

(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国际局

(43) 国际公布日
2024 年 3 月 14 日 (14.03.2024)



(10) 国际公布号
WO 2024/050869 A1

- (51) 国际专利分类号:
G06F 21/57 (2013.01) **H04L 9/28** (2006.01)
- (21) 国际申请号: PCT/CN2022/119945
- (22) 国际申请日: 2022 年 9 月 20 日 (20.09.2022)
- (25) 申请语言: 中文
- (26) 公布语言: 中文
- (30) 优先权:
202211087946.7 2022年9月7日 (07.09.2022) CN
- (71) 申请人: 南方科技大学 (SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY) [CN/CN]; 中国广东省深圳市南山区学苑大道 1088 号, Guangdong 518055 (CN)。
- (72) 发明人: 张殷乾 (ZHANG, Yinqian); 中国广东省深圳市南山区学苑大道 1088 号, Guangdong 518055

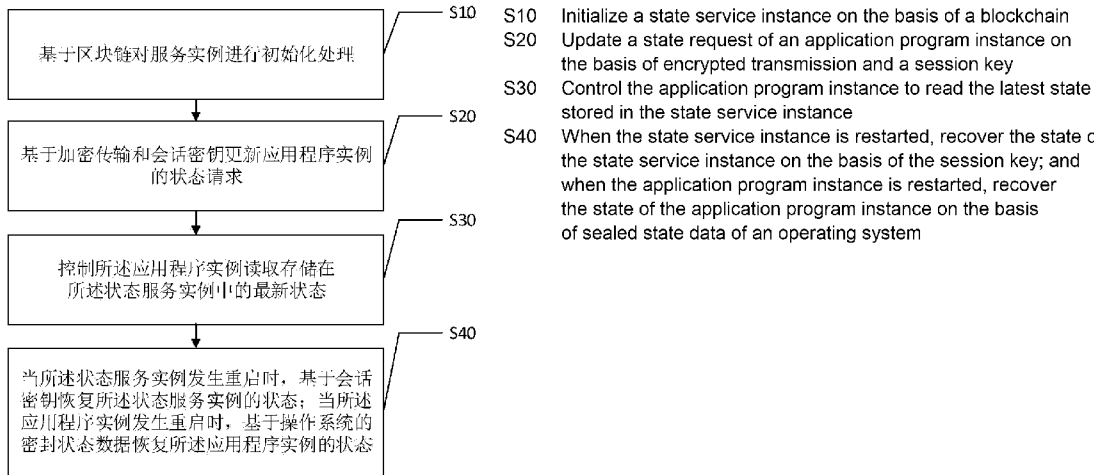
(CN)。彭维 (PENG, Wei); 中国广东省深圳市南山区学苑大道 1088 号, Guangdong 518055 (CN)。牛健宇 (NIU, Jianyu); 中国广东省深圳市南山区学苑大道 1088 号, Guangdong 518055 (CN)。

(74) 代理人: 深圳市君胜知识产权代理事务所 (普通合伙) (JOHNSON INTELLECTUAL PROPERTY AGENCY (SHENZHEN)); 中国广东省深圳市南山区粤海街道高新区社区高新南七道 20 号深圳国家工程实验室大楼 A503, Guangdong 518000 (CN)。

(81) 指定国 (除另有指明, 要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK,

(54) Title: DECENTRALIZED TRUST-BASED TEE STATE CONTINUITY PROTECTION METHOD UNDER PUBLIC CLOUD

(54) 发明名称: 基于去中心化信任的公有云 TEE 状态连续性保护方法



(57) Abstract: Disclosed in the present invention are a decentralized trust-based TEE state continuity protection method and system under a public cloud, an intelligent terminal, and a storage medium. The method comprises: initializing a state service instance on the basis of a blockchain; updating a state request of an application program instance on the basis of encrypted transmission and a session key; controlling the application program instance to read the latest state thereof stored in the state service instance; when the state service instance is restarted, recovering the state of the state service instance on the basis of the session key; and when the application program instance is restarted, recovering the state of the application program instance on the basis of sealed state data of an operating system. According to the present invention, the dependence on a centralized trusted entity is eliminated by using the blockchain, and depending only on the blockchain for initialization improves the state updating and reading speeds.



MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW。

(84) 指定国(除另有指明, 要求每一种可提供的地区保护): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。

本国际公布:

— 包括国际检索报告(条约第21条(3))。

(57) 摘要: 本发明公开了一种基于去中心化信任的公有云下TEE状态连续性保护方法、系统、智能终端及存储介质, 所述方法包括: 基于区块链对状态服务实例进行初始化处理; 基于加密传输和会话密钥更新应用程序实例的状态请求; 控制所述应用程序实例读取自己存储在所述状态服务实例中的最新状态; 当所述状态服务实例发生重启时, 基于会话密钥恢复所述状态服务实例的状态; 当所述应用程序实例发生重启时, 基于操作系统的密封状态数据恢复所述应用程序实例的状态。本发明中, 使用区块链消除了对中心化可信实体的依赖, 且只依赖区块链进行初始化, 提高了状态更新和读取的速度。

基于去中心化信任的公有云下 TEE 状态连续性保护方法

技术领域

本发明涉及计算机技术领域，尤其涉及的是一种基于去中心化信任的公有云下 TEE 状态连续性保护方法、系统、智能终端及计算机可读存储介质。

背景技术

随着科技不断发展，越来越多的厂商选择将业务部署到公有云上，在带来便利的同时，隐私保护和数据安全问题也日益严峻，由于操作系统的复杂性以及安全漏洞、安全机制容易被绕过等问题，把用户指纹、证书等机密数据和普通应用数据保存在公有云上无法保证其机密性、完整性和可用性。为此，业界提出了可信执行环境（Trusted Execution Environment，TEE）的设计，例如亚马逊云，谷歌云，阿里巴巴云平台使用 TEE 技术为用户提供机密计算服务，但 TEE 保护的应用程序仍然会受到回滚攻击或者分叉攻击，攻击者通过这些攻击可以将 TEE 的状态回滚到过时的版本或被分叉成多个版本，导致 TEE 的状态连续性被破坏。

现有技术中，通常利用硬件可信计数器提供一个安全容器，远程用户将代码和数据上载到安全容器中，在对容器执行计算时，安全容器用来保护数据的机密性和完整性，但硬件可信计数器的状态更新和读取的速度慢，不能满足云环境中对更新状态的要求。

因此，现有技术还有待改进和发展。

发明内容

本发明的主要目的在于提供一种基于去中心化信任的公有云下 TEE 状态连续性保护方法，旨在解决现有技术中硬件可信计数器的状态更新和读取的速度慢，不能满足云环境中对更新状态的要求的问题。

为了实现上述目的，本发明第一方面提供一种基于去中心化信任的公有云下 TEE 状态连续性保护方法，所述基于去中心化信任的公有云下 TEE 状态连续性保护方法包括：基于区块链对状态服务实例进行初始化处理；基于加密传输和会话密钥更新应用程序实例的状态请求；控制所述应用程序实例读取存储在所述状态服务实例中的最新状态；当所述状态服务实例发生重启时，基于会话密钥恢复所述状态服务实例的状态；当所述应用程序实例发生重启时，基于操作系统的密封状态数据恢复所述应用程序实例的状态。

可选的，所述基于去中心化信任的公有云下 TEE 状态连续性保护方法还包括：通过所述区块链的第一接口将所述状态服务实例的身份标识号码和公钥写到所述区块链上，并记录初始化信息；通过所述区块链的第二接口根据所述状态服务实例的身份标识号码和公钥在所述区块链上查询初始化信息；通过所述区块链的第三接口在所述应用程序实例更新状态摘要时，向所述状态服务实例请求更新是否合法；通过所述区块链的第四接口获取所述应用程序实例的状态摘要。

可选的，所述基于区块链对所述状态服务实例进行初始化处理，具体包括：将任意一个所述状态服务实例设置为领导者，通过远程证明相互验证所述领导者和其他状态服务实例的身份，并建立所述领导者与其他状态服务实例的安全通信信道；

控制所述领导者和其他状态服务实例生成非对称密钥对，并将所述状态服务

实例的公钥发送给所述领导者，所述领导者将接收到的公钥列表分发给其他状态服务实例；

控制每个所述状态服务实例向所述区块链发送读请求，所述读请求包括所述状态服务实例的唯一标识符；

控制每个所述状态服务实例向所述区块链发送写请求，当收到来自所述区块链的确认字符时，控制每个所述状态服务实例创建本地内存，所述本地内存包括其他状态服务实例状态条目的状态服务实例状态表和空的应用程序实例状态表；

控制本地状态服务实例将所述公钥列表放置到所述本地状态服务实例的本地内存以生成状态服务实例配置表，并控制所述本地状态服务实例通过运行经过验证的密钥协议与其他状态服务实例建立成对会话密钥；

密封并存储所述本地状态服务实例的密钥对、所述状态服务实例配置表和所述应用程序实例状态表。

可选的，所述基于加密传输和会话密钥更新所述应用程序实例的状态请求，具体包括：

当所述应用程序实例接收到用户请求时，密封并存储所述应用程序实例的状态快照到磁盘上，并更新所述应用程序实例的状态摘要，向所述本地状态服务实例发送状态更新请求；

当所述本地状态服务实例接收到状态更新请求时，存储所述状态更新请求到请求队列；

当处理所述状态更新请求时，控制所述本地状态服务实例向所述本地状态服务实例和其他状态服务实例发送所述状态更新请求；

当所述本地状态服务实例和其他状态服务实例接收到所述状态更新请求时，更新所述本地状态服务实例和其他状态服务实例状态内存中所述本地状态服务实例的状态摘要并发送回复消息；

当所述本地状态服务实例收到超过 $f+1$ 个所述回复消息时，所述本地状态服务实例向其他状态服务实例发送决定消息，其中， f 为共识节点容错数；

当其他状态服务实例收到所述决定消息后，检查其他状态服务实例是否已回复所述准备消息，若是，则控制所述状态服务实例发回包含状态摘要的确认字符；

当所述本地状态服务实例收到超过 $f+1$ 个所述确认字符时，控制所述本地状态服务实例更新所述应用程序实例的状态摘要并向所述应用程序实例回复包含状态摘要的确认字符；

