



(12)发明专利申请

(10)申请公布号 CN 106126269 A

(43)申请公布日 2016.11.16

(21)申请号 201610397190.4

(22)申请日 2016.06.07

(71)申请人 中国建设银行股份有限公司
地址 100032 北京市西城区金融大街25号

(72)发明人 何伟明 邓玉 廖敏飞 吴孟晴
李文鹏 刘丽娟 董思 许腾
谭世殊 解敏

(74)专利代理机构 广州三环专利代理有限公司
44202
代理人 温旭 郝传鑫

(51)Int.Cl.
G06F 9/445(2006.01)

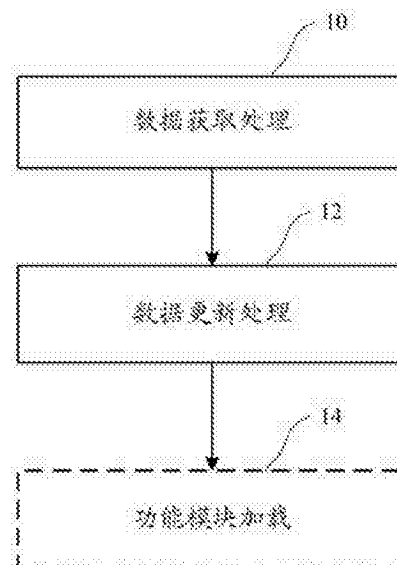
权利要求书1页 说明书5页 附图2页

(54)发明名称

应用更新方法、装置、系统及应用设计方法

(57)摘要

本发明公开了一种应用更新方法、装置、系统及应用设计方法,其中,所述应用更新方法包括:从服务器获取应用中待更新功能模块的更新数据;将所述更新数据保存至所述应用的数据目录,其中,在所述应用的数据目录中,所述应用的不同功能模块分别独立地保存。采用本发明,能够进行细粒度、低流量、易维护的更新处理;为用户提供更新和使用的无缝连接,提高用户体验;有效保证功能模块的安全调用和更新。



1. 一种应用更新方法,其特征在于,所述方法包括:
从服务器获取应用中待更新功能模块的更新数据;
将所述更新数据保存至所述应用的数据目录,其中,在所述应用的数据目录中,所述应用的不同功能模块分别独立地保存。
2. 如权利要求1所述的方法,其特征在于,所述方法还包括:
在将所述更新数据保存至所述应用的数据目录之后,加载所述待更新功能模块。
3. 如权利要求1所述的方法,其特征在于,从服务器获取应用中待更新功能模块的更新数据包括:
从服务器获取所述待更新功能模块的压缩包,所述压缩包中包含所述更新数据。
4. 如权利要求3所述的方法,其特征在于,将所述更新数据保存至所述应用的数据目录包括:
以替代所述待更新功能模块的原始压缩包的方式将从所述服务器获取的压缩包保存至所述应用的数据目录。
5. 如权利要求1所述的方法,其特征在于,所述方法还包括:
根据所述更新数据中携带的数字签名对所述更新数据进行有效性校验。
6. 一种应用更新装置,其特征在于,所述装置包括:
数据获取模块,用于从服务器获取应用中待更新功能模块的更新数据;
数据更新模块,用于将所述更新数据保存至所述应用的数据目录,其中,在所述应用的数据目录中,所述应用的不同功能模块分别独立地保存。
7. 如权利要求6所述的装置,其特征在于,所述装置还包括:
加载模块,用于在所述数据更新模块将所述更新数据保存至所述应用的数据目录之后,加载所述待更新功能模块。
8. 如权利要求6所述的装置,其特征在于,所述数据获取模块用于从所述服务器获取所述待更新功能模块的压缩包,所述压缩包中包含所述更新数据。
9. 如权利要求8所述的装置,其特征在于,所述数据更新模块用于以替代所述待更新功能模块的原始压缩包的方式将从所述服务器获取的压缩包保存至所述应用的数据目录。
10. 如权利要求6所述的装置,其特征在于,所述装置还包括:
校验模块,用于根据所述更新数据中携带的数字签名对所述更新数据进行有效性校验。
11. 一种应用更新系统,其特征在于,所述系统包括:
如权利要求6-10中任一项所述的应用更新装置;
服务器,保存有应用的待更新模块的更新数据。
12. 一种应用设计方法,其特征在于,所述方法包括:
为应用设计一个主模块,该主模块具有如权利要求6-10中任一项所述的应用更新装置的功能;
为所述应用设计至少一个用于实现具体功能的功能模块,并将所述功能模块独立地保存于所述应用的数据目录中。

