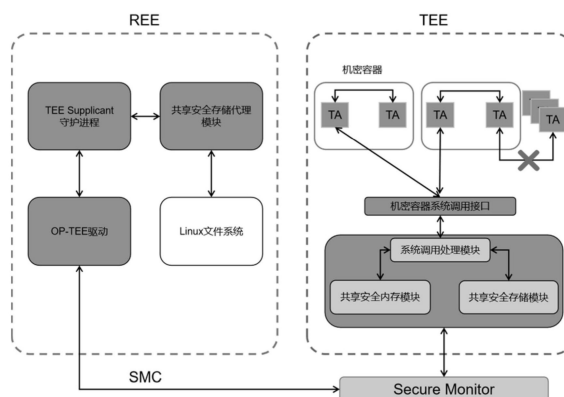




(43) 申请公布日 2025.06.27

G06F 9/54 (2006.01)

本发明提供一种基于硬件可信环境的轻量级机密容器构建方法,属于机密容器技术领域,包括共享安全内存模块、共享安全存储模块及系统调用分发模块,本发明在支持ARM TrustZone技术的端侧嵌入式设备利用已有的开源可信执行环境操作系统OP-TEE实现了轻量级机密容器;通过基于机密容器唯一标识设计,在TEE内部实现多个机密容器,同一个机密容器内的多可信应用具有细粒度的安全内存和安全存储条件共享,降低了机密容器内部可信应用之间的通信开销。机密容器的安全内存和安全存储与不可信世界保持强隔离状态;本发明降低了安全敏感应用的开发与部署难度,在保障硬件级数据机密性与完整性的同时,显著提升资源利用效率与系统可扩展性。



1. 一种基于硬件可信环境的轻量级机密容器构建方法,其特征在于:所述方法包括以下步骤:

S1:在物理内存中的OP-TEE OS之中划分出可信执行环境(TEE)与普通执行环境(REE),所述可信执行环境与普通执行环境之间通过SMC进行通信;

S2:所述可信执行环境(TEE)内部支持多个机密容器同时存在,每个所述机密容器内均支持多个可信应用(TA),每一个所述机密容器内的共享资源都受到机密容器所在的安全域的保护;

S3:设计并实现共享安全内存模块、共享安全存储模块以及系统调用分发模块,共享安全内存和共享安全存储以系统调用的形式向可信应用(TA)提供服务;

S4:所述可信应用(TA)权限设置为对共享物理内存的可读可写可执行权限,不同的所述机密容器,分别有不同的所述共享安全物理内存使用;

S5:同一个所述机密容器内的可信应用(TA)之间可以共享安全内存和共享安全存储,不同的所述机密容器之间的可信应用(TA)无法共享对方的安全内存和安全存储,任何尝试性的访问都会被相应的检查机制所拦截。

2. 根据权利要求1所述的一种基于硬件可信环境的轻量级机密容器构建方法,其特征在于:S3中,所述共享安全内存模块负责管理机密容器内不同可信应用(TA)之间的共享安全内存,在可信应用(TA)内存地址映射过程中,所述共享安全内存模块将会接管原生OP-TEE的地址映射过程,通过修改内存映射机制,实现同一个容器内部的可信应用(TA),其映射后的虚拟地址空间内部有一块虚拟内存来自相同的物理内存。

3. 根据权利要求1所述的一种基于硬件可信环境的轻量级机密容器构建方法,其特征在于:S3中,所述共享安全存储模块结合内部加解密模块负责管理机密容器内不同可信应用(TA)之间的共享安全存储,通过设计结合GP标准密钥链与机密容器标识的共享文件加密密钥(SFEK),并用于共享安全存储文件的加解密运算,接着设计共享安全存储模块,实现同一机密容器内部的存储共享。

4. 根据权利要求3所述的一种基于硬件可信环境的轻量级机密容器构建方法,其特征在于:所述共享安全存储模块在可信执行环境(TEE)和普通执行环境(REE)两侧均有相关模块,在所述可信执行环境(TEE)侧,设计共享安全存储模块,负责为可信应用(TA)提供系统调用,在所述REE侧,增加共享安全存储代理模块,负责处理TEE侧发送的系统调用,实现具体文件系统功能。

5. 根据权利要求1所述的一种基于硬件可信环境的轻量级机密容器构建方法,其特征在于:S3中,所述系统调用分发模块负责处理共享安全内存模块及共享安全存储模块的系统调用,并通过结合所设计的机密容器唯一标识码(SID),在OP-TEE的TEE侧划分出安全域,接着通过域控手段实现机密容器功能并保证其安全性。

6. 根据权利要求1-5任一项所述的一种基于硬件可信环境的轻量级机密容器构建方法,其特征在于:所述共享安全内存模块和共享安全存储模块分别与系统调用处理模块连接,所述系统调用处理模块连接有机密容器系统调用接口,所述机密容器系统调用接口分别与不同的可信应用(TA)连接。

7. 根据权利要求1所述的一种基于硬件可信环境的轻量级机密容器构建方法,其特征在于:所述普通执行环境(REE)内部设有OP-TEE驱动,所述OP-TEE驱动依次连接有TEE

