(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 107851167 A
(43)申请公布日 2018.03.27

(21)申请号 201680044488.5
(22)申请日 2016.07.15
(30)优先权数据
3995/CHE/2015 2015.07.31 IN
14/865,570 2015.09.25 US
(85)PCT国际申请进入国家阶段日
2018.01.29
(86)PCT国际申请的申请数据
PCT/US2016/042381 2016.07.15
(87)PCT国际申请的公布数据
WO2017/023510 EN 2017.02.09
(71)申请人 微软技术许可有限责任公司
地址 美国华盛顿州
(72)发明人 M·科斯塔 O·T·霍德森
S·K·拉亚马尼 M·佩纳多
M·E·拉希诺维奇 K·瓦斯瓦尼
(74)专利代理机构 北京市金杜律师事务所 11256
代理人 王茂华 黄捷
(51)Int.Cl.
*G06F 21/62*(2006.01)
*G06F 21/60*(2006.01)
*G06F 9/50*(2006.01)

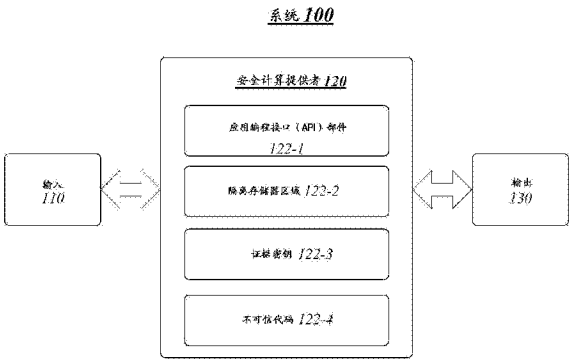
权利要求书2页 说明书19页 附图13页

(54)发明名称

在计算环境中保护计算数据的技术

(57)摘要

用于在计算环境中保护计算数据免于不可信代码的影响的技术。这些技术涉及计算环境内的隔离环境和用于执行密钥交换协议的应用编程接口(API)部件,该协议确保从隔离环境传达出的数据的数据完整性和数据机密性。隔离环境包括用于存储代码包的隔离存储器区域。密钥交换协议还涉及用于被存储在隔离环境中的代码包的验证过程,以确定一个或多个所交换的加密密钥是否已经被损害。如果签名成功认证一个或多个密钥,则到隔离环境的安全通信信道被建立,并且对代码包的功能性的访问被启用。其他实施例被描述和要求保护。




CN 107851167 A

1.一种装置,包括:

逻辑电路;以及

逻辑,在所述逻辑电路上操作以:在计算环境中配置隔离存储器区域,用于与代码进行安全通信,所述代码在与所述隔离存储器区域不同的存储器区域中可执行;使用与所述计算环境相对应的证据密钥来生成经签名数据,所述经签名数据包括被保护的加密密钥和用于验证所述被保护的加密密钥的签名;以及将所述经签名数据传达到远程可信部件,以访问被存储在所述隔离存储器区域中的秘密代码。

2.根据权利要求1所述的装置,其中所述逻辑进一步操作以用私有证据密钥来生成所述签名,所述私有证据密钥特定地对应于控制所述计算环境的安全计算提供者。

3.根据权利要求1所述的装置,其中所述逻辑进一步操作以生成用于在所述隔离存储器区域中运行的可信代码的加密密钥。

4.根据权利要求1所述的装置,其中所述逻辑进一步操作以将密钥-值对存储在缓冲器中,所述密钥-值对被传达到在所述隔离存储器区域或所述远程可信部件中运行的可信代码。

5.根据权利要求1所述的装置,其中所述逻辑进一步被配置为:在分布式文件系统上生成代码包的密码学摘要,以及使用所述密码学摘要来验证所述被保护的加密密钥的所述签名。

6.根据权利要求1所述的装置,其中所述逻辑进一步操作以处理指向在所述隔离存储器区域内部运行的可信代码或在所述隔离存储器区域外部运行的不可信代码的通信原语,其中所述逻辑进一步操作以处理通信原语,所述通信原语操作以调用在所述隔离存储器区域外部运行的所述不可信代码上的函数,或在所述隔离存储器区域内部运行的所述可信代码上的函数。

7.根据权利要求1所述的装置,所述逻辑进一步操作以使用公开证据密钥来验证所述签名,以及解密所述被保护的加密密钥以提取所述加密密钥。

8.一种包括至少一个计算机可读存储介质的产品,所述计算机可读存储介质包括指令,所述指令当被执行时,使得系统执行以下操作:

生成计算数据,所述计算数据与在计算环境的隔离存储器区域内执行的一组计算相对应;

使用加密密钥来保护所述计算数据以生成被保护的计算数据;以及

调用原语以将所述被保护的计算数据传达到在所述隔离存储器区域的外部运行的代码。

9.根据权利要求8所述的产品,还包括指令,所述指令当被执行时,使得所述系统处理使用与所述计算环境相关联的私钥所生成的所述被保护的数据的签名,以及调用原语来将所述被保护的计算数据和所述签名传达到远程可信部件。

10.根据权利要求8所述的产品,还包括指令,所述指令当被执行时,使得系统使用与在远程机器上运行的远程可信部件相对应的公钥来保护所述加密密钥,其中所述公钥与所述产品通信。

11.根据权利要求10所述的产品,还包括指令,所述指令当被执行时,使得系统使用所述加密密钥来解密被保护的用户密钥以提取用户密钥,以及使用所述用户密钥来解密所述

隔离存储器区域中的秘密代码。

12.一种方法,包括:

在计算环境中生成隔离存储器区域以存储代码包,所述隔离存储器区域仅对于在所述隔离存储器区域中运行的代码可访问;

使用与所述计算环境的安全计算提供者相对应的私有证据密钥来生成签名;

使用所述签名以及所述代码包的密码学摘要,对所述代码包执行验证过程;以及

将针对所述代码包的验证结果传达到远程可信部件。

13.根据权利要求12所述的方法,还包括:传达被保护的用户密钥以将所述代码包的秘密代码变换成应用代码。

14.根据权利要求12所述的方法,还包括:将经加密的代码包加载到所述隔离存储器区域中。

15.根据权利要求12所述的方法,还包括:使用与所述计算环境相对应的公钥来处理包括经签名数据的消息,以及生成验证结果,所述验证结果指示消息是否源自所述隔离存储器区域中的所述可信部件。

# 在计算环境中保护计算数据的技术

## 背景技术

[0001] 对于开发人员(例如,软件应用开发人员)而言,保证他们所使用的数据机密性和数据完整性,变得越来越重要,尤其是当其程序涉及敏感数据时。复杂的安全威胁使得政府和私有组织由于延迟和支出用于预防/减轻这些威胁而花费大量资金。随着这些组织转向在网络上使用基于云的服务,而非维护本地部署的(on-premises)硬件,攻击者就有更多机会利用软件漏洞,并且危及其他组织的安全。由于云计算环境的分层特权结构,所以在这样的计算基础上运行的程序可能从特权软件代码继承软件漏洞,诸如操作系统部件或管理程序部件。

[0002] 各种代码分区方案为恶意攻击提供了相当多的机会,并且降低了在具有不同特权级别的单独执行环境中执行这些部分的益处和实用性。大量可信代码也会抑制关于正确性的任何有意义的检查。更进一步地,代码分区方案通常需要基本上手动的任务,其证明是容易出错和缓慢的。

[0003] 就这些和其他考虑而言,当前的改进是需要的。

## 发明内容

[0004] 以下给出简化的发明内容,以便提供对本文中所描述的一些新颖实施例的基本理解。本发明内容并非是广泛概述,也不是旨在标识关键/重要元件或描绘其范围。其唯一目的是以简化形式提出一些概念,作为稍后呈现的具体实施方式的序言。

[0005] 各种实施例通常指向经由安全硬件抽象在计算环境中提供安全计算的技术。如本文中所描述的,计算环境由安全计算提供者控制,并且可以指代基于云的环境或本地部署的(例如,本地)计算环境。安全计算提供者通常包括合适的安全硬件部件,诸如安全处理器。在计算环境的隔离存储器区域中,可以存储安全硬件不可知的并且与任何安全计算提供者一起操作的代码包。根据本文中所描述的各种实施例,使用经签名数据来验证代码包为可信代码并且认证消息数据源自隔离存储器区域的技术使得能够由不同的提供者进行安全计算。由代码包生成的消息数据可以被用于使用各种机制(诸如本文中所描述的那些机制以及还涵盖具有类似特征的那些机制)在隔离存储器区域中的可信代码与在远程机器中远程存储的可信代码之间共享秘密。

[0006] 一些实施例具体指向使得能够访问被存储在隔离存储器区域中的代码包的技术。代码包可以实现功能性,其被配置为对存储在外部存储装置中的数据执行一组计算。为代码包提供安全计算涉及在维持用于在隔离存储器区域和不可信代码部件之间进行通信的原语编程模型的同时,将部分或全部的包的数据和代码与不可信代码部件(例如,特权软件,诸如操作系统部件或虚拟机监控器部件)隔离。一般而言,原语编程模型是底层(安全)硬件的抽象,其仍然提供对所存储的数据的安全计算。安全计算可以通过在隔离代码包与在远程机器上运行的一个或多个远程可信部件之间建立一个或多个安全通信信道来增强。因为隔离代码包独立于底层硬件、软件和/或固件而操作,所以本文中所描述的各种实施例可以以任何硬件配置来实现。

[0007] 在一个实施例中，例如，一种装置可以包括逻辑，其在逻辑电路上操作以在计算环境中配置隔离存储器区域，用于与在隔离存储器区域之外运行的代码进行安全通信；使用与计算环境相对应的证据密钥(attestation key)生成经签名数据，该经签名数据包括被保护的加密密钥和用于认证被保护的加密密钥的签名；以及向远程可信部件传达经签名数据以访问被存储在隔离存储器区域中的秘密代码。其他实施例被描述并要求保护。

[0008] 为了实现前述和相关目的，在本文中结合以下描述和附图对某些说明性方面进行描述。这些方面指示可以实践本文中所公开的原理的各种方式，并且其所有方面和等同物均旨在处于所要求保护的主题的范围内。结合附图考虑下面的具体实施方式，其他优点和新颖特征将变得明显。

## 附图说明

[0009] 图1图示了在计算环境中保护计算数据的系统的实施例。

[0010] 图2图示了用于可信部件的操作环境的实施例。

[0011] 图3图示了用于具有支持区域的可信部件的操作环境的实施例。

[0012] 图4图示了可信部件与远程可信部件之间的密钥交换协议的实施例。

[0013] 图5图示了可信部件和远程可信部件之间的安全通信信道的实施例。

[0014] 图6图示了用于孤立存储器区域中的可信代码的隔离环境的实施例。

[0015] 图7图示了用于安全计算提供者的系统的实施例。

[0016] 图8图示了用于图1的系统的集中系统的实施例。

[0017] 图9图示了用于图1的系统的逻辑流程的实施例。

[0018] 图10图示了用于图5的可信部件的逻辑流程的实施例。

[0019] 图11图示了用于图4的远程可信部件的逻辑流程的实施例。

[0020] 图12图示了计算体系架构的实施例。

[0021] 图13图示了通信体系架构的实施例。

## 具体实施方式

[0022] 各种实施例指向计算环境中的应用编程接口(API)部件，其操作以在计算环境中隔离可信代码与不可信代码，并且当这种数据正在被不可信代码处理时，保护由该可信代码生成的数据。一般而言，API部件通过实现原语编程模型来提供安全硬件抽象层，通过该模型，不可信代码和可信代码建立安全连接或通信信道。不可信代码和可信代码二者均可以使用原语编程模型的原语函数来生成和管理隔离环境。经由原语编程模型，可信代码实现了用于保护在隔离环境和不可信代码之间的网络数据通信的加密协议。

[0023] 如本文中所描述的，隔离环境可以包括被存储在计算设备的存储器的隔离存储器区域中的各种计算机代码和数据。在一些实施例中，API部件的原语编程模型由不可信代码使用以配置隔离存储器区域，该隔离存储器区域具有通往可在与隔离存储器区域不同的存储器区域中执行的代码的安全通信信道，该代码包括在隔离存储器区域的外部运行的特权代码部件。除了其他之外，原语编程模型可以使得能够向不可信代码安全传达数据，以用于由不可信代码的功能中的一个功能处理、用于存储在外部存储装置中、和/或用于通过网络传输到远程机器。由API部件实现的原语编程模型还可以实现附加管理功能，诸如文件系统

操作、线程化、同步化、存储器分配和/或类似的功能。在一些实施例中，计算数据与签名(或另一认证代码)一起被传达到远程机器，并且用通过原语编程模型生成的加密密钥进行保护。签名确保加密密钥的完整性，并且向远程机器验证隔离存储器区域中的计算机代码尚未受到损害或毁坏。

[0024] 应用开发框架(例如，安全Hadoop)对其数据和代码进行分区，使得一部分与特权软件(例如，操作系统)隔离，但需要与计算环境内的底层硬件兼容。API部件的原语编程模型提供了与任何底层硬件的互操作性。在一些实施例中，API部件实现产生通信原语以安全地传达数据的功能。尽管这些通信原语中的一些通信原语可以包含或类似已知进程间通信的原语，但是本公开所设想的实施例不限于任何特定构造。API部件实现最少数目的功能来实现安全计算和安全通信，同时将对隔离存储器区域的访问仅限于该区域中的代码。因此，即使特权软件受到恶意管理员的损害或操作，攻击者也无法访问隔离存储器区域中的数据和代码。

[0025] 因此，实施例可以改善针对操作员、设备或网络的可负担性、可扩展性、模块性、可扩展性或互操作性。

[0026] 我们描述了云中的不可信代码如何使用由用户提供的一些代码创建隔离区域，以及隔离区域内部的可信代码如何与外部的代码进行通信。我们还描述了远程机器中的可信代码如何与隔离区域内部的可信代码建立安全通道。

[0027] 一般参照本文中所使用的符号和术语，下面的具体实施方式可以根据在计算机或计算机网络上执行的程序流程流程来呈现。本领域技术人员使用这些流程流程性描述和表示来将其工作的实质最有效地传达给本领域其他技术人员。

[0028] 流程这里通常被设想为导致所期望的结果的自洽操作序列。这些操作是需要对物理量进行物理操纵的那些操作。通常但不一定，这些量采取能够被存储、传送、组合、比较和以其他方式操纵的电、磁或光信号的形式。主要由于习惯用法，有时证明将这些信号称为比特、值、元素、符号、字符、术语、数字等是方便的。然而，应当指出，所有这些和类似术语都要与适当的物理量相关联，并且仅仅是适用于那些量的方便标签。

[0029] 进一步地，所执行的操纵经常以术语形式被提及，这些术语通常与由人类操作员执行的心理操作相关联，诸如添加或比较。在大多数情况下，在本文中所描述的形成一个或多个实施例的一部分的操作中的任一操作中，人类操作员的这种能力不是必需的或者是期望的。相反，这些操作是机器操作。用于执行各种实施例的操作的有用机器包括通用数字计算机或类似设备。

