(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)

WIPO | PCT

(10) 国际公布号
WO 2024/051311 A1

(51) 国际专利分类号:
G06F 21/57 (2013.01) *G06F 9/54* (2006.01)

(21) 国际申请号: PCT/CN2023/103965

(22) 国际申请日: 2023 年 6 月 29 日 (29.06.2023)

(25) 申请语言: 中文

(26) 公布语言: 中文

(30) 优先权:
202211091500.1 2022年9月7日 (07.09.2022) CN

(71) 申请人: 华 为 技 术 有 限 公 司 **(HUAWEI TECHNOLOGIES CO., LTD.)** [CN/CN]; 中 国 广 东 省 深 圳 市 龙 岗 区 坂 田 华 为 总 部 办 公 楼, Guangdong 518129 (CN)。

(72) 发明人: 程 琨 **(CHENG, Kun)**; 中 国 广 东 省 深 圳 市 龙 岗 区 坂 田 华 为 总 部 办 公 楼, Guangdong 518129 (CN)。 王 宇 **(WANG, Yu)**; 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。 罗 睿 平 **(LUO, Ruiping)**; 中 国 广 东 省 深 圳 市 龙 岗 区 坂 田 华 为 总 部 办 公 楼, Guangdong 518129 (CN)。 王城**(WANG, Cheng)**; 中 国 广 东 省 深 圳 市 龙 岗 区 坂 田 华 为 总 部 办 公楼, Guangdong 518129 (CN)。

(74) 代理人: 深 圳 中 一 联 合 知 识 产 权 代 理 有 限 公 司 **(SHENZHEN ZHONGYI UNION INTELLECTUAL PROPERTY AGENCY CO., LTD.)**; 中 国 广 东 省 深 圳 市 福 田 区 莲 花 街 道 紫 荆 社 区 深 南 大 道 6008 号 深 圳 特 区 报 业 大 厦 33层, Guangdong 518034 (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, MG, MK, MN, MU, 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。

(54) **Title:** DATA PROCESSING METHOD, TERMINAL DEVICE AND READABLE STORAGE MEDIUM

(54) 发明名称: 数据处理方法、终端设备和可读存储介质

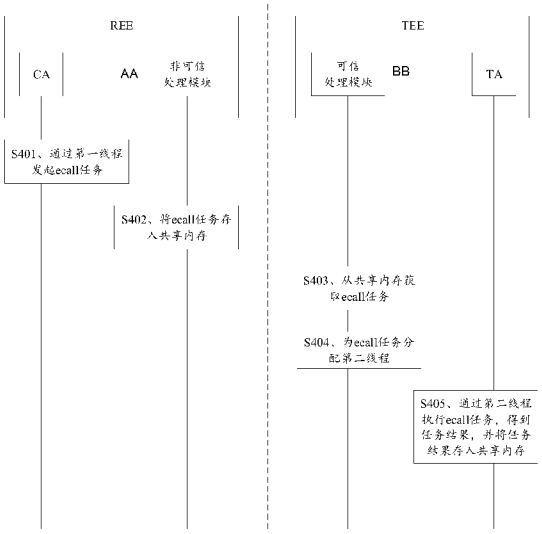

图 4

S401 Initiate an ecall task by means of a first thread
S402 Store the ecall task in a shared memory
S403 Acquire the ecall task from the shared memory
S404 Allocate a second thread for the ecall task
S405 Execute the ecall task by means of the second thread to obtain a task result, and store the task result in the shared memory
AA Untrusted processing module
BB Trusted processing module

(57) Abstract: The embodiments of the present application relate to the technical field of computers. Provided are a data processing method, a terminal device and a readable storage medium. The terminal device deploys an REE and a TEE. The data processing method comprises: in an REE, storing in a shared memory an ecall task initiated by a CA by means of a first thread; and in a TEE, acquiring the ecall task from the shared memory, executing the ecall task by a TA by means of a second thread, and storing a task result in the shared memory. The shared memory is applied in advance for the CA in the REE and can be accessed by the REE and the TEE. The second thread is created in the TEE, and the second thread and the first thread are bound with different processing units. The shared memory is applied in advance, obviating data copying; the threads executing the ecall task in the REE and the TEE are bound with the different processing units, and the second thread is created in the TEE, thus reducing the switching overhead, and increasing data processing efficiency.

WO 2024/051311 A1

[见续页]

(84) 指定国(除另有指明，要求每一种可提供的地区保护): ARIPO (BW, CV, 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, ME, 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: 本申请实施例涉及计算机技术领域，提供一种数据处理方法、终端设备和可读存储介质。终端设备部署REE和TEE。数据处理方法包括：在REE中，将由CA通过第一线程发起的ecall任务存入共享内存；在TEE中，从共享内存获取ecall任务，由TA通过第二线程执行ecall任务，将任务结果存入共享内存。共享内存是在REE中针对CA预先申请的可被REE和TEE访问，第二线程在TEE中被创建，第二线程和第一线程绑定的处理单元不同。通过预先申请共享内存，避免了数据拷贝，通过REE和TEE中执行ecall任务的线程绑定的处理单元不同，且在TEE中创建第二线程，减小了切换开销，提高了数据处理效率。

数据处理方法、终端设备和可读存储介质

本申请要求于2022年09月07日提交国家知识产权局、申请号为202211091500.1、申请名称为“数据处理方法、终端设备和可读存储介质”的中国专利申请的优先权，其全部内容通过引用结合在本申请中。

技术领域

本申请实施例涉及计算机技术领域，尤其涉及一种数据处理方法、终端设备和可读存储介质。

背景技术

随着移动互联网的发展，终端设备的应用越来越广泛。为了保证终端设备中数据的安全性，出现了以 ARM TrustZone 为代表的安全框架。其中，ARM 的含义为高级精简指令集处理器（advanced RISC machines），RISC 的含义为精简指令集计算机（reduced instruction set computer），TrustZone 的含义为信任区。

在 ARM TrustZone 框架下，终端设备的软硬件资源划分到两个世界中，称为安全世界（secure world）和正常世界（normal world）。安全世界对应的执行环境称为可信执行环境（trusted execution environment，TEE），TEE 中运行可信应用（trusted application，TA）。正常世界对应的执行环境称为富执行环境（rich execution environment，REE），REE 中运行普通应用（common application，CA）。REE 和 TEE 相互隔离，TEE 的安全要求更高。

目前，REE 与 TEE 之间进行数据传递的开销很大，数据处理效率很低。

发明内容

本申请实施例提供一种数据处理方法、终端设备和可读存储介质，提高了数据处理效率。

第一方面，提供了一种数据处理方法，应用于终端设备，终端设备部署可信执行环境和非可信执行环境；方法包括：在第一执行环境中，将由第一应用程序通过第一线程发起的任务存入共享内存；任务用于调用第二执行环境中第二应用程序提供的软件功能；共享内存是在非可信执行环境中针对第一应用程序预先申请的且用于非可信执行环境和可信执行环境访问的物理内存区域；第一执行环境为非可信执行环境、第二执行环境为可信执行环境，或者，第一执行环境为可信执行环境、第二执行环境为非可信执行环境；在第二执行环境中，从共享内存获取任务；在第二执行环境中，为任务分配第二线程；第二线程是在第二执行环境中创建的，第二线程和第一线程分别绑定的处理单元不同；在第二执行环境中，由第二应用程序通过第二线程执行任务，得到任务结果，并将任务结果存入共享内存。

第一方面提供的数据处理方法，通过在第一执行环境中针对第一应用程序预先申请共享内存，将执行任务的相关数据等存储在共享内存中，可信执行环境和非可信执行环境通过访问共享内存完成了数据传递，不需要数据的多次拷贝，缩短了可信执行环境和非可信执行环境之间的数据传输路径，提高了数据处理的效率。而且，由于第一应用程序发起任务的第一线程和第二应用程序执行任务的第二线程分别绑定的处理单元不同，使得第一线程绑定的处理单元不需要进行 CPU 模式切换，减小了 CPU 的切换开销，提高了处理效率。而且，由于第二线程本身就在第二执行环境中被创建，使得第二线程绑定的处理单元不需要进行 CPU 模式切换就可以在第二执行环境中执行第一应用程序发起的任务，减小了 CPU 的切换开销，提高了处理效率。

一种可能的实现方式中，在第一执行环境中，将任务存入共享内存，包括：在第一执行环境中，序列化任务，得到任务的序列化消息；在第一执行环境中，将任务的序列化消息存入消息队列；消息队列存储在共享内存中，用于存储第一应用程序发起的多个任务分别对应的序列化消息；在第一执行环境中，将任务的序列化消息的执行标志位设置为第一数值；其中，第一数值用于指

示序列化消息的状态为待执行状态。

在该实现方式中，通过将任务的执行标志位设置为第一数值，指示任务尚未处理，使得第二执行环境可以快速获得尚未处理的任务，提高了处理效率。

一种可能的实现方式中，在第二执行环境中，从共享内存获取任务，包括：在第二执行环境中，访问消息队列，获取执行标志位为第一数值的任务的序列化消息；在第二执行环境中，解析任务的序列化消息，得到任务。

一种可能的实现方式中，在第二执行环境中，由第二应用程序通过第二线程执行任务，得到任务结果，并将任务结果存入共享内存之后，还包括：在第二执行环境中，将任务的序列化消息的执行标志位设置为第二数值；其中，第二数值用于指示序列化消息的状态为已执行状态。

在该实现方式中，通过将任务的执行标志位设置为第二数值，指示任务已经处理，使得第一执行环境可以快速得知任务的状态，提高了处理效率。

一种可能的实现方式中，在第一执行环境中，将由第一应用程序通过第一线程发起的任务存入共享内存之前，还包括：在第一执行环境中，由第一应用程序通过第一线程发起任务，并注册回调函数。

在该实现方式中，通过注册回调函数，第一应用程序发起任务后，不需要等待任务结果，而是可以通过第一线程进行后续应用业务逻辑的处理。缩短了第一应用程序发起任务后的等待时间，解决了第二执行环境执行任务阻塞第一应用程序业务逻辑执行的问题。

一种可能的实现方式中，方法还包括：在第一执行环境中，通过第三线程调用回调函数；第三线程和第一线程不同。

在该实现方式中，通过第三线程调用回调函数，可以自动完成对 TEE 侧返回的传出参数和函数返回值的处理。

一种可能的实现方式中，第三线程的数量小于或等于第一应用程序发起的任务的数量。

一种可能的实现方式中，任务包括传入参数和传出参数，传入参数和传出参数的存储位置位于共享内存中；在第一执行环境中，将由第一应用程序通过第一线程发起的任务存入共享内存之前，还包括：在第一执行环境中，获取共享内存在第一执行环境中的第一基准地址、传入参数在第一执行环境中的第一输入地址和传出参数在第一执行环境中的第一输出地址；在第一执行环境中，将第一输入地址转换为相对于第一基准地址的第一输入偏移量，将第一输出地址转换为相对于第一基准地址的第一输出偏移量。

一种可能的实现方式中，在第二执行环境中，由第二应用程序通过第二线程执行任务之前，还包括：在第二执行环境中，获取共享内存在第二执行环境中的第二基准地址；在第二执行环境中，根据第二基准地址和第一输入偏移量获取传入参数在第二执行环境中的第二输入地址，根据第二基准地址和第一输出偏移量获取传出参数在第二执行环境中的第二输出地址。

一种可能的实现方式中，第一执行环境为可信执行环境，第二执行环境为非可信执行环境，任务为 ocall 任务。

一种可能的实现方式中，第一执行环境为非可信执行环境，第二执行环境为可信执行环境，任务为 ecall 任务。

一种可能的实现方式中，在第二执行环境中针对第一应用程序建立有线程池，线程池中包括多个线程，线程池中的线程是在第二执行环境中创建的；第二线程为线程池中的空闲线程。

一种可能的实现方式中，方法还包括：在第二执行环境中，获取线程池中空闲线程的第一数量；在第二执行环境中，获取第一应用程序发起的至少一个任务中未执行的任务的第二数量；在第二执行环境中，根据第一数量和第二数量，调整线程池中线程的数量。

在该实现方式中，根据线程池中空闲线程的第一数量和尚未处理的任务的第二数量，实时调整线程池中线程的数量，避免了 CPU 浪费，提高了数据处理效率。

一种可能的实现方式中，在第二执行环境中，根据第一数量和第二数量，调整线程池中线程的数量，包括：若第一数量大于第二数量，且线程池中线程的总数大于最小阈值，则在第二执行环境中，在线程池中销毁一个线程；或者，若第一数量小于第二数量，且线程池中线程的总数小于最大阈值，则在第二执行环境中，在线程池中创建至少一个线程；至少一个线程的数量小于或

等于最大阈值与线程池中线程的总数之差。

一种可能的实现方式中，在第一执行环境中，将由第一应用程序通过第一线程发起的任务存入共享内存之前，还包括：在第一执行环境中，向第二执行环境发送最小阈值、最大阈值和中间阈值；中间阈值大于最小阈值且小于最大阈值；在第二执行环境中，针对第一应用程序建立线程池，线程池中线程的数量为中间阈值。

一种可能的实现方式中，在第一执行环境中，将由第一应用程序通过第一线程发起的任务存入共享内存之前，还包括：在第一执行环境中，获取共享内存的内存大小；在第一执行环境中，根据内存大小分配共享内存；在第一执行环境中，向第二执行环境发送共享内存的相关信息。

一种可能的实现方式中，内存大小包括消息队列占用的第一内存大小和应用数据占用的第二内存大小；消息队列存储在共享内存中，用于存储第一应用程序发起的多个任务分别对应的序列化消息；共享内存包括第一内存大小的第一内存空间和第二内存大小的第二内存空间。

一种可能的实现方式中，第一内存空间和第二内存空间的地址连续。

第二方面，提供了一种终端设备，终端设备部署可信执行环境和非可信执行环境。可信执行环境部署可信应用程序和可信处理模块。非可信执行环境部署非可信应用程序和非可信处理模块。第一执行环境中的第一处理模块，用于将第一应用程序通过第一线程发起的任务存入共享内存；任务用于调用第二执行环境中第二应用程序提供的软件功能；共享内存是在非可信执行环境中针对第一应用程序预先申请的且用于非可信执行环境和可信执行环境访问的物理内存区域；第一执行环境为非可信执行环境、第二执行环境为可信执行环境、第一处理模块为非可信处理模块、第二处理模块为可信处理模块，或者，第一执行环境为可信执行环境、第二执行环境为非可信执行环境、第一处理模块为可信处理模块、第二处理模块为非可信处理模块；第二执行环境中的第二处理模块，用于从共享内存获取任务；第二执行环境中的第二处理模块，用于为任务分配第二线程；第二线程是在第二执行环境中创建的，第二线程和第一线程分别绑定的处理单元不同；第二执行环境中的第二应用程序，用于通过第二线程执行任务，得到任务结果，并将任务结果存入共享内存。

一种可能的实现方式中，第一执行环境中的第一处理模块，用于：在第一执行环境中，序列化任务，得到任务的序列化消息；在第一执行环境中，将任务的序列化消息存入消息队列；消息队列存储在共享内存中，用于存储第一应用程序发起的多个任务分别对应的序列化消息；在第一执行环境中，将任务的序列化消息的执行标志位设置为第一数值；其中，第一数值用于指示序列化消息的状态为待执行状态。

一种可能的实现方式中，第二执行环境中的第二处理模块，用于：在第二执行环境中，访问消息队列，获取执行标志位为第一数值的任务的序列化消息；在第二执行环境中，解析任务的序列化消息，得到任务。

一种可能的实现方式中，第二执行环境中的第二处理模块，还用于：在第二应用程序通过第二线程执行任务，得到任务结果，并将任务结果存入共享内存之后，在第二执行环境中，将任务的序列化消息的执行标志位设置为第二数值；其中，第二数值用于指示序列化消息的状态为已执行状态。