Suppllicant守护进程、共享安全存储代理模块,所述共享安全存储代理模块最终连接Linux文件系统。

8.根据权利要求1所述的一种基于硬件可信环境的轻量级机密容器构建方法,其特征在于:所述机密容器除在OP-TEE系统中实现之外,还保留OP-TEE在TA强隔离方面的设计,可信应用(TA)可正常运行于机密容器外。

9.根据权利要求1所述的一种基于硬件可信环境的轻量级机密容器构建方法,其特征在于:所述机密容器内部的可信应用(TA)也可以使用原生OP-TEE所提供的服务,拥有独享的安全存储文件。

10.根据权利要求1所述的一种基于硬件可信环境的轻量级机密容器构建方法,其特征在于:本方法设计实现的模块是原生OP-TEE机制的补充与扩展,与原生OP-TEE之中的模块不存在冲突。

## 一种基于硬件可信环境的轻量级机密容器构建方法

### 技术领域

[0001] 本发明属于机密容器技术领域,特别涉及一种基于硬件可信环境的轻量级机密容器构建方法。

### 背景技术

[0002] 随着物联网与边缘计算技术的快速发展,嵌入式及端侧设备已广泛应用于智能制造、自动驾驶及智慧医疗等领域,这些设备的规模化部署在提升实时性与本地化处理能力的同时,也面临着严峻的安全挑战。由于端侧设备常暴露于开放物理环境,且受限于计算资源与能耗约束,难以部署复杂的安全机制,使其成为侧信道攻击、恶意代码注入等威胁的主要目标,进而加剧了关键数据泄露和系统完整性破坏的风险。

[0003] 可信执行环境(Trusted Execution Environment, TEE)是一种基于硬件扩展的异构安全架构,其通过构建软硬件协同的隔离执行环境,实现可信计算基(Trusted Computing Base, TCB)最小化与信任链(Chain of Trust, CoT)完整性保障。相较于依赖软件权限控制与密码学算法的传统方案,TEE依托硬件强制隔离机制,在安全飞地(Secure Enclave)内提供代码/数据机密性、运行时完整性及抗侧信道攻击能力,并基于动态度量机制确保从加载到执行的全生命周期可信验证。该技术已成为支撑机密计算等隐私敏感型计算范式的核心基础设施。

[0004] 为应对安全威胁,可信执行环境(TEE)技术通过硬件隔离机制为敏感计算提供了有效保护。目前,移动设备和嵌入式设备大多配备了支持TEE的硬件(如ARM TrustZone)。在物联网与边缘计算场景中,嵌入式及端侧设备因其物理暴露性、资源受限性及计算实时性要求,面临严峻的安全挑战。此类设备受制于低算力与高能耗约束,难以部署复杂安全协议,易受物理攻击、恶意代码注入及侧信道攻击威胁,导致关键数据泄露与系统完整性破坏风险显著加剧。尽管TEE技术理论上可为端侧提供轻量级安全防护,其实践部署仍存在三方面缺陷:首先,可信应用的开发因必须使用可信执行环境提供的少量接口,甚至专用开发工具,导致其开发、维护和迁移困难;其次,TEE的安全世界开发需定制化编程模型,导致安全功能与普通业务逻辑的协同复杂度高,开发效率低下;最后,传统TEE技术缺乏轻量级隔离单元支持,难以适配容器化部署需求,阻碍资源受限环境下安全性与灵活性的平衡。

[0005] 为简化安全应用的开发与部署难度,并实现系统级隔离,学术界与工业界开始探索在TEE环境中构建机密容器。机密容器技术通过融合硬件隔离与容器化封装,为应用栈及其运行时环境提供更强的安全保护。然而,现有研究多聚焦云端场景,基于Intel SGX等方案构建机密容器,但其依赖的富资源环境与端侧设备的异构性、低算力特性存在固有矛盾,导致端侧机密虚拟机难以直接迁移。因此,亟需设计一种轻量级机密容器架构,以满足端侧设备在安全性、资源利用和灵活性方面的需求。

### 发明内容

[0006] 与现有技术相比,本发明通过设计一种基于硬件可信环境的轻量级机密容器构建

方法,实现在端侧嵌入式设备或移动设备中实现机密容器,简化安全应用的开发与部署难度,并实现系统级隔离,显著提高端侧可信执行环境的易用性。

[0007] 本发明为解决上述问题采用如下的技术方案:

[0008] 一种基于硬件可信环境的轻量级机密容器构建方法,所述方法包括以下步骤:

[0009] S1:在物理内存中的OP-TEE OS之中划分出可信执行环境(TEE)与普通执行环境(REE),所述可信执行环境与普通执行环境之间通过SMC进行通信;

[0010] S2:所述可信执行环境内部支持多个机密容器同时存在,每个所述机密容器内均支持多个可信应用(TA),每一个所述机密容器内的共享资源都受到机密容器所在的安全域的保护;

[0011] S3:设计并实现共享安全内存模块、共享安全存储模块以及系统调用分发模块,共享安全内存和共享安全存储以系统调用的形式向可信应用(TA)提供服务;

[0012] S4:所述可信应用(TA)权限设置为对共享物理内存的可读可写可执行权限,不同的所述机密容器,分别有不同的所述共享安全物理内存使用;

