

业务通信方法、系统、装置及电子设备

相关申请的交叉引用

本申请基于申请号为 202011222173.X、申请日为 2020 年 11 月 05 日的中国专利申请提出，并要求该中国专利申请的优先权，该中国专利申请的全部内容在此引入本申请作为参考。

5 技术领域

本申请涉及通信技术，尤其涉及一种业务通信方法、系统、装置、电子设备、计算机可读存储介质及计算机程序产品。

背景技术

10 在业务场景中，常常会出现接口调用的情况，例如终端设备的某个进程去调用该终端设备的另一个进程的接口，又例如终端设备的进程去调用服务器的进程的接口。接口调用的目的是实现业务通信，例如发送或者请求特定的数据等。

在相关技术提供的方案中，对于业务接入进程（即调用方进程）的调用，通常是由业务接入进程来约定一个固定的密钥，并基于该密钥来进行后续的加密业务通信。但是，该密钥容易被恶意进程利用，导致业务通信的安全性不足，无法对业务进行有效保障。

15 发明内容

本申请实施例提供一种业务通信方法，包括：

接收业务接入进程发送的鉴权请求；

对所述业务接入进程进行同步验证处理，对所述业务接入进程进行异步验证处理；

根据对所述业务接入进程的同步验证处理结果，确定为所述业务接入进程分配的业务密钥信息；

20 将所述业务密钥信息发送至所述业务接入进程，以基于所述业务密钥信息进行与所述业务接入进程之间的加密业务通信；

根据对所述业务接入进程的异步验证处理结果，控制与所述业务接入进程之间用于承载所述加密业务通信的通信连接。

25 本申请实施例提供一种业务通信系统，包括业务接入客户端、安全客户端及安全服务器；其中，所述业务接入客户端运行有业务接入进程；

所述安全客户端，用于：

接收所述业务接入进程发送的鉴权请求；

对所述业务接入进程进行同步验证处理，通知所述安全服务器对所述业务接入进程进行异步验证处理；

30 根据对所述业务接入进程的同步验证处理结果，确定为所述业务接入进程分配的业务密钥信息；

将所述业务密钥信息发送至所述业务接入进程，以基于所述业务密钥信息进行与所述业务接入进程之间的加密业务通信；

根据所述安全服务器对所述业务接入进程的异步验证处理结果，控制与所述业务接入进程之间用于承载所述加密业务通信的通信连接。

35 本申请实施例提供一种业务通信装置，包括：

接收模块，配置为接收业务接入进程发送的鉴权请求；

验证模块，配置为对所述业务接入进程进行同步验证处理，对所述业务接入进程进行异步验证处理；

40 确定模块，配置为根据对所述业务接入进程的同步验证处理结果，确定为所述业务接入进程分配的业务密钥信息；

发送模块，配置为将所述业务密钥信息发送至所述业务接入进程，以基于所述业务密钥信息进行与所述业务接入进程之间的加密业务通信；

连接控制模块,配置为根据对所述业务接入进程的异步验证处理结果,控制与所述业务接入进程之间用于承载所述加密业务通信的通信连接。

本申请实施例提供一种电子设备,包括:

存储器,用于存储可执行指令;

5 处理器,用于执行所述存储器中存储的可执行指令时,实现本申请实施例提供的业务通信方法。

本申请实施例提供一种计算机可读存储介质,存储有可执行指令,用于引起处理器执行时,实现本申请实施例提供的业务通信方法。

本申请实施例提供一种计算机程序产品,包括可执行指令,所述可执行指令被处理器执行时实现本申请实施例提供的业务通信方法。

10 附图说明

图1是本申请实施例提供的业务通信系统的一个架构示意图;

图2是本申请实施例提供的结合区块链网络的业务通信系统的一个架构示意图;

图3是本申请实施例提供的终端设备的一个架构示意图;

图4A是本申请实施例提供的业务通信方法的一个流程示意图;

15 图4B是本申请实施例提供的业务通信方法的一个流程示意图;

图4C是本申请实施例提供的业务通信方法的一个流程示意图;

图4D是本申请实施例提供的业务通信方法的一个流程示意图;

图4E是本申请实施例提供的业务通信方法的一个流程示意图;

图5是本申请实施例提供的安全管理端的策略管理界面的一个示意图;

20 图6是本申请实施例提供的进程信息的一个示意图;

图7是本申请实施例提供的安全管理端的零信任网关界面的一个示意图;

图8是本申请实施例提供的零信任网关的配置界面的一个示意图;

图9是本申请实施例提供的安全管理端的业务系统界面的一个示意图;

图10是本申请实施例提供的业务系统的配置界面的一个示意图;

25 图11是本申请实施例提供的业务系统的配置界面的一个示意图;

图12是本申请实施例提供的业务系统的配置界面的一个示意图;

图13是本申请实施例提供的安全管理端的策略管理界面的一个示意图;

图14是本申请实施例提供的安全客户端界面的一个示意图;

图15是本申请实施例提供的安全客户端界面的一个示意图;

30 图16是本申请实施例提供的安全客户端界面的一个示意图;

图17是本申请实施例提供的访问过程的一个示意图;

图18是本申请实施例提供的访问过程的一个示意图;

图19是本申请实施例提供的安全服务器级联部署的一个示意图;

图20是本申请实施例提供的异步验证处理的一个示意图。

35 具体实施方式

为了使本申请的目的、技术方案和优点更加清楚,下面将结合附图对本申请作进一步地详细描述,所描述的实施例不应视为对本申请的限制,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,都属于本申请保护的范围。

40 在以下的描述中,涉及到“一些实施例”,其描述了所有可能实施例的子集,但是可以理解,“一些实施例”可以是所有可能实施例的相同子集或不同子集,并且可以在不冲突的情况下相互结合。

在以下的描述中,所涉及的术语“第一\第二\第三”仅仅是是区别类似的对象,不代表针对对象的特定排序,可以理解地,“第一\第二\第三”在允许的情况下可以互换特定的顺序或先后次序,以使这里描述的本申请实施例能够以除了在这里图示或描述的以外的顺序实施。在以下的描述中,所涉及的术语“多个”是指至少两个。

45 除非另有定义,本文所使用的所有的技术和科学术语与属于本申请的技术领域的技术人员通常理解的含义相同。本文中所使用的术语只是为了描述本申请实施例的目的,不是旨在限制本申请。

对本申请实施例进行进一步详细说明之前,对本申请实施例中涉及的名词和术语进行说明,本

申请实施例中涉及的名词和术语适用于如下的解释。

1) 业务接入进程: 指作为调用方的进程, 本申请实施例对业务接入进程的类型不做限定, 例如业务接入进程可以是应用进程(如某个会议应用的进程), 也可以是专门用于对应用进程的业务请求进行代理的进程。

5 2) 同步(Synchronous)验证处理: 需要等待同步验证处理完成, 即得到同步验证处理结果时, 才能执行其他的步骤。在本申请实施例中, 涉及对业务接入进程的同步验证处理、以及对应用进程的同步验证处理, 举例来说, 只有在得到对业务接入进程的同步验证处理结果时, 才执行确定为业务接入进程分配的业务密钥信息的步骤。

10 3) 异步(Asynchronous)验证处理: 在异步验证处理的期间, 可以执行其他的步骤。在本申请实施例中, 涉及对业务接入进程的异步验证处理、以及对应用进程的同步验证处理, 举例来说, 在对业务接入进程进行异步验证处理期间, 可以与业务接入进程进行加密业务通信。

4) 业务密钥信息: 用于进行加密业务通信, 业务密钥信息至少包括密钥, 还可以包括密钥标识等信息。

15 5) 签名信息: 泛指与数字签名(Digital Signature)相关的信息, 例如可以包括数字签名本身以及证书信息等。

6) 哈希(Hash): 将任意长度的输入通过散列算法(也称哈希处理)变换成固定长度的输出, 该输出即为 Hash 值(也称 Hash 结果), 从而, 可以通过得到的 Hash 值来对输入进行标识。散列算法包括信息摘要(Message-Digest, MD)算法及安全散列算法(Secure Hash Algorithm, SHA)等。

20 7) 对称密钥: 指发送数据的一方和接收数据的一方使用相同的密钥进行加密处理和解密处理, 本申请实施例对对称密钥的生成方式不做限定, 例如可通过高级加密标准(Advanced Encryption Standard, AES)算法来生成对称密钥。

25 8) 非对称密钥对: 包括公钥及私钥, 发送数据的一方通过公钥对数据进行加密处理, 接收数据的一方通过私钥对加密后的数据进行解密处理, 或者, 也可以用私钥对数据进行加密处理, 用公钥对加密后的数据进行解密处理。本申请实施例对生成非对称密钥对的方式不做限定, 例如可通过 RSA 算法进行生成。

9) 业务网关: 用于将接收到的业务请求转发给相应的业务服务器, 以实现对业务请求的代理。在本申请实施例中, 业务网关可以通过软件形式实现, 也可以通过硬件形式实现。

10) 业务服务器: 用于提供业务资源的服务器, 例如, 对于会议业务来说, 会议应用的后台服务器即为业务服务器, 用于给会议应用的联网运行提供数据支持。

30 11) 区块链(Blockchain): 由区块(Block)形成的加密的、链式的交易的存储结构。

12) 区块链网络(Blockchain Network): 通过共识的方式将新区块纳入区块链的一系列的节点的集合。

35 本申请实施例提供一种业务通信方法、系统、装置、电子设备及计算机可读存储介质, 能够提升业务通信的安全性。下面说明本申请实施例提供的电子设备的示例性应用, 本申请实施例提供的电子设备可以实施为终端设备, 也可以实施为服务器。

参见图 1, 图 1 是本申请实施例提供的业务通信系统 100 的一个架构示意图, 终端设备 400 通过网络 300 连接服务器 200, 网络 300 可以是广域网或者局域网, 又或者是二者的组合。其中, 终端设备 400 运行有业务接入客户端。

40 在一些实施例中, 以电子设备是终端设备为例, 本申请实施例提供的业务通信方法可以由终端设备实现。例如, 终端设备 400 中的业务接入客户端, 可以通过运行的业务接入进程, 调用终端设备 400 中的应用客户端的应用进程的接口, 即应用客户端会接收到业务接入进程发送的鉴权请求。此时, 应用客户端可以对业务接入进程进行同步验证处理及异步验证处理, 根据同步验证处理结果确定分配的业务密钥信息, 并基于业务密钥信息进行与业务接入进程之间的加密业务通信, 应用客户端在得到异步验证处理结果时, 还可以根据异步验证处理结果控制与业务接入进程之间的通信连接。基于建立的加密业务通信, 业务接入进程可实现多种业务目的, 对此不做限定。

45 举例来说, 业务接入进程可以是即时通信应用客户端(以下简称为客户端 A)的应用进程, 所调用的是文件管理客户端(以下简称为客户端 B)的应用进程, 当客户端 A 中的某条文件链接(如分享在会话界面中的文件链接)被用户触发时, 客户端 A 的应用进程会调用客户端 B 的应用进程的接口, 以与客户端 B 的应用进程建立加密业务通信。在已建立加密业务通信的基础上, 客户端 A 的应用进程可以向客户端 B 的应用进程发送包括该文件链接的业务请求, 客户端 B 的应用进程从所管理的文件中查询该文件链接对应的文件(即响应数据), 并将查询到的文件发送至客户端 A 的应用

进程, 以在客户端 A 的界面中进行显示。

在一些实施例中, 以电子设备是服务器为例, 本申请实施例提供的业务通信方法可以由服务器实现。例如, 终端设备 400 中的业务接入客户端可以通过运行的业务接入进程调用服务器 200 中运行的进程的接口, 即服务器 200 会接收到业务接入进程发送的鉴权请求。此时, 服务器 200 可以对业务接入进程进行同步验证处理及异步验证处理, 以建立与业务接入进程之间的加密业务通信。举例来说, 业务接入客户端可以是某应用的客户端, 服务器 200 是该应用的后台服务器, 业务接入客户端可以通过业务接入进程与服务器 200 中的进程建立加密业务通信, 从而向服务器 200 中的进程请求应用数据 (即响应数据)。

在一些实施例中, 本申请实施例提供的业务通信方法也可以由终端设备及服务器协同实现。例如, 终端设备 400 中运行有安全客户端, 服务器 200 为安全服务器。当安全客户端接收到业务接入进程发送的鉴权请求时, 安全客户端对业务接入进程进行同步验证处理, 并通知安全服务器对业务接入进程进行异步验证处理, 以与业务接入进程建立加密业务通信。

在一些实施例中, 终端设备 400 或服务器 200 可以通过运行计算机程序来实现本申请实施例提供的业务通信方法, 例如, 计算机程序可以是操作系统中的原生程序或软件模块; 可以是本地 (Native) 应用程序 (APP, Application), 即需要在操作系统中安装才能运行的程序; 也可以是小程序, 即只需要下载到浏览器环境中就可以运行的程序; 还可以是能够嵌入至任意 APP 中的小程序, 其中, 该小程序组件可以由用户控制运行或关闭。总而言之, 上述计算机程序可以是任意形式的应用程序、模块或插件。