一种可能的实现方式中，第一执行环境中的第一应用程序，用于：在第一执行环境中，通过第一线程发起任务，并注册回调函数。

一种可能的实现方式中，第一执行环境中的第一处理模块，还用于：在第一执行环境中，通过第三线程调用回调函数；第三线程和第一线程不同。

一种可能的实现方式中，第三线程的数量小于或等于第一应用程序发起的任务的数量。

一种可能的实现方式中，任务包括传入参数和传出参数，传入参数和传出参数的存储位置位于共享内存中；第一执行环境中的第一处理模块，还用于：在将第一应用程序通过第一线程发起的任务存入共享内存之前，在第一执行环境中，获取共享内存在第一执行环境中的第一基准地址、传入参数在第一执行环境中的第一输入地址和传出参数在第一执行环境中的第一输出地址；在第一执行环境中，将第一输入地址转换为相对于第一基准地址的第一输入偏移量，将第一输出地址转换为相对于第一基准地址的第一输出偏移量。

一种可能的实现方式中，第二执行环境中的第二处理模块，还用于：在第二应用程序通过第二线程执行任务之前，在第二执行环境中，获取共享内存在第二执行环境中的第二基准地址；在第二执行环境中，根据第二基准地址和第一输入偏移量获取传入参数在第二执行环境中的第二输入地址，根据第二基准地址和第一输出偏移量获取传出参数在第二执行环境中的第二输出地址。

一种可能的实现方式中，第一执行环境为可信执行环境，第二执行环境为非可信执行环境，任务为 ocall 任务。

一种可能的实现方式中，第一执行环境为非可信执行环境，第二执行环境为可信执行环境，任务为 ecall 任务。

一种可能的实现方式中，在第二执行环境中针对第一应用程序建立有线程池，线程池中包括多个线程，线程池中的线程是在第二执行环境中创建的；第二线程为线程池中的空闲线程。

一种可能的实现方式中，第二执行环境中的第二处理模块，还用于：在第二执行环境中，获取线程池中空闲线程的第一数量；在第二执行环境中，获取第一应用程序发起的至少一个任务中未执行的任务的第二数量；在第二执行环境中，根据第一数量和第二数量，调整线程池中线程的数量。

一种可能的实现方式中，第二执行环境中的第二处理模块，用于：若第一数量大于第二数量，且线程池中线程的总数大于最小阈值，则在第二执行环境中，在线程池中销毁一个线程；或者，若第一数量小于第二数量，且线程池中线程的总数小于最大阈值，则在第二执行环境中，在线程池中创建至少一个线程；至少一个线程的数量小于或等于最大阈值与线程池中线程的总数之差。

一种可能的实现方式中，第一执行环境中的第一处理模块，还用于：在将第一应用程序通过第一线程发起的任务存入共享内存之前，在第一执行环境中，向第二执行环境发送最小阈值、最大阈值和中间阈值；中间阈值大于最小阈值且小于最大阈值；第二执行环境中的第二处理模块，还用于：在第二执行环境中，针对第一应用程序建立线程池，线程池中线程的数量为中间阈值。

一种可能的实现方式中，第一执行环境中的第一处理模块，还用于：在将第一应用程序通过第一线程发起的任务存入共享内存之前，在第一执行环境中，获取共享内存的内存大小；在第一执行环境中，根据内存大小分配共享内存；在第一执行环境中，向第二执行环境发送共享内存的相关信息。

一种可能的实现方式中，内存大小包括消息队列占用的第一内存大小和应用数据占用的第二内存大小；消息队列存储在共享内存中，用于存储第一应用程序发起的多个任务分别对应的序列化消息；共享内存包括第一内存大小的第一内存空间和第二内存大小的第二内存空间。

一种可能的实现方式中，第一内存空间和第二内存空间的地址连续。

第三方面，提供一种终端设备，包括处理器，处理器用于与存储器耦合，并读取存储器中的指令并根据指令使得终端设备执行第一方面提供的方法。

第四方面，提供一种程序，该程序在被处理器执行时用于执行第一方面提供的方法。

第五方面，提供一种计算机可读存储介质，计算机可读存储介质中存储有指令，当指令在计算机或处理器上运行时，实现第一方面提供的方法。

第六方面，提供一种程序产品，所述程序产品包括计算机程序，所述计算机程序存储在可读存储介质中，设备的至少一个处理器可以从所述可读存储介质读取所述计算机程序，所述至少一个处理器执行所述计算机程序使得该设备实施第一方面提供的方法。

附图说明

图 1A~图 1B 为本申请实施例提供的终端设备的一组架构示意图；

图 1C 为本申请实施例提供的终端设备的另一种架构示意图；

图 2 为本申请实施例提供的终端设备的又一种架构示意图；

图 3A~图 3D 为本申请实施例提供的共享内存的一组示意图；

图 4 为本申请实施例提供的数据处理方法的一种流程图；

图 5 为本申请实施例提供的数据处理方法的另一种流程图；

图 6 为本申请实施例提供的数据处理方法的又一种流程图；

图 7 为本申请实施例提供的数据处理方法的又一种流程图；

图 8 为本申请实施例提供的终端设备的一种结构示意图。

具体实施方式

下面结合附图描述本申请实施例。

本申请实施例提供的数据处理方法，应用于终端设备。本申请实施例对终端设备的名称不做限定，例如，终端设备也称为：终端、电子设备、计算机设备、计算设备或计算机系统等。常见的终端设备例如包括：手机、平板电脑、笔记本电脑、台式电脑、服务器、可穿戴设备（例如智能手表或智能手环）、智能家庭设备（例如智能音响或智能电视）、车载智能设备、无人驾驶设备、虚拟现实（virtual reality，VR）设备、增强现实（augmented reality，AR）设备、混合现实（mix reality，MR）设备以及人工智能（artificial intelligence，AI）设备等。

首先，为了方便理解本申请，对相关概念进行说明。

1.安全世界、非安全世界

终端设备上运行有操作系统（operating system，OS），OS 上部署有应用程序（application，APP），OS 和 APP 为终端设备提供了丰富的功能。随着移动互联网的发展，APP 的类型和数量越来越多，终端设备支持的功能也越来越多，对用户隐私和信息安全提出了更高的要求。

为了提高终端设备运行的安全性，可以将终端设备的软硬件资源划分到两个世界中，称为安全世界和非安全世界。安全世界和非安全世界为相对概念。在安全世界中执行需要保密的操作，存储需要保密的数据，例如，进行指纹识别、人脸识别、密码处理、数据加解密、安全认证等，而其余操作在非安全世界中执行。

需要说明，本申请实施例对安全世界、非安全世界的名称不做限定。例如，安全世界也称为安全域。非安全世界也称为非安全域、正常世界（normal world）或普通世界（common world）。在终端设备不同的体系架构中，安全世界、非安全世界的名称可能不同。

2.可信执行环境、非可信执行环境、可信应用程序、非可信应用程序

安全世界对应的执行环境称为可信执行环境，非安全世界对应的执行环境称为非可信执行环境。可信执行环境和非可信执行环境为相对概念。可信执行环境和非可信执行环境相互隔离，从而，可以在可信执行环境中执行需要保密的操作，存储需要保密的数据，保护关键程序逻辑、高价值数据等。

可信执行环境中部署有可信应用程序，非可信执行环境中部署有非可信应用程序。可信应用程序和非可信应用程序为相对概念。比如，可信应用程序包括但不限于指纹识别程序、人脸识别程序等。

需要说明，本申请实施例对可信执行环境、非可信执行环境、可信执行程序、非可信执行程序的名称不做限定。在终端设备不同的体系架构中，这些概念的名称可能不同，可以参见后续关于 GP 架构和 Intel SGX 架构的描述。

3.ecall 任务、ocall 任务

在非可信执行环境中发起的、用于调用可信执行环境函数接口的行为，称为入口调用（entry call，ecall）任务或 ecall 调用，被调用的可信执行环境的函数称为 ecall 函数。

相对应的，在可信执行环境中发起的、用于调用非可信执行环境函数接口的行为，称为出口调用（out call，ocall）任务或 ocall 调用，被调用的非可信执行环境的函数称为 ocall 函数。

4.程序、应用程序、进程、线程

程序（program），通常指一段可执行的代码。

应用程序（application），包括多个程序，可以提供特定的功能。例如，指纹识别应用程序提供指纹识别功能，人脸识别应用程序提供人脸识别功能。

进程（process），指程序的一次运行活动，是操作系统进行资源分配的基本单位。可以理解为，一个运行起来的程序就是一个进程。进程具有自己的地址空间。一个进程可以包括一个线程，

此时，进程是单线程的。一个进程也可以包括多个线程，多个线程可以并行执行不同的任务，此时，进程是多线程的。

线程（thread），包含在进程之中，是操作系统进行运算调度的基本单位。可以理解为，一条线程是进程中一个单一顺序的控制流。线程有时也称为轻量级进程（light weight process，LWP）。

5.CPU 绑定

中央处理器（central processing unit，CPU）绑定，是指在多核 CPU 的系统中将进程或线程绑定到指定的 CPU 核上去执行。

通过 CPU 绑定，确保了进程或线程在指定的 CPU 核上运行，避免了操作系统在多个 CPU 核之间频繁调度切换应用程序而导致的 CPU 缓存失效、降低缓存命中率的问题，提高了应用程序对 CPU 的使用效率。

6.GP、ARM TrustZone、TEE、REE、TA、CA

全球平台组织（GlobalPlatform，GP）制定了可信执行环境的开放架构和应用编程接口（application programming interface，API）标准。在 GP 架构中，非安全世界称为普通世界或正常世界。安全世界对应的可信执行环境称为 TEE，非安全世界对应的非可信执行环境称为 REE。TEE 中运行的可信应用程序称为 TA，REE 中运行的非可信应用程序称为 CA。

其中，ARM TrustZone 是基于 GP 标准开发的一种架构。ARM TrustZone 基于硬件的安全功能，在处理器层次引入了安全世界和非安全世界，任何时刻处理器仅在其中的一个世界内运行。处理器运行在安全世界时，运行安全世界的代码；处理器运行在非安全世界时，运行非安全世界的代码。安全世界和非安全世界通过监控模式（Monitor Mode）进行转换。安全世界和非安全世界的硬件隔离，为保护应用程序的代码和数据提供了有效机制。

示例性的，图 1A 为本申请实施例提供的终端设备的一种架构示意图，终端设备采用 GP 架构或 ARM TrustZone 架构。如图 1A 所示，终端设备包括硬件层和软件层。硬件层包括但不限于：处理器和存储介质。软件层包括 REE 和 TEE。REE 侧部署有 REE 操作系统和 CA，REE 操作系统也称为富操作系统（rich operating system，Rich OS），比如，现有的 Linux 操作系统或 Windows 操作系统。TEE 侧部署有 TEE 操作系统和 TA，TEE 操作系统也称为可信操作系统（trusted operating system，Trusted OS）。

REE 侧和 TEE 侧均包括应用态（或用户态）和内核态。REE 侧的内核态包括 REE 操作系统，REE 侧的应用态包括 CA。TEE 侧的内核态包括 TEE 操作系统，TEE 侧的应用态包括 TA。

CA 可以发起 ecall 任务，调用 TEE 侧提供的 API 来完成某些安全的、受保护的业务逻辑。相反的，TA 也可以发起 ocall 任务，调用 REE 侧提供的 API 来完成一些通常的业务逻辑。示例性的，一些应用场景为：在终端设备解锁、应用程序登陆、金融支付等场景中，CA 为解锁应用或金融支付类应用，TA 为指纹识别应用或者人脸识别应用。CA 发起 ecall 任务，调用 TEE 侧的 TA 提供的指纹识别或者人脸识别功能，CA 和 TA 二者协同，完成终端设备的解锁、应用程序的登陆、指纹支付或者人脸支付。

其中，REE 和 TEE 运行在各自独立的物理内存上，即内存隔离。REE 侧不能访问 TEE 侧的物理内存，反过来，TEE 侧不能访问 REE 侧的物理内存。比如，在图 1A 中，REE 侧对应的物理内存称为 REE 内存，TEE 侧对应的物理内存称为 TEE 内存。

其中，处理器包括多个处理器核心，标记为核心 0~核心 n。任何时刻，处理器核心运行在 REE 侧或者运行在 TEE 侧。比如，核心 0 在 REE 中运行，处于 REE 模式或 REE 状态；核心 1 在 TEE 中运行，处于 TEE 模式或 TEE 状态。

需要说明，处理器包括多个处理单元。本申请实施例对处理器的类型和名称，处理单元的类型、名称和数量不作限定。例如，处理器为 CPU，处理单元为 CPU 核心（或 CPU 核、CPU 内核）。CPU 核心是 CPU 的重要组成部分，用来完成计算、接受或存储命令、处理数据等。

存储介质，也称为存储器，用于存储可执行的程序代码和/或数据，可执行的程序代码可以包括指令。存储介质可以包括存储程序区和存储数据区。存储介质可以包括高速随机存取存储器和非易失性存储器，例如，至少一个磁盘存储器件，硬盘（hard disk drive，HDD），固态硬盘（solid-state drive，SS），闪存器件，通用闪存存储器（universal flash storage，UFS）等，还可以包括易失性

存储器（volatile memory），例如随机存取存储器（random- access memory，RAM）等。

示例性的，参见图 1B。当 REE 侧 CA 发起 ecall 任务时，CA 和 TA 之间通过 REE 通信代理、硬件层的消息通道以及 TEE 通信代理交互数据。REE 通信代理、消息通道以及 TEE 通信代理为 CA 和 TA 建立了安全的通信通道。具体的，CA 调用 TEE 客户端 API 来和对应的 TA 实现通信，TA 调用 TEE 内部 API 来使用 TEE 提供的编程资源实现相关功能。

但是，由于 REE 与 TEE 的内存隔离，导致数据传递必须依赖操作系统与内存进行拷贝传递，使得 CA 和 TA 之间的数据传递经历多次拷贝，交互开销较大。而且，发起 ecall 任务的 CA 所在的处理器核心（比如，核心 0）要从 REE 模式切换到 TEE 模式，在 TEE 中执行 TA 中的 ecall 函数，执行完毕后，核心 0 再从 TEE 模式切换回 REE 模式，在 REE 中继续执行 CA 应用，CPU 切换的开销较大，数据处理效率很低。

7.Intel（因特尔）软件保护执行模式（software guard eXecution，SGX）架构

Intel SGX 是一种满足可信计算需求的技术，它与 ARM TrustZone 类似，可以实现可信执行环境。

示例性的，图 1C 为本申请实施例提供的终端设备的另一种架构示意图，终端设备采用 Intel SGX 架构或 SGX 架构，示出了软件层。如图 1C 所示，非可信执行环境包括操作系统和应用程序。操作系统比如为 Linux 操作系统或 Windows 操作系统。应用程序可以分为两个部分：安全部分（或可信部分、可信应用程序）和非安全部分（或不可信部分、非可信应用程序）。当应用程序需要处理机密数据时，它会创建一个位于可信内存的安全区（称为 enclave，或可信区域、可信执行环境），用来存放 enclave 代码和 enclave 数据，然后调用可信函数（也称为 enclave 函数，它是由开发者创建的、专门在安全区内执行的函数）。一旦可信函数被调用，应用程序就会在可信区域执行并明文访问安全区内的 enclave 代码和 enclave 数据，除此之外，试图从安全区外访问安全区内的行为都会被处理器拒绝。当可信函数执行完毕后，安全区内的 enclave 数据依然保留在可信内存中，而应用程序返回到不可信区域（或非可信执行环境）继续执行，并且失去了对可信内存的访问权限。