[0013] S5:同一个所述机密容器内的可信应用(TA)之间可以共享安全内存和共享安全存储,不同的所述机密容器之间的可信应用(TA)无法共享对方的安全内存和安全存储,任何尝试性的访问都会被相应的检查机制所拦截。

[0014] 进一步的,S3中,所述共享安全内存模块负责管理机密容器内不同可信应用(TA)之间的共享安全内存,在可信应用(TA)内存地址映射过程中,所述共享安全内存模块将会接管原生OP-TEE的地址映射过程,通过修改内存映射机制,实现同一个容器内部的可信应用(TA),其映射后的虚拟地址空间内部有一块虚拟内存来自相同的物理内存。

[0015] 进一步的,S3中,所述共享安全存储模块结合内部加解密模块负责管理机密容器内不同可信应用(TA)之间的共享安全存储,通过设计结合GP标准密钥链与机密容器标识的共享文件加密密钥(SFEK),并用于共享安全存储文件的加解密运算,接着设计共享安全存储模块,实现同一机密容器内部的存储共享。

[0016] 进一步的,所述共享安全存储模块在可信执行环境(TEE)和普通执行环境(REE)两侧均有相关模块,在所述可信执行环境(TEE)侧,设计共享安全存储模块,负责为可信应用(TA)提供系统调用,在所述REE侧,增加共享安全存储代理模块,负责处理TEE侧发送的系统调用,实现具体文件系统功能。

[0017] 进一步的,S3中,所述系统调用分发模块负责处理共享安全内存模块及共享安全存储模块的系统调用,并通过结合所设计的机密容器唯一标识码(SID),在OP-TEE的TEE侧划分出安全域,接着通过域控手段实现机密容器功能并保证其安全性。

[0018] 进一步的,所述共享安全内存模块和共享安全存储模块分别与系统调用处理模块连接,所述系统调用处理模块连接有机密容器系统调用接口,所述机密容器系统调用接口分别与不同的可信应用(TA)连接。

[0019] 进一步的,所述普通执行环境(REE)内部设有OP-TEE驱动,所述OP-TEE驱动依次连接有TEE Supplicant守护进程、共享安全存储代理模块,所述共享安全存储代理模块最终连接Linux文件系统。

[0020] 进一步的,所述机密容器除在OP-TEE系统中实现之外,还保留OP-TEE在可信应用(TA)强隔离方面的设计,可信应用(TA)可正常运行于机密容器外。

[0021] 进一步的,所述机密容器内部的可信应用(TA)也可以使用原生OP-TEE所提供的服务,拥有独享的安全存储文件。

[0022] 进一步的,本方法设计实现的模块是原生OP-TEE机制的补充与扩展,与原生OP-TEE之中的模块不存在冲突。

[0023] 本发明的有益效果在于:

[0024] 1、本发明在端侧支持ARM TrustZone设备的TEE侧实现了多机密容器,机密容器内部可信应用(TA)支持内存与存储有条件共享、机密容器之间相互隔离的安全策略;通过融合硬件隔离与容器化封装实现应用栈全生命周期的系统级保护,简化安全应用的开发与部署难度,并优化隔离粒度、重构信任传递机制以及精简运行时开销,实现了安全防护强度与资源利用效率的协同优化,平衡了可信执行环境的易用性与安全性。

[0025] 2、原生OP-TEE中不同的可信应用(TA)可用内存地址空间相互隔离,隔离机制的是由页表机制所支持,未设计共享内存区域,TrustZone原生设计机制TA间通信需要世界切换,具有较大的性能开销,本方法为优化此问题,在实现的机密容器内部,为需要频繁通信的TA设计实现共享内存机制,共享安全内存机制通过共享安全内存模块实现;利用OP-TEE原生设计机制在物理内存介于内核和用于TA的RAM之间存在一块允许用户自行使用保留内存,本方法通过修改此保留内存的物理地址的划分,用于共享安全内存。

## 附图说明

[0026] 为了更清楚地说明本发明具体实施方式,下面将对具体实施方式描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图是本发明的一些实施方式,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据这些附图获得其他的附图。

[0027] 图1是本发明的机密容器整体结构图;

[0028] 图2是本发明机密容器内存布局示意图;

[0029] 图3是本发明共享安全内存映射流程图;

[0030] 图4是本发明共享安全存储密钥链生成流程。

## 具体实施方式

[0031] 为使本领域技术人员更好的理解本发明的技术方案,下面结合附图和具体实施例对本发明作详细说明。

[0032] 本发明公开了一种基于硬件可信环境的轻量级机密容器构建方法,此方法在支持ARM TrustZone技术的端侧嵌入式设备利用已有的开源可信执行环境操作系统OP-TEE (Open Source Project Trusted Execution Environment),实现一个轻量级机密容器,通过基于机密容器唯一标识的设计,在TrustZone内部划分出多个用于机密容器的安全域;进一步实现,在同一个机密容器内的多可信应用具有细粒度的安全内存和安全存储共享,且在机密容器的安全内存和安全存储中仍然保持可信执行环境的强隔离状态。

