(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 108683712 B
(45)授权公告日 2021.04.27

(21)申请号 201810376903.8

(22)申请日 2018.04.25

(65)同一申请的已公布的文献号
申请公布号 CN 108683712 A

(43)申请公布日 2018.10.19

(73)专利权人 咪咕文化科技有限公司
地址 100032 北京市西城区德胜门外大街11号5幢400室(德胜园区)
专利权人 中国移动通信集团有限公司

(72)发明人 任化强 李琳 周冰 周效军 赵家成

(74)专利代理机构 北京派特恩知识产权代理有限公司 11270
代理人 张荣 张颖玲

(51)Int.Cl.
H04L 29/08(2006.01)
H04L 9/32(2006.01)
H04L 9/08(2006.01)
G06F 21/51(2013.01)

(56)对比文件
CN 104836784 A,2015.08.12
CN 103793633 A,2014.05.14
US 2017346807 A1,2017.11.30

审查员 陈相玫

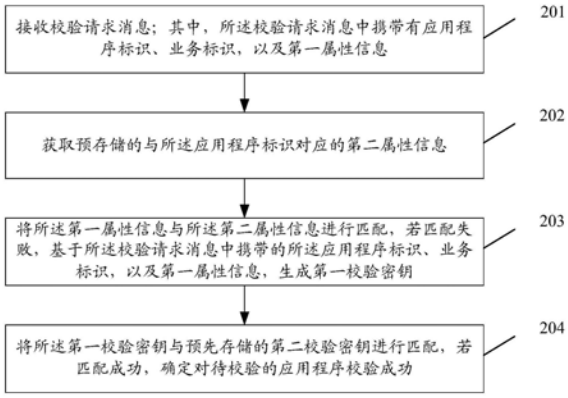
权利要求书2页 说明书16页 附图4页

(54)发明名称

应用程序校验及校验密钥的生成方法、装置及存储介质

(57)摘要

本发明公开了一种应用程序校验方法，包括：接收校验请求消息，其中，所述校验请求消息中携带有应用程序标识、业务标识，以及第一属性信息；获取预存储的与所述应用程序标识对应的第二属性信息；将所述第一属性信息与所述第二属性信息进行匹配，若匹配失败，则基于所述校验请求消息中携带的所述应用程序标识、业务标识，以及第一属性信息，生成第一校验密钥；将所述第一校验密钥与预先存储的第二校验密钥进行匹配，若匹配成功，确定对待校验的应用程序校验成功。本发明还同时公开了一种应用程序校验装置、一种应用程序校验密钥的生成方法和装置，以及存储介质。



CN 108683712 B

1.一种应用程序校验密钥的生成方法,其特征在于,所述方法包括:

当获取的第一请求消息中携带有应用程序标识时,从已存储的应用程序的业务标识中获取与所述应用程序标识对应的业务标识;其中,所述第一请求消息为申请备份的应用程序发送的用于请求生成校验密钥的消息;

判断所述第一请求消息中携带的业务标识,与所获取的业务标识是否匹配;

若匹配成功,根据所述应用程序标识、所述第一请求消息中携带的业务标识,以及所述申请备份的应用程序的属性信息,生成所述校验密钥;

其中,所述属性信息包括所述申请备份的应用程序的包名和包签名,所述校验密钥为用于对所述申请备份的应用程序进行校验的密钥。

2.根据权利要求1所述的应用程序校验密钥的生成方法,其特征在于,在所述从已存储的应用程序的业务标识中获取与所述应用程序标识对应的业务标识之前,所述方法还包括:

获取未携带有应用程序标识的第二请求消息,其中,所述第二请求消息中携带业务标识;

为所述第二请求消息中的业务标识分配对应的应用程序标识;

存储所述第二请求消息中的业务标识,以及所分配的应用程序标识。

3.根据权利要求1所述的应用程序校验密钥的生成方法,其特征在于,所述根据所述应用程序标识、所述第一请求消息中携带的业务标识,以及所述申请备份的应用程序的属性信息,生成所述校验密钥,包括:

将所述应用程序标识、所述第一请求消息中携带的业务标识,以及所述包名和包签名进行组合,生成相应的字符串信息;

对所述字符串信息进行哈希处理,得到与所述字符串信息对应的哈希值,将所述哈希值确定为所述校验密钥。

4.一种应用程序校验方法,其特征在于,所述方法包括:

接收校验请求消息,其中,所述校验请求消息中携带有应用程序标识、业务标识,以及第一属性信息;

获取预存储的与所述应用程序标识对应的第二属性信息;

将所述第一属性信息与所述第二属性信息进行匹配,若匹配失败,则基于所述校验请求消息中携带的所述应用程序标识、业务标识,以及第一属性信息,生成第一校验密钥;

将所述第一校验密钥与预先存储的第二校验密钥进行匹配,若匹配成功,确定对待校验的应用程序校验成功;

其中,所述第二校验密钥是基于预存储的应用程序标识、业务标识,以及第二属性信息生成的;所述第一属性信息包括所述待校验的应用程序的包名和包签名。

5.根据权利要求4所述的应用程序校验方法,其特征在于,所述基于所述校验请求消息中携带的所述应用程序标识、业务标识,以及第一属性信息,生成第一校验密钥,包括:

将所述校验请求消息中携带的所述应用程序标识、所述业务标识,以及所述待校验的应用程序的包名和包签名进行组合,生成相应的字符串信息;

对所述字符串信息进行哈希处理,得到与所述字符串信息对应的第一哈希值,将所述第一哈希值确定为所述第一校验密钥。

6.一种应用程序校验密钥的生成装置，其特征在于，所述装置包括：获取模块、判断模块和生成模块；其中，

所述获取模块，用于当获取的第一请求消息中携带有应用程序标识时，从已存储的应用程序的业务标识中获取与所述应用程序标识对应的业务标识；其中，所述第一请求消息为申请备份的应用程序发送的用于请求生成校验密钥的消息；

所述判断模块，用于判断所述第一请求消息中携带的业务标识，与所获取的业务标识是否匹配；

所述生成模块，用于在所述判断模块判定匹配成功时，根据所述应用程序标识、所述第一请求消息中携带的业务标识，以及所述申请备份的应用程序的属性信息，生成所述校验密钥；

其中，所述属性信息包括所述申请备份的应用程序的包名和包签名，所述校验密钥为用于对所述申请备份的应用程序进行校验的密钥。

7.一种应用程序校验装置，其特征在于，所述装置包括：接收模块、获取模块、匹配模块、生成模块和校验模块；其中，

所述接收模块，用于接收校验请求消息，其中，所述校验请求消息中携带有应用程序标识、业务标识，以及第一属性信息；

所述获取模块，用于获取预存储的与所述应用程序标识对应的第二属性信息；

所述匹配模块，用于将所述第一属性信息与所述第二属性信息进行匹配；

所述生成模块，用于所述匹配模块匹配失败时，基于所述校验请求消息中携带的所述应用程序标识、业务标识，以及第一属性信息，生成第一校验密钥；

所述校验模块，用于将所述第一校验密钥与预先存储的第二校验密钥进行匹配，若匹配成功，确定对待校验的应用程序校验成功；

其中，所述第二校验密钥是基于预存储的应用程序标识、业务标识，以及第二属性信息生成的；所述第一属性信息包括所述待校验的应用程序的包名和包签名。

8.一种计算机可读存储介质，其上存储有可执行程序，其特征在于，所述可执行程序被处理器执行时实现如权利要求1至3任一项所述的应用程序校验密钥的生成方法的步骤，或者如权利要求4或5所述的应用程序校验方法的步骤。

9.一种应用程序校验密钥的生成装置，包括存储器、处理器及存储在存储器上并能够由所述处理器运行的可执行程序，其特征在于，所述处理器运行所述可执行程序时执行如权利要求1至3任一项所述的应用程序校验密钥的生成方法的步骤。

10.一种应用程序校验装置，包括存储器、处理器及存储在存储器上并能够由所述处理器运行的可执行程序，其特征在于，所述处理器运行所述可执行程序时执行如权利要求4或5所述的应用程序校验方法的步骤。

应用程序校验及校验密钥的生成方法、装置及存储介质

技术领域

[0001] 本发明涉及计算机技术领域，尤其涉及一种应用程序校验及校验密钥的生成方法、装置及存储介质。

背景技术

[0002] 随着终端技术的不断发展，越来越多的应用程序(APP，Application)被开发和安装使用。为避免用户下载安装的APP为不安全的非法APP，从而影响终端的使用安全，通常需对下载至终端上的待安装使用的APP进行校验，以识别软件来源及软件开发者的真实身份。

[0003] 目前，可使用应用程序密钥(APP KEY，Application KEY)对APP的包名、签名等信息进行加密和解密。相关技术中，由于APP与APP KEY一一对应，因此，当对不同的APP进行合法性验证时，后台服务器都需要使用不同的APP KEY分别对相应的APP进行校验。可见，针对每一个APP，后台服务器都需要有单独的校验过程。

[0004] 然而，随着业务需求的快速发展，尤其是云游戏的不断兴起，目前已出现在属于同一类业务的APP上集成多个APP的情况，比如，在一个游戏类的APP上集成多个游戏APP，若采用现有技术提供的技术方案，对集成的每个游戏APP都单独执行一次与APP KEY的交互过程，则针对一类业务的多个APP进行批量的校验过程需消耗较长的时间，从而降低校验效率，大大影响用户的使用体验。