应用更新方法、装置、系统及应用设计方法

技术领域

[0001] 本发明涉及应用程序更新领域,更为具体而言,涉及应用更新方法、装置、系统及应用设计方法。

背景技术

[0002] 目前,以安卓(Android)应用为例,业界流行的应用升级方法主要包括如下两种:

[0003] 1.全量下载、覆盖安装的方式。具体而言,在应用需要升级的时候,服务端通知客户端新版本的相关信息,然后客户端下载更新包到本地覆盖安装应用,从而实现应用的升级。

[0004] 2.增量下载、覆盖安装的方式。具体而言,客户端上送其版本信息到服务端;服务端根据其版本信息将新版本的apk包(应用的压缩安装包)与客户端对应版本的apk包做一个差异对比;客户端下载包含差异内容的差异增量包到本地,与原apk包合并后得到新的apk包,然后覆盖安装该新版本的apk包。

[0005] 此外,在CN201210410435号专利文献中,公开了一种根据用户使用应用的频率来分配应用更新的优先级以期能够最大化的减少不必要(从用户角度来看)的应用更新的方案。在CN201310157527号专利文献中,公开了一种通过类加载器和应用程序运行指令解析、在无用户干预的情况下实现应用程序的自动增量更新的方案。在CN201310359406号专利文献中,公开了一种通过维护一个拦截列表实现了对部分不安全软件升级的拦截,从而在一定程度上保护的应用升级的安全的方案。

[0006] 但是,无论是全量升级还是增量升级,都无可避免的使用到了覆盖安装的方式,只是下载安装包的形式不同。对于全量升级的方式而言,需要下载整个安装包,下载流量大,速度比较慢。对于增量安装的形式,需要同时维护所有旧版本与当前最新版本之间的差异增量包,以便使用各个版本的用户都能够得到升级服务。两种升级方式对比而已,增量升级的升级包更小,但是维护更加麻烦。

[0007] 考虑这样的应用场景:只升级应用中的某个模块,而并不改变其余模块的实现。为了实现这个微小的升级,如果使用全量升级的方式,则用户会有比较大的流量负担。如果使用增量升级的方式,则需要为该模块变动修订一个版本,并维护所有旧版本与新版本之间的差量包,当修改频繁的时候,服务端所需要维护的差异增量包就会以平方级数增加,增加维护压力。

发明内容

[0008] 鉴于现有技术的上述缺陷,本发明实施方式提供了一种应用更新方法、装置、系统及应用设计方法,能够以细粒度、低流量、易维护的升级方式实现应用升级。

[0009] 具体地,本发明实施方式提供了一种应用更新方法,其包括:

[0010] 从服务器获取应用中待更新功能模块的更新数据;

[0011] 将所述更新数据保存至所述应用的数据目录,其中,在所述应用的数据目录中,所

述应用的不同功能模块分别独立地保存。

[0012] 相应地,本发明实施方式还提供了一种应用更新装置,其包括:

[0013] 数据获取模块,用于从服务器获取应用中待更新功能模块的更新数据;