[0033] 本发明包括三个部分具体为:共享安全内存模块、共享安全存储模块及系统调用分发模块,本发明将机密容器与安全域结合,实现了机密容器内部共享、机密容器之间、机密容器内外相互隔离的安全策略,降低可信应用之间的通信开销,平衡了可信执行环境的

易用性与安全性。

[0034] 实施例1

[0035] 如图1所示,本发明的机密容器构件方法包括以下步骤:

[0036] S1:在物理内存中的OP-TEE OS之中划分出可信执行环境(TEE)与普通执行环境(REE),可信执行环境与普通执行环境之间通过SMC进行通信。

[0037] 普通执行环境(REE)内部设有OP-TEE驱动,OP-TEE驱动依次连接有TEE Suppllicant守护进程、共享安全存储代理模块,共享安全存储代理模块最终连接Linux文件系统。

[0038] S2:可信执行环境内部支持多个机密容器同时存在,每个机密容器内均支持多个可信应用(TA),每一个机密容器内的共享资源都受到机密容器所在的安全域的保护。

[0039] S3:设计并实现共享安全内存模块、共享安全存储模块以及系统调用分发模块,共享安全内存和共享安全存储以系统调用的形式向可信应用(TA)提供服务。

[0040] 共享安全内存模块和共享安全存储模块分别与系统调用处理模块连接,系统调用处理模块连接有机密容器系统调用接口,机密容器系统调用接口分别与不同的可信应用(TA)连接。

[0041] 共享安全内存模块负责管理机密容器内不同可信应用(TA)之间的共享安全内存,在可信应用(TA)内存地址映射过程中,共享安全内存模块将会接管原生OP-TEE的地址映射过程,通过修改内存映射机制,实现同一个容器内部的可信应用(TA),其映射后的虚拟地址空间内部有一块虚拟内存来自相同的物理内存。

[0042] 共享安全存储模块结合内部加解密模块负责管理机密容器内不同可信应用(TA)之间的共享安全存储,通过设计结合GP标准密钥链与机密容器标识的共享文件加密密钥(Shared File Encrypt Key,SFEK),并用于共享安全存储文件的加解密运算,接着设计共享安全存储模块,实现同一机密容器内部的存储共享。

[0043] 共享安全存储模块在可信执行环境(TEE)和普通执行环境(REE)两侧均有相关模块,在可信执行环境(TEE)侧,设计共享安全存储模块,负责为TA提供系统调用,在REE侧,增加共享安全存储代理模块,负责处理TEE侧发送的系统调用,实现具体文件系统功能。

[0044] 系统调用分发模块负责处理共享安全内存模块及共享安全存储模块的系统调用,并通过结合所设计的机密容器唯一标识码(Shared ID,SID),在OPTEE的TEE侧划分出安全域,接着通过域控手段实现机密容器功能并保证其安全性。

[0045] S4:可信应用(TA)权限设置为对共享物理内存的可读可写可执行权限,不同的机密容器,分别有不同的共享安全物理内存使用。

[0046] S5:同一个机密容器内的可信应用(TA)之间可以共享安全内存和共享安全存储,不同的机密容器之间的可信应用(TA)无法共享对方的安全内存和安全存储,任何尝试性的访问都会被相应的检查机制所拦截。

[0047] 需要说明的是,机密容器除在OP-TEE系统中实现之外,还保留OP-TEE在TA强隔离方面的设计,TA可正常运行于机密容器外;机密容器内部的TA也可以使用原生OP-TEE所提供的服务,拥有独享的安全存储文件;本方法设计实现的模块是原生OP-TEE机制的补充与扩展,与原生OP-TEE之中的模块不存在冲突。

[0048] 实施例2

[0049] 作为优选实施例,本实施例对于共享安全内存模块、共享安全存储模块及系统调用分发模块进行进一步限定,其余技术特征同实施例1。

[0050] 本方法实现在OP-TEE的TEE侧支持多个机密容同时存在,且在机密容器内部可以支持多个TA的安全内存和共享安全存储,每一个机密容器内的共享资源都受到机密容器所在的安全域的保护。如图1所示,在机密容器内存在两个TA,其之间可以共享安全内存和共享安全存储,而不同的机密容器之间的TA无法共享对方的安全内存和安全存储,任何尝试性的访问都会被相应的检查机制所拦截。

[0051] 除在OP-TEE系统中实现机密容器之外,本方法保留OP-TEE在TA强隔离方面的设计,如TA可正常运行于机密容器外。同样的,机密容器内部的TA也可以使用原生OP-TEE所提供的服务,例如机密容器内部TA仍然可以使用原有的安全存储机制,拥有独享的安全存储文件。本方法设计实现的模块是原生OP-TEE机制的补充与扩展,与原生OP-TEE之中的模块不存在冲突。

[0052] 原生OP-TEE中不同的可信应用(TA)可用内存地址空间相互隔离,隔离机制的是由页表机制所支持,未设计共享内存区域,TrustZone原生设计机制TA间通信需要世界切换,具有较大的性能开销,本方法为优化此问题,在实现的机密容器内部,为需要频繁通信的TA设计实现共享内存机制,共享安全内存机制通过共享安全内存模块实现;利用OP-TEE原生设计机制在物理内存介于内核和用于TA的RAM之间存在一块允许用户自行使用保留内存,本方法通过修改此保留内存的物理地址的划分,用于共享安全内存。