发明内容

[0005] 有鉴于此，本发明实施例期望提供一种应用程序校验及校验密钥的生成方法、装置及存储介质，至少用以解决相关技术中难以有效提高对多个APP进行批量校验的效率的问题。

[0006] 为达到上述目的，本发明实施例的技术方案是这样实现的：

[0007] 第一方面，本发明实施例提供一种应用程序校验密钥的生成方法，所述方法包括：

[0008] 当获取的第一请求消息中携带有应用程序标识时，从已存储的应用程序的业务标识中获取与所述应用程序标识对应的业务标识；其中，所述第一请求消息为申请备份的应用程序发送的用于请求生成校验密钥的消息；

[0009] 判断所述第一请求消息中携带的业务标识，与所获取的业务标识是否匹配；

[0010] 若匹配成功，根据所述应用程序标识、所述第一请求消息中携带的业务标识，以及所述申请备份的应用程序的属性信息，生成所述校验密钥；

[0011] 其中，所述属性信息包括所述申请备份的应用程序的包名和包签名，所述校验密钥为用于对所述申请备份的应用程序进行校验的密钥。

[0012] 第二方面，本发明实施例还提供一种应用程序校验方法，所述方法包括：

[0013] 接收校验请求消息，其中，所述校验请求消息中携带有应用程序标识、业务标识，以及第一属性信息；

[0014] 获取预存储的与所述应用程序标识对应的第二属性信息；

[0015] 将所述第一属性信息与所述第二属性信息进行匹配，若匹配失败，则基于所述校验请求消息中携带的所述应用程序标识、业务标识，以及第一属性信息，生成第一校验密钥；

[0016] 将所述第一校验密钥与预先存储的第二校验密钥进行匹配，若匹配成功，确定对待校验的应用程序校验成功；

[0017] 其中，所述第二校验密钥是基于预存储的应用程序标识、业务标识，以及第二属性信息生成的；所述第一属性信息包括所述待校验的应用程序的包名和包签名。

[0018] 第三方面，本发明实施例还提供一种应用程序校验密钥的生成装置，所述装置包括：获取模块、判断模块和生成模块；其中，

[0019] 所述获取模块，用于当获取的第一请求消息中携带有应用程序标识时，从已存储的应用程序的业务标识中获取与所述应用程序标识对应的业务标识；其中，所述第一请求消息为申请备份的应用程序发送的用于请求生成校验密钥的消息；

[0020] 所述判断模块，用于判断所述第一请求消息中携带的业务标识，与所获取的业务标识是否匹配；

[0021] 所述生成模块，用于在所述判断模块判定匹配成功时，根据所述应用程序标识、所述第一请求消息中携带的业务标识，以及所述申请备份的应用程序的属性信息，生成所述校验密钥；

[0022] 其中，所述属性信息包括所述申请备份的应用程序的包名和包签名，所述校验密钥为用于对所述申请备份的应用程序进行校验的密钥。

[0023] 第四方面，本发明实施例还提供一种应用程序校验装置，所述装置包括：接收模块、获取模块、匹配模块、生成模块和校验模块；其中，

[0024] 所述接收模块，用于接收校验请求消息，其中，所述校验请求消息中携带有应用程序标识、业务标识，以及第一属性信息；

[0025] 所述获取模块，用于获取预存储的与所述应用程序标识对应的第二属性信息；

[0026] 所述匹配模块，用于将所述第一属性信息与所述第二属性信息进行匹配；

[0027] 所述生成模块，用于所述匹配模块匹配失败时，基于所述校验请求消息中携带的所述应用程序标识、业务标识，以及第一属性信息，生成第一校验密钥；

[0028] 所述校验模块，用于将所述第一校验密钥与预先存储的第二校验密钥进行匹配，若匹配成功，确定对待校验的应用程序校验成功；

[0029] 其中，所述第二校验密钥是基于预存储的应用程序标识、业务标识，以及第二属性信息生成的；所述第一属性信息包括所述待校验的应用程序的包名和包签名。

[0030] 第五方面，本发明实施例还提供一种存储介质，其上存储有可执行程序，所述可执行程序被处理器执行时实现本发明实施例提供的应用程序校验密钥的生成方法的步骤，或者本发明实施例提供的应用程序校验方法的步骤。

[0031] 第六方面，本发明实施例还提供一种应用程序校验密钥的生成装置，包括存储器、处理器及存储在存储器上并能够由所述处理器运行的可执行程序，所述处理器运行所述可执行程序时执行本发明实施例提供的应用程序校验密钥的生成方法的步骤。

[0032] 第七方面，本发明实施例还提供一种应用程序校验装置，包括存储器、处理器及存储在存储器上并能够由所述处理器运行的可执行程序，所述处理器运行所述可执行程序时

执行本发明实施例提供的应用程序校验方法的步骤。

[0033] 本发明实施例所提供的应用程序校验及校验密钥的生成方法、装置及存储介质，通过接收校验请求消息，所述校验请求消息中携带有应用程序标识、业务标识，以及第一属性信息；获取预存储的与所述应用程序标识对应的第二属性信息；将所述第一属性信息与所述第二属性信息进行匹配，若匹配失败，则基于所述校验请求消息中携带的所述应用程序标识、业务标识，以及第一属性信息，生成第一校验密钥；将所述第一校验密钥与预先存储的第二校验密钥进行匹配，若匹配成功，确定对待校验的应用程序校验成功。如此，可以允许同一个应用程序标识下增加至少两个包签名，通过根据校验请求消息中的应用程序标识、业务标识，以及第一属性信息所生成的第一校验密钥，与第二校验密钥进行匹配来确定校验是否成功，这样，针对同一应用程序标识的多个应用程序来说，仅需执行一次与应用程序交互加解密密钥的过程，可以有效缩短对多个应用程序进行校验的时间，以及提高校验效率，从而大大提升用户的使用体验。

附图说明

[0034] 图1为本发明实施例提供的一种应用程序校验密钥的生成方法的实现流程示意图；

[0035] 图2为本发明实施例提供的一种应用程序校验方法的实现流程示意图；

[0036] 图3为本发明实施例提供的一种基于应用程序校验的系统架构示意图；

[0037] 图4为本发明实施例提供的一种应用程序校验方法的具体实现交互流程示意图；

[0038] 图5为本发明实施例提供的一种应用程序校验密钥的生成装置的功能结构示意图；

[0039] 图6为本发明实施例提供的一种应用程序校验装置的功能结构示意图；

[0040] 图7为本发明实施例提供的一种应用程序校验密钥的生成装置的硬件结构示意图；

[0041] 图8为本发明实施例提供的一种应用程序校验装置的硬件结构示意图。

具体实施方式

[0042] 为了能够更加详尽地了解本发明实施例的特点与技术内容，下面结合附图对本发明实施例的实现进行详细阐述，所附附图仅供参考说明之用，并非用来限定本发明。

[0043] 图1为本发明实施例提供的一种应用程序校验密钥的生成方法的实现流程示意图，该应用程序校验密钥的生成方法可应用于服务器中；如图1所示，本发明实施例中的应用程序校验密钥的生成方法的实现流程，可以包括如下步骤：

[0044] 步骤101：当获取的第一请求消息中携带有应用程序标识时，从已存储的应用程序的业务标识中获取与所述应用程序标识对应的业务标识。

[0045] 在本实施例中，所述第一请求消息为申请备份的应用程序发送的用于请求生成校验密钥的消息。

[0046] 这里，对于多个应用程序申请集成私有软件开发工具包(SDK，Software Development Kit)来说，比如在一个游戏类的应用程序上集成多个子游戏应用程序，可以将每个子游戏的应用程序集成于同一个应用程序标识下，即每个子游戏的应用程序对应的

应用程序标识相同，由于应用程序标识与加解密密钥具有一一对应的关系，因此，可以使用同一个加解密密钥来实现对每个子游戏的应用程序进行加密或解密。这样，针对多个子游戏应用程序而言，仅需执行一次应用程序与加解密密钥交互的过程，减少交互过程中不必要的时间浪费。

[0047] 需要说明的是，在一个游戏类的应用程序上集成的多个子游戏应用程序所对应的业务标识也是相同的。

[0048] 在本实施例中，在本步骤101中执行从已存储的应用程序的业务标识中获取与所述应用程序标识对应的业务标识之前，该应用程序校验密钥的生成方法还可以包括以下步骤：

[0049] 获取未携带有应用程序标识的第二请求消息，其中，所述第二请求消息中携带业务标识；

[0050] 为所述第二请求消息中的业务标识分配对应的应用程序标识；

[0051] 存储所述第二请求消息中的业务标识，以及所分配的应用程序标识。

[0052] 在本实施例中，具体来说，当所获取的第二请求消息中未携带有应用程序标识时，也就是说，在申请备份阶段中，并未使用已有的应用程序标识进行备份，而是申请新的应用程序标识，即使用新申请的应用程序标识进行备份，因此，后台服务器将为所述申请备份的应用程序分配唯一的应用程序标识，以及分配与所分配的应用程序标识具有对应关系的加解密密钥；然后，将所分配的应用程序标识、加解密密钥，以及与所分配的应用程序标识具有对应关系的包名、包签名、业务标识一并存储至后台数据库中，并作为应用程序校验的依据之一。这里，需要强调的是，本发明实施例中所涉及的申请备份阶段为后续的校验阶段做了充足的准备，所述申请备份阶段即为预先生成和存储用于应用程序校验的校验密钥的阶段，在该阶段中，还可以预先存储后续校验过程中使用的应用程序标识，以及与应用程序标识对应的业务标识和应用程序的属性信息。