当所述应用程序实例收到所述确认字符时，将其状态更新为最新状态并发布相关输出。

可选的，所述控制所述应用程序实例读取所述状态服务实例的最新状态，具体包括：

控制所述应用程序实例调用函数向所述本地状态服务实例发送状态读取请求；

当所述本地状态服务实例收到所述状态读取请求时，检索全部所述状态服务实例的最新状态；

检查每个所述状态服务实例的状态服务实例状态表，并将存储的状态摘要发回给所述本地状态服务实例；

当收到 $f+1$ 个所述状态摘要时，选择索引最大的状态摘要作为所述本地状态服务实例的状态摘要。

可选的，所述当所述状态服务实例发生重启时，基于会话密钥恢复所述状态

服务实例的状态，具体包括：

控制重启后的所述状态服务实例与其他状态服务实例建立新的会话密钥；

控制重启后的所述状态服务实例获取所述状态服务实例和其他状态服务实例的状态摘要；

选择索引最大的状态摘要作为重启后的所述状态服务实例的状态摘要；

控制所述状态服务实例从操作系统获取密封的状态摘要，将所述状态摘要与从其他状态服务实例获取的状态摘要进行匹配，若匹配，则控制所述状态服务实例恢复所述状态服务实例的状态摘要。

可选的，所述当所述应用程序实例发生重启时，基于操作系统的密封状态数据恢复所述应用程序实例的状态，具体包括：

控制重启后的所述应用程序实例向所述本地状态服务实例请求最新状态摘要；

控制所述应用程序实例获取所述操作系统的密封状态数据，将所述密封状态数据与所述最新状态摘要进行匹配，若匹配成功，则更新重启后所述应用程序实例的状态摘要并发布输出。

本发明第二方面提供一种基于去中心化信任的公有云下 TEE 状态连续性保护系统，其特征在于，所述基于去中心化信任的公有云下 TEE 状态连续性保护系统包括：

初始化模块，用于基于区块链对状态服务实例进行初始化处理；

状态更新模块，用于基于加密传输和会话密钥更新应用程序实例的状态请求；

状态读取模块，用于控制所述应用程序实例读取所述状态服务实例的最新状态；

重启模块，用于当所述状态服务实例发生重启时，基于会话密钥恢复所述状态服务实例的状态；还用于当所述应用程序实例发生重启时，基于操作系统的密封状态数据恢复所述应用程序实例的状态。

本发明第三方面提供一种智能终端，所述智能终端包括存储器、处理器以及存储在所述存储器上并可在所述处理器上运行的基于去中心化信任的公有云下 TEE 状态连续性保护程序，所述基于去中心化信任的公有云下 TEE 状态连续性保护程序被所述处理器执行时实现如上任意一项所述基于去中心化信任的公有云下 TEE 状态连续性保护方法的步骤。

本发明第四方面提供一种计算机可读存储介质，所述计算机可读存储介质上存储有基于去中心化信任的公有云下 TEE 状态连续性保护程序，所述基于去中心化信任的公有云下 TEE 状态连续性保护程序被处理器执行时实现如上任意一项所述基于去中心化信任的公有云下 TEE 状态连续性保护方法的步骤。

由上可见，本发明实施例提供一种基于去中心化信任的公有云下 TEE 状态连续性保护方法，所述基于去中心化信任的公有云下 TEE 状态连续性保护方法包括：基于区块链对状态服务实例进行初始化处理；基于加密传输和会话密钥更新应用程序实例的状态请求；控制所述应用程序实例读取所述状态服务实例的最新状态；当所述状态服务实例发生重启时，基于会话密钥恢复所述状态服务实例的状态；当所述应用程序实例发生重启时，基于操作系统的密封状态数据恢复所述应用程序实例的状态。与现有技术相比，本发明使用区块链消除了对中心化可信实体的依赖，且只依赖区块链进行初始化，提高了状态更新和读取的速度。

附图说明

为了更清楚地说明本发明实施例中的技术方案，下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍，显而易见地，下面描述中的附图仅仅是本发明的一些实施例，对于本领域普通技术人员来讲，在不付出创造性劳动性的前提下，还可以根据这些附图获得其它的附图。

图 1 是本本发明实施例提供一种基于去中心化信任的公有云下 TEE 状态连续性保护方法的流程示意图；

图 2 是本本发明实施例提供的步骤 S10 的流程示意图；

图 3 是本本发明实施例提供的系统初始化协议的交互示意图；

图 4 是本本发明实施例提供的本地状态服务实例的内存布局示意图；

图 5 是本本发明实施例提供的步骤 S20 的流程示意图；

图 6 是本本发明实施例提供的状态更新协议的交互示意图；

图 7 是本本发明实施例提供的步骤 S30 的流程示意图；

图 8 是本本发明实施例提供的状态读取协议的交互示意图；

图 9 是本本发明实施例提供的基于去中心化信任的公有云下 TEE 状态连续性保护系统的结构示意图；

图 10 是本本发明实施例提供的基于去中心化信任的公有云下 TEE 状态连续性保护系统的架构示意图；

图 11 是本本发明实施例提供的智能终端的内部结构原理框图。

具体实施方式

以下描述中，为了说明而不是为了限定，提出了诸如特定系统结构、技术之类的具体细节，以便透彻理解本发明实施例。然而，本领域的技术人员应当清楚，在没有这些具体细节的其它实施例中也可以实现本发明。在其它情况下，省略对众所周知的系统、装置、电路以及方法的详细说明，以免不必要的细节妨碍本发明的描述。

应当理解，当在本说明书和所附权利要求书中使用时，术语“包括”指示所描述特征、整体、步骤、操作、元素和/或组件的存在，但并不排除一个或多个其它特征、整体、步骤、操作、元素、组件和/或其集合的存在或添加。

还应当理解，在本发明说明书中所使用的术语仅仅是出于描述特定实施例的目的而并不意在限制本发明。如在本发明说明书和所附权利要求书中所使用的那样，除非上下文清楚地指明其它情况，否则单数形式的“一”、“一个”及“该”意在包括复数形式。

还应当进一步理解，在本发明说明书和所附权利要求书中使用的术语“和/或”是指相关联列出的项中的一个或多个的任何组合以及所有可能组合，并且包括这些组合。

下面结合本发明实施例的附图，对本发明实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例仅仅是本发明的一部分实施例，而不是全部的实施例。基于本发明中的实施例，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其它实施例，都属于本发明保护的范围。

在下面的描述中阐述了很多具体细节以便于充分理解本发明，但是本发明还可以采用其它不同于在此描述的其它方式来实现，本领域技术人员可以在不违背本发明内涵的情况下做类似推广，因此本发明不受下面公开的具体实施例的限制。

随着科技不断发展，越来越多的厂商选择将业务部署到公有云上，例如英

特尔软件保护扩展 (Software Guard Extension, SGX) 能够使应用程序直接计算, 不会泄露机密数据。例如, 英特尔 SGX 为受保护的计算提供安全的隔离环境飞地 (Enclave), 并将代码和数据存储在飞地 (Enclave) 中, Enclave 是一个被保护的内容容器, 用于存放应用程序敏感数据和代码; 现有技术中通常通过 TEE 为用户提供机密计算服务, 旨在增强客户外包数据或代码的安全和隐私性。但 TEE 中的状态应用程序依赖于不受信任的操作系统来存储状态, 因此会遭受回滚或分叉攻击, 它们的状态可能回滚到正确但已经过时的版本或被分叉成多个版本, 导致状态连续性造成破坏, 另外, 当系统发生重启、断电或崩溃时, 受保护模块应在状态连续性中断之前恢复到相同的状态。

状态连续性一旦违规将会产生严重后果, 例如支付、信用系统、可信存储、智能合约等都会受到影响, 例如, 在 TEE 中实现的支付系统中, 状态连续性一旦违规, 攻击者可以通过还原其账户余额的状态在多个支付中花费相同的硬币 (即双花攻击)。

针对上述问题现有技术中有基于硬件的解决方案, 例如, 使用定时计数器做状态更新, 不但只有有限次的写操作, 且状态更新和读取的速度较慢, 不能满足云环境中对更新状态的要求。基于软件的解决方案, 例如, 将可信计数器装载到单个可信服务器或分布式集合服务器, 则依赖于可信的管理员做系统初始化, 使得这类方案仍然存在依赖中心化信任的问题。

为了解决现有技术中的问题, 本发明提供一种基于去中心化信任的公有云下 TEE 状态连续性保护方法, 该方法依赖于区块链 (即去中心化信任) 和基于 TEE 的分布式系统, 能够为云上用户的受保护 TEE 应用实例提供高性能的状态连续性保护, 例如能够支持应用在云 TEE 环境下无限且快速的状态更新。

示例性方法

如图 1 所示, 本发明实施例提供一种基于去中心化信任的公有云下 TEE 状态连续性保护方法的流程示意图, 具体的, 所述基于去中心化信任的公有云下 TEE 状态连续性保护方法包括:

步骤 S10: 基于区块链对状态服务实例进行初始化处理;

步骤 S20: 基于加密传输和会话密钥更新应用程序实例的状态请求;

步骤 S30: 控制所述应用程序实例读取自身存储在所述状态服务实例中的最新状态;

步骤 S40: 当所述状态服务实例发生重启时, 基于会话密钥恢复所述状态服务实例的状态; 当所述应用程序实例发生重启时, 基于操作系统的密封状态数据恢复所述应用程序实例的状态。

本实施例中, 仅在初始化处理中使用区块链, 既奠定了去中心化的信任, 消除了对中央可信实体的依赖, 同时尽可能减少与区块链的互动, 使得系统可以减少高昂的区块链交互开销, 降低成本。

需要说明的是, 本实施例可以应用在由云服务商管理可信平台上, 例如, 支持英特尔 SGX 的平台, 其中, 每个支持 SGX 的可信平台运行多个飞地 (enclave) 实例, 具体的, 所述飞地实例可以分为以下两种类型:

应用程序实例 (Application Enclave, AE), AE 是由独立软件供应商创建和上传的应用程序, 用于处理客户的请求并返回相应的输出。

状态服务实例 (State Enclave, SE), SE 是为 AE 提供状态连续性服务的 enclave 实例。每个 AE 都可以链接到同一平台上的本地 SE (即目标 SE), 并根据所述方法保护其状态连续性。