[0030] 各种实施例还涉及用于执行这些操作的装置或系统。该设备可以出于所需目的而被专门建造，或者可以包括如通过被存储在计算机中的计算机程序被选择性地激活或重新配置的通用计算机。本文中所呈现的流程并不固有地涉及特定计算机或其他装置。各种通用机器可以与按照本文中的教导编写的程序一起使用，或者可能证明构造更专业的装置以执行所需方法步骤是方便的。从所给出的描述中可以看出针对多种这些机器的所需结构。

[0031] 现在参考附图，其中相同的附图标记始终用于指代相同的元件。在以下描述中，出于解释的目的，对许多具体细节进行阐述，以提供对其的透彻理解。然而，明显的是，可以在没有这些具体细节的情况下，实践新颖实施例。在其他实例中，众所周知的结构和设备以框图形式示出，以便便于对其的描述。意图是覆盖与所要求保护的主题相一致的所有修改、等

同物和备选方案。

[0032] 图1图示了针对系统100的框图。在一个实施例中，系统100可以包括具有安全计算提供者120和一个或多个部件122-a的计算机实现的系统100。尽管图1中所示的系统100在特定拓扑中具有有限数目的元件，但是可以领会，系统100可以根据针对给定实现方式的需要在备选拓扑中包括更多或更少的元件。

[0033] 值得指出的是，如本文中所使用的“a”和“b”和“c”以及类似标志符是旨在表示任何正整数的变量。因此，例如，如果实现方式设置a＝5的值，则部件122-a的完整集合可以包括部件122-1、122-2、122-3和122-4。实施例在该上下文中不受限制。

[0034] 系统100可以包括安全计算提供者120，其控制计算环境。安全计算提供者120通常可以被布置为向在本地或远程操作的若干个计算设备提供计算服务。计算环境的一个示例包括具有运行各种应用的虚拟机的形式的处理资源和存储资源的配置。一个物理计算机可以被抽象为几个虚拟机，并且可替代地，两个或更多个物理计算设备可以分配处理能力和/或存储空间给在计算框架上的执行处理作业，例如，对大数据集执行并行计算的大集合。

[0035] 本文中所描述的各种实施例指的是应用设计接口(API)部件122-1，其操作以生成包括若干个原语函数的原语编程服务。原语函数可以表示最少数目的原语，其适合于支持不同的安全计算提供者并且促进与每个提供者的功能性的交互。API部件122-1可以进一步操作以提供对该原语编程服务的访问，例如，对隔离存储器区域122-2内的可信代码部件的访问。经由API部件122-1，在隔离存储器区域122-2内运行的可信代码可以与在远程机器上运行的远程可信部件执行密钥交换协议。

[0036] 密钥交换协议的一个特征是证据密钥122-3，其对于安全计算提供者120是私有的，用于认证从隔离存储器区域122-2传达出的数据。证据密钥122-3可以具体地与安全计算提供器120相对应；因此，使用该密钥为一些数据生成数字签名确保了当从由安全计算提供者120控制的计算环境中传送出时该数据的完整性。该数据可以使用与证据密钥122-3相对应的公钥来验证。因为远程可信部件可以确认数据的真实性，所以安全计算提供者120可以确保数据在处在隔离存储器区域122-2的外部时没有被损害。

[0037] 来自证据密钥122-3的数字签名和来自API部件122-1的被保护的加密密钥的组合方面为在隔离存储器区域122-2与该区域外部的任何可信部件之间被传达的数据提供附加的数据机密性和完整性。在可信部件包括远程存储的代码的实施例中，因为被保护的加密密钥通过对于该代码已知的方案被加密，所以加密密钥不太可能被损害。因此，远程存储的代码可以确保与隔离存储器区域122-2的通信是安全的。

[0038] 根据一个实施例，不可信代码122-4在隔离存储器区域122-2外部的存储器区域中被执行，并且通过API部件122-1与该区域通信。一旦密钥交换协议成功完成，应用代码就可以例如与其他可信部件并行地对所存储的数据运行计算的集合。在隔离存储器区域中运行的应用代码可以使用I/O控制代码来指示不可信代码122-4执行各种计算任务。因此，API部件122-1提供了在隔离存储器区域122-2中运行的代码与在隔离存储器区域122-2外部的存储器区域中运行的部件之间的安全通信。这些部件可以包括远程机器上的远程存储的代码或不可信代码122-4。

[0039] 不可信代码122-4可以在存储器内创建隔离环境，并且用计算机代码和/或数据来配置该隔离环境。为了通过示例说明，API部件122-1实现以下函数IsolatedRegionCreate

(),其当被调用时,创建隔离环境并且将由packagePath自变量指定的计算机代码包加载到该环境中:HANDLE

[0040] IsolatedRegionCreate(

[0041] _In_LPCTSTR packagePath,

[0042] _In_ISOLATION_PROVIDER isolationProvider,

[0043] _In_opt_CALL_OUT_HANDLER callOutHandler

[0044] )

[0045] 在上文示例中,isolationProvider标识安全计算服务的底层提供者,例如,VSM。packagePath自变量可以指代全局或云文件系统而非本地文件系统中的文件数据。callOutHandler标识不可信代码中的函数,其可以处理从区域内部发送的IO控制代码。该包是代码(例如,可信应用代码)和数据的容器。一个示例包是可从移动应用平台下载的移动应用包。该包还包括配置参数,诸如区域的大小。

[0046] 不可信代码122-4可以调用隔离存储器区域122-2中的代码。实现这一点的一种方式是向隔离存储器区域122-2发送输入/输出(IO)控制代码(例如,IOCTL代码):

[0047] IRIO_RESULT

[0048] IsolatedRegionIOControl(

[0049] _In_HANDLE region,

[0050] _In_DWORD callInId,

[0051] _In_reads_bytes_opt_(inputBufferBytes)LPCVOID inputBuffer,

[0052] _In_SIZE_T inputBufferBytes,

[0053] _Out_writes_bytes_to_opt_(outputBufferBytes,*bytesReturned)LPVOID outputBuffer,

[0054] _In_SIZE_T outputBufferBytes,

[0055] _Out_opt_PSIZE_T bytesReturned

[0056] )

[0057] 区域自变量定义了隔离存储器区域122-2的地址或位置。callInID自变量标识了隔离存储器区域122-2中的函数,其被配置为处理来自不可信代码122-4的控制代码(或其他通信原语)。callInID自变量可以由代码包附带的信息提供。inputBuffer自变量和outputBuffer自变量是分别存储控制代码和返回结果的存储器缓冲器。最后,不可信代码122-4可以例如通过调用以下函数来破坏隔离存储器区域:

[0058] VOID IsolatedRegionClose(_In_HANDLE region)

[0059] 如本文中所指出的,由API部件122-1实现的功能可以被扩展以执行诸如存储器管理功能之类的附加任务。作为示例,VirtualAlloc()和VirtualFree()可以在隔离存储器区域122-2内部实现,以动态分配/释放虚拟存储器。

[0060] 图2图示了用于系统100的操作环境200的实施例。如图2所示,API部件122-1从在隔离存储器区域202内运行的可信代码202接收诸如通信原语的控制指示。一些控制指示指令API部件122-1将经签名数据214传达到远程机器。在远程机器上运行的远程可信部件206可以对经签名数据122-3执行验证过程以确定这种数据是否已经被盗用和/或保护远程机器免于恶意活动。

[0061] 根据一个示例实施例，可信代码202和远程可信部件206参与密钥交换协议，通过该协议，一个或多个加密密钥通过不可信代码被安全地传达。可信代码202的一个示例实现方式经由函数调用来调用通信原语以指令API部件122-1生成加密密钥。可信代码202例如通过使用远程可信部件206的公钥对加密密钥进行加密来保护加密密钥。可信代码202调用另一原语函数来请求用于被保护的加密密钥的签名，其当时被存储在经签名数据214中。应当领会，可以实现许多备选的密钥交换协议。比如，作为一个备选方案，可信代码202可以使用另一加密方案。

[0062] 在一个实施例中，远程可信部件206对代码(例如，函数库)和数据进行加密，并且将它们绑定到代码包208中。用于生成加密代码包208的加密密钥被称为用户密钥210。代码包208的一部分可以是公开代码并且可以被存储在可信代码202中。另一部分可以包括支持代码文件并且还可以被存储在可信代码中。另一部分可以作为隔离存储器区域122-2中的秘密代码212而保持安全，直到建立了安全通信信道为止。在一些实施例中，代码包208包括元数据，其用于标识一个或多个函数，其处理来自在除隔离存储器区域122-2以外的存储器区域中执行的代码的通信原语(例如，I/O代码)。元数据中的每个函数定义都可以实现即时通信并且控制用于不可信代码的应用功能性。

[0063] 在一个实施例中，远程可信部件206使用加密密钥来保护用户密钥210，其是在将秘密代码212传送到隔离存储器区域122-2之前初始加密秘密代码212的加密密钥。如本文中关于图1所描述的，秘密代码212可以构成计算机代码包208的一部分，其可以被安置于隔离存储器区域122-2中，以对存储在外部存储装置中的数据执行安全计算。根据一个实施例，秘密代码212包括要对用于相当数目的客户端计算设备的实质数据集执行的并行处理作业(例如，映射函数和约简(reduce)函数)。

[0064] 可信部件202接收用户密钥210并且解密秘密代码212以访问定义要对存储的数据执行的一组计算的并行处理作业信息。秘密代码212将并行处理作业分布在隔离存储器区域122-2内的一个或多个资源中，以生成计算数据，该计算数据使用在密钥交换协议期间生成的加密密钥来保护。可信部件202请求用于被保护的计算数据的签名，并且签名和被保护的计算数据都作为经签名数据214被传达到远程可信部件206。使用一个或多个通信原语(例如，I/O控制代码)，可信代码202生成消息以存储经签名数据214，并且将该消息写入到用于传达到API部件122-1的存储器缓冲器。应当领会，消息内容可以被用于以多种方式(例如，Diffie-Hellman密钥交换)在可信代码202和远程可信部件206之间建立共享秘密数据。本文中所描述的实施例支持这些安全信道建立机制中的几个安全信道建立机制，并且提供用于选择特定机制的机制。

[0065] 远程可信部件206转而使用API部件122-1来验证消息内容的完整性和机密性。一个示例实现方式确定签名是否由与计算环境的安全计算提供者120(例如，而非恶意提供者)相对应的私有证据密钥产生，以及消息内容是否由可信代码202生成。另一示例实现方式确定被保护的计算数据是否由秘密代码212(例如，而非受损害的代码包)产生。使用一个或多个通信原语(例如，I/O控制码)，可信部件202将经签名数据214写入和/或读取到存储器缓冲器，该经签名数据214被传达到API部件122-1。

[0066] 经签名数据214可以包括签名或由安全计算提供者的证据密钥生成的另一认证码。在一些实施例中，该密钥可以是公钥密码学方案下的私钥，并且特定地与相关联的安全

计算提供者相对应。在一些实施例中，经签名数据还包括对于在安全计算提供者处操作的不可信代码部件不知晓的加密密钥。因为加密密钥被保护使得不受不可信部件影响，所以密钥可以被传达到在隔离存储器区域外部运行的代码，而不会受到损害。如果该代码在远程计算机上运行，则代码可以通过检查签名以确定密钥在处于隔离存储器区域的外部时是否受到损害来验证密钥。因此，经签名数据214向远程机器的系统验证密钥的完整性和机密性。

[0067] 图3图示了具有用于支持部件304的支持区域的、隔离存储器区域122-2的操作环境300的实施例。在该实施例中，可信代码202与支持部件304一起操作以实现针对存储在外部存储装置中的数据的安全计算。

[0068] 为了通过示例说明，安全计算提供者120可以操作云计算环境，其中每个机器在存储器中创建支持区域302，该支持区域与机器上的其他地方运行的不可信代码隔离。比如，使用API部件122-1，不可信代码可以用IsolatedRegionCreate()创建支持区域302，并且用支持部件304加载该区域。支持部件304实现一个或多个管理功能，其允许远程可信部件206向云计算环境安全地发送私有计算机代码以供存储在隔离存储器区域122-2中。用于实现支持部件304的代码是不重要的，并且可以公开。

[0069] 如本文中所描述的，不可信代码通过IsolatedRegionIOControl()函数调用来调用支持部件304中的函数。支持部件304按照对于另一代码部件不知晓的加密方案来生成公有-私有密钥对。这个公开-私有加密密钥对可以特定于已经被分配给隔离存储器区域122-2的处理资源。支持部件304调用API部件122-1上的原语函数以生成用于加密私钥的密封密钥。支持部件304调用另一原语函数来生成针对公钥的签名。使用签名和被保护的私有加密密钥，支持部件304参与与远程机器的密钥交换协议。

[0070] 为了通过示例说明，考虑隔离存储器区域122-2在图3中被描绘为具有秘密代码212，其是用于对存储的数据执行一组计算的被保护的计算机代码。秘密代码212内的函数可以用远程可信部件206已知的秘密密钥加密。在API部件122-1验证了上文所提及的公钥的证据之后，远程可信部件206使用公钥对针对秘密代码212的秘密密钥进行加密。因此，该秘密密钥可以被称为用户密钥，其类似于图2的用户密钥208。

[0071] 为了保护支持区域302与隔离存储器区域122-2(在该实例中支持部件304构成为在隔离存储器区域122-2的外部运行的代码)之间的通信，支持部件304和可信代码202启动密钥交换协议，使得可信代码202接收用于解密秘密代码212的秘密密钥，并且支持部件304接收用于保护秘密密钥以及未来可能的其他通信的私钥。一旦解密，秘密代码212的函数就被合并到可信代码202中，并且私钥可以被那些函数用来保护被写入到不可信代码或从该不可信代码读取的数据(例如，加密值对)。

[0072] 图4图示了可信部件202与远程可信部件206之间的密钥交换协议400的实施例。

[0073] 如本文中所描述的，在计算环境120中运行的不可信代码调用API部件122-1上的函数以创建隔离存储器区域并且将计算机代码加载到该区域中。执行该计算机代码生成可信部件202并且启动密钥交换协议400。API部件122-1的一个示例实现方式将该包加载到隔离存储器区域中，并且向可信部件202发送用于远程可信部件的公钥。API部件122-1可以将公钥存储在消息的存储器缓冲器中，该消息被传达到可信部件202以启动设置过程402。该公钥可以特定于特定的远程机器。该消息还包括配置参数，诸如区域的大小。

[0074] 为了开始设置过程402，可信部件202可以调用原语函数调用404，并且作为响应，API部件122-1生成并且返回加密密钥以保护可信部件202与远程可信部件之间的通信。这些密钥允许可信代码对数据进行加密，将其保存在外部存储装置中，然后在后续执行中将其解密。