[0053] 步骤102：判断所述第一请求消息中携带的业务标识，与所获取的业务标识是否匹配。

[0054] 这里，在本步骤102中判断第一请求消息中携带的业务标识，与所获取的业务标识是否匹配之前，还可以从第一请求消息中解析出所携带的业务标识，然后再将解析出的业务标识与所获取的业务标识进行匹配。需要特别说明的是，这里对业务标识进行匹配验证的目的是为了判断两个关联的应用程序是否同属于一类业务。

[0055] 步骤103：若匹配成功，根据所述应用程序标识、所述第一请求消息中携带的业务标识，以及所述申请备份的应用程序的属性信息，生成所述校验密钥。

[0056] 在本实施例中，所述属性信息可以包括所述申请备份的应用程序的包名和包签名，其中，所述校验密钥为用于对所述申请备份的应用程序进行校验的密钥。

[0057] 在本实施例中，对于本步骤103中的根据所述应用程序标识、所述第一请求消息中携带的业务标识，以及所述申请备份的应用程序的属性信息，生成所述校验密钥来说，具体可以采用如下方式来实现：

[0058] 将所述应用程序标识、所述第一请求消息中携带的业务标识，以及所述包名和包签名进行组合，生成相应的字符串信息；

[0059] 对所述字符串信息进行哈希处理，得到与所述字符串信息对应的哈希值，将所述

哈希值确定为所述校验密钥。

[0060] 这里，在执行完本步骤103之后，该应用程序校验密钥的生成方法还可以包括：

[0061] 将所述校验密钥，以及从所述请求消息中解析出的所述应用程序标识存储至本地数据库中。

[0062] 需要说明的是，可以将生成的校验密钥存储在本地注册的哈希表中，或存储在一个数据库表中，以便为后续的对待校验的应用程序进行校验作依据。

[0063] 图2为本发明实施例提供的一种应用程序校验方法的实现流程示意图，该应用程序校验方法可应用于服务器中；如图2所示，本发明实施例中的应用程序校验方法的实现流程，可以包括以下步骤：

[0064] 步骤201：接收校验请求消息；其中，所述校验请求消息中携带有应用程序标识、业务标识，以及第一属性信息。

[0065] 在本实施例中，所述第一属性信息可以包括所述待校验的应用程序的包名和包签名。

[0066] 这里，在后台服务器接收校验请求消息之前，接入服务器如应用的SDK先从本地缓存中查看是否有缓存的加密的包名和包签名信息，如果本地缓存中有缓存的加密的包名和包签名信息，则直接使用加解密密钥进行解密，并将获取到的包名和包签名与解密得到的包名和包签名进行匹配，若匹配成功，则可以确定校验通过；否则校验未通过。当接入服务器从本地缓存中未获取到加密的应用程序的包名和包签名时，则再向后台服务器发送校验请求。

[0067] 步骤202：获取预存储的与所述应用程序标识对应的第二属性信息。

[0068] 在本实施例中，预先在后台数据库中存储与应用程序标识具有对应关系的第二属性信息，其中，所述第二属性信息可以包括预存储的应用程序的包名和包签名，这样，根据对应关系，即可确定预存储的与所述应用程序标识对应的应用程序的包名和包签名。

[0069] 步骤203：将所述第一属性信息与所述第二属性信息进行匹配，若匹配失败，基于所述校验请求消息中携带的所述应用程序标识、业务标识，以及第一属性信息，生成第一校验密钥。

[0070] 在本实施例中，对于本步骤203中的基于所述校验请求消息中携带的所述应用程序标识、业务标识，以及第一属性信息，生成第一校验密钥来说，可以采用如下方式来实现：

[0071] 将所述校验请求消息中携带的所述应用程序标识、所述业务标识，以及所述待校验的应用程序的包名和包签名进行组合，生成相应的字符串信息；

[0072] 对所述字符串信息进行哈希处理，得到与所述字符串信息对应的第一哈希值，将所述第一哈希值确定为所述第一校验密钥。

[0073] 这里，对于将校验请求消息中携带的应用程序标识、业务标识，以及待校验的应用程序的包名和包签名进行组合的组合方式可以有多种形式，这里不做具体限定。需要说明的是，在本发明实施例中，可以采用哈希算法如Hash算法，对所述字符串信息进行哈希处理，得到与所述字符串信息对应的第一哈希值，即将任意长度的字符串信息的二进制值映射为较短的固定长度的字符串信息的二进制值，这里不再赘述。

[0074] 在本实施例中，所述校验请求消息中携带的包名和包签名为加密的包名和包签名；

[0075] 对于本步骤203中的将所述第一属性信息与所述第二属性信息进行匹配来说，可以具体包括以下步骤：

[0076] 从所述校验请求消息中解析出所述加密的包名和包签名；

[0077] 基于加解密密钥，对所述加密的包名和包签名进行解密，获得解密后的包名和包签名；

[0078] 根据所述应用程序标识，从后台数据库中查询预存储的与所述应用程序标识对应的包名和包签名，将所述解密后的包名和包签名与所查询到的包名和包签名进行匹配。

[0079] 在本实施例中，所述校验请求消息中携带的业务标识为加密的业务标识；

[0080] 对于本步骤203中的在基于所述校验请求消息中携带的所述应用程序标识、业务标识，以及第一属性信息，生成第一校验密钥之前，该应用程序校验方法还可以包括以下步骤：

[0081] 从所述校验请求消息中解析出所述加密的业务标识；

[0082] 基于加解密密钥，对所述加密的业务标识进行解密，获得解密后的业务标识；

[0083] 根据所述应用程序标识，从所述后台数据库中查询预存储的与所述应用程序标识对应的业务标识，将所述解密后的业务标识与所查询到的业务标识进行匹配；

[0084] 若匹配成功，则基于所述校验请求消息中携带的所述应用程序标识、业务标识，以及第一属性信息，生成第一校验密钥。

[0085] 步骤204：将所述第一校验密钥与预先存储的第二校验密钥进行匹配，若匹配成功，确定对待校验的应用程序校验成功。

[0086] 在本实施例中，所述第二校验密钥是基于预存储的应用程序标识、业务标识，以及第二属性信息生成的。

[0087] 这里，第二属性信息包括预存储的应用程序的包名和包签名。

[0088] 具体来说，对于预先存储的第二校验密钥的生成过程，可以采用如下方式来实现：

[0089] 将所述预存储的应用程序标识、业务标识，以及应用程序的包名和包签名进行组合，生成相应的字符串信息；

[0090] 对所述字符串信息进行哈希处理，得到与所述字符串信息对应的第二哈希值，将所述第二哈希值确定为所述第二校验密钥。

[0091] 这里，对于将所述预存储的应用程序标识、业务标识，以及应用程序的包名和包签名进行组合的组合方式同样可以有多种形式，这里不做具体限定。需要说明的是，在本发明实施例中，可以采用哈希算法如Hash算法对字符串信息进行哈希处理，得到与字符串信息对应的第二哈希值，即将任意长度的字符串信息的二进制值映射为较短的固定长度的字符串信息的二进制值，这里不再赘述。

[0092] 这里，可以将所生成的第二校验密钥存储在本地注册的哈希表中，或存储在一个数据库表中，以便为后续的应用程序校验作依据。

[0093] 在本实施例中，对于本步骤204中的将所述第一校验密钥与预先存储的第二校验密钥进行匹配来说，具体可以包括如下步骤：

[0094] 基于所述第一哈希值，向本地注册的哈希表中查询是否存在与所述第一哈希值相匹配的第二哈希值；

[0095] 当确定所述哈希表中存在与所述第一哈希值相匹配的第二哈希值时，则确定对所

述待校验的应用程序校验成功；

[0096] 其中，所述第二哈希值为预存储的与所述第二校验密钥对应的哈希值。

[0097] 这里，在后台服务器对所述待校验的应用程序校验成功之后，将校验结果返回给接入服务器，当接入服务器接收到校验成功的反馈消息后，接入服务器可以使用加解密密钥将获取到的第二属性信息进行加密，并将加密的第二属性信息缓存至本地，供后续对应用程序校验使用。

[0098] 采用本发明实施例的技术方案，根据校验请求消息中的应用程序标识、业务标识，以及第一属性信息所生成的第一校验密钥，与第二校验密钥进行匹配来确定校验是否成功，这样，针对同一应用程序标识的多个应用程序来说，仅需执行一次与应用程序交互加解密密钥的过程，在保证SDK被安全调用的情况下，可以有效缩短对多个应用程序进行校验的时间，以及提高校验效率，从而大大提升用户的使用体验。

[0099] 下面对本发明实施例提出的应用程序校验方法的具体实现过程做进一步地详细说明。

[0100] 图3为本发明实施例提供的一种基于应用程序校验的系统架构示意图，该系统架构涉及四个对象的交互，来完成备份与校验的过程，如图3所示，该系统架构主要包括客户端(应用程序APP)、接入服务器(应用的SDK)、管理服务器和后台服务器(SDK后台)四个功能模块；其中，各模块提供的功能如下：