[0053] OP-TEE对安全内存的类型有着严格的区分,因此本方法通过将机密容器内的TA权限设置为对共享物理内存的可读可写可执行权限,在TA内存地址映射过程中,共享安全内存模块将会接管原生OP-TEE的地址映射过程,通过修改内存映射机制,实现同一个容器内部的TA,其映射后的虚拟地址空间内部有一块虚拟内存来自相同的物理内存。

[0054] 本方法静态地将共享物理内存的大小设置为0x1000,即4KB大小;此设计是兼容效率考虑,由于OP-TEE所使用的环境是可能为32位系统,在开启页表机制的情况下,一个页面的标准大小为4KB,在这种情况下共享物理内存的大小是一个页大小,此时建立内存映射可以直接进行页对齐,提高内存映射的效率。

[0055] 如图2所示,对于在机密容器内部的TA,如机密容器的两个TA,其虚拟地址空间都有一块来自相同物理内存的映射;对于TA1,可以通过访问共享安全内存的方式间接地访问到TA2的虚拟地址空间之中的共享安全内存,但是TA1无法访问TA2其他虚拟地址空间。

[0056] 不同机密容器,分别有不同的共享安全物理内存使用,机密容器和共享安全物理内存是一一对应的关系;本方法设计了基于Key-Value型的对应机制,将机密容器唯一标识码设置为Key,共享安全物理内存的起始地址设置为Value,以全局链表的形式存储在OP-TEE OS之中,因此,机密容器中的TA无法,跨容器访问到其他容器中的TA的地址空间。

[0057] TA具体的共享安全内存分配过程如图3所示:首先,机密容器内部第一个需要加载动态物理地址的TA执行打开会话,在打开会话的过程中通过传入参数判断本TA是否处在一个机密容器内部,如果不在一个机密容器内部则执行原来OP-TEE正常的内存映射流程;如果此TA在一个机密容器中,那么通过传入的机密容器唯一标识码判断此机密容器唯一标识码对应的物理内存是否存在(即此TA是否为此机密容器内第一个被加载进TEE的TA),如果存在,则直接使用已经分配好的物理内存,建立物理内存到虚拟内存的映射关系;如果不存

在,则先进行物理内存分配操作,在分配完成后建立物理内存的映射关系,最终完成加载。

[0058] 对于共享安全内存的释放过程设置,因为用于共享的安全物理内存的生命周期,伴随着整个机密容器的生命周期存在,在首次使用进行分配后,直到机密容器内部的TA最后一次使用完毕才会进行释放,因此共享内存模块在每一块用于共享的安全物理内存设置有一个引用计数,如果其被释放,则引用次数会减少一次,如果引用计数已经被减少到零,那么该容器最后一个TA退出前,调用释放函数那么OP-TEE OS将会释放此机密容器对应的这块物理内存。

[0059] 为实现机密容器内部TA的共享安全存储,本方法首先设计结合GP标准密钥链与机密容器标识的共享文件加密密钥(Shared File Encrypt Key,SFEK),并用于将用于共享安全存储文件的加解密运算;接着设计共享安全存储模块,实现同一机密容器内部的存储共享。

[0060] 共享文件加密密钥设计如下:

[0061] 全球平台(Global Platform,GP)标准组织规定,没有自身文件系统的TEE可以依赖正常世界的文件系统,但是要使用密码学算法保证安全,并且安全存储文件要和设备绑定,原生OP-TEE采取的方案是构建密钥链,将硬件标识和芯片ID等信息加入密钥链之中,最终生成用于安全存储的密钥,因此结合原有OP-TEE方案,本发明设计的共享文件加密密钥生成过程如图4所示。

[0062] 在密钥链中,与硬件相关的密钥是硬件安全存储密钥(Hardware Secure Storage Key,HSSK),是由硬件唯一密钥(Hardware Unique Key,HUK)和芯片唯一标识(Chip Identification,ChipID)与容器安全密钥(Container Secure Key,CSK)一起生成,其中HUK是厂商在设备出厂时写入协处理器或其他硬件之中,仅可在TEE侧读取,ChipID同样由芯片厂商在出厂时写入硬件;上述两个硬件标识由厂商保证唯一性并在硬件上不可更改;CSK是设计方案中针对安全存储所设计的机密容器相关信息的唯一标识,此密钥可以由开发者自行指定,是针对实际使用场景之中所设计的兼容性考虑,此设计可以保证各使用者和各厂商自行指定密钥链之中的一部分,防止暴力碰撞。HSSK计算过程如下:

[0063]  $HSSK = \text{HMAC}_{\text{SHA256}}(\text{HUK}, \text{ChipID} || \text{CSK})$

[0064]  $\text{HMAC}_{\text{SHA256}}$ 是指哈希运算消息认证码(Hash-based Message Authentication Code),利用SHA256算法生成一个不可逆的密文结果,HSSK将与设备和使用的机密容器方案版本唯一绑定。