可见，在 SGX 架构中，可信执行环境是宿主进程的一部分，这意味着，应用程序包含自己的代码（称为应用代码）、数据（称为应用数据）和 enclave，参见图 1C 中左侧部分。而 enclave 也包含自己的代码（称为 enclave 代码）和自己的数据（称为 enclave 数据），参见图 1C 中右侧部分。SGX 可以保护 enclave 代码和 enclave 数据的机密性和完整性。

为了方便说明，本申请实施例以 GP 架构作为示例，对本申请实施例提供的数据处理方法进行说明。

本申请实施例在图 1A 和图 1B 所示 GP 架构的基础上，提供了终端设备的另一种架构图。示例性的，图 2 为本申请实施例提供的终端设备的又一种架构示意图。在图 2 中，存储介质中增加了共享内存，REE 的应用态增加了非可信处理模块，TEE 的应用态增加了可信处理模块。其中，非可信处理模块用于在 REE 中处理 REE 或 TEE 中应用程序发起任务过程中的相关数据。可信处理模块用于在 TEE 中处理 REE 或 TEE 中应用程序发起任务过程中的相关数据。

基于图 2 所示的 GP 架构，本申请实施例提供的数据处理方法，通过预先申请共享内存，REE 侧和 TEE 侧可以访问共享内存中的数据，完成了 REE 侧和 TEE 侧之间的数据传递，不需要数据的多次拷贝，缩短了 REE 和 TEE 之间数据传输的路径，提高了数据处理的效率。而且，REE 侧和 TEE 侧中执行同一任务的线程绑定的处理单元不同，TEE 侧中执行任务的线程在 TEE 侧中被创建，减小了 CPU 的切换开销，提高了数据处理效率。

下面，对本申请实施例中的概念进行说明。

1.第一执行环境、第一应用程序、第二执行环境、第二应用程序

在本申请实施例中，第一执行环境为非可信执行环境、第二执行环境为可信执行环境，或者，第一执行环境为可信执行环境、第二执行环境为非可信执行环境。

第一应用程序为第一执行环境中的应用程序，第二应用程序为第二执行环境中的应用程序。

2.任务

在本申请实施例中，任务是指在第一执行环境中由第一应用程序通过第一线程发起的、用于调用第二执行环境中第二应用程序提供的软件功能。

举例说明。假设，第一执行环境为 REE，第一应用程序为 CA，第二执行环境为 TEE，第二应用程序为 TA。在一个场景中，用户使用支付类 APP 通过指纹识别进行支付。在该场景中，CA 为支付类 APP，TA 为指纹识别应用程序。在 REE 中，CA（支付类 APP）运行的过程中，发起 ecall 任务，调用 TEE 中的 TA（指纹识别应用程序）提供的指纹识别功能，在支付类 APP 中完成指纹支付。

可选的，任务包括 ecall 任务和 ocall 任务。

3.共享内存、第一内存空间、第二内存空间

在本申请实施例中，共享内存是指，在非可信执行环境中针对第一应用程序预先申请的、且用于非可信执行环境和可信执行环境访问的内存区域。

举例说明。假设，第一执行环境为 REE，第一应用程序为 CA，CA 发起 ecall 任务。通常，终端设备的物理内存管理由非可信执行环境执行。在 REE 中，针对 CA 预先申请一块物理内存区域，REE 和 TEE 均可以访问该物理内存区域。那么，当 CA 发起 ecall 任务时，可以将 ecall 任务的相关数据存入共享内存，从而，TEE 侧可以从共享内存获取该 ecall 任务的相关数据，之后执行该 ecall 任务。相应的，在 TEE 侧执行 ecall 任务结束后，可以将 ecall 任务的任务结果存储在共享内存中，从而，REE 侧可以从共享内存中获取该 ecall 任务的任务结果，完成数据传输。由于不需要数据拷贝的情况下完成了数据的传输，因此提高了数据处理的效率。

其中，共享内存中存储有任务调用的函数的传入参数和传出参数。

可选的，共享内存可以为，在非可信执行环境中针对第一应用程序发起的任务预先申请的、且用于非可信执行环境和可信执行环境访问的物理内存区域。

在该实现方式中，由于共享内存的内存大小用于执行第一应用程序发起的任务，根据第一应用程序发起的任务预先申请共享内存，提高了预先申请共享内存的合理性和准确性，提高了内存资源的利用率。

需要说明，本申请实施例对共享内存的内存大小和位置不作限定。当第一应用程序不同时，共享内存的内存大小和位置可以不同。比如，第一应用程序包括 CA1 和 CA2。CA1 的共享内存大小为 5MB，起始地址为 0x40000000。CA2 的共享内存大小为 1MB，起始地址为 0x50000000。

可选的，共享内存的内存大小包括消息队列占用的第一内存大小和应用数据占用的第二内存大小。其中，消息队列用于存储第一应用程序发起的多个任务分别对应的序列化消息。应用数据包括执行第一应用程序发起的任务的过程中的交互数据。本申请实施例对应用数据的名称不作限定，例如，也可以称为交互数据、任务数据或共享数据等。

可选的，第一内存大小可以根据第一应用程序发起的任务的总数量进行估算。举例说明。假设，CA 发起 100 个任务，每个任务在消息队列中占用的内存大小为 0.05MB，那么，第一内存大小可以为 100*0.05MB=5MB。可选的，第一内存大小还可以包括防溢容量值。比如，防溢容量值为 1MB，那么，CA 发起 100 个任务时，第一内存大小为：100*0.05MB+1MB=6MB。

可选的，如图 2 所示，共享内存包括第一内存空间和第二内存空间。第一内存空间的大小为第一内存大小，用于存储消息队列。第二内存空间的大小为第二内存大小，用于存储应用数据。本申请实施例对第一内存空间和第二内存空间的名称不作限定。例如，第二内存空间也称为数据缓冲区。

本申请实施例对第一内存大小和第二内存大小的取值不作限定，对第一内存空间和第二内存空间的起始位置不作限定。

可选的，为了便于内存管理，第一内存空间和第二内存空间的地址连续。

可选的，第一应用程序声明初始化配置时，可以包括下列中的至少一项：共享内存的内存大小、第一内存大小、第二内存大小和第一应用程序发起的任务的数量。在非可信执行环境中，调用非可信操作系统提供的 API 可以申请分配共享内存。

下面结合图 3A~图 3D，对共享内存进行示例性说明。假设，第一应用程序为 CA。

可选的，在一种实现方式中，如图 3A 所示，在 REE 中，根据共享内存的内存大小申请一块

地址连续的物理内存区域，作为 CA 的共享内存。

可选的，在另一种实现方式中，如图 3B 所示，在 REE 中，根据第一内存大小申请一块地址连续的物理内存区域作为 CA 共享内存中的第一内存空间，用于存储消息队列；根据第二内存大小申请一块地址连续的物理内存区域作为 CA 共享内存中的第二内存空间，用于存储应用数据。其中，第一内存空间和第二内存空间的地址连续，且第一内存空间位于第二内存空间的地址之后。

可选的，在又一种实现方式中，如图 3C 所示，根据第一内存大小和第二内存大小分别申请第一内存空间和第二内存空间。图 3C 与图 3B 的相同点在于第一内存空间和第二内存空间的地址连续，区别在于，在图 3C 中，第一内存空间位于第二内存空间的地址之前。

可选的，在又一种实现方式中，如图 3D 所示，根据第一内存大小和第二内存大小分别申请第一内存空间和第二内存空间。图 3D 与图 3B、图 3C 的区别在于：在图 3D 中，第一内存空间和第二内存空间的地址不连续。

4.消息队列、序列化、反序列化

消息队列存储在共享内存中，用于存储第一应用程序发起的多个任务分别对应的序列化消息。

本申请实施例对消息队列可以存储的任务的数量或者存储的序列化消息的数量不作限定，应用程序不同时，应用程序发起的任务的数量通常不同，消息队列的容量不同。

其中，序列化是一种数据处理机制。计算设备执行输入/输出（input/output，IO）操作时，数据要以流的形式进行读写。序列化可以将对象转换成计算设备可以识别的字节流。通过序列化，可以更方面的进行数据的存储和传输。

与序列化相反的操作称为反序列化，是指将接收到的字节流还原成程序能识别的对象。

在本申请实施例中，任务存入消息队列之前，需要对任务进行序列化，生成任务的序列化消息。相应的，从消息队列中取出任务对应的序列化消息后，需要对任务对应的序列化消息进行反序列化或者进行解析，从而恢复出任务。

5.线程池、最小阈值、中间阈值、最大阈值

在本申请实施例中，在可信执行环境中，针对非可信执行环境中的非可信应用程序建立有线程池。线程池中包括多个线程，线程池中的线程均是在可信执行环境中创建的。线程池中的线程用于执行非可信应用程序发起的 ecall 任务。

可选的，线程池中线程的数量具有最小值和最大值，分别称为最小阈值和最大阈值。本申请实施例对最小阈值和最大阈值的数量不作限定。通过最小阈值，确保了在可信执行环境中存在线程用于执行非可信应用程序发起的 ecall 任务，避免了 ecall 任务无法执行的情况。通过最大阈值，限制了在可信执行环境中执行非可信应用程序发起的 ecall 任务的线程最大值，避免了因为 ecall 任务较少而造成的资源浪费。

可选的，线程池中线程的数量可以根据非可信应用程序发起的 ecall 任务的数量在最小阈值和最大阈值之间实时调整。通过线程池中线程数量的实时调整，避免了由于非可信应用程序发起的 ecall 任务较多而造成的没有线程可用的情况，也避免了由于非可信应用程序发起的 ecall 任务较少而造成的资源浪费，提高了可信执行环境侧的资源利用率，提高了 ecall 任务的处理效率。

可选的，在可信执行环境中初始建立线程池时，线程池中线程的数量可以为中间阈值，其中，最小阈值<中间阈值<最大阈值。

可选的，最小阈值、最大阈值和中间阈值中的至少一项，可以为可信执行环境侧设置的，或者，为非可信执行环境侧设置的，或者，为可信执行环境侧和非可信执行环境侧相互协商后确定的。

可选的，非可信执行环境中的非可信应用程序不同时，线程池对应的最小阈值、最大阈值和中间阈值中的至少一项，可以相同，也可以不同。

下面举例说明。

假设，非可信应用程序包括 CA1 和 CA2。在可信执行环境中，针对 CA1 建立线程池 1，对应的最小阈值、最大阈值和中间阈值分别为 2、5 和 10。初始建立线程时，针对 CA1 建立 5 个线程。针对 CA2 建立线程池 2，对应的最小阈值、最大阈值和中间阈值分别为 3、6 和 10。初始建立线程时，针对 CA2 建立 6 个线程。

6.回调函数

回调函数是指一个被作为参数传递的函数。

使用回调函数的过程可以包括：（1）定义一个回调函数；（2）提供函数实现的一方在初始化的时候，将回调函数的函数指针注册给调用者；（3）当特定事件或条件满足时，调用者使用函数指针调用回调函数对事件进行处理。

通过回调函数，可以把需要调用的函数的指针作为参数传递给一个函数，以便该函数在处理相似事件的时候可以灵活的使用不同的方法。

下面通过具体的实施例对本申请的技术方案进行详细说明。下面的实施例可以相互结合，对于相同或相似的概念或过程可能在某些实施例中不再赘述。

本申请实施例中的术语“第一”、“第二”、“第三”、“第四”等（如果存在）是用于区别类似的对象，而不必用于描述特定的顺序或先后次序。

在本申请的一个实施例中，提供了一种数据处理方法，执行主体为终端设备。终端设备中部署有可信执行环境和非可信执行环境。本实施例对终端设备实现可信执行环境的方式不作限定。比如，终端设备可以采用 GP 架构，也可以采用 SGX 架构。可信执行环境和非可信执行环境中均部署有应用程序和处理模块。其中，可信执行环境中部署的应用程序称为可信应用程序，处理模块称为可信处理模块。非可信执行环境中部署的应用程序称为非可信应用程序，处理模块称为非可信处理模块。

本实施例提供的数据处理方法，适用于终端设备在第一执行环境中由第一应用程序通过第一线程发起任务的场景。为了方便说明，第一执行环境中部署的应用程序称为第一应用程序，部署的处理模块称为第一处理模块。第二执行环境中部署的应用程序称为第二应用程序，部署的处理模块称为第二处理模块。

可选的，在一种实现方式中，第一执行环境为非可信执行环境、第一应用程序为非可信应用程序、第二执行环境为可信执行环境、第二应用程序为可信应用程序、任务为 ecall 任务。

可选的，在另一种实现方式中，第一执行环境为可信执行环境、第一应用程序为可信应用程序、第二执行环境为非可信执行环境、第二应用程序为非可信应用程序、任务为 ocall 任务。

可以理解，终端设备采用的架构不同时，第一执行环境和第二执行环境不同时，本实施例提供的数据处理方法，至少可以应用于如下场景：

场景一：终端设备采用 GP 架构，可以参见图 2 所示结构。第一执行环境为 REE、第一应用程序为 CA、第一处理模块为非可信处理模块、第二执行环境为 TEE、第二应用程序为 TA、第二处理模块为可信处理模块。终端设备在 REE 中由 CA 通过第一线程发起 ecall 任务。

场景二：终端设备采用 GP 架构，可以参见图 2 所示结构。第一执行环境为 TEE、第一应用程序为 TA、第一处理模块为可信处理模块、第二执行环境为 REE、第二应用程序为 CA、第二处理模块为非可信处理模块。终端设备在 TEE 中由 TA 通过第一线程发起 ocall 任务。

场景三：终端设备采用 SGX 架构，可以参见图 1C 所示结构。第一执行环境为非可信执行环境、第一应用程序为应用程序的非安全部分、第一处理模块为非可信处理模块、第二执行环境为 Enclave、第二应用程序为应用程序的安全部分、第二处理模块为可信处理模块。终端设备在非可信执行环境中由应用程序的非安全部分通过第一线程发起 ecall 任务。

场景四：终端设备采用 SGX 架构，可以参见图 1C 所示结构。第一执行环境为 Enclave、第一应用程序为应用程序的安全部分、第一处理模块为可信处理模块，第二执行环境为非可信执行环境、第二应用程序为应用程序的非安全部分、第二处理模块为非可信处理模块。终端设备在 Enclave 中由应用程序的安全部分通过第一线程发起 ocall 任务。

为了方便说明，本实施例以场景一作为示例，对本实施例提供的数据处理方法进行说明。

示例性的，图 4 为本申请实施例提供的数据处理方法的一种流程图。如图 4 所示，本实施例提供的数据处理方法，可以包括：

S401、在 REE 中，CA 通过第一线程发起 ecall 任务。

具体的，CA 发起 ecall 任务，调用 TEE 中 TA 提供的软件功能。第一线程是 CA 在 REE 中创

建的线程，绑定有 CPU 核。本实施例对第一线程绑定的 CPU 核不作限定。比如，CPU 包括核心 0~核心 31，第一线程绑定的 CPU 核是核心 0。

ecall 任务可以包括但不限于下列信息：调用的 ecall 函数的函数名、传入参数、传出参数和函数返回值类型。

S402、在 REE 中，非可信处理模块将 ecall 任务存入共享内存。

其中，共享内存是在 REE 中针对 CA 预先申请的且用于 REE 和 TEE 访问的内存区域。共享内存的含义可以参见本申请前面的相关描述，此处不再赘述。

其中，ecall 函数的传入参数和传出参数的存储位置也位于共享内存中。

由于 REE 和 TEE 都可以访问共享内存，非可信处理模块将 CA 发起的 ecall 任务存入共享内存后，TEE 侧通过对共享内存进行访问，可以获取该 ecall 任务，从而在不需要数据拷贝的情况下完成了 REE 侧向 TEE 侧的数据传递，缩短了 GP 架构下的数据传输路径，提高了数据处理的效率。

