(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 107851160 B
(45)授权公告日 2022.04.01

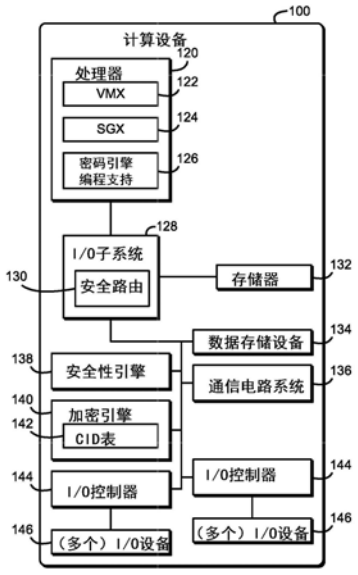
(21)申请号 201680042122.4
(22)申请日 2016.06.20
(65)同一申请的已公布的文献号
申请公布号 CN 107851160 A
(43)申请公布日 2018.03.27
(30)优先权数据
62/194,763 2015.07.20 US
62/195,148 2015.07.21 US
62/195,703 2015.07.22 US
14/974,948 2015.12.18 US
(85)PCT国际申请进入国家阶段日
2018.01.17
(86)PCT国际申请的申请数据
PCT/US2016/038392 2016.06.20
(87)PCT国际申请的公布数据
WO2017/014886 EN 2017.01.26
(73)专利权人 英特尔公司
地址 美国加利福尼亚州
(72)发明人 S·查博拉 R·拉尔 R·萨希塔 R·艾尔巴茨 幸滨
(74)专利代理机构 上海专利商标事务所有限公司 31100
代理人 李炜 黄嵩泉
(51)Int.Cl.
*G06F 21/60*(2006.01)
*G06F 13/28*(2006.01)
*G06F 9/455*(2006.01)
(56)对比文件
US 2007204073 A1,2007.08.30
CN 103795717 A,2014.05.14
US 2014188732 A1,2014.07.03
US 2012047580 A1,2012.02.23
US 7716389 B1,2010.05.11
CN 102986163 A,2013.03.20
US 2014095918 A1,2014.04.03 (续)
审查员 王春圆
权利要求书4页 说明书19页 附图6页

(54)发明名称

# 用于在ISA控制下进行多个共存可信执行环境的可信I/O的技术

(57)摘要

用于对加密引擎进行安全编程的技术包括具有加密引擎和一个或多个I/O控制器的计算设备。所述计算设备建立一个或多个可信执行环境(TEE)。TEE针对DMA通道生成对所述加密引擎进行编程的请求。所述计算设备可以验证已签名清单,所述已签名清单指示被准许对DMA通道进行编程的TEE,并且如果通过验证,则判定是否准许所述TEE对所请求的DMA通道进行编程。所述计算设备可以针对保护所述DMA通道的请求而记录所述TEE,并且可以针对解除保护DMA通道的请求而判定所述编程TEE是否匹配所记录的TEE。如果所述编程TEE匹配所记录的TEE,则所述计算设备可以允许要解除保护所述DMA通道的所述请求。描述并要求保护了其他实施例。




CN 107851160 B

[转续页]

[接上页]

(56)对比文件 CN 101621520 A,2010.01.06

US 2008282093 A1,2008.11.13

1.一种用于安全的加密引擎编程的计算设备,所述计算设备包括:

配置模块,用于由所述计算设备的可信执行环境请求对所述计算设备的加密引擎进行编程以保护直接存储器存取(DMA)通道;以及

清单验证模块,用于:(i)响应于对所述加密引擎进行编程的请求,验证所述计算设备的已签名清单,其中,所述已签名清单指示被准许对DMA通道进行编程的一个或多个可信执行环境,(ii)响应于对所述已签名清单的验证,判定是否准许所述可信执行环境基于所述已签名清单对所述DMA通道进行编程,以及(iii)响应于确定了所述可信执行环境被准许对所述DMA通道进行编程,对所述DMA通道进行编程。

2.如权利要求1所述的计算设备,其中,所述可信执行环境包括用所述计算设备的处理器的安全飞地支持建立的安全飞地、或虚拟机监视器。

3.如权利要求1所述的计算设备,进一步包括处理器,其中:

请求对所述计算设备的所述加密引擎进行编程以保护所述DMA通道包括:调用处理器指令来生成包封的编程信息;

所述处理器包括所述清单验证模块;并且

验证所述计算设备的所述已签名清单包括:响应于对所述处理器指令的调用而验证所述已签名清单。

4.如权利要求1所述的计算设备,其中:

请求对所述计算设备的所述加密引擎进行编程以保护所述DMA通道包括:由所述可信执行环境调用第一处理器指令来生成包封的编程信息,其中,所述包封的编程信息指示所述可信执行环境;并且

基于所述已签名清单来判定是否准许所述可信执行环境对所述DMA通道进行编程进一步包括:基于所述包封的编程信息来判定所述可信执行环境。

5.如权利要求4所述的计算设备,进一步包括处理器,其中:

生成对所述计算设备的所述加密引擎进行编程以保护所述DMA通道的请求包括:由所述计算设备的不可信软件利用所述包封的编程信息来调用第二处理器指令;并且

所述处理器包括所述清单验证模块。

6.如权利要求4所述的计算设备,其中,所述加密引擎包括所述清单验证模块。

7.一种用于安全的加密引擎编程的方法,所述方法包括:

由计算设备的可信执行环境生成对所述计算设备的加密引擎进行编程以保护直接存储器存取(DMA)通道的请求;

响应于生成对所述加密引擎进行编程的所述请求,由所述计算设备验证所述计算设备的已签名清单,其中,所述已签名清单指示被准许对DMA通道进行编程的一个或多个可信执行环境;

响应于验证所述已签名清单,由所述计算设备基于所述已签名清单来判定是否准许所述可信执行环境对所述DMA通道进行编程;以及

响应于确定了准许所述可信执行环境对所述DMA通道进行编程,由所述计算设备对所述DMA通道进行编程。

8.如权利要求7所述的方法,其中,所述可信执行环境包括用所述计算设备的处理器的安全飞地支持建立的安全飞地、或虚拟机监视器。

9.如权利要求7所述的方法,其中:

生成对所述计算设备的所述加密引擎进行编程以保护所述DMA通道的所述请求包括:调用处理器指令来生成包封的编程信息;

验证所述计算设备的所述已签名清单包括:响应于调用所述处理器指令,由所述计算设备的处理器来验证所述已签名清单;

基于所述已签名清单来判定是否准许所述可信执行环境对所述DMA通道进行编程包括:由所述处理器基于所述已签名清单来判定是否准许所述可信执行环境对所述DMA通道进行编程;并且

响应于确定了准许所述可信执行环境对所述DMA通道进行编程而对所述DMA通道进行编程包括:由所述处理器响应于确定了准许所述可信执行环境对所述DMA通道进行编程而生成所述包封的编程信息。

10.如权利要求7所述的方法,其中:

生成对所述计算设备的所述加密引擎进行编程以保护DMA通道的所述请求包括:由所述可信执行环境调用第一处理器指令来生成包封的编程信息,其中,所述包封的编程信息指示所述可信执行环境;并且

基于所述已签名清单来判定是否准许所述可信执行环境对所述DMA通道进行编程进一步包括:基于所述包封的编程信息来判定所述可信执行环境。

11.如权利要求10所述的方法,其中:

生成对所述计算设备的所述加密引擎进行编程以保护所述DMA通道的所述请求包括:由所述计算设备的不可信软件利用所述包封的编程信息来调用第二处理器指令;

验证所述计算设备的所述已签名清单包括:由所述计算设备的处理器响应于调用所述第二处理器指令而验证所述已签名清单;

基于所述已签名清单来判定是否准许所述可信执行环境对所述DMA通道进行编程包括:由所述处理器判定是否准许所述可信执行环境对所述DMA通道进行编程;并且

响应于确定了准许所述可信执行环境对所述DMA通道进行编程而对所述DMA通道进行编程包括:由所述处理器响应于确定了准许所述可信执行环境对所述DMA通道进行编程而对所述DMA通道进行编程。

12.如权利要求10所述的方法,其中:

验证所述计算设备的所述已签名清单包括:由所述加密引擎验证所述已签名清单;

基于所述已签名清单来判定是否准许所述可信执行环境对所述DMA通道进行编程包括:由所述加密引擎判定是否准许所述可信执行环境对所述DMA通道进行编程;并且

响应于确定了准许所述可信执行环境对所述DMA通道进行编程而对所述DMA通道进行编程包括:由所述加密引擎响应于确定了准许所述可信执行环境对所述DMA通道进行编程而对所述DMA通道进行编程。

13.一种用于安全的加密引擎编程的计算设备,所述计算设备包括:

配置模块,用于由可信执行环境针对直接存储器存取(DMA)通道生成对所述计算设备的加密引擎进行编程的请求;以及

所有权验证模块,用于:(i)判定对所述加密引擎进行编程的所述请求是否包括将所述DMA通道编程为缺乏安全的请求;(ii)响应于确定了对所述加密引擎进行编程的所述请求

包括将所述DMA通道编程为缺乏安全的请求，判定所述可信执行环境是否与同所述DMA通道相关联的所记录的可信执行环境相匹配，以及(iii)响应于确定了所述可信执行环境与同所述DMA通道相关联的所述记录的可信执行环境相匹配，允许对所述加密引擎进行编程的所述请求。

14.如权利要求13所述的计算设备，其中，所述可信执行环境包括用所述计算设备的处理器的安全飞地支持建立的安全飞地、或虚拟机监视器。

15.如权利要求13所述的计算设备，其中，对所述加密引擎进行编程的所述请求包括指示所述可信执行环境的包封的编程信息。

16.如权利要求13所述的计算设备，其中，所述加密引擎包括所述所有权验证模块。

17.如权利要求13所述的计算设备，进一步包括处理器，其中，所述处理器包括所述所有权验证模块。

18.如权利要求17所述的计算设备，其中：

生成对所述加密引擎进行编程的所述请求包括：(i)由所述可信执行环境调用第一处理器指令来生成包封的编程信息，其中，所述包封的编程信息指示所述可信执行环境，以及(ii)由所述计算设备的不可信软件利用所述包封的编程信息来调用第二处理器指令；并且

判定对所述加密引擎进行编程的所述请求是否包括将所述DMA通道编程为缺乏安全的请求包括：响应于对所述第二处理器指令的调用，由所述处理器判定对所述加密引擎进行编程的所述请求是否包括将所述DMA通道编程为缺乏安全的请求。

19.如权利要求13所述的计算设备，其中，判定所述可信执行环境是否与同所述DMA通道相关联的所述记录的可信执行环境相匹配包括：将同对所述加密引擎进行编程的所述请求相关联的通道编程密钥与同所述DMA通道相关联的加密密钥进行比较。

20.一种用于安全的加密引擎编程的方法，所述方法包括：

针对直接存储器存取(DMA)通道，由计算设备的可信执行环境生成对所述计算设备的加密引擎进行编程的请求；

由所述计算设备判定对所述加密引擎进行编程的所述请求是否包括将所述DMA通道编程为缺乏安全的请求；

响应于确定了对所述加密引擎进行编程的所述请求包括将所述DMA通道编程为缺乏安全的请求，由所述计算设备判定所述可信执行环境是否与同所述DMA通道相关联的所记录的可信执行环境相匹配；以及

响应于确定了所述可信执行环境与同所述DMA通道相关联的所述记录的可信执行环境相匹配，由所述计算设备允许对所述加密引擎进行编程的所述请求。

21.如权利要求20所述的方法，其中，所述可信执行环境包括用所述计算设备的处理器的安全飞地支持建立的安全飞地、或虚拟机监视器。

22.如权利要求20所述的方法，其中：

判定对所述加密引擎进行编程的所述请求是否包括将所述DMA通道编程为缺乏安全的请求包括：由所述加密引擎判定对所述加密引擎进行编程的所述请求是否包括将所述DMA通道编程为缺乏安全的请求；

判定所述可信执行环境是否与所述记录的可信执行环境相匹配包括：由所述加密引擎判定所述可信执行环境是否与所述记录的可信执行环境相匹配；并且

允许对所述加密引擎进行编程的所述请求包括:由所述加密引擎允许对所述加密引擎进行编程的所述请求。

23.如权利要求20所述的方法,其中:

判定对所述加密引擎进行编程的所述请求是否包括将所述DMA通道编程为缺乏安全的请求包括:由所述计算设备的处理器判定对所述加密引擎进行编程的所述请求是否包括将所述DMA通道编程为缺乏安全的请求;