[0065]  $\text{STSK} = \text{HMAC}_{\text{SHA256}}(\text{SID}, \text{HSSK})$

[0066] SID是用于共享安全存储的机密容器唯一标识码,结合HSSK生成共享TA存储密钥(Shared TASTorage Key,STSK)。

[0067]  $\text{SFEK} = \text{AES\_CBC}(\text{STSK})$

[0068] 为进一步保证安全性,对STSK进行AES CBC算法加密,得到最终的共享文件加密密钥(Shared File Encrypt Key,SFEK),SFEK最终将用于共享安全存储文件的加解密运算。

[0069] 共享安全存储模块的设计如下:

[0070] 由于OP-TEE的文件系统依赖于REE,根据TEE侧和REE侧的交互逻辑,共享安全存储模块在TEE和REE两侧均有相关模块,如图1所示:在TEE侧,设计共享安全存储模块,负责为TA提供系统调用;在REE侧,增加共享安全存储代理模块,负责处理TEE侧发送的系统调用,

实现具体文件系统功能。

[0071] 在共享安全存储模块的内部,设计加解密模块,负责提供密钥链生成功能、密钥分发和管理、加解密算法,加解密模块中实现AES算法,也可以对接OP-TEE之中提供的其他密码学算法接口,如SM2、SM3、SM4等算法。

[0072] 共享安全存储模块位于OP-TEE OS之中,实现基本的文件系统操作,具体有共享安全存储文件的打开、关闭、读取、写入等功能,上述功能以系统调用的方式供机密容器内的TA使用,OP-TEE OS在接到系统调用后,在共享安全存储模块中将会判断是否需要调用加密或解密操作,如果需要就进行对应处理,处理完成后或者不需要加解密操作将会直接发起RPC请求,通过OP-TEE驱动,TEE侧的文件操作最终会通过远程系统调用发送到REE侧。

[0073] 共享安全存储代理模块依赖于存在于REE侧的OP-TEE client,OP-TEE client中包含TEE Supplicant守护进程,守护进程会持续监听来自TEE侧的RPC请求,在接到共享安全存储的RPC请求后,TEE Supplicant守护进程将会将其分发到共享安全存储代理模块,共享安全存储代理模块会进一步根据共享安全存储之中的系统调用号调用Linux文件系统服务;相关数据在REE均以密文方式传输和存储,当Linux文件系统服务处理完成后,处理结果将会以函数返回值或函数参数的形式返回,在返回到共享安全存储模块之中,会再次判断是否需要加解密操作,最终返回给机密容器内部的TA;为进一步保证机密容器安全,共享安全存储文件机制没有在REE侧设置有权限控制、软连接等操作,权限直接归属于TEE Supplicant守护进程或者REE侧的root权限,其他低权限用户无法对共享安全存储文件进行文件系统操作。

[0074] 为处理共享安全内存模块及共享安全存储模块所出发的系统调用,发明设计实现系统调用分发模块,该模块通过接管系统调用请求,动态判断其是否属于机密容器专用(如共享内存/存储操作);为避免模块之间耦合和代码设计,对OP-TEE产生的安全性影响,模块仅设计必要系统调用,并通过接口参数精简、调用路径固化等优化,将TCB复杂度降至最低,并设计安全校验机制,严格限定系统调用仅服务于合法机密容器内的可信应用,在实现机密容器功能调用的同时,进一步保证容器安全性。本发明所设计新增的系统调用如下表所示:

功能	用户态系统调用	OP-TEE OS内实现
获取共享安全内存地址	TEE_GET_SHARED_VA	_utee_get_shared_va
释放共享安全物理内存	TEE_FREE_SHARED_PA	_utee_free_shared_pa
打开共享安全存储文件	TEE_TCON_OPEN	_utee_tcon_open
关闭共享安全存储文件	TEE_TCON_CLOSE	_utee_tcon_close
读取共享安全存储文件	TEE_TCON_READ	_utee_tcon_read
写入共享安全存储文件	TEE_TCON_WRITE	_utee_tcon_write

[0076] 本发明中所实现的系统调用均设有安全域检查机制。系统调用会判断调用它的TA是否处于一个机密容器内部。若TA产生错误访问如跨容器访问时,则系统调用将不会执行,直接返回错误值,本模块实现的系统调用只会向机密容器内部的TA提供服务。共享安全内存相关的系统调用的安全域检查由硬件MMU保证,其获取到的是物理内存存在虚拟地址空间之中的映射地址。TA独享的用户态地址无法被其他TA获取。与共享安全存储相关的系统调用则会检查传入的SID参数,如果其传入的SID值错误,则会直接返回错误值,不会进一步执

行。

[0077] 以上通过实施例对本发明进行了详细说明,但内容仅为本发明的较佳实施例,不能被认为用于限定本发明的实施范围。凡依本发明申请范围所作的均等变化与改进等,均应仍归属于本发明的专利涵盖范围之内。