[0101] 客户端：主要为集成私有的SDK的多个应用程序；

[0102] 接入服务器：可以为具有某种功能的私有SDK，可以调用SDK，供多个应用程序申请集成使用；

[0103] 管理服务器：应用程序的管理终端，可以通过后台服务器的申请接入服务，对应用程序的包名和包签名等信息进行备份；

[0104] 后台服务器：主要用于提供给接入服务器相关服务，比如对应用程序进行备份与校验的服务。

[0105] 基于上述图3所示的基于应用程序校验的系统架构，下面对本发明实施例的应用程序校验方法的具体实现交互过程进行说明。图4为本发明实施例提供的一种应用程序校验方法的具体实现交互流程示意图，如图4所示，在对所述应用程序进行校验之前，还包括应用程序申请备份的过程，下面分别对应用程序申请备份的过程和对应用程序进行校验的过程进行说明。

[0106] 其中，应用程序申请备份的实现流程，可以包括以下步骤：

[0107] 步骤401：管理服务器向后台服务器发送申请备份的请求消息。

[0108] 步骤402：后台服务器根据请求消息中是否携带应用程序标识的情况，备份应用程序的属性信息。

[0109] 这里，管理服务器在向后台服务器申请备份的过程中，可以先判断是否需要使用已有的应用程序标识进行备份，也即判断发送的请求消息中是否携带有应用程序标识，若所发送的请求消息中携带有应用程序标识，则确定需要使用已有的应用程序标识进行备份，此时，从已存储的应用程序的业务标识中获取与所述应用程序标识对应的业务标识，并将获取的业务标识和已有的应用程序标识一并传输至后台服务器。

[0110] 在本实施例中，若确定所发送的请求消息中未携带有应用程序标识，即使用新申

请的应用程序标识进行备份，则需要为获取的请求消息中携带的业务标识分配对应的应用程序标识，将请求消息中的业务标识，以及所分配的应用程序标识和应用程序的属性信息一并进行存储。

[0111] 在本实施例中，若确定所发送的请求消息中携带有应用程序标识，即使用已有的应用程序标识进行备份，此时，根据应用程序标识，从数据库中查询从已存储的应用程序的业务标识中获取与应用程序标识对应的业务标识；将请求消息中携带的业务标识与所获取的业务标识进行匹配，若匹配成功，则根据所述应用程序标识、所述请求消息中携带的业务标识，以及所述申请备份的应用程序的属性信息，生成第二校验密钥。将所生成的第二校验密钥存储在本地注册的哈希表中，或存储在一个数据库表中，以便为后续的应用程序校验作依据。

[0112] 这里，对业务标识进行匹配验证的目的是为了判断两个关联的应用程序是否同属于一类业务。

[0113] 这里，所述属性信息可以包括所述申请备份的应用程序的包名和包签名，其中，所述第二校验密钥为用于对所述申请备份的应用程序进行校验的密钥。

[0114] 这里，根据所述应用程序标识、所述请求消息中携带的业务标识，以及所述申请备份的应用程序的属性信息，生成第二校验密钥，可以包括：将所述应用程序标识、所述请求消息中携带的业务标识，以及所述包名和包签名进行组合，生成相应的字符串信息；

[0115] 对所述字符串信息进行哈希处理，得到与所述字符串信息对应的第二哈希值，将所述哈希值确定为第二校验密钥。

[0116] 这里，可以将所生成的第二校验密钥存储在本地注册的哈希表中，或存储在一个数据库表中，以便为后续的应用程序校验作依据。

[0117] 步骤403：后台服务器将备份的应用程序的属性信息、应用程序标识和业务标识返回给管理服务器进行存储。

[0118] 其中，对应用程序进行校验的实现流程，可以包括以下步骤：

[0119] 步骤404：应用程序申请备份成功后，客户端可以从管理服务器获取对应的应用程序标识等信息，向接入服务器发送启动调用SDK服务的请求。

[0120] 在本实施例中，客户端每次调用SDK服务时，都会在请求中携带有应用程序标识、加解密密钥与业务标识，供校验使用。

[0121] 步骤405：接入服务器接收到请求后，会获取应用程序的属性信息，并向后台服务器发送校验请求消息，其中，所述校验请求消息中携带有应用程序标识、业务标识，以及应用程序的属性信息。

[0122] 这里，应用程序的属性信息可以包括应用程序的包名和包签名。下面给出安卓系统中获取包名和包签名的实现代码：

[0123] 获取包签名的实现代码如下所示：PackageInfo packageInfo＝context.getPackageManager().getPackageInfo(appPackage,packageManager.GET_SIGNATURES)；

[0124] 获取包名的实现代码如下所示：Context.getPackageName()。

[0125] 这里，接入服务器获取到应用程序的属性信息之后，先从本地缓存中查看是否有缓存的加密的包名和包签名信息，如果本地缓存中有缓存的加密的包名和包签名信息，则

直接使用加解密密钥进行解密，并将获取到的包名和包签名与解密得到的包名和包签名进行匹配，若匹配成功，则可以确定校验通过；否则校验未通过。当接入服务器从本地缓存中未获取到加密的应用程序的包名和包签名时，则再向后台服务器发送校验请求。

[0126] 步骤406：后台服务器接收到校验请求消息后，根据应用程序标识查询数据库，对待校验的应用程序进行校验处理。

[0127] 在本实施例中，后台服务器根据应用程序标识从数据库中查询预存储的与应用程序标识对应的属性信息。将校验请求消息中携带的属性信息与查询到的属性信息进行匹配，若匹配失败，则基于校验请求消息中携带的应用程序标识、业务标识，以及属性信息，生成第一校验密钥。

[0128] 在本实施例中，对于生成第一校验密钥的具体实现过程，可以包括以下步骤：将所述校验请求消息中携带的所述应用程序标识、所述业务标识，以及所述待校验的应用程序的包名和包签名进行组合，生成相应的字符串信息；对字符串信息进行哈希处理，得到与字符串信息对应的第一哈希值，将第一哈希值确定为第一校验密钥。将所述第一校验密钥与预先存储的第二校验密钥进行匹配，若匹配成功，确定对待校验的应用程序校验成功。

[0129] 这里，对于将校验请求消息中携带的所述应用程序标识、所述业务标识，以及所述待校验的应用程序的包名和包签名进行组合的组合方式可以有多种形式，这里不做具体限定。需要说明的是，在本发明实施例中，可以采用哈希算法如Hash算法，对所述字符串信息进行哈希处理，这里不再赘述。

[0130] 具体来说，对于如何将第一校验密钥与预先存储的第二校验密钥进行匹配，可以采用如下方式来实现：基于第一哈希值，向本地注册的哈希表中查询是否存在与第一哈希值相匹配的第二哈希值；当确定哈希表中存在与第一哈希值相匹配的第二哈希值时，则确定对待校验的应用程序校验成功；其中，第二哈希值为预存储的与第二校验密钥对应的哈希值。

[0131] 步骤407：将校验结果反馈给接入服务器。

[0132] 这里，在后台服务器对所述待校验的应用程序校验成功之后，将校验结果返回给接入服务器，当接入服务器接收到校验成功的反馈消息后，接入服务器可以使用加解密密钥将获取到的第二属性信息进行加密，并将加密的第二属性信息缓存至本地，供后续对应用程序校验使用。

[0133] 采用本发明实施例的技术方案，根据校验请求消息中的应用程序标识、业务标识，以及第一属性信息所生成的第一校验密钥，与第二校验密钥进行匹配来确定校验是否成功，这样，针对同一应用程序标识的多个应用程序来说，仅需执行一次与应用程序交互加解密密钥的过程，在保证SDK被安全调用的情况下，可以有效缩短对多个应用程序进行校验的时间，以及提高校验效率，从而大大提升用户的使用体验。

[0134] 为了实现上述应用程序校验密钥的生成方法，本发明实施例还提供了一种应用程序校验密钥的生成装置，该应用程序校验密钥的生成装置可应用于服务器中，图5为本发明实施例提供的一种应用程序校验密钥的生成装置的功能结构示意图；如图5所示，该应用程序校验密钥的生成装置可以包括：获取模块51、判断模块52和生成模块53。下面对各程序模块进行详细说明。其中，

[0135] 所述获取模块51，用于当获取的第一请求消息中携带有应用程序标识时，从已存

储的应用程序的业务标识中获取与所述应用程序标识对应的业务标识;其中,所述第一请求消息为申请备份的应用程序发送的用于请求生成校验密钥的消息;

[0136] 所述判断模块52,用于判断所述第一请求消息中携带的业务标识,与所获取的业务标识是否匹配;

[0137] 所述生成模块53,用于在所述判断模块52判定匹配成功时,根据所述应用程序标识、所述第一请求消息中携带的业务标识,以及所述申请备份的应用程序的属性信息,生成所述校验密钥;

[0138] 其中,所述属性信息包括所述申请备份的应用程序的包名和包签名,所述校验密钥为用于对所述申请备份的应用程序进行校验的密钥。

[0139] 在本实施例中,在所述获取模块51从已存储的应用程序的业务标识中获取与所述应用程序标识对应的业务标识之前,所述获取模块51还用于获取未携带有应用程序标识的第二请求消息,其中,所述第二请求消息中携带业务标识。

[0140] 该应用程序校验密钥的生成装置还可以包括:分配模块,用于为所述第二请求消息中的业务标识分配对应的应用程序标识;

[0141] 存储模块,用于存储所述第二请求消息中的业务标识,以及所分配的应用程序标识。

[0142] 在本实施例中,对于所述生成模块53根据所述应用程序标识、所述第一请求消息中携带的业务标识,以及所述申请备份的应用程序的属性信息,生成所述校验密钥来说,可以采用如下方式实现:

[0143] 将所述应用程序标识、所述第一请求消息中携带的业务标识,以及所述包名和包签名进行组合,生成相应的字符串信息;