[0075] 在原语函数404的以下示例实现方式中，对IsolatedAppGetKey的函数调用请求与KeyID相对应的、具有参数为缓冲器keyBuffer中的keyBufferBytesRequired和keyBufferBytes的密钥：

```
BOOL
IsolatedAppGetKey(
    _In_ KeyId keyId,
[0076]  _Out_writes_bytes_to_(keyBufferBytes,
*keyBufferBytesRequired) LPVOID keyBuffer,
    _In_ SIZE_T keyBufferBytes,
    _Always_(_Out_) PSIZE_T keyBufferBytesRequired )
```

[0077] 在另一操作中，可信部件202可以用公钥来保护加密密钥并且调用原语函数406，并且作为响应，API部件122生成并且返回被保护的加密密钥的数字签名。在又一操作中，可信部件202可以调用原语函数408以将数字签名和被保护的加密密钥写入存储器缓冲器中，并且将消息传达到API部件122-1。作为响应，API部件122-1将消息发送到远程可信部件206。

[0078] 在原语函数404的以下示例实现方式中，对IsolatedAppSignMessage的函数调用指示API部件以生成针对消息内容的数字签名并且将数字签名存储在outputBuffer中。

```
BOOL
IsolatedAppSignMessage(
    _In_reads_bytes_(messageBytes) LPCVOID message,
[0079]  _ In_ SIZE_T messageBytes,
    Out_writes_bytes_to_(outputBufferBytes,
*outputBufferBytesRequired) LPVOID outputBuffer,
    _In_ SIZE_T outputBufferBytes,
    _Always_(_Out_) PSIZE_T  outputBufferBytesRequired
[0080] )
```

[0081] 远程可信部件206可以从存储器缓冲器读取数据并且提取数字签名和被保护的加密密钥。在一个操作中，远程可信部件调用原语函数410以生成包的真实拷贝的密码学摘要。在另一操作中，远程可信部件206调用原语函数412以确定数字签名是否由具有密码学摘要的包生成。API部件122-1可以发送验证结果，其指示存储器缓冲器中的数据是安全的，

或者指示数据已经被盗用,或至少是不正确的。

[0082] 在原语函数404的以下示例实现方式中,对IsolatedAppIoControl的函数调用传达inputBuffer中的IO控制代码,并且API部件122-1执行IO控制代码并且返回outputBuffer中的结果:

[0083]

```
IRIO_RESULT
IsolatedAppIoControl(
    _In_          DWORD     callOutId,
    _In_reads_bytes_opt_(inputBufferBytes) LPCVOID inputBuffer,
    _In_          SIZE_T    inputBufferBytes,
    _Out_writes_bytes_to_opt_(outputBufferBytes,      *bytesReturned)
LPVOID    outputBuffer,
    _In_          SIZE_T    outputBufferBytes,
    _Out_opt_ PSIZE_T bytesReturned  )
```

[0084] 可信代码202调用上文原语函数以指示API部件122-1向远程可信部件传达存储器缓冲器inputBuffer中的消息。

[0085] 建立安全通信信道的一个示例机制操作这些函数来验证该消息源自隔离存储器区域中的可信代码202:

[0086]

```
BOOL
IsolatedRegionGetDigest(
    _In_ LPCTSTR packagePath,
    _Out_writes_bytes_to_(regionDigestBytes,
*regionDigestBytesRequired)
    LPVOID regionDigest,
```

```
    _In_ SIZE_T regionDigestBytes,
    _Out_ PSIZE_T regionDigestBytesRequired
)
BOOL
IsolatedRegionCheckSignature(
    _In_        ISOLATION_PROVIDER isolationProvider,
    _In_reads_(regionDigestBytes) LPCVOID regionDigest,
    _In_ SIZE_T regionDigestBytes,
    _In_reads_(messageBytes) LPCVOID message,
    _In_ SIZE_T messageBytes,
    _In_reads_(signatureBytes) LPCVOID signature,
    _In_ SIZE_T signatureBytes
)
```

[0087]

[0088] IsolatedRegionGetDigest()函数返回密码学摘要,其确定性地标识位于本地文件系统或全局或基于云的文件系统中的、在packagePath中指示的地址处的代码包。该代码包可以指示应用的干净或未被毁坏的版本。密码学摘要可以作为regionDigest自变量与安全计算提供者120的标识符一起被传递到原语函数IsolatedRegionCheckSignature()。如果缓冲器中的消息由隔离区域上的代码产生,则该函数返回布尔值“真”,该隔离区域由安全计算提供者创建。例如,该函数可以使用与安全计算提供者120相对应的公开证据密钥来认证该签名,以确认消息内容没有被损害。作为另一示例,该函数可以通过将上述密码学摘要与针对这种代码产生的摘要进行比较来验证在隔离存储器区域内部运行的代码尚未受到损害,并且匹配指示代码包的未更改的副本。然而,不匹配指示隔离存储器区域内部的代码与干净版本不一样。因此,经签名消息/证据消息的内容可以被用于以多种方式(例如,Diffie-Hellman密钥交换)在可信代码202与远程可信部件206之间共享秘密数据。本文中所描述的实施例支持这些安全信道建立机制中的几个安全信道建立机制,并且让用户选择使用哪一个。

[0089] 除了隔离存储器区域122-2之外,以下描述适用于实现支持部件的一个或多个示例实施例,诸如图3的支持部件304。该支持部件调用API部件122-1上的原语函数IsolatedAppGetKey()来生成加密密钥以用作加密处理器私钥的密封密钥。支持部件可以调用原语函数IsolatedAppSignMessage()来签名处理器公钥,然后发布密钥。

[0090] 在用户开发应用代码(例如,映射和约简函数)时,远程可信部件使用秘密密钥编译和加密应用代码,并且将经加密的应用代码与公开代码绑定以产生代码包(例如,代码库,诸如动态链接库(DLL)文件)。远程可信部件206可以使用函数IsolatedRegionCheckSignature()验证处理器公钥的证据,然后使用处理器公钥对被用于加密应用代码的秘密密钥进行加密。

[0091] 云计算环境中的不可信代码通过函数IsolatedRegionCreate()将代码包加载到隔离存储器区域中，并且使用函数IsolatedRegionIOControl()以指示公开代码生成新的随机对称密钥来与隔离存储器区域建立安全通信信道。用于该区域的新密钥使用处理器公钥进行加密，并且用户区域通过调用函数IsolatedAppSignMessage()来获取签名。然后，经加密的新密钥被发送到支持区域。支持区域使用IsolatedRegionCheckSignature()来验证签名，并且解密经加密的新密钥。然后，支持区域解密用于应用代码(其已使用处理器公钥加密)的秘密密钥，使用来自隔离存储器区域的新密钥加密该秘密密钥，并且将经加密的秘密密钥发送到隔离存储器区域。在该区域内运行的可信代码解密该秘密密钥并且解密应用代码，然后通过调用函数IsolatedRegionIOControl()来调用应用代码中的函数，以向用于处理指向应用代码的IO控制代码的处理程序(handler)传达原语IO控制代码(或另一控制指令)。可信代码通过将IO控制代码指向不可信代码中的特定处理程序来准备用于外部存储装置的被保护的计算数据。例如，可信代码调用函数IsolatedAppIOControl()来将IO控制代码指向不可信代码的处理程序，以指示该处理程序读取经加密的密钥-值对；并且在对那些“对”执行计算之后，可信代码调用函数IsolatedAppIOControl()来将IO控制代码指向不可信代码的处理程序，以指示该处理程序写入经加密的密钥-值对。

[0092] 图5图示了可信代码202和远程可信部件206之间的安全通信信道500的实施例。当在隔离存储器区域中运行的应用代码正在执行用于并行处理作业的一组计算时，可信代码202调用原语函数502以从不可信码122-4读取数据(例如，加密值对)。不可信代码122-4返回加密值对。可信代码202可以调用该函数并且对加密值对执行一个或多个计算。在另一操作中，可信代码202调用原语函数以将加密值对写入到外部存储装置，并且不可信代码122-4在完成时返回确认。

[0093] 为了将这些“对”发送到远程可信部件，可信代码202调用原语函数404来处理用于认证加密值对的数字签名。为了发送具有数字签名和加密值对的消息，可信代码202调用原语函数406以将消息传达到远程可信部件。在解密被保护的计算数据之前，远程可信部件调用原语函数412来验证数字签名。

[0094] 图6图示了用于隔离存储器区域122-2中的可信代码和数据的隔离环境600的实施例。应用代码602可以是加密形式的秘密代码。公开代码604包括用于应用代码602的接口，用于对存储的数据执行一组计算以产生计算数据606。应用代码602可以使用公开代码604来保护具有加密密钥608的计算数据606，并且将安全计算数据606传达到隔离存储器区域122-2外部的不可信代码。公开代码604还可以通过向API部件请求签名610并且在消息的存储器缓冲器中传达该签名610来将签名610传递到在与隔离存储器122-2不同的存储器区域中执行的代码。如本文中所描述的，在不同的存储区域中执行的代码可以指的是在云计算环境中运行的不可信代码或由远程机器执行的远程存储的代码。

[0095] 在一些实施例中，公开代码604可以利用元数据612来标识应用代码602的存储器区域，该存储器区域包括一个或多个函数，其被配置为处理来自在隔离存储器区域122-2外部运行的不可信代码的控制指令(例如，通信原语，诸如I/O控制代码)。经由诸如本文中所描述的那些应用编程接口(API)部件的应用编程接口部件，不可信代码可以调用原语函数来传达控制指令以调用这些函数中的一些函数。因此，元数据612通过(例如，较低级别)进程间通信原语使得能够访问由应用代码602实现的复杂功能性。

[0096] 图7图示了用于安全计算提供者720的计算环境700的实施例。安全计算提供者720描绘了图1的安全计算提供者120的备选方案。在该实施例中，图1的支持部件3可以访问在处理单元中运行的应用代码。应当领会，计算环境700表示图1的系统100的一个备选方案，并且在本公开中设想了其他备选方案和修改。

[0097] 处于计算环境700控制下的安全计算提供者720包括处理器电路730和隔离存储器区域750，该隔离存储器区域750还包括支持部件304和处理器密钥752。隔离存储器区域750可以被配置为类似于图1的隔离存储器区域122-2。应用代码602使用处理器电路730来正在对数据执行一组计算。如本文中所描述的，支持部件302生成一个或多个处理器密钥752以保护处理单元级别的计算。因此，被保护的计算数据608可以被快速解密/加密，从而在安全风险很小或没有的情况下，增强了计算吞吐量。

[0098] 图8图示了集中式系统800的框图。该集中式系统800可以在单个计算实体中(诸如完全在单个设备820内)实现用于系统100的一些或全部结构和/或操作。

[0099] 设备820可以包括任何电子设备，其能够接收、处理和发送用于系统100的信息。电子设备的示例可以包括但不限于超移动设备、移动设备、个人数字助理(PDA)、移动计算设备、智能电话、电话、数字电话、蜂窝电话、电子书阅读器、手机、单向寻呼机、双向寻呼机、消息收发设备、计算机、个人计算机(PC)、台式电脑、膝上型电脑、笔记本电脑、上网本电脑、手持电脑、平板电脑、服务器、服务器阵列或服务器群、web服务器、网络服务器、互联网服务器、工作站、微型计算机、大型计算机、超级计算机、网络家电、web家电、分布式计算系统、多处理器系统、基于处理器的系统、消费电子产品、可编程消费电子产品、游戏设备、电视机、数字电视、机顶盒、无线接入点、基站、用户站、移动用户中心、无线电网络控制器、路由器、集线器、网关、网桥、交换机、机器或其组合。本实施例在该上下文中不受限制。

[0100] 设备820可以使用处理部件830来执行用于系统100的处理操作或逻辑。处理部件830可以包括各种硬件元件、软件元件或两者的组合。硬件元件的示例可以包括设备、逻辑设备、部件、处理器、微处理器、电路、处理器电路、电路元件(例如，晶体管、电阻器、电容器、电感器等)、集成电路、专用集成电路(ASIC)、可编程逻辑设备(PLD)、数字信号处理器(DSP)、现场可编程门阵列(FPGA)、专用标准产品(ASSP)、片上系统(SOC)、复杂可编程逻辑设备(CPLD)、存储器单元、逻辑门、寄存器、半导体设备、芯片、微芯片、芯片组等。软件元件的示例可以包括软件部件、程序、应用、计算机程序、应用程序、系统程序、软件开发程序、机器程序、操作系统软件、中间件、固件、软件模块、例程、子例程、功能、方法、程序、软件接口、应用编程接口(API)、指令集、计算代码、计算机代码、代码段、计算机代码段、单词、值、符号或其任何组合。确定是否使用硬件元件和/或软件元件来实现实施例可以按照任何数目的因素而变化，诸如期望的计算速率、功率水平、耐热性、处理周期预算、输入数据速率、输出数据速率、存储器资源、数据总线速度和其他设计或性能约束，如给定实现方式所期望的。

[0101] 设备820可以使用通信部件840来执行用于系统100的通信操作或逻辑。通信部件840可以实现任何公知的通信技术和协议，诸如适合与分组交换网络(例如，诸如互联网的公共网络、诸如企业内联网的私有网络等)、电路交换网络(例如，公共交换电话网络)或分组交换网络和电路交换网络的组合(具有合适的网关和转换器)一起使用的技术。通信部件840可以包括各种类型的标准通信元件，诸如一个或多个通信接口、网络接口、网络接口卡(NIC)、无线电、无线发射器/接收器(收发器)、有线和/或无线通信介质、物理连接器等。作

为示例而非限制，通信介质812，842包括有线通信介质和无线通信介质。有线通信介质的示例可以包括导线、电缆、金属引线、印刷电路板(PCB)、背板、交换结构(switch fabrics)、半导体材料、双绞线、同轴电缆、光纤、传播信号等。无线通信媒体的示例可以包括声学、射频(RF)频谱、红外和其他无线介质。

[0102] 设备820可以经由通信部件840分别使用通信信号814，844分别通过通信介质812，842与其他设备810，850通信。设备810，850可以针对给定实现方式的需要位于设备820的内部或外部。

[0103] 如本文中所描述的，在远程机器上运行的可信部件期望安全通信信道，其具有在图8的计算环境的隔离存储器区域中运行的应用代码。安全计算提供者的API部件可以经由原语编程模型建立具有隔离存储器区域的安全通信信道。使用调用通信原语的函数调用，不可信代码和可信代码通过实现用于保护诸如传输层安全性(TLS)、安全套接字层(SSL)和/或类似物的网络数据业务的加密协议来建立安全连接或通信信道。

[0104] 本文中所包括的是表示用于执行所公开的体系架构的新颖方面的示例性方法的流程图集合。虽然出于解释的简化目的，本文中所示出的一种或多种方法(例如，以流程图表或流程图的形式)被示出和描述为一系列动作，但是应当理解和领会，方法不受动作的次序的限制，因为一些动作可以据此以不同次序发生和/或与除了本文中所示出和描述的动作之外的其他动作同时发生。例如，本领域的技术人员将理解和领会，方法可以可替代地诸如在状态图上被表示为一系列相关状态或事件。此外，新颖的实现方式可能并不需要在方法中所图示的所有动作。