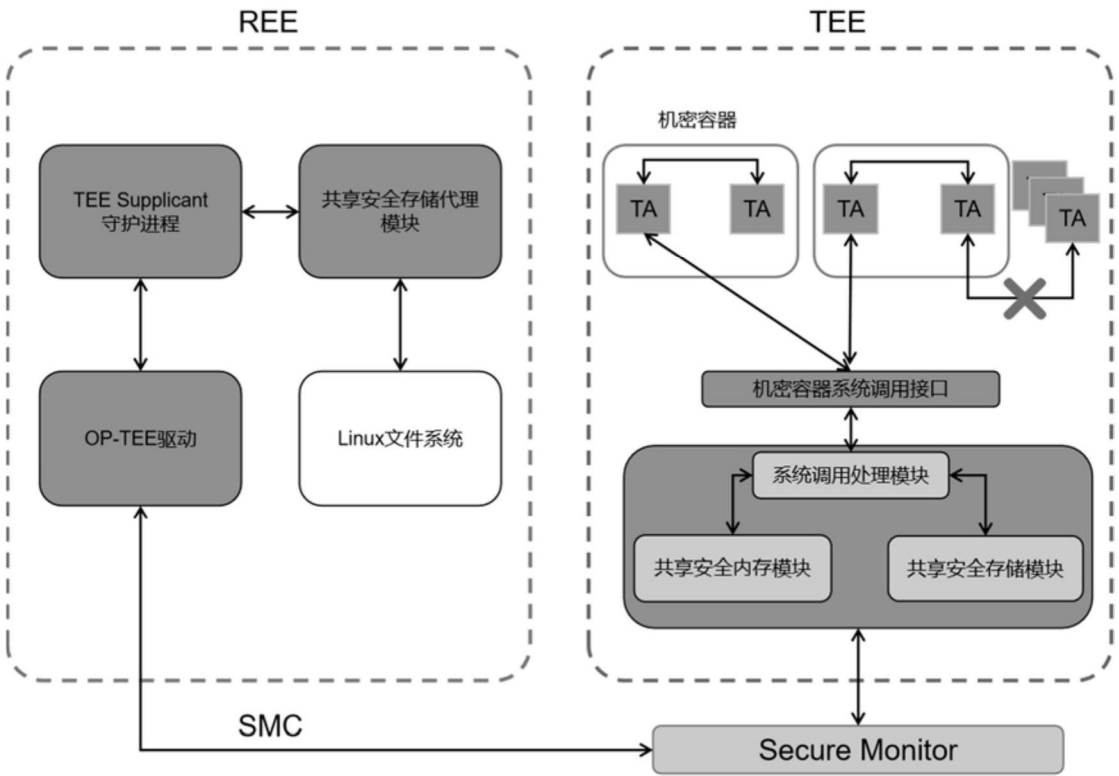


图1

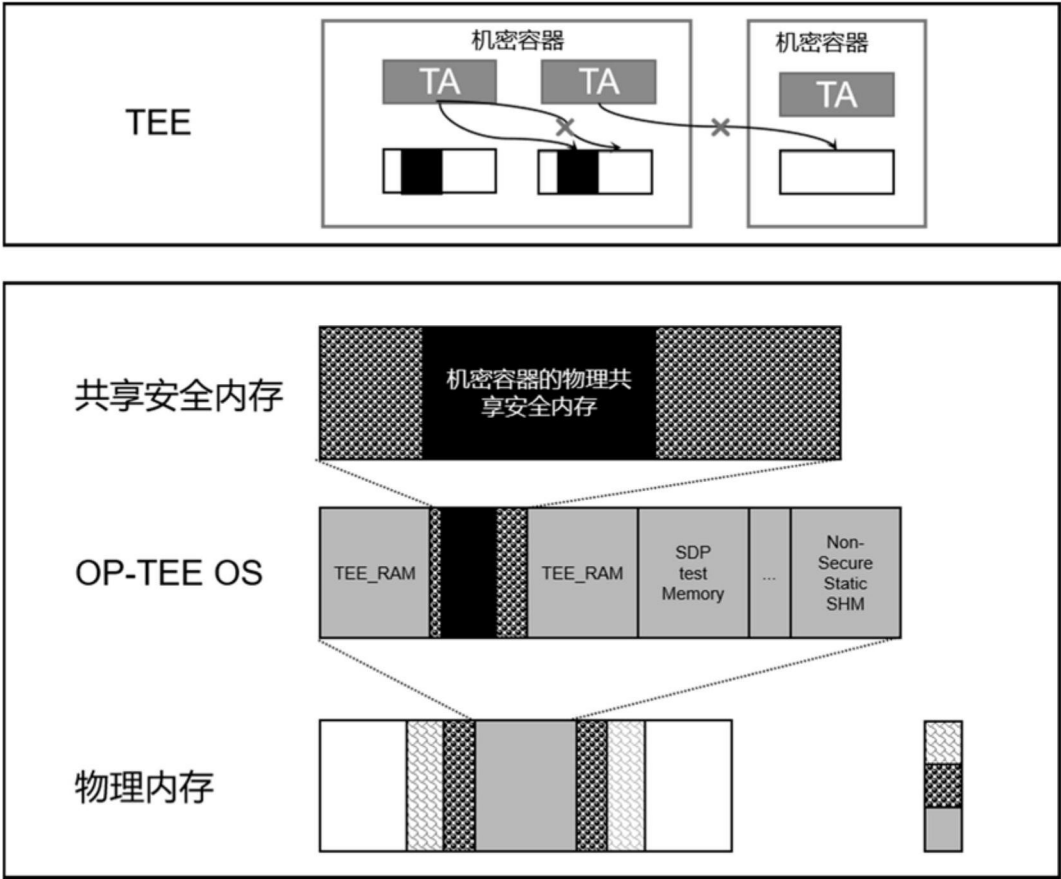


图2