在一些实施例中, 服务器 200 可以是独立的物理服务器, 也可以是多个物理服务器构成的服务器集群或者分布式系统, 还可以是提供云服务、云数据库、云计算、云函数、云存储、网络服务、云通信、中间件服务、域名服务、安全服务、CDN、以及大数据和人工智能平台等基础云计算服务的云服务器, 其中, 云服务可以是异步验证服务, 供终端设备 400 进行调用。终端设备 400 可以是智能手机、平板电脑、笔记本电脑、台式计算机、智能电视、智能音箱、智能手表等, 但并不局限于此。终端设备以及服务器可以通过有线或无线通信方式进行直接或间接地连接, 本申请实施例中不做限制。

参见图 2, 图 2 是本申请实施例提供的结合区块链网络的业务通信系统 110 的一个架构示意图, 包括区块链网络 500 (区块链网络 500 通常包括多个节点, 这里示例性地示出了节点 510)、认证中心 600 及电子设备 700, 电子设备 700 可以是服务器 (例如图 1 所示的服务器 200) 或者终端设备 (例如图 1 所示的终端设备 400), 根据实际应用场景而定。其中, 认证中心 600 用于向电子设备 700 下发数字证书。

电子设备 700 可接入区块链网络 500, 成为区块链网络 500 的客户端节点, 进而查询存储于区块链中的数据, 其中, 区块链可用于存储业务通信过程中的各种信息。例如, 电子设备 700 可以查询存储于区块链中的黑名单, 以根据黑名单对业务接入进程进行同步验证处理以及异步验证处理中的至少之一。

以本申请实施例提供的电子设备是终端设备为例说明, 可以理解的, 对于电子设备是服务器的情况, 图 3 中示出的结构中的部分 (例如用户接口、呈现模块和输入处理模块) 可以缺省。参见图 3, 图 3 是本申请实施例提供的终端设备 400 的结构示意图, 图 3 所示的终端设备 400 包括: 至少一个处理器 410、存储器 450、至少一个网络接口 420 和用户接口 430。终端 400 中的各个组件通过总线系统 440 耦合在一起。可理解, 总线系统 440 用于实现这些组件之间的连接通信。总线系统 440 除包括数据总线之外, 还包括电源总线、控制总线和状态信号总线。但是为了清楚说明起见, 在图 3 中将各种总线都标为总线系统 440。

处理器 410 可以是一种集成电路芯片, 具有信号的处理能力, 例如通用处理器、数字信号处理器 (DSP, Digital Signal Processor), 或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件等, 其中, 通用处理器可以是微处理器或者任何常规的处理器等。

用户接口 430 包括使得能够呈现媒体内容的一个或多个输出装置 431, 包括一个或多个扬声器和/或一个或多个视觉显示屏。用户接口 430 还包括一个或多个输入装置 432, 包括有助于用户输入的用户接口部件, 比如键盘、鼠标、麦克风、触屏显示屏、摄像头、其他输入按钮和控件。

存储器 450 可以是可移除的, 不可移除的或其组合。示例性的硬件设备包括固态存储器, 硬盘驱动器, 光盘驱动器等。存储器 450 可选地包括在物理位置上远离处理器 410 的一个或多个存储设备。

存储器 450 包括易失性存储器或非易失性存储器, 也可包括易失性和非易失性存储器两者。非

易失性存储器可以是只读存储器 (ROM, Read Only Memory), 易失性存储器可以是随机存取存储器 (RAM, Random Access Memory)。本申请实施例描述的存储器 450 旨在包括任意适合类型的存储器。

5 在一些实施例中, 存储器 450 能够存储数据以支持各种操作, 这些数据的示例包括程序、模块和数据结构或者其子集或超集, 下面示例性说明。

操作系统 451, 包括用于处理各种基本系统服务和执行硬件相关任务的系统程序, 例如框架层、核心库层、驱动层等, 用于实现各种基础业务以及处理基于硬件的任务;

10 网络通信模块 452, 用于经由一个或多个 (有线或无线) 网络接口 420 到达其他计算设备, 示例性的网络接口 420 包括: 蓝牙、无线相容性认证 (WiFi)、和通用串行总线 (USB, Universal Serial Bus) 等;

呈现模块 453, 用于经由一个或多个与用户接口 430 相关联的输出装置 431 (例如, 显示屏、扬声器等) 使得能够呈现信息 (例如, 用于操作外围设备和显示内容和信息的用户接口);

输入处理模块 454, 用于对一个或多个来自一个或多个输入装置 432 之一的一个或多个用户输入或互动进行检测以及翻译所检测的输入或互动。

15 在一些实施例中, 本申请实施例提供的业务通信装置可以采用软件方式实现, 图 3 示出了存储在存储器 450 中的业务通信装置 455, 其可以是程序和插件等形式的软件, 包括以下软件模块: 接收模块 4551、验证模块 4552、确定模块 4553、发送模块 4554 及连接控制模块 4555, 这些模块是逻辑上的, 因此根据所实现的功能可以进行任意的组合或进一步拆分。将在下文中说明各个模块的功能。

20 将结合本申请实施例提供的电子设备的示例性应用和实施, 说明本申请实施例提供的业务通信方法。

参见图 4A, 图 4A 是本申请实施例提供的业务通信方法的一个流程示意图, 将结合图 4A 示出的步骤进行说明。

在步骤 101 中, 接收业务接入进程发送的鉴权请求。

25 这里, 电子设备接收业务接入进程发送的鉴权请求, 例如, 可以在电子设备中预先约定一个鉴权接口 (如某个特定的端口), 当电子设备接收到业务接入进程对该鉴权接口的调用请求时, 将该调用请求作为鉴权请求。其中, 业务接入进程可以是某个应用客户端的应用进程, 也可以用于对应用客户端进行代理的进程, 将在后文进行阐述。值得说明的是, 本申请实施例对业务的类型不做限定, 例如可以是即时通信业务或视频业务等。

30 在步骤 102 中, 对业务接入进程进行同步验证处理, 对业务接入进程进行异步验证处理。

这里, 针对业务接入进程, 进行同步验证处理及异步验证处理, 以验证其合法性。本申请实施例对同步验证处理及异步验证处理的执行顺序不做限定, 例如可以同时执行, 又例如, 也可以在同步验证处理结果为验证成功时, 再执行异步验证处理。

35 本申请实施例对同步验证处理及异步验证处理的处理方式同样不做限定, 例如, 同步验证处理及异步验证处理的验证对象可以相同, 验证对象如业务接入进程的进程信息, 两者之间的区别在于同步验证处理只执行一次, 而异步验证处理是周期性地执行多次。

在步骤 103 中, 根据对业务接入进程的同步验证处理结果, 确定为业务接入进程分配的业务密钥信息。

40 例如, 可以根据对业务接入进程的同步验证处理结果, 判断是否确定为业务接入进程分配的业务密钥信息。其中, 业务密钥信息至少包括密钥, 还可以包括密钥标识等。业务密钥信息可以是针对业务接入进程预先分配好的, 也可以是实时分配的, 对此不做限定。

值得说明的是, 为了进一步提升安全性, 电子设备还可以周期性地对分配给业务接入进程的业务密钥信息进行更新, 并对更新前的业务密钥信息进行失效处理 (也称无效处理), 以触发业务接入进程重新发送鉴权请求, 从而获取更新后的业务密钥信息。

45 在一些实施例中, 鉴权请求包括鉴权请求地址; 可以通过这样的方式来实现上述的确定为业务接入进程分配的业务密钥信息: 针对不同业务接入进程发送的鉴权请求地址分配不同的业务密钥信息, 针对同一业务接入进程发送的不同鉴权请求地址分配不同的业务密钥信息; 针对不同业务接入进程发送的鉴权请求地址分配不同的业务密钥信息, 针对同一业务接入进程发送的不同鉴权请求地址分配相同的业务密钥信息; 针对不同业务接入进程发送的鉴权请求地址分配相同的业务密钥信息, 针对同一业务接入进程发送的不同鉴权请求地址, 分配相同的业务密钥信息。

50 这里, 电子设备可以预先约定多个鉴权请求地址, 以便业务接入进程根据鉴权请求地址发送鉴

权请求。电子设备在确定为业务接入进程分配的业务密钥信息时，可以根据业务接入进程本身以及鉴权请求地址这两个因素，实现多种分配方案。

举例来说，第一种分配方案是，针对不同业务接入进程发送的鉴权请求地址分配不同的业务密钥信息，并针对同一业务接入进程发送的不同鉴权请求地址分配不同的业务密钥信息。该种分配方案的安全性相较于其他分配方案来说更高，同时分配的业务密钥信息的数量也较多，电子设备需要耗费更多的存储资源来存储分配的业务密钥信息。

第二种分配方案是，针对不同业务接入进程发送的鉴权请求地址分配不同的业务密钥信息，针对同一业务接入进程发送的不同鉴权请求地址分配相同的业务密钥信息，即仅区分业务接入进程。较之第一种分配方案，第二种分配方案需要耗费的存储资源更少。

第三种分配方案是，不区分业务接入进程及鉴权请求地址，即分配统一的业务密钥信息，该种方案需要耗费的存储资源最少。

根据实际应用场景中的需求，可以选用上述三种分配方案中的任意一个，例如某企业对安全性需求最高，则可以选用上述的第一种分配方案。通过上述方式，提升了分配业务密钥信息的灵活性。

在一些实施例中，可以通过这样的方式来实现上述的根据对业务接入进程的同步验证处理结果，确定为业务接入进程分配的业务密钥信息：当对业务接入进程的同步验证处理结果为验证成功时，确定为业务接入进程分配的业务密钥信息。

这里，当对业务接入进程的同步验证处理结果为验证成功时，确定为业务接入进程分配的业务密钥信息；当对业务接入进程的同步验证处理结果为验证失败时，证明业务接入进程不合法，则拒绝为业务接入进程分配业务密钥信息，还可以断开与业务接入进程之间的通信连接。即，对业务接入进程的同步验证处理，实质上可以看作是对业务接入进程是否合法的初步验证。

在一些实施例中，当对业务接入进程的同步验证处理结果为验证失败时，可以向业务接入进程发送错误信息，以通知业务接入进程重新发送鉴权请求，或者与业务接入进程断开通信连接。另外，若在得到对业务接入进程的同步验证处理结果为验证失败时，正在执行对业务接入进程的异步验证处理，则可以中断对业务接入进程的异步验证处理，以节省计算资源。

在步骤 104 中，将业务密钥信息发送至业务接入进程，以基于业务密钥信息进行与业务接入进程之间的加密业务通信。

电子设备将为业务接入进程分配的业务密钥信息发送至业务接入进程，如此，便可基于业务密钥信息进行与业务接入进程之间的加密业务通信。例如，业务密钥信息包括一个对称密钥，则业务接入进程可以根据对称密钥对数据进行加密处理，并将加密后的数据发送至电子设备，以使电子设备根据对称密钥对接收到的加密后的数据进行解密处理；同样地，电子设备可以根据对称密钥对数据进行加密处理，并将加密后的数据发送至业务接入进程，以使业务接入进程根据对称密钥对加密后的数据进行解密处理。

在步骤 105 中，根据对业务接入进程的异步验证处理结果，控制与业务接入进程之间用于承载加密业务通信的通信连接。

这里，在得到对业务接入进程的异步验证处理结果时，根据该异步验证处理结果控制与业务接入进程之间的用于承载加密业务通信的通信连接。本申请实施例对通信连接的类型不做限定，例如可以是套接字（Socket）连接。

在一些实施例中，可以通过这样的方式来实现上述的根据对业务接入进程的异步验证处理结果，控制与业务接入进程之间用于承载加密业务通信的通信连接：当对业务接入进程的异步验证处理结果为验证成功时，维持与业务接入进程之间用于承载加密业务通信的通信连接；当对业务接入进程的异步验证处理结果为验证失败时，断开与业务接入进程之间用于承载加密业务通信的通信连接。

这里，当对业务接入进程的异步验证处理结果为验证成功时，证明业务接入进程合法，则继续维持与业务接入进程之间用于承载加密业务通信的通信连接；当对业务接入进程的异步验证处理结果为验证失败时，证明业务接入进程不合法，则断开与业务接入进程之间用于承载加密业务通信的通信连接，以避免对业务造成损害（例如业务数据被窃取）。即，对业务接入进程的异步验证处理，实质上可以看作是对业务接入进程是否合法的再度验证。

如图 4A 所示，本申请实施例通过同步验证处理和异步验证处理相结合的方式，能够对业务接入进程的合法性进行有效验证；同时，通过电子设备下发的业务密钥信息进行加密业务通信，能够降低业务密钥信息被恶意进程窃取并利用的概率，提升业务通信的安全性。