判定所述可信执行环境是否与所述记录的可信执行环境相匹配包括:由所述处理器判定所述可信执行环境是否与所述记录的可信执行环境相匹配;以及

允许对所述加密引擎进行编程的所述请求包括:由所述处理器允许对所述加密引擎进行编程的所述请求。

24.如权利要求23所述的方法,其中:

生成对所述加密引擎进行编程的所述请求包括:(i)由所述可信执行环境调用第一处理器指令来生成包封的编程信息,其中,所述包封的编程信息指示所述可信执行环境,以及(ii)由所述计算设备的不可信软件利用所述包封的编程信息来调用第二处理器指令;并且

判定对所述加密引擎进行编程的所述请求是否包括将所述DMA通道编程为缺乏安全的请求包括:响应于对所述第二处理器指令的调用,由所述处理器判定对所述加密引擎进行编程的所述请求是否包括将所述DMA通道编程为缺乏安全的请求。

# 用于在ISA控制下进行多个共存可信执行环境的可信I/O的技术

[0001] 相关申请的交叉引用

[0002] 本申请要求于2015年12月18日提交的题为“TECHNOLOGIES FOR TRUSTED I/O FOR MULTIPLE COEXISTING TRUSTED EXECUTION ENVIRONMENTS UNDER ISA CONTROL(用于在ISA控制下的多个共存可信执行环境的可信I/O的技术)”的美国发明专利申请序列号14/974,948的优先权,所述美国发明专利申请根据35U.S.C.§119(e)要求以下各项的优先权:于2015年7月20日提交的题为“CRYPTOGRAPHIC PROTECTION OF I O DATA FOR DMA CAPABLE I/O CONTROLLERS(用于有DMA能力的I/O控制器的I/O数据的加密保护)”的美国临时专利申请序列号62/194,763;于2015年7月21日提交的题为“CRYPTOGRAPHIC PROTECTION OF I/O DATA FOR DMA CAPABLE I/O CONTROLLERS(用于有DMA能力的I/O控制器的I/O数据的加密保护)”的美国临时专利申请序列号62/195,148;以及于2015年7月22日提交的题为“TECHNOLOGIES FOR TRUSTED I/O FOR MULTIPLE TRUSTED EXECUTION ENVIRONMENTS UNDER PROCESSOR INSTRUCTION SET CONTROL(用于在处理器指令集控制下的多个可信执行环境的可信I/O的技术)”的美国临时专利申请序列号62/195,703。

## 背景技术

[0003] 对于安全性,典型的计算设备可以依靠软件代理,例如反恶意软件代理。但是,难以跟上用户设备上越来越多的恶意软件攻击。为了对抗恶意软件威胁,通过在可信执行环境(TEE)内部运行安全性敏感软件以保护安全性敏感软件是一种趋势。TEE提供了即使系统的其他部分受到损害仍可以保护秘密的通过安全检查的环境。TEE的示例包括Intel®软件防护扩展(Intel® SGX)、安全虚拟机(VM)、以及融合式安全引擎(CSE)。TEE虽然对保护TEE内的秘密很有用,但可能不保护被传达进入和/或离开安全“容器(container)”的I/O数据,例如用户数据和传感器数据。可信I/O的安全性要求因使用情况和设备而异,并且涉及保密性、完整性、活跃度、以及重放保护的风格和组合。

[0004] 某些系统可以使用虚拟机监视器(VMM)、管理程序、或其他虚拟化技术来提供可信执行环境。基于虚拟化的可信执行环境可以被称为VMM TEE。特别地,某些版本的Microsoft®Windows™操作系统可以包括基于虚拟化的环境。一些计算设备可以包括多个互不信任的可信执行环境。特别地,许多计算设备可以包括SGX TEE和VMM TEE。

## 附图说明

[0005] 在附图中通过示例的方式而不是通过限制的方式来展示了本文中所描述的概念。为了说明的简单和清楚起见,附图中所展示的元件不一定按比例绘制。在认为适当的情况下,在附图当中已经重复了参考标号以表示相应或相似的元件。

[0006] 图1是用于对中央加密引擎进行安全编程的计算设备的至少一个实施例的简化框图;

[0007] 图2是可由图1的计算设备建立的环境的至少一个实施例的简化框图;

[0008] 图3是可由图1至图3的计算设备建立的系统架构的至少一个实施例的简化框图；

[0009] 图4是可由图1至图2的计算设备执行的用于对中央加密引擎进行安全编程的方法的至少一个实施例的简化流程图；

[0010] 图5是可由图1至图2的计算设备执行的用于清单检查的方法的至少一个实施例的简化流程图；并且

[0011] 图6是可由图1至图2的计算设备执行的用于所有权检查的方法的至少一个实施例的简化流程图。

## 具体实施方式

[0012] 虽然本公开的概念易于经历各种修改和替代形式，但是在附图中已经通过示例的方式示出了其特定实施例并且将在本文中对其进行详细描述。然而，应当理解的是，不意在将本公开的概念限制于所公开的特定形式，而相反，意图是覆盖与本公开和所附权利要求书一致的所有修改形式、等效形式和替代形式。

[0013] 在说明书中提到的“一个实施例”、“实施例”、“说明性实施例”等指示所描述的实施例可以包括具体特征、结构或特性，但每一个实施例可能或者可能不一定包括所述具体特征、结构或特性。此外，此些短语不一定指相同的实施例。进一步地，当关于实施例而描述了特定特征、结构或特性时，应当认为的是，无论是否进行了明确描述，结合其他实施例来实现这种特征、结构或特性都在本领域的技术人员的知识内。另外，应当认识到，包括在采用“A、B和C中至少一个”形式的列表中的项可意指(A)；(B)；(C)；(A和B)；(A和C)；(B和C)；或(A、B和C)。类似地，采用“A、B或C中的至少一者”的形式列出的项可以意指(A)；(B)；(C)；(A和B)；(A和C)；(B和C)；或(A、B和C)。

[0014] 在一些情况下，可以在硬件、固件、软件或其任何组合中实施所公开的实施例。所公开的实施例还可以被实施为由一个或多个暂态或非暂态机器可读(例如，计算机可读)存储介质所承载的或存储于其上的指令，所述指令可以由一个或多个处理器读取和执行。机器可读存储介质可以被实施为任何存储设备、机制、或用于存储或传输采用机器可读形式的信息的其他物理结构(例如，易失性或非易失性存储器、介质盘或其他介质设备)。

[0015] 在附图中，可以采用特定安排和/或排序来示出一些结构特征或方法特征。然而，应当理解的是，可能不需要这种特定的安排和/或排序。相反，在一些实施例中，可以采用与在说明性附图中所示出的方式和/或顺序不同的方式和/或顺序来安排这种特征。另外，在特定的图中包括结构性特征或方法特征并不意味着暗示在所有实施例中都需要这种特征，并且在某些实施例中，可以不包括这种特征或者这种特征可以与其他特征组合。

[0016] 现在参考图1，在说明性实施例中，用于安全I/O的计算设备100包括处理器120、主存储器132、硬件加密引擎140、以及与一个或多个I/O设备146通信的一个或多个I/O控制器144等其他组件。在使用中，加密引擎140在平台I/O控制器144与存储器132之间提供经由直接存储器存取操作(DMAed)传送的数据的即时加密和解密。每个DMA事务用表示与特定I/O设备146或I/O设备组146相关联的数据流的通道ID(CID)来标记。加密引擎140使用CID来可靠地标识必须被保护的事务，检索相应的加密密钥，并且对DMA数据执行适当的加密操作。由可信软件使用处理器120的一个或多个专用指令，例如用通道信息和相关联的加密密钥对加密引擎140进行编程。计算设备100还建立多个互不信任的可信执行环境，如安全飞地

和VMM。在一些实施例中，计算设备100可以准许每个TEE基于系统清单来控制一个或多个DMA通道。另外地或可替代地，在一些实施例中，计算设备100可以跟踪每个DMA通道的所有权，并且仅准许拥有特定DMA通道的TEE使DMA通道缺乏安全。因此，计算设备100可以允许多个TEE对加密引擎140进行编程，同时防止不可信软件(包括不同的TEE)破坏对加密引擎140的编程。因此，计算设备100可以使得中央密码引擎140的硬件可信I/O能力可用于计算设备100的任何TEE。另外，在计算设备100中，每个TEE是在计算设备100上的其他TEE的可信代码基(TCB)之外。

[0017] 计算设备100可以被实施为能够执行在此所描述的功能的任何类型的计算设备或计算机设备，包括但不限于计算机、台式计算机、工作站、服务器、膝上型计算机、笔记本计算机、平板计算机、移动计算设备、可穿戴计算设备、网络电器、web电器、分布式计算系统、基于处理器的系统和/或消费电子设备。如图1中所示，计算设备100示意性地包括处理器120、输入/输出子系统128、存储器132、数据存储设备134以及通信电路系统136。当然，在其他实施例中，计算设备100可以包括其他或附加组件，如台式计算机中常见的那些组件(例如，各种输入/输出设备)。另外，在一些实施例中，说明性组件中的一个或多个说明性组件可以结合在另一组件中，或以其他方式形成其一部分。例如，在一些实施例中，存储器132或其一部分可以结合到处理器120中。

[0018] 处理器120可以被实施为能够执行在本文中所描述的功能的任何类型的处理器。处理器120可以被实施为(多个)单核或多核处理器、数字信号处理器、微控制器或其他处理器或处理/控制电路。如所示，处理器120可以包括硬件虚拟化支持122、安全飞地支持124、以及密码引擎编程支持126。

[0019] 硬件虚拟化支持122支持计算设备100对操作系统、应用、以及其他软件的虚拟化执行。硬件虚拟化支持122可以通过提供两种执行模式来包括虚拟机扩展(VMX)支持：VMX根模式和VMX非根模式。VMX根模式允许执行软件具有对计算设备100及其硬件资源的广泛控制。管理程序、虚拟机监视器(VMM)或主机操作系统(OS)可以在VMX根模式下执行。VMX非根模式限制访问某些硬件指令，同时仍实施处理器120的普通环/权限系统。一个或多个客户OS可以在VMX非根模式下执行。类似于在没有虚拟化的情况下的执行，那些客户OS可以在环零中执行。硬件虚拟化支持122还可以支持扩展页表(EPT)，所述EPT可以被实施为硬件辅助的第二级页地址转换。硬件虚拟化支持122可以被实施为例如Intel®VT-x技术。

[0020] 安全飞地支持124允许处理器120建立被称为安全飞地的可信执行环境，在所述可信执行环境中可以测量、验证和/或以其他方式确定执行代码是真实的。此外，安全飞地中所包括的代码和数据可以被加密或以其他方式被保护不被在安全飞地以外执行的代码访问。例如，安全飞地中包含的代码和数据可以在被执行或者被存储在处理器120的某个受保护的高速缓冲存储器中的同时由处理器120的硬件保护机制来保护。安全飞地中包含的代码和数据可以在被存储在共享高速缓存或主存储器132中时被加密。安全飞地支持124可以被实施为一组处理器指令扩展，其允许处理器120在存储器132中建立一个或多个安全飞地。例如，安全飞地支持124可以被实施为为Intel®软件防护扩展(SGX)技术。

[0021] 密码引擎编程支持126允许处理器120对加密引擎140进行编程以提供对I/O数据的加密保护。特别地，处理器120可以启用或禁用对某些I/O通道的加密，并且可以向加密引擎140安全地提供加密密钥。密码引擎编程支持126可以被实施为一个或多个专用处理器指

令(例如,指令EBINDTIO、UNWRAP、或其他指令)以及处理器120的相关联的硬件、微码、固件或其他组件。处理器120的密码引擎编程支持126可以允许可信软件对加密引擎140进行编程,同时防止不可信软件对加密引擎140进行编程。

[0022] 存储器132可以被实施为能够执行此处所描述的功能的任何类型的易失性或非易失性存储器或数据存储设备。在运算中,存储器132可存储在计算设备100运算期间使用的各种数据和软件,如运算系统、应用、程序、函数库和驱动程序。存储器132以通信方式经由I/O子系统128耦合到处理器120,所述I/O子系统128可被实施为电路系统和/或组件以促进与计算设备100的处理器120、存储器132和/或其它组件的输入/输出操作。例如,I/O子系统128可以被实施为或以其他方式包括用于促进输入/输出操作的存储器控制器中枢、输入/输出控制中枢、平台控制器中枢、集成控制电路系统、固件设备、通信链路(即,点到点的链路、总线链路、导线、线缆、光导、印刷电路板迹线等)和/或其他组件及子系统。I/O子系统128可以进一步包括安全路由支持130。安全路由支持130包括用于确保在流氓软件的影响下I/O数据无法在结构128中误传的硬件支持。安全路由支持130可以与加密引擎140一起使用以提供I/O数据的加密保护。在一些实施例中,I/O子系统128可以形成片上系统(SoC)的一部分并且可以与计算设备100的处理器120、存储器132以及其他组件一起结合在单个集成电路芯片上。