本实施例中，所述状态服务实例通过区块链接口与所述区块链进行交互，所述区块链使用拜占庭容错共识作为骨干协议，进一步实现了键值存储，使用户能够通过身份标识号码（Identity Document，ID）识别将元组 $\langle key, blob \rangle$ 写入区块链，其中 *blob* 是任意与键映射的字符串，客户的身份 ID 既可以是账本标识符，也可以是数字的公钥签名方案，只有密钥所有者有权更新关联值，而任何一方都可以使用密钥读取关联的值，密钥的创建者可以加密关联的值以保护数据隐私。

本实施例中，通过所述区块链的第一接口将所述状态服务实例的身份标识号码和公钥写到所述区块链上，并记录初始化信息；通过所述区块链的第二接口根据所述状态服务实例的身份标识号码和公钥在所述区块链上查询初始化信息；通过所述区块链的第三接口在所述应用程序实例更新状态摘要时，向所述状态服务实例请求更新是否合法；通过所述区块链的第四接口获取所述应用程序实例的状态摘要。

具体的，所述第一接口为 `Blockchain.write(ID,  $\langle key, blob \rangle$ )`：用于状态服务实例把请求记录在区块链。ID 是区块链的标识，*key* 和 *blob* 是一对键值存储。如果写入成功，就会返回一个的 σ 。客户端可以运行签名算法进行验证。

所述第二接口为 `Blockchain.read(ID, key)  $\rightarrow$  (blob,  $\sigma$ )`：读取算法返回与 *key* 和验证器 σ 关联的 *blob*。状态服务实例通过 σ 可以检查返回值的完整性。

所述第三接口为 `writeState(value)`：用于在应用程序实例更新自身状态时，向状态服务实例请求更新是否合法，如果合法，返回一个确认字符以确认应用程序实例。

所述第四接口为 `readState()  $\rightarrow$  value`：用于获取所述应用程序实例的状态摘要。需要注意的是，若应用程序实例没有之前写入任何值，则返回一个空值。

通过以上接口，本实施例中应用程序实例可以实行先记录后执行，当应用程序实例更新其状态时，首先密封并存储一个状态快照在磁盘上，然后调用函数 `writeState()` 来存储关联的状态摘要，一旦应用程序实例收到确认字符，就可以更新其状态并发布输出。

当应用程序实例需要验证重新启动后操作系统提供的数据的新鲜度时，调用函数 `readState()` 获取其最新的状态摘要，检查状态摘要是否与未密封的状态快照数据匹配。本发明中，每个应用程序实例都与一个本地状态服务实例链接，本地状态服务实例提供多个接口提高证明速度和状态管理灵活性。

需要说明的是，本实施例的目的是实现一个初始化协议，其中，所有的状态服务实例自主完成配置，不涉及任何受信任的管理员，需要注意的是，在初始化的过程中应该确保只有一组合法的状态服务实例具有已知的在支持 SGX 的可信平台上运行的身份。如果没有此保证，应用程序实例可以链接到多个组，且每个组可以存储不同的状态，将导致违反状态连续性。

由于缺乏可信性，在初始化过程中存在两个挑战：第一、所有状态服务实例必须以分布式方式构建 PKI 方式，这通常会导致较高的消息复杂性。第二、对保证只有一个初始化组，一个集中式系统通常使用受信任的管理员来提供密钥配置。

为了应对第一个挑战，本实施例中采用领导者模式进行初始化，将任意一个所述状态服务实例设置为领导者，领导者可以由云服务商委派，且领导者选择不影响安全性。首先，领导者通过远程证明相互验证其他所述状态服务实例，并与其他所述状态服务实例建立通信通道，互相证明彼此的身份后建立信任链条，然

后，领导者可以收集和分发其他状态服务实例的公钥。

为了应对第二个挑战，本实施例依赖于与区块链的交互，在初始化期间，一个状态服务实例必须在账本上发布记录 $\langle uid, blob \rangle$ ，其中 uid 是平台上飞地（Enclave）程序唯一的标识符，且 $blob$ 可以是配置信息；当另一个状态服务实例在同一个平台开始初始化，它也会生成相同的 uid ，在区块链中找到之前的记录，然后中止初始化。因此，本实施例中在同一个平台不会出现分叉状态服务实例。整个初始化是自主完成的不涉及任何受信任的管理员。

本实施例中，如图 2 所示，图 2 是本发明实施例提供的基于区块链对所述状态服务实例进行初始化处理的流程示意图，所述步骤 S10 具体包括：

步骤 S101：将任意一个所述状态服务实例设置为领导者，通过远程证明相互验证所述领导者和其他状态服务实例的身份，并建立所述领导者与其他状态服务实例的安全通信通道；

步骤 S102：控制所述领导者和其他状态服务实例生成非对称密钥对，并将所述状态服务实例的公钥发送给所述领导者，所述领导者将接收到的公钥列表分发给其他状态服务实例；

步骤 S103：控制每个所述状态服务实例向所述区块链发送读请求，所述读请求包括所述状态服务实例的唯一标识符；在同一可信执行环境平台上的多个应用实例具有相同的标识符；

步骤 S104：控制每个所述状态服务实例向所述区块链发送写请求，当收到来自所述区块链的确认字符时，控制每个所述状态服务实例创建本地内存，所述本地内存包括其他状态服务实例状态条目的状态服务实例状态表和空的应用程序实例状态表；

步骤 S105：控制本地状态服务实例将所述公钥列表放置到所述本地状态服务实例的本地内存以生成状态服务实例配置表，并控制所述本地状态服务实例通过运行经过验证的密钥协议与其他状态服务实例建立成对会话密钥；

步骤 S106：密封并存储所述本地状态服务实例的密钥对、所述状态服务实例配置表和所述应用程序实例状态表。

需要说明的是，所述步骤 S101 至步骤 S104 为初始化协议架构的方法，具体的，如图 3 所示，图 3 是本发明实施例提供的系统初始化协议的交互示意图，通过步骤 S101 建立安全的沟通渠道。领导者使用远程证明相互证明每个其他状态服务实例，同时与所有其他状态服务实例建立安全的通信通道。需要注意的是，在创建时，每个状态服务实例（包括领导者）都需要计算它的飞地（Enclave）身份，并比较其与其他人的身份，一起验证被证明的飞地（Enclave）是否是状态服务实例。领导者使用远程证明来相互验证其他每个状态服务实例，同时与这些状态服务实例建立安全的通信通道，在创建时，每个状态服务实例（包括领导者）都会计算其 Enclave 标识（即 MRENCLAVE），并将其标识与其他 Enclave 的身份进行比较，以验证已证明的 Enclave 是否为状态服务实例。

通过步骤 S102 进行身份建立，所有状态服务实例（包括领导者）生成一个非对称密钥对 (pk, sk) ，并将它们的公钥发送给领导者，领导者收到公钥后分发收到的公钥列表（由 $Cert$ 表示）到其他状态服务实例，以便他们可以相互验证。

通过步骤 S103 进行第一次初始化检查，每个状态服务实例发送一个读请求到第二接口 $Blockchain.read(ID, uid)$ 的分类帐，其中 ID 为状态服务实例的账本标识符（由它的 pk 生成），而 uid 是状态服务实例在启用 SGX 的可信平台上的唯一标识符。当另一个状态服务实例在同一个平台开始初始化，它也会生成相同

的 *uid*，状态服务实例向区块链发送读取请求，区块链根据请求来返回 *blob*，如果 *blob* 为空，说明该状态服务实例没有被初始化，是第一次加入群组。如果返回的 *blob* 不为空，说明已经有另一个状态服务实例初始化，则状态服务实例将中止初始化。

通过步骤 S104 记录初始化，状态服务实例通过第一接口 `Blockchain.write(ID, <uid, H(Cert)>)` 发送写请求，当收到一个来自区块链的确认字符 (ACK) 时，状态服务实例创建一个包含其他状态服务实例的状态条目的状态服务实例状态表和一个空的应用程序实例状态表，状态服务实例将公钥列表 *pkSE_i* 放到状态服务实例配置表并建立成对话密钥 *keySE_i* 与所有其他状态服务实例通过运行经过身份验证的密钥协议，如图 4 所示，图 4 是本发明实施例提供的本地状态服务实例的内存布局示意图，其中的数据灰色矩形被密封用于状态快照，状态服务实例密封并存储其密钥对、状态服务实例配置表 and 应用程序实例状态表，最终状态服务实例完成初始化并服务应用程序实例的请求。

需要说明的是，步骤 S103 中的标识符 *uid* 是 **Enclave** 身份标识，使其他人能够验证身份报价是由同一个平台发出。状态服务实例应在步骤 S104 执行期间保持活动状态。因为无法区分正常崩溃是否来自创建分叉组的恶意行为，如果状态服务实例在收到 *blob* 之前崩溃，则无法初始化。然而，因为初始化过程只会发生一次，本地状态服务实例的身份可以硬编码到应用程序实例实现或由独立软件开发商提供。应用程序实例可以与本地状态服务实例运行经过身份验证的密钥建立协议，并通过该协议共享密钥 *k*；状态服务实例为应用程序实例创建一个状态条目，包括唯一的身份应用程序实例（即使用应用程序实例代码的哈希）、共享密钥和一个空的状态摘要，可参见图 4 中的应用程序实例状态表。

本实施例中，如图 5 所示，图 5 是本发明实施例提供的基于加密传输和会话密钥更新所述应用程序实例的状态请求的流程示意图，所述步骤 S20 包括以下步骤：

步骤 S201：当所述应用程序实例接收到用户请求时，密封并存储所述应用程序实例的状态快照到磁盘上，并更新所述应用程序实例的状态摘要，向所述本地状态服务实例发送状态更新请求；

步骤 S202：当所述本地状态服务实例接收到状态更新请求时，存储所述状态更新请求到请求队列；

步骤 S203：当处理所述状态更新请求时，控制所述本地状态服务实例向全部状态服务实例（包括本地状态服务实例和其他状态服务实例）发送所述状态更新请求；

步骤 S204：当全部状态服务实例（包括本地状态服务实例和其他状态服务实例）接收到所述状态更新请求时，更新全部状态服务实例（包括本地状态服务实例和其他状态服务实例）状态内存中所述本地状态服务实例的状态摘要并发送回复消息；