[0014] 数据更新模块,用于将所述更新数据保存至所述应用的数据目录,其中,在所述应用的数据目录中,所述应用的不同功能模块分别独立地保存。

[0015] 此外,本发明实施方式还提供了一种应用更新系统,其包括前述的应用更新装置以及服务器,所述服务器保存有应用的待更新模块的更新数据。

[0016] 此外,本发明实施方式还提供了一种应用设计方法,所述方法包括:

[0017] 为应用设计一个主模块,该主模块具有前述应用更新装置的功能;

[0018] 为应用设计至少一个用于实现具体功能的功能模块,并将所述功能模块独立地保存于所述应用的数据目录中。

[0019] 采用本发明实施方式具有下述有益效果:

[0020] 通过从服务器获取待更新功能模块的更新数据并将所述更新数据保存至所述应用的数据目录,能够对模块化设计的应用进行细粒度、低流量、易维护的更新处理;

[0021] 待更新功能模块在下载完成后即可使用,无需重新安装应用,为用户提供更新和使用的无缝连接,提高用户体验;

[0022] 通过在调用功能模块和使用下载的更新数据前进行有效性校验,能够有效保证功能模块的安全调用和更新。

附图说明

[0023] 图1是根据本发明实施例的一种应用更新方法的流程示意图;

[0024] 图2是根据本发明实施例的一种安卓应用的架构示意图及其与服务器通讯的示意图;

[0025] 图3是根据本发明实施例的一种应用更新装置的框图;

[0026] 图4是根据本发明实施例的一种应用更新系统的示意图。

具体实施方式

[0027] 为了便于理解本发明技术方案的各个方面、特征以及优点,下面结合附图对本发明进行具体描述。应当理解,下述的各种实施方式只用于举例说明,而非用于限制本发明的保护范围。

[0028] 图1是根据本发明实施例的一种应用更新方法的流程示意图,参照图1,所述方法包括:

[0029] 10:数据获取处理。具体而言,包括:从服务器获取应用中待更新功能模块的更新数据。

[0030] 需要说明的是,在本发明提及的应用是指运行于操作系统上的应用程序(app)。在本发明中的各个实施例中主要以安卓应用程序为例进行说明,但这并不表示本发明仅适用于安卓应用程序的更新,Android、ios和PC上的应用都可以,理论上所有的应用程序都可适用该方法。本领域技术人员根据本发明提供的思想而进行的方案改进、应用场景扩展等,均落在本发明的保护范围内。

[0031] 12:数据更新处理。具体而言,包括:将所述更新数据保存至所述应用的数据目录。其中,在所述应用的数据目录中,所述应用的不同功能模块分别独立地保存。其中,独立地保存是指功能模块分别作为一个整体保存。例如,一个功能模块对应一个压缩包。

[0032] 需要说明的是,在本发明中提及的应用是指的具有如下架构特点的应用:应用包括主模块和至少一个功能模块,其中,由主模块加载功能模块从而实现应用的具体功能。由于不同应用所包含的具体功能不同,例如,页面跳转、文字处理、语音处理等功能,因此,本发明不对所述具体功能进行限制。此外,主模块还可以具有与服务器通讯的功能以及数字签名校验功能。

[0033] 示例性地,图2是根据本发明实施例的一种安卓应用的架构示意图及其与服务器通讯的示意图。如图2所示,安卓应用包括应用主模块和多个功能模块。其中,各个功能模块分别用于实现应用的具体功能。主模块则主要用于与服务器通讯以及调用功能模块以实现应用的具体功能。

[0034] 具体而言,参照图2中的步骤1、2、3、4。步骤1和步骤2表示应用(客户端)与服务器进行交互以实现如下过程:从服务器获取功能模块更新列表以确定需要更新的功能模块;从服务器下载需要更新的功能模块。步骤3和步骤4表示主模块与功能模块之间的交互,例如,当针对功能模块1进行更新时,包括如下过程:应用对下载的新功能模块1进行有效性校验,在校验成功后用其替换数据目录中的旧的功能模块1;应用重新加载更新后的功能模块1。