S403、在 TEE 中，可信处理模块从共享内存获取任务。

具体的，TEE 侧的可信处理模块访问共享内存，获取共享内存中存储的任务。

S404、在 TEE 中，可信处理模块为 ecall 任务分配第二线程。

其中，第二线程是在 TEE 中创建的，第二线程和第一线程分别绑定的处理单元不同。

可选的，处理单元为处理器的 CPU 核。本实施例对第二线程绑定的 CPU 核不作限定。比如，CPU 包括核心 0~核心 31，第二线程绑定的 CPU 核是核心 10。

在相关技术的一种实现方式中，由 REE 侧的 CA 创建第二线程，绑定 CPU 核，比如，核心 10。之后，将第二线程绑定的核心 10 从 REE 侧切换到 TEE 侧，核心 10 从 REE 模式切换到 TEE 模式。

而在本实施例中，由于第二线程是在 TEE 侧被创建的，第二线程绑定的处理单元（核心 10）初始就为 TEE 模式，不需要进行 CPU 切换，因而提高了数据处理效率。

S405、在 TEE 中，TA 通过第二线程执行 ecall 任务，得到任务结果，并将任务结果存入共享内存。

具体的，第二线程可以调用 TA 提供的函数入口，执行 ecall 任务，将传出参数和函数返回值写入到 ecall 函数中的参数指定的共享区域。

下面对本步骤举例说明。在 REE 侧，CA 发起 ecall 任务的线程为第一线程，绑定的处理单元为核心 0。在 TEE 侧，TA 执行 ecall 任务调用的 ecall 函数的线程为第二线程，绑定 CPU 核 10。由于第二线程和第一线程绑定的 CPU 核不同，TA 执行 ecall 函数时，不需要将 REE 侧的第一线程绑定的 CPU 核（核心 0）切换到 TEE 侧，因此减少了 CPU 的切换开销，提高了处理效率。

而且，由于 REE 和 TEE 都可以访问共享内存，TA 执行 ecall 任务后将任务结果存入共享内存。REE 侧通过对共享内存进行访问，可以获取该 ecall 任务的任务结果，从而在不需要数据拷贝的情况下完成了 TEE 侧向 REE 侧的数据传递，缩短了 GP 架构下的数据传输路径，提高了数据处理的效率。

可见，在本实施例中，以 GP 架构作为示例，提供了一种数据处理方法，适用于终端设备在第一执行环境中由第一应用程序通过第一线程发起任务的场景。通过在非可信执行环境中针对第一应用程序预先申请共享内存，将执行任务的相关数据，例如，任务调用函数的传入参数和传出参数，任务结果等存储在共享内存中，可信执行环境和非可信执行环境通过访问共享内存完成了数据传递，不需要数据的多次拷贝，缩短了 REE 和 TEE 之间的数据传输路径，提高了数据处理的效率。而且，由于第一应用程序发起任务的第一线程和第二应用程序执行任务的第二线程分别绑定的处理单元不同，使得第一线程绑定的处理单元不需要进行 CPU 模式切换，减小了 CPU 的切换开销，提高了处理效率。而且，由于第二线程本身就在第二执行环境中被创建，使得第二线程绑定的处理单元不需要进行 CPU 模式切换就可以在第二执行环境中执行第一应用程序发起的任务，减小了 CPU 的切换开销，提高了处理效率。

可选的，本实施例提供的数据处理方法，在 S402 中非可信处理模块将 ecall 任务存入共享内存之前，还可以包括：

在 REE 中，非可信处理模块获取 ecall 任务的指示信息。其中，指示信息用于确定 ecall 任务

是否采用本实施例提供的数据处理方法。

在 REE 中，非可信处理模块根据 ecall 任务的指示信息，确定 ecall 任务是否采用本实施例提供的数据处理方法。

若确定 ecall 任务采用本实施例提供的数据处理方法，则执行后续的 S403~S405。

若确定 ecall 任务不采用本实施例提供的数据处理方法，则根据现有的 ecall 任务处理流程进行处理。

可见，通过 ecall 任务的指示信息判断 CA 发起的 ecall 任务是否执行本实施例提供的数据处理方法，提高了本实施例提供的数据处理方法与现有 ecall 任务处理流程的兼容性，提高了数据处理的灵活性。

可选的，指示信息可以包括下列中的至少一项：ecall 任务的配置信息或者 ecall 任务调用的 ecall 函数的关键字。

其中，函数的关键字参数是指在调用函数时，通过语法发送的参数。在本申请实施例中，通过函数调用关键字方法，可以判断 CA 发起的 ecall 任务是否执行本实施例提供的数据处理方法。

可选的，本实施例提供的数据处理方法，在 S402 中非可信处理模块将 ecall 任务存入共享内存之前，还可以包括：

在 REE 中，非可信处理模块获取共享内存在 REE 中的第一基准地址、传入参数在 REE 中的第一输入地址和传出参数在 REE 中的第一输出地址。

在 REE 中，非可信处理模块将第一输入地址转换为相对于第一基准地址的第一输入偏移量，将第一输出地址转换为相对于第一基准地址的第一输出偏移量。

其中，共享内存在 REE 中的第一基准地址，可以反映共享内存在 CA 地址空间的布局信息，例如，起始位置、偏移量等。ecall 任务还包括第一输入偏移量和第一输出偏移量。

可选的，本实施例提供的数据处理方法，在 S404 中可信处理模块为任务分配第二线程之前，还可以包括：

在 TEE 中，可信处理模块获取共享内存在 TEE 中的第二基准地址。

在 TEE 中，可信处理模块根据第二基准地址和第一输入偏移量获取传入参数在 TEE 中的第二输入地址，根据第二基准地址和第一输出偏移量获取传出参数在 TEE 中的第二输出地址。

其中，共享内存在 TEE 中的第二基准地址，可以反映共享内存在 TA 地址空间的布局信息，例如，起始位置、偏移量等。

具体的，在 GP 架构中，TEE 侧和 REE 侧的地址空间是完全隔离的。在 REE 侧，共享内存的地址映射由 REE 操作系统负责。在 TEE 侧，共享内存的地址映射由 TEE 操作系统负责。共享内存在 REE 侧和 TEE 侧会被不同的操作系统映射到 CA 进程地址和 TA 进程地址的不同区域中。这样，当 CA 发起 ecall 任务时，位于共享内存中的传入参数、传出参数等将无法被 TA 访问。反之亦然，TA 传递的位于共享内存中的参数，也无法被 CA 访问。因此，在 REE 侧，将 CA 传递的位于共享内存中的参数的地址，转换为相对于共享内存基准地址的偏移量，从而在 TEE 侧，可以根据参数地址的偏移量转换成 TEE 地址空间中的地址，TA 可以成功访问。反之亦然。

下面举例说明。

假设，共享内存在 REE 中的第一基准地址是共享内存在 CA 地址空间的起始地址，为 0x40000000，长度为 1MB。传入参数在 REE 中的第一输入地址为 0x40001000，长度为 8 字节。那么，传入参数在 REE 中的第一输入地址相对于共享内存在 REE 中的第一基准地址的偏移量为：0x40001000-0x40000000=0x1000，即，第一输入偏移量为 0x1000。

可选的，在计算地址的偏移量之前，判断传入参数的地址是否会超过内存地址的范围。比如，在该示例中，传入参数的长度为 8 字节，第一输入地址为 0x40001000，而共享内存的长度为 1MB，起始地址为 0x40000000，传入参数没有超过内存地址的范围。

相应的，假设，共享内存在 TEE 中的第二基准地址为 0x32000000，长度为 1MB。根据第二基准地址 0x32000000 以及传入参数的第一输入偏移量 0x1000，可以计算得到，传入参数在 TEE 中的第二输入地址为：0x32000000+0x1000=0x32001000。

可选的，在 TEE 侧，得到传入参数在 TEE 中的第二输入地址后，可以判断传入参数的地址是

否会超过内存地址的范围，确保对共享内存的访问不越界。

可选的，本实施例提供的数据处理方法，在 S401 中 CA 通过第一线程发起 ecall 任务之前，还可以包括：

在 REE 中，非可信处理模块获取共享内存的内存大小。

在 REE 中，非可信处理模块根据内存大小预先分配共享内存。

在 REE 中，向 TEE 发送共享内存的相关信息。

可选的，共享内存的相关信息至少包括共享内存的物理地址范围。

可选的，向 TEE 发送共享内存的相关信息，可以采用现有技术中的信息传输通道。例如，图 1B 所示架构中的 REE 通信代理和 TEE 通信代理。

可选的，本实施例提供的数据处理方法，还可以包括：

在 CA 完成与 TA 的交互后，在 REE 中，非可信处理模块释放共享内存。

可选的，在图 4 所示实施例的基础上，图 5 为本申请实施例提供的数据处理方法的另一种流程图，提供了 S402 和 S403 的实现方式。其中，S402 包括 S4021~S4023。S403 包括 S4031~S4032。S401、S403~S405 可以参见图 4 所示实施例中的描述，此处不再赘述。

如图 5 所示，本实施例提供的数据处理方法，可以包括：

S401、在 REE 中，CA 通过第一线程发起 ecall 任务。

S4021、在 REE 中，非可信处理模块序列化 ecall 任务，得到 ecall 任务的序列化消息。

其中，序列化的含义可以参见本申请前面的相关描述，此处不再赘述。

S4022、在 REE 中，非可信处理模块将 ecall 任务的序列化消息存入消息队列。其中，消息队列存储在共享内存中，用于存储 CA 发起的多个 ecall 任务分别对应的序列化消息。

S4023、在 REE 中，非可信处理模块将 ecall 任务的序列化消息的执行标志位设置为第一数值。

其中，第一数值用于指示消息队列中序列化消息的状态为待执行状态，以使 TEE 侧访问共享内存后从消息队列中获取尚未处理的 ecall 任务。

第一数值和第二数值为相对概念。第二数值用于指示消息队列中序列化消息的状态为已执行状态，以向 REE 侧通知该 ecall 任务执行完成。

第一数值和第二数值的取值不同。本申请实施例对第一数值和第二数值的取值不作限定。例如，第一数值为 0，第二数值为 1；或者，第一数值为 1，第二数值为 0。

S4031、在 TEE 中，可信处理模块访问消息队列，获取执行标志位为第一数值的 ecall 任务的序列化消息。

其中，本实施例对 TEE 侧的可信处理模块访问消息队列的规则不作限定。例如，可以通过轮询的方式，或者通过配置指定线程监控对应消息队列的方式，发现消息队列中执行标志位为第一数值的尚未处理的 ecall 任务。

S4032、在 TEE 中，可信处理模块解析 ecall 任务的序列化消息，得到 ecall 任务。

其中，解析和反序列化具有相同含义。通过对尚未处理的 ecall 任务的序列化消息进行解析，将序列化消息还原为 ecall 任务调用的 ecall 函数。

S404、在 TEE 中，可信处理模块为 ecall 任务分配第二线程。

S405、在 TEE 中，TA 通过第二线程执行 ecall 任务，得到任务结果，并将任务结果存入共享内存。

S406、在 TEE 中，可信处理模块将 ecall 任务的序列化消息的执行标志位设置为第二数值。

可见，本实施例提供的数据处理方法，以 GP 架构作为示例，适用于终端设备在第一执行环境中由第一应用程序通过第一线程发起任务的场景。在非可信执行环境中针对第一应用程序预先申请共享内存，非可信执行环境和可信执行环境都可以访问共享内存，共享内存中存储有消息队列和执行任务过程中的相关数据，通过将任务的执行标志位设置为不同的数值，指示任务是否已经处理。通过预先申请共享内存，实现了非可信执行环境和可信执行环境之间的数据传递，不需要数据拷贝，缩短了数据传递路径，提高了数据处理的效率。通过设置第一应用程序发起任务的第一线程和第二应用程序执行任务的第二线程分别绑定的处理单元不同，减小了 CPU 的切换开销，

提高了数据处理效率。而且，由于第二线程在第二执行环境中被创建，减小了 CPU 的切换开销，提高了数据处理效率。

在本申请的又一个实施例中，提供了一种数据处理方法，执行主体为终端设备。终端设备中部署有可信执行环境和非可信执行环境。本实施例对终端设备实现可信执行环境的方式不作限定。比如，终端设备可以采用 GP 架构，也可以采用 SGX 架构。可信执行环境和非可信执行环境中均部署有应用程序和处理模块。其中，可信执行环境中部署的应用程序称为可信应用程序，处理模块称为可信处理模块。非可信执行环境中部署的应用程序称为非可信应用程序，处理模块称为非可信处理模块。

本实施例提供的数据处理方法，适用于终端设备在第一执行环境中由第一应用程序通过第一线程发起任务的场景。

终端设备采用的架构不同时，第一执行环境和第二执行环境不同时，本实施例提供的数据处理方法，至少可以应用于上述的场景一~场景四。

为了方便说明，本实施例以场景一作为示例，对本实施例提供的数据处理方法进行说明。

示例性的，图 6 为本申请实施例提供的数据处理方法的又一种流程图。如图 6 所示，本实施例提供的数据处理方法，可以包括：

S601、在 REE 中，CA 通过第一线程发起 ecall 任务，并注册回调函数。

具体的，CA 发起 ecall 任务，调用 TEE 中 TA 提供的软件功能。第一线程是 CA 在 REE 中创建的线程。CA 在发起 ecall 任务时，提供回调函数的地址。

ecall 任务可以包括但不限于下列信息：调用的 ecall 函数的函数名、传入参数、传出参数和函数返回值类型。ecall 任务还包括回调函数。

S602、在 REE 中，非可信处理模块通过第三线程调用回调函数。其中，第三线程和第一线程不同。

具体的，第三线程是 CA 在 REE 中创建的线程。REE 侧确定 TEE 侧执行 ecall 任务结束，通过第三线程调用回调函数，进而自动完成对 TEE 侧返回的传出参数和函数返回值的处理。

可见，本实施例提供的数据处理方法，CA 通过第一线程发起 ecall 任务的同时注册回调函数，CA 发起 ecall 任务后，不需要等待 TEE 侧执行 ecall 任务的任务结果，而是可以通过第一线程进行后续应用业务逻辑的处理。在 TEE 侧执行 ecall 任务结束后，REE 侧通过第三线程调用回调函数，自动完成对 TEE 侧返回的传出参数和函数返回值的处理。通过注册回调函数，缩短了 REE 侧发起 ecall 任务后的等待时间，解决了 TEE 侧 ecall 任务执行阻塞 REE 侧应用业务逻辑执行的问题。

可选的，第三线程的数量小于或等于 CA 发起的 ecall 任务的数量。

下面举例说明。

如果第三线程的数量等于 CA 发起的 ecall 任务的数量，由于二者数量相等，每个第三线程可以执行一个 ecall 任务，提高了数据处理效率。假设，CA 发起 3 个 ecall 任务，分别为：ecall 任务 1~ecall 任务 3。第三线程为 3 个，分别为线程 31~线程 33。在一种实现方式中，线程 31 调用 ecall 任务 1 对应的回调函数、线程 32 调用 ecall 任务 2 对应的回调函数、线程 33 调用 ecall 任务 3 对应的回调函数。本实施例对第三线程和 ecall 任务的回调函数之间的对应关系不作限定。