在一些实施例中，参见图 4B，图 4B 是本申请实施例提供的业务通信方法的一个流程示意图，图 4A 示出的步骤 102 可以通过步骤 201 至步骤 204 实现，将结合各步骤进行说明。

在步骤 201 中, 将业务接入进程的进程路径与设定的安全目录之间的匹配结果, 作为第一验证处理结果。

在本申请实施例中, 对业务接入进程的同步验证处理可以包括对进程路径的验证处理以及对签名信息的验证处理。对于前者, 可以根据实际应用场景预先设定安全目录, 并将业务接入进程的进程路径与安全目录进行匹配处理, 并将得到的匹配结果作为第一验证处理结果。其中, 当业务接入进程的进程路径位于安全目录中时, 确定进程路径与安全目录之间的匹配结果为成功, 即第一验证处理结果为验证成功; 当业务接入进程的进程路径并未位于安全目录中时, 证明业务接入进程不合法, 确定进程路径与安全目录之间的匹配结果为失败, 即第一验证处理结果为验证失败。

在步骤 202 中, 对业务接入进程的签名信息进行验证处理, 得到第二验证处理结果。

这里, 签名信息泛指与业务接入进程的数字签名相关的信息, 可以根据实际应用场景中的安全性需求来确定需要进行验证处理的签名信息的具体内容。对业务接入进程的签名信息进行验证处理后, 得到第二验证处理结果。

在一些实施例中, 可以通过这样的方式来实现上述的对业务接入进程的签名信息进行验证处理, 得到第二验证处理结果: 根据签名信息是否包括数字签名的结果、数字签名的有效性验证处理结果、数字签名的签名方与签名方黑名单之间的匹配结果、以及签名信息中的证书信息与证书信息黑名单之间的匹配结果中的至少之一, 确定第二验证处理结果。

本申请实施例提供了对业务接入进程的签名信息进行验证处理的四种影响因素, 以下进行分别说明。

1) 签名信息是否包括数字签名的结果。例如, 当签名信息包括数字签名时, 确定第二验证处理结果为验证成功; 当签名信息未包括数字签名时, 确定第二验证处理结果为验证失败。

2) 对签名信息中的数字签名进行有效性验证处理得到的有效性验证处理结果。例如, 可以直接将有效性验证处理结果作为第二验证处理结果, 有效性验证处理的方式在后文进行阐述。

3) 签名信息中的数字签名的签名方与签名方黑名单之间的匹配结果。例如, 签名方黑名单包括有多个恶意签名方, 当业务接入进程的数字签名的签名方与签名方黑名单中的任意一个恶意签名方相同, 即匹配成功时, 确定第二验证处理结果为验证失败; 当业务接入进程的数字签名的签名方与签名方黑名单中的所有恶意签名方均不同, 即匹配失败时, 确定第二验证处理结果为验证成功。

4) 签名信息中的证书信息与证书信息黑名单之间的匹配结果。其中, 证书指的是数字证书 (Digital Certificate)。例如, 证书信息黑名单包括有多个恶意证书信息, 当业务接入进程的证书信息与证书信息黑名单中的任意一个恶意证书信息相同, 即匹配成功时, 确定第二验证处理结果为验证失败; 当业务接入进程的证书信息与证书信息黑名单中的所有恶意证书信息均不同, 即匹配失败时, 确定第二验证处理结果为验证成功。这里, 证书信息可以是证书链 (Certificate Chain) 信息, 例如包括根证书信息、中级证书信息及签名证书信息, 证书信息的格式根据业务接入进程实际的证书签发情况而定。

上述的 1) 至 4) 的影响因素的验证力度递增, 验证的时长同样递增, 可以综合实际应用场景中的安全性需求和效率需求, 选用上述的至少一种影响因素进行验证处理。以选用上述的第 1) 种和第 2) 种方式为例进行说明, 则当签名信息包括数字签名、且对数字签名的有效性验证处理结果为验证成功时, 才确定第二验证处理结果为验证成功; 当签名信息未包括数字签名、或者对数字签名的有效性验证处理结果为验证失败时, 确定第二验证处理结果为验证失败。通过上述方式, 提升了对业务接入进程的签名信息进行验证处理的灵活性。

在一些实施例中, 还包括: 确定签名信息中的数字签名、以及与数字签名对应的解密密钥; 根据解密密钥对数字签名进行解密处理, 得到业务接入进程的进程文件的第一哈希结果; 对业务接入进程的进程文件进行哈希处理, 得到第二哈希结果; 将第一哈希结果与第二哈希结果之间的匹配结果, 作为数字签名的有效性验证处理结果。

这里, 提供了对数字签名进行有效性验证处理的一种示例。数字签名可以是对业务接入进程的进程文件进行哈希处理, 再对得到的第一哈希结果进行加密处理得到, 其中, 进程文件可以是指业务接入进程的可移植且可执行 (Portable Executable, PE) 文件。因此, 在对数字签名进行有效性验证处理时, 可以首先确定签名信息中的数字签名、以及与数字签名对应的解密密钥, 其中, 解密密钥可以从业务接入进程的证书信息中解析得到, 如从签名证书中解析得到。

然后, 根据解密密钥对数字签名进行解密处理, 得到第一哈希结果, 同时, 对业务接入进程的进程文件进行哈希处理, 得到第二哈希结果, 这里的哈希处理所用的算法同样可以从证书信息中得到, 如此, 可以保证计算第一哈希结果所用的算法与计算第二哈希结果所用的算法一致。最后, 将

第一哈希结果与第二哈希结果之间的匹配结果，作为有效性验证处理结果，例如，当第一哈希结果与第二哈希结果相同，即匹配成功时，确定有效性验证处理结果为验证成功；当第一哈希结果与第二哈希结果不同时，证明进程文件经过了篡改，则确定有效性验证处理结果为验证失败。当然，上述方式仅仅是有效性验证处理的一种示例，并不构成对有效性验证处理的限定，例如还可以通过调用特定的应用程序接口（Application Programming Interface, API），如 WinVerifyTrust 来实现有效性验证处理。

在步骤 203 中，根据第一验证处理结果及第二验证处理结果，确定对业务接入进程的同步验证处理结果。

例如，当第一验证处理结果及第二验证处理结果均为验证成功时，确定对业务接入进程的同步验证处理结果为验证成功；当第一验证处理结果及第二验证处理结果中的任意一个为验证失败时，确定对业务接入进程的同步验证处理结果为验证失败。如此，结合进程路径和签名信息两个方面来进行对业务接入进程的同步验证处理，能够有效提升得到的同步验证处理结果的准确性。

在步骤 204 中，周期性地将业务接入进程的进程信息与进程信息黑名单进行匹配，并将得到的匹配结果作为对业务接入进程的异步验证处理结果。

这里，进程信息可以包括进程文件的 MD5 值、进程文件的 Hash 值、进程路径以及证书信息中的至少之一，可以根据实际应用场景进行设定。进程信息黑名单则包括多个恶意进程的进程信息，进程信息黑名单可以是随着时间推移而不断更新的，类似于病毒库。在对业务接入进程进行异步验证处理的过程中，可以周期性地将业务接入进程的进程信息与进程信息黑名单（例如当前最新的进程信息黑名单）进行匹配，当业务接入进程的进程信息与进程信息黑名单中的任意一个进程信息相同，即匹配成功时，确定对业务接入进程的异步验证处理结果为验证失败；当业务接入进程的进程信息与进程信息黑名单中的所有进程信息均不同，即匹配失败时，确定对业务接入进程的异步验证处理结果为验证成功。

值得说明的是，在进程信息包括多种类型的内容（如 MD5 值、Hash 值等）的情况下，可以设定当进程信息 A 包括的某个内容（如 MD5 值）与进程信息 B 包括的同类型的内容相同时，确定进程信息 A 与进程信息 B 相同，从而加强安全管控的效率。

另外，可以周期性地将业务接入进程的进程信息发送至特定的云服务，以使云服务将业务接入进程的进程信息与进程信息黑名单进行匹配，从而减轻本地的处理压力。

如图 4B 所示，本申请实施例提供了对业务接入进程进行同步验证处理及异步验证处理的一些示例，能够有效提升得到的同步验证处理结果及异步验证处理结果的准确性，从而增强后续进行业务通信的安全性。

在一些实施例中，参见图 4C，图 4C 是本申请实施例提供的业务通信方法的一个流程示意图，图 4A 示出的步骤 104 可以通过步骤 301 至步骤 307 实现，将结合各步骤进行说明。

在步骤 301 中，根据鉴权请求中的公钥对业务密钥信息进行加密处理。

为了提升传输业务密钥信息过程中的安全性和保密性，在本申请实施例中，可以在步骤 101 之前，由业务接入进程生成非对称密钥对，并将非对称密钥对中的公钥添加至鉴权请求中，其中，非对称密钥对包括公钥以及与公钥配套的私钥。

电子设备在确定出为业务接入进程分配的业务密钥信息后，可以利用业务接入进程发送的公钥对业务密钥信息进行加密处理。

在步骤 302 中，将加密后的业务密钥信息发送至业务接入进程，以使业务接入进程根据私钥对加密后的业务密钥信息进行解密处理。

这里，电子设备将加密后的业务密钥信息发送至业务接入进程，以使业务接入进程根据存储的私钥对加密后的业务密钥信息进行解密处理。由于除业务接入进程外的其他进程未持有私钥，因此，就算加密后的业务密钥信息被其他进程窃取，其他进程也无法得到实际的业务密钥信息。通过上述方式，能够提升业务密钥信息的保密性。

在步骤 303 中，接收业务接入进程发送的业务请求；其中，业务请求包括密钥标识、以及利用对称密钥加密后的请求数据。

这里，以业务密钥信息包括密钥标识及对称密钥为例，说明加密业务通信的过程。业务接入进程在接收到业务密钥信息后，可以基于业务密钥信息生成调用请求，该调用请求包括密钥标识、以及利用对称密钥加密后的请求数据，这里对请求数据的类型不做限定，可以根据实际应用场景确定。

值得说明的是，可以在电子设备中预先约定一个业务接口（如某个特定的端口），当电子设备接收到业务接入进程对该业务接口的调用请求时，将该调用请求作为业务请求。其中，可以为多种业

务设定不同的业务接口，也可以设定统一的业务接口。

在步骤 304 中，在已分配的对称密钥中，查询与业务请求中的密钥标识对应的对称密钥。

- 5 电子设备在分配业务密钥信息的同时，将业务密钥信息进行存储，其中每个业务密钥信息包括一个密钥标识以及与密钥标识对应的对称密钥。电子设备在接收到业务接入进程发送的业务请求时，在已分配的所有对称密钥中，查询与业务请求中的密钥标识对应的对称密钥。若未查询到与密钥标识对应的对称密钥，则可以发送相应的错误信息给业务接入进程，以使业务接入进程在接收到错误信息时重新发送业务请求。

- 10 在一些实施例中，步骤 304 之后，还包括：当查询到的对称密钥的使用参数满足过期参数条件时，将过期信息发送至业务接入进程，以使业务接入进程在接收到过期信息时重新发送鉴权请求；其中，使用参数包括使用次数以及使用时长中的至少一种。

对于业务密钥信息，可以设定过期参数条件，当某个业务密钥信息的使用参数满足过期参数条件时，电子设备可以对该业务密钥信息进行更新，其中，过期参数条件可以是最大使用次数以及有效使用时长中的至少一种，当然也可以是其他的条件，对此不做限定。

- 15 电子设备在根据业务请求中的密钥标识进行查询后，若查询到的对称密钥的使用参数（等同于对称密钥所属的业务密钥信息的使用参数）满足过期参数条件，则将过期信息发送至业务接入进程，以使业务接入进程在接收到过期信息时重新发送鉴权请求，从而请求更新后的业务密钥信息。通过上述方式，能够实现业务密钥信息的动态更新，进一步提升业务通信的安全性。

在步骤 305 中，根据查询到的对称密钥，对业务请求中的加密后的请求数据进行解密处理。

在步骤 306 中，对解密处理得到的请求数据进行响应处理，得到响应数据。

- 20 这里，可以根据实际的业务策略或者规则对解密处理得到的请求数据进行响应处理，得到响应数据。例如，解密处理得到的请求数据是一个视频链接，则可以获取与该视频链接对应的视频数据，作为响应数据。值得说明的是，进行的响应处理也可以是将解密处理得到的请求数据进行存储，即并非一定要得到响应数据，这里仅以得到响应数据的情况进行示例说明。

- 25 在一些实施例中，业务请求还包括时间戳及第一校验信息；其中，第一校验信息是业务接入进程对密钥标识、对称密钥、时间戳、以及利用对称密钥加密后的请求数据进行哈希处理得到的；还可以通过这样的方式实现上述的对解密处理得到的请求数据进行响应处理：对密钥标识、查询到的对称密钥、时间戳、以及加密后的请求数据进行哈希处理，得到第二校验信息；当第一校验信息与第二校验信息之间的匹配结果为匹配成功时，对解密处理得到的请求数据进行响应处理。