[0023] 数据存储设备134可以被实施为被配置用于对数据进行短期或长期存储的任何类型的一个或多个设备(如例如,存储器设备和电路、存储器卡、硬盘驱动器、固态驱动器或其他数据存储设备)。在一些实施例中,数据存储设备134可用于存储一个或多个安全飞地的内容。安全飞地的内容当由数据存储设备134存储时可被加密以防止未授权的访问。

[0024] 计算设备100的通信电路系统136可以被实施为能够通过网络实现计算设备100与其他远程设备之间的通信的任何通信电路、设备或其集合。通信电路系统136可以被配置用于使用任何一种或多种通信技术(例如,有线或无线通信)以及相关联的协议(例如,以太网、Bluetooth®、 Wi-Fi®、WiMAX等)来实现这种通信。

[0025] 在一些实施例中,计算设备100可以包括安全性引擎138,所述安全性引擎可以被实施为能够对计算设备100提供的安全相关的服务的任何(多个)硬件组件或电路系统。特别地,安全性引擎138可以包括能够独立地且安全地执行来自处理器120的固件和/或其他代码的微处理器、微控制器、或其他嵌入式控制器。因此,安全性引擎138可以被用来建立与由处理器120执行的代码分开的可信执行环境。安全性引擎138可以通过诸如主机嵌入式控制器接口(HECI)等专用总线与计算设备100的处理器120和/或其他组件进行通信。安全性引擎138还可以提供对计算设备100的远程配置、控制或管理。在所示实施例中,安全性引擎138被实施为并入计算设备100的片上系统(SoC)中的融合式安全和管理引擎(CSME)。在一些实施例中,安全性引擎138可以被实施为可管理性引擎、带外处理器、可信平台模块(TPM)、或其他安全性引擎设备或设备集合。此外,在一些实施例中,安全性引擎138还能够使用通信电路系统136或独立于计算设备100的状态(例如,独立于主处理器120的状态)的专用通信电路进行通信,也被称为“带外”通信。

[0026] 加密引擎140可以被实施为能够执行本文所描述的功能的任何微控制器、微处理器、功能块、逻辑或其他电路或电路集合。在对存储器132的一个或多个直接存储器存取(DMA)操作中,加密引擎140可以对由I/O控制器144读取或写入的I/O数据进行加密和/或解

密。加密引擎140包括内部通道标识符(CID)表142,加密引擎140使用所述表来动态地标识待保护的(多个)DMA通道。可以由可信软件例如使用处理器120的密码引擎编程支持126来控制和/或编程所述CID表142。CID表142的加密密钥和/或其他秘密信息对于不可信软件是不可用的。在一些实施例中,加密引擎140可以与I/O子系统128和/或处理器120一起并入计算设备100的片上系统(SoC)中。

[0027] 类似地,I/O控制器144可以被实施为能够执行本文所描述的功能的任何嵌入式控制器、微控制器、微处理器、功能块、逻辑或其他电路或电路集合。在一些实施例中,I/O控制器144中的一个或多个可以被嵌入到计算设备100的另一组件中,如I/O子系统128和/或处理器120。另外或可替代地,I/O控制器144中的一个或多个可以经由诸如PCI Express(PCIe)或其他I/O连接等扩展总线连接到I/O子系统128和/或处理器120。如下面进一步描述的,I/O控制器144例如通过外围通信总线(例如,USB、蓝牙等)与一个或多个I/O设备146通信。I/O设备146可以被实施为任何I/O设备,如人机界面设备、键盘、鼠标、触摸屏、麦克风、相机和其他输入设备、以及显示器和其他输出设备。如上所述,使用被称为通道标识符(CID)的标识符来唯一地标识I/O控制器144和相关联的DMA通道(其对应于附接的I/O设备146)。每个I/O控制器144可以用每个DMA事务(例如作为事务层分组(TLP)前缀的一部分)来断言合适的CID,以便唯一地标识DMA事务的来源并提供活跃度保护。CID还使得I/O能够与不同设备146隔离开。

[0028] 在使用中,加密引擎140监听由I/O控制器144生成到存储器132中的所有DMA事务。在去往或来自能够参与可信I/O的设备146的每个事务中,加密引擎140引用CID表142以在CID表142中找到对应于DMA通道的CID。匹配表示通道当前受保护,并且加密引擎140应使用与通道相关联的通道密钥来保护被写入到存储器132和/或从存储器132中读取的数据(取决于通道的方向)。

[0029] 现在参照图2,在说明性实施例中,计算设备100在操作期间建立环境200。说明性环境200包括配置模块202、清单验证模块204、以及所有权验证模块206。环境200的不同模块可以被实施为硬件、固件、微码、软件或其组合。这样,在一些实施例中,环境200的模块中的一个或多个模块可以被实施为电子设备的电路系统或集合(例如,配置电路系统202、清单验证电路系统204、以及所有权验证电路系统206)。应当理解,在这样的实施例中,配置电路系统202、清单验证电路系统204、和/或所有权验证电路系统206中的一者或多者可以形成计算设备100的处理器120、I/O子系统128、加密引擎140、和/或其他组件中的一者或多者的一部分。另外,在一些实施例中,说明性模块中的一个或多个说明性模块可以形成另一个模块的一部分和/或说明性模块中的一个或多个说明性模块可以彼此独立。

[0030] 配置模块202被配置用于由计算设备100的可信执行环境请求针对直接存储器存取(DMA)通道对加密引擎140进行编程(例如,以保护DMA通道或解除保护DMA通道)。可信执行环境可以被实施为例如用处理器120的安全飞地支持124建立的安全飞地、或计算设备100的虚拟机监视器(VMM)。配置模块202可以被进一步被配置用于向计算设备100安全地供应已签名清单,所述已签名清单指示被准许对DMA通道进行编程的一个或多个可信执行环境。在一些实施例中,配置模块202可以被配置用于将所述已签名清单的地址编程到加密引擎140中。

[0031] 清单验证模块204被配置用于响应于对加密引擎140进行编程的请求而验证计算

设备100的已签名清单。清单验证模块204被进一步配置用于如果已签名清单通过验证，则判定是否准许发起编程请求的可信执行环境基于所述已签名清单对所请求的DMA通道进行编程，并且如果可信执行环境被准许则对所请求的DMA通道进行编程。如果可信执行环境不被准许对DMA通道进行编程，则清单验证模块204可以被配置用于放弃对加密引擎140进行编程的请求。

[0032] 所有权验证模块206被配置用于判定对加密引擎140进行编程的请求是将DMA通道编程为安全的请求还是将DMA通道编程为缺乏安全的请求。关于将DMA通道编程为安全的请求，所有权验证模块206被配置用于记录所述编程可信执行环境的标识符。关于将DMA通道编程为缺乏安全的请求，所有权验证模块206被配置用于判定所述编程可信执行环境是否与所记录的可信执行环境相匹配。所有权验证模块206被配置用于如果编程可信执行环境与所记录的可信执行环境相匹配，则允许对加密引擎140进行编程的请求，并且如果编程可信执行环境与所记录的可信执行环境不匹配，则放弃对加密引擎140进行编程的请求。

[0033] 现在参考图3，图示300展示了可由计算设备100建立的系统架构。所述系统架构可以包括不可信I/O堆栈，所述不可信I/O堆栈包括应用302、设备驱动器304、过滤驱动器306、以及总线驱动器308。不可信I/O堆栈可以经由加密引擎140从I/O控制器144接收未受保护的(即，明文)I/O数据，并且照常处理I/O数据。所述系统架构还可以包括可信I/O堆栈，所述可信I/O堆栈包括应用飞地310和设备驱动器飞地(DDE)312。飞地310、312各自可以使用处理器120的安全飞地支持124来建立，并且因此可以是可信的。如所示，飞地310、312各自可以供应有与一个或多个DMA通道相关联的加密密钥。因此，应用飞地310和/或DDE 312可以安全地解密和处理经由加密引擎140从I/O设备146生成的安全I/O数据。如所示，安全飞地310、312可以经由不可信I/O堆栈的多个部分(如总线驱动器308和/或过滤驱动器306)接收安全I/O数据。特别地，I/O控制和路由可以由不可信I/O堆栈执行，并且因为安全I/O数据的有效载荷被加密，因此安全I/O数据仍然受到保护。因此，不可信I/O堆栈不需要被包括在计算设备100的TCB中。在一些实施例中，包括过滤驱动器306、总线驱动器308和/或其他不可信I/O组件的不可信I/O堆栈可以重新使用或以其他方式与计算设备100的普通操作系统共享。

[0034] 如所示，系统架构300进一步包括一个或多个可执行环境314、密码引擎驱动器316、以及解包封引擎318，所述解包封引擎可以用于对加密引擎140进行编程。可信执行环境314中的每一个可以被实施为保持或以其他方式访问与一个或多个DMA通道相关联的加密密钥的可信代码。每个可信执行环境314与其他可信执行环境314互不信任；也就是说，每个可信执行环境314在其他可信执行环境314的可信赖代码基之外。例如，说明性实施例包括两个可信执行环境314a、314b。可信执行环境314a可以被实施为虚拟机监视器、管理程序或受处理器120的虚拟化支持122保护的其他系统管理代码。可信执行环境314b可以被实施为安全飞地，即用处理器120的安全飞地支持124保护的用户级(例如，环3)代码。如下面进一步描述的，可信执行环境314可以向可信I/O堆栈供应加密密钥和/或使用处理器120的密码引擎编程支持126对加密引擎140进行编程。在一些实施例中，可信执行环境314还可以应付来自可信应用的通道编程请求。特别地，可信执行环境314可以执行一个或多个专用处理器指令以准备包括包封的通道编程信息的二进制blob，所述通道编程信息包括可用于对加密引擎140进行编程的包封的加密密钥。

[0035] 可信执行环境314可以将二进制blob提供给密码引擎驱动器316，所述密码引擎驱动器可以被实施为内核级不可信软件组件。密码引擎驱动器316将二进制blob提供给解包封引擎318，所述解包封引擎可以对二进制blob进行解包封和验证，并且如果通过验证，则将通道编程信息编程到加密引擎140中。因此，密码引擎驱动器316可以允许计算设备100的操作系统、VMM或其他控制软件来管理对加密引擎140的编程，而不需要操作系统访问用于DMA通道的明文加密密钥。在说明性实施例中，解包封引擎318由处理器120的硬件和/或微码资源提供；然而，在一些实施例中，解包封引擎318的功能可以由加密引擎140或计算设备100的其他组件执行。

[0036] 现在参考图4，在使用中，计算设备100可以执行用于对加密引擎140进行安全编程的方法400。方法400可以由计算设备100的硬件、固件、处理器微码、软件或其他执行资源来执行。方法400始于框402，在所述框中，可信执行环境(TEE)314生成用于对DMA通道进行编程的密钥。TEE 314可以被实施为计算设备100的任何可信组件，例如安全飞地(例如，密码引擎飞地(CEE)或其他安全飞地)、虚拟机监视器或管理程序、或其他可信组件。所述密钥可以包括用于保护通过DMA通道传输的I/O数据的通道密钥。因此，TEE 314还可以将通道密钥提供给计算设备100的可以访问受保护I/O数据的应用飞地310、设备驱动器飞地312、和/或其他可信组件。TEE 314不向其他TEE 314提供所述通道密钥。例如，安全飞地TEE 314可以不将所述密钥提供给计算设备100的VMM。

[0037] 在框404中，TEE 314将通道编程信息准备用于对DMA通道进行编程。所述通道编程信息可以包括如在框402中确定的加密密钥以及其他编程信息，如有待编程的DMA通道的通道标识符(CID)、编程命令、可以用于认证和重放保护的随机现时值、以及其他编程信息。CID可以被实施为唯一地标识与计算设备100的I/O控制器144相连接的特定I/O设备146的标识符。例如，CID可以包括标识I/O控制器144的控制器标识符字段以及标识I/O设备146的设备号字段。为了准备编程信息，TEE 314例如可以在存储器中分配用于存储所述编程信息的结构，也被称为“二进制blob”。在一些实施例中，所述编程信息可以存储在被称为BIND_STRUCT的结构中，所述结构可以包括下面在表1中描述的字段。