步骤 S205：当所述本地状态服务实例收到超过 $f+1$ 个所述回复消息时，所述本地状态服务实例向其他状态服务实例发送决定消息，其中， f 为共识节点容错数；

步骤 S206：当其他状态服务实例收到所述决定消息后，检查其他状态服务实例是否已回复所述准备消息，若是，则控制所述状态服务实例发回包含状态摘要的确认字符；

步骤 S207：当所述本地状态服务实例收到超过 $f+1$ 个所述确认字符时，控

制所述本地状态服务实例更新所述应用程序实例的状态摘要并向所述应用程序实例回复包含状态摘要的确认字符；

步骤 S208：当所述应用程序实例收到所述确认字符时，将其状态更新为最新状态并发布相关输出。

需要说明的是，所述步骤 S201 至步骤 S207 为状态更新协议的架构方法，具体的，如图 6 所示，图 6 是本发明实施例提供的状态更新协议的交互示意图，也就是说，当本地状态服务实例准备好处理应用程序实例的状态更新请求时，遵循图 6 所示的状态更新协议，所述状态更新协议基于回声广播协议，应用程序实例和状态服务实例的消息通过加密方式传输和使用秘密会话密钥更新所述应用程序实例的状态请求。此外，可以为每条消息添加随机数以防止消息重播攻击。

举例说明，当接收到用户请求 I_i 时，应用程序实例首先密封并存储状态磁盘上的快照 $\langle S_i, I_i, SD_{i-1} \rangle$ ，其中状态摘要 SD_{i-1} 为 $H(S_{i-1}||I_{i-1})$ 。然后，应用程序实例调用函数 $writeState(SD_i)$ 更新它的状态摘要，为了清楚起见，假设需要随机性 R_i 附有 S_i ；本地状态服务实例收到状态更新请求时，将其缓存在先进先出(FIFO)队列中，一旦请求在头部并且没有其他服务请求，状态服务实例密封并存储它在磁盘上的状态快照 $\langle S_j, I_j, SD_{j-1} \rangle$ ，其中 I_j 是应用程序实例的状态摘要 SD_i ；状态服务实例向全部状态服务实例（包括当前状态服务实例和其他状态服务实例）发送准备消息 $\langle Prepare, SD_j(j, seq) \rangle$ ；当接收到 Prepare 消息时，每个状态服务实例都会更新状态内存中本地状态服务实例的摘要并发送回复消息，所述回复消息包含 SD_j 的消息；收到超过 $f+1$ 个回复消息后，本地状态服务实例向组中所有状态服务实例发送决定消息 $\langle Decide, SD_j, (j, seq) \rangle$ ；当状态服务实例收到决定消息后，检查是否已回显相关的准备消息。如果是，则发回包含 SD_j 的确认字符。当接收到超过 $f+1$ 个确认字符时，本地状态服务实例更新应用程序实例的状态摘要并返回包含 SD_i 的确认字符。当应用程序实例接收到确认字符后，应用程序实例将其状态更新为 S_{i+1} 并且发布相关的输出 O_i 。

需要说明的是，所述状态更新协议可以在较小的传输控制协议中实现，首先，在处理请求时，应用程序实例必须进行密封和存储本地磁盘中的状态快照。对于某些应用程序，如支付，整个状态很大，因此此操作会产生加密数据和输入输出的繁重任务；其次，状态服务实例依次处理应用程序实例的状态更新。序列化会限制状态服务实例每秒处理请求的能力（即吞吐量）；此外，当支持的数量应用程序实例和关联请求很大，吞吐量低会导致到长时间的服务延迟，每次更新涉及两轮状态服务实例之间的消息交换，是影响的主要因素延迟和吞吐量。

为了解决上述问题，本实施例中，采用以下方式优化所述状态更新协议：

(1) 利用检查点减少密封和存储数据的大小，此方式适用于应用程序实例和状态服务实例。为了清楚起见，我们从应用程序实例的角度描述优化，需要说明的是，由于程序是确定性的，可以恢复最新状态执行所有历史输入，也就是说，应用程序实例只需要密封并将输入请求存储在磁盘上，同时记录摘要状态服务实例中的输入历史，对于大多数应用程序，输入通常是远小于整个应用程序状态，因此可以减少密封和存储数据的大小。

(2) 利用批处理处理提升系统处理请求的能力。使状态服务实例可以同时服务多个运行不同应用程序的应用程序实例,即不需要串行处理不同应用程序实例的状态更新请求。在步骤 S202 中状态服务实例获取一批来自队列的请求,并将这些请求作为一个请求。然后,状态服务实例遵循步骤 S203 至步骤 S207 处理这些请求。执行上述步骤后,状态服务实例会一一处理并发送确认字符。批处理的开销可以忽略不计,原因有以下两个:首先,密封和存储数据的大小仅略有增加执行时间,其次,协议传输的网络消息(即准备消息和决定消息)的大小保持不变,因为输出大小无论输入的大小如何,散列函数的值都是固定的。

当应用程序实例想要检查其状态或密封状态的新鲜度时。应用程序实例调用函数 `readState()` 获取最新的来自本地状态服务实例的状态摘要。当接收到状态读取请求时,本地状态服务实例无法直接返回应用程序实例的记录状态。主要是因为可能因为存在多个本地状态服务实例(即分叉攻击),因此请求的状态服务实例保存的状态不是最新的。为了解决这个问题,应用程序实例必须检查它的状态是否是最新的,如果是,则安全地返回应用程序实例的状态摘要。所述控制所述应用程序实例读取所述状态服务实例的最新状态,如图 7 所示。图 7 是本发明实施提供的所述应用程序实例读取所述状态服务实例的最新状态的流程示意图,步骤 S30 包括以下步骤:

步骤 S301: 控制所述应用程序实例调用函数向所述本地状态服务实例发送状态读取请求;

步骤 S302: 当所述本地状态服务实例收到所述状态读取请求时,检索全部所述状态服务实例的最新状态;

步骤 S303: 检查每个所述状态服务实例的状态服务实例状态表,并将存储的状态摘要发回给所述本地状态服务实例;

步骤 S304: 当收到 $f+1$ 个所述状态摘要时,选择索引最大的状态摘要作为所述本地状态服务实例的状态摘要。

需要说明的是,所述步骤 S301 至步骤 S304 为状态读取协议的架构方法,具体的,如图 8 所示。图 8 是本发明实施例提供的状态读取协议的交互示意图,应用程序实例调用 `readState()` 函数向目标状态服务实例发送状态读取请求,本地状态服务实例收到状态读取请求时,检索组中所有状态服务实例的最新状态,每个状态服务实例检查其状态服务实例状态表,并将存储的状态摘要发回本地状态服务实例,在收到至少 $f+1$ 个回复后,本地状态服务实例选择具有最大索引和序列号的状态摘要;然后,本地状态服务实例将选择的状态摘要与前一次的状态摘要进行比较,如果它的状态是最新的,则发回记录的状态摘要给应用程序实例,否则,不回复。

会话密钥作为状态服务实例的身份,在同一平台上只允许有一个正在工作中的状态服务实例。然而,现有技术中的重启协议存在安全问题,即对手可以利用该协议创建两个并行运行的系统,然后启动对 TEE 应用程序的回滚和分叉攻击。为了解决上述问题,本实施例中当所述状态服务实例发生重启时,基于会话密钥恢复所述状态服务实例的状态;也就是说,重启后的状态服务实例必须与其他状态服务实例建立新的会话密钥。

当所述状态服务实例发生重启时,所述基于会话密钥恢复所述状态服务实例的状态包括以下步骤:

控制重启后的所述状态服务实例与其他状态服务实例建立新的会话密钥;

需要说明的是,当某一个状态服务实例(SE)被强制重启后,重启后的状

态服务实例（SE）会失去所有的内存数据，但是之前建立的会话控制（session）会保存在状态服务实例（SE）中，此时重启后的状态服务实例（SE）无法重新加入集群。本实施例中，每一个状态服务实例（SE）会存在一个“ping”的功能，检查自己是否仍与其他状态服务实例保持链接。

控制重启后的所述状态服务实例执行所述步骤 S302 至所述步骤 S304 以获取所述状态服务实例和其他状态服务实例的状态摘要；

选择索引最大的状态摘要作为重启后的所述状态服务实例的状态摘要，重启的状态服务实例（SE）需要恢复内存数据，因此选择最新的状态摘要；

控制所述状态服务实例从操作系统获取密封的状态摘要，将所述状态摘要与从其他状态服务实例获取的状态摘要进行匹配，若匹配，则控制所述状态服务实例恢复状态摘要对应的状态。

具体的，重启后的服务实例需要与其他状态服务实例建立新的会话密钥，因为它已经失去了以前的全部数据。其次，在会话密钥建立之前，其他状态服务实例必须检查他们是否仍维持与其他状态服务实例的链接，即他们的会话密钥是否仍处于活动状态，为此，每个状态服务实例发送一个 ping 请求向其他状态服务实例发送消息并等待 ACK，如果他们无法收到来自其他状态服务实例的 ACK，则拒绝与重启状态服务实例建立新的会话密钥；重启状态服务实例执行状态读取协议中的步骤 S302 至步骤 S304 以获取其和其他状态服务实例的最新状态摘要，步骤 S302 至步骤 S304 可以在一个读取请求中；重启状态服务实例选择索引最大的状态摘要，也就是最新的状态摘要作为每个状态服务实例（包括重启状态服务实例）的序列号。为了重启后的状态服务实例和群组保持同步，增加序列号并重新执行状态更新协议中步骤 S202 至步骤 S206 来更新其最新的状态摘要。如此，可以保证最新的状态摘要被安全地记录在至少 $f+1$ 个状态服务实例上。

当所述应用程序实例发生重启时，所述基于操作系统的密封状态数据恢复所述应用程序实例的状态，包括以下步骤：

控制重启后的所述应用程序实例执行所述步骤 S301 至所述步骤 S304 向所述本地状态服务实例请求最新状态摘要；

控制所述应用程序实例获取所述操作系统的密封状态数据，将所述密封状态数据与所述最新状态摘要进行匹配，若匹配成功，则更新重启后所述应用程序实例的状态摘要并发布输出。

具体的，重启后的重启应用程序实例执行状态读取协议向本地状态服务实例请求最新的状态摘要；并从操作系统获得重启应用程序实例的最新密封状态，将最新密封状态与所述本地状态服务实例的状态摘要进行匹配，如果这两个匹配，将应用程序实例的状态更新为最新状态并发布相关输出。