[0105] 图9图示了用于图1的系统的逻辑流程900的实施例。该逻辑流程900可以表示由本文中所描述的一个或多个实施例执行的操作中的一些或全部操作。在图9所示的图示的实施例中，逻辑流程900可以由图1的API部件122-1执行，以在框906处建立可信代码与在隔离存储器区域外部运行的代码之间的安全通信信道。

[0106] 例如，当计算环境中的不可信代码根据某些参数(例如，大小)调用原语函数来配置隔离存储器区域时，在框902处，逻辑流程900可以在计算环境中生成隔离存储器区域，并且将代码包存储在该区域中。逻辑流程900可以使用私有证据密钥来生成签名，该私有证据密钥特定地对应于安全计算提供者120，其控制计算环境。

[0107] 逻辑流程900可以在框906处执行验证过程，在该验证过程期间，代码包被认证为在远程机器中运行的远程可信部件。逻辑流程900可以执行验证过程以完成密钥交换协议，使得可信代码向远程可信部件提供使用公开密钥和签名保护的加密密钥以认证加密密钥。例如，公钥可以与证书相对应，该证书与已经请求计算服务的远程机器相对应。作为响应，远程可信部件返回关于如何访问代码包的数据，例如通过使用加密密钥保护用户密钥，并且经由对API部件的函数调用向可信代码传达该被保护的用户密钥。函数调用可能导致控制代码传达到可信代码，其提示该代码读取被保护的用户密钥，解密用户密钥，然后解密代码包中的秘密代码以访问代码包的功能性。

[0108] 比如，API部件和不可信代码可以指令隔离存储器区域中的可信代码来启动用于密钥交换协议的设置过程。API部件可以生成加密密钥并且将其传达到隔离存储器区域。在可信代码保护加密密钥之后，可信代码请求API部件使用与控制计算环境的安全计算提供者相对应的私钥来生成签名。API部件将包括签名的消息传达到在远程机器上运行的远程

可信部件。在密钥交换协议之后，API部件对消息执行验证过程以提取消息的内容并且确定如何访问代码包。如果成功，则验证过程证明经签名消息/证据消息源自隔离存储器区域中的可信代码。API部件可以将被保护的用户密钥传达到在隔离存储器区域中运行的可信代码。API部件可以生成用于干净代码包的密码学摘要，并且在该摘要与始发该消息的代码包的密码学摘要之间执行比较。

[0109] 逻辑流程900可以继续在框908处将验证结果传达到在远程机器上运行的远程可信部件。框908处的逻辑流程完成了对隔离存储器区域和在该区域中运行的代码的安全通信信道的建立。至少由于这个原因，逻辑流程900可以继续执行到更高级别的计算。这些计算涉及比进程间通信原语(例如，IO控制代码)更复杂的控制指令。为了说明，作为一个选项，逻辑流程可以继续进行到框908，并且将安全数据(例如，加密值对)存储在云文件系统中，其中外部存储装置作为一个文件系统出现。作为另一选项，逻辑流程900可以将IO控制代码传达到不可信代码，并且调用函数(例如，硬件驱动器函数)。实施例不限于此示例。

[0110] 图10图示了逻辑流程1000的一个实施例。该逻辑流程1000可以表示由本文中所描述的一个或多个实施例执行的操作中的一些或全部操作。

[0111] 在图10所示的图示的实施例中，逻辑流程1000可以在框1002处处理用户密钥并且解密存储在隔离存储器区域中的秘密代码包。经解密的代码包(即，现在可信的代码)中的映射和约简函数可以定义一组计算。逻辑流程1000在框1004处可以运行映射和约简函数以对所存储的数据执行一组计算并且生成计算数据。逻辑流程1000可以使用对于在隔离存储器区域外部运行的不可信代码不知晓的加密密钥来保护计算数据。逻辑流程1000在框1006处可以调用原语函数来生成用于安全计算数据的签名。逻辑流程1000在框1008处调用通信原语，其操作以将被保护的计算数据写入不可信代码。例如，逻辑流程1000可以指示不可信代码将被保护的计算数据存储在外部存储装置中。实施例不限于该示例。

[0112] 图11图示了用于图4的远程可信部件的逻辑流程的实施例。逻辑流程1100可以表示由本文中所描述的一个或多个实施例执行的操作中的一些或全部操作。

[0113] 在图11所示的图示的实施例中，逻辑流程1100在框1102处开始，其中逻辑流程1100加密代码包并且将加密的代码包与公钥一起传达到计算环境。逻辑流程1100可以在框1104处处理经签名消息/证据消息，并且从该消息中的存储器缓冲器中提取经签名数据。逻辑流程1100在框1106启动过程，以验证经签名数据源自秘密代码包，因而尚未被篡改或损害。如果被验证，则经签名消息/证据消息的内容可以被用于以多种方式(例如，Diffie-Hellman密钥交换)在隔离环境中的可信代码与远程机器中的远程可信部件之间共享秘密数据。本文中所描述的实施例支持这些安全信道建立机制中的几个安全信道建立机制，并且让用户选择使用哪一个。

[0114] 逻辑流程1200可以在框1106处执行关于经签名数据是否已经被损害的确定，并且在框1108处拒绝与隔离环境的连接，或者在框1110处接受连接。用于验证过程的多个示例实施例在本文中被描述，并且这些示例中的任一个示例都可以用于实施这样的确定。比如，如果经签名数据不能使用针对安全计算提供者的公开证据密钥来验证，则这样看来，消息内容已被污染。作为另一示例，如果秘密代码包的密码学摘要与存储在远程机器上的版本的摘要不匹配，那么代码包可能已经被更改，指示隔离环境已经被损害。如果逻辑流程1100确定经签名数据是安全的，则逻辑流程1000在框1110处可以解密被存储在消息中的密码密

钥并且保护用于传达到可信代码的用户密钥。如本文中所描述的，用户密钥包括用于解密秘密代码包的加密密钥。实施例不限于该示例。

[0115] 图12图示了适于实现如先前所描述的各种实施例的示例性计算体系架构1200的实施例。在一个实施例中，计算体系架构1200可以包括或被实现为电子设备的一部分。电子设备的示例可以包括参考图8描述的那些设备等等。实施例在该上下文中不受限制。

[0116] 如在本申请中所使用的，术语“系统”和“部件”旨在指代计算机相关实体，其可以是硬件、硬件和软件的组合、软件、或执行中的软件，其示例由示例性计算体系架构1200提供。例如，部件可以是但不限于在处理器上运行的进程、处理器、硬盘驱动器、(光和/或磁存储介质的)多个存储驱动器、对象、可执行文件、执行线程、程序和/或计算机。通过说明，在服务器上运行的应用和服务器都可以是部件。一个或多个部件可以驻留在进程和/或执行线程内，并且部件可以位于一台计算机上和/或分布在两台或更多台计算机之间。进一步地，部件可以通过各种类型的通信介质彼此通信地耦合以协调操作。协调可能涉及信息的单向或双向交换。比如，部件可以传达形式为通过通信介质传达的信号的信息。该信息可以被实现为分配给各种信号线的信号。在这种分配中，每个消息都是信号。然而，其他实施例可以可替代地采用数据消息。这样的数据消息可以通过各种连接发送。示例性连接包括并行接口、串行接口和总线接口。

[0117] 计算体系架构1200包括各种通用计算元件，诸如一个或多个处理器、多核处理器、协处理器、存储器单元、芯片组、控制器、外设、接口、振荡器、定时设备、视频卡、声卡、多媒体输入/输出(I/O)部件、电源等。然而，实施例不限于由计算体系架构1200来实现。

[0118] 如图12所示，计算体系架构1200包括处理单元1204、系统存储器1206和系统总线1208。处理单元1204可以是各种商业上可用的处理器中的任一种处理器，包括但不限于AMD®Athlon®、Duron®和Opteron®处理器；ARM®应用、嵌入式和安全处理器；IBM®和Motorola®DragonBall®和PowerPC®处理器；IBM和Sony®Cell处理器；Intel®Celeron®、Core(2)Duo®、Itanium®、Pentium®、Xeon®和XScale®处理器；以及类似处理器。双微处理器、多核处理器和其他多处理器体系架构也可以被采用为处理单元1204。

[0119] 系统总线1208为系统部件提供到处理单元1204的接口，该系统部件包括但不限于系统存储器1206。系统总线1208可以是几种类型的总线结构中的任一种，其可以进一步互连到存储器总线(具有或不具有存储器控制器)、外围总线和使用多种商业上可获得的总线体系架构中的任一种的本地总线。接口适配器可以经由插槽体系架构连接到系统总线1208。示例插槽体系架构可以包括但不限于加速图形端口(AGP)、卡总线、(扩展)工业标准体系架构((E)ISA)、微通道体系架构(MCA)、NuBus、外围部件互连(扩展)(PCI(X))、PCI Express、个人计算机存储器卡国际协会(PCMCIA)等。

[0120] 计算体系架构1200可以包括或实现各种制品。制品可以包括用于存储逻辑的计算机可读存储介质。计算机可读存储介质的示例可以包括能够存储电子数据的任何有形介质，包括易失性存储器或非易失性存储器、可移除或不可移除存储器、可擦除或不可擦除存储器、可写入或可重写存储器等等。逻辑的示例可以包括使用诸如源代码、编译代码、解释代码、可执行代码、静态代码、动态代码、面向对象的代码、视觉代码等的任何合适类型的代码实现的可执行计算机程序指令。实施例还可以至少部分地被实现为包含在非暂态计算机

可读介质中或其上的指令，其可以被一个或多个处理器读取和执行以使得能够执行本文中所描述的操作。

[0121] 系统存储器1206可以包括一个或多个较高速存储器单元形式的各种类型的计算机可读存储介质，诸如只读存储器(ROM)、随机存取存储器(RAM)、动态RAM(DRAM)、双数据速率DRAM(DDRAM)、同步DRAM(SDRAM)、静态RAM(SRAM)、可编程ROM(PROM)、可擦除可编程ROM(EPROM)、电可擦除可编程ROM(EEPROM)、闪存、聚合物存储器(诸如铁电聚合物存储器)、奥氏存储器、相变或铁电存储器、硅氧化氮氧化硅(SONOS)存储器、磁卡或光卡，设备阵列(诸如独立磁盘冗余阵列(RAID)驱动器)、固态存储器设备(例如，USB存储器、固态驱动器(SSD))以及适合于存储信息的任何其他类型的存储介质。在图12所示的图示的实施例中，系统存储器1206可以包括非易失性存储器1210和/或易失性存储器1212。基本输入/输出系统(BIOS)可以被存储在非易失性存储器1210中。

[0122] 计算机1202可以包括一个或多个较低速度存储器单元形式的各种类型的计算机可读存储介质，包括内部(或外部)硬盘驱动器(HDD)1214、从可移除磁盘1218读取或向其写入的磁性软盘驱动器(FDD)1216、以及从可移除光盘1222读取或向其写入的光盘驱动器1220(例如，CD-ROM或DVD)。HDD 1214、FDD 1216和光盘驱动器1220可以分别通过HDD接口1224、FDD接口1226和光驱接口1228连接到系统总线1208。用于外部驱动器实现方式的HDD接口1224可以包括通用串行总线(USB)和IEEE 1394接口技术中的至少一个或两个。

[0123] 驱动器和相关联的计算机可读介质提供数据、数据结构、计算机可执行指令等的易失性和/或非易失性存储。例如，若干个程序模块可以被存储在驱动器和存储器单元1210，1212中，包括操作系统1230、一个或多个应用程序1232、其他程序模块1234和程序数据1236。在一个实施例中，一个或多个应用程序1232、其他程序模块1234和程序数据1236可以包括例如系统100的各种应用和/或部件。

[0124] 用户可以通过一个或多个有线/无线输入设备(例如，键盘1238和诸如鼠标1240的指示设备)将命令和信息录入到计算机1202中。其他输入设备可以包括麦克风、红外(IR)遥控器、射频(RF)遥控器、游戏垫、触笔、读卡器、软件狗、指纹读取器、手套、图形输入板、游戏杆、键盘、视网膜读取器、触摸屏(例如，电容式、电阻式等)、轨迹球、触控板、传感器、触针等。这些和其他输入设备通常通过被耦合到系统总线1208的输入设备接口1242连接到处理单元1204，但是可以通过其他接口(诸如并行端口、IEEE 1394串行端口、游戏端口、USB端口、IR接口等等)连接。

[0125] 监视器1244或其他类型的显示设备也经由诸如视频适配器1246的接口连接到系统总线1208。监视器1244可以在计算机1202的内部或外部。除了监视器1244之外，计算机通常包括其他外围输出设备，诸如扬声器、打印机等等。

[0126] 计算机1202可以经由到一个或多个远程计算机(诸如远程计算机1248)的有线和/或无线通信使用逻辑连接在联网环境中操作。远程计算机1248可以是工作站、服务器计算机、路由器、个人计算机、便携式计算机、基于微处理器的娱乐设施、对等设备或其他公共网络节点，并且通常包括相对于计算机1202描述的许多或全部元件，尽管为了简洁起见，仅存储器/存储设备1250被图示。所描绘的逻辑连接包括到局域网(LAN)1252和/或更大网络(例如，广域网(WAN)1254)的有线/无线连接。这种LAN和WAN联网环境在办公室和公司中是司空见惯的，并且促进企业范围的计算机网络，诸如内联网，所有这些都可以连接到全球通信网

络，例如，互联网。

[0127] 当在LAN联网环境中使用时，计算机1202通过有线和/或无线通信网络接口或适配器1256连接到LAN 1252。适配器1256可以促进到LAN 1252的有线和/或无线通信，其还可以包括设置在其上的无线接入点，其用于与适配器1256的无线功能性进行通信。

[0128] 当在WAN联网环境中使用时，计算机1202可以包括调制解调器1258，或者被连接到WAN 1254上的通信服务器，或者具有用于通过WAN 1254(诸如通过互联网)建立通信的其他器件。可以是内置或外置的、以及是有线设备和/或无线设备的调制解调器1258经由输入设备接口1242连接到系统总线1208。在联网环境中，相对于计算机1202或其部分描绘的程序模块可以被存储在远程存储器/存储设备1250中。应当领会，所示的网络连接是示例性的，并且可以使用在计算机之间建立通信链路的其他器件。

[0129] 计算机1202可操作来使用IEEE802系列标准与有线和无线设备或实体进行通信，诸如在操作上被设置在无线通信(例如，IEEE802.11空中(over-the-air)调制技术)中的无线设备。这至少包括Wi-Fi(或无线保真)、WiMax和Bluetooth™无线技术等等。因此，通信可以如对于常规网络那样是预先定义的结构，或者仅仅是至少两个设备之间的自组织(ad hoc)通信。Wi-Fi网络使用被称为IEEE 802.11x(a、b、g、n等)的无线电技术来提供安全、可靠、快速的无线连接。Wi-Fi网络可以被用于将计算机彼此连接，连接到互联网，以及连接到有线网络(其使用IEEE 802.3相关介质和功能)。