[0038]

| 偏移名称 | 偏移量 | 大小（B） | 描述 | 由…设定 |
|---|---|---|---|---|
| BTID | 0 | 4 | 目标设备 | 软件 |
| BTSVN | 4 | 4 | 目标安全性版本号 | 软件 |
| BTDATA | 8 | 24 | 目标特定的数据（例如包括CID） | 软件 |
| TKEY | 32 | 16 | 用于目标的通道密钥 | 软件 |
| NONCE | 48 | 8 | 用于认证响应的现时值 | 软件 |
| SEQID | 56 | 8 | 用于生成初始化向量（IV）的种子 | 硬件 |
| WRAPPING_TEE | 64 | 4 | TEE 生成经包封的通道信息（0：SGX，1：VMM） | 硬件 |
| MAC | 68 | 16 | 加密密钥、策略、目标ID、SVN、BTDATA、NONCE、SEQID、以及 WRAPPING_TEE 上的MAC | 硬件 |
| RSVD | 84 | 44 | 保留 | 硬件 |

[0039] 表1.绑定密钥结构(BIND_STRUCT)。

[0040] 如所示,BIND_STRUCT结构可以包括由处理器120的硬件设定的字段,包括序列号(SEQID)、消息认证码(MAC)、以及WRAPPING_TEE字段。WRAPPING_TEE字段可以由处理器120的微码设定,以指示调用TEE 314。这个字段仅可由硬件设定以允许调用TEE 314被安全地传达到加密引擎140。除了填充WRAPPING_TEE字段之外,处理器120的微码可以在MAC计算中包括这个字段以确保不可信软件无法在没有检测的情况下修改这个字段。下面进一步描述由处理器120生成这些字段。在说明性实施例中,WRAPPING_TEE字段支持标识两个TEE 314中的一个(说明性地,Intel®SGX安全飞地或VMM)。然而,应当理解的是,在一些实施例中,可以扩展WRAPPING_TEE字段以支持附加的TEE 314。

[0041] 仍然参考图4,在框406中,TEE 314调用处理器120的处理器指令来生成包封的编程信息。例如,TEE 314可以调用EBINDTIO指令和/或BINDTIO指令来生成包封的编程信息。TEE 314可以将包括通道编程的BIND_STRUCT作为参数传递给所述处理器指令。处理器120对通道编程信息的密钥进行加密以生成加密密钥。处理器120可以使用仅由处理器120和解包封引擎318(其可以被实施为处理器120和/或加密引擎140)已知的密钥包封密钥(KWK)来对密钥进行加密。处理器120可以生成序列号并将其包括在包封的编程信息中以用于重放保护的,并且处理器120可以在通道编程信息上生成用于完整性保护的MAC。

[0042] 在一些实施例中,在框408中,处理器120可以在包封的编程信息中标识编程TEE 314。例如,如上面结合表1所描述的,处理器120可以在包封的编程信息中设定一个位以用于指示调用TEE 314是VMM还是安全飞地。处理器120可以使用任何适当的技术来确定调用TEE 314的标识。例如,处理器120可以判定处理器120当前是否在安全飞地中执行,或者处理器120可以判定处理器120是否在VMX根模式下执行。处理器120可以修改BIND_STRUCT以包含所述包封的编程信息。处理器120可以在包封TEE位上生成MAC,以便完整地保护调用TEE 314的标识。

[0043] 在一些实施例中,在框410中,计算设备100可以将处理器120配置用于允许从虚拟机监视器(VMM)或不是安全飞地的其他TEE 314中调用EBINDTIO指令。例如,处理器120可以默认地不允许执行来自安全飞地之外的EBINDTIO、EBIND或类似指令。在一些实施例中,处理器120可以建立被称为IA32_EBIND_VMM_ENABLE的模型特定的寄存器(MSR),所述模型特定寄存器可以由BIOS(或其他平台固件)使用以指示允许从VMM内执行EBINDTIO。在执行EBINDTIO或EBIND指令期间,处理器120可以检查所述指令是从安全飞地还是VMM内部执行,并且可以允许来自VMM的编程尝试仅在IA32_EBIND_VMM_ENABLE MSR被所述固件适当地设定的情况下成功。尽管被展示为在方法400的执行期间进行配置,但是应当理解,在一些实施例中,处理器120可以在不同的时间进行配置,例如在执行预引导固件环境期间。

[0044] 在一些实施例中,在框412中,计算设备100可以验证编程TEE 314被授权对与所述编程信息相关联的DMA通道进行编程。特别地,计算设备100可以判定TEE 314是否已经被已安全地供应到计算设备100的系统清单所授权。所述系统清单可以描述可由每个TEE 314控制的分开的设备组。以下结合图5描述了用于验证编程TEE 314被授权对DMA通道进行编程的方法的一个可能实施例。

[0045] 在生成包封的编程信息之后,在框414中,TEE 314将所述包封的编程信息提供给计算设备100的不可信软件,如加密引擎驱动器316。由于所述包封的编程信息已经被加密

并且被绑定到解包封引擎318，通道编程信息中的敏感数据(例如，通道编程密钥)可以不被不可信软件访问。不可信软件可以检查所述包封的编程信息的未受保护的字段(例如，BTDATA字段)，以判定是否允许编程尝试。因此，诸如密码引擎驱动器316的内核模式软件可以管理加密引擎140的编程，而无需被信任或以其他方式能够访问受保护的I/O数据。

[0046] 在框416中，不可信软件(例如，密码引擎驱动器316)调用解包封引擎318以对编程信息解除包封并且将DMA通道安全地编程到加密引擎140中。解包封引擎318可以将通道编程信息复制到加密引擎140的CID表142的适当条目中。在一些实施例中，在框418中，密码引擎驱动器316可以调用诸如UNWRAP指令的处理器指令，所述处理器指令使解包封引擎318(例如，处理器120和/或加密引擎140)对通道编程密钥进行解密、验证通道编程信息、或以其他方式对DMA通道进行编程。UNWRAP指令可以被实施为内核级(例如，环0)指令。在一些实施例中，UNWRAP指令可以生成虚拟机退出(VMExit)，从而允许VMM和/或管理程序管理UNWRAP指令的虚拟化。通过允许VMM管理加密引擎140的编程，UNWRAP指令允许计算设备100的最高权限组件(例如，VMM)对DMA通道资源的使用具有最终控制。因此，UNWRAP指令可以防止权限反转，例如通过允许VMM判定是否允许用户级安全飞地314安全地访问DMA通道，而不将安全飞地314的秘密暴露给VMM。

[0047] 在一些实施例中，在框420中，计算设备100可以验证编程TEE 314被授权对与所述编程信息相关联的DMA通道进行编程。特别地，计算设备100可以判定TEE 314是否已经被已安全地供应到计算设备100的系统清单所授权。以下结合图5描述了用于验证编程TEE 314被授权对DMA通道进行编程的方法的一个可能实施例。另外或可替代地，在一些实施例中，计算设备100可以跟踪和验证所编程的DMA通道的所有权。特别地，计算设备100可以在将DMA通道编程为安全时记录编程TEE 314的标识，并且可以验证同一TEE 314生成将DMA通道编程为缺乏安全的请求。以下结合图6描述了用于验证DMA通道的所有权的方法的一个可能实施例。

[0048] 在对加密引擎140进行编程之后，加密引擎140可以生成可由TEE 314使用的经认证的响应，以验证通道编程是成功的。在编程之后，方法400循环回到框402，在所述框中，一个或多个TEE 314可以生成附加的编程请求。

[0049] 现在参考图5，在使用中，计算设备100可以执行用于系统清单检查的方法500。方法500可以由计算设备100的硬件、固件、处理器微码、软件或其他执行资源来执行。例如，在一些实施例中，可以由处理器120的微码响应于如上面结合图4的框412所描述的EBINDTIO指令的调用和/或响应于如上结合图4的框420所描述的UNWRAP指令的调用而执行方法500。另外或可替代地，在一些实施例中，可以由加密引擎140响应于如上结合图4的框420所描述的编程尝试而执行方法500。

[0050] 方法500开始于框502，在所述框中，设备制造商或其他可信实体将已签名的系统清单供应到计算设备100。例如用制造商的私人密钥对系统清单进行签名，以允许计算设备100验证所述系统清单并检测对所述系统清单进行修改的尝试。所述清单可以包括I/O设备146的列表(以通道ID的形式)和被允许控制I/O设备146的相关联的TEE 314。例如，在一些实施例中，生物计量传感器设备146可以单独由计算设备100的虚拟机监视器(VMM)控制，从而允许VMM使用生物计量传感器设备146来进行安全的用户认证和登录。可以使用任何安全供应技术将系统清单供应到计算设备100。另外，虽然被展示为在方法500的执行期间发生，

但是可以在任何适当的时间供应所述清单，例如在计算设备100的制造、整合、和/或初始配置期间。

[0051] 在框504中，计算设备100对系统清单的地址进行配置。计算设备100可以将系统清单的地址配置成使得它对于负责读取和/或验证所述清单的计算设备100的组件(如加密引擎140或处理器120)是可用的。例如，计算设备100的平台固件可以在执行预引导固件环境期间将系统清单的地址编程到加密引擎140中。所述地址可以指向系统清单的任何适当的存储位置，如存储器132的固件保留部、数据存储设备134的固件卷、和/或其他存储位置。尽管被展示为在方法500的执行期间发生，但是清单的地址可以在任何适当的时间被编程，例如在计算设备100的引导期间。

[0052] 在框506中，计算设备100判定是否已经尝试了DMA通道编程。例如，计算设备100可以判定TEE 314是否已经请求了生成包封的编程信息，如上面结合图4的框406所描述的那样。作为另一示例，计算设备100可以判定不可信软件是否已经请求了对编程信息进行解包封和/或编程，如上面结合图4的框416所描述的那样。如果未检测到通道编程尝试，则方法500循环回到框506以对于通道编程尝试继续进行监测。如果检测到通道编程尝试，则方法500前进到框508。

[0053] 在框508中，计算设备100读取系统清单并且验证所述清单的签名。为了验证签名，计算设备100可以使用任何密码技术来验证清单未被修改。例如，计算设备100可以使用计算设备100的制造商的公开密钥来验证所述签名。所述公开密钥可以在制造期间嵌入计算设备100中，例如通过将公开密钥烧制成多个熔丝(fuse)，所述熔丝可以在初始化期间由加密引擎140拉动并随后用于验证。另外或可替代地，在一些实施例中，加密引擎140可以用证书授权机构(CA)的公开密钥进行硬编码，并且可以请求CA发送制造商的已签名公开密钥，加密引擎140然后可以验证并检索制造商的公开密钥。如上所述，已签名清单的验证可以由加密引擎140执行和/或可以被卸载到处理器120。例如，可以响应于EBINDTIO或UNWRAP指令的调用而由处理器120的微码执行已签名清单的验证。

[0054] 在框510中，计算设备100判定系统清单是否被成功验证。若否，则方法500前进到框512，在所述框中，计算设备100放弃DMA通道编程请求并且记录错误。在放弃所述编程请求之后，相关联的DMA通道的状态可以保持不变。在放弃所述编程请求之后，方法500循环回到框506，在所述框中，计算设备100对于另外的通道编程请求进行监测。

[0055] 返回参考框510，如果系统清单通过验证，则方法500前进到框514，在所述框中，计算设备100判定对于当前TEE 314是否允许DMA通道编程。例如，对于所请求的DMA通道，计算设备100可以通过CID来搜索系统签名，并且判定是否允许当前的TEE 314对所述DMA通道进行编程。计算设备100可以使用任何适当的技术来确定当前TEE 314的标识。例如，处理器120可以通过判定处理器120是否以一个或多个特定执行模式执行来标识当前TEE314。处理器120可以判定它是在安全飞地执行还是以VMX根模式下(例如，在VMM内)执行。作为另一示例，如上所述，计算设备100可以例如通过检查BIND_STRUCT对象的WRAPPING_TEE字段基于包封的编程信息来标识所述编程TEE 314。在框516中，计算设备100检查是否允许DMA通道编程。若否，则方法500分支到框512，在所述框中，计算设备100放弃所述编程请求并且记录错误，如上所述。如果DMA通道编程被允许，则方法500分支到框518。

[0056] 在框518中，计算设备100继续进行DMA通道编程请求。例如，如上面结合图4的框

406所描述的，处理器120可以继续生成包封的编程信息。作为另一示例，如上面结合图4的框416所描述的，处理器120和/或加密引擎140可以继续对包封的编程信息进行解包封和/或对DMA通道进行编程。在DMA编程请求被允许之后，DMA通道的状态可以改变，例如通过使DMA通道进入安全操作或通过使DMA通道脱离安全操作。在允许DMA通道编程之后，方法500循环回到框506，在所述框中，计算设备100对附加的通道编程请求进行监测。