[0035] 在安装具有图2所示架构的安卓应用时,可以从apk包中将各个功能模块所对应的压缩包(例如,jar格式的压缩包,一个压缩包对应一个功能模块)拷贝到应用中对应的数据目录中,也可以从网络下载jar包至对应的数据目录。当应用需要调用某功能模块时,从数据目录中进行读取、校验和加载即可。也就是说,在应用安装之后,压缩包就以与应用相独立又可被应用调用的形式存在。apk包只打包具有调用功能、通讯功能、加解密保护功能等的核心模块即可,而其它功能模块均可以采用本发明提供的方法进行动态更新。

[0036] 采用本发明提供的方法,通过处理10和12,能够对模块化设计的应用进行细粒度(例如,仅针对需要更新的功能模块)、低流量、易维护(例如,服务器端仅需保存各功能模块的更新的压缩包,无需维护不同版本的差异增量包)的更新处理。此外,待更新功能模块在下载完成后即可使用,无需重新安装应用,为用户提供更新和使用的无缝连接,提高用户体验。

[0037] 可选地,在本实施例的一种实现方式中,如图1中虚线框所示,在处理12之后,还可以进行处理14:功能模块加载。具体而言,加载所述待更新功能模块,从而实现该待更新功能模块在更新后的具体功能。

[0038] 可选地,在本实施例的一种实现方式中,处理10具体可包括:从服务器获取待更新功能模块的压缩包,所述压缩包中包含所述更新数据。例如,首先以服务器推送或者客户端上拉的方式获取功能模块的版本信息,然后根据版本信息判断是否需要更新,针对需要更新的功能模块(即,待更新功能模块)进行处理10。

[0039] 可选地,在本实施例的一种实现方式中,处理12具体可包括:以替代所述待更新功能模块的原始压缩包的方式将从所述服务器获取的压缩包保存至所述应用的数据目录。

[0040] 可选地,在本实施例的一种实现方式中,所述方法还包括:根据更新数据中携带的

数字签名对所述更新数据进行有效性校验。例如,在处理10之后、处理12之前,或者在调用功能模块前进行所述有效性校验。

[0041] 在本实现方式中,可以在各个功能模块中打入一份服务器端的有效公钥证书,以便在调用功能模块之前和获取更新数据之后进行有效性校验。例如,如图2所示,在平时启动应用并调用功能模块时,首先由主模块校验功能模块的有效性,如果功能模块有损坏或者是非法模块,则删除该模块,提示用户并退出应用;否则,调用功能模块以实现具体功能。

[0042] 采用本实现方式,通过在调用功能模块和使用下载的更新数据前进行有效性校验,能够有效保证功能模块的安全调用和更新。

[0043] 以上对本发明的方法实施例及其实现方式进行了说明,除非特别指出,本发明方法实施例中的各个实现方式可以适当地进行组合,这均落在本发明的保护范围内。

[0044] 图3是根据本发明实施例的一种应用更新装置的框图,参照图3,应用更新装置包括数据获取模块30和数据更新模块32。其中,数据获取模块30用于从服务器获取应用中待更新功能模块的更新数据。数据更新模块32,用于将所述更新数据保存至所述应用的数据目录,其中,在所述应用的数据目录中,所述应用的不同功能模块分别独立地保存。

[0045] 采用本实施例提供的应用更新装置,能够对模块化设计的应用进行细粒度、低流量、易维护的更新处理。此外,待更新功能模块在下载完成后即可使用,无需重新安装应用,为用户提供更新和使用的无缝连接,提高用户体验。

[0046] 可选地,在本实施例的一种实现方式中,如图3中的虚线框所示,应用更新装置还可以包括加载模块34,用于在数据更新模块32将所述更新数据保存至所述应用的数据目录之后,加载所述待更新功能模块。