[0144] 对所述字符串信息进行哈希处理,得到与所述字符串信息对应的哈希值,将所述哈希值确定为所述校验密钥。

[0145] 在本实施例中,所述分配模块,具体用于当所获取的第二请求消息中未携带有应用程序标识时,为所述申请备份的应用程序分配唯一的应用程序标识,以及与所分配的应用程序标识对应的加解密密钥;

[0146] 所述存储模块,还用于将所述所分配的应用程序标识、加解密密钥,以及与所分配的应用程序标识具有对应关系的包名、包签名、业务标识一并存储至后台数据库中,并作为应用程序校验的依据之一。

[0147] 在本实施例中,所述存储模块,还可用于在所述生成模块53生成所述校验密钥之后,将所述校验密钥,以及从所述请求消息中解析出的所述应用程序标识存储至本地数据库中。

[0148] 需要说明的是:上述实施例提供的应用程序校验密钥的生成装置在进行校验密钥的生成过程时,仅以上述各程序模块的划分进行举例说明,实际应用中,可以根据需要而将上述处理分配由不同的程序模块完成,即将应用程序校验密钥的生成装置的内部结构划分成不同的程序模块,以完成以上描述的全部或者部分处理。另外,上述实施例提供的应用程序校验密钥的生成装置与应用程序校验密钥的生成方法实施例属于同一构思,其具体实现过程详见方法实施例,这里不再赘述。

[0149] 在实际应用中,应用程序校验密钥的生成装置中的上述各程序模块均可由服务器

上的中央处理器(CPU,Central Processing Unit)、微处理器(MPU,Micro Processor Unit)、数字信号处理器(DSP,Digital Signal Processor)、或现场可编程门阵列(FPGA,Field Programmable Gate Array)等实现。

[0150] 为了实现上述应用程序校验方法,本发明实施例还提供了一种应用程序校验装置,该应用程序校验装置可应用于服务器中,图6为本发明实施例提供的一种应用程序校验装置的功能结构示意图;如图6所示,该应用程序校验装置可以包括:接收模块61、获取模块62、匹配模块63、生成模块64和校验模块65。下面对各程序模块进行详细说明。其中,

[0151] 所述接收模块61,用于接收校验请求消息;其中,所述校验请求消息中携带有应用程序标识、业务标识,以及第一属性信息;

[0152] 所述获取模块62,用于获取预存储的与所述应用程序标识对应的第二属性信息;

[0153] 所述匹配模块63,用于将所述第一属性信息与所述第二属性信息进行匹配;

[0154] 所述生成模块64,用于在所述匹配模块63匹配失败时,基于所述校验请求消息中携带的所述应用程序标识、业务标识,以及第一属性信息,生成第一校验密钥;

[0155] 所述校验模块65,用于将所述第一校验密钥与预先存储的第二校验密钥进行匹配,若匹配成功,确定对待校验的应用程序校验成功;

[0156] 其中,所述第二校验密钥是基于预存储的应用程序标识、业务标识,以及第二属性信息生成的;所述第一属性信息包括所述待校验的应用程序的包名和包签名。

[0157] 在本实施例中,对于所述生成模块64基于所述校验请求消息中携带的所述应用程序标识、业务标识,以及第一属性信息,生成第一校验密钥来说,可以采用如下方式实现:

[0158] 将所述校验请求消息中携带的所述应用程序标识、所述业务标识,以及所述待校验的应用程序的包名和包签名进行组合,生成相应的字符串信息;

[0159] 对所述字符串信息进行哈希处理,得到与所述字符串信息对应的第一哈希值,将所述第一哈希值确定为所述第一校验密钥。

[0160] 需要说明的是,所述第二属性信息包括预存储的应用程序的包名和包签名。

[0161] 在本实施例中,所述第二校验密钥是基于预存储的应用程序标识、业务标识,以及第二属性信息生成的,可以采用如下方式实现:

[0162] 将所述预存储的应用程序标识、业务标识,以及应用程序的包名和包签名进行组合,生成相应的字符串信息;

[0163] 对所述字符串信息进行哈希处理,得到与所述字符串信息对应的第二哈希值,将所述第二哈希值确定为所述第二校验密钥。

[0164] 在本实施例中,所述校验请求消息中携带的包名和包签名为加密的包名和包签名;

[0165] 对于所述匹配模块63将所述第一属性信息与所述第二属性信息进行匹配来说,可以采用如下方式来实现:

[0166] 从所述校验请求消息中解析出所述加密的包名和包签名;

[0167] 基于加解密密钥,对所述加密的包名和包签名进行解密,获得解密后的包名和包签名;

[0168] 根据所述应用程序标识,从后台数据库中查询预存储的与所述应用程序标识对应的包名和包签名,将所述解密后的包名和包签名与所查询到的包名和包签名进行匹配。

[0169] 在本实施例中，所述校验请求消息中携带的业务标识为加密的业务标识；

[0170] 在所述生成模块64基于所述校验请求消息中携带的所述应用程序标识、业务标识，以及第一属性信息，生成第一校验密钥之前，所述应用程序校验装置还可以包括：

[0171] 解析模块，用于从所述校验请求消息中解析出所述加密的业务标识；

[0172] 所述获取模块62，还用于基于加解密密钥，对所述加密的业务标识进行解密，获得解密后的业务标识；

[0173] 查询模块，用于根据所述应用程序标识，从所述后台数据库中查询预存储的与所述应用程序标识对应的业务标识；

[0174] 所述匹配模块63，还用于将所述解密后的业务标识与所查询到的业务标识进行匹配；若匹配成功，基于所述校验请求消息中携带的所述应用程序标识、业务标识，以及第一属性信息，生成第一校验密钥；

[0175] 对于所述匹配模块63将所述第一校验密钥与预先存储的第二校验密钥进行匹配来说，可以采用如下方式来实现：

[0176] 基于所述第一哈希值，向本地注册的哈希表中查询是否存在与所述第一哈希值相匹配的第二哈希值；

[0177] 当确定所述哈希表中存在与所述第一哈希值相匹配的第二哈希值时，则确定对所述待校验的应用程序校验成功；

[0178] 其中，所述第二哈希值为预存储的与所述第二校验密钥对应的哈希值。

[0179] 需要说明的是：上述实施例提供的应用程序校验装置在对应用程序进行校验时，仅以上述各程序模块的划分进行举例说明，实际应用中，可以根据需要而将上述处理分配由不同的程序模块完成，即将应用程序校验装置的内部结构划分成不同的程序模块，以完成以上描述的全部或者部分处理。另外，上述实施例提供的应用程序校验装置与应用程序校验方法实施例属于同一构思，其具体实现过程详见方法实施例，这里不再赘述。

[0180] 在实际应用中，应用程序校验装置中的上述各程序模块均可由服务器上的CPU、MPU、DSP或FPGA等实现。

[0181] 为了实现上述应用程序校验密钥的生成方法，本发明实施例还提供了一种应用程序校验密钥的生成装置的硬件结构。现在将参考附图描述实现本发明实施例的应用程序校验密钥的生成装置，该应用程序校验密钥的生成装置可以以各种形式的服务器来实施。下面对本发明实施例的应用程序校验密钥的生成装置的硬件结构做进一步说明，可以理解，图7仅仅示出了应用程序校验密钥的生成装置的示例性结构而非全部结构，根据需要可以实施图7示出的部分结构或全部结构。

[0182] 参见图7，图7为本发明实施例提供的一种应用程序校验密钥的生成装置的硬件结构示意图，实际应用中可以应用于前述运行应用程序的各种形式的服务器，图7所示的应用程序校验密钥的生成装置700包括：至少一个处理器701、存储器702、用户接口703和至少一个网络接口704。所述应用程序校验密钥的生成装置700中的各个组件通过总线系统705耦合在一起。可以理解，总线系统705用于实现这些组件之间的连接通信。总线系统705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7中将各种总线都标为总线系统705。

[0183] 其中，用户接口703可以包括显示器、键盘、鼠标、轨迹球、点击轮、按键、按钮、触感

板或者触摸屏等。

[0184] 可以理解，存储器702可以是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。

[0185] 本发明实施例中的存储器702用于存储各种类型的数据以支持应用程序校验密钥的生成装置700的操作。这些数据的示例包括：用于在应用程序校验密钥的生成装置700上操作的任何计算机程序，如可执行程序7021和操作系统7022，实现本发明实施例的应用程序校验密钥的生成方法的程序可以包含在可执行程序7021中。

[0186] 本发明实施例揭示的应用程序校验密钥的生成方法可以应用于处理器701中，或者由处理器701实现。处理器701可能是一种集成电路芯片，具有信号的处理能力。在实现过程中，上述应用程序校验密钥的生成方法的各步骤可以通过处理器701中的硬件的集成逻辑电路或者软件形式的指令完成。上述的处理器701可以是通用处理器、DSP，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。处理器701可以实现或者执行本发明实施例中提供的各应用程序校验密钥的生成方法、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。结合本发明实施例所提供的应用程序校验密钥的生成方法的步骤，可以直接体现为硬件译码处理器执行完成，或者用译码处理器中的硬件及软件模块组合执行完成。软件模块可以位于存储介质中，该存储介质位于存储器702，处理器701读取存储器702中的信息，结合其硬件完成本发明实施例提供的应用程序校验密钥的生成方法的步骤。

[0187] 本实施例中，所述应用程序校验密钥的生成装置700包括存储器702、处理器701及存储在存储器702上并能够由所述处理器701运行的可执行程序7021，所述处理器701运行所述可执行程序7021时实现：