如果第三线程的数量小于 CA 发起的 ecall 任务的数量，那么，存在至少一个第三线程，该第三线程调用至少 2 个 ecall 任务对应的回调函数。假设，CA 发起 3 个 ecall 任务，分别为：ecall 任务 1~ecall 任务 3。第三线程为 2 个，分别为线程 31、线程 32。在一种实现方式中，线程 31 调用 ecall 任务 1 和 ecall 任务 2 对应的回调函数、线程 32 调用 ecall 任务 3 对应的回调函数。在另一种实现方式中，线程 31 调用 ecall 任务 1 对应的回调函数、线程 32 调用 ecall 任务 2 和 ecall 任务 3 对应的回调函数。本实施例对第三线程和 ecall 任务的回调函数之间的对应关系不作限定。可选的，当一个第三线程调用至少 2 个 ecall 任务对应的回调函数时，可以按照时间顺序依次调用回调函数。

在本申请的又一个实施例中，提供了一种数据处理方法，执行主体为终端设备。终端设备中部署有可信执行环境和非可信执行环境。本实施例对终端设备实现可信执行环境的方式不作限定。比如，终端设备可以采用 GP 架构，也可以采用 SGX 架构。可信执行环境和非可信执行环境中均部署有应用程序和处理模块。其中，可信执行环境中部署的应用程序称为可信应用程序，处理模块称为可信处理模块。非可信执行环境中部署的应用程序称为非可信应用程序，处理模块称为非可信处理模块。

本实施例提供的数据处理方法，适用于终端设备在第一执行环境中由第一应用程序通过第一线程发起任务的场景。其中，第一执行环境为非可信执行环境、第一应用程序为非可信应用程序、第二执行环境为可信执行环境、第二应用程序为可信应用程序、任务为 ecall 任务。

终端设备采用的架构不同时，本实施例提供的数据处理方法，至少可以应用于上述的场景一和场景三。

为了方便说明，本实施例以场景一作为示例，对本实施例提供的数据处理方法进行说明。

示例性的，图 7 为本申请实施例提供的数据处理方法的又一种流程图。如图 7 所示，本实施例提供的数据处理方法，可以包括：

S701、在 TEE 中，可信处理模块针对 REE 中的 CA 建立线程池，线程池中包括多个线程。

其中，线程池中的每个线程均是在 TEE 中创建的。

其中，线程池的含义可以参见本申请前面的相关描述，此处不再赘述。

S401、在 REE 中，CA 通过第一线程发起 ecall 任务。

S404、在 TEE 中，可信处理模块为 ecall 任务分配第二线程。

其中，第二线程为线程池中的空闲线程。

其中，S401 和 S404 可以参见图 4 所示实施例，此处不再赘述。

S702、在 TEE 中，可信处理模块获取线程池中空闲线程的第一数量。

S703、在 TEE 中，可信处理模块获取 CA 发起的至少一个 ecall 任务中未执行的 ecall 任务的第二数量。

S704、在 TEE 中，可信处理模块根据第一数量和第二数量，调整线程池中线程的数量。

本实施例提供的数据处理方法，在 TEE 中，针对 REE 中的 CA 建立有线程池，线程池中包括多个线程。根据线程池中空闲线程的第一数量和 CA 发起的尚未处理的 ecall 任务的第二数量，实时调整线程池中线程的数量，避免了由于 CA 发起的 ecall 任务较多而造成 TEE 侧没有线程可用的情况，也避免了由于 CA 发起的 ecall 任务较少而造成 TEE 侧的 CPU 浪费，提高了 TEE 侧线程管理的灵活性，提高了 TEE 侧的资源利用率，也提高了 ecall 任务的处理效率。

可选的，S704 中，在 TEE 中，可信处理模块根据第一数量和第二数量，调整线程池中线程的数量，可以包括：

若第一数量大于第二数量，且线程池中线程的总数大于最小阈值，则在 TEE 中，可信处理模块在线程池中销毁一个线程。或者，

若第一数量小于第二数量，且线程池中线程的总数小于最大阈值，则在 TEE 中，可信处理模块在线程池中创建至少一个线程。至少一个线程的数量小于或等于最大阈值与线程池中线程的总数之差。

其中，本实施例对创建的至少一个线程的数量不做限定。但是，创建至少一个线程后需要保证线程池中所有线程的总数小于或等于最大阈值。

可选的，在一种实现方式中，至少一个线程的数量为 1。即，每次新创建 1 个线程。

可选的，在另一种实现方式中，至少一个线程的数量为最小阈值。

举例说明。假设，第一数量为 3、第二数量为 5、最大阈值为 10、最小阈值为 2、线程池中线程的总数当前为 7。那么，可信处理模块可以在线程池中创建最小阈值个线程，即创建 2 个线程。创建 2 个线程后，线程池中线程的总数更新为 7+2=9 个。之后继续判断第一数量和第二数量的大小。假设，第一数量为 2、第二数量为 3、线程池中线程的总数当前为 9。那么，如果可信处理模块在线程池中创建最小阈值个线程，线程池中线程的总数将更新为 9+2=11，超过了最大阈值 10。所以，可信处理模块可以在线程池中创建 1 个线程。创建 1 个线程后，线程池中线程的总数更新

为 9+1=10 个，达到最大阈值。

其中，线程池中线程的数量具有最小值和最大值，分别称为最小阈值和最大阈值。

具体的，当线程池中空闲线程的数量较多，大于 CA 发起的尚未处理的 ecall 任务的数量时，则在线程池中销毁一个线程，直至线程池中线程的数量达到最小阈值，确保了在 TEE 中存在线程用于执行 CA 发起的 ecall 任务，避免了 ecall 任务无法执行的情况。

当线程池中空闲线程的数量较少，小于 CA 发起的尚未处理的 ecall 任务的数量时，则在线程池中新创建至少一个线程，逐步或快速的增加线程池中线程的数量，直至线程池中线程的数量达到最大阈值。通过限制 TEE 中执行 CA 发起的 ecall 任务的线程最大值，避免了 TEE 侧的资源浪费。

可选的，S701 中，可信处理模块针对 REE 中的 CA 建立线程池，可以包括：

在 REE 中，向 TEE 发送最小阈值、最大阈值和中间阈值。中间阈值大于最小阈值且小于最大阈值。

在 TEE 中，可信处理模块针对 CA 建立线程池，线程池中线程的数量为中间阈值。

可选的，向 TEE 发送最小阈值、最大阈值和中间阈值，可以采用现有技术中的信息传输通道。例如，图 1B 所示架构中的 REE 通信代理和 TEE 通信代理。

在本实施方式中，TEE 侧根据 REE 侧设置的最小阈值、最大阈值和中间阈值建立针对 CA 的线程池，提高了管理线程池的合理性和准确性。

可选的，本实施例提供的数据处理方法，还可以包括：

在 CA 完成与 TA 的交互后，在 REE 中，非可信处理模块调用销毁接口函数。销毁接口函数用于指示在 TEE 中销毁线程池中的所有线程。

在 TEE 中，可信处理模块根据销毁接口函数，销毁线程池中的所有线程。

需要说明的是，本申请的各个实施例之间可以相互结合，对结合方式不作限定。比如，图 4 所示的实施例和图 6 所示的实施例可以相互结合。又比如，当第一执行环境为非可信执行环境、第二执行环境为可信执行环境时，图 4 所示实施例和图 7 所示实施例可以相互结合。

可以理解的是，终端设备为了实现上述功能，其包含了执行各个功能相应的硬件和/或软件模块。结合本文中所公开的实施例描述的各示例的算法步骤，本申请能够以硬件或硬件和计算机软件的结合形式来实现。某个功能究竟以硬件还是计算机软件驱动硬件的方式来执行，取决于技术方案的特定应用和设计约束条件。本领域技术人员可以结合实施例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申请的范围。

本申请实施例可以根据上述方法示例对终端设备进行功能模块的划分，例如，可以对应各个功能划分各个功能模块，也可以将两个或两个以上的功能集成在一个模块中。需要说明的是，本申请实施例中对模块的划分是示意性的，仅仅为一种逻辑功能划分，实际实现时可以有另外的划分方式。需要说明的是，本申请实施例中模块的名称是示意性的，实际实现时对模块的名称不做限定。

如图 2 所示，本申请实施例提供一种终端设备，终端设备部署可信执行环境和非可信执行环境。可信执行环境部署可信应用程序和可信处理模块。非可信执行环境部署非可信应用程序和非可信处理模块。

第一执行环境中的第一处理模块，用于将第一应用程序通过第一线程发起的任务存入共享内存；所述任务用于调用第二执行环境中第二应用程序提供的软件功能；所述共享内存是在所述非可信执行环境中针对所述第一应用程序预先申请的且用于所述非可信执行环境和所述可信执行环境访问的物理内存区域；所述第一执行环境为所述非可信执行环境、所述第二执行环境为所述可信执行环境、所述第一处理模块为所述非可信处理模块、第二处理模块为所述可信处理模块，或者，所述第一执行环境为所述可信执行环境、所述第二执行环境为所述非可信执行环境、所述第一处理模块为所述可信处理模块、第二处理模块为所述非可信处理模块；

所述第二执行环境中的所述第二处理模块，用于从所述共享内存获取所述任务；

所述第二执行环境中的所述第二处理模块，用于为所述任务分配第二线程；所述第二线程是在所述第二执行环境中创建的，所述第二线程和所述第一线程分别绑定的处理单元不同；

所述第二执行环境中的所述第二应用程序，用于通过所述第二线程执行所述任务，得到任务结果，并将所述任务结果存入所述共享内存。

可选的，所述第一执行环境中的所述第一处理模块，用于：

在所述第一执行环境中，序列化所述任务，得到所述任务的序列化消息；

在所述第一执行环境中，将所述任务的序列化消息存入消息队列；所述消息队列存储在所述共享内存中，用于存储所述第一应用程序发起的多个任务分别对应的序列化消息；

在所述第一执行环境中，将所述任务的序列化消息的执行标志位设置为第一数值；其中，第一数值用于指示序列化消息的状态为待执行状态。

可选的，所述第二执行环境中的所述第二处理模块，用于：

在所述第二执行环境中，访问所述消息队列，获取执行标志位为所述第一数值的所述任务的序列化消息；

在所述第二执行环境中，解析所述任务的序列化消息，得到所述任务。

可选的，所述第二执行环境中的所述第二处理模块，还用于：

在所述第二应用程序通过所述第二线程执行所述任务，得到任务结果，并将所述任务结果存入所述共享内存之后，在所述第二执行环境中，将所述任务的序列化消息的所述执行标志位设置为第二数值；其中，第二数值用于指示序列化消息的状态为已执行状态。

可选的，所述第一执行环境中的所述第一应用程序，用于：

在所述第一执行环境中，通过所述第一线程发起所述任务，并注册回调函数。

可选的，所述第一执行环境中的所述第一处理模块，还用于：

在所述第一执行环境中，通过第三线程调用所述回调函数；所述第三线程和所述第一线程不同。

可选的，所述第三线程的数量小于或等于所述第一应用程序发起的任务的数量。

可选的，所述任务包括传入参数和传出参数，所述传入参数和所述传出参数的存储位置位于所述共享内存中；

所述第一执行环境中的所述第一处理模块，还用于：

在将所述第一应用程序通过第一线程发起的任务存入共享内存之前，在所述第一执行环境中，获取所述共享内存在所述第一执行环境中的第一基准地址、所述传入参数在所述第一执行环境中的第一输入地址和所述传出参数在所述第一执行环境中的第一输出地址；

在所述第一执行环境中，将所述第一输入地址转换为相对于所述第一基准地址的第一输入偏移量，将所述第一输出地址转换为相对于所述第一基准地址的第一输出偏移量。

可选的，所述第二执行环境中的所述第二处理模块，还用于：

在所述第二应用程序通过所述第二线程执行所述任务之前，在所述第二执行环境中，获取所述共享内存在所述第二执行环境中的第二基准地址；

在所述第二执行环境中，根据所述第二基准地址和所述第一输入偏移量获取所述传入参数在所述第二执行环境中的第二输入地址，根据所述第二基准地址和所述第一输出偏移量获取所述传出参数在所述第二执行环境中的第二输出地址。

可选的，所述第一执行环境为所述可信执行环境，所述第二执行环境为所述非可信执行环境，所述任务为 ocall 任务。

可选的，所述第一执行环境为所述非可信执行环境，所述第二执行环境为所述可信执行环境，所述任务为 ecall 任务。

可选的，在所述第二执行环境中针对所述第一应用程序建立有线程池，所述线程池中包括多个线程，所述线程池中的线程是在所述第二执行环境中创建的；

所述第二线程为所述线程池中的空闲线程。

可选的，所述第二执行环境中的所述第二处理模块，还用于：

在所述第二执行环境中，获取所述线程池中空闲线程的第一数量；

在所述第二执行环境中，获取所述第一应用程序发起的至少一个任务中未执行的任务的第二数量；

在所述第二执行环境中，根据所述第一数量和所述第二数量，调整所述线程池中线程的数量。

可选的，所述第二执行环境中的所述第二处理模块，用于：

若所述第一数量大于所述第二数量，且所述线程池中线程的总数大于最小阈值，则在所述第二执行环境中，在所述线程池中销毁一个线程；或者，

若所述第一数量小于所述第二数量，且所述线程池中线程的总数小于最大阈值，则在所述第二执行环境中，在所述线程池中创建至少一个线程；所述至少一个线程的数量小于或等于所述最大阈值与所述线程池中线程的总数之差。

可选的，所述第一执行环境中的所述第一处理模块，还用于：

在将所述第一应用程序通过第一线程发起的任务存入共享内存之前，在所述第一执行环境中，向所述第二执行环境发送最小阈值、最大阈值和中间阈值；所述中间阈值大于所述最小阈值且小于所述最大阈值；

所述第二执行环境中的所述第二处理模块，还用于：

在所述第二执行环境中，针对所述第一应用程序建立所述线程池，所述线程池中线程的数量为所述中间阈值。

可选的，所述第一执行环境中的所述第一处理模块，还用于：

在将所述第一应用程序通过第一线程发起的任务存入共享内存之前，在所述第一执行环境中，获取所述共享内存的内存大小；

在所述第一执行环境中，根据所述内存大小分配所述共享内存；

在所述第一执行环境中，向所述第二执行环境发送所述共享内存的相关信息。

可选的，所述内存大小包括消息队列占用的第一内存大小和应用数据占用的第二内存大小；所述消息队列存储在所述共享内存中，用于存储所述第一应用程序发起的多个任务分别对应的序列化消息；

所述共享内存包括所述第一内存大小的第一内存空间和所述第二内存大小的第二内存空间。

可选的，所述第一内存空间和所述第二内存空间的地址连续。

请参考图 8，其示出了本申请实施例提供的终端设备的另一种结构。终端设备包括：处理器 801、存储器 804 和总线 805。可选的，还可以包括接收器 802 和发射器 803。处理器 801 包括一个或者多个处理核心，处理器 801 通过运行软件程序以及模块，从而执行各种功能的应用以及信息处理。存储器 804 可用于存储至少一个程序指令，处理器 801 用于执行至少一个程序指令，以实现上述实施例的技术方案。接收器 802 和发射器 803 可以实现为一个通信组件，该通信组件可以是一块基带芯片。存储器 804 通过总线 805 和处理器 801 相连。其实现原理和技术效果与上述方法相关实施例类似，此处不再赘述。

本领域技术人员可以理解，为了便于说明，图 8 仅示出了一个存储器和处理器。在实际的终端设备中，可以存在多个处理器和存储器。存储器也可以称为存储介质或者存储设备等，本申请实施例对此不做限制。

在本申请实施例中，处理器可以是通用处理器、数字信号处理器、专用集成电路、现场可编程门阵列或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件，可以实现或者执行本申请实施例中的公开的各方法、步骤及逻辑框图。通用处理器可以是微处理器或者任何常规的处理器等。结合本申请实施例所公开的方法的步骤可以直接体现为硬件处理器执行完成，或者用处理器中的硬件及软件模块组合执行完成。