[0057] 现在参考图6，在使用中，计算设备100可以执行用于DMA通道所有权检查的方法600。方法600可以由计算设备100的硬件、固件、处理器微码、软件或其他执行资源来执行。例如，在一些实施例中，可以由处理器120的微码响应于如上面结合图4的框422所描述的对UNWRAP指令的调用而执行方法600。另外或可替代地，在一些实施例中，可以由加密引擎140响应于如上结合图4的框422所描述的编程尝试而执行方法600。

[0058] 方法600开始于框602，在所述框中，计算设备100判定是否已经尝试了DMA通道编程。例如，计算设备100可以判定不可信软件是否已经请求了对包封的编程信息进行解包封和/或编程，如上面结合图4的框416所描述的那样。如果未检测到通道编程尝试，则方法600循环回到框602以对于通道编程尝试继续进行监测。如果检测到通道编程尝试，则方法600前进到框604。

[0059] 在框604中，计算设备100判定通道编程请求是否是将DMA通道编程为安全的请求。加密引擎140可以支持若干个编程命令，包括将通道编程为安全、将通道编程为缺乏安全、以及其他命令。计算设备100可以检查所述包封的编程信息以判定TEE 314是否已经请求了将DMA通道编程为安全。例如，BIND_STRUCT对象可以包括一个或多个包含所请求的编程命令的字段。例如，BIND_STRUCT对象字段的一个或多个字段可以包括将用于将通道编程为安全的命令或用于将通道编程为缺乏安全的命令。如果通道编程请求不是将通道编程为安全的请求，则方法600向前分支到下面描述的框610。如果通道编程请求是将通道编程为安全的请求，则方法600前进到框606。

[0060] 在框606中，计算设备100确定发起通道编程请求的可信执行环境(TEE)314。计算设备100可以使用任何适当的技术来标识发源TEE 314。例如，计算设备100可以例如通过检查BIND_STRUCT对象的WRAPPING_TEE字段基于包封的编程信息来标识发源TEE 314。在框608中，计算设备100记录与有待被编程为安全的DMA通道相关联的起源TEE 314的标识。计算设备100可以使用任何技术来记录编程TEE 314。例如，加密引擎140的CID表142可以包括用于以每个条目标识相关联的TEE 314的字段。作为另一示例，可以记录用于保护DMA通道的加密密钥(其是编程TEE 314的秘密)，以标识相关联的TEE 314。

[0061] 在框610中，计算设备100判定通道编程请求是否是将DMA通道编程为缺乏安全的请求。例如，计算设备100可以判定BIND_STRUCT对象是否包括解除保护通道的请求。如果所述编程请求不是将DMA通道编程为缺乏安全的请求，则方法600向前分支到框616，在所述框中，计算设备100继续进行DMA通道编程。计算设备100可以将所请求的DMA通道编程为安全或者执行任何其他所请求的命令(例如，用不同的密钥对通道进行重新编程或查询通道密钥)。例如，如上面结合图4的框416所描述的，处理器120和/或加密引擎140可以继续对包封的编程信息进行解包封和/或对DMA通道进行编程。在对DMA通道进行编程之后，方法600循环回到框602以对附加的通道编程请求继续进行监测。

[0062] 返回参考框610，如果编程请求是将DMA通道编程为缺乏安全的请求，则方法600向

前分支到框612，在所述框中，对于DMA通道，计算设备100将所述编程TEE 314与所记录的TEE 314进行比较。如上面结合框608所描述的，当最初将DMA通道编程为安全时，计算设备100存储编程TEE 314的标识。因此，通过将对DMA通道编程为缺乏安全的请求的编程TEE 314与所记录的TEE 314进行比较，计算设备100判定将DMA通道编程为安全的同一TEE314是否试图将DMA通道编程为缺乏安全。计算设备100可以使用任何适当的技术来将所述编程TEE 314与所记录的TEE 314进行比较。例如，计算设备100可以将编程TEE 314与记录在用于所述DMA通道的加密引擎140的CID表142中的TEE 314的标识进行比较。作为另一示例，计算设备100可以将解除保护DMA通道的请求中所包含的通道编程密钥与与当前存储在CID表142中并用于保护DMA通道的加密密钥进行比较。

[0063] 在框614中，计算设备100判定编程TEE 314和所记录的TEE 314是否匹配。若是，则方法600分支到框616，在所述框中，计算设备100进行DMA通道编程。例如，如上面结合图4的框416所描述的，处理器120和/或加密引擎140可以继续对包封的编程信息进行解包封和/或将DMA通道编程为缺乏安全。在对所述通道进行编程之后，方法600循环回到框602以对于附加的通道编程请求继续进行监测。

[0064] 返回参考框614，如果编程TEE 314和所记录的TEE 314不匹配，则方法600分支到框618，在所述框中，计算设备100放弃所述编程请求并且记录错误。在放弃所述编程请求之后，相关联的DMA通道的状态可以保持不变。因此，DMA通道可以保持免受编程TEE 314的影响。在放弃所述编程请求之后，方法600循环回到框602，在所述框中，计算设备100对于另外的通道编程请求进行监测。

[0065] 应当理解的是，在一些实施例中，方法400、500和/或600可以被实施为存储在计算机可读介质上的各种指令，所述指令可以由计算设备100的处理器120、加密引擎140和/或其他组件来执行，以使得计算设备100执行相应的方法400、500和/或600。所述计算机可读介质可以被实施为能够由计算设备100读取的任何类型的介质，包括但不限于存储器132、数据存储设备134、处理器120的微码、加密引擎140的存储器、加密引擎140的固件、和/或其他介质。

[0066] 示例

[0067] 下文提供本文中所公开的技术的说明性示例。这些技术的实施例可包括下文描述的示例中的任何一者或多者以及其任何组合。

[0068] 示例1包括一种用于安全的加密引擎编程的计算设备，所述计算设备包括：配置模块，用于由所述计算设备的可信执行环境请求对所述计算设备的加密引擎进行编程以保护直接存储器存取(DMA)通道；以及清单验证模块，用于：(i)响应于对所述加密引擎进行编程的请求，验证所述计算设备的已签名清单，其中，所述已签名清单指示被准许对DMA通道进行编程的一个或多个可信执行环境，(ii)响应于对所述已签名清单的验证，判定是否准许所述可信执行环境基于所述已签名清单对所述DMA通道进行编程，以及(iii)响应于确定了所述可信执行环境被准许对所述DMA通道进行编程，对所述DMA通道进行编程。

[0069] 示例2包括如示例1所述的主题，并且其中，所述可信执行环境包括用所述计算设备的处理器的安全飞地支持建立的安全飞地、或虚拟机监视器。

[0070] 示例3包括如示例1和2中任一项所述的主题，并且其中，所述清单验证模块进一步用于响应于确定了所述可信执行环境不被准许对所述DMA通道进行编程而放弃对所述加密

引擎进行编程的所述请求。

[0071] 示例4包括如示例1至3中任一项所述的主题，并且其中，所述配置模块进一步用于将所述已签名清单安全地供应到所述计算设备。

[0072] 示例5包括如示例1至4中任一项所述的主题，并且其中：所述配置模块进一步用于将所述已签名清单的地址编程到所述加密引擎中；其中，验证所述计算设备的所述已签名清单包括通过所述已签名清单的所述地址来读取所述已签名清单。

[0073] 示例6包括如示例1至5中任一项所述的主题，并且其中，验证所述计算设备的所述已签名清单包括使用制造商公开密钥来验证所述已签名清单。

[0074] 示例7包括如示例1至6中任一项所述的主题，并且进一步包括处理器，其中：请求对所述计算设备的所述加密引擎进行编程以保护所述DMA通道包括调用处理器指令来生成包封的编程信息；所述处理器包括所述清单验证模块；并且验证所述计算设备的所述已签名清单包括响应于对所述处理器指令的调用而验证所述已签名清单。

[0075] 示例8包括如示例1至7中任一项所述的主题，并且其中：请求对所述计算设备的所述加密引擎进行编程以保护所述DMA通道包括由所述可信执行环境调用第一处理器指令来生成包封的编程信息，其中，所述包封的编程信息指示所述可信执行环境；并且基于所述已签名清单来判定是否准许所述可信执行环境对所述DMA通道进行编程进一步包括基于所述包封的编程信息来判定所述可信执行环境。

[0076] 示例9包括如示例1至8中任一项所述的主题，并且进一步包括处理器，其中：生成对所述计算设备的所述加密引擎进行编程以保护所述DMA通道的请求包括由所述计算设备的不可信软件利用所述包封的编程信息来调用第二处理器指令；并且所述处理器包括所述清单验证模块。

[0077] 示例10包括如示例1至9中任一项所述的主题，并且其中，所述加密引擎包括所述清单验证模块。

[0078] 示例11包括一种用于安全的加密引擎编程的计算设备，所述计算设备包括：配置模块，用于由可信执行环境针对直接存储器存取(DMA)通道生成对所述计算设备的加密引擎进行编程的请求；以及所有权验证模块，用于：(i)判定对所述加密引擎进行编程的所述请求是否包括将所述DMA通道编程为缺乏安全的请求；(ii)响应于确定了对所述加密引擎进行编程的所述请求包括将所述DMA通道编程为缺乏安全的请求，判定所述可信执行环境是否与同所述DMA通道相关联的所记录的可信执行环境相匹配，以及(iii)响应于确定了所述可信执行环境与同所述DMA通道相关联的所述记录的可信执行环境相匹配，允许对所述加密引擎进行编程的所述请求。

[0079] 示例12包括如示例11所述的主题，并且其中，所述可信执行环境包括用所述计算设备的处理器的安全飞地支持建立的安全飞地、或虚拟机监视器。

[0080] 示例13包括如示例11和12中任一项所述的主题，并且其中，所述所有权验证模块进一步用于响应于确定了所述可信执行环境与同所述DMA通道相关联的所述记录的可信执行环境不匹配而放弃对所述加密引擎进行编程的所述请求。

[0081] 示例14包括如示例11和至13中任一项所述的主题，并且其中，所述所有权验证模块进一步用于：判定对所述加密引擎进行编程的所述请求是否包括将所述DMA通道编程为安全的请求；以及响应于确定了对所述加密引擎进行编程的所述请求包括将所述DMA通道

编程为安全的请求，记录与所述DMA通道相关联的所述可信执行环境。

[0082] 示例15包括如示例11至14中任一项所述的主题，并且其中，对所述加密引擎进行编程的所述请求包括指示所述可信执行环境的包封编程信息。

[0083] 示例16包括如示例11至15中任一项所述的主题，并且其中，所述加密引擎包括所述所有权验证模块。

[0084] 示例17包括如示例11至16中任一项所述的主题，并且进一步包括处理器，其中，所述处理器包括所述所有权验证模块。

[0085] 示例18包括如示例11至17中任一项所述的主题，并且其中：生成对所述加密引擎进行编程的所述请求包括：(i)由所述可信执行环境调用第一处理器指令来生成包封的编程信息，其中，所述包封的编程信息指示所述可信执行环境，以及(ii)由所述计算设备的不可信软件利用所述包封的编程信息来调用第二处理器指令；并且判定对所述加密引擎进行编程的所述请求是否包括将所述DMA通道编程为缺乏安全的请求包括由所述处理器响应于对所述第二处理器指令的调用而判定对所述加密引擎进行编程的所述请求是否包括将所述DMA通道编程为缺乏安全的请求。

[0086] 示例19包括如示例11至18中任一项所述的主题，并且其中，判定所述可信执行环境是否与同所述DMA通道相关联的所述记录的可信执行环境相匹配包括将同对所述加密引擎进行编程的所述请求相关联的通道编程密钥与同所述DMA通道相关联的加密密钥进行比较。

[0087] 示例20包括一种用于安全的加密引擎编程的方法，所述方法包括：由计算设备的可信执行环境生成对所述计算设备的加密引擎进行编程以保护直接存储器存取(DMA)通道的请求；响应于生成对所述加密引擎进行编程的所述请求，由所述计算设备验证所述计算设备的已签名清单，其中，所述已签名清单指示被准许对DMA通道进行编程的一个或多个可信执行环境；响应于验证所述已签名清单，由所述计算设备基于所述已签名清单来判定是否准许所述可信执行环境对所述DMA通道进行编程；以及响应于确定了准许所述可信执行环境对所述DMA通道进行编程，由所述计算设备对所述DMA通道进行编程。

[0088] 示例21包括如示例20所述的主题，并且其中，所述可信执行环境包括用所述计算设备的处理器的安全飞地支持建立的安全飞地、或虚拟机监视器。

[0089] 示例22包括如示例20和21中任一项所述的主题，并且进一步包括由所述计算设备响应于确定了所述可信执行环境不被准许对所述DMA通道进行编程而放弃对所述加密引擎进行编程的所述请求。

