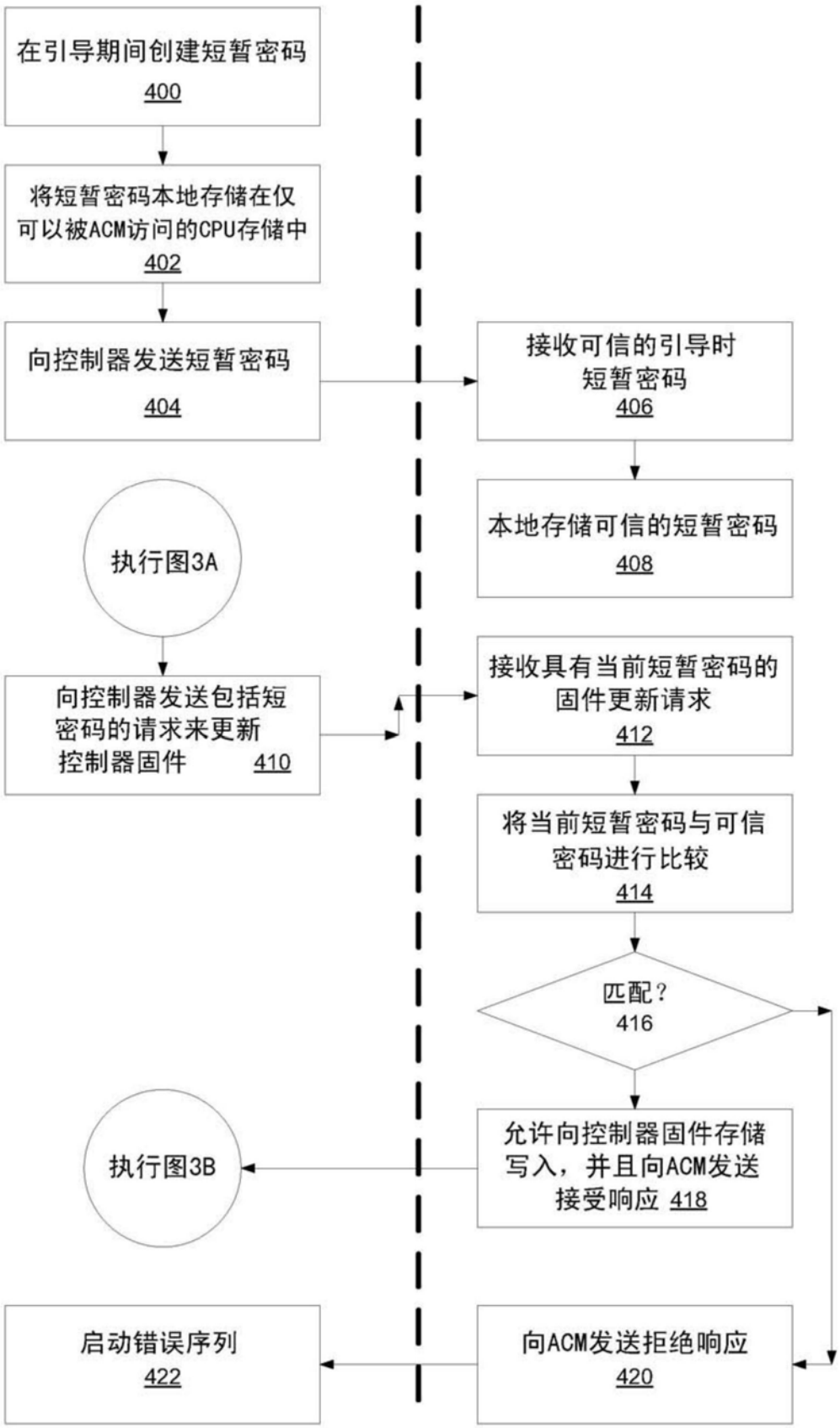

在引导期间创建短暂密码
400
将短暂密码本地存储在仅可以被ACM访问的CPU存储中
402
向控制器发送短暂密码
404
接收可信的引导时短暂密码
406
本地存储可信的短暂密码
408
执行图3A
向控制器发送包括短密码的请求来更新控制器固件
410
接收具有当前短暂密码的固件更新请求
412
将当前短暂密码与可信密码进行比较
414
匹配？
416
执行图3B
允许向控制器固件存储写入，并且向ACM发送接受响应
418
启动错误序列
422
向ACM发送拒绝响应
420

图4