由上可见，本发明实施例提供一种基于去中心化信任的公有云下 TEE 状态连续性保护方法，所述基于去中心化信任的公有云下 TEE 状态连续性保护包括：基于区块链对状态服务实例进行初始化处理；基于加密传输和会话密钥更新应用程序实例的状态请求；控制所述应用程序实例读取所述状态服务实例的最新状态；当所述状态服务实例发生重启时，基于会话密钥恢复所述状态服务实例的状态；当所述应用程序实例发生重启时，基于操作系统的密封状态数据恢复所述应用程序实例的状态。与现有技术相比，本发明使用区块链消除了对中心化可信实体的依赖，且只依赖区块链进行初始化，提高了状态更新和读取的速度。

示例性设备

如图 9 中所示，对应于所述基于去中心化信任的公有云下 TEE 状态连续性

保护方法，本发明实施例还提供一种基于去中心化信任的公有云下 TEE 状态连续性保护系统，所述基于去中心化信任的公有云下 TEE 状态连续性保护系统包括：

初始化模块 100，用于基于区块链对状态服务实例进行初始化处理；

状态更新模块 200，用于基于加密传输和会话密钥更新应用程序实例的状态请求；

状态读取模块 300，用于控制所述应用程序实例读取所述状态服务实例的最新状态；

重启模块 400，用于当所述状态服务实例发生重启时，基于会话密钥恢复所述状态服务实例的状态；还用于当所述应用程序实例发生重启时，基于操作系统的密封状态数据恢复所述应用程序实例的状态。

如图 10 所示，所述系统包括：可信平台及区块链，所述应用程序实例（AE）和所述状态服务实例（SE）运行在所述可信平台，所述应用程序实例处理客户的请求并返回相应的输出，所述状态服务实例为所述应用程序实例提高状态连续性保护。

具体地，本实施例中，所述基于去中心化信任的公有云下 TEE 状态连续性保护系统及其各个模块的具体功能还可以参照所述基于去中心化信任的公有云下 TEE 状态连续性保护方法中的对应描述，在此不再赘述。

基于所述实施例，本发明还提供了一种智能终端，其原理框图可以如图 11 所示。所述智能终端包括存储器、处理器以及存储在所述存储器上并可在所述处理器上运行的基于去中心化信任的公有云下 TEE 状态连续性保护程序，所述基于去中心化信任的公有云下 TEE 状态连续性保护程序被所述处理器执行时实现如上任意一项所述基于去中心化信任的公有云下 TEE 状态连续性保护方法的步骤。

本领域技术人员可以理解，图 11 中示出的原理框图，仅仅是与本发明方案相关的部分结构的框图，并不构成对本发明方案所应用于其上的智能终端的限定，具体地智能终端可以包括比图中所示更多或更少的部件，或者组合某些部件，或者具有不同的部件布置。

在一个实施例中，所述基于去中心化信任的公有云下 TEE 状态连续性保护程序被所述处理器执行时进行以下操作指令：

基于区块链对状态服务实例进行初始化处理；

基于加密传输和会话密钥更新应用程序实例的状态请求；

控制所述应用程序实例读取所述状态服务实例的最新状态；

当所述状态服务实例发生重启时，基于会话密钥恢复所述状态服务实例的状态；当所述应用程序实例发生重启时，基于操作系统的密封状态数据恢复所述应用程序实例的状态。

本发明实施例还提供一种计算机可读存储介质，所述计算机可读存储介质上存储有基于去中心化信任的公有云下 TEE 状态连续性保护程序，所述基于去中心化信任的公有云下 TEE 状态连续性保护程序被处理器执行时实现如上任意一项所述基于去中心化信任的公有云下 TEE 状态连续性保护方法的步骤。

应理解，所述实施例中各步骤的序号大小并不意味着执行顺序的先后，各过程的执行顺序应以其功能和内在逻辑确定，而不对本发明实施例的实施过程构成任何限定。

所属领域的技术人员可以清楚地了解到，为了描述的方便和简洁，仅以所

述各功能单元、模块的划分进行举例说明，实际应用中，可以根据需要而将所述功能分配由不同的功能单元、模块完成，即将所述装置的内部结构划分成不同的功能单元或模块，以完成以上描述的全部或者部分功能。实施例中的各功能单元、模块可以集成在一个处理单元中，也可以是各个单元单独物理存在，也可以两个或两个以上单元集成在一个单元中，所述集成的单元既可以采用硬件的形式实现，也可以采用软件功能单元的形式实现。另外，各功能单元、模块的具体名称也只是为了便于相互区分，并不用于限制本发明的保护范围。所述系统中单元、模块的具体工作过程，可以参考前述方法实施例中的对应过程，在此不再赘述。

在所述实施例中，对各个实施例的描述都各有侧重，某个实施例中未详述或记载的部分，可以参见其它实施例的相关描述。

本领域普通技术人员可以意识到，结合本文中所公开的实施例描述的各实施例的单元及算法步骤，能够以电子硬件、或者计算机软件和电子硬件的结合来实现。这些功能究竟是以硬件还是软件方式来执行，取决于技术方案的特定应用和设计约束条件。专业技术人员可以对每个特定的应用来使用不同方法来实现所描述的功能，但是这种实现不应认为超出本发明的范围。

在本发明所提供的实施例中，应该理解到，所揭露的装置/终端设备和方法，可以通过其它的方式实现。例如，以上所描述的装置/终端设备实施例仅仅是示意性的，例如，所述模块或单元的划分，仅仅为一种逻辑功能划分，实际实现时可以由另外的划分方式，例如多个单元或组件可以结合或者可以集成到另一个系统，或一些特征可以忽略，或不执行。

所述集成的模块/单元如果以软件功能单元的形式实现并作为独立的产品销售或使用，可以存储在一个计算机可读存储介质中。基于这样的理解，本发明实现所述实施例方法中的全部或部分流程，也可以通过计算机程序来指令相关的硬件来完成，所述计算机程序可存储于一计算机可读存储介质中，该计算机程序在被处理器执行时，可实现所述各个方法实施例的步骤。其中，所述计算机程序包括计算机程序代码，所述计算机程序代码可以为源代码形式、对象代码形式、可执行文件或某些中间形式等。所述计算机可读介质可以包括：能够携带所述计算机程序代码的任何实体或装置、记录介质、U 盘、移动硬盘、磁碟、光盘、计算机存储器、只读存储器（ROM，Read-Only Memory）、随机存取存储器（RAM，Random Access Memory）、电载波信号、电信信号以及软件分发介质等。需要说明的是，所述计算机可读存储介质包含的内容可以根据司法管辖区内立法和专利实践的要求进行适当的增减。

以上所述实施例仅用以说明本发明的技术方案，而非对其限制；尽管参照前述实施例对本发明进行了详细的说明，本领域的普通技术人员应当理解：其依然可以对前述各实施例所记载的技术方案进行修改，或者对其中部分技术特征进行等同替换；而这些修改或者替换，并不是相应技术方案的本质脱离本发明各实施例技术方案的精神和范围，均应包含在本发明的保护范围之内。

权利要求书

1. 一种基于去中心化信任的公有云下 TEE 状态连续性保护方法，其特征在于，所述基于去中心化信任的公有云下 TEE 状态连续性保护方法包括：

基于区块链对状态服务实例进行初始化处理；

基于加密传输和会话密钥更新应用程序实例的状态请求；

控制所述应用程序实例读取存储在所述状态服务实例中的最新状态；

当所述状态服务实例发生重启时，基于会话密钥恢复所述状态服务实例的状态；当所述应用程序实例发生重启时，基于操作系统的密封状态数据恢复所述应用程序实例的状态。

2. 根据权利要求1所述的基于去中心化信任的公有云下 TEE 状态连续性保护方法，其特征在于，所述基于去中心化信任的公有云下 TEE 状态连续性保护方法还包括：

通过所述区块链的第一接口将所述状态服务实例的身份标识号码和公钥写到所述区块链上，并记录初始化信息；

通过所述区块链的第二接口根据所述状态服务实例的身份标识号码和公钥在所述区块链上查询初始化信息；

通过所述区块链的第三接口在所述应用程序实例更新状态摘要时，向所述状态服务实例请求更新是否合法；

通过所述区块链的第四接口获取所述应用程序实例的状态摘要。

3. 根据权利要求1所述的基于去中心化信任的公有云下 TEE 状态连续性保护方法，其特征在于，所述基于区块链对所述状态服务实例进行初始化处理，具体包括：

将任意一个所述状态服务实例设置为领导者，通过远程证明相互验证所述领导者和其他状态服务实例的身份，并建立所述领导者与其他状态服务实例的安全通信信道；

控制所述领导者和其他状态服务实例生成非对称密钥对，并将所述状态服务实例的公钥发送给所述领导者，所述领导者将接收到的公钥列表分发给其他状态服务实例；

控制每个所述状态服务实例向所述区块链发送读请求，所述读请求包括所述状态服务实例的唯一标识符；

控制每个所述状态服务实例向所述区块链发送写请求，当收到来自所述区块链的确认字符时，控制每个所述状态服务实例创建本地内存，所述本地内存包括其他状态服务实例状态条目的状态服务实例状态表和空的应用程序实例状态表；

控制本地状态服务实例将所述公钥列表放置到所述本地状态服务实例的本地内存以生成状态服务实例配置表，并控制所述本地状态服务实例通过运行经过验证的密钥协议与其他状态服务实例建立成对会话密钥；

密封并存储所述本地状态服务实例的密钥对、所述状态服务实例配置表和所述应用程序实例状态表。

4. 根据权利要求3所述的基于去中心化信任的公有云下 TEE 状态连续性保护方法，其特征在于，所述基于加密传输和会话密钥更新所述应用程序实例的状态请求，具体包括：