在本申请实施例中，存储器可以是非易失性存储器，比如硬盘(hard disk drive，HDD)或固态硬盘(solid-state drive，SS)等，还可以是易失性存储器(volatile memory)，例如随机存取存储器(random- access memory，RAM)。存储器是能够用于携带或存储具有指令或数据结构形式的期望的程序代码并能够由计算机存取的任何其他介质，不限于此。

本申请实施例中的存储器还可以是电路或者其它任意能够实现存储功能的装置，用于存储程

序指令和/或数据。本申请各实施例提供的方法中，可以全部或部分地通过软件、硬件、固件或者其任意组合来实现。当使用软件实现时，可以全部或部分地以计算机程序产品的形式实现。所述计算机程序产品包括一个或多个计算机指令。在计算机上加载和执行所述计算机程序指令时，全部或部分地产生按照本申请实施例所述的流程或功能。所述计算机可以是通用计算机、专用计算机、计算机网络、网络设备、用户设备或者其他可编程装置。所述计算机指令可以存储在计算机可读存储介质中，或者从一个计算机可读存储介质向另一个计算机可读存储介质传输，例如，所述计算机指令可以从一个网站站点、计算机、服务器或数据中心通过有线(例如同轴电缆、光纤、数字用户线(digital subscriber line, DSL)或无线(例如红外、无线、微波等)方式向另一个网站站点、计算机、服务器或数据中心进行传输。所述计算机可读存储介质可以是计算机可以存取的任何可用介质或者是包含一个或多个可用介质集成的服务器、数据中心等数据存储设备。所述可用介质可以是磁性介质(例如，软盘、硬盘、磁带)、光介质(例如，数字视频光盘(digital video disc, DWD)、或者半导体介质(例如，SSD)等。

本申请实施例提供一种计算机程序产品，当所述计算机程序产品在终端设备运行时，使得所述终端设备执行上述实施例中的技术方案。其实现原理和技术效果与上述相关实施例类似，此处不再赘述。

本申请实施例提供一种计算机可读存储介质，其上存储有程序指令，所述程序指令被终端设备执行时，使得所述终端设备执行上述实施例的技术方案。其实现原理和技术效果与上述相关实施例类似，此处不再赘述。

综上所述，以上实施例仅用以说明本申请的技术方案，而非对其限制；尽管参照前述实施例对本申请进行了详细的说明，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，并不使相应技术方案的本质脱离本申请各实施例技术方案的范围。

权 利 要 求 书

1.一种数据处理方法，其特征在于，应用于终端设备，所述终端设备部署可信执行环境和非可信执行环境；所述方法包括：

在第一执行环境中，将由第一应用程序通过第一线程发起的任务存入共享内存；所述任务用于调用第二执行环境中第二应用程序提供的软件功能；所述共享内存是在所述非可信执行环境中针对所述第一应用程序预先申请的且用于所述非可信执行环境和所述可信执行环境访问的物理内存区域；所述第一执行环境为所述非可信执行环境、所述第二执行环境为所述可信执行环境，或者，所述第一执行环境为所述可信执行环境、所述第二执行环境为所述非可信执行环境；

在所述第二执行环境中，从所述共享内存获取所述任务；

在所述第二执行环境中，为所述任务分配第二线程；所述第二线程是在所述第二执行环境中创建的，所述第二线程和所述第一线程分别绑定的处理单元不同；

在所述第二执行环境中，由所述第二应用程序通过所述第二线程执行所述任务，得到任务结果，并将所述任务结果存入所述共享内存。

2.根据权利要求 1 所述的方法，其特征在于，所述在所述第一执行环境中，将所述任务存入共享内存，包括：

在所述第一执行环境中，序列化所述任务，得到所述任务的序列化消息；

在所述第一执行环境中，将所述任务的序列化消息存入消息队列；所述消息队列存储在所述共享内存中，用于存储所述第一应用程序发起的多个任务分别对应的序列化消息；

在所述第一执行环境中，将所述任务的序列化消息的执行标志位设置为第一数值；其中，第一数值用于指示序列化消息的状态为待执行状态。

3.根据权利要求 2 所述的方法，其特征在于，所述在所述第二执行环境中，从所述共享内存获取所述任务，包括：

在所述第二执行环境中，访问所述消息队列，获取执行标志位为所述第一数值的所述任务的序列化消息；

在所述第二执行环境中，解析所述任务的序列化消息，得到所述任务。

4.根据权利要求 2 所述的方法，其特征在于，所述在所述第二执行环境中，由所述第二应用程序通过所述第二线程执行所述任务，得到任务结果，并将所述任务结果存入所述共享内存之后，还包括：

在所述第二执行环境中，将所述任务的序列化消息的所述执行标志位设置为第二数值；其中，第二数值用于指示序列化消息的状态为已执行状态。

5.根据权利要求 1-4 中任一项所述的方法，其特征在于，所述在第一执行环境中，将由第一应用程序通过第一线程发起的任务存入共享内存之前，还包括：

在所述第一执行环境中，由所述第一应用程序通过所述第一线程发起所述任务，并注册回调函数。

6.根据权利要求 5 所述的方法，其特征在于，所述方法还包括：

在所述第一执行环境中，通过第三线程调用所述回调函数；所述第三线程和所述第一线程不同。

7.根据权利要求 6 所述的方法，其特征在于，所述第三线程的数量小于或等于所述第一应用程序发起的任务的数量。

8.根据权利要求 1-7 中任一项所述的方法，其特征在于，所述任务包括传入参数和传出参数，所述传入参数和所述传出参数的存储位置位于所述共享内存中；

所述在第一执行环境中，将由第一应用程序通过第一线程发起的任务存入共享内存之前，还包括：

在所述第一执行环境中，获取所述共享内存在所述第一执行环境中的第一基准地址、所述传入参数在所述第一执行环境中的第一输入地址和所述传出参数在所述第一执行环境中的第一输出地址；

在所述第一执行环境中，将所述第一输入地址转换为相对于所述第一基准地址的第一输入偏

移量，将所述第一输出地址转换为相对于所述第一基准地址的第一输出偏移量。

9.根据权利要求 8 所述的方法，其特征在于，所述在所述第二执行环境中，由所述第二应用程序通过所述第二线程执行所述任务之前，还包括：

在所述第二执行环境中，获取所述共享内存在所述第二执行环境中的第二基准地址；

在所述第二执行环境中，根据所述第二基准地址和所述第一输入偏移量获取所述传入参数在所述第二执行环境中的第二输入地址，根据所述第二基准地址和所述第一输出偏移量获取所述传出参数在所述第二执行环境中的第二输出地址。

10.根据权利要求 1-9 中任一项所述的方法，其特征在于，所述第一执行环境为所述可信执行环境，所述第二执行环境为所述非可信执行环境，所述任务为 ocall 任务。

11.根据权利要求 1-9 中任一项所述的方法，其特征在于，所述第一执行环境为所述非可信执行环境，所述第二执行环境为所述可信执行环境，所述任务为 ecall 任务。

12.根据权利要求 11 所述的方法，其特征在于，在所述第二执行环境中针对所述第一应用程序建立有线程池，所述线程池中包括多个线程，所述线程池中的线程是在所述第二执行环境中创建的；

所述第二线程为所述线程池中的空闲线程。

13.根据权利要求 12 所述的方法，其特征在于，所述方法还包括：

在所述第二执行环境中，获取所述线程池中空闲线程的第一数量；

在所述第二执行环境中，获取所述第一应用程序发起的至少一个任务中未执行的任务的第二数量；

在所述第二执行环境中，根据所述第一数量和所述第二数量，调整所述线程池中线程的数量。

14.根据权利要求 13 所述的方法，其特征在于，所述在所述第二执行环境中，根据所述第一数量和所述第二数量，调整所述线程池中线程的数量，包括：

若所述第一数量大于所述第二数量，且所述线程池中线程的总数大于最小阈值，则在所述第二执行环境中，在所述线程池中销毁一个线程；或者，

若所述第一数量小于所述第二数量，且所述线程池中线程的总数小于最大阈值，则在所述第二执行环境中，在所述线程池中创建至少一个线程；所述至少一个线程的数量小于或等于所述最大阈值与所述线程池中线程的总数之差。

15.根据权利要求 12 所述的方法，其特征在于，所述在第一执行环境中，将由第一应用程序通过第一线程发起的任务存入共享内存之前，还包括：

在所述第一执行环境中，向所述第二执行环境发送最小阈值、最大阈值和中间阈值；所述中间阈值大于所述最小阈值且小于所述最大阈值；

在所述第二执行环境中，针对所述第一应用程序建立所述线程池，所述线程池中线程的数量为所述中间阈值。

16.根据权利要求 11 所述的方法，其特征在于，所述在第一执行环境中，将由第一应用程序通过第一线程发起的任务存入共享内存之前，还包括：

在所述第一执行环境中，获取所述共享内存的内存大小；

在所述第一执行环境中，根据所述内存大小分配所述共享内存；

在所述第一执行环境中，向所述第二执行环境发送所述共享内存的相关信息。

17.根据权利要求 16 所述的方法，其特征在于，所述内存大小包括消息队列占用的第一内存大小和应用数据占用的第二内存大小；所述消息队列存储在所述共享内存中，用于存储所述第一应用程序发起的多个任务分别对应的序列化消息；

所述共享内存包括所述第一内存大小的第一内存空间和所述第二内存大小的第二内存空间。

18.根据权利要求 17 所述的方法，其特征在于，所述第一内存空间和所述第二内存空间的地址连续。

19.一种终端设备，其特征在于，包括处理器，所述处理器用于与存储器耦合，并读取存储器中的指令并根据所述指令使得所述终端设备执行如权利要求 1-18 中任一项所述的方法。

20.一种计算机可读存储介质，其特征在于，所述计算机可读存储介质存储有计算机指令，当

所述计算机指令在终端设备上运行时，使得所述终端设备执行如权利要求 1-18 中任一项所述的方法。

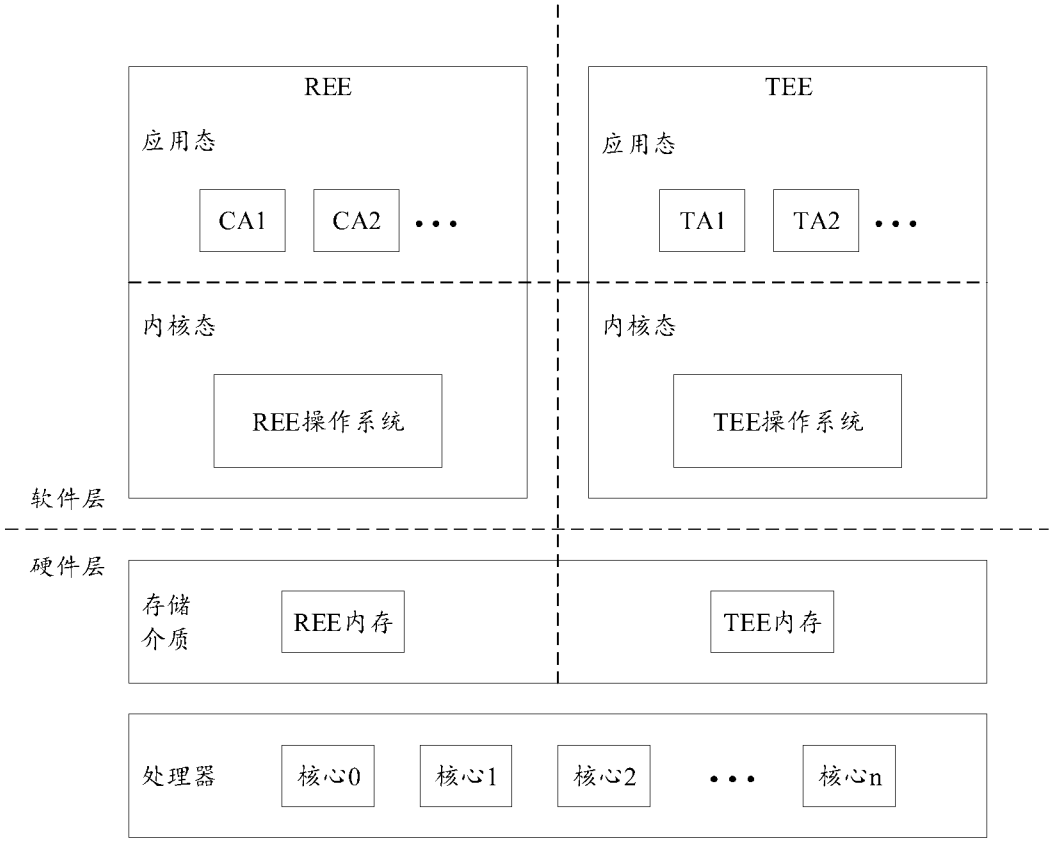
REE
应用态
CA1
CA2
• • •
内核态
REE操作系统
TEE
应用态
TA1
TA2
• • •
内核态
TEE操作系统
软件层
硬件层
存储
介质
REE内存
TEE内存
处理器
核心0
核心1
核心2
• • •
核心n

图 1A

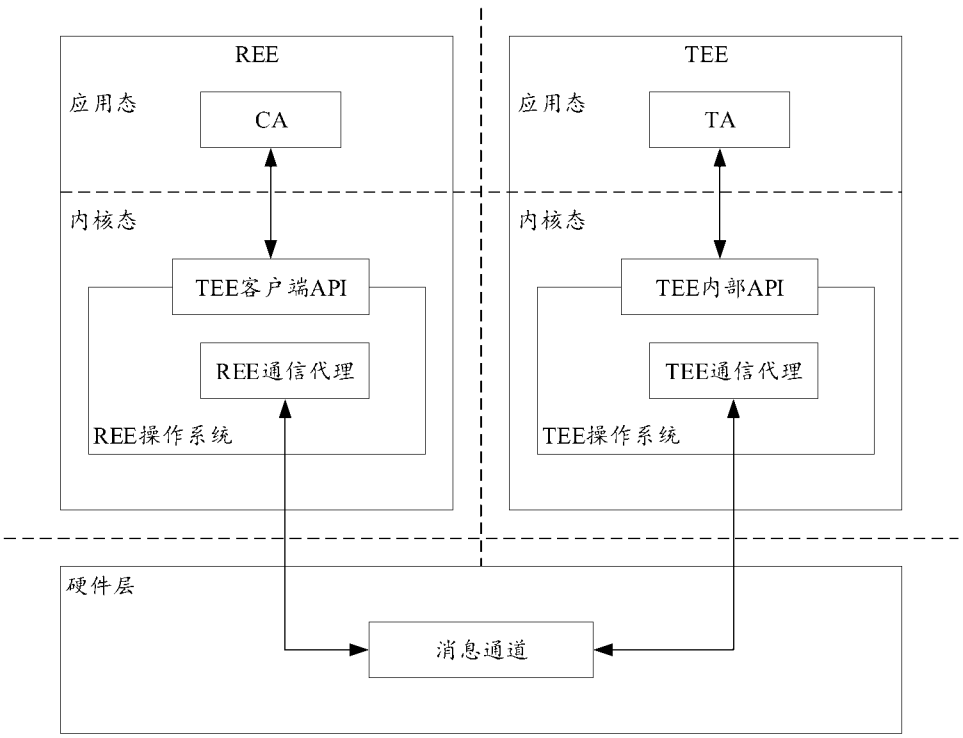
REE
应用态
CA
内核态
TEE客户端API
REE通信代理
REE操作系统
TEE
应用态
TA
内核态
TEE内部API
TEE通信代理
TEE操作系统
硬件层
消息通道