[0047] 可选地,在本实施例的一种实现方式中,数据获取模块30具体用于从服务器获取待更新功能模块的压缩包,所述压缩包中包含所述更新数据。

[0048] 可选地,在本实施例的一种实现方式中,数据更新模块32具体用于以替代所述待更新功能模块的原始压缩包的方式将从所述服务器获取的压缩包保存至所述应用的数据目录。

[0049] 可选地,在本实施例的一种实现方式中,如图3中的虚线框所示,应用更新装置还可以包括校验模块36,用于根据更新数据(例如,压缩包)中携带的数字签名对更新数据进行有效性校验。采用该实现方式,能够有效保证功能模块的安全调用和更新。

[0050] 以上结合附图对本发明的装置实施例及其实现方式进行了说明,其中,关于相关名词、术语、范围的解释,以及关于各个模块所执行或可执行的处理的详细说明,请参见方法实施例中的相应说明,此处不再赘述。此外,除非特别指出,本发明装置实施例中的各个实现方式可以适当地进行组合,这均落在本发明的保护范围内。

[0051] 此外,结合图2和图3,本领域技术人员应当理解,图2所示的主模块可以具有图3所示的实施例或其实现方式的功能。

[0052] 图4是根据本发明实施例的一种应用更新系统的示意图,参照图4,所述系统包括应用更新装置和保存有应用的待更新模块的更新数据的服务器。其中,关于应用更新装置及其与服务器之间的交互的说明,请参照前文,此处不再赘述。

[0053] 本发明还提供一种应用设计方法以及采用该方法设计的应用。其中,所述应用设计方法包括:为应用设计一个主模块,该主模块具有图3所示应用更新装置的实施例和/或

其实现方式的功能(换言之,主模块具有数据获取模块30、数据更新模块32,或者进一步包括加载模块34和/或校验模块36);为所述应用设计至少一个用于实现具体功能的功能模块,并将所述功能模块独立地保存于所述应用的数据目录中。

[0054] 通过以上的实施方式的描述,本领域的技术人员可以清楚地了解到本发明可借助软件结合硬件平台的方式来实现。基于这样的理解,本发明的技术方案对背景技术做出贡献的全部或者部分可以以软件产品的形式体现出来,该计算机软件产品可以存储在存储介质中,如ROM/RAM、磁碟、光盘等,包括若干指令用以使得一台计算机设备(可以是个人计算机,服务器,或者网络设备等)执行本发明各个实施例或者实施例的某些部分所述的方法。

[0055] 本领域技术人员应当理解,以上所公开的仅为本发明的实施方式而已,当然不能以此来限定本发明之权利范围,依本发明实施方式所作的等同变化,仍属本发明权利要求所涵盖的范围。