当所述应用程序实例接收到用户请求时，密封并存储所述应用程序实例的状态快照到磁盘上，并更新所述应用程序实例的状态摘要，向所述本地状态服务实例发送状态更新请求；

当所述本地状态服务实例接收到状态更新请求时，存储所述状态更新请求到请求队列；

当处理所述状态更新请求时，控制所述本地状态服务实例向所述本地状态服务实例和其他状态服务实例发送所述状态更新请求；

当所述本地状态服务实例和其他状态服务实例接收到所述状态更新请求时，更新所述本地状态服务实例和其他状态服务实例状态内存中所述本地状态服务实例的状态摘要并发送回复消息；

当所述本地状态服务实例收到超过 $f+1$ 个所述回复消息时，所述本地状态服务实例向其他状态服务实例发送决定消息，其中， f 为共识节点容错数；

当其他状态服务实例收到所述决定消息后，检查其他状态服务实例是否已回复准备消息，若是，则控制所述状态服务实例发回包含状态摘要的确认字符；

当所述本地状态服务实例收到超过 $f+1$ 个所述确认字符时，控制所述本地状态服务实例更新所述应用程序实例的状态摘要并向所述应用程序实例回复包含状态摘要的确认字符；

当所述应用程序实例收到所述确认字符时，将其状态更新为最新状态并发布相关输出。

5. 根据权利要求4所述的基于去中心化信任的公有云下 TEE 状态连续性保护方法，其特征在于，所述控制所述应用程序实例读取所述状态服务实例的最新状态，具体包括：

控制所述应用程序实例调用函数向所述本地状态服务实例发送状态读取请求；

当所述本地状态服务实例收到所述状态读取请求时，检索全部所述状态服务实例的最新状态；

检查每个所述状态服务实例的状态服务实例状态表，并将存储的状态摘要发回给所述本地状态服务实例；

当收到 $f+1$ 个所述状态摘要时，选择索引最大的状态摘要作为所述本地状态服务实例的状态摘要。

6. 根据权利要求5所述的基于去中心化信任的公有云下 TEE 状态连续性保护方法，其特征在于，所述当所述状态服务实例发生重启时，基于会话密钥恢复所述状态服务实例的状态，具体包括：

控制重启后的所述状态服务实例与其他状态服务实例建立新的会话密钥；

控制重启后的所述状态服务实例获取所述状态服务实例和其他状态服务实例的状态摘要；

选择索引最大的状态摘要作为重启后的所述状态服务实例的状态摘要；

控制所述状态服务实例从操作系统获取密封的状态摘要，将所述状态摘要与从其他状态服务实例获取的状态摘要进行匹配，若匹配，则控制所述状态服务实例恢复所述状态服务实例的状态摘要。

7. 根据权利要求6所述的基于去中心化信任的公有云下 TEE 状态连续性保护方法，其特征在于，所述当所述应用程序实例发生重启时，基于操作系统的密封状态数据恢复所述应用程序实例的状态，具体包括：

控制重启后的所述应用程序实例向所述本地状态服务实例请求最新状态摘要；

控制所述应用程序实例获取所述操作系统的密封状态数据，将所述密封状态数据与所述最新状态摘要进行匹配，若匹配成功，则更新重启后所述应用程序

序实例的状态摘要并发布输出。

8. 一种基于去中心化信任的公有云下 TEE 状态连续性保护系统,其特征在
于,所述基于去中心化信任的公有云下 TEE 状态连续性保护系统包括:

初始化模块,用于基于区块链对状态服务实例进行初始化处理;

状态更新模块,用于基于加密传输和会话密钥更新应用程序实例的状态请
求;

状态读取模块,用于控制所述应用程序实例读取所述状态服务实例的最新
状态;

重启模块,用于当所述状态服务实例发生重启时,基于会话密钥恢复所述
状态服务实例的状态;还用于当所述应用程序实例发生重启时,基于操作系统的
密封状态数据恢复所述应用程序实例的状态。

9. 一种智能终端,其特征在在于,所述智能终端包括存储器、处理器以及存
储在所述存储器上并可在所述处理器上运行的基于去中心化信任的公有云下
TEE 状态连续性保护程序,所述基于去中心化信任的公有云下 TEE 状态连续性
保护程序被所述处理器执行时实现如权利要求 1-7 任意一项所述基于去中心化
信任的公有云下 TEE 状态连续性保护方法的步骤。

10. 一种计算机可读存储介质,其特征在在于,所述计算机可读存储介质上
存储有基于去中心化信任的公有云下 TEE 状态连续性保护程序,所述基于去中
心化信任的公有云下 TEE 状态连续性保护程序被处理器执行时实现如权利要
求 1-7 任意一项所述基于去中心化信任的公有云下 TEE 状态连续性保护方法
的步骤。