[0130] 图13图示了适于实现如先前所描述的各种实施例的示例性通信体系架构1300的框图。通信体系架构1300包括各种常见通信元件，诸如发射器、接收器、收发器、无线电、网络接口、基带处理器、天线、放大器、滤波器、电源等。然而，实施例不限于由通信体系架构1300来实现。

[0131] 如图13所示，通信体系架构1300包括一个或多个客户端1302和服务器1304。客户端1302可以实现客户端设备910。服务器1304可以实现服务器设备950。客户端1302和服务器1304可操作地连接到一个或多个相应的客户数据存储装置1308和服务器数据存储装置1310，其可以被用来存储客户端1302和服务器1304相应的本地信息，诸如cookie和/或相关联的上下文信息。

[0132] 客户端1302和服务器1304可以使用通信框架1306在彼此之间传达信息。通信框架1306可以实现任何公知的通信技术和协议。通信框架1306可以被实现为分组交换网络(例如，诸如互联网的公共网络、诸如企业内联网的私有网络等)、电路交换网络(例如，公共交换电话网络)、或者分组交换网络和电路交换网络的组合(具有合适网关和转换器)。

[0133] 通信框架1306可以实现各种网络接口，这些网络接口被布置为接受、传达和连接到通信网络。网络接口可以被认为是输入输出接口的专用形式。网络接口可以采用连接协议，其包括但不限于直接连接、以太网(例如，粗、细、双绞线10/100/1000Base T等等)、令牌环、无线网络接口、蜂窝网络接口、IEEE 802.11ax网络接口、IEEE 802.16网络接口、IEEE 802.20网络接口等。进一步地，多个网络接口可以被用于与各种通信网络类型接合。例如，多个网络接口可以被用来允许用于通过广播、多播和单播网络的通信。如果处理要求指示了更大的速度和容量，则分布式网络控制器体系架构可以类似地被用来汇集、负载平衡、以及以其他方式增加客户端1302和服务器1304所需的通信带宽。通信网络可以是有线网络和/或无线网络中的任一个和其组合，其包括但不限于直接互连、安全定制连接、私有网络

(例如,企业内联网)、公共网络(例如,互联网)、个人区域网络(PAN)、本地局域网(LAN)、城域网(MAN)、作为互联网上的节点的操作任务(OMNI)、广域网(WAN)、无线网络、蜂窝网络和其他通信网络。

[0134] 本公开的各种实施例包括一种装置,其包括逻辑电路和逻辑,该逻辑在逻辑电路上操作以在计算环境中配置隔离存储器区域,用于保护与可在与隔离存储器区域不同的存储器区域中执行的代码进行的通信;使用与计算环境相对应的证据密钥来生成经签名数据,该经签名数据包括被保护的加密密钥和用于认证被保护的加密密钥的签名;以及将经签名数据传达到远程可信部件以访问存储在隔离存储器区域中的秘密代码。

[0135] 前一段落的装置可以包括逻辑,其进一步操作以使用特定地与控制计算环境的安全计算提供者相对应的私有证据密钥来生成签名。前一段落的装置可以包括逻辑,其进一步操作以生成用于在隔离存储器区域中运行的可信代码的加密密钥。前一段落的装置可以包括逻辑,其进一步操作以将密钥-值对存储在缓冲器中,该密钥-值对被传达到在隔离存储器区域中运行的可信代码或远程可信部件。前一段落的装置可以包括逻辑,其进一步被配置为在分布式文件系统上生成代码包的密码学摘要,并且使用该密码学摘要来验证被保护的加密密钥的签名。前一段落的装置可以包括逻辑,其进一步操作以处理指向在隔离存储器区域内部运行的可信代码或在隔离存储器区域的外部运行的不可信代码的通信原语。前一段落的装置可以包括逻辑,其进一步操作以处理通信原语,该通信原语操作以调用在隔离存储器区域的外部运行的不可信码或在隔离存储器区域的内部运行的可信代码上的函数。前一段落的装置可以包括逻辑,其进一步操作以使用公开证据密钥来验证签名并且解密被保护的加密密钥以提取加密密钥。先前段落中所描述的实施例还可以与本段落中的具体公开的备选方案中的一个或多个备选方案组合。

[0136] 本公开的各种实施例还包括产品,其包括至少一个计算机可读存储介质,该计算机可读存储介质包括指令,该指令当被执行时,使得系统通过在计算环境的隔离存储器区域内执行一组计算来生成计算数据,使用加密密钥来保护计算数据以生成被保护的计算数据,并且调用原语以将被保护的计算数据传达到在隔离存储器区域的外部运行的代码。

[0137] 前一段落的产品还可以包括指令,其当被执行时,使得系统处理使用与计算环境相关联的私钥生成的安全数据的签名,并且调用原语以向远程可信部件传达被保护的计算数据和签名。前一段落的产品可以进一步包括指令,其当被执行时,使得系统调用原语函数来生成加密密钥,以及调用另一原语函数来使用加密密钥生成签名。前一段落的产品可以进一步包括指令,其当被执行时,使得系统使用与在远程机器上运行的远程可信部件相对应的公钥来保护加密密钥。在上文所描述的产品的一个或多个实施例中,公钥与产品通信。前一段落的产品可以进一步包括指令,其当被执行时,使得系统使用加密密钥来解密被保护的用户密钥以提取用户密钥并且使用该用户密钥来解密隔离存储器区域中的秘密代码。先前段落中所描述的实施例还可以与该段落中的具体公开的备选方案中的一个或多个备选方案组合。

[0138] 本公开的各种实施例还包括一种方法,其包括以下步骤:在计算环境中生成隔离存储器区域以存储代码包,其中隔离存储器区域仅可由在隔离存储器区域中运行的代码访问;使用与计算环境相对应的私有证据密钥生成签名,使用代码包的签名和密码学摘要对代码包执行验证过程;以及将代码包的验证结果传达到远程可信部件。

[0139] 前一段落的方法还可以包括传达被保护的用户密钥以将代码包的秘密代码变换成应用代码的步骤。前一段落的方法还可以包括将加密的代码包加载到隔离存储器区域中的步骤。前一段落的方法可以进一步包括生成加密密钥以保护应用代码与在隔离存储器区域的外部运行的代码之间的通信的步骤。前一段落的方法可以进一步包括生成加密密钥以保护应用代码与在隔离存储器区域的外部运行的代码之间的通信的步骤。前一段落的方法还可以包括以下步骤:使用与计算环境相对应的公钥来处理包括经签名数据的消息,并且生成指示消息是否源自隔离存储器区域中的可信部件的验证结果。先前段落中所描述的实施例还可以与该段落中的具体公开的备选方案中的一个或多个备选方案组合。

[0140] 一些实施例可以使用表达式“一个实施例”或“一实施例”及其派生词来描述。这些术语意指结合实施例描述的特定特征、结构或特性被包括在至少一个实施例中。说明书中各处出现的短语“在一个实施例中”不一定都是指相同的实施例。进一步地,可以使用表达式“耦合”和“连接”及其派生词来描述一些实施例。这些术语不一定旨在作为彼此的同义词。例如,可以使用术语“连接”和/或“耦合”来描述一些实施例,以指示两个或更多个元件彼此直接物理或电接触。然而,术语“耦合”还可以意指两个或更多个元件彼此没有直接接触,而是仍然彼此协作或交互。

[0141] 需要强调的是,提供了本公开的摘要以允许读者快速确定技术公开的性质。提交该摘要,要理解的是,它不会被用于解释或限制权利要求的范围或含义。另外,在前面的具体实施方式中,可以看出,出于使本公开流畅简洁的目的,各种特征在单个实施例中被组合在一起。本公开的方法不被解释为反映所要求保护的实施例需要比每个权利要求中明确记载的特征更多的特征的意图。相反,如以下权利要求所反映的,发明主题在于少于单个公开实施例的所有特征。因此,以下权利要求由此被并入到具体实施方式中,其中每个权利要求本身作为单独的实施例。在所附权利要求中,术语“包括(including)”和“其中(in which)”分别被用作相应术语“包括(comprising)”和“其中(wherein)”的简明英语的等同物。此外,术语“第一”、“第二”、“第三”等仅被用作标签,并且不旨在对其对象施加数字要求。

[0142] 上文所描述的内容包括所公开的体系架构的示例。当然,不可能对部件和/或方法的每个可设想的组合进行描述,但是本领域的普通技术人员可以认识到许多其他组合和排列是可能的。因而,新颖的体系架构旨在涵盖落入所附权利要求的精神和范围内的所有这样的更改、修改和变型。

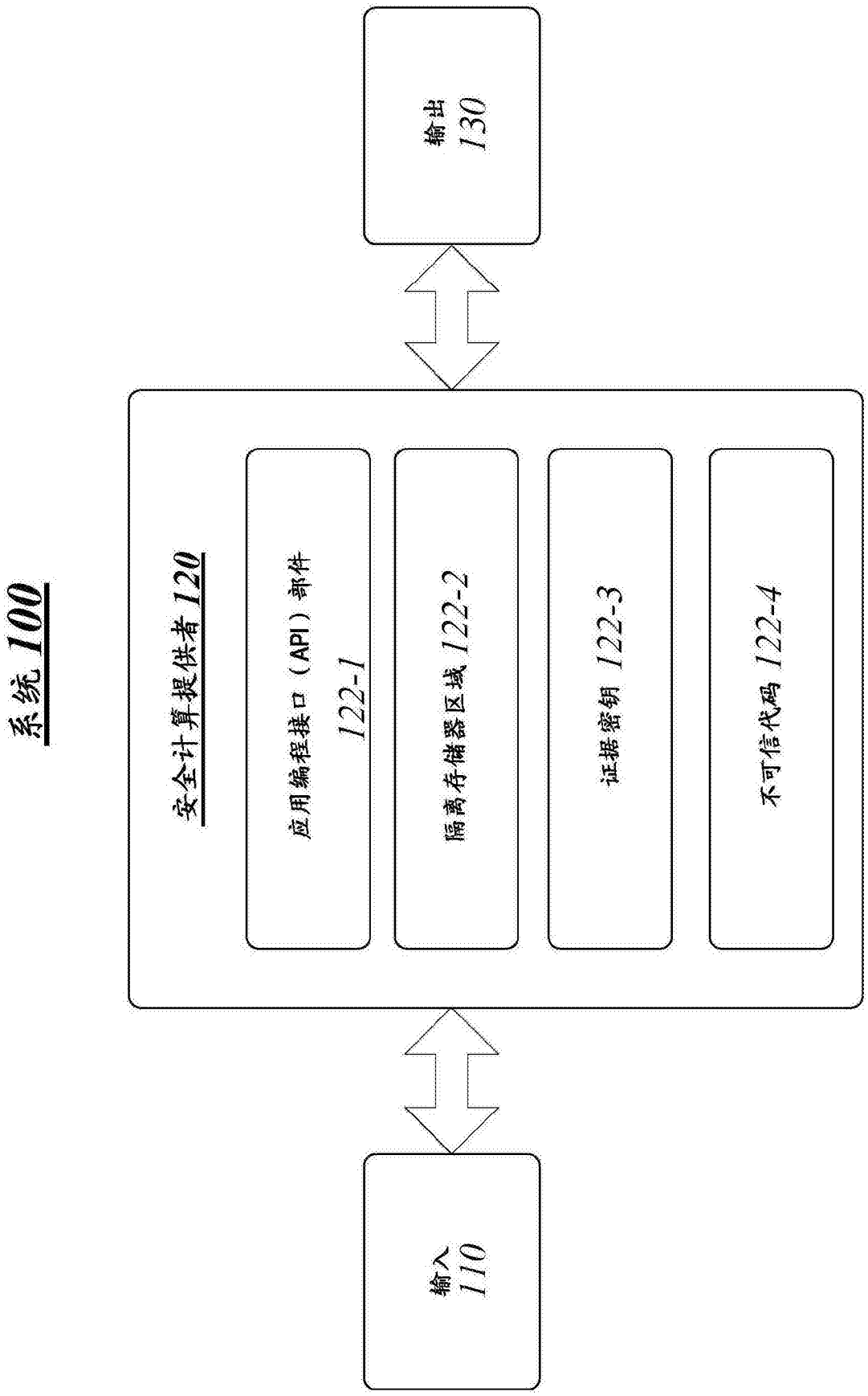

系统 100
安全计算提供者 120
应用编程接口（API）部件 122-1
隔离存储器区域 122-2
证据密钥 122-3
不可信代码 122-4
输入 110
输出 130

图1

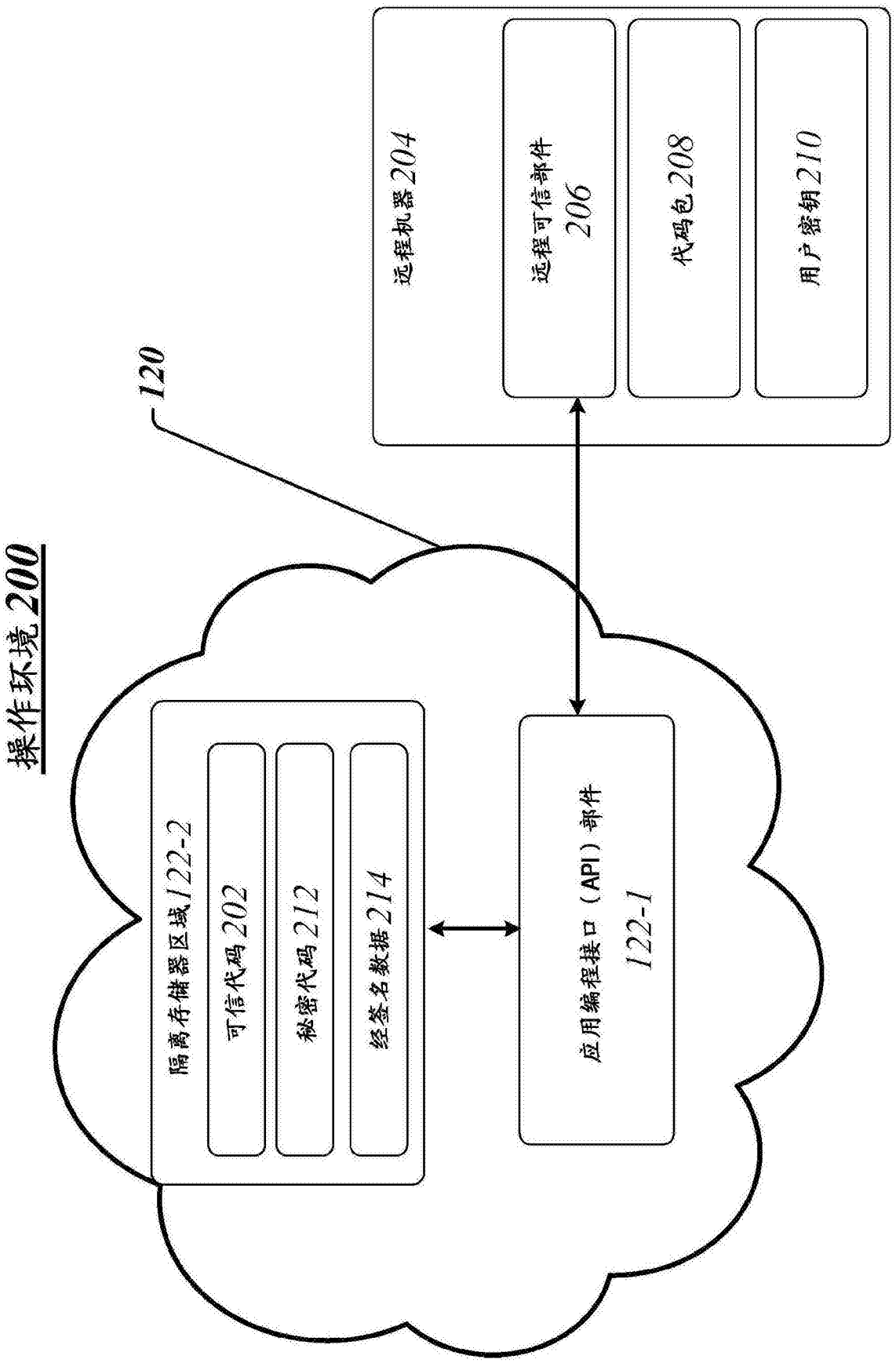

操作环境 200
120
隔离存储器区域 122-2
可信代码 202
秘密代码 212
经签名数据 214
应用编程接口（API）部件
122-1
远程机器 204
远程可信部件
206
代码包 208
用户密钥 210