[0188] 当获取的第一请求消息中携带有应用程序标识时，从已存储的应用程序的业务标识中获取与所述应用程序标识对应的业务标识；其中，所述第一请求消息为申请备份的应用程序发送的用于请求生成校验密钥的消息；判断所述第一请求消息中携带的业务标识，与所获取的业务标识是否匹配；若匹配成功，根据所述应用程序标识、所述第一请求消息中携带的业务标识，以及所述申请备份的应用程序的属性信息，生成所述校验密钥；其中，所述属性信息包括所述申请备份的应用程序的包名和包签名，所述校验密钥为用于对所述申请备份的应用程序进行校验的密钥。

[0189] 作为一种实施方式，所述处理器701运行所述可执行程序7021时实现：

[0190] 在所述从已存储的应用程序的业务标识中获取与所述应用程序标识对应的业务标识之前，获取未携带有应用程序标识的第二请求消息，其中，所述第二请求消息中携带业务标识；为所述第二请求消息中的业务标识分配对应的应用程序标识；存储所述第二请求消息中的业务标识，以及所分配的应用程序标识。

[0191] 作为一种实施方式，所述处理器701运行所述可执行程序7021时实现：

[0192] 将所述应用程序标识、所述第一请求消息中携带的业务标识，以及所述包名和包签名进行组合，生成相应的字符串信息；对所述字符串信息进行哈希处理，得到与所述字符串信息对应的哈希值，将所述哈希值确定为所述校验密钥。

[0193] 在示例性实施例中，本发明实施例还提供了一种存储介质，所述存储介质可为光盘、闪存或磁盘等存储介质，可选为非瞬间存储介质。

[0194] 其中，所述存储介质上存储有可执行程序7021，所述可执行程序7021被处理器701执行时实现：

[0195] 当获取的第一请求消息中携带有应用程序标识时，从已存储的应用程序的业务标识中获取与所述应用程序标识对应的业务标识；其中，所述第一请求消息为申请备份的应用程序发送的用于请求生成校验密钥的消息；判断所述第一请求消息中携带的业务标识，与所获取的业务标识是否匹配；若匹配成功，根据所述应用程序标识、所述第一请求消息中携带的业务标识，以及所述申请备份的应用程序的属性信息，生成所述校验密钥；其中，所述属性信息包括所述申请备份的应用程序的包名和包签名，所述校验密钥为用于对所述申请备份的应用程序进行校验的密钥。

[0196] 作为一种实施方式，所述可执行程序7021被处理器701执行时实现：

[0197] 在所述从已存储的应用程序的业务标识中获取与所述应用程序标识对应的业务标识之前，获取未携带有应用程序标识的第二请求消息，其中，所述第二请求消息中携带业务标识；为所述第二请求消息中的业务标识分配对应的应用程序标识；存储所述第二请求消息中的业务标识，以及所分配的应用程序标识。

[0198] 作为一种实施方式，所述可执行程序7021被处理器701执行时实现：

[0199] 将所述应用程序标识、所述第一请求消息中携带的业务标识，以及所述包名和包签名进行组合，生成相应的字符串信息；对所述字符串信息进行哈希处理，得到与所述字符串信息对应的哈希值，将所述哈希值确定为所述校验密钥。

[0200] 为了实现上述应用程序校验方法，本发明实施例还提供了一种应用程序校验装置的硬件结构。图8为本发明实施例提供的一种应用程序校验装置的硬件结构示意图，图8所示的应用程序校验装置800包括：至少一个处理器801、存储器802、用户接口803和至少一个网络接口804。其中，应用程序校验装置800中各组成结构的功能与图7中示出的应用程序校验密钥的生成装置700中的各组成结构的功能属于同一构思，应用程序校验装置800的具体组成结构详见应用程序校验密钥的生成装置700的硬件结构的组成，这里不再赘述。

[0201] 本实施例中，所述应用程序校验装置800包括存储器802、处理器801及存储在存储器802上并能够由所述处理器801运行的可执行程序8021，所述处理器801运行所述可执行程序8021时实现：

[0202] 接收校验请求消息，其中，所述校验请求消息中携带有应用程序标识、业务标识，以及第一属性信息；获取预存储的与所述应用程序标识对应的第二属性信息；将所述第一属性信息与所述第二属性信息进行匹配，若匹配失败，则基于所述校验请求消息中携带的所述应用程序标识、业务标识，以及第一属性信息，生成第一校验密钥；将所述第一校验密钥与预先存储的第二校验密钥进行匹配，若匹配成功，确定对待校验的应用程序校验成功；其中，所述第二校验密钥是基于预存储的应用程序标识、业务标识，以及第二属性信息生成的；所述第一属性信息包括所述待校验的应用程序的包名和包签名。

[0203] 作为一种实施方式，所述处理器801运行所述可执行程序8021时实现：

[0204] 将所述校验请求消息中携带的所述应用程序标识、所述业务标识，以及所述待校验的应用程序的包名和包签名进行组合，生成相应的字符串信息；对所述字符串信息进行哈希处理，得到与所述字符串信息对应的第一哈希值，将所述第一哈希值确定为所述第一校验密钥。

[0205] 在示例性实施例中，本发明实施例还提供了一种存储介质，所述存储介质可为光盘、闪存或磁盘等存储介质，可选为非瞬间存储介质。

[0206] 其中，所述存储介质上存储有可执行程序8021，所述可执行程序8021被处理器801执行时实现：

[0207] 接收校验请求消息，其中，所述校验请求消息中携带有应用程序标识、业务标识，以及第一属性信息；获取预存储的与所述应用程序标识对应的第二属性信息；将所述第一属性信息与所述第二属性信息进行匹配，若匹配失败，则基于所述校验请求消息中携带的所述应用程序标识、业务标识，以及第一属性信息，生成第一校验密钥；将所述第一校验密钥与预先存储的第二校验密钥进行匹配，若匹配成功，确定对待校验的应用程序校验成功；其中，所述第二校验密钥是基于预存储的应用程序标识、业务标识，以及第二属性信息生成的；所述第一属性信息包括所述待校验的应用程序的包名和包签名。

[0208] 作为一种实施方式，所述可执行程序8021被处理器801执行时实现：

[0209] 将所述校验请求消息中携带的所述应用程序标识、所述业务标识，以及所述待校验的应用程序的包名和包签名进行组合，生成相应的字符串信息；对所述字符串信息进行哈希处理，得到与所述字符串信息对应的第一哈希值，将所述第一哈希值确定为所述第一校验密钥。

[0210] 本发明实施例的应用程序校验方法中通过根据校验请求消息中的应用程序标识、业务标识，以及第一属性信息所生成的第一校验密钥，与第二校验密钥进行匹配来确定校验是否成功，这样，针对同一应用程序标识的多个应用程序来说，仅需执行一次与应用程序交互加解密密钥的过程，可以有效缩短对多个应用程序进行校验的时间，提高校验效率。

[0211] 本发明实施例所记载的各技术方案之间，在不冲突的情况下，可以任意组合。

[0212] 本领域内的技术人员应明白，本发明的实施例可提供为方法、系统、或可执行程序产品。因此，本发明可采用硬件实施例、软件实施例、或结合软件和硬件方面的实施例的形式。而且，本发明可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质(包括但不限于磁盘存储器和光学存储器等)上实施的可执行程序产品的形式。

[0213] 本发明是参照根据本发明实施例的方法、设备(系统)、和可执行程序产品的流程图和/或方框图来描述的。应理解可由可执行程序指令实现流程图和/或方框图中的每一流程和/或方框、以及流程图和/或方框图中的流程和/或方框的结合。可提供这些可执行程序指令到通用计算机、专用计算机、嵌入式处理机或参考可编程数据处理设备的处理器以产生一个机器，使得通过计算机或参考可编程数据处理设备的处理器执行的指令产生用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的装置。

[0214] 这些可执行程序指令也可存储在能引导计算机或参考可编程数据处理设备以特定方式工作的计算机可读存储器中，使得存储在该计算机可读存储器中的指令产生包括指令装置的制造品，该指令装置实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能。

[0215] 这些可执行程序指令也可装载到计算机或参考可编程数据处理设备上，使得在计算机或参考可编程设备上执行一系列操作步骤以产生计算机实现的处理，从而在计算机或参考可编程设备上执行的指令提供用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的步骤。

[0216] 以上所述，仅为本发明的较佳实施例而已，并非用于限定本发明的保护范围，凡在本发明的精神和原则之内所作的任何修改、等同替换和改进等，均应包含在本发明的保护范围之内。