图 1B

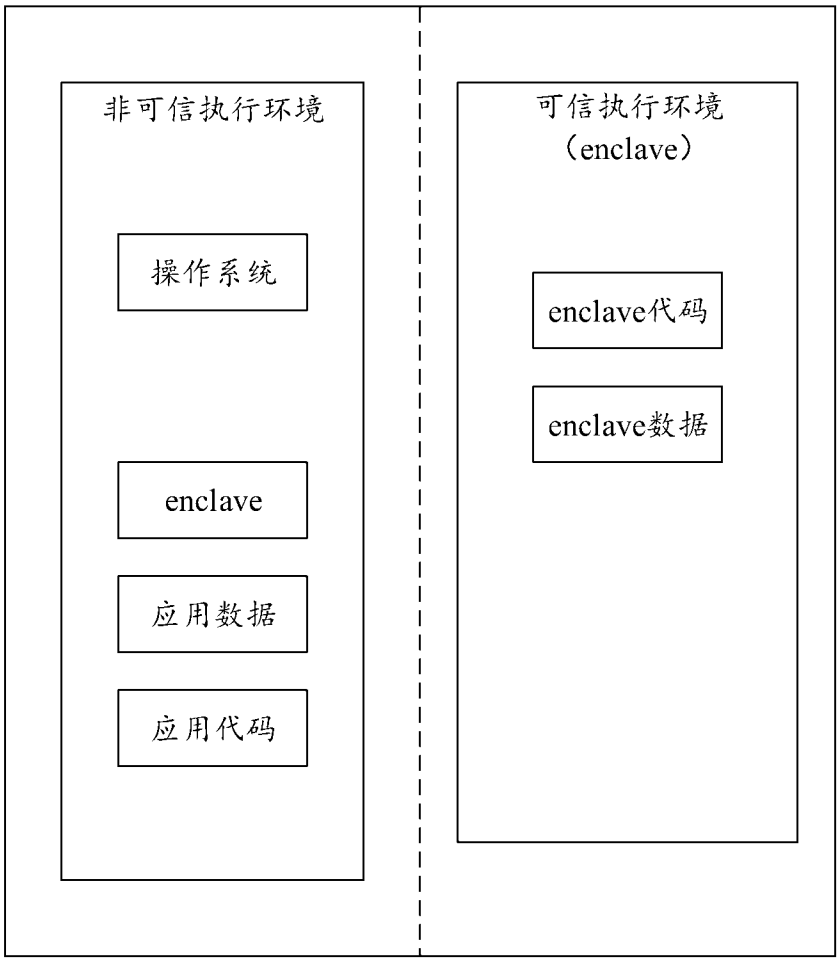
非可信执行环境
操作系统
enclave
应用数据
应用代码
可信执行环境
（enclave）
enclave代码
enclave数据

图 1C

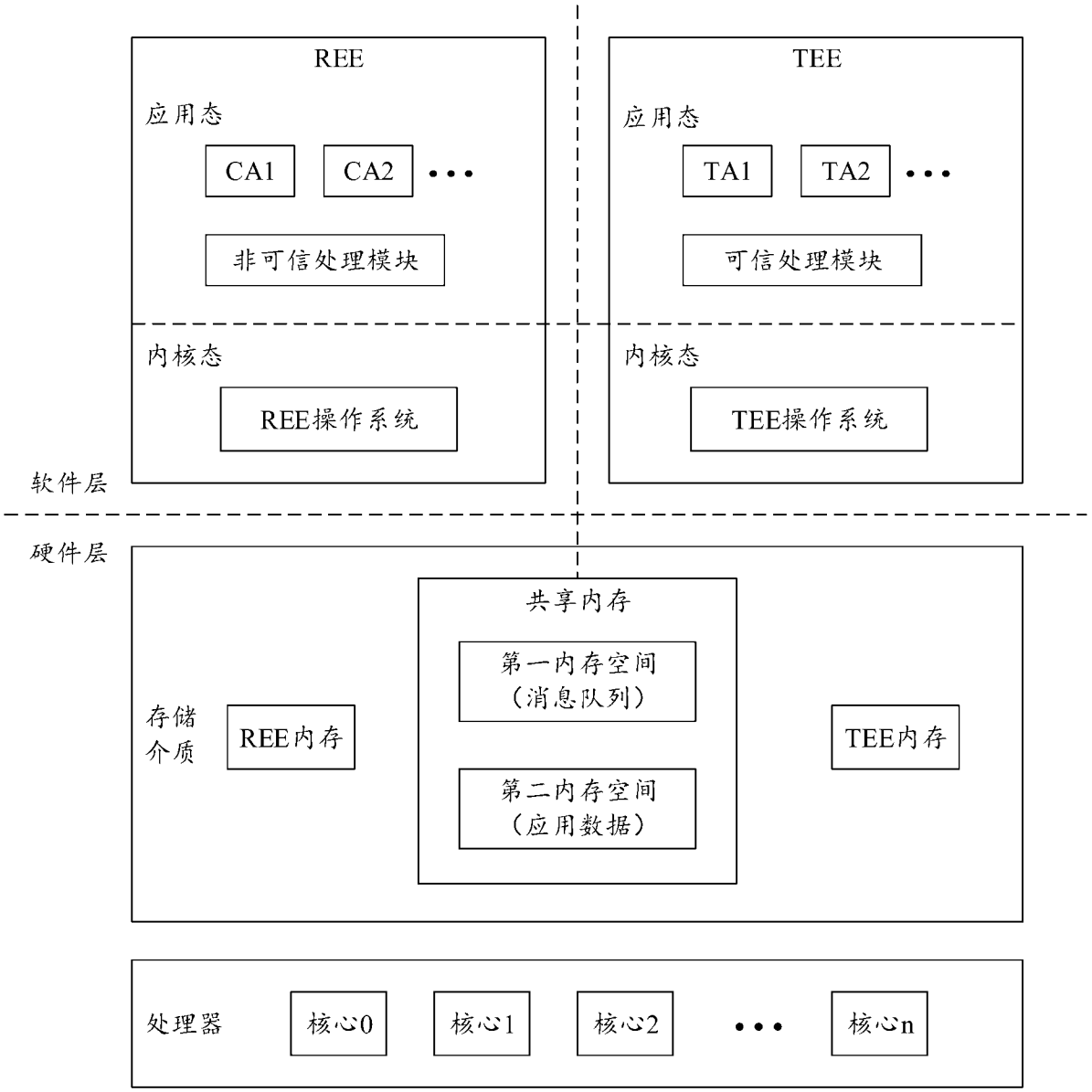
REE
应用态
CA1
CA2
非可信处理模块
内核态
REE操作系统
TEE
应用态
TA1
TA2
可信处理模块
内核态
TEE操作系统
软件层
硬件层
共享内存
第一内存空间
（消息队列）
第二内存空间
（应用数据）
存储
介质
REE内存
TEE内存
处理器
核心0
核心1
核心2
核心n

图 2


物理
内存
共享内存

图 3A

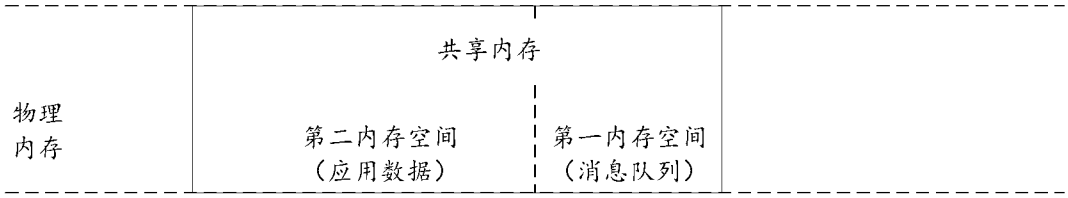
物理
内存
共享内存
第二内存空间
（应用数据）
第一内存空间
（消息队列）

图 3B

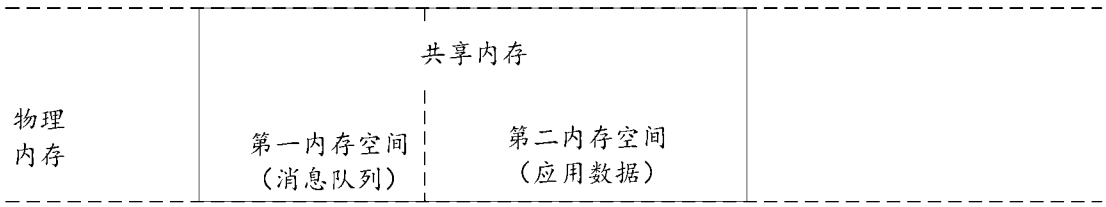
物理
内存
共享内存
第一内存空间
（消息队列）
第二内存空间
（应用数据）

图 3C

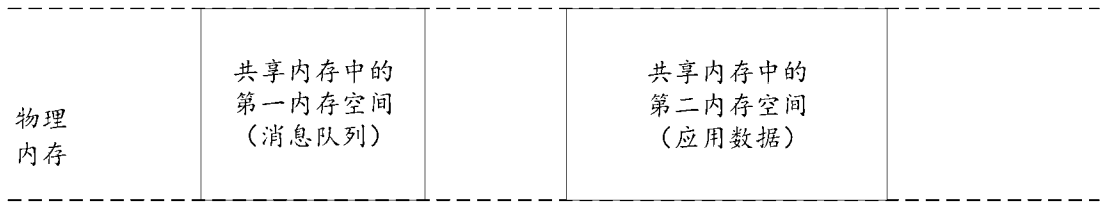
物理
内存
共享内存中的
第一内存空间
（消息队列）
共享内存中的
第二内存空间
（应用数据）

图 3D

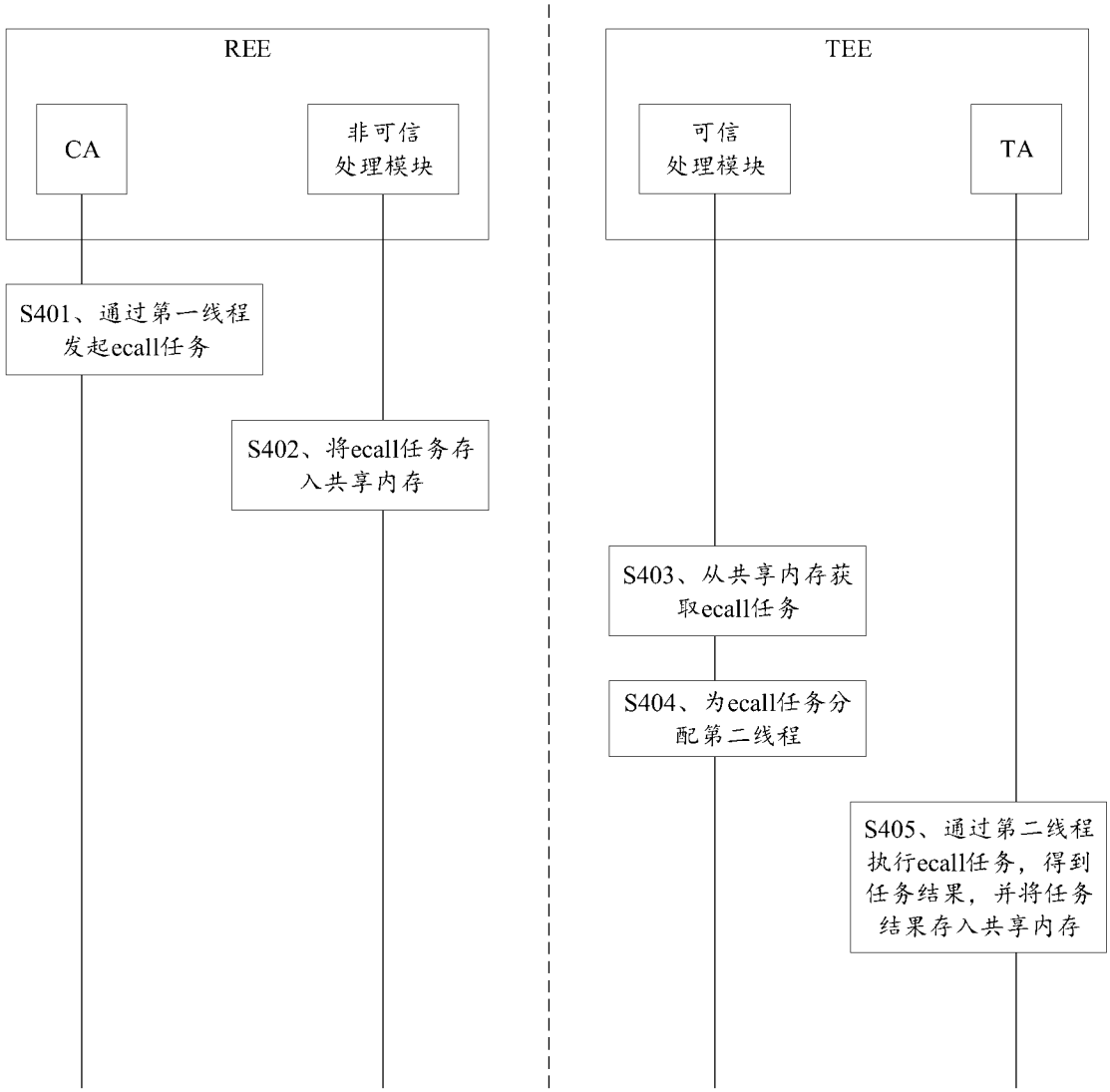
REE
CA
非可信
处理模块
TEE
可信
处理模块
TA
S401、通过第一线程
发起ecall任务
S402、将ecall任务存
入共享内存
S403、从共享内存获
取ecall任务
S404、为ecall任务分
配第二线程
S405、通过第二线程
执行ecall任务，得到
任务结果，并将任务
结果存入共享内存