- 30 这里，业务接入进程可以对密钥标识、对称密钥、当前的时间戳、以及利用对称密钥加密后的请求数据进行哈希处理，得到第一校验信息，并将第一校验信息添加至业务请求中，即业务请求包括密钥标识、利用对称密钥加密后的请求数据、时间戳以及第一校验信息。电子设备在通过业务请求中的密钥标识查询得到相应的对称密钥后，对业务请求中的密钥标识、查询到的对称密钥、业务请求中的时间戳、以及业务请求中的加密后的请求数据进行哈希处理，得到第二校验信息。其中，哈希处理所采用的算法可以预先约定，保证算法统一。

- 35 当第一校验信息与第二校验信息相同，即匹配成功时，证明业务请求中的数据未经篡改，此时电子设备对解密处理得到的请求数据进行响应处理；当第一校验信息与第二校验信息不同，即匹配失败时，证明业务请求中的数据经过了篡改，电子设备可以将相应的错误信息发送给业务接入进程，以使业务接入进程在接收到错误信息时重新发送业务请求。值得说明的是，在本申请实施例中，可以先比对第一校验信息与第二校验信息，再根据查询到的对称密钥对业务请求中的加密后的请求数据进行解密处理；也可以先根据查询到的对称密钥对业务请求中的加密后的请求数据进行解密处理，再比对第一校验信息与第二校验信息。通过上述方式，能够准确判断业务请求中的数据是否经过篡改，从而确定是否进行响应处理。

在步骤 307 中，根据公钥对响应数据进行加密处理，并将加密后的响应数据发送至业务接入进程，以使业务接入进程根据私钥对加密后的响应数据进行解密处理。

- 45 电子设备在通过响应处理得到响应数据后，根据业务接入进程发送的公钥对响应数据进行加密处理，并将加密后的响应数据发送至业务接入进程。如此，业务接入进程在接收到加密后的响应数据时，便可根据存储的私钥对其进行解密处理，得到响应数据，完成一次加密业务通信。

如图 4C 所示，本申请实施例结合业务接入进程生成的非对称密钥对、以及电子设备下发的业务密钥信息进行加密业务通信，能够有效保障业务通信的安全性和隐私性。

- 50 在一些实施例中，参见图 4D，图 4D 是本申请实施例提供的业务通信方法的一个流程示意图，图 4A 示出的步骤 104 之后，还可以在步骤 401 中，接收业务接入进程发送的凭证请求；其中，凭

证请求是业务接入进程在拦截到应用进程的业务请求时发送的，业务请求的目的地址为业务服务器的地址。

在本申请实施例中，业务接入进程可以用于拦截应用进程（例如与业务接入进程运行于同一终端设备上的应用进程）发送的业务请求，其中，该业务请求的目的地址为业务服务器的地址，这里的地址可以包括 IP 地址（或域名）及端口。这里，对拦截业务请求的方式不做限定，例如可以通过特定的虚拟网卡实现拦截。

业务接入进程在拦截到业务请求时，生成凭证请求，该凭证请求的内容可以包括业务请求中的源地址（即应用进程的地址）及目的地址，还可以包括应用进程的进程识别号（Process Identification, PID）。

10 在步骤 402 中，对应用进程进行同步验证处理。

电子设备在接收到业务接入进程发送的凭证请求时，对凭证请求所指向的应用进程进行同步验证处理。本质上来说，凭证请求也是业务接入进程发送的业务请求中的一种，凭证请求所请求的响应数据即为业务凭证。

15 在一些实施例中，还可以通过这样的方式实现上述的对应用进程进行同步验证处理：确定应用进程所在设备中处于登录态的用户账号，并获取用户账号对应的可信应用进程的进程信息、以及用户账号对应的可访问的业务服务器的地址；将应用进程的进程信息与可信应用进程的进程信息之间的匹配结果，作为第三验证处理结果；将业务请求所请求的业务服务器的地址与可访问的业务服务器的地址之间的匹配结果，作为第四验证处理结果；获取应用进程所在设备的设备信息，并将设备信息及设备安全条件之间的匹配结果，作为第五验证处理结果；根据第三验证处理结果、第四验证处理结果及第五验证处理结果，确定对应用进程的同步验证处理结果。

20 这里，提供了对应用进程进行同步验证处理的一种示例。首先，可以根据实际应用场景，针对多个用户账号中的每个用户账号，设定用户账号对应的可信应用进程以及可访问的业务服务器，电子设备可以将可信应用进程的进程信息以及可访问的业务服务器的地址存储在本地。

25 电子设备在接收到业务接入进程发送的凭证请求时，可以根据凭证请求中的 PID 确定相应的应用进程，确定应用进程所在设备中处于登录态的用户账号，同时获取应用进程所在设备的设备信息。针对处于登录态的用户账号，当应用进程的进程信息与该用户账号对应的任意一个可信应用进程的进程信息相同，即匹配成功时，确定第三验证处理结果为验证成功；当应用进程的进程信息与该用户账号对应的所有可信应用进程的进程信息均不同时，确定第三验证处理结果为验证失败。同样针对处于登录态的用户账号，当凭证请求中的目的地址与该用户账号对应的任意一个可访问的业务服务器的地址相同，即匹配成功时，确定第四验证处理结果为验证成功；当凭证请求中的目的地址与该用户账号对应的所有可访问的业务服务器的地址均不同时，确定第四验证处理结果为验证失败。

30 此外，还将设备信息与设定的设备安全条件进行匹配，若设备信息符合设备安全条件，即匹配成功，则确定第五验证处理结果为验证成功；否则，确定第五验证处理结果为验证失败。这里，对设备安全条件不做限定，例如可以是设备信息中不包括病毒和系统漏洞。

35 最后，当第三验证处理结果、第四验证处理结果及第五验证处理结果均为验证成功时，确定对应用进程的同步验证处理结果为验证成功；当第三验证处理结果、第四验证处理结果及第五验证处理结果中的任意一个为验证失败时，确定对应用进程的同步验证处理结果为验证失败。值得说明的是，也可以根据上述的第三验证处理结果、第四验证处理结果及第五验证处理结果中的至少一个，来确定对应用进程的同步验证处理结果，从而提升同步验证处理的效率。通过上述方式，能够提升得到的同步验证处理结果的准确性和有效性。

40 在一些实施例中，步骤 401 之后，还包括：周期性地应用进程的进程信息与进程信息黑名单进行匹配，并将得到的匹配结果作为对应用进程的异步验证处理结果；根据对应用进程的异步验证处理结果，控制与业务接入进程之间用于承载加密业务通信的通信连接。

45 除了对应用进程进行同步验证处理之外，还可以对应用进程进行异步验证处理。这里，对应用进程进行异步验证处理的过程，与上述对业务接入进程进行异步验证处理的过程类似。在得到对应用进程的异步验证处理结果后，若该异步验证处理结果为验证成功，则维持与业务接入进程之间的通信连接；若该异步验证处理结果为验证失败，则断开与业务接入进程之间的通信连接。

在一些实施例中，在任意步骤之间，还包括：向区块链网络发送查询请求，以查询区块链中存储的黑名单。

50 本申请实施例也可结合区块链技术实现。在本申请实施例中，区块链可用于存储黑名单，包括但不限于上述的签名方黑名单、证书信息黑名单及进程信息黑名单。下面，以电子设备接入区块链

网络以实现查询黑名单为例,说明区块链网络的示例性应用。

为了便于理解,以图 2 示出的架构进行说明,电子设备 700 接入区块链网络 500,成为区块链网络 500 的客户端节点。电子设备 700 在需要查询黑名单时,将查询请求以交易形式发送至区块链网络 500,在交易中指定实现查询操作需要调用的智能合约以及向智能合约传递的参数,交易还携带了电子设备 700 签署的数字签名(例如,使用电子设备 700 的数字证书,对交易的摘要进行加密得到),并将交易广播到区块链网络 500。其中,数字证书可由电子设备 700 向认证中心 600 进行登记注册得到。

区块链网络 500 中的节点 510 在接收到交易时,对交易携带的数字签名进行验证,数字签名验证成功后,根据交易中携带的电子设备 700 的身份标识,确认电子设备 700 是否具有交易权限,数字签名和权限验证中的任何一个验证判断都将导致交易失败。验证成功后签署节点 510 自己的数字签名,并继续在区块链网络 500 中广播。

区块链网络 500 中具有排序功能的节点 510 接收到验证成功的交易后,将交易填充到新的区块中,并广播到区块链网络 500 中提供共识服务的节点。

区块链网络 500 中的提供共识服务的节点 510 对新区块进行共识过程以达成一致,提供账本功能的节点 510 将新区块追加到区块链的尾部,并执行新区块中的交易:对于查询黑名单的交易,从状态数据库中查询黑名单,并将查询到的黑名单发送至电子设备 700。值得说明的是,状态数据库以键值对的形式存储数据,且状态数据库存储的数据通常与区块链存储的数据相同,在响应查询交易时,可以优先根据状态数据库中的数据响应,从而提升响应效率。当然,在不存在状态数据库的情况下,可以直接根据区块链中的数据进行响应。由于区块链中的数据具有不可篡改的特性,故通过上述方式,保证了获取到的黑名单的准确性。另外,区块链中存储的数据并不限于黑名单,例如还可以包括分配的业务密钥信息、用户账号对应的可信应用进程的进程信息、用户账号对应的可访问的业务服务器的地址等。

在步骤 403 中,根据对应用进程的同步验证处理结果,确定为应用进程分配的业务凭证及网关地址,并将业务凭证及网关地址发送至业务接入进程,以使业务接入进程将业务凭证及业务请求发送至网关地址对应的业务网关。

例如,当对应用进程的同步验证处理结果为验证成功时,电子设备确定为应用进程分配的业务凭证及网关地址,并将业务凭证及网关地址发送至业务接入进程;对应用进程的同步验证处理结果为验证失败时,可以断开与业务接入进程之间的通信连接,或者通知业务接入进程与业务服务器进行直连。这里的网关地址是指有权限访问业务服务器(这里指应用进程的业务请求所请求的业务服务器)的业务网关的地址,在本申请实施例中,可以针对不同的业务服务器设定不同的有权限访问的业务网关,也可以统一设定有权限访问的业务网关,对此不做限定。电子设备可以将分配的业务凭证存储在本地,以便后续验证,其中,对业务凭证的类型不做限定,例如可以是一个随机的字符串。

业务接入进程在接收到业务凭证及网关地址时,可以首先将业务凭证发送至网关地址对应的业务网关。业务网关在接收到业务凭证时,将业务凭证发送至电子设备以进行验证处理,若业务凭证与电子设备中存储的业务凭证相同,即验证成功,则业务网关可以将验证成功的信息发送至业务接入进程。业务接入进程在接收到验证成功的信息时,将应用进程的业务请求发送至业务网关,以使业务网关将业务请求转发至业务服务器,如此,业务服务器便可对接收到的业务请求中的请求数据进行响应处理。对于业务服务器通过响应处理得到的响应数据,按照相反的通信方向发送给应用进程即可。此外,若业务网关对接收到的业务凭证验证失败,则业务网关不支持对业务接入进程发送过来的业务请求进行转发,例如,业务网关可以与业务接入进程断开通信连接。

值得说明的是,在上述的步骤 401 至步骤 403 中,业务接入进程与电子设备之间的交互,均属于加密业务通信的范畴。

如图 4D 所示,业务接入进程可以作为应用进程的代理,实现对业务服务器的访问,适用于零信任场景,如零信任无边界办公的场景。

在一些实施例中,可以通过业务通信系统来实现业务通信方法,其中,业务通信系统包括业务接入客户端、安全客户端及安全服务器。参见图 4E,图 4E 是本申请实施例提供的业务通信方法的一个流程示意图,将结合图 4E 示出的步骤进行说明。

在步骤 501 中,安全客户端接收业务接入进程发送的鉴权请求。

这里,安全客户端及业务接入客户端可以部署于同一个终端设备中,也可以部署在不同的终端设备中。

在步骤 502 中, 安全客户端对业务接入进程进行同步验证处理。

这里, 对业务接入进程进行的同步验证处理可以包括对进程路径和签名信息的验证处理。值得说明的是, 在一些实施例中, 业务接入客户端可以是安全客户端的一个组件, 安全客户端在对业务接入进程的进程路径进行验证处理时, 安全目录可以是安全客户端的安装目录。

5 在步骤 503 中, 安全服务器对业务接入进程进行异步验证处理。

这里, 安全客户端可以通知安全服务器对业务接入进程进行异步验证处理, 安全服务器可以将对业务接入进程的异步验证处理结果返回给安全客户端。其中, 安全客户端可以将业务接入进程的进程信息发送至安全服务器, 以便于进行异步验证处理。