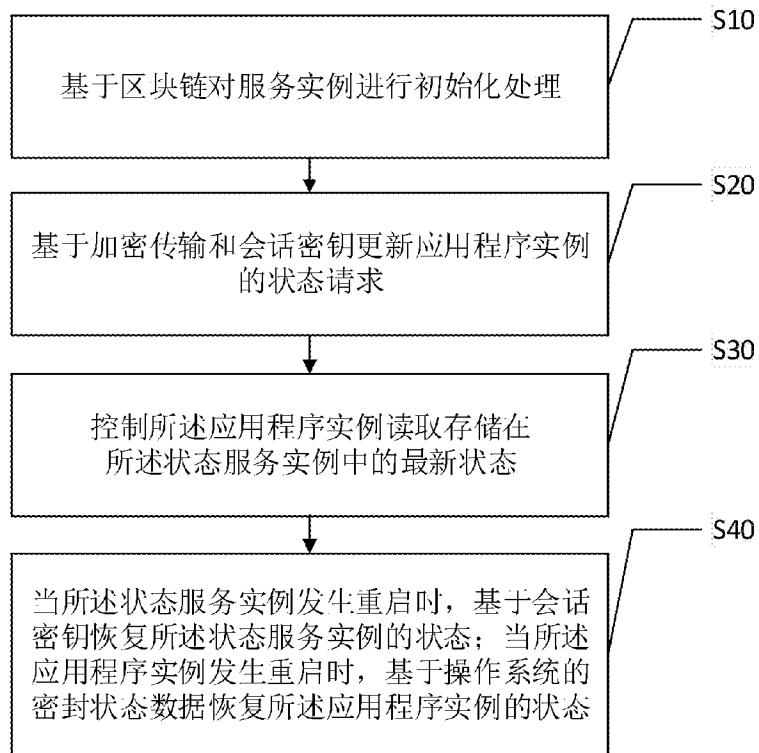


图 1

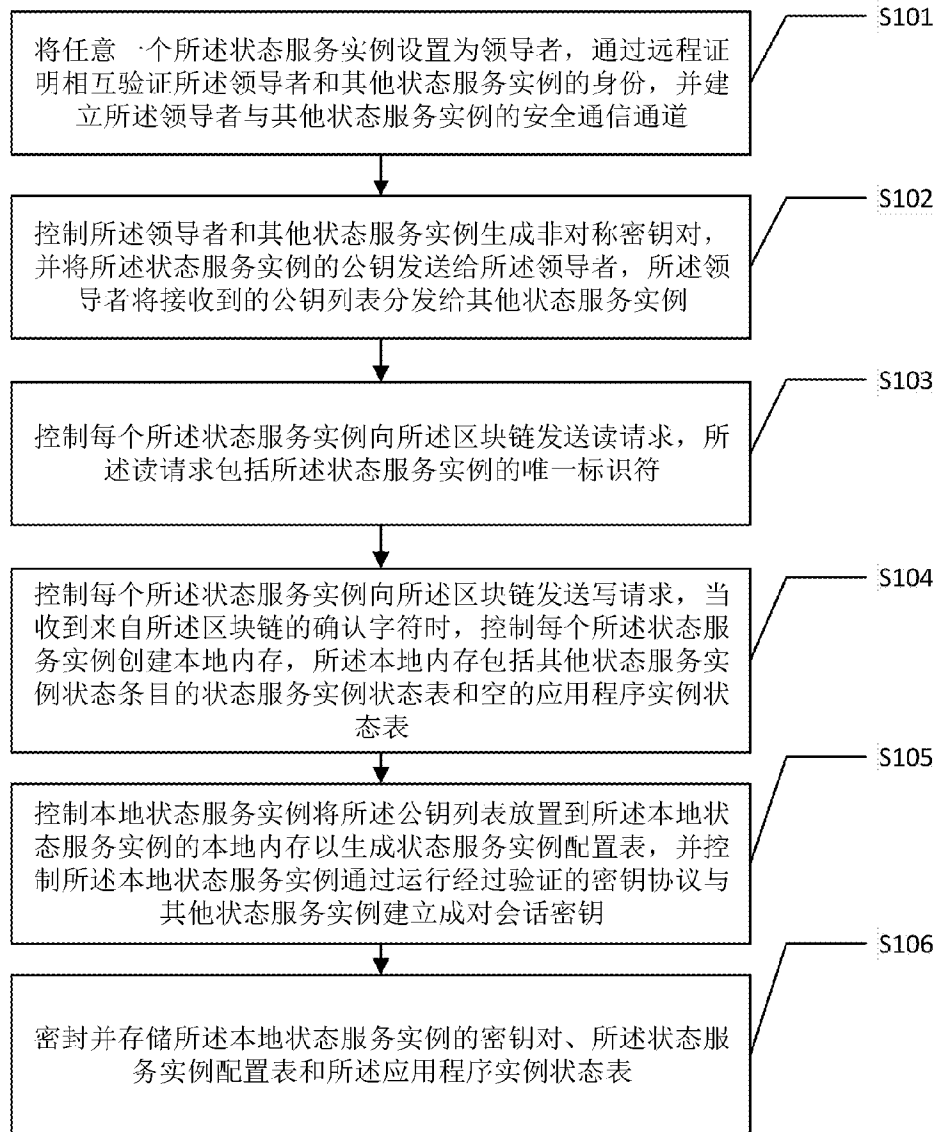


图 2

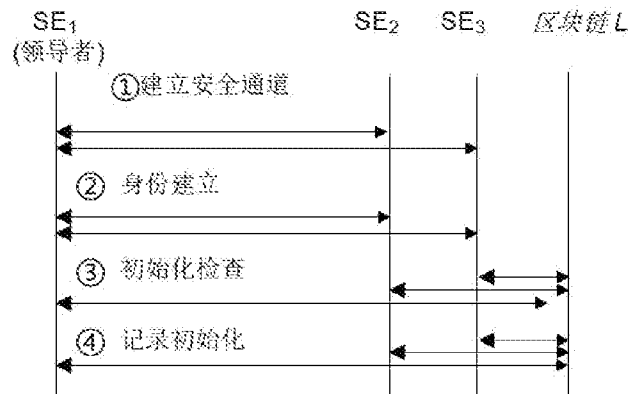


图 3



图 4

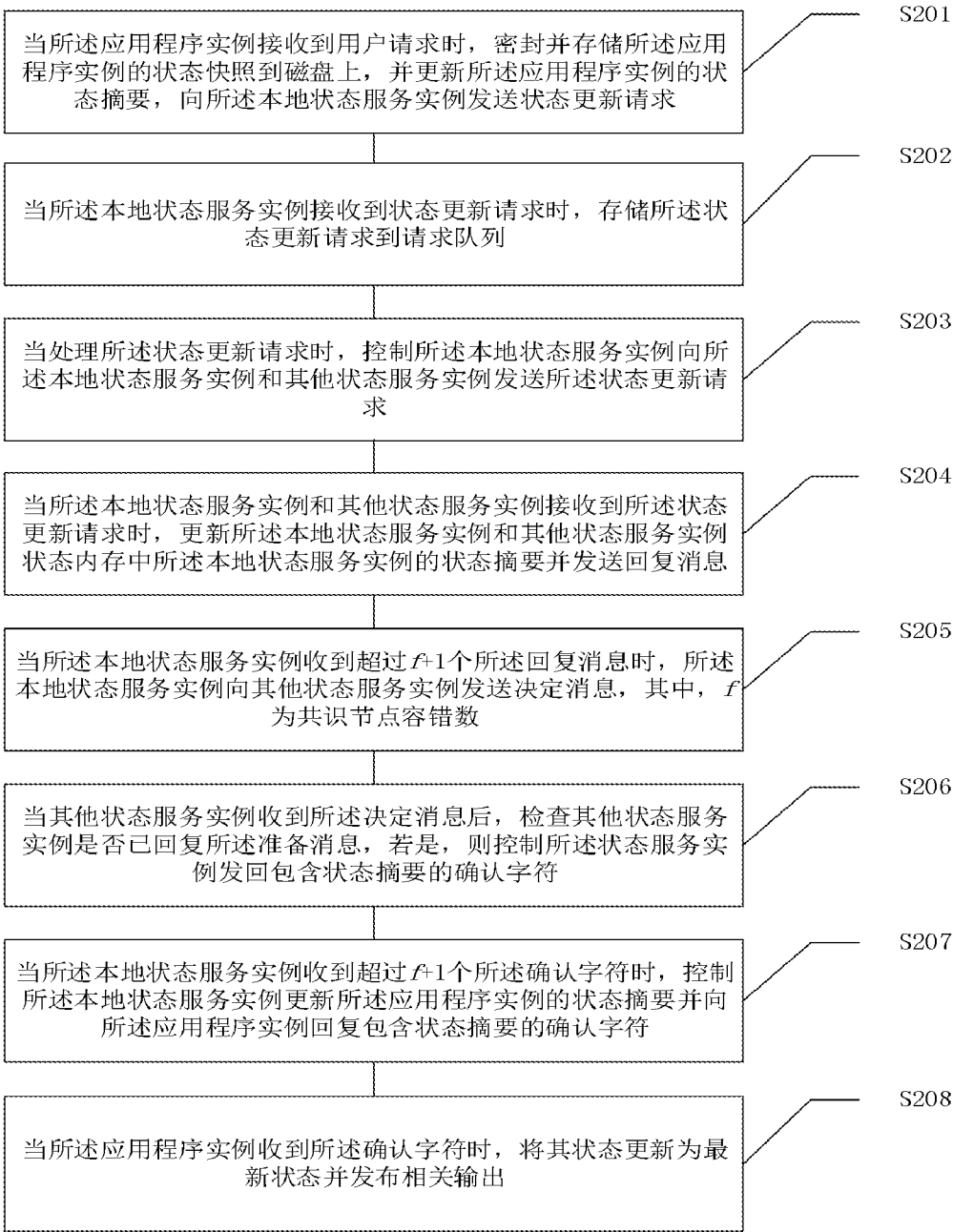


图 5

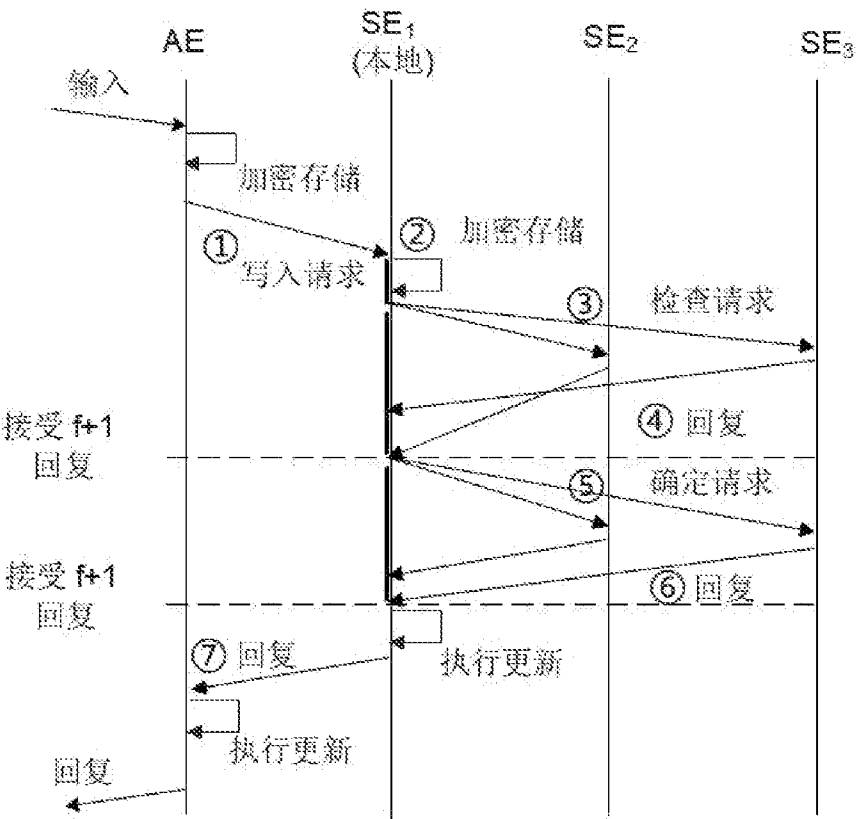


图 6

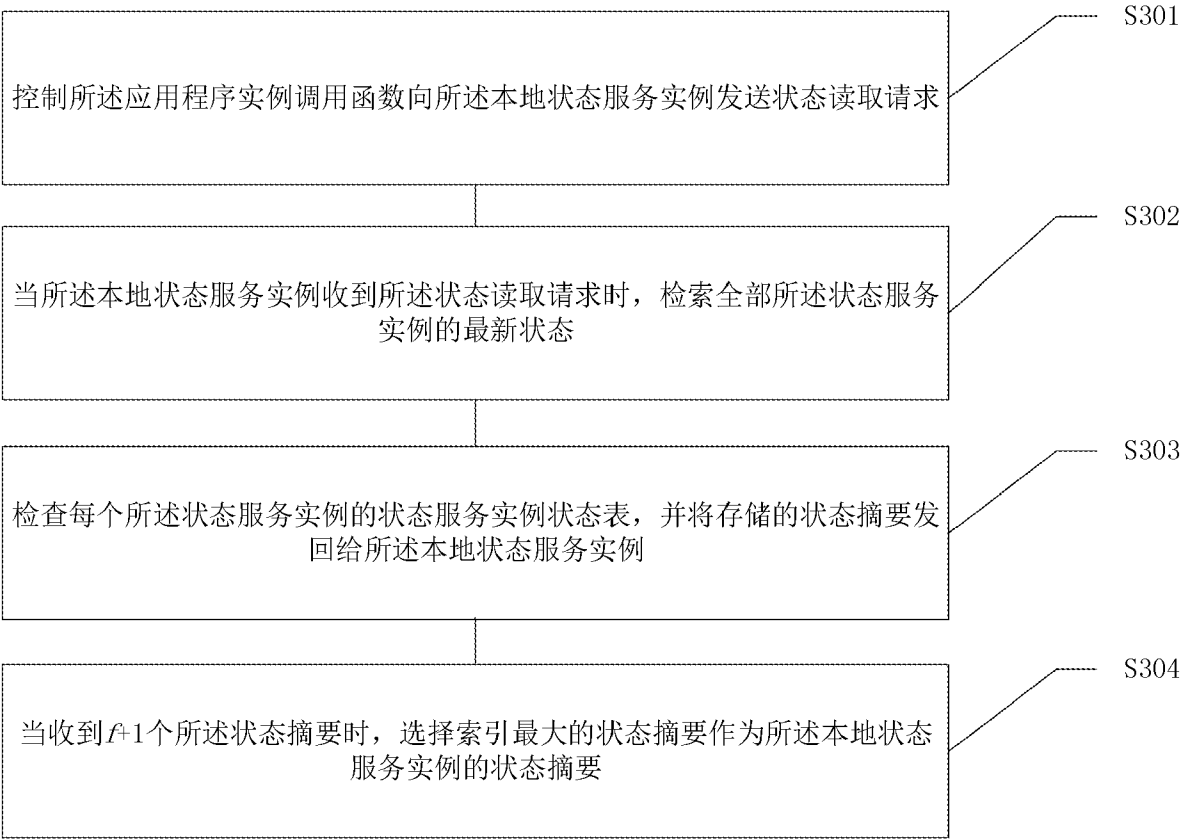


图 7

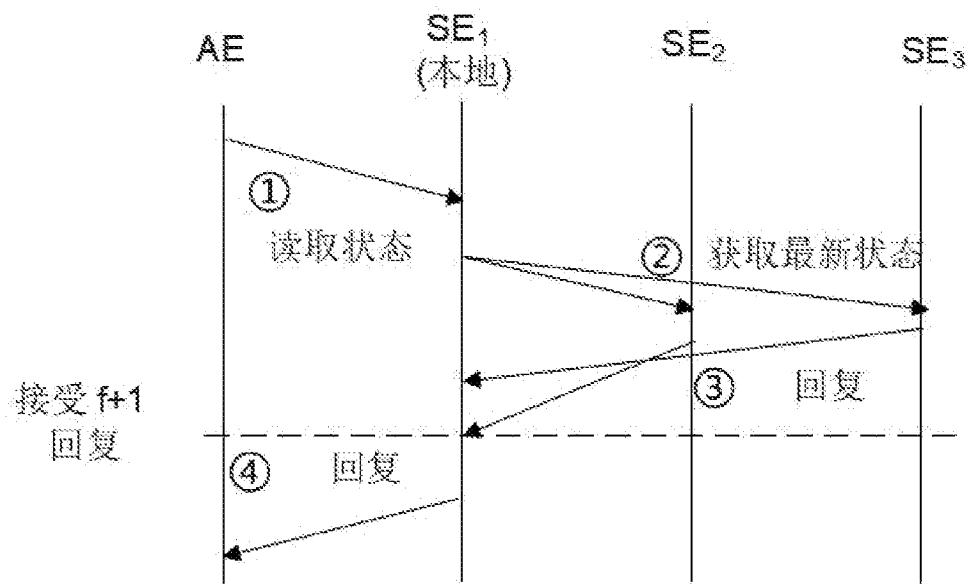


图 8

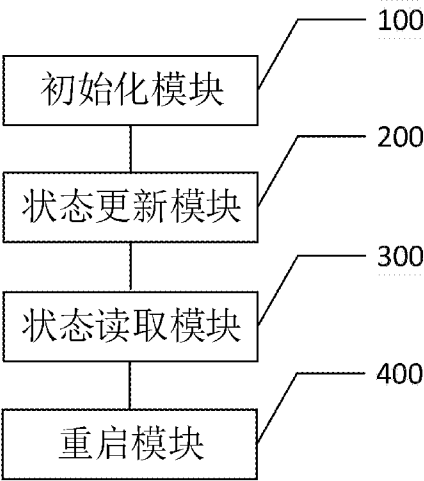


图 9

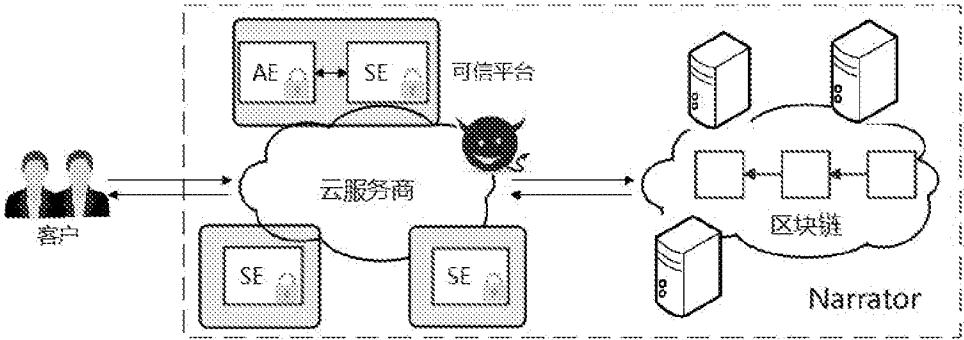


图 10

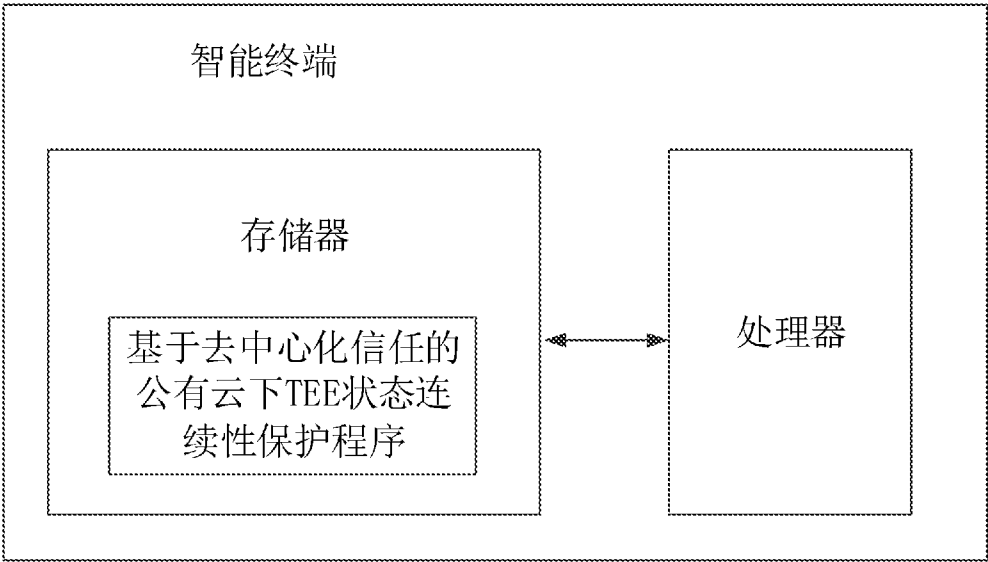


图 11

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2022/119945

A. CLASSIFICATION OF SUBJECT MATTER

G06F21/57(2013.01)i;H04L9/28(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC: G06F; H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

VEN, CNTXT, CNABS, USTXT, WOTXT, CNKI, IEEE: 可信执行环境, 飞地, 实例, 区块链, 加密, 密钥, 状态服务实例, 应用程序实例, 重启, 恢复, TEE, SGX, enclave, application enclave, state enclave, blockchain, encrypt+, key, restart, recover, renew

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2019220603 A1 (INTEL CORP.) 18 July 2019 (2019-07-18) description, paragraphs 78-82, and figures 6-10	1-10
A	CN 114117522 A (SHANGHAI JIAO TONG UNIVERSITY) 01 March 2022 (2022-03-01) entire document	1-10
A	CN 114547656 A (BEIJING UNIVERSITY OF TECHNOLOGY) 27 May 2022 (2022-05-27) entire document	1-10
A	CN 114826684 A (XIDIAN UNIVERSITY) 29 July 2022 (2022-07-29) entire document	1-10
A	WO 2018104326 A1 (ETH ZURICH) 14 June 2018 (2018-06-14) entire document	1-10



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

“A” document defining the general state of the art which is not considered to be of particular relevance

“D” document cited by the applicant in the international application

“E” earlier application or patent but published on or after the international filing date

“L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

“O” document referring to an oral disclosure, use, exhibition or other means

“P” document published prior to the international filing date but later than the priority date claimed

“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

“X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

“Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

“&” document member of the same patent family