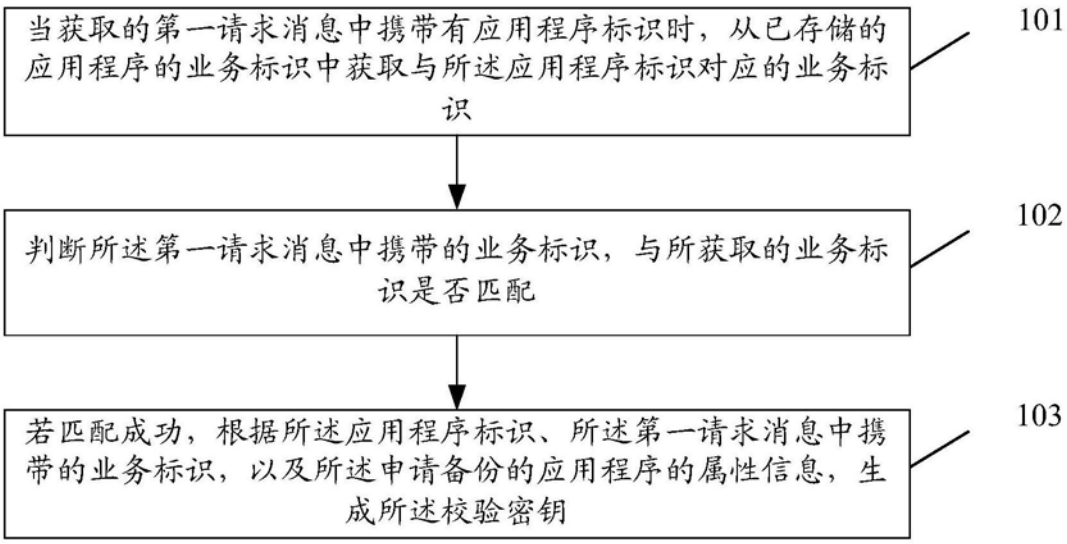

当获取的第一请求消息中携带有应用程序标识时，从已存储的应用程序的业务标识中获取与所述应用程序标识对应的业务标识
101
判断所述第一请求消息中携带的业务标识，与所获取的业务标识是否匹配
102
若匹配成功，根据所述应用程序标识、所述第一请求消息中携带的业务标识，以及所述申请备份的应用程序的属性信息，生成所述校验密钥
103

图1

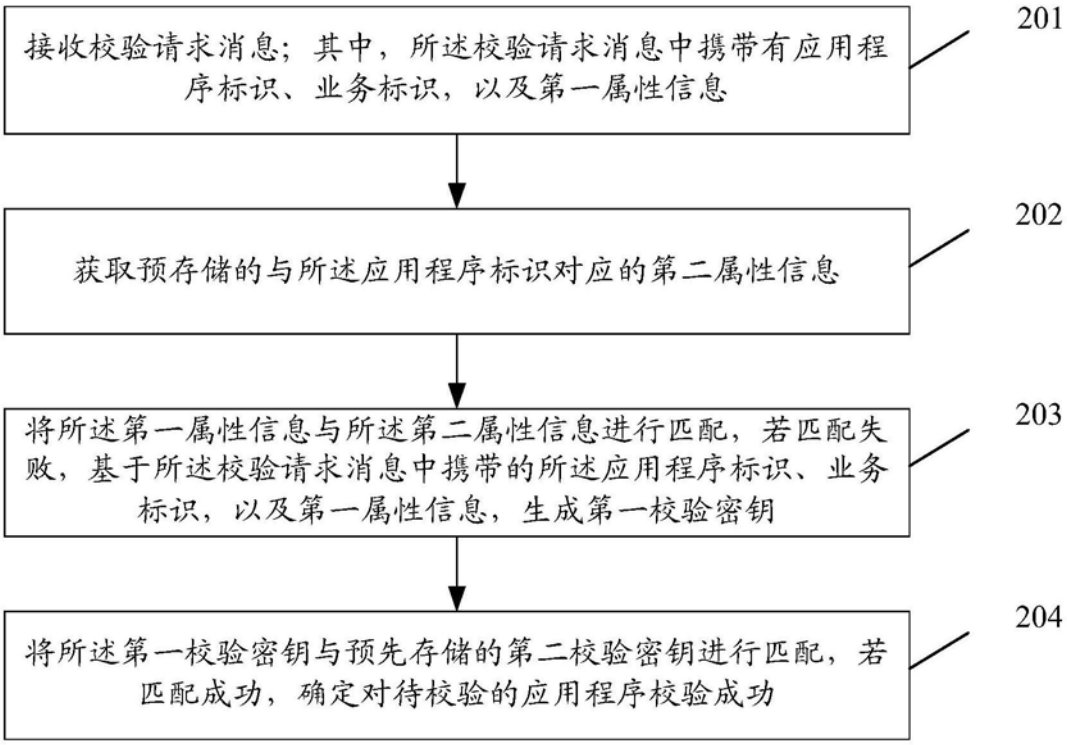

接收校验请求消息；其中，所述校验请求消息中携带有应用程序标识、业务标识，以及第一属性信息
201
获取预存储的与所述应用程序标识对应的第二属性信息
202
将所述第一属性信息与所述第二属性信息进行匹配，若匹配失败，基于所述校验请求消息中携带的所述应用程序标识、业务标识，以及第一属性信息，生成第一校验密钥
203
将所述第一校验密钥与预先存储的第二校验密钥进行匹配，若匹配成功，确定对待校验的应用程序校验成功
204