10 在步骤 504 中, 安全客户端根据对业务接入进程的同步验证处理结果, 确定为业务接入进程分配的业务密钥信息。

例如, 当对业务接入进程的同步验证处理结果为验证成功时, 安全客户端向安全服务器请求业务密钥信息, 即业务密钥信息的实际生成方可以是安全服务器。安全客户端可以将获取到的业务密钥信息存储在本地, 以便后续进行加密业务通信。

15 在步骤 505 中, 安全客户端将业务密钥信息发送至业务接入进程, 以基于业务密钥信息进行与业务接入进程之间的加密业务通信。

20 在一些实施例中, 步骤 505 之后, 还包括: 安全客户端接收业务接入进程发送的凭证请求; 其中, 凭证请求是业务接入进程在拦截到应用进程的业务请求时发送的, 业务请求的目的地址为业务服务器的地址; 安全客户端通知安全服务器对应用进程进行同步验证处理; 安全服务器根据对应用进程的同步验证处理结果, 确定为应用进程分配的业务凭证及网关地址, 并将业务凭证及网关地址通过安全客户端发送至业务接入进程; 业务接入客户端通过业务接入进程, 将业务凭证及业务请求发送至网关地址对应的业务网关; 其中, 业务网关用于对接收到的业务凭证进行验证处理, 并在对业务凭证的验证处理结果为验证成功时, 将接收到的业务请求发送至业务服务器; 业务服务器用于对接收到的业务请求中的请求数据进行响应处理。

25 例如, 安全客户端、业务接入进程及应用进程可运行于同一个终端设备中。安全客户端可以将获取到的凭证请求、应用进程所在设备中处于登录态的用户账号、应用进程的进程信息、以及应用进程所在设备的设备信息发送至安全服务器, 以使安全服务器对应用进程进行同步验证处理, 其中, 处于登录态的用户账号可以是指安全客户端中处于登录态的用户账号。

在一些实施例中, 安全服务器还可以对应用进程进行异步验证处理。

30 在步骤 506 中, 安全客户端根据安全服务器对业务接入进程的异步验证处理结果, 控制与业务接入进程之间用于承载加密业务通信的通信连接。

如图 4E 所示, 本申请实施例可以通过业务接入客户端、安全客户端及安全服务器来实现加密业务通信, 其中, 业务接入客户端可以作为安全客户端的一个组件, 便于安装至终端设备 (例如企业员工持有的终端设备) 中, 适用于零信任无边界办公等场景。

35 下面, 将说明本申请实施例在一个实际的应用场景中的示例性应用, 为了便于理解, 以零信任无边界办公的场景进行示例说明, 其中, 零信任 (Zero Trust) 是指不信任网络内部和外部的任何用户/设备/应用。在本申请实施例中, 业务通信系统可以包括安全客户端、安全服务器以及安全管理端, 其中, 安全管理端用于配置在安全客户端和/或安全服务器中应用的策略、规则等, 安全管理端可以提供人机交互界面 (如 Web 界面), 从而便于管理员 (如企业管理员) 进行配置。

40 例如, 管理员可以在安全管理端配置零信任策略, 其中, 零信任策略包括用户账号可使用的应用 (可信应用) 以及可访问的业务系统。作为示例, 参见图 5 所示的安全管理端的策略管理界面的一个示意图。在图 5 中, 零信任策略可以从组织架构中的用户账号 (或用户账号组)、可信应用、业务系统 (对应上文的业务服务器) 这三方面进行配置, 下面进行分别说明。

45 1) 用户账号或用户账号组: 在本申请实施例中, 零信任策略的粒度可以为单个用户账号。如果为某个用户账号组配置零信任策略, 则该用户账号组包括的所有用户账号可以共享同一个零信任策略。

2) 可信应用: 终端设备可访问内部业务系统的应用载体, 该应用载体由安全管理端授信。在本申请实施例中, 应用进程的进程信息可以包括应用名称、进程名称、应用 MD5 值及签名信息等。

50 3) 业务系统: 可以用于提供企业内部的业务资源、数据、开发环境、测试环境、运维环境和正式生产环境等, 是访问主体 (人/终端设备/应用) 需要访问的对象, 即业务系统是访问客体。在一些情况中, 业务系统可以是业务服务器 (或者是业务服务器集群)。

值得说明的是, 用户账号组是企业组织架构里的一个节点, 在图 5 中, 节点从上至下依次为全

网账号、用户账号组及用户账号，即全网账号是所有用户账号组的父节点，用户账号组是该用户账号组包括的所有用户账号的父节点。举例来说，图 5 中的“lemon”用户账号属于“testgroup”用户账号组，“testgroup”用户账号组又属于全网账号。

5 本申请实施例还提供了如图 6 所示的进程信息（也称进程特征）的一个示意图，在图 6 中，示出了一个会议应用进程的进程信息，可以包括进程名称、应用名称、操作系统（即用于运行应用进程的终端设备的操作系统）、厂家、签名信息、送检结果（即异步验证处理结果）、版本、MD5 值及 SHA256。

10 在配置业务系统之前，管理员可以在安全管理端的零信任网关（对应上文的业务网关）界面进行零信任网关的相关配置，作为示例，提供了如图 7 所示的安全管理端的零信任网关界面的示意图，以及如图 8 所示的零信任网关的配置界面示意图，图 8 示出的配置界面可以是对图 7 示出的某个网关（如图 4）进行触发而显示的，当然对此并不做限定。管理员可以根据实际需求，对零信任网关的名称、IP 地址、域名、端口、以及优先访问的 IP 地址段等属性进行配置。其中，若对某个网关配置了优先访问的 IP 地址段，则当访问主体（如某个终端设备）的 IP 地址与该优先访问的 IP 地址段匹配成功（即落入优先访问的 IP 地址段）时，可以优先使用该网关转发该访问主体的业务请求；

15 当该访问主体的 IP 地址与该优先访问的 IP 地址段匹配失败时，可以选择使用该网关或者不使用该网关，根据实际应用场景而定。

本申请实施例还提供了如图 9 所示的安全管理端的业务系统界面的示意图。在图 9 中，可以查询具体的某个业务系统，还可以支持添加、拷贝及删除（如批量删除）业务系统。根据实际应用场景的不同，还可以创建不同的系统组合，其中，系统组合包括至少一个业务系统。在图 9 中，以系统组合 1 为例，示出了系统组合 1 中的业务系统 1 至 4。

20

作为示例，本申请实施例还提供了如图 10、图 11 及图 12 所示的业务系统的配置界面的示意图。对于每个业务系统，可以配置其 IP 地址、所属的 IP 地址段、域名以及端口，还可以根据实际的业务情况，配置可访问该业务系统的网关（即具有访问权限的网关），并自定义可访问该业务系统的多个网关的顺序。举例来说，对于可访问某个业务系统的 4 个网关，按照从前至后的顺序依次为网关 1、网关 2、网关 3 及网关 4，则访问主体在访问该业务系统时，同样按照该顺序进行依次访问，例如首先访问网关 1，若访问失败则访问网关 2，以此类推。

25

对业务系统及可信应用配置完成之后，即可针对用户账号或用户账号组下发相关的配置，即进行零信任策略的下发。作为示例，本申请实施例提供了如图 13 所示的安全管理端的策略管理界面的一个示意图，在图 13 中，为用户账号“123”配置了可信应用（即图 13 中的“xx 会议”）、以及通过可信应用可访问的业务系统（即图 13 中的“系统组合 2”）。值得说明的是，本申请实施例可以提供继承功能，即是将配置随着从上到下的用户账号层级关系自动复制下来，避免管理员反复勾选操作。例如，用户账号“123”属于用户账号组“test”，则可以将对用户账号组“test”的配置，复制给用户账号“123”。当然，也可以关闭继承功能，以对用户账号“123”进行针对性配置。

30

安全管理端可以将管理员配置的零信任策略部署至安全服务器中，当然也可以部署至安全客户端中，例如将部分零信任策略部署至安全客户端中。安全客户端用于安装至作为访问主体的终端设备中，这里的终端设备如企业员工使用的个人电脑等。作为示例，本申请实施例提供了如图 14 所示的安全客户端的界面示意图，在图 14 中，提供了扫码登录及账号登录两种方式，其中，账号登录是指通过输入用户名和密码的方式进行登录。值得说明的是，安全客户端还可以集成病毒查杀、合规检测、漏洞修复及电脑工具等功能，又例如，如图 15 所示，在成功登录之后，安全客户端可以提供实时防护策略、杀毒防护引擎及安全加固策略，可以根据实际应用场景对安全客户端进行功能的添加和删除。

35

40

本申请实施例还提供了如图 16 所示的安全客户端的界面示意图，在登录后，安全客户端可以获得针对处于登录态的用户账号配置的零信任策略，从该零信任策略中获取可信应用并进行显示。例如，在图 16 中，示出了与用户账号“lemon”对应的可信应用“xx 会议”。依据安全管理端下发的零信任策略，终端设备可以根据配置的可信应用来访问配置的业务系统，适用于无边界办公，如远程办公的场景。

45

接下来，以无边界办公的场景为例，说明实际的访问过程。本申请实施例提供了如图 17 所示的访问过程示意图，在图 17 中，业务通信系统可以充当零信任网络安全服务提供方，基于访问代理（对应上文的业务接入客户端）和零信任网关为访问主体提供统一入口，以使访问主体通过统一入口对访问客体进行访问，即通过网络发送业务请求。其中，业务通信系统可以为统一入口提供鉴权操作，只有通过鉴权的业务请求才能由访问代理转发给零信任网关，从而通过零信任网关代理对实际业务

50

系统的访问。

作为示例,本申请实施例还提供了如图 18 所示的访问过程示意图,图 18 中的核心模块包括安全客户端、安全服务器、访问代理及零信任网关(智能网关),接下来进行分别说明。

1)安全客户端:安装在访问主体(如企业员工的终端设备)中的 Agent,可以用于验证处于登录态的用户账号是否可信、终端设备是否可信以及应用是否可信。安全客户端可以将应用进程的进程特征(对应上文的进程信息)发送给安全服务器,即进行进程送检(对应上文的针对应用进程的异步验证处理)。

2)访问代理:也称代理客户端,用于劫持终端设备的流量,即拦截终端设备中的应用进程发送的业务请求,其中,可以通过 TUN/TAP 虚拟网卡实现流量劫持,但劫持方式并不限于此。

3)零信任网关:可以部署在企业应用程序和数据资源的入口,负责对每一个用于访问企业内部业务系统的业务请求进行验证、授权及转发。其中,零信任网关可以通过软件形式构建。

4)安全服务器:负责通过策略控制引擎(策略中心)对业务流量进行安全调度,按照用户账号-终端设备-应用的颗粒度对应用进程的业务请求进行授权。安全服务器可以用于验证安全客户端中处于登录态的用户账号是否可信,还可以用于验证设备硬件信息和设备安全状态,还可以用于检测应用进程是否可信,如是否有漏洞,是否有病毒木马等。安全服务器可以周期性地向云服务器发起文件送检,若识别出发送业务请求的应用进程为恶意进程,则通知安全客户端执行异步阻断操作,例如断开与访问代理之间的通信连接,其中,云服务器用于提供恶意进程的云检测服务(或称云查询服务、云识别服务等)。

举例来说,终端设备(访问主体)可以通过应用进程发送针对访问客体的业务请求,访问代理在劫持到该业务请求时,向安全客户端发送票据请求(对应上文的凭证请求),票据请求包括业务请求中的源 IP 地址(或者源域名)、源端口、目的 IP 地址(或者目的域名)及目的端口,还包括应用进程对应的 PID,其中,票据即对应上文的业务凭证,PID 用于唯一地标识应用进程。安全客户端在接收到访问代理发送的票据请求时,根据其中的 PID 采集应用进程的 MD5 值、进程路径、进程最近修改时间、版权信息及签名信息等,连同票据请求中的源 IP 地址(或者源域名)、源端口、目的 IP 地址(或者目的域名)以及目的端口,还连同安全客户端中处于登录态的用户账号的信息等,向安全服务器申请票据。安全服务器在接收到这些信息后,获取与安全客户端中处于登录态的用户账号对应的零信任策略,并根据获取到的零信任策略,确定该用户账号是否具有通过应用进程(指发起业务请求的应用进程)访问业务系统(指业务请求所请求的业务系统)的权限,若具有权限,则安全服务器可以将生成的票据、票据最大使用次数以及票据有效使用时长发送至安全客户端,以使安全客户端转发至访问代理。

在票据申请的过程中,可以包括同步和异步两个方面。例如,在同步验证处理中,安全客户端向安全服务器申请票据,安全服务器对应用进程的进程特征进行同步验证处理,例如验证应用进程是否为可信应用、用户账号是否拥有通过应用进程访问相应的业务系统的权限等。若安全服务器得到的同步验证处理结果为验证成功,则安全服务器正常响应票据、票据最大使用次数以及票据有效使用时长给安全客户端,由安全客户端发送给访问代理,例如安全客户端可以通过本地进程通信(即 Socket 通信连接)的方式发送至访问代理。