图2

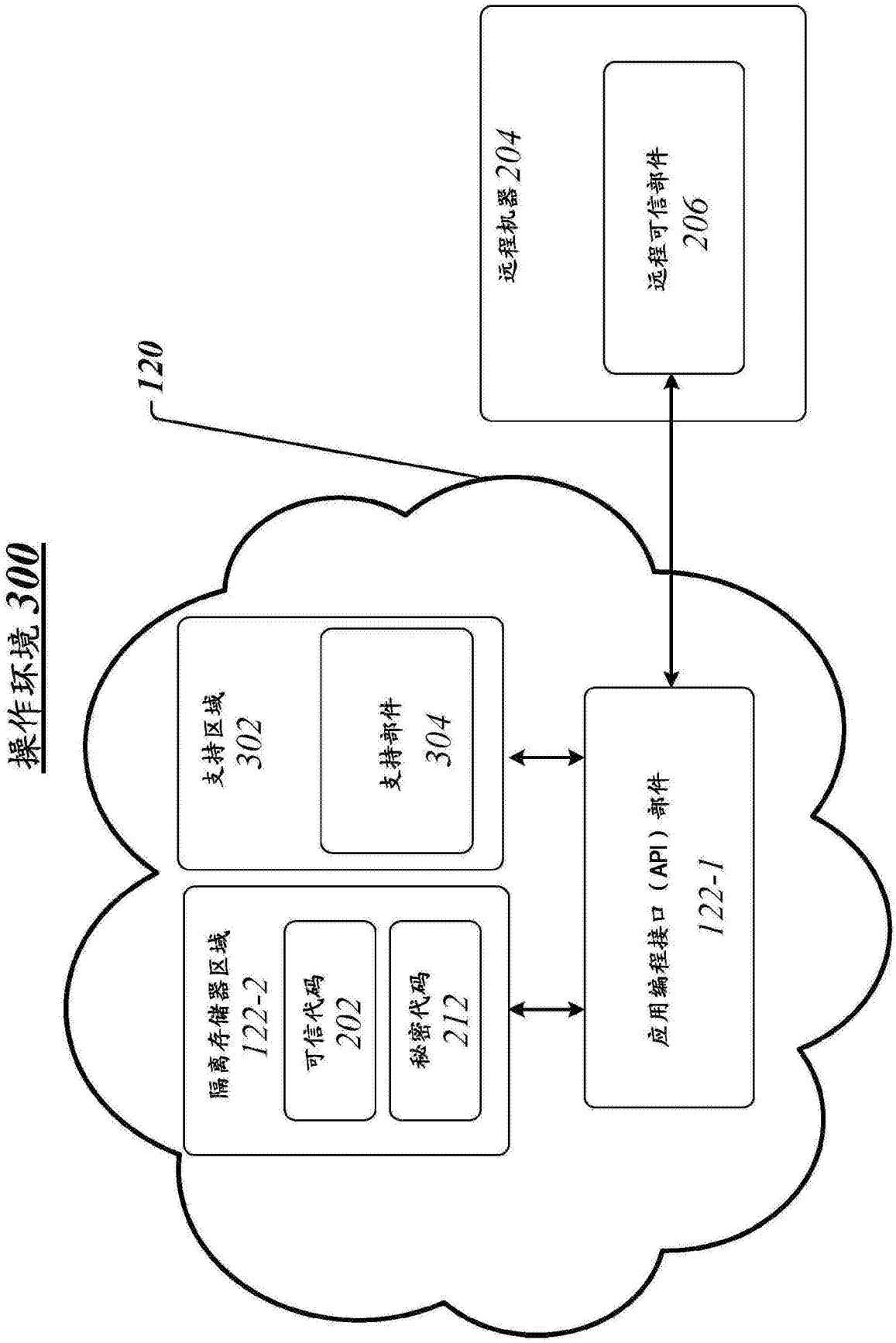

操作环境 300
120
远程机器 204
远程可信部件 206
支持区域 302
支持部件 304
隔离存储器区域 122-2
可信代码 202
秘密代码 212
应用编程接口（API）部件 122-1

图3

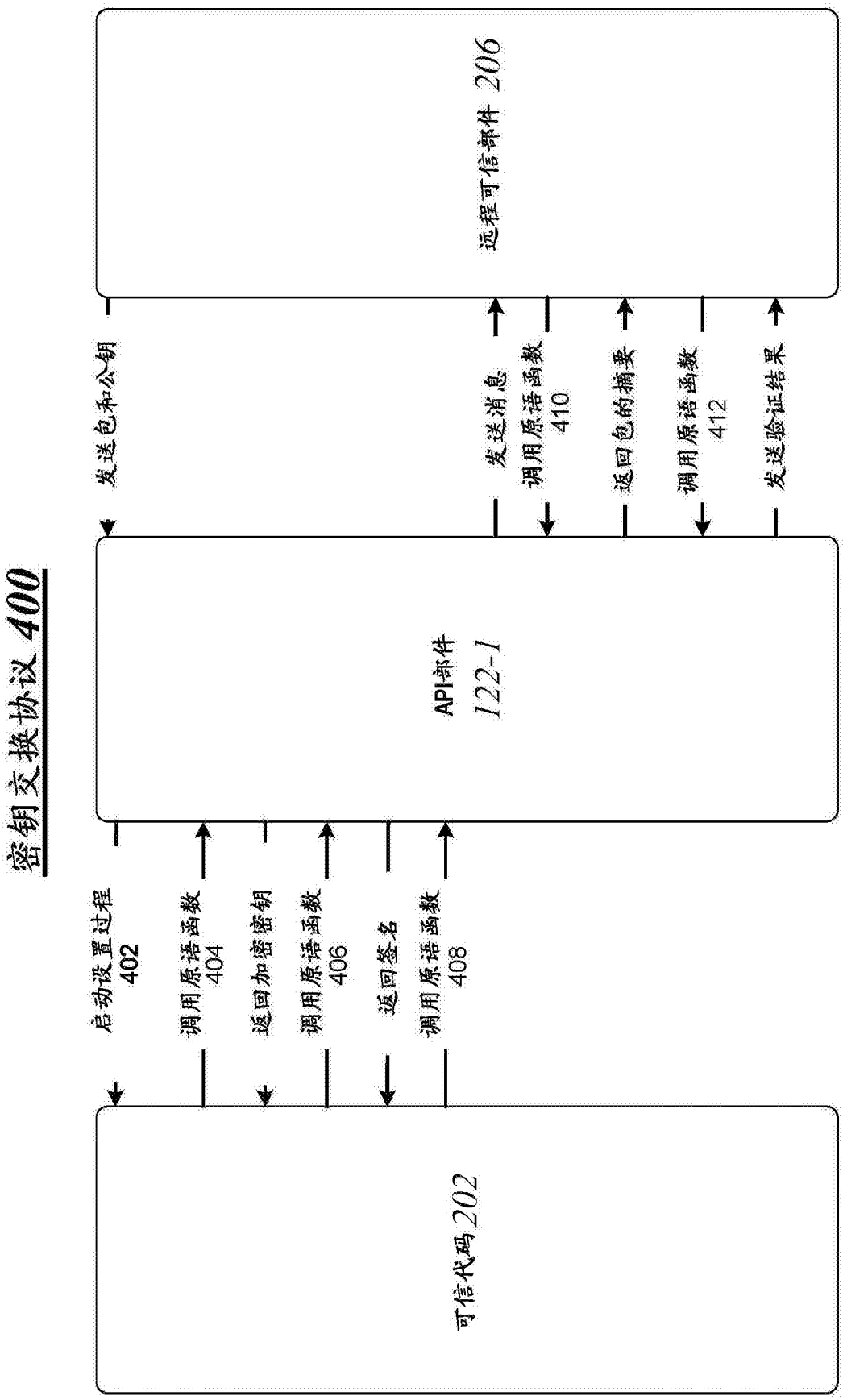

密钥交换协议 400
可信代码 202
API部件 122-1
远程可信部件 206
启动设置过程 402
调用原语函数 404
返回加密密钥
调用原语函数 406
返回签名
调用原语函数 408
发送包和公钥
发送消息
调用原语函数 410
返回包的摘要
调用原语函数 412
发送验证结果

图4

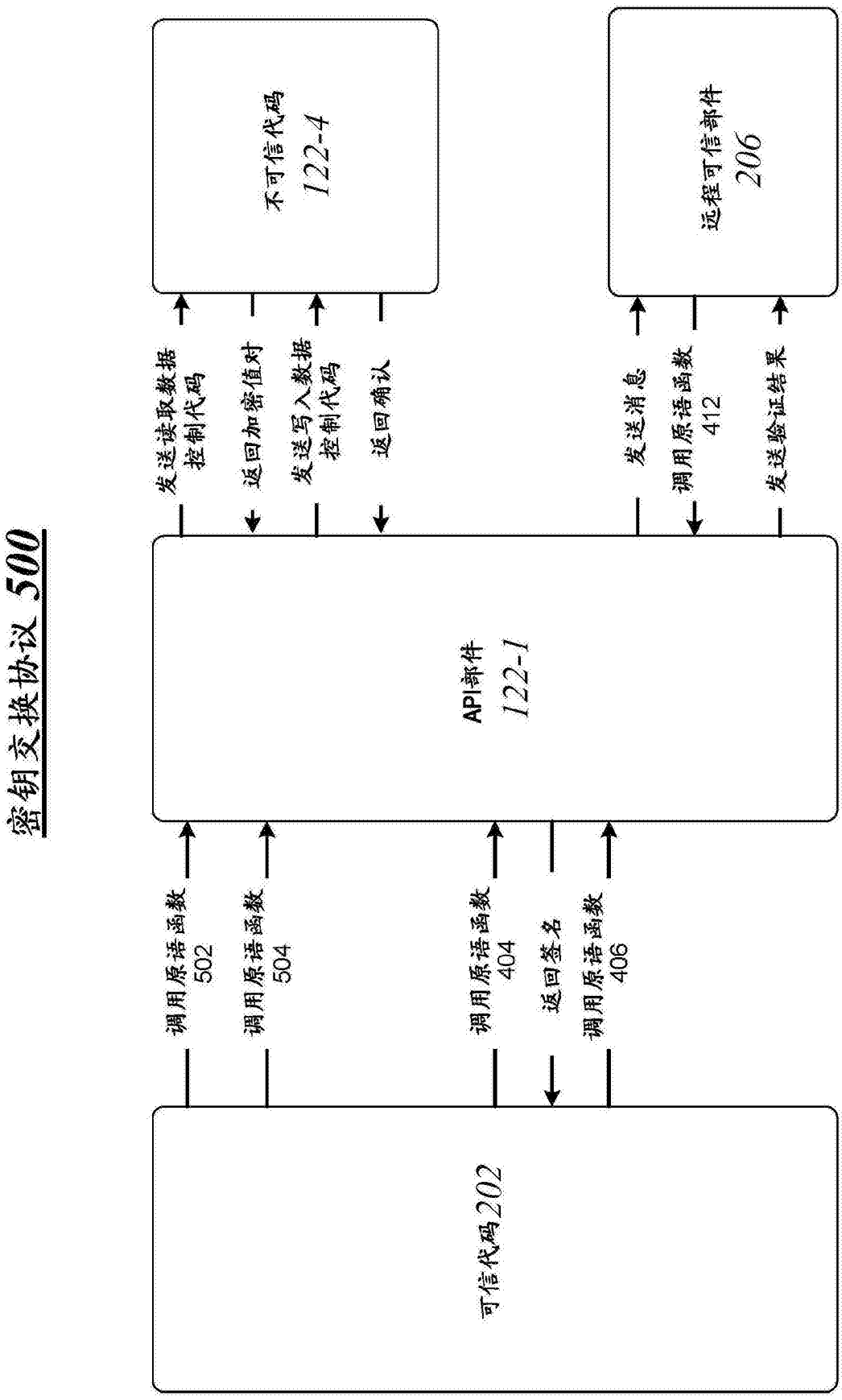

密钥交换协议 500
可信代码202
API部件
122-1
不可信代码
122-4
远程可信部件
206
调用原语函数
502
调用原语函数
504
调用原语函数
404
返回签名
调用原语函数
406
发送读取数据
控制代码
返回加密值对
发送写入数据
控制代码
返回确认
发送消息
调用原语函数
412
发送验证结果