图2

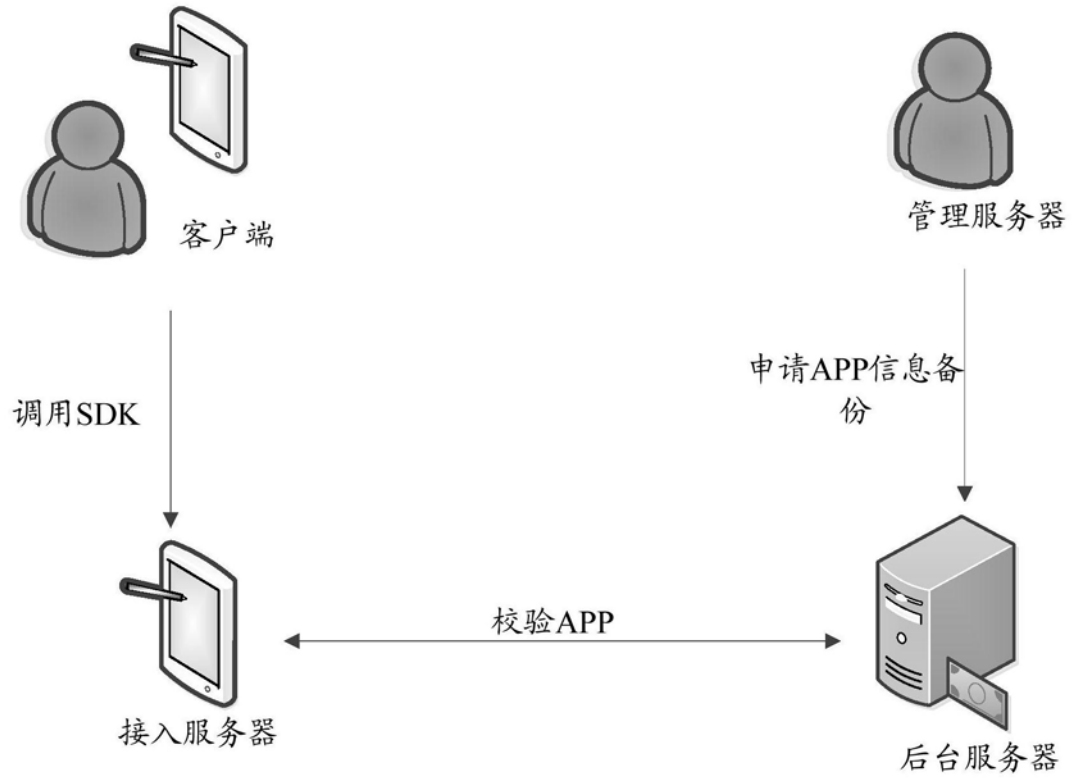

客户端
管理服务器
调用SDK
申请APP信息备份
校验APP
接入服务器
后台服务器

图3

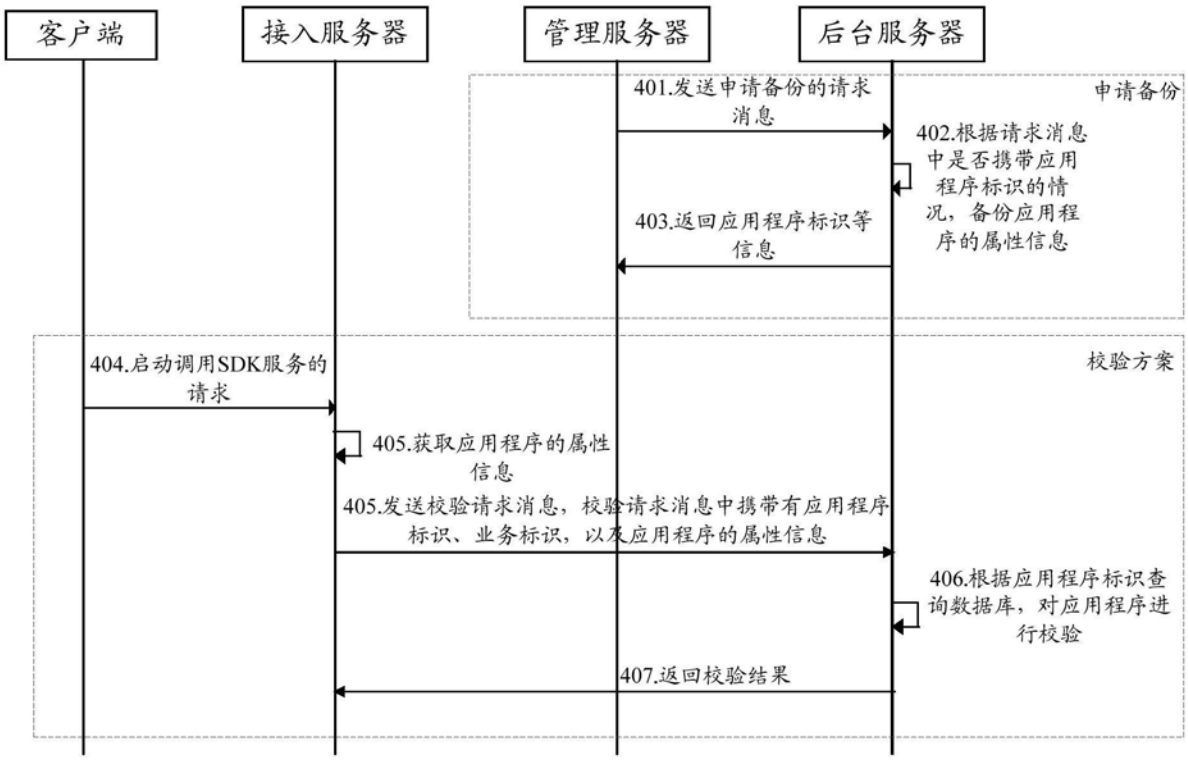

客户端
接入服务器
管理服务器
后台服务器
申请备份
401.发送申请备份的请求消息
402.根据请求消息中是否携带应用程序标识的情况，备份应用程序的属性信息
403.返回应用程序标识等信息
校验方案
404.启动调用SDK服务的请求
405.获取应用程序的属性信息
405.发送校验请求消息，校验请求消息中携带有应用程序标识、业务标识，以及应用程序的属性信息
406.根据应用程序标识查询数据库，对应用程序进行校验
407.返回校验结果

图4

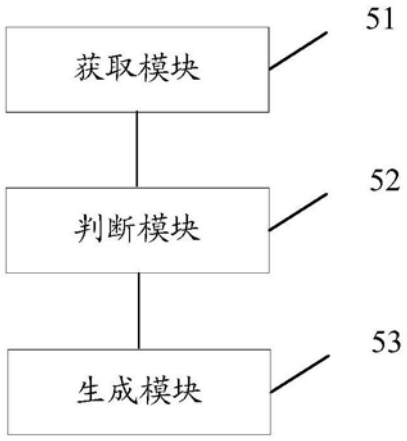


图5

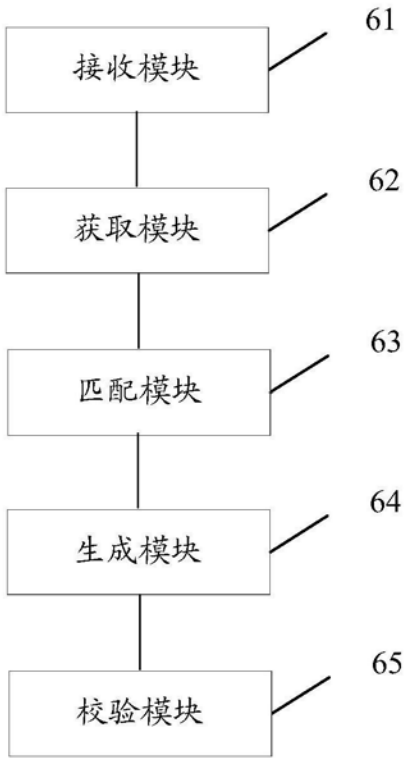


图6

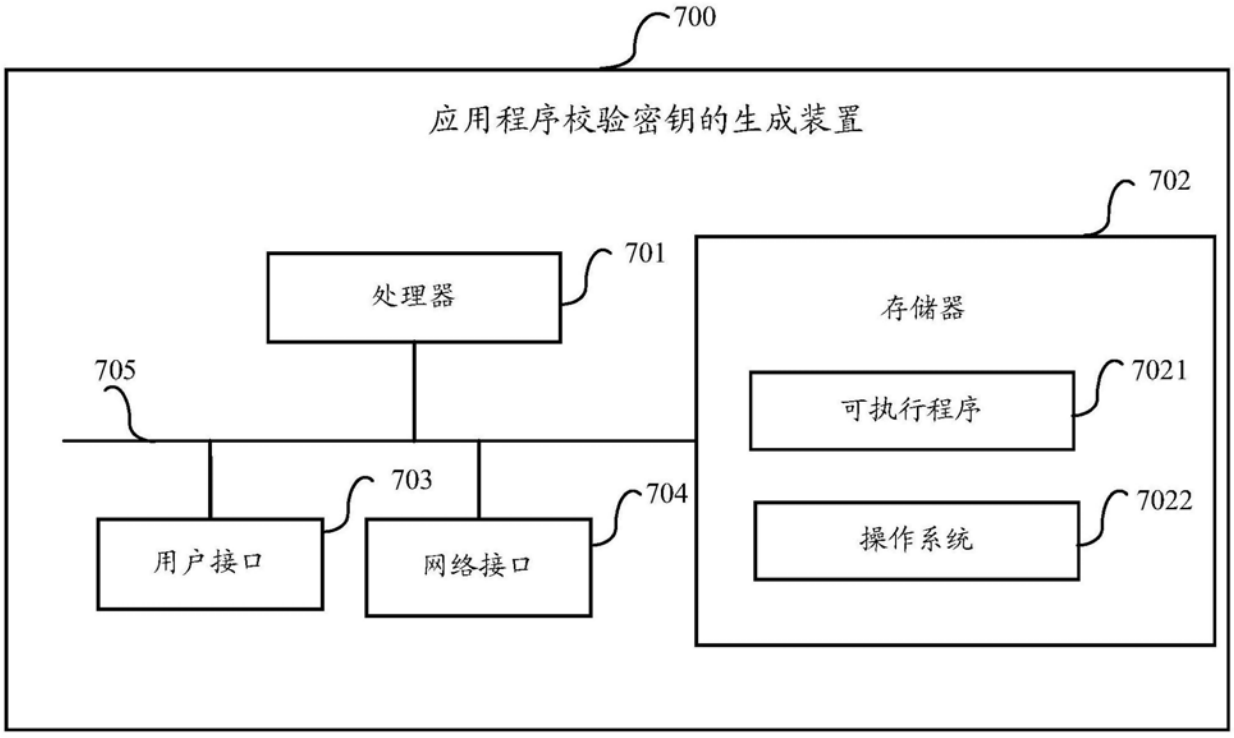

700
应用程序校验密钥的生成装置
702
701
处理器
存储器
705
7021
可执行程序
703
704
7022
用户接口
网络接口
操作系统

图7

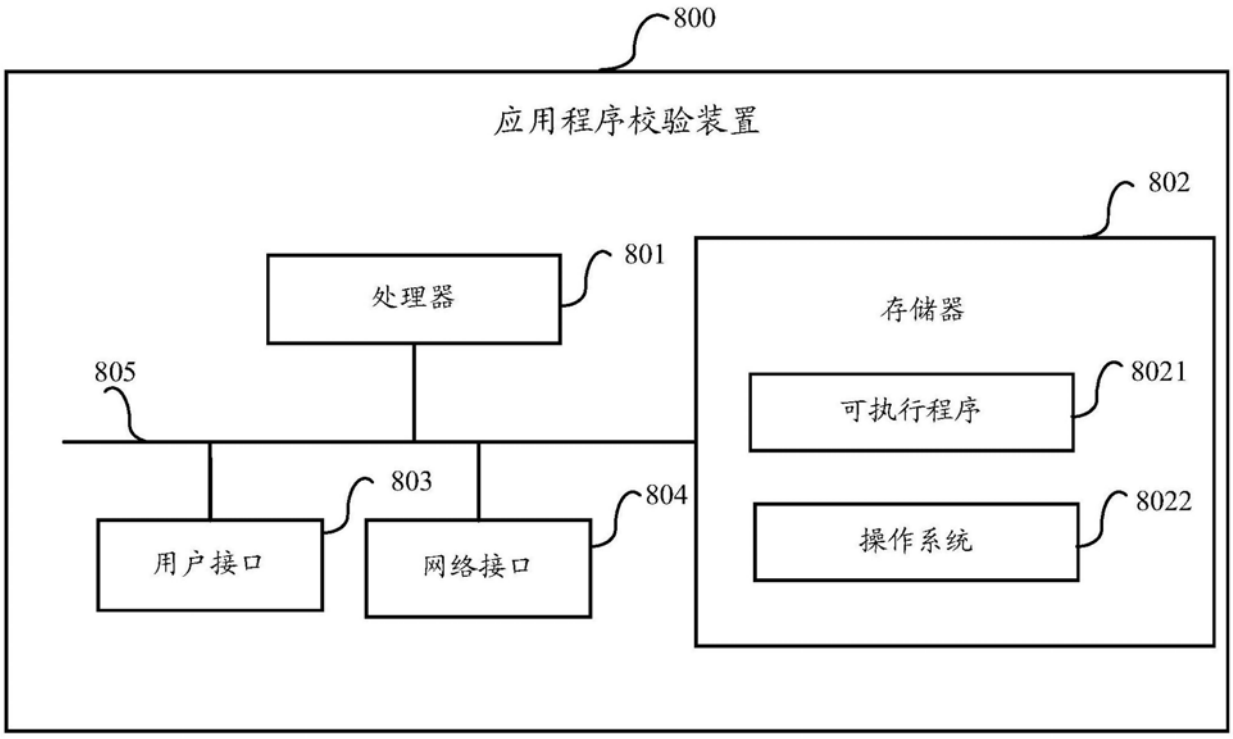

800
应用程序校验装置
802
801
处理器
存储器
805
8021
可执行程序
803
804
8022
用户接口
网络接口
操作系统

图8