在异步验证处理中,安全服务器可以根据应用进程的进程特征,周期性地向特定的云服务器发送文件送检请求。若安全服务器从云服务器获取到应用进程为恶意进程,则可以通知安全客户端执行异步阻断操作,例如断开与访问代理之间的通信连接。

访问代理在获取到票据后,可以向零信任网关发送票据,以使零信任网关进行验证。例如,访问代理可以将票据添加在一个 HTTPS 请求的 Authorization 首部字段中,并将该 HTTPS 请求发送至零信任网关。零信任网关在接收到该 HTTPS 请求时,解析出 Authorization 首部字段中的票据,并向安全服务器校验票据。若零信任网关对票据验证成功,则零信任网关与访问代理成功建立连接,之后访问代理便可将原始的业务请求(这里指应用进程发送的业务请求)发送给零信任网关,由零信任网关将业务请求转发至对应的业务系统,以代理应用进程对业务系统的网络访问;若零信任网关对票据验证失败,则断开访问代理与零信任网关之间的通信连接,由访问代理直接将原始的业务请求发送至对应的业务系统,在零信任无边界办公的场景中,为了保证网络访问的安全性,这种访问通常会失败。

在本申请实施例中,安全服务器既能够通过单一部署方式,适应中型企事业单位和政府,也能够通过分布式级联部署方式,适应大型企业集团和多级垂直政府电子政务系统。

例如,为了保障零信任网络访问核心业务的稳定运行,可以采取零信任核心服务异地多活的方

案。核心的基础业务部署在总控节点，各个不同的业务部署在不同的二级业务节点中。作为示例，本申请实施例提供了如图 19 所示的级联部署的示意图，在图 19 中，安全服务器采取级联部署模式，包括总控节点（总控服务器）和二级业务节点（服务器节点）。总控节点部署有核心的基础服务，例如心跳服务、策略同步服务及设备管控服务等。总控节点的配置和数据可以周期性地同步给各二级业务节点；同时，当二级业务节点有数据和配置需要更改时，可以通知总控节点去修改，总控节点修改完成之后，同步给相应的二级业务节点即可。

在本申请实施例中，访问代理的业务接入进程可以调用安全客户端中的接口来实现数据传递，为了保证安全性，可以对访问代理的业务接入进程进行同步验证处理及异步验证处理，以下进行分别说明。

1) 同步验证处理。

访问代理与安全客户端建立 Socket 连接后，安全客户端通过访问代理的 IP 地址和端口号，获取业务接入进程的 PID。然后，安全客户端根据业务接入进程的 PID 获取业务接入进程的进程路径，检测该进程路径是否位于安全客户端的安装目录（对应上文的安全目录）内。在本申请实施例中，访问代理可以是业务通信系统的一个组件，在正常安装后，访问代理通常位于安全客户端的安装目录中。并且，在本申请实施例中，可以对安全客户端的安装目录进行目录保护，阻止其他非业务通信系统的服务或进程对该安装目录进行操作。

通过检测业务接入进程的进程路径是否在安全客户端的安装目录内，可以初步检测接入进程的合法性。若业务接入进程的进程路径位于该安装目录内，则可以进一步获取业务接入进程的进程文件的签名信息，并进行验证处理。在企业的安全管控强度较低的情况下，可以放松（授权）未具有数字签名的业务接入进程对安全客户端的接口访问；在企业的安全管控强度较高的情况下，可限制只有具备数字签名的业务接入进程才能访问安全客户端的接口。根据安全管控强度的不同，可以实行以下几种针对签名信息的验证处理。

①检测业务接入进程的进程文件是否具备数字签名，若具备数字签名，即确定对签名信息验证成功。

②安全客户端本地对数字签名进行有效性验证处理，例如调用 WinVerifyTrust 的 API 进行有效性验证处理。若有效性验证处理为验证成功，则确定对签名信息验证成功。

③在安全客户端本地进行有效性验证处理的基础上，获取数字签名的签名人姓名（对应上文的签名方），并将签名人姓名与签名人黑名单进行匹配。当签名人姓名不位于签名人黑名单中，即匹配失败时，确定对签名信息验证成功。

④安全客户端获取业务接入进程的进程文件的证书链信息（对应上文的证书信息），例如，证书链信息可以包括摘要算法、根证书信息、中级证书信息、签名证书信息、签名状态、签名人姓名、时间戳及签名验证错误信息。其中，根证书信息包括根证书的名称、序列号及到期时间等，中级证书信息及签名证书信息以此类推。另外，签名状态可以用于表示数字签名可用、数字签名被篡改、证书不可信、证书过期、证书吊销及其他错误中的一种。安全客户端在获取到证书链信息后，可以将证书链信息与证书链信息黑名单（对应上文的证书信息黑名单）进行匹配，若匹配失败，则确定对签名信息验证成功。

以上 4 种方式的安全管控强度逐渐增强，相应地，验证处理所耗费的时长也会变长，可以根据实际应用场景中的安全管控的强度需求以及效率需求，选用至少一种方式实现对签名信息的验证处理。

安全客户端在对业务接入进程的进程路径和签名信息均验证成功时，可以先允许业务接入进程的接口调用，进入接口鉴权环节。

2) 异步验证处理。

为了充分验证业务接入进程是否是恶意进程，安全客户端可以异步获取业务接入进程的进程特征，并发送至安全服务器，以进行异步验证处理。其中，业务接入进程的进程特征包括但不限于业务接入进程的 MD5 值、Hash 值、进程路径及签名信息（包括证书链信息）。安全服务器接收到业务接入进程的进程特征后，可以周期性地向云服务器发送文件送检请求，云服务器根据存储的不断更新的进程特征黑名单（对应上文的进程信息黑名单），判断业务接入进程是否为恶意进程。若云服务器判断出业务接入进程为恶意进程，则安全服务器会通知安全客户端，以使安全客户端断开与业务接入进程之间的 Socket 连接。

作为示例，本申请实施例提供了如图 20 所示的异步验证处理的示意图，图 20 中的编号表示异步验证处理的步骤，将结合示出的各个步骤进行说明。

①访问代理尝试调用安全客户端的相关接口，即向安全客户端发送鉴权请求。

②安全客户端对业务接入进程进行同步验证处理（包括对进程路径及签名信息的验证处理），若同步验证处理结果为验证成功，则采集业务接入进程的进程特征。

③安全客户端将业务接入进程的进程特征发送至安全服务器，即进行异步送检。

5 ④安全服务器将业务接入进程的进程特征发送至云服务器，以便云服务器判断业务接入进程是否为恶意进程。

⑤云服务器向安全服务器发送异步验证处理的结果，即业务接入进程是否为恶意进程。

⑥若业务接入进程为恶意进程，则安全服务器通知安全客户端执行连接阻断操作。

⑦安全客户端主动断开与业务接入进程之间的 Socket 连接。

10 接下来，对安全客户端与业务接入进程之间的接口鉴权环节进行阐述，为了便于理解，以步骤形式进行说明。

1) 业务接入进程生成非对称密钥对。

业务接入进程在调用安全客户端的接口时，需要获取安全客户端针对此次调用的授权。首先，由业务接入进程生成非对称密钥对，非对称密钥对中的私钥 `privatekey` 可以由业务接入进程进行内
15 存加密保存，非对称密钥对中的公钥 `pubkey` 用于发送给安全客户端。本申请实施例对生成非对称密
钥对的方式不做限定，例如可以通过 RSA 算法进行生成。

2) 业务接入进程调用安全客户端的鉴权接口。

这里，业务接入进程向安全客户端的鉴权接口发送鉴权请求，鉴权请求的示例如下。

GET `http://localhost:[port]/[path]/[req_acc]/[pubkey]`

20 其中，[port]表示安全客户端所提供的安全服务的监听端口，[path]表示服务路由（对应上文的鉴
权请求地址），[req_acc]表示请求标识，即用于表示请求鉴权。另外，上述示例中的
{pubkey}=urlencode(base64encode(pubkey))，即{pubkey}是对业务接入进程生成的公钥 `pubkey` 进行
base64encode 之后，再进行 urlencode 得到的结果。在本申请实施例中，执行 base64encode 和 urlencode
的目的都是为了便于进行网络传输。

25 3) 安全客户端响应鉴权请求。

安全客户端在接收到鉴权请求时，通过上述的同步和异步相结合的方法检测业务接入进程的合
法性。在对业务接入进程的同步验证处理结果为验证成功时，安全客户端请求安全服务器生成对称
密钥 `aes_key`，用于后续的加密业务通信，其中，对称密钥可以通过 AES 算法或其他算法生成。安
全客户端通过鉴权请求中的 {pubkey} 对对称密钥进行加密，并将加密后的对称密钥发送至业务接入
30 进程，完成首次授权。

这里，安全客户端向业务接入进程返回的数据包的示例如下。

```
{
  "code": 0 // 针对鉴权请求的响应码，如果是 0 则表示响应正常
  "msg": "" // 响应描述信息
  "data": base64encode(encrypt(plain_data, pubkey))
}
```

其中，`plain_data` 表示数据包主体的明文信息，其内容的示例如下。

```
{
  "access_id": "id" // 访问 id，对应上文的密钥标识
  "access_key": base64encode(aes_key) // 访问动态密钥
}
```

其中，`encrypt(plain_data, pubkey)` 表示安全客户端根据业务接入进程发送的公钥 `pubkey` 对数
据包主体的明文信息 `plain_data` 进行加密后得到的结果，即 `encrypt` 表示加密处理。`access_key` 是安
全客户端对对称密钥 `aes_key` 执行 base64encode 得到的结果，这里，安全服务器可以将 `access_id` 与
45 `access_key` 之间的映射关系进行存储，还可以存储 `access_key` 的最大使用次数以及有效使用时长等
属性。安全客户端也可以将 `access_id` 及对应的 `access_key`（或者对应的 `aes_key`）存储在本地，以便
后续查询。

4) 业务接入进程接收并解析安全客户端响应的数据包。

业务接入进程接收到安全客户端返回的数据包时，解析出数据包中的响应码 `code`，若响应码 `code`
50 的值为 0，则表示处理正常，业务接入进程进一步通过 base64decode 解析出该数据包的 `data` 的 base64
明文，再通过内存中加密保存的私钥 `privatekey` 对 base64 明文进行解密处理，得到 `plain_data`。

然后,业务接入进程从 plain_data 中解析出 access_id 和 access_key,并加密存储在内存中。在本申请实施例中,access_id 和 access_key 可以是一对一的关系。

5) 业务接入进程调用安全客户端的业务接口。

这里,业务接口可以是用于申请票据的接口,当然也可以是其他的业务接口。当业务接入进程调用安全客户端的业务接口时,对发送给业务接口的原始参数 raw_param (对应上文的请求数据) 经过如下处理,形成 call_param。

①使用对称密钥 aes_key 对原始参数 raw_param 进行加密处理,即 encrypt(raw_param, aes_key)。其中,对原始参数 raw_param 的内容不做限定,可以根据实际业务来具体设定。

②针对第①步得到的结果执行 base64encode,即 base64encode(encrypt(raw_param, aes_key))。

③针对第②步得到的结果执行 urlencode,即 urlencode(base64encode(encrypt(raw_param, aes_key))),得到 call_param。

当业务接入进程调用安全客户端的业务接口时,业务接入进程同时生成自定义校验信息 signature (对应上文的第一校验信息),生成的步骤如下所示。

①业务接入进程从内存加密数据中取出 access_id,以作为第一部分(命名为 A)。

②对对称密钥 aes_key 执行 base64encode,得到第二部分(命名为 B),即 B=base64encode(aes_key)。

③对参数 call_param 执行 base64encode,得到第三部分(命名为 C),即 C=base64encode(call_param)。

④将当前的时间戳 timestamp 作为第四部分(命名为 D)。

⑤将上述 4 步得到的字符串相加,再计算 MD5 值,得到自定义校验信息 signature,即 signature=MD5(A+B+C+D)。

在得到自定义校验信息 signature 后,对其进行 base64encode 得到 Authorization 公共头信息,例如可以表示为:

Authorization:"IPN base64encode({"access_id": "", "signature": "", "timestamp": ""})"

其中,timestamp 表示在生成 signature 过程中的时间戳,即上述的 D。值得说明的是,业务接入进程向安全客户端发送的业务请求包括上述的 call_param 及 Authorization 公共头信息。

6) 安全客户端解析业务接入进程向业务接口发送的业务请求,并检测业务请求是否合法。

当安全客户端接收到业务请求时,对业务请求中的 call_param 进行 urlencode,然后进行 base64decode,得到 encrypt(raw_param, aes_key)。然后,使用 aes_key 对 encrypt(raw_param, aes_key) 进行解密处理得到 raw_param,即业务接入进程的原始参数。

对于业务请求中的 Authorization 公共头信息,安全客户端首先进行 base64decode 得到明文,即 access_id、自定义校验信息 signature 及时间戳 timestamp。然后,安全客户端根据 access_id 在本地进行查询,得到相应的 access_key (或者 aes_key),然后将查询到的 access_key 发送至安全服务器,以检测查询到的 access_key 是否过期。若查询到的 access_key 已过期,则安全客户端触发鉴权接口的调用重认证,即是使业务接入进程重新发送鉴权请求。