[0090] 示例23包括如示例20至22中任一项所述的主题，并且进一步包括将所述已签名清单安全地供应到所述计算设备。

[0091] 示例24包括如示例20至23中任一项所述的主题，并且进一步包括由所述计算设备将所述已签名清单的地址编程到所述加密引擎中；其中，验证所述计算设备的所述已签名清单包括通过所述已签名清单的所述地址来读取所述已签名清单。

[0092] 示例25包括如示例20至24中任一项所述的主题，并且其中，验证所述计算设备的所述已签名清单包括使用制造商公开密钥来验证所述已签名清单。

[0093] 示例26包括如示例20至25中任一项所述的主题，并且其中：生成对所述计算设备的所述加密引擎进行编程以保护所述DMA通道的所述请求包括调用处理器指令来生成包封

的编程信息;验证所述计算设备的所述已签名清单包括响应于调用所述处理器指令由所述计算设备的处理器来验证所述已签名清单;基于所述系统清单来判定是否准许所述可信执行环境对所述DMA通道进行编程包括由所述处理器基于所述系统清单来判定是否准许所述可信执行环境对所述DMA通道进行编程;并且响应于确定了准许所述可信执行环境对所述DMA通道进行编程而对所述DMA通道进行编程包括由所述处理器响应于确定了准许所述可信执行环境对所述DMA通道进行编程而生成所述包封的编程信息。

[0094] 示例27包括如示例20至26中任一项所述的主题,并且其中:生成对所述计算设备的所述加密引擎进行编程以保护DMA通道的所述请求包括由所述可信执行环境调用第一处理器指令来生成包封的编程信息,其中,所述包封的编程信息指示所述可信执行环境;并且基于所述已签名清单来判定是否准许所述可信执行环境对所述DMA通道进行编程进一步包括基于所述包封的编程信息来判定所述可信执行环境。

[0095] 示例28包括如示例20至27中任一项所述的主题,并且其中:生成对所述计算设备的所述加密引擎进行编程以保护所述DMA通道的所述请求包括由所述计算设备的不可信软件利用所述包封的编程信息来调用第二处理器指令;验证所述计算设备的所述已签名清单包括由所述计算设备的处理器响应于调用所述第二处理器指令而验证所述已签名清单;基于所述系统清单来判定是否准许所述可信执行环境对所述DMA通道进行编程包括由所述处理器判定是否准许所述可信执行环境对所述DMA通道进行编程;并且响应于确定了准许所述可信执行环境对所述DMA通道进行编程而对所述DMA通道进行编程包括由所述处理器响应于确定了准许所述可信执行环境对所述DMA通道进行编程而对所述DMA通道进行编程。

[0096] 示例29包括如示例20至28中任一项所述的主题,并且其中:验证所述计算设备的所述已签名清单包括由所述加密引擎验证所述已签名清单;基于所述系统清单来判定是否准许所述可信执行环境对所述DMA通道进行编程包括由所述加密引擎判定是否准许所述可信执行环境对所述DMA通道进行编程;并且响应于确定了准许所述可信执行环境对所述DMA通道进行编程而对所述DMA通道进行编程包括由所述加密引擎响应于确定了准许所述可信执行环境对所述DMA通道进行编程而对所述DMA通道进行编程。

[0097] 示例30包括一种用于安全的加密引擎编程的方法,所述方法包括:针对直接存储器存取(DMA)通道,由计算设备的可信执行环境生成对所述计算设备的加密引擎进行编程的请求;由所述计算设备判定对所述加密引擎进行编程的所述请求是否包括将所述DMA通道编程为缺乏安全的请求;响应于确定了对所述加密引擎进行编程的所述请求包括将所述DMA通道编程为缺乏安全的请求,由所述计算设备判定所述可信执行环境是否与同所述DMA通道相关联的所记录的可信执行环境相匹配;以及响应于确定了所述可信执行环境与同所述DMA通道相关联的所述记录的可信执行环境相匹配,由所述计算设备允许对所述加密引擎进行编程的所述请求。

[0098] 示例31包括如示例30所述的主题,并且其中,所述可信执行环境包括用所述计算设备的处理器的安全飞地支持建立的安全飞地、或虚拟机监视器。

[0099] 示例32包括如示例30和31中任一项所述的主题,并且进一步包括响应于确定了所述可信执行环境与同所述DMA通道相关联的所述记录的可信执行环境不匹配,由所述计算设备放弃对所述加密引擎进行编程的所述请求。

[0100] 示例33包括如示例30至32中任一项所述的主题,并且进一步包括:由所述计算设

备判定对所述加密引擎进行编程的所述请求是否包括将所述DMA通道编程为安全的请求；以及由所述计算设备响应于确定了对所述加密引擎进行编程的所述请求包括将所述DMA通道编程为安全的请求，记录与所述DMA通道相关联的所述可信执行环境。

[0101] 示例34包括如示例30至33中任一项所述的主题，并且其中，生成对所述加密引擎进行编程的所述请求包括生成指示所述可信执行环境的包封编程信息。

[0102] 示例35包括如示例30至34中任一项所述的主题，并且其中：判定对所述加密引擎进行编程的所述请求是否包括将所述DMA通道编程为缺乏安全的请求包括由所述加密引擎判定对所述加密引擎进行编程的所述请求是否包括将所述DMA通道编程为缺乏安全的请求；判定所述可信执行环境是否与所述记录的可信执行环境相匹配包括由所述加密引擎判定所述可信执行环境是否与所述记录的可信执行环境相匹配；并且允许对所述加密引擎进行编程的所述请求包括由所述加密引擎允许对所述加密引擎进行编程的所述请求。

[0103] 示例36包括如示例30至35中任一项所述的主题，并且其中：判定对所述加密引擎进行编程的所述请求是否包括将所述DMA通道编程为缺乏安全的请求包括由所述计算设备的处理器判定对所述加密引擎进行编程的所述请求是否包括将所述DMA通道编程为缺乏安全的请求；判定所述可信执行环境是否与所述记录的可信执行环境相匹配包括由所述处理器判定所述可信执行环境是否与所述记录的可信执行环境相匹配；并且允许对所述加密引擎进行编程的所述请求包括由所述处理器允许对所述加密引擎进行编程的所述请求。

[0104] 示例37包括如示例30至36中任一项所述的主题，并且其中：生成对所述加密引擎进行编程的所述请求包括：(i)由所述可信执行环境调用第一处理器指令来生成包封的编程信息，其中，所述包封的编程信息指示所述可信执行环境，以及(ii)由所述计算设备的不可信软件利用所述包封的编程信息来调用第二处理器指令；并且判定对所述加密引擎进行编程的所述请求是否包括将所述DMA通道编程为缺乏安全的请求包括由所述处理器响应于对所述第二处理器指令的调用而判定对所述加密引擎进行编程的所述请求是否包括将所述DMA通道编程为缺乏安全的请求。

[0105] 示例38包括如示例30至37中任一项所述的主题，并且其中，判定所述可信执行环境是否与同所述DMA通道相关联的所述记录的可信执行环境相匹配包括将同对所述加密引擎进行编程的所述请求相关联的通道编程密钥与同所述DMA通道相关联的加密密钥进行比较。

[0106] 示例39包括一种计算设备，所述计算设备包括：处理器；以及存储器，所述存储器具有存储于其中的多条指令，所述多条指令当被所述处理器执行时致使所述计算设备执行如示例20至38中任一项所述的方法。

[0107] 示例40包括一种或多种机器可读存储介质，所述一种或多种机器可读存储介质包括存储于其上的多条指令，所述多条指令响应于被执行而使计算设备执行如示例20至38中任一项所述的方法。

[0108] 示例41包括一种计算设备，所述计算设备包括用于执行示例20至38中任一项所述的方法的装置。

[0109] 示例42包括一种用于安全的加密引擎编程的计算设备，所述计算设备包括：用于由所述计算设备的可信执行环境生成对所述计算设备的加密引擎进行编程以保护直接存储器存取(DMA)通道的请求的装置；用于响应于生成对所述加密引擎进行编程的所述请求

而验证所述计算设备的已签名清单的装置,其中,所述已签名清单指示被准许对DMA通道进行编程的一个或多个可信执行环境;用于响应于验证所述已签名清单而基于所述已签名清单来判定是否准许所述可信执行环境对所述DMA通道进行编程的装置;以及用于响应于确定了准许所述可信执行环境对所述DMA通道进行编程而对所述DMA通道进行编程的装置。

[0110] 示例43包括如示例42所述的主题,并且其中,所述可信执行环境包括用所述计算设备的处理器的安全飞地支持建立的安全飞地、或虚拟机监视器。

[0111] 示例44包括如示例42和43中任一项所述的主题,并且进一步包括用于响应于确定了所述可信执行环境不被准许对所述DMA通道进行编程而放弃对所述加密引擎进行编程的所述请求的装置。

[0112] 示例45包括如示例42至44中任一项所述的主题,并且进一步包括用于将所述已签名清单安全地供应到所述计算设备的装置。

[0113] 示例46包括如示例42至45中任一项所述的主题,并且进一步包括:用于将所述已签名清单的地址编程到所述加密引擎中的装置;其中,所述用于验证所述计算设备的所述已签名清单的装置包括用于通过所述已签名清单的所述地址来读取所述已签名清单的装置。

[0114] 示例47包括如示例42至46中任一项所述的主题,并且其中,所述用于验证所述计算设备的所述已签名清单的装置包括用于使用制造商公开密钥来验证所述已签名清单的装置。

[0115] 示例48包括如示例42至47中任一项所述的主题,并且其中:所述用于生成对所述计算设备的所述加密引擎进行编程以保护所述DMA通道的所述请求的装置包括用于调用处理器指令来生成包封编程信息的装置;所述用于验证所述计算设备的所述已签名清单的装置包括响应于调用所述处理器指令由所述计算设备的处理器来验证所述已签名清单的装置;所述用于基于所述系统清单来判定是否准许所述可信执行环境对所述DMA通道进行编程的装置包括由所述处理器基于所述系统清单来判定是否准许所述可信执行环境对所述DMA通道进行编程的装置;并且所述用于响应于确定了准许所述可信执行环境对所述DMA通道进行编程而对所述DMA通道进行编程的装置包括用于由所述处理器响应于确定了准许所述可信执行环境对所述DMA通道进行编程而生成所述包封编程信息的装置。

[0116] 示例49包括如示例42至48中任一项所述的主题,并且其中:所述用于生成对所述计算设备的所述加密引擎进行编程以保护DMA通道的所述请求的装置包括由所述可信执行环境调用第一处理器指令来生成包封编程信息的装置,其中,所述包封的编程信息指示所述可信执行环境;并且所述用于基于所述已签名清单来判定是否准许所述可信执行环境对所述DMA通道进行编程的装置进一步包括用于基于所述包封的编程信息来判定所述可信执行环境的装置。

[0117] 示例50包括如示例42至49中任一项所述的主题,并且其中:所述用于生成对所述计算设备的所述加密引擎进行编程以保护所述DMA通道的所述请求的装置包括用于由所述计算设备的不可信软件利用所述包封的编程信息来调用第二处理器指令的装置;所述用于验证所述计算设备的所述已签名清单的装置包括由所述计算设备的处理器响应于调用所述第二处理器指令而验证所述已签名清单的装置;所述用于基于所述系统清单判定是否准许所述可信执行环境对所述DMA通道进行编程的装置包括由所述处理器判定是否准许所述

可信执行环境对所述DMA通道进行编程的装置;并且所述用于响应于确定了准许所述可信执行环境对所述DMA通道进行编程而对所述DMA通道进行编程的装置包括用于由所述处理器响应于确定了准许所述可信执行环境对所述DMA通道进行编程而对所述DMA通道进行编程的装置。

[0118] 示例51包括如示例42至50中任一项所述的主题,并且其中:所述用于验证所述计算设备的所述已签名清单的装置包括用于由所述加密引擎验证所述已签名清单的装置;所述用于基于所述系统清单来判定是否准许所述可信执行环境对所述DMA通道进行编程的装置包括由所述加密引擎判定是否准许所述可信执行环境对所述DMA通道进行编程的装置;并且所述用于响应于确定了准许所述可信执行环境对所述DMA通道进行编程而对所述DMA通道进行编程的装置包括用于由所述加密引擎响应于确定了准许所述可信执行环境对所述DMA通道进行编程而对所述DMA通道进行编程的装置。