图5

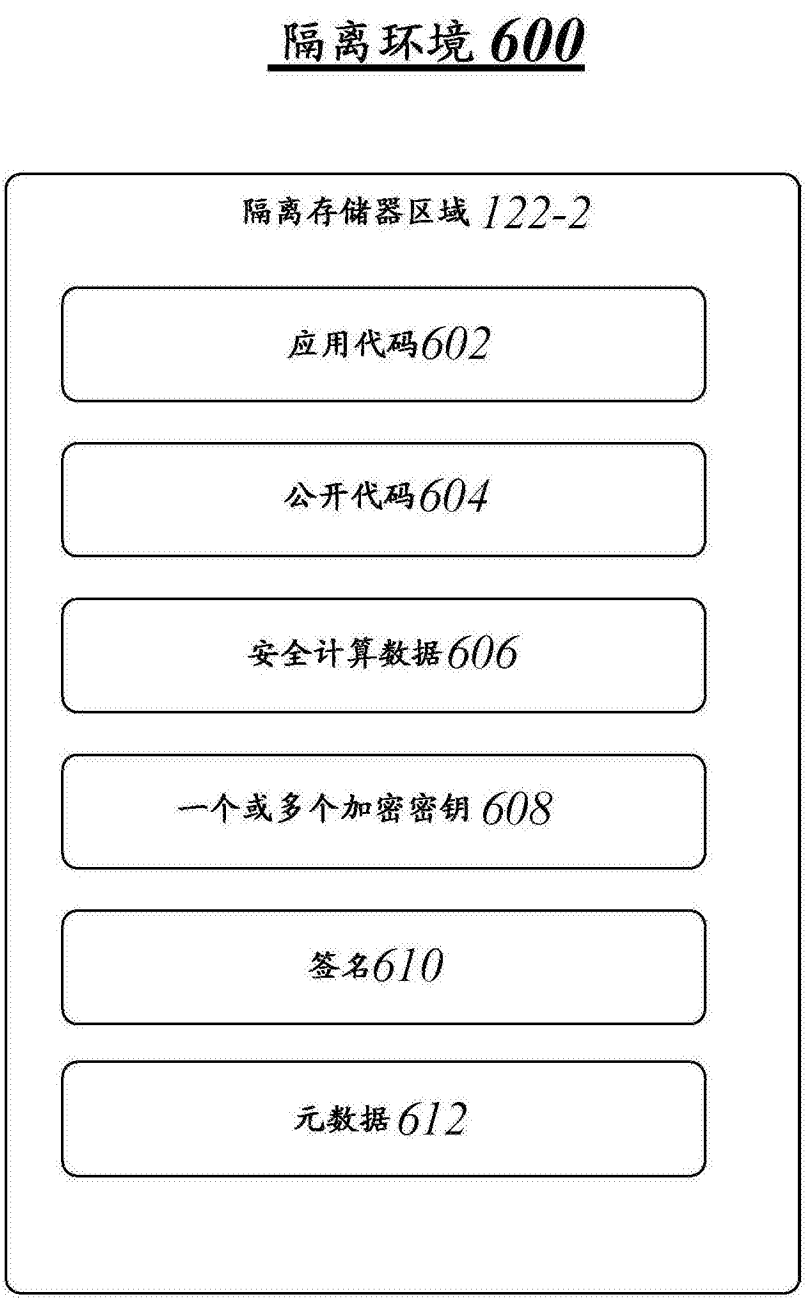

隔离环境600
隔离存储器区域122-2
应用代码602
公开代码604
安全计算数据606
一个或多个加密密钥608
签名610
元数据612

图6

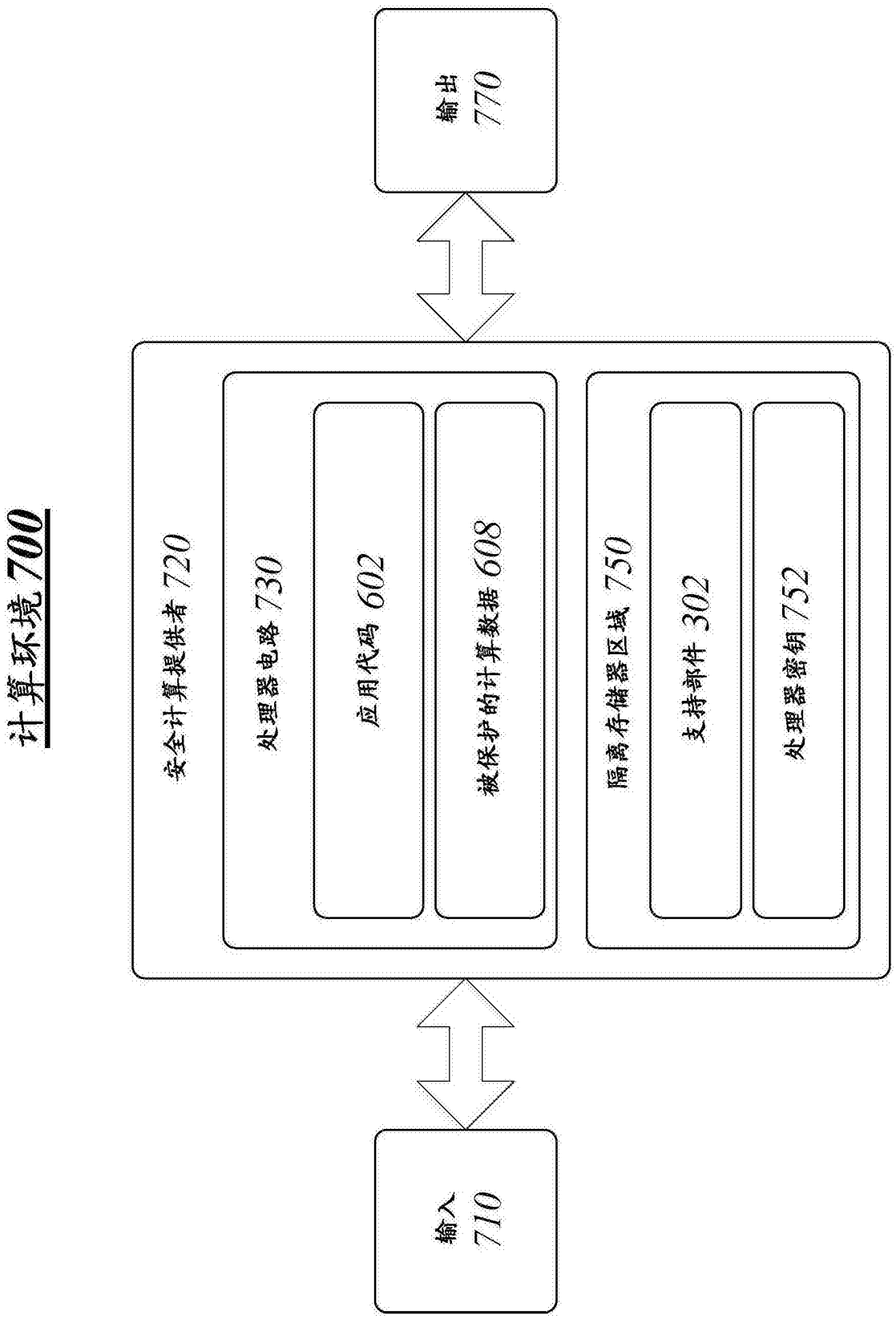

计算环境 700
安全计算提供者 720
处理器电路 730
应用代码 602
被保护的计算数据 608
隔离存储器区域 750
支持部件 302
处理器密钥 752
输入 710
输出 770

图7

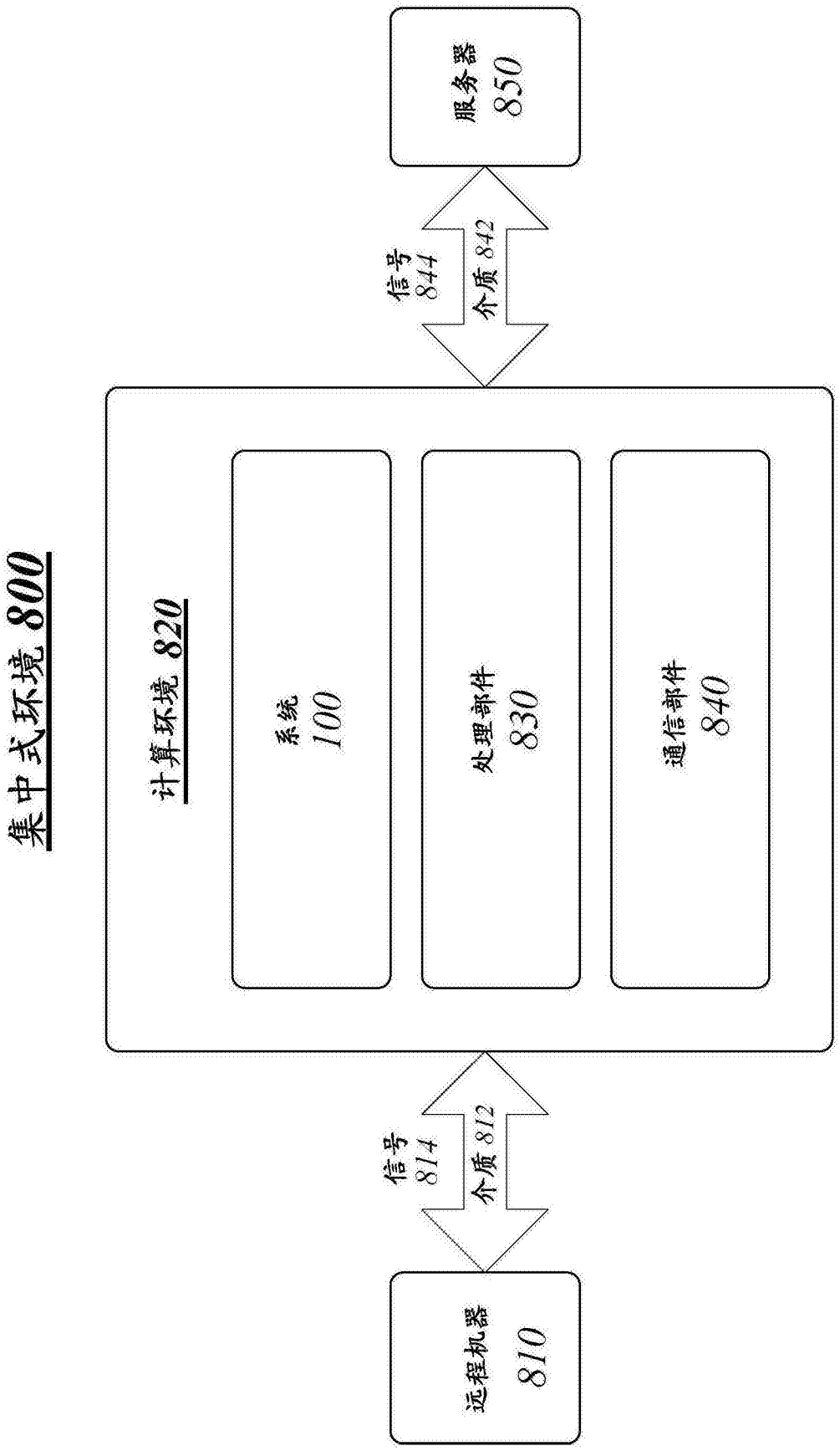

集中式环境 800
计算环境 820
系统 100
处理部件 830
通信部件 840
远程机器 810
信号 814
介质 812
信号 844
介质 842
服务器 850

图8

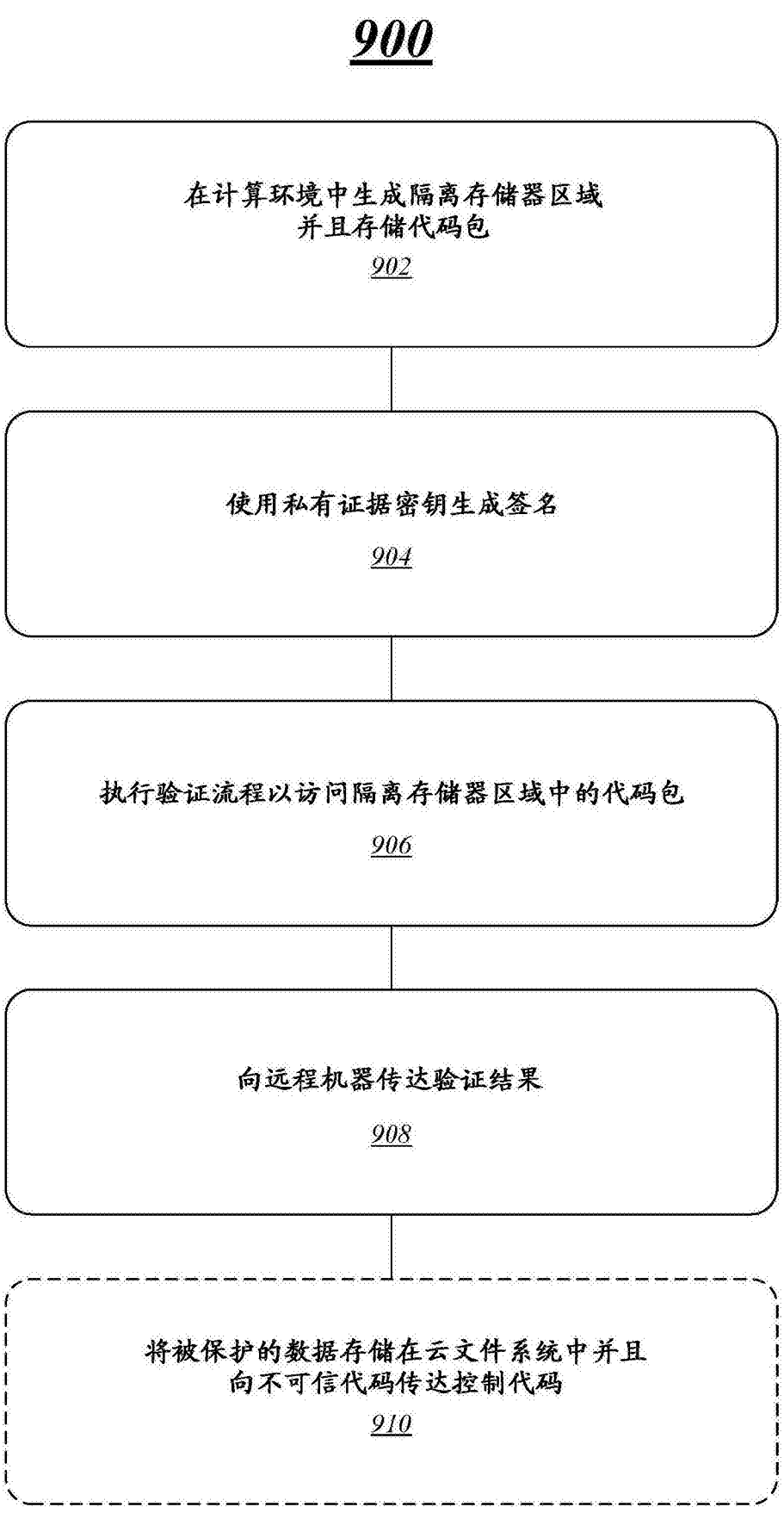

900
在计算环境中生成隔离存储器区域
并且存储代码包
902
使用私有证据密钥生成签名
904
执行验证流程以访问隔离存储器区域中的代码包
906
向远程机器传达验证结果
908
将被保护的数据存储在云文件系统中并且
向不可信代码传达控制代码
910