若查询到的 access_key 未过期,即未失效,则安全客户端根据业务请求中的 call_param、Authorization 公共头信息中的 access_id、Authorization 公共头信息中的 timestamp、以及查询到的 access_key,重新生成校验信息 Signature' (对应上文的第二校验信息),即 Signature' = MD5(access_id + base64encode(aes_key) + base64encode(call_param) + timestamp),这里的 base64encode(aes_key)即为查询到的 access_key。

安全客户端将生成的校验信息 Signature' 与业务接入进程发送的自定义校验信息 Signature 进行比较,若两者相同,则安全客户端确定业务请求合法;若两者不同,证明业务请求中的参数经过篡改或伪造,安全客户端确定业务请求不合法,业务接入进程对业务接口的调用失败。

7) 安全客户端对业务请求中的原始参数进行响应处理得到响应数据,并返回给业务接入进程。

若安全客户端确定出业务请求合法,即业务接入进程针对业务接口的调用合法,则安全客户端对业务请求中的原始参数 raw_param 进行响应处理,得到响应数据 resultdata,这里,对响应数据 resultdata 的具体内容不做限定,可以根据实际业务进行设定。安全客户端返回给业务接入进程的数据包的示例如下。

```
{
  "retcode": 0 // 针对业务请求的响应码
  "msg": "" // 响应描述信息,可以根据实际应用场景进行设定
  "data": base64encode(encrypt(resultdata, pubkey))
}
```

}

其中, retcode 表示针对业务接入进程发送的业务请求的响应码, 若 retcode 是 0, 则表示针对业务接口的调用成功; 若 retcode 是 1, 则表示针对业务接口的调用失败; 若 retcode 是 2, 则表示 access_key (或者 aes_key) 已过期, 需要业务接入进程重新发送鉴权请求, 即重新执行步骤 2)。

5 在本申请实施例, 可以在安全管理端周期性地更新接口鉴权所需的对称密钥, 在每次更新时, 安全服务器可以将更新前的所有 access_id 及 access_key (或者 aes_key) 设置为失效, 从而触发业务接入进程重新调用安全客户端的鉴权接口, 以获取更新后的 access_id 和 access_key。

针对 access_id 和 access_key (或者 aes_key), 本申请实施例可以在安全管理端配置不同的粒度, 例如可从业务接入进程本身、以及业务接入进程调用鉴权接口时发送的服务路由两方面进行区分。
10 举例来说, 业务接入进程 A 调用鉴权接口时的服务路由包括 path1、path2 和 path3, 业务接入进程 B 调用鉴权接口时的服务路由包括 path4 和 path5, 则粒度划分的方案可包括如下三种:

1) 安全服务器同时区分业务接入进程和服务路由, 即针对不同的业务接入进程分配不同的 access_id 和 access_key 组合对, 针对同一业务接入进程的不同服务路由也分配不同的 access_id 和 access_key 组合对。举例来说, 针对业务接入进程 A 的 path1、path2 及 path3, 分别分配三组不同的
15 access_id 和 access_key 组合对; 针对业务接入进程 B 的 path4 及 path5, 分别分配另外两组不同的 access_id 和 access_key 组合对。

2) 安全服务器仅区分业务接入进程, 不区分服务路由, 即针对不同的业务接入进程分配不同的 access_id 和 access_key 组合对, 针对同一业务接入进程的不同服务路由分配相同的 access_id 和 access_key 组合对。举例来说, 针对业务接入进程 A 的 path1、path2 及 path3, 统一分配一组 access_id
20 和 access_key 组合对; 针对业务接入进程 B 的 path4 及 path5, 统一分配另一组 access_id 和 access_key 组合对。

3) 安全服务器不区分业务接入进程及服务路由, 即针对所有业务接入进程的所有服务路由, 统一分配一组 access_id 和 access_key 组合对。

在实际应用场景中, 管理员可以根据企业的管控程度, 采用上述的任意一种粒度划分的方案。

25 针对 access_id 和 access_key (或者 aes_key) 的更换机制, 安全服务器可以配置最大使用次数以及有效使用时长中的至少一种, 从而实现 access_id 和 access_key 的定期更新, 提升安全性。

在本申请实施例中, 安全客户端通过对业务接入进程的进程路径及签名信息进行验证处理, 能够实现对业务接入进程的合法性的有效验证; 同时, 通过安全客户端的进程特征异步送检以及安全服务器的云查服务检测, 能够准确验证业务接入进程是否为恶意进程。本申请实施例既提升了对业务接入进程的验证准确率, 也提升了合规的业务接入进程接入安全客户端的效率。
30

此外, 本申请实施例提供了通过动态密钥和自定义校验信息验证的接口鉴权方案, 能够根据安全服务器下发的对称密钥进行加密业务通信, 避免密钥存储被破解的问题; 同时, 可以周期性地更新业务密钥信息, 从而加强接口调用的安全性; 还允许管理员根据各业务的管控程度, 通过不同的粒度划分方案来区分业务密钥信息, 提升对不同业务及不同场景的适用性。

35 下面继续说明本申请实施例提供的终端设备 400 实施为软件模块的示例性结构, 在一些实施例中, 如图 3 所示, 存储在存储器 450 的业务通信装置 455 中的软件模块可以包括: 接收模块 4551, 用于接收业务接入进程发送的鉴权请求; 验证模块 4552, 用于对业务接入进程进行同步验证处理, 对业务接入进程进行异步验证处理; 确定模块 4553, 用于根据对业务接入进程的同步验证处理结果, 确定为业务接入进程分配的业务密钥信息; 发送模块 4554, 用于将业务密钥信息发送至业务接入进
40 程, 以基于业务密钥信息进行与业务接入进程之间的加密业务通信; 连接控制模块 4555, 用于根据对业务接入进程的异步验证处理结果, 控制与业务接入进程之间用于承载加密业务通信的通信连接。

在一些实施例中, 验证模块 4552, 还用于: 将业务接入进程的进程路径与设定的安全目录之间的匹配结果, 作为第一验证处理结果; 对业务接入进程的签名信息进行验证处理, 得到第二验证处理结果; 根据第一验证处理结果及第二验证处理结果, 确定对业务接入进程的同步验证处理结果。

45 在一些实施例中, 验证模块 4552, 还用于根据签名信息是否包括数字签名的结果、数字签名的有效性验证处理结果、数字签名的签名方与签名方黑名单之间的匹配结果、以及签名信息中的证书信息与证书信息黑名单之间的匹配结果中的至少之一, 确定第二验证处理结果。

在一些实施例中, 验证模块 4552, 还用于: 确定签名信息中的数字签名、以及与数字签名对应的解密密钥; 根据解密密钥对数字签名进行解密处理, 得到业务接入进程的进程文件的第一哈希结果; 对业务接入进程的进程文件进行哈希处理, 得到第二哈希结果; 将第一哈希结果与第二哈希结果之间的匹配结果, 作为数字签名的有效性验证处理结果。
50

在一些实施例中,鉴权请求包括业务接入进程生成的非对称密钥对中的公钥;其中,非对称密钥对包括公钥以及与公钥对应的私钥;发送模块 4554,还用于:根据公钥对业务密钥信息进行加密处理;将加密后的业务密钥信息发送至业务接入进程,以使业务接入进程根据私钥对加密后的业务密钥信息进行解密处理。

- 5 在一些实施例中,业务密钥信息包括密钥标识及对称密钥;业务通信装置 455 还包括:业务接收模块,用于接收业务接入进程发送的业务请求;其中,业务请求包括密钥标识、以及利用对称密钥加密后的请求数据;查询模块,用于在已分配的对称密钥中,查询与业务请求中的密钥标识对应的对称密钥;解密模块,用于根据查询到的对称密钥,对业务请求中的加密后的请求数据进行解密处理;响应模块,用于对解密处理得到的请求数据进行响应处理,得到响应数据;加密发送模块,用于根据公钥对响应数据进行加密处理,并将加密后的响应数据发送至业务接入进程,以使业务接入进程根据私钥对加密后的响应数据进行解密处理。

- 10 在一些实施例中,业务请求还包括时间戳及第一校验信息;其中,第一校验信息是业务接入进程对密钥标识、对称密钥、时间戳、以及利用对称密钥加密后的请求数据进行哈希处理得到的;响应模块,还用于:对密钥标识、查询到的对称密钥、时间戳、以及加密后的请求数据进行哈希处理,得到第二校验信息;当第一校验信息与第二校验信息之间的匹配结果为匹配成功时,对解密处理得到的请求数据进行响应处理。

在一些实施例中,业务通信装置 455 还包括:过期处理模块,用于当查询到的对称密钥的使用参数满足过期参数条件时,将过期信息发送至业务接入进程,以使业务接入进程在接收到过期信息时重新发送鉴权请求;其中,使用参数包括使用次数以及使用时长中的至少一种。

- 20 在一些实施例中,鉴权请求包括鉴权请求地址;确定模块 4553,还用于执行以下任意一种处理:针对不同业务接入进程发送的鉴权请求地址分配不同的业务密钥信息,针对同一业务接入进程发送的不同鉴权请求地址分配不同的业务密钥信息;针对不同业务接入进程发送的鉴权请求地址分配不同的业务密钥信息,针对同一业务接入进程发送的不同鉴权请求地址分配相同的业务密钥信息;针对不同业务接入进程发送的鉴权请求地址分配相同的业务密钥信息,针对同一业务接入进程发送的不同鉴权请求地址分配相同的业务密钥信息。

在一些实施例中,验证模块 4552,还用于:周期性地将业务接入进程的进程信息与进程信息黑名单进行匹配,并将得到的匹配结果作为对业务接入进程的异步验证处理结果。

- 30 在一些实施例中,确定模块 4553,还用于:当对业务接入进程的同步验证处理结果为验证成功时,确定为业务接入进程分配的业务密钥信息;当对业务接入进程的同步验证处理结果为验证失败时,执行以下任意一种处理:通知业务接入进程重新发送鉴权请求;断开与业务接入进程之间用于传输鉴权请求的通信连接;中断对业务接入进程的异步验证处理。

在一些实施例中,连接控制模块 4555,还用于:当对业务接入进程的异步验证处理结果为验证成功时,维持与业务接入进程之间用于承载加密业务通信的通信连接;当对业务接入进程的异步验证处理结果为验证失败时,断开与业务接入进程之间用于承载加密业务通信的通信连接。

- 35 在一些实施例中,业务通信装置 455 还包括:加密通信模块,用于:在加密业务通信的过程中,执行以下处理:接收业务接入进程发送的凭证请求;其中,凭证请求是业务接入进程在拦截到应用进程的业务请求时发送的,业务请求的目的地址为业务服务器的地址;对应用进程进行同步验证处理;根据对应应用进程的同步验证处理结果,确定为应用进程分配的业务凭证及网关地址,并将业务凭证及网关地址发送至业务接入进程,以使业务接入进程将业务凭证及业务请求发送至网关地址对应的业务网关;其中,业务网关用于对接收到的业务凭证进行验证处理,并在对业务凭证的验证处理结果为验证成功时,将接收到的业务请求发送至业务服务器;业务服务器用于对接收到的业务请求中的请求数据进行响应处理。

- 40 在一些实施例中,加密通信模块还用于:确定应用进程所在设备中处于登录态的用户账号,并获取用户账号对应的可信应用进程的进程信息、以及用户账号对应的可访问的业务服务器的地址;将应用进程的进程信息与可信应用进程的进程信息之间的匹配结果,作为第三验证处理结果;将业务请求所请求的业务服务器的地址与可访问的业务服务器的地址之间的匹配结果,作为第四验证处理结果;获取应用进程所在设备的设备信息,并将设备信息与设备安全条件之间的匹配结果,作为第五验证处理结果;根据第三验证处理结果、第四验证处理结果及第五验证处理结果,确定对应用进程的同步验证处理结果。

- 50 在一些实施例中,加密通信模块还用于:周期性地将应用进程的进程信息与进程信息黑名单进行匹配,并将得到的匹配结果作为对应用进程的异步验证处理结果;根据对应用进程的异步验证处

理结果，控制与业务接入进程之间用于承载加密业务通信的通信连接。

5 本申请实施例提供了一种计算机程序产品或计算机程序，该计算机程序产品或计算机程序包括计算机指令（可执行指令），该计算机指令存储在计算机可读存储介质中。计算机设备的处理器从计算机可读存储介质读取该计算机指令，处理器执行该计算机指令，使得该计算机设备执行本申请实施例上述的业务通信方法

本申请实施例提供一种存储有可执行指令的计算机可读存储介质，其中存储有可执行指令，当可执行指令被处理器执行时，将引起处理器执行本申请实施例提供的方法，例如，如图 4A、图 4B、图 4C、图 4D 及图 4E 示出的业务通信方法。