[0119] 示例52包括一种用于安全的加密引擎编程的计算设备,所述计算设备包括:用于针对直接存储器存取(DMA)通道由所述计算设备的可信执行环境生成对所述计算设备的加密引擎进行编程的请求的装置;用于判定对所述加密引擎进行编程的所述请求是否包括将所述DMA通道编程为缺乏安全的请求的装置;用于响应于确定了对所述加密引擎进行编程的所述请求包括将所述DMA通道编程为缺乏安全的请求而判定所述可信执行环境是否与同所述DMA通道相关联的所记录的可信执行环境相匹配的装置;以及用于响应于确定了所述可信执行环境与同所述DMA通道相关联的所述记录的可信执行环境相匹配而允许对所述加密引擎进行编程的所述请求的装置。

[0120] 示例53包括如示例52所述的主题,并且其中,所述可信执行环境包括用所述计算设备的处理器的安全飞地支持建立的安全飞地、或虚拟机监视器。

[0121] 示例54包括如示例52和53中任一项所述的主题,并且进一步包括用于响应于确定了所述可信执行环境与同所述DMA通道相关联的所述记录的可信执行环境不匹配而放弃对所述加密引擎进行编程的所述请求的装置。

[0122] 示例55包括如示例52至54中任一项所述的主题,并且进一步包括:用于判定对所述加密引擎进行编程的所述请求是否包括将所述DMA通道编程为安全的请求的装置;以及用于响应于确定了对所述加密引擎进行编程的所述请求包括将所述DMA通道编程为安全的请求而记录与所述DMA通道相关联的所述可信执行环境的装置。

[0123] 示例56包括如示例52至55中任一项所述的主题,并且其中,所述用于生成对所述加密引擎进行编程的所述请求的装置包括用于生成指示所述可信执行环境的包封编程信息的装置。

[0124] 示例57包括如示例52至56中任一项所述的主题,并且其中:所述用于判定对所述加密引擎进行编程的所述请求是否包括将所述DMA通道编程为缺乏安全的请求的装置包括用于由所述加密引擎判定对所述加密引擎进行编程的所述请求是否包括将所述DMA通道编程为缺乏安全的请求的装置;所述用于判定所述可信执行环境是否与所述记录的可信执行环境相匹配的装置包括用于由所述加密引擎判定所述可信执行环境是否与所述记录的可信执行环境相匹配的装置;并且所述用于允许对所述加密引擎进行编程的所述请求的装置包括用于由所述加密引擎允许对所述加密引擎进行编程的所述请求的装置。

[0125] 示例58包括如示例52至57中任一项所述的主题,并且其中:所述用于判定对所述

加密引擎进行编程的所述请求是否包括将所述DMA通道编程为缺乏安全的请求的装置包括用于由所述计算设备的处理器判定对所述加密引擎进行编程的所述请求是否包括将所述DMA通道编程为缺乏安全的请求的装置;所述用于判定所述可信执行环境是否与所述记录的可信执行环境相匹配的装置包括用于由所述处理器判定所述可信执行环境是否与所述记录的可信执行环境相匹配的装置;并且所述用于允许对所述加密引擎进行编程的所述请求的装置包括用于由所述处理器允许对所述加密引擎进行编程的所述请求的装置。

[0126] 示例59包括如示例52至58中任一项所述的主题,并且其中:所述用于生成对所述加密引擎进行编程的所述请求的装置包括:(i)用于由所述可信执行环境调用第一处理器指令来生成包封编程信息的装置,其中,所述包封的编程信息指示所述可信执行环境,以及(ii)用于由所述计算设备的不可信软件利用所述包封的编程信息来调用第二处理器指令的装置;并且所述用于判定对所述加密引擎进行编程的所述请求是否包括将所述DMA通道编程为缺乏安全的请求的装置包括用于由所述处理器响应于对所述第二处理器指令的调用而判定对所述加密引擎进行编程的所述请求是否包括将所述DMA通道编程为缺乏安全的请求的装置。

[0127] 示例60包括如示例52至59中任一项所述的主题,并且其中,所述用于判定所述可信执行环境是否与同所述DMA信道相关联的所述记录的可信执行环境相匹配的装置包括用于将同对所述加密引擎进行编程的所述请求相关联的信道编程密钥与同所述DMA信道相关联的加密密钥进行比较的装置。

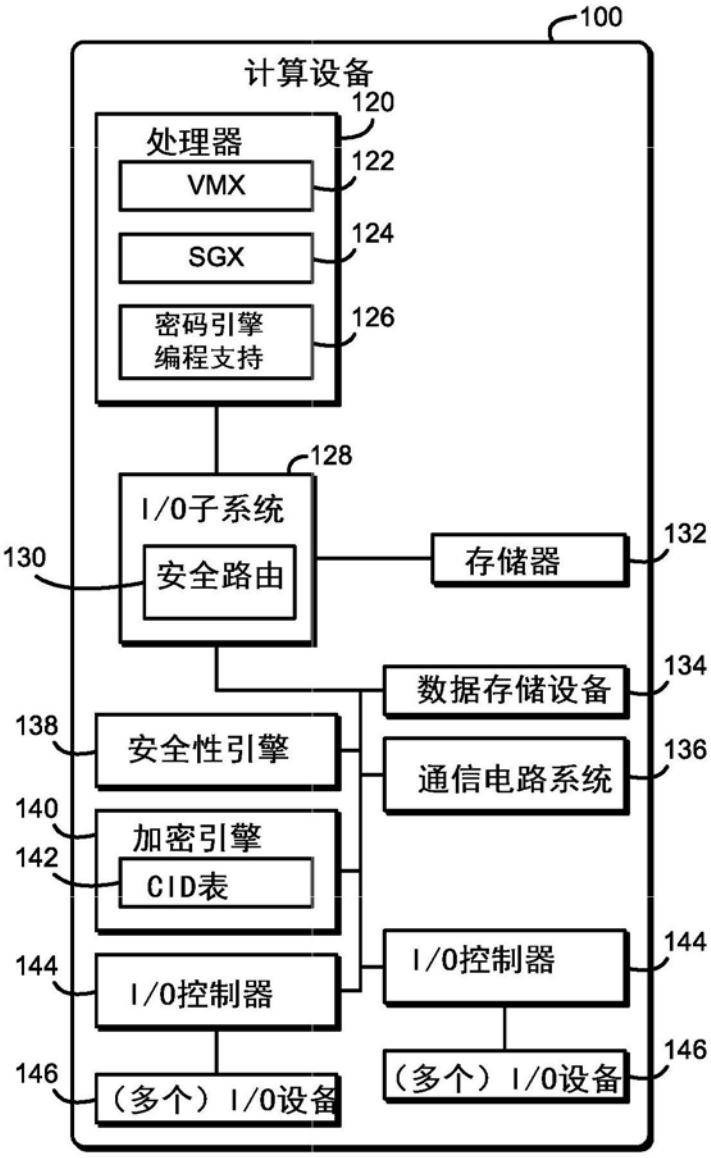

100
计算设备
120
处理器
122
VMX
124
SGX
126
密码引擎
编程支持
128
I/O子系统
130
安全路由
132
存储器
134
数据存储设备
138
安全性引擎
136
通信电路系统
140
加密引擎
142
CID表
144
I/O控制器
144
I/O控制器
146
（多个）I/O设备
146
（多个）I/O设备

图1

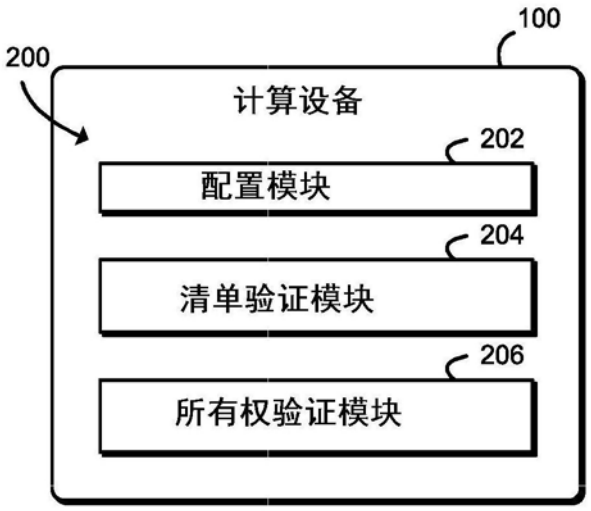



图2

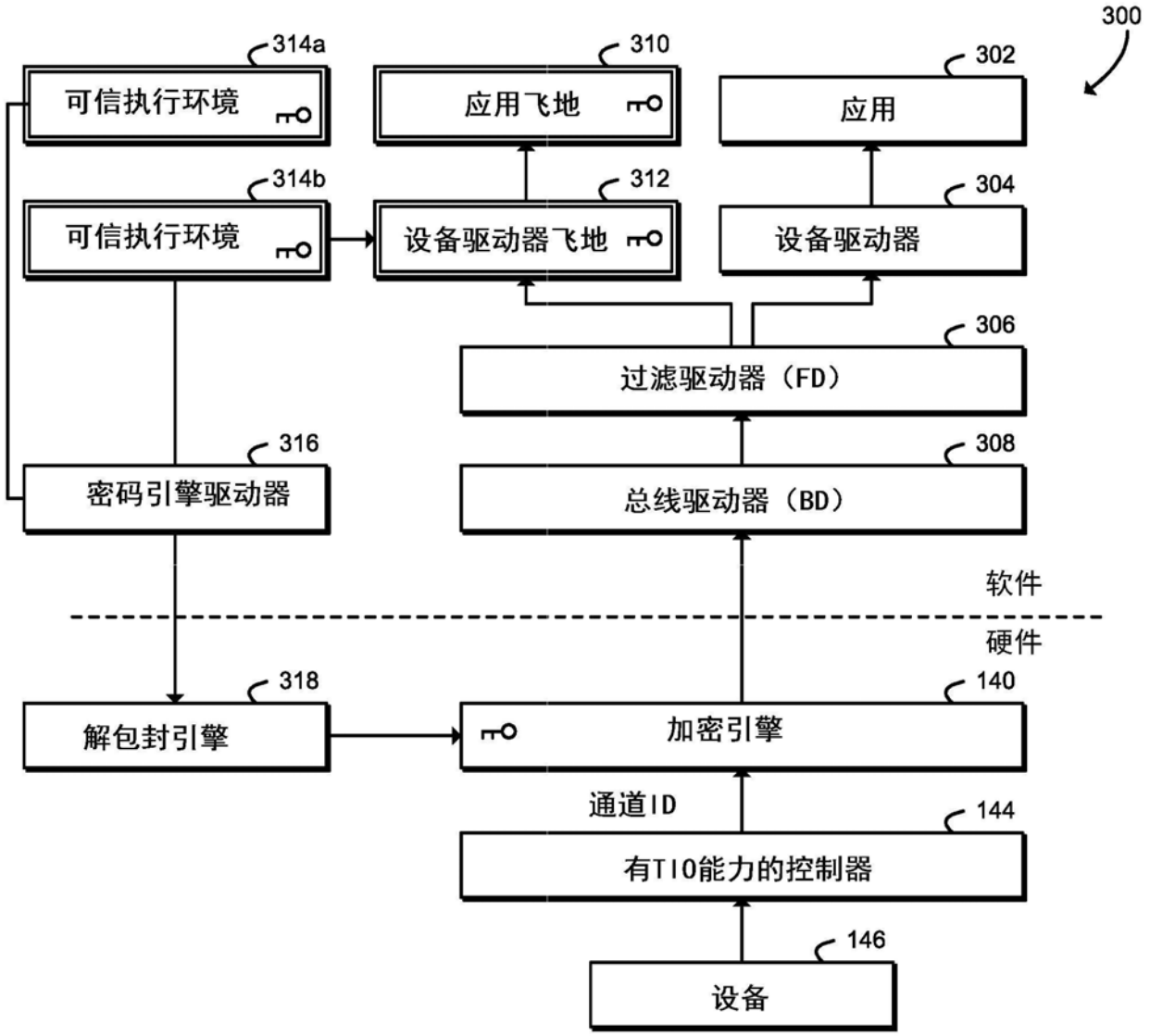

300
314a
可信执行环境
310
应用飞地
302
应用
314b
可信执行环境
312
设备驱动器飞地
304
设备驱动器
306
过滤驱动器（FD）
316
密码引擎驱动器
308
总线驱动器（BD）
软件
硬件
318
解包封引擎
140
加密引擎
通道ID
144
有TIO能力的控制器
146
设备