图 4

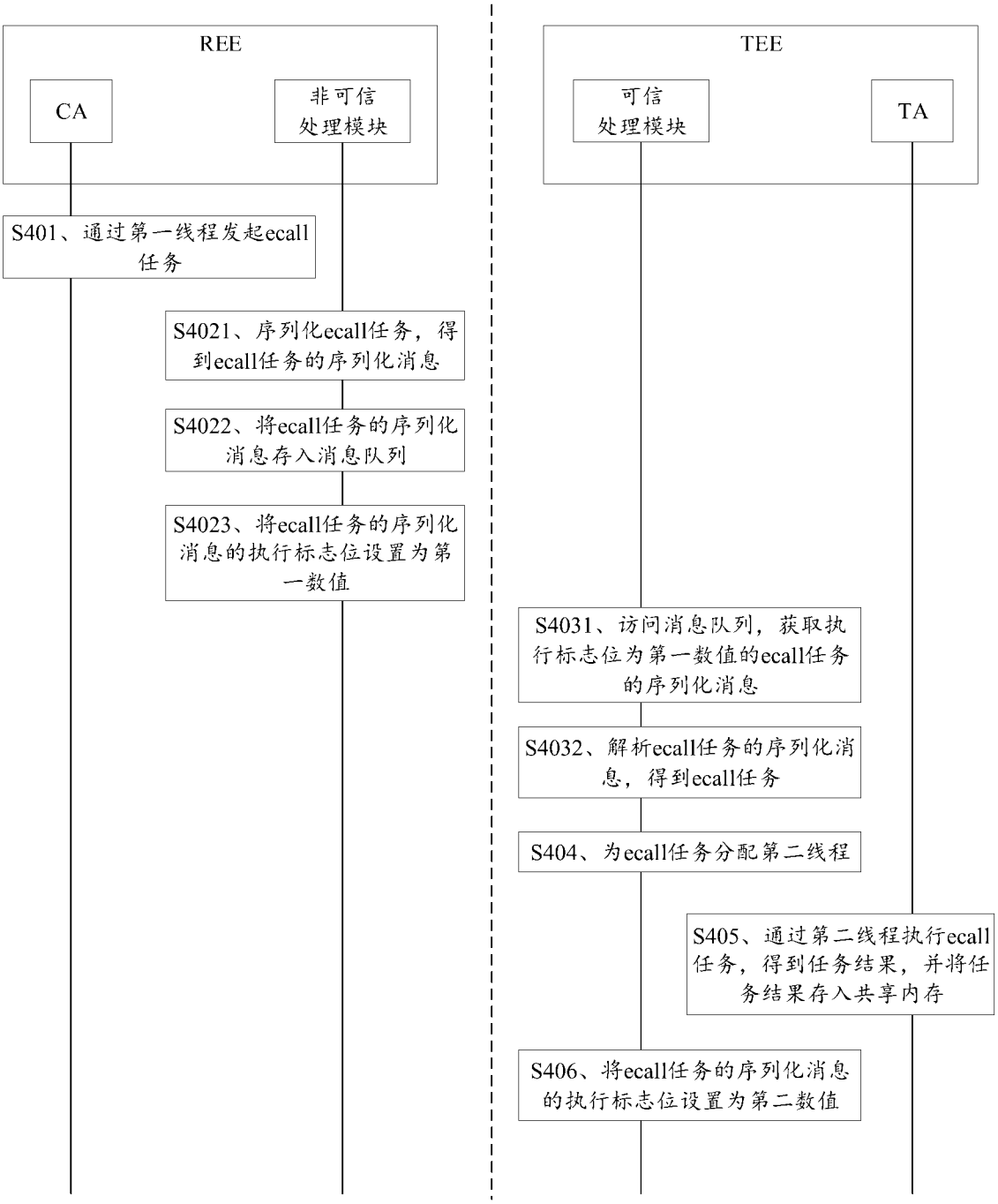
REE
CA
非可信
处理模块
TEE
可信
处理模块
TA
S401、通过第一线程发起ecall任务
S4021、序列化ecall任务，得到ecall任务的序列化消息
S4022、将ecall任务的序列化消息存入消息队列
S4023、将ecall任务的序列化消息的执行标志位设置为第一数值
S4031、访问消息队列，获取执行标志位为第一数值的ecall任务的序列化消息
S4032、解析ecall任务的序列化消息，得到ecall任务
S404、为ecall任务分配第二线程
S405、通过第二线程执行ecall任务，得到任务结果，并将任务结果存入共享内存
S406、将ecall任务的序列化消息的执行标志位设置为第二数值

图 5

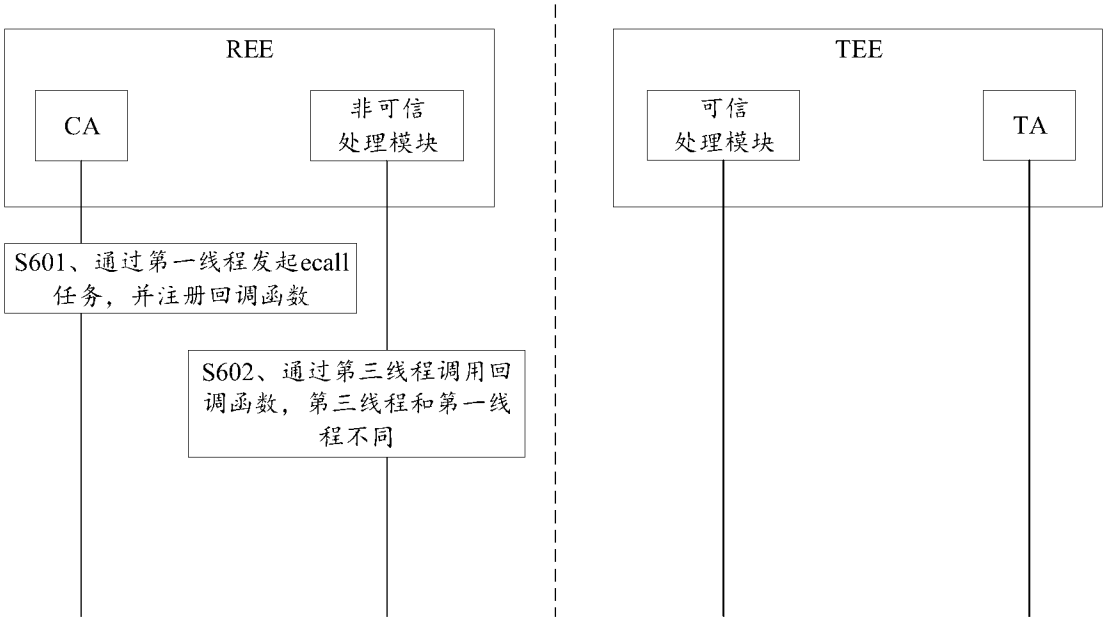
REE
CA
非可信
处理模块
TEE
可信
处理模块
TA
S601、通过第一线程发起ecall
任务，并注册回调函数
S602、通过第三线程调用回
调函数，第三线程和第一线
程不同

图 6

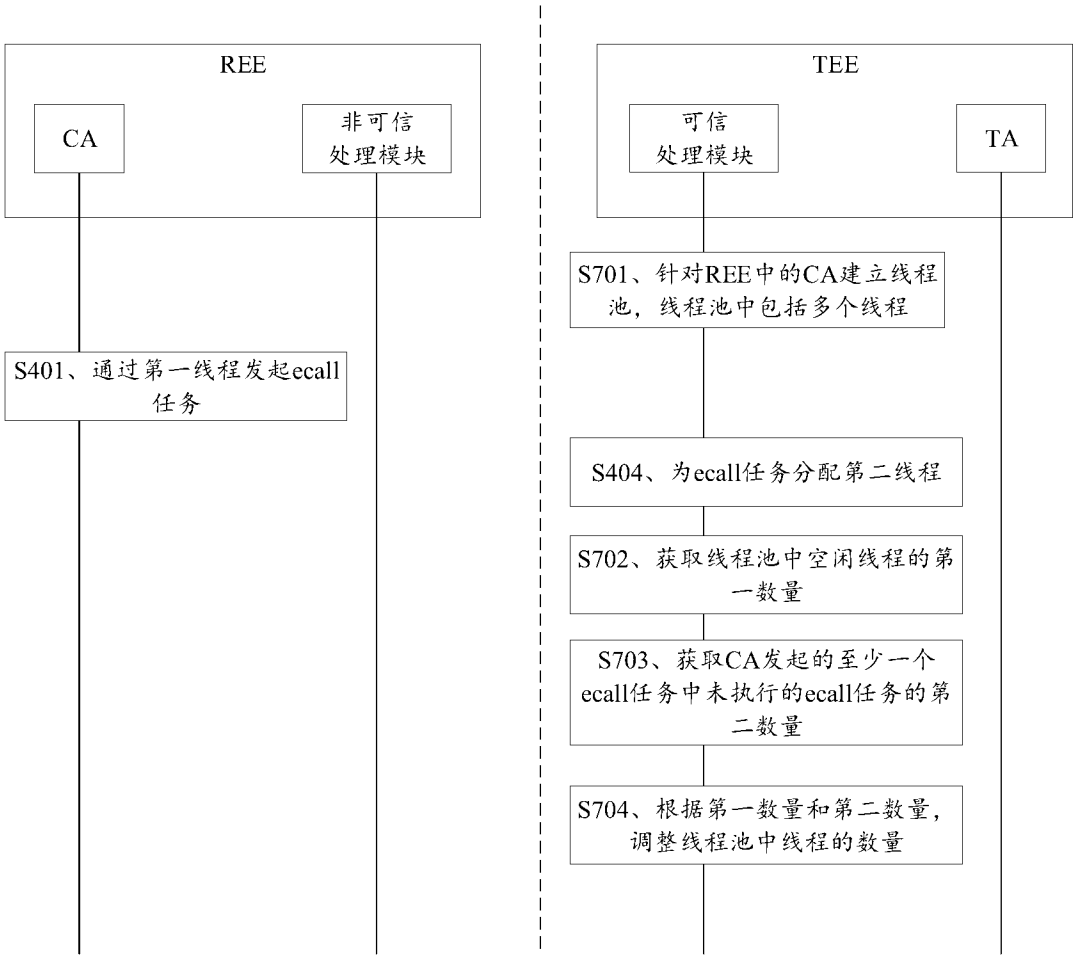
REE
CA
非可信
处理模块
TEE
可信
处理模块
TA
S401、通过第一线程发起ecall
任务
S701、针对REE中的CA建立线程
池，线程池中包括多个线程
S404、为ecall任务分配第二线程
S702、获取线程池中空闲线程的第
一数量
S703、获取CA发起的至少一个
ecall任务中未执行的ecall任务的第
二数量
S704、根据第一数量和第二数量，
调整线程池中线程的数量

图 7

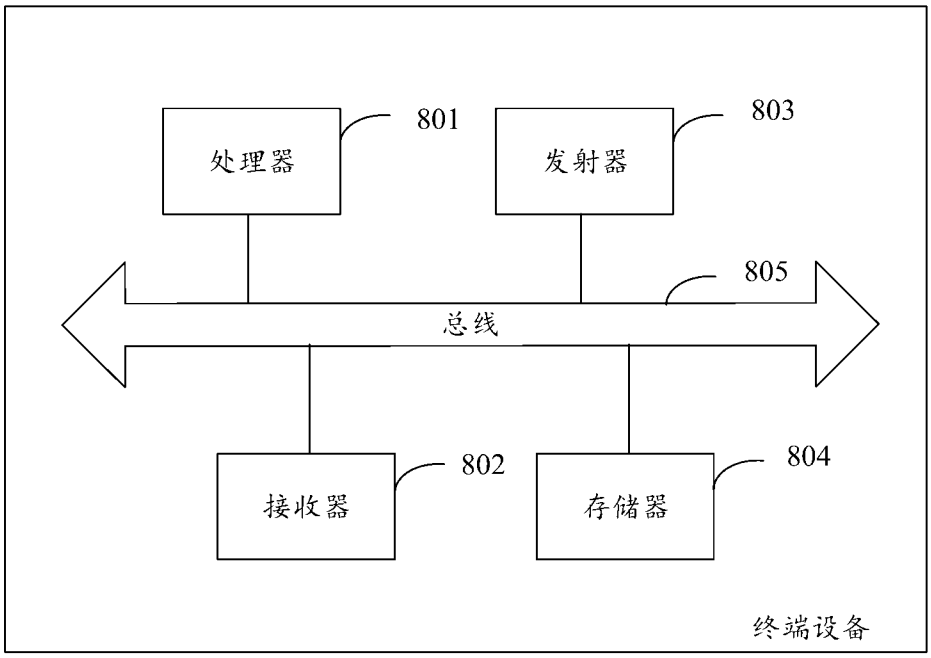

图 8

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2023/103965

A. CLASSIFICATION OF SUBJECT MATTER

G06F21/57(2013.01)i; G06F9/54(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: G06F21/-; G06F9/-

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)

CNABS, CNTXT, ENTXTC, CNKI: 可信, 安全, 环境, 第一应用, 第二应用, 普通应用, 安全应用, 可信应用, 共享, 公用, 共用, 内存, 空间, 线程, 绑定, 亲和, 不同, CPU内核, CPU核, 处理单元; VEN, ENTXT, IEEE: Trusted Execution Environment, TEE, REE, Rich Execution Environment, CA, TA, secure application, trusted application, shared memory, thread, different, CPU

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	CN 111859395 A (NATIONAL UNIVERSITY OF DEFENSE TECHNOLOGY OF PLA) 30 October 2020 (2020-10-30) description, paragraphs 0019-0032	1-20
Y	CN 111585999 A (SHENZHEN GENEW TECHNOLOGIES CO., LTD.) 25 August 2020 (2020-08-25) description, paragraph 0051	1-20
A	CN 109460373 A (ALIBABA GROUP HOLDING LIMITED) 12 March 2019 (2019-03-12) entire document	1-20
A	CN 110442462 A (ALIBABA GROUP HOLDING LIMITED) 12 November 2019 (2019-11-12) entire document	1-20
A	CN 111858004 A (NATIONAL UNIVERSITY OF DEFENSE TECHNOLOGY OF PLA) 30 October 2020 (2020-10-30) entire document	1-20

☑ 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	Date of mailing of the international search report
06 September 2023	**16 September 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

International application No.
PCT/CN2023/103965

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2018288095 A1 (AT&T INTELLECTUAL PROPERTY I, L. P.) 04 October 2018 (2018-10-04) entire document	1-20

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2023/103965

Patent document cited in search report	Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
CN 111859395 A	30 October 2020	None	
CN 111585999 A	25 August 2020	CN 111585999 B	29 March 2022
CN 109460373 A	12 March 2019	WO 2019047745 A1	14 March 2019
		CN 109460373 B	26 August 2022
CN 110442462 A	12 November 2019	WO 2021008111 A1	21 January 2021
		US 10896075 B1	19 January 2021
		US 2021019202 A1	21 January 2021
		TW 202119209 A	16 May 2021
		US 10884830 B1	05 January 2021
		US 2021019203 A1	21 January 2021
		CN 110442462 B	28 July 2020
		TW 730630 B1	11 June 2021
CN 111858004 A	30 October 2020	None	
US 2018288095 A1	04 October 2018	US 10805349 B2	13 October 2020

国际检索报告

国际申请号
PCT/CN2023/103965

A. 主题的分类

G06F21/57(2013.01)i; G06F9/54(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: G06F21/-; G06F9/-

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））

CNABS, CNTXT, ENTXTC, CNKI: 可信, 安全, 环境, 第一应用, 第二应用, 普通应用, 安全应用, 可信应用, 共享, 公用, 共用, 内存, 空间, 线程, 绑定, 亲和, 不同, CPU内核, CPU核, 处理单元; VEN, ENTXT, IEEE: Trusted Execution Environment, TEE, REE, Rich Execution Environment, CA, TA, secure application, trusted application, shared memory, thread, different, CPU

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
Y	CN 111859395 A (中国人民解放军国防科技大学) 2020年10月30日 (2020-10-30) 说明书0019-0032段	1-20
Y	CN 111585999 A (深圳震有科技股份有限公司) 2020年8月25日 (2020-08-25) 说明书0051段	1-20
A	CN 109460373 A (阿里巴巴集团控股有限公司) 2019年3月12日 (2019-03-12) 全文	1-20
A	CN 110442462 A (阿里巴巴集团控股有限公司) 2019年11月12日 (2019-11-12) 全文	1-20
A	CN 111858004 A (中国人民解放军国防科技大学) 2020年10月30日 (2020-10-30) 全文	1-20
A	US 2018288095 A1 (AT&T INTELLECTUAL PROPERTY I,L.P.) 2018年10月4日 (2018-10-04) 全文	1-20

☐ 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	国际检索报告邮寄日期
2023年9月6日	2023年9月16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	受权官员 倪礼 电话号码 (+86) 028-62967622

国际检索报告
关于同族专利的信息

国际申请号
PCT/CN2023/103965

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)
CN	111859395	A	2020年10月30日	无			
CN	111585999	A	2020年8月25日	CN	111585999	B	2022年3月29日
CN	109460373	A	2019年3月12日	WO	2019047745	A1	2019年3月14日
				CN	109460373	B	2022年8月26日
CN	110442462	A	2019年11月12日	WO	2021008111	A1	2021年1月21日
				US	10896075	B1	2021年1月19日
				US	2021019202	A1	2021年1月21日
				TW	202119209	A	2021年5月16日
				US	10884830	B1	2021年1月5日
				US	2021019203	A1	2021年1月21日
				CN	110442462	B	2020年7月28日
				TW	730630	B1	2021年6月11日
CN	111858004	A	2020年10月30日	无			
US	2018288095	A1	2018年10月4日	US	10805349	B2	2020年10月13日