图9

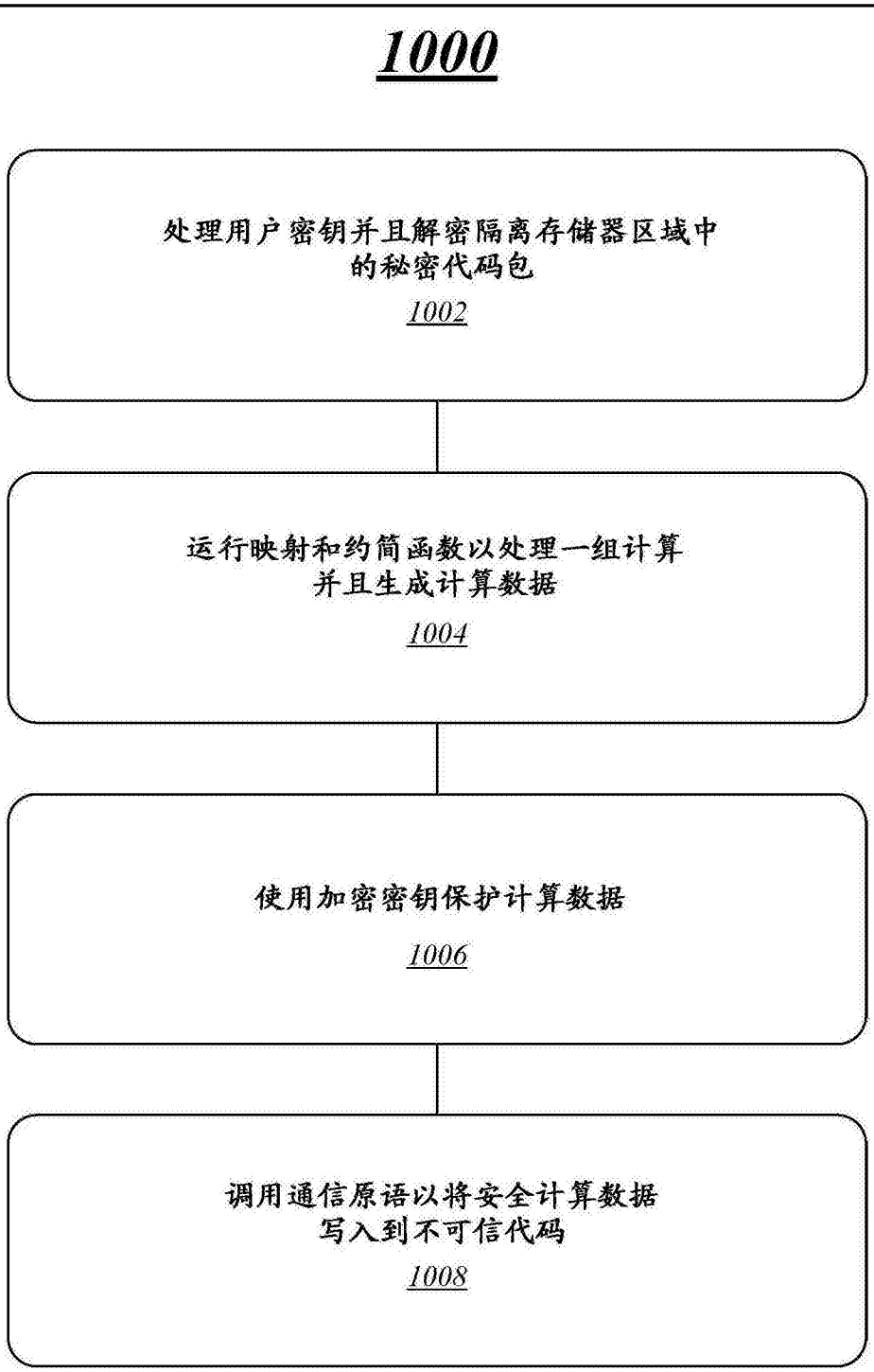

1000
处理用户密钥并且解密隔离存储器区域中的秘密代码包
1002
运行映射和约简函数以处理一组计算并且生成计算数据
1004
使用加密密钥保护计算数据
1006
调用通信原语以将安全计算数据写入到不可信代码
1008

图10

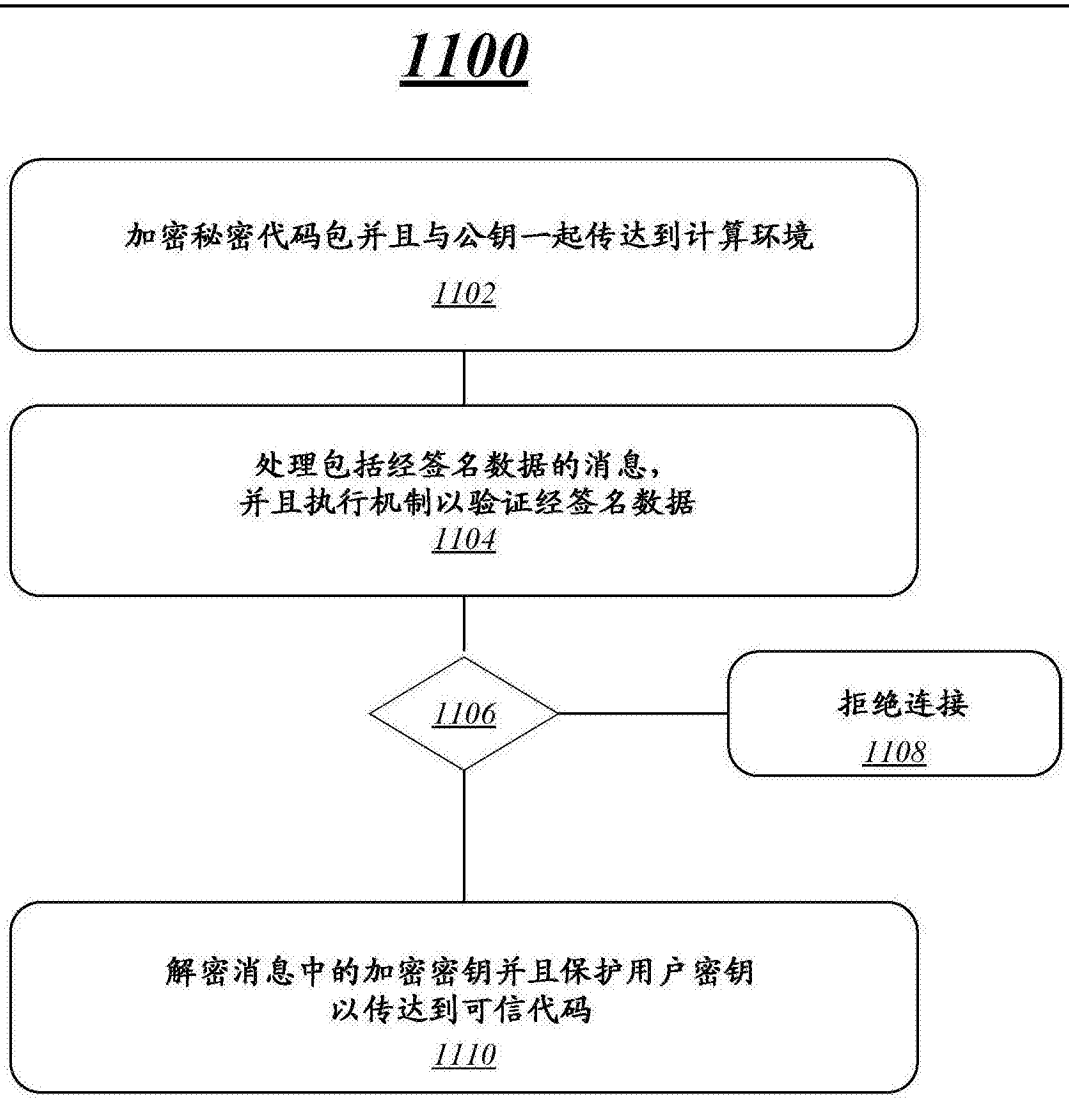

1100
加密秘密代码包并且与公钥一起传达到计算环境
1102
处理包括经签名数据的消息，
并且执行机制以验证经签名数据
1104
1106
拒绝连接
1108
解密消息中的加密密钥并且保护用户密钥
以传达到可信代码
1110

图11

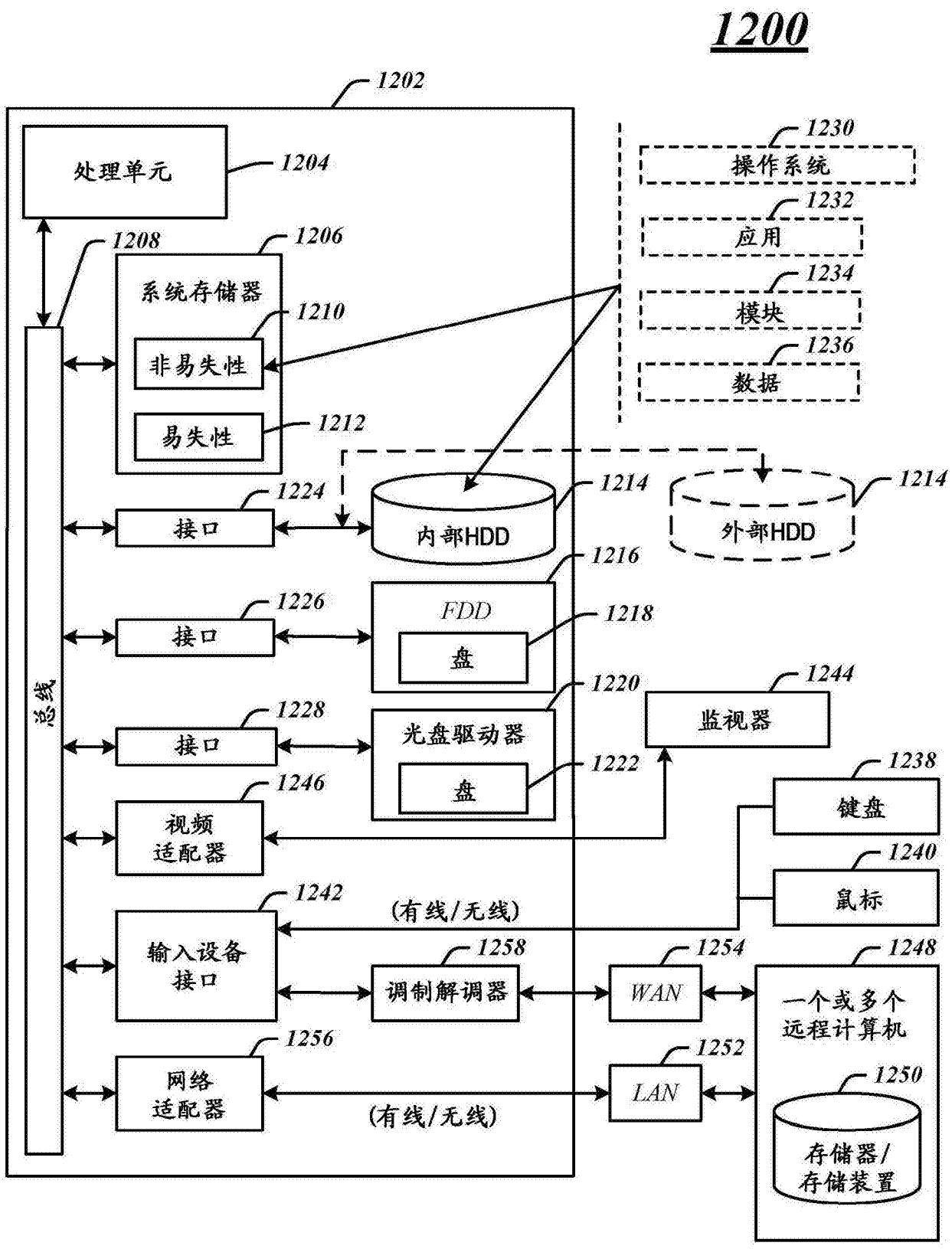

1200
1202
处理单元
1204
1208
1206
系统存储器
1210
非易失性
易失性
1212
1230
操作系统
1232
应用
1234
模块
1236
数据
1224
接口
内部HDD
1214
外部HDD
1214
1216
1226
接口
FDD
盘
1218
1220
1228
接口
光盘驱动器
盘
1222
1244
监视器
1238
键盘
1246
视频
适配器
1240
鼠标
1242
(有线/无线)
输入设备
接口
1258
调制解调器
1254
WAN
1248
一个或多个
远程计算机
1256
网络
适配器
1252
LAN
1250
存储器/
存储装置
(有线/无线)
总线

图12

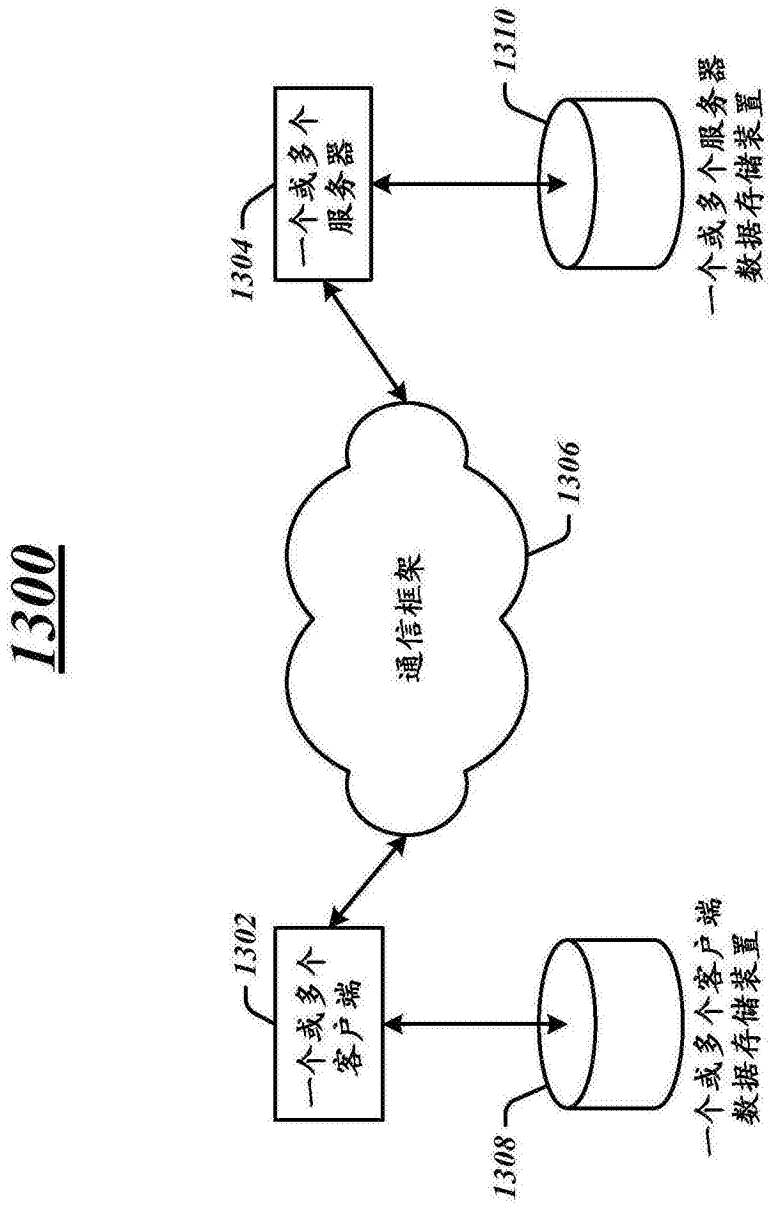

1300
一个或多个客户端
1302
一个或多个服务器
1304
通信框架
1306
一个或多个客户端数据存储装置
1308
一个或多个服务器数据存储装置
1310

图13