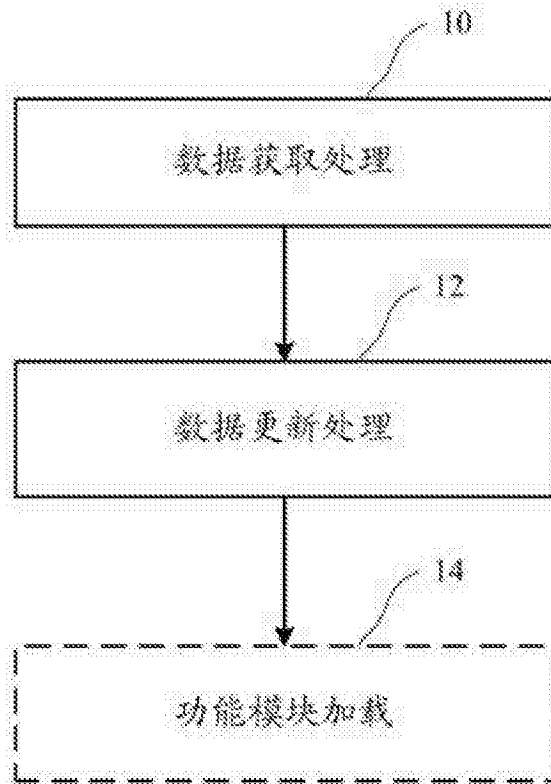


图1

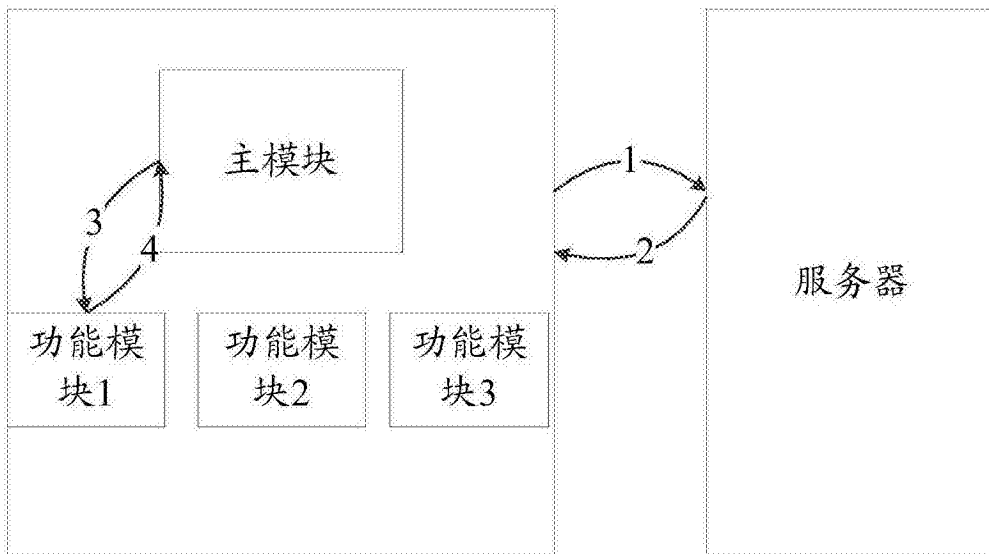


图2