10 在一些实施例中，计算机可读存储介质可以是 FRAM、ROM、PROM、EPROM、EEPROM、闪存、磁表面存储器、光盘、或 CD-ROM 等存储器；也可以是包括上述存储器之一或任意组合的各种设备。

15 在一些实施例中，可执行指令可以采用程序、软件、软件模块、脚本或代码的形式，按任意形式的编程语言（包括编译或解释语言，或者声明性或过程性语言）来编写，并且其可按任意形式部署，包括被部署为独立的程序或者被部署为模块、组件、子例程或者适合在计算环境中使用的其它单元。

作为示例，可执行指令可以但不一定对应于文件系统上的文件，可以可被存储在保存其它程序或数据的文件的一部分，例如，存储在超文本标记语言（HTML，Hyper Text Markup Language）文档中的一个或多个脚本中，存储在专用于所讨论的程序的单个文件中，或者，存储在多个协同文件（例如，存储一个或多个模块、子程序或代码部分的文件）中。

20 作为示例，可执行指令可被部署为在一个计算设备上执行，或者在位于一个地点的多个计算设备上执行，又或者，在分布在多个地点且通过通信网络互连的多个计算设备上执行。

以上，仅为本申请的实施例而已，并非用于限定本申请的保护范围。凡在本申请的精神和范围之内所作的任何修改、等同替换和改进等，均包含在本申请的保护范围之内。

权利要求书

- 1、一种业务通信方法，由电子设备执行，所述方法包括：
接收业务接入进程发送的鉴权请求；
对所述业务接入进程进行同步验证处理，对所述业务接入进程进行异步验证处理；
5 根据对所述业务接入进程的同步验证处理结果，确定为所述业务接入进程分配的业务密钥信息；
将所述业务密钥信息发送至所述业务接入进程，以基于所述业务密钥信息进行与所述业务接入进程之间的加密业务通信；
根据对所述业务接入进程的异步验证处理结果，控制与所述业务接入进程之间用于承载所述加密业务通信的通信连接。
- 10 2、根据权利要求1所述的方法，其中，所述对所述业务接入进程进行同步验证处理，包括：
将所述业务接入进程的进程路径与设定的安全目录之间的匹配结果，作为第一验证处理结果；
对所述业务接入进程的签名信息进行验证处理，得到第二验证处理结果；
根据所述第一验证处理结果及所述第二验证处理结果，确定对所述业务接入进程的同步验证处理结果。
- 15 3、根据权利要求2所述的方法，其中，所述对所述业务接入进程的签名信息进行验证处理，得到第二验证处理结果，包括：
根据所述签名信息是否包括数字签名的结果、所述数字签名的有效性验证处理结果、所述数字签名的签名方与签名方黑名单之间的匹配结果、以及所述签名信息中的证书信息与证书信息黑名单之间的匹配结果中的至少之一，确定第二验证处理结果。
- 20 4、根据权利要求3所述的方法，其中，所述方法还包括：
确定所述签名信息中的数字签名、以及与所述数字签名对应的解密密钥；
根据所述解密密钥对所述数字签名进行解密处理，得到所述业务接入进程的进程文件的第一哈希结果；
对所述业务接入进程的进程文件进行哈希处理，得到第二哈希结果；
25 将所述第一哈希结果与所述第二哈希结果之间的匹配结果，作为所述数字签名的有效性验证处理结果。
- 5、根据权利要求1所述的方法，其中，所述鉴权请求包括所述业务接入进程生成的非对称密钥对中的公钥；其中，所述非对称密钥对包括所述公钥以及与所述公钥对应的私钥；
所述将所述业务密钥信息发送至所述业务接入进程，包括：
30 根据所述公钥对所述业务密钥信息进行加密处理；
将加密后的所述业务密钥信息发送至所述业务接入进程，以使所述业务接入进程根据所述私钥对加密后的所述业务密钥信息进行解密处理。
- 6、根据权利要求5所述的方法，其中，所述业务密钥信息包括密钥标识及对称密钥；
所述将加密后的所述业务密钥信息发送至所述业务接入进程之后，所述方法还包括：
35 接收所述业务接入进程发送的业务请求；其中，所述业务请求包括所述密钥标识、以及利用所述对称密钥加密后的请求数据；
在已分配的对称密钥中，查询与所述业务请求中的所述密钥标识对应的对称密钥；
根据查询到的对称密钥，对所述业务请求中的加密后的请求数据进行解密处理；
对解密处理得到的请求数据进行响应处理，得到响应数据；
40 根据所述公钥对所述响应数据进行加密处理，并将加密后的所述响应数据发送至所述业务接入进程，以使所述业务接入进程根据所述私钥对加密后的所述响应数据进行解密处理。
- 7、根据权利要求6所述的方法，其中，所述业务请求还包括时间戳及第一校验信息；其中，所述第一校验信息是所述业务接入进程对所述密钥标识、所述对称密钥、所述时间戳、以及利用所述对称密钥加密后的请求数据进行哈希处理得到的；
45 所述对解密处理得到的请求数据进行响应处理，包括：
对所述密钥标识、所述查询到的对称密钥、所述时间戳、以及所述加密后的请求数据进行哈希处理，得到第二校验信息；
当所述第一校验信息与所述第二校验信息之间的匹配结果为匹配成功时，对解密处理得到的请求数据进行响应处理。

8、根据权利要求6所述的方法，其中，所述在已分配的对称密钥中，查询与所述业务请求中的所述密钥标识对应的对称密钥之后，所述方法还包括：

当查询到的对称密钥的使用参数满足过期参数条件时，将过期信息发送至所述业务接入进程，以使所述业务接入进程在接收到所述过期信息时重新发送鉴权请求；

5 其中，所述使用参数包括使用次数以及使用时长中的至少一种。

9、根据权利要求1所述的方法，其中，所述鉴权请求包括鉴权请求地址；

所述确定为所述业务接入进程分配的业务密钥信息，包括：

执行以下任意一种处理：

10 针对不同所述业务接入进程发送的鉴权请求地址分配不同的业务密钥信息，针对同一所述业务接入进程发送的不同鉴权请求地址分配不同的业务密钥信息；

针对不同所述业务接入进程发送的鉴权请求地址分配不同的业务密钥信息，针对同一所述业务接入进程发送的不同鉴权请求地址分配相同的业务密钥信息；

针对不同所述业务接入进程发送的鉴权请求地址分配相同的业务密钥信息，针对同一所述业务接入进程发送的不同鉴权请求地址分配相同的业务密钥信息。

15 10、根据权利要求1所述的方法，其中，所述根据对所述业务接入进程的同步验证处理结果，确定为所述业务接入进程分配的业务密钥信息，包括：

当对所述业务接入进程的同步验证处理结果为验证成功时，确定为所述业务接入进程分配的业务密钥信息；

所述方法还包括：

20 当对所述业务接入进程的同步验证处理结果为验证失败时，执行以下任意一种处理：

通知所述业务接入进程重新发送鉴权请求；

断开与所述业务接入进程之间用于传输所述鉴权请求的通信连接；

中断对所述业务接入进程的异步验证处理。

25 11、根据权利要求1所述的方法，其中，所述根据对所述业务接入进程的异步验证处理结果，控制与所述业务接入进程之间用于承载所述加密业务通信的通信连接，包括：

当对所述业务接入进程的异步验证处理结果为验证成功时，维持与所述业务接入进程之间用于承载所述加密业务通信的通信连接；

当对所述业务接入进程的异步验证处理结果为验证失败时，断开与所述业务接入进程之间用于承载所述加密业务通信的通信连接。

30 12、根据权利要求1至11任一项所述的方法，其中，所述方法还包括：

在所述加密业务通信的过程中，执行以下处理：

接收所述业务接入进程发送的凭证请求；其中，所述凭证请求是所述业务接入进程在拦截到应用进程的业务请求时发送的，所述业务请求的目的地址为业务服务器的地址；

对所述应用进程进行同步验证处理；

35 根据对所述应用进程的同步验证处理结果，确定为所述应用进程分配的业务凭证及网关地址，并将所述业务凭证及所述网关地址发送至所述业务接入进程，以使

所述业务接入进程将所述业务凭证及所述业务请求发送至所述网关地址对应的业务网关；

40 其中，所述业务网关用于对接收到的所述业务凭证进行验证处理，并在对所述业务凭证的验证处理结果为验证成功时，将接收到的所述业务请求发送至所述业务服务器；所述业务服务器用于对接收到的所述业务请求中的请求数据进行响应处理。

13、根据权利要求12所述的方法，其中，所述对所述应用进程进行同步验证处理，包括：

确定所述应用进程所在设备中处于登录态的用户账号，并获取所述用户账号对应的可信应用进程的进程信息、以及所述用户账号对应的可访问的业务服务器的地址；

45 将所述应用进程的进程信息与所述可信应用进程的进程信息之间的匹配结果，作为第三验证处理结果；

将所述业务请求所请求的业务服务器的地址与所述可访问的业务服务器的地址之间的匹配结果，作为第四验证处理结果；

获取所述应用进程所在设备的设备信息，并将所述设备信息与设备安全条件之间的匹配结果，作为第五验证处理结果；

50 根据所述第三验证处理结果、所述第四验证处理结果及所述第五验证处理结果，确定对所述应用进程的同步验证处理结果。

14、根据权利要求 12 所述的方法，其中，所述接收所述业务接入进程发送的凭证请求之后，所述方法还包括：

周期性地所述应用进程的进程信息与进程信息黑名单进行匹配，并将得到的匹配结果作为对所述应用进程的异步验证处理结果；

- 5 根据对所述应用进程的异步验证处理结果，控制与所述业务接入进程之间用于承载所述加密业务通信的通信连接。

15、根据权利要求 1 至 11 任一项所述的方法，其中，所述对所述业务接入进程进行异步验证处理，包括：

- 10 周期性地所述业务接入进程的进程信息与进程信息黑名单进行匹配，并将得到的匹配结果作为对所述业务接入进程的异步验证处理结果。

16、一种业务通信系统，包括业务接入客户端、安全客户端及安全服务器；其中，所述业务接入客户端运行有业务接入进程；

所述安全客户端，用于：

接收所述业务接入进程发送的鉴权请求；

- 15 对所述业务接入进程进行同步验证处理，通知所述安全服务器对所述业务接入进程进行异步验证处理；

根据对所述业务接入进程的同步验证处理结果，确定为所述业务接入进程分配的业务密钥信息；

将所述业务密钥信息发送至所述业务接入进程，以基于所述业务密钥信息进行与所述业务接入进程之间的加密业务通信；

- 20 根据所述安全服务器对所述业务接入进程的异步验证处理结果，控制与所述业务接入进程之间用于承载所述加密业务通信的通信连接。

17、根据权利要求 16 所述的系统，其中，

所述安全客户端，还用于：

- 25 接收所述业务接入进程发送的凭证请求；其中，所述凭证请求是所述业务接入进程在拦截到应用进程的业务请求时发送的，所述业务请求的目的地址为业务服务器的地址；

通知所述安全服务器对所述应用进程进行同步验证处理；

所述安全服务器，用于根据对所述应用进程的同步验证处理结果，确定为所述应用进程分配的业务凭证及网关地址，并将所述业务凭证及所述网关地址通过所述安全客户端发送至所述业务接入进程；

- 30 所述业务接入客户端，还用于将所述业务凭证及所述业务请求通过所述业务接入进程发送至所述网关地址对应的业务网关；

其中，所述业务网关用于对接收到的所述业务凭证进行验证处理，并在对所述业务凭证的验证处理结果为验证成功时，将接收到的所述业务请求发送至所述业务服务器；所述业务服务器用于对接收到的所述业务请求中的请求数据进行响应处理。

- 35 18、一种业务通信装置，所述装置包括：

接收模块，配置为接收业务接入进程发送的鉴权请求；

验证模块，配置为对所述业务接入进程进行同步验证处理，对所述业务接入进程进行异步验证处理；

- 40 确定模块，配置为根据对所述业务接入进程的同步验证处理结果，确定为所述业务接入进程分配的业务密钥信息；

发送模块，配置为将所述业务密钥信息发送至所述业务接入进程，以基于所述业务密钥信息进行与所述业务接入进程之间的加密业务通信；

连接控制模块，配置为根据对所述业务接入进程的异步验证处理结果，控制与所述业务接入进程之间用于承载所述加密业务通信的通信连接。

- 45 19、一种电子设备，所述电子设备包括：

存储器，用于存储可执行指令；

处理器，用于执行所述存储器中存储的可执行指令时，实现权利要求 1 至 15 任一项所述的业务通信方法。

- 50 20、一种计算机可读存储介质，存储有可执行指令，所述可执行指令被处理器执行时实现权利要求 1 至 15 任一项所述的业务通信方法。

21、一种计算机程序产品，包括可执行指令，所述可执行指令被处理器执行时实现权利要求 1 至 15 任一项所述的业务通信方法。