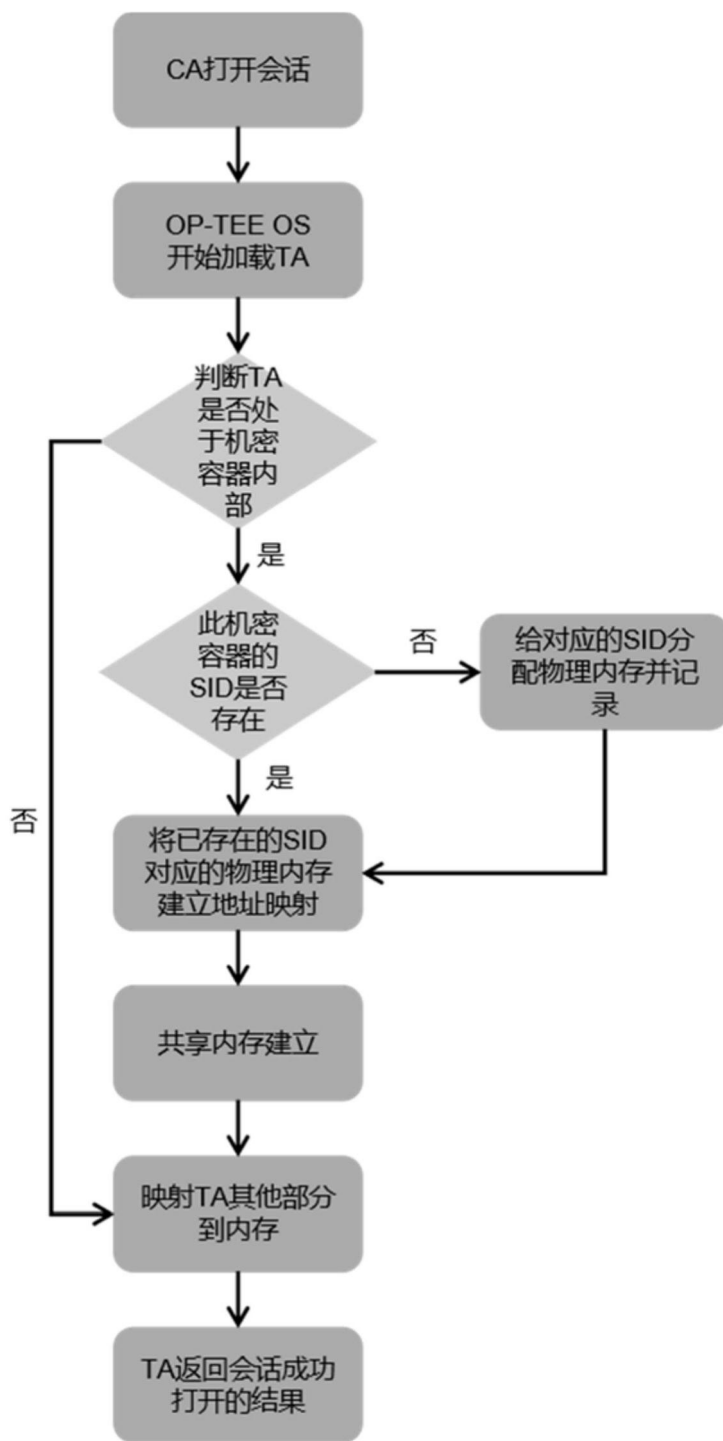


图3

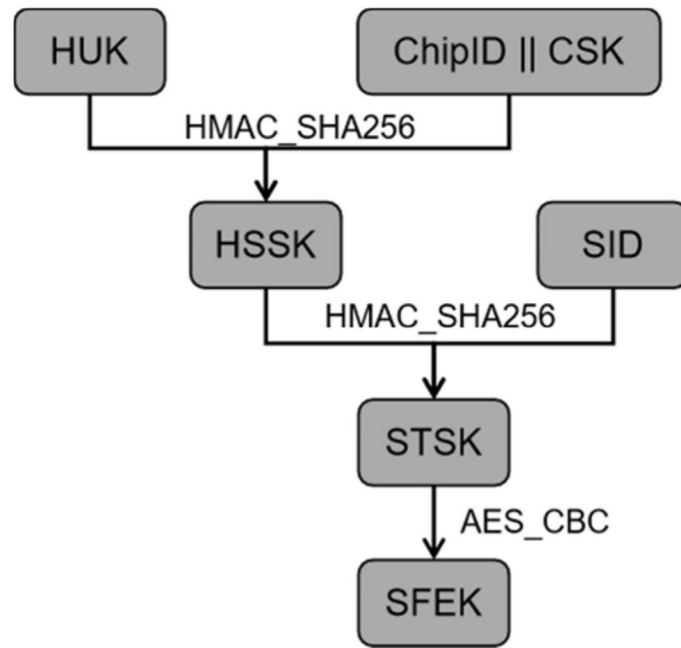


图4