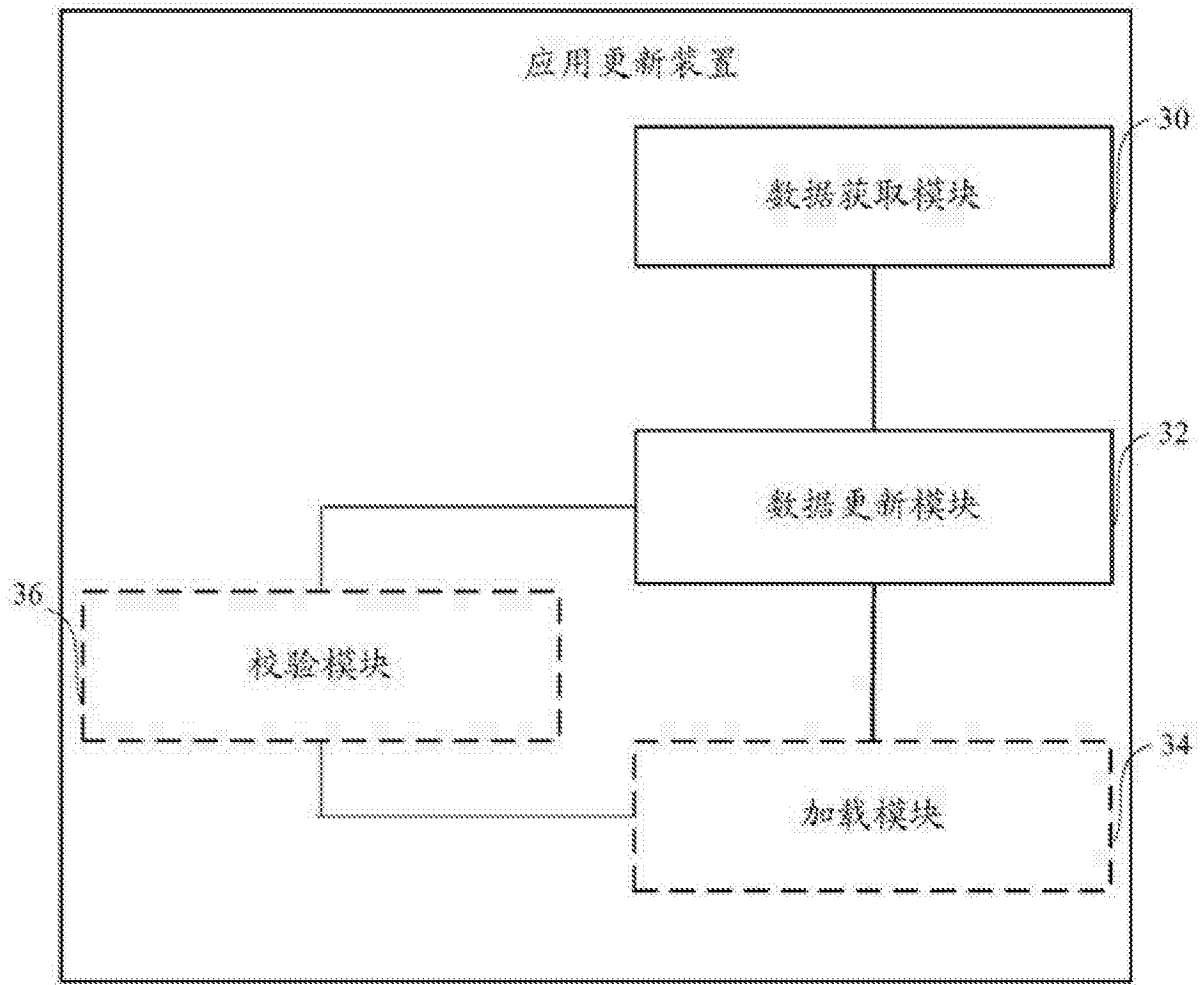


图3

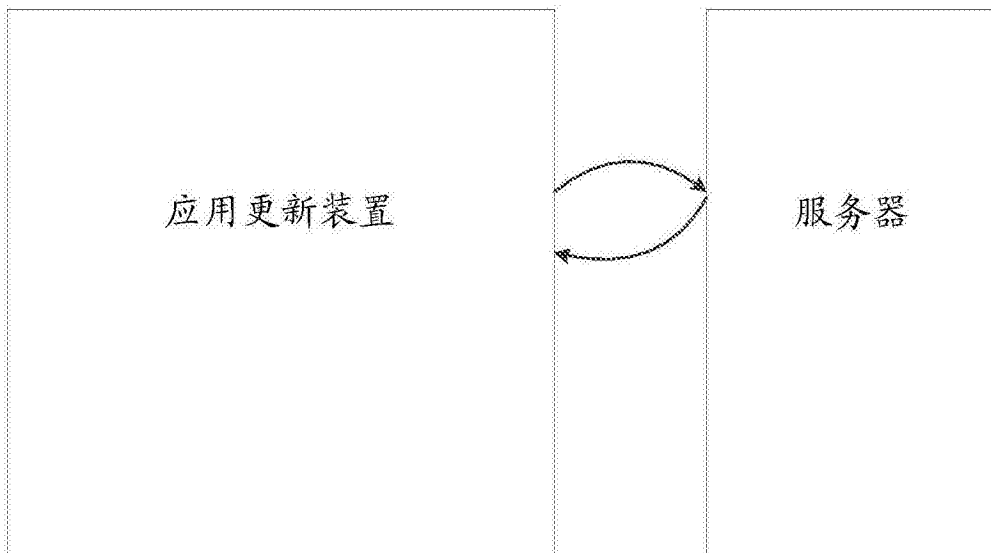


图4