图3

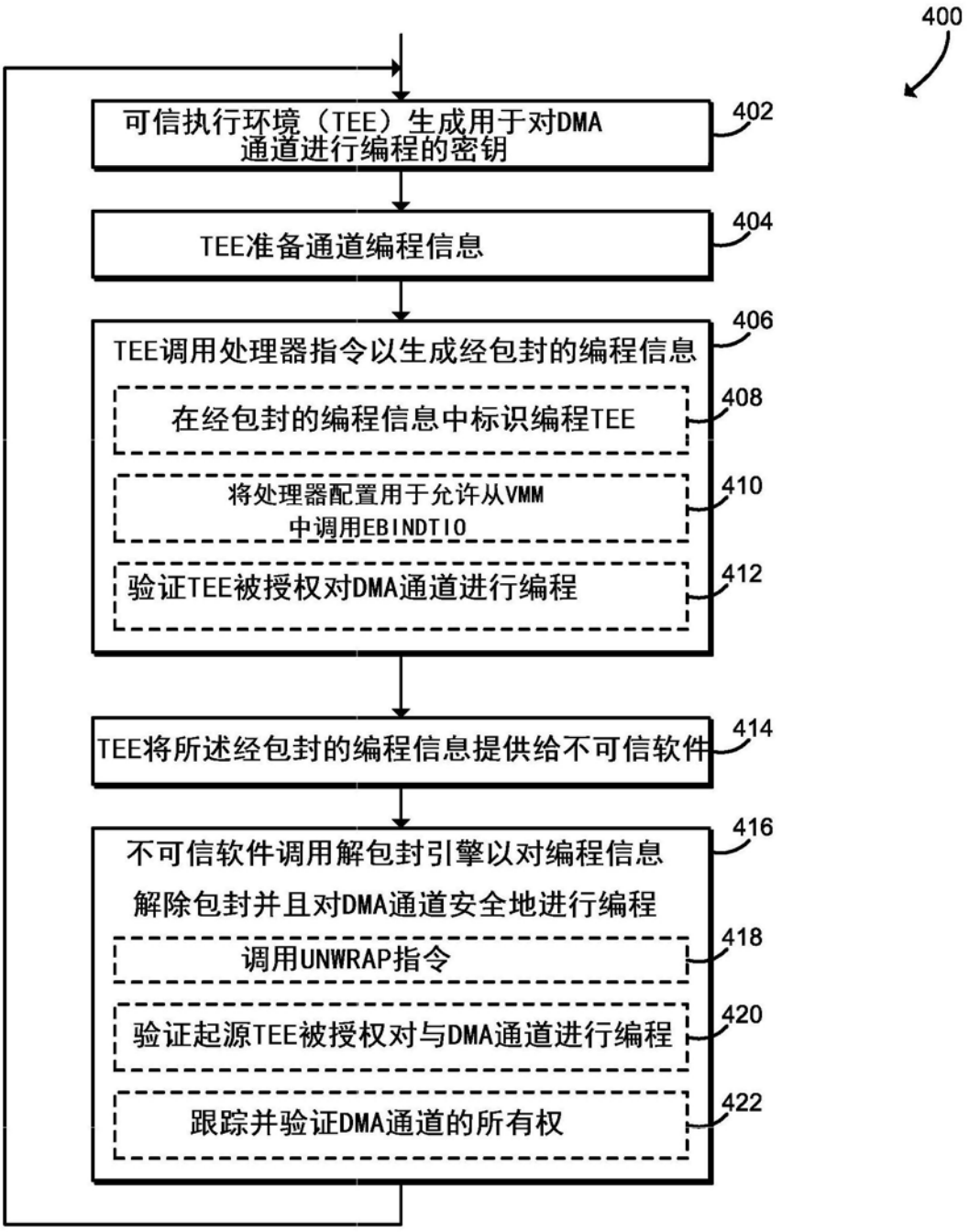

400
可信执行环境（TEE）生成用于对DMA通道进行编程的密钥
402
TEE准备通道编程信息
404
TEE调用处理器指令以生成经包封的编程信息
406
在经包封的编程信息中标识编程TEE
408
将处理器配置用于允许从VMM中调用EBINDTIO
410
验证TEE被授权对DMA通道进行编程
412
TEE将所述经包封的编程信息提供给不可信软件
414
不可信软件调用解包封引擎以对编程信息解除包封并且对DMA通道安全地进行编程
416
调用UNWRAP指令
418
验证起源TEE被授权对与DMA通道进行编程
420
跟踪并验证DMA通道的所有权
422

图4

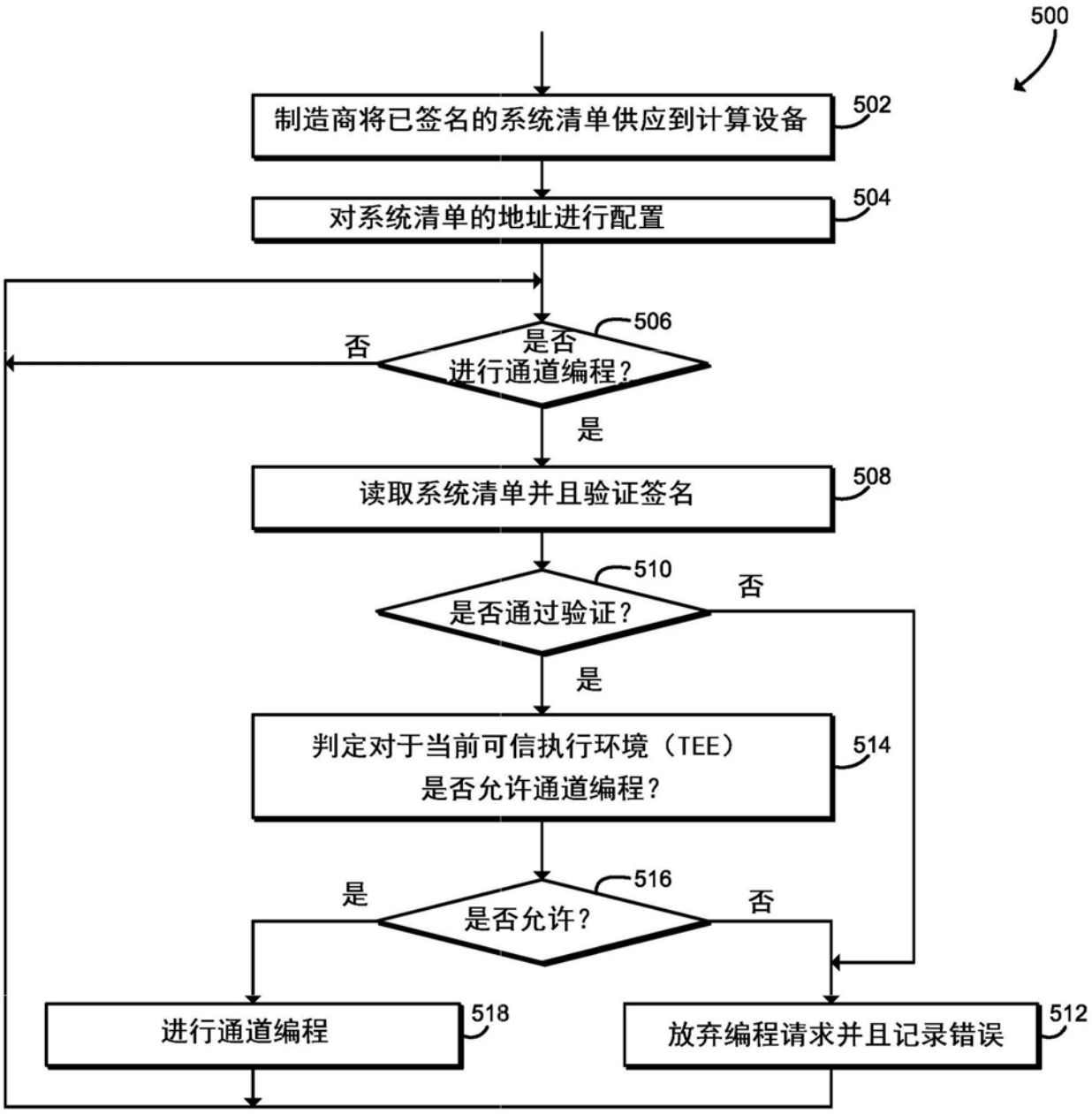

500
制造商将已签名的系统清单供应到计算设备
502
对系统清单的地址进行配置
504
是否
进行通道编程?
506
否
是
读取系统清单并且验证签名
508
是否通过验证?
510
否
是
判定对于当前可信执行环境（TEE）
是否允许通道编程?
514
是否允许?
516
是
否
进行通道编程
518
放弃编程请求并且记录错误
512

图5

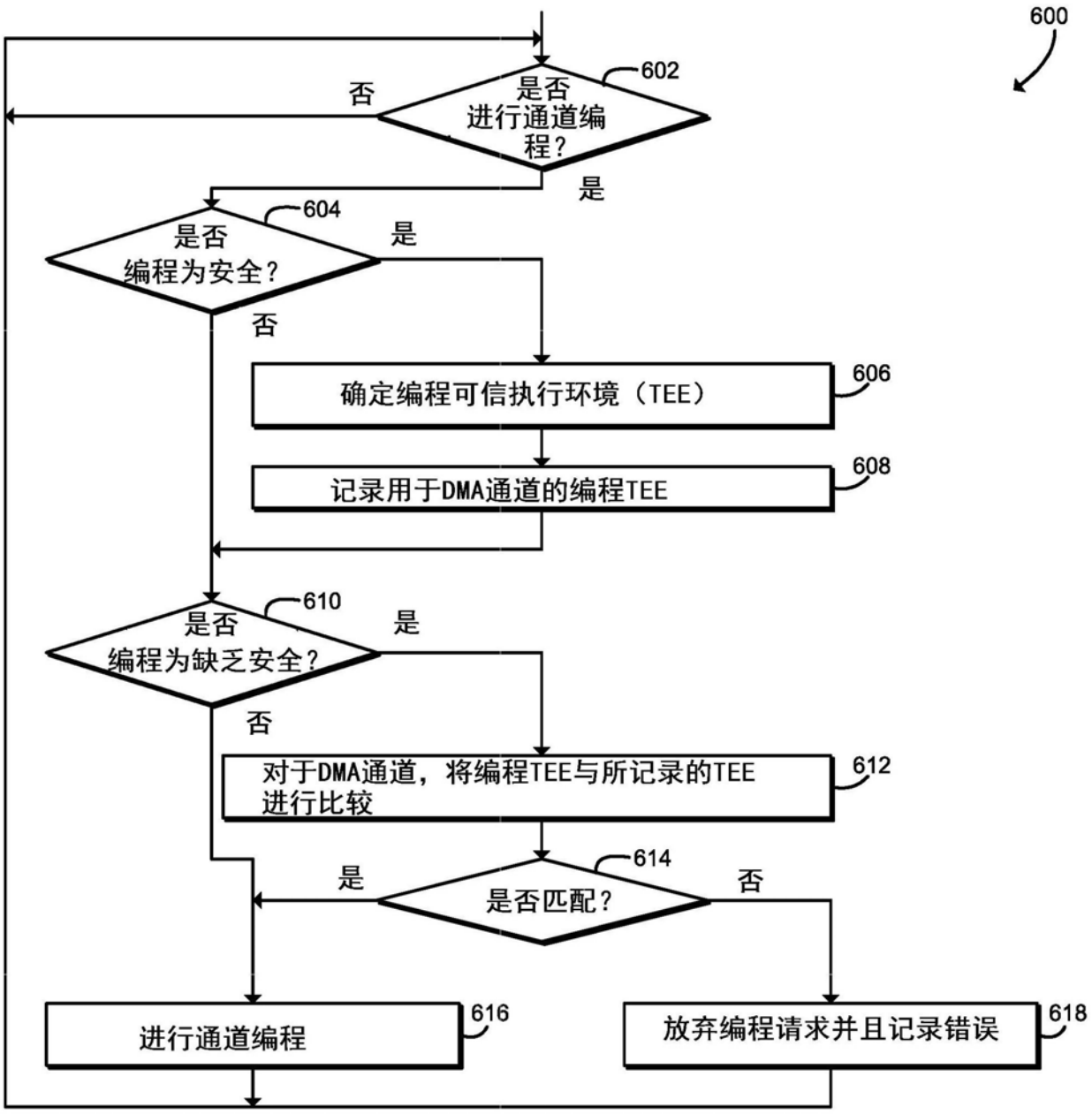

600
602
是否进行通道编程？
否
是
604
是否编程为安全？
是
否
606
确定编程可信执行环境（TEE）
608
记录用于DMA通道的编程TEE
610
是否编程为缺乏安全？
是
否
612
对于DMA通道，将编程TEE与所记录的TEE进行比较
614
是否匹配？
是
否
616
进行通道编程
618
放弃编程请求并且记录错误

图6