Date of the actual completion of the international search

05 May 2023

Date of mailing of the international search report

10 May 2023

Name and mailing address of the ISA/CN

China National Intellectual Property Administration (ISA/
CN)
China No. 6, Xitucheng Road, Jimenqiao, Haidian District,
Beijing 100088

Authorized officer

Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2022/119945

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
US	2019220603	A1	18 July 2019	None			
CN	114117522	A	01 March 2022	None			
CN	114547656	A	27 May 2022	None			
CN	114826684	A	29 July 2022	None			
WO	2018104326	A1	14 June 2018	EP	3333747	A1	13 June 2018

国际检索报告

国际申请号

PCT/CN2022/119945

A. 主题的分类 G06F21/57(2013.01)i;H04L9/28(2006.01)i 按照国际专利分类(IPC)或者同时按照国家分类和IPC两种分类		
B. 检索领域 检索的最低限度文献(标明分类系统和分类号) IPC: G06F; H04L 包含在检索领域中的除最低限度文献以外的检索文献 在国际检索时查阅的电子数据库(数据库的名称, 和使用的检索词(如使用)) VEN, CNTXT, CNABS, USTXT, WOTXT, CNKI, IEEE: 可信执行环境、飞地、实例、区块链、加密、密钥、状态服务实例、应用程序实例、重启、恢复、TEE、SGX、enclave、application enclave、state enclave、blockchain、encrypt+、key、restart、recover、renew		
C. 相关文件		
类 型*	引用文件, 必要时, 指明相关段落	相关的权利要求
A	US 2019220603 A1 (INTEL CORPORATION) 2019年7月18日 (2019 - 07 - 18) 说明书78-82段, 图6-10	1-10
A	CN 114117522 A (上海交通大学) 2022年3月1日 (2022 - 03 - 01) 全文	1-10
A	CN 114547656 A (北京工业大学) 2022年5月27日 (2022 - 05 - 27) 全文	1-10
A	CN 114826684 A (西安电子科技大学) 2022年7月29日 (2022 - 07 - 29) 全文	1-10
A	WO 2018104326 A1 (ETH ZURICH) 2018年6月14日 (2018 - 06 - 14) 全文	1-10
☐ 其余文件在C栏的续页中列出。 ☒ 见同族专利附件。		
* 引用文件的具体类型: “A” 认为不特别相关的表示了现有技术一般状态的文件 “D” 申请人在国际申请中引证的文件 “E” 在国际申请日的当天或之后公布的在先申请或专利 “L” 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件(如具体说明的) “O” 涉及口头公开、使用、展览或其他方式公开的文件 “P” 公布日先于国际申请日但迟于所要求的优先权日的文件 “T” 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件 “X” 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性 “Y” 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性 “&” 同族专利的文件		
国际检索实际完成的日期 2023年5月5日		国际检索报告邮寄日期 2023年5月10日
ISA/CN的名称和邮寄地址 中国国家知识产权局 中国北京市海淀区蓟门桥西土城路6号 100088		授权官员 郝宁 电话号码 (+86) 010-53961326

国际检索报告
关于同族专利的信息

国际申请号

PCT/CN2022/119945

检索报告引用的专利文件			公布日 (年/月/日)	同族专利	公布日 (年/月/日)
US	2019220603	A1	2019年7月18日	无	
CN	114117522	A	2022年3月1日	无	
CN	114547656	A	2022年5月27日	无	
CN	114826684	A	2022年7月29日	无	
WO	2018104326	A1	2018年6月14日	EP 3333747 A1	2018年6月13日