



(12) 发明专利

(10) 授权公告号 CN 102065077 B

(45) 授权公告日 2013. 12. 18

(21) 申请号 201010542441. 6

(22) 申请日 2010. 11. 11

(73) 专利权人 中国联合网络通信集团有限公司  
地址 100033 北京市西城区金融大街 21 号

(72) 发明人 加雄伟

(74) 专利代理机构 北京同立钧成知识产权代理有限公司 11205

代理人 王申

(51) Int. Cl.

H04L 29/06 (2006. 01)

H04L 9/32 (2006. 01)

G06F 11/28 (2006. 01)

(56) 对比文件

CN 101789967 A, 2010. 07. 28, 摘要, 权利要求 1、5, 说明书第 2、5 页, 附图 1、2、5.

WO 00/65763 A2, 2000. 11. 02, 全文.

CN 101789967 A, 2010. 07. 28, 摘要, 权利要求 1、5, 说明书第 2、5 页, 附图 1、2、5.

CN 101404053 A, 2009. 04. 08, 全文.

US 6226784 B1, 2001. 05. 01, 全文.

CN 101339595 A, 2009. 01. 07, 全文.

孙青, 蒋伟, 陈波. 《代码签名技术及应用探讨》. 《电脑编程技巧与维护》. 2009, 第 21-26 页.

康金辉. 《基于数字校园网的客户端软件分发方法》. 《陕西理工学院学报(自然科学版)》. 2008, 第 24 卷(第 4 期), 全文.

黄君毅. 《基于 PKI/CA 架构的加密签名系统设计及实现》. 《万方数据-中山大学硕士学位论文》. 2005, 第 2、3、5、6 章.

审查员 汪辉

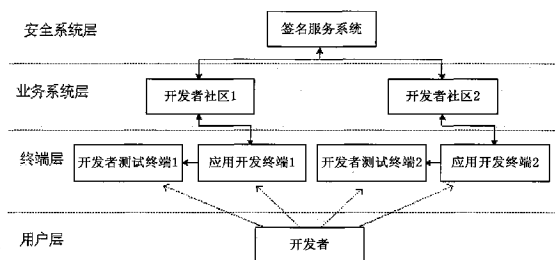
权利要求书3页 说明书10页 附图7页

(54) 发明名称

终端应用软件分发方法及系统

(57) 摘要

本发明提供一种终端应用软件分发方法及系统, 其中方法包括: 通过应用开发终端将开发者注册成为开发者社区的用户; 由所述应用开发终端开发应用软件; 由开发者测试终端测试所述应用软件; 由所述应用开发终端对通过测试的应用软件进行打包并提交给所述开发者社区。本发明无需限定于特定的终端产品, 具有较高的通用性及安全性, 可以由运营商搭建可管理、可运营、安全的应用软件可控分发体系; 并且, 该方法从开发者开发和测试应用软件的阶段便可以控制应用软件的分发, 因此具有较高的可控性。



1. 一种终端应用软件分发方法,其特征在于包括:
  - 通过应用开发终端将开发者注册成为开发者社区的用户;
  - 由所述应用开发终端开发应用软件;
  - 由开发者测试终端测试所述应用软件;
  - 由所述应用开发终端对通过测试的应用软件进行打包并提交给所述开发者社区;
  - 其中,所述通过应用开发终端将开发者注册成为开发者社区的用户包括:
    - 所述应用开发终端接收到来自于开发者的注册申请后,根据约定的加密算法生成证书申请信息,并向开发者社区发送包含所述证书申请信息的证书申请;
    - 所述开发者社区根据所述证书申请信息及与签名服务系统的约定生成开发者证书,所述开发者证书包括第一硬件标识串;
    - 所述签名服务系统对开发者证书进行签名;
    - 所述开发者社区将签名后的开发者证书反馈给所述应用开发终端;
    - 所述应用开发终端存储所述开发者证书并向所述开发者社区反馈注册申请处理结果;
  - 所述由开发者测试终端测试所述应用软件包括:
    - 所述应用开发终端根据所述应用软件生成签名文件;
    - 根据所述签名文件生成测试用安装包;
    - 根据所述测试用安装包生成测试用授权许可文件;
    - 将所述测试用安装包及所述测试用授权许可文件传输给所述开发者测试终端;
    - 当确认所述授权许可文件的合法性及有效性后,在所述开发者测试终端上安装并测试所述测试用安装包;
    - 所述在所述开发者测试终端上安装并测试所述测试用安装包包括:
      - 从所述测试用安装包中分离出应用软件、能力文件和签名文件,并找到相应的安装包标识;
      - 根据所述安装包标识查找相应的测试用授权许可文件;
      - 对所述测试用授权许可文件进行检查,当该测试用授权许可文件中的签名证书与所述签名文件中的签名证书相同且均为开发者证书时,从所述开发者证书中分离出所述第一硬件标识串,并获取所述开发者测试终端的第二硬件标识串;
      - 判断所述第一硬件标识串与所述第二硬件标识串是否匹配,当匹配时,安装所述应用软件及能力文件。
2. 根据权利要求1所述的方法,其特征在于所述应用开发终端根据所述应用软件生成签名文件包括:
  - 生成所述应用软件的摘要;
  - 生成所述应用软件相应能力文件的摘要;
  - 计算所述应用软件的摘要与所述能力文件的摘要的签名信息;
  - 根据所述签名信息生成签名文件。
3. 根据权利要求1所述的方法,其特征在于所述根据所述测试用安装包生成测试用授权许可文件包括:
  - 根据所述测试用安装包生成购买信息;

按约定的摘要算法及摘要加密算法,生成加密后的所述购买信息的摘要;  
根据所述购买信息的摘要按约定的规则生成所述测试用授权许可文件。

4. 一种终端应用软件分发系统,其特征在于包括应用开发终端、开发者测试终端及开发者社区服务器,其中:

所述应用开发终端用于将开发者注册成为开发者社区的用户,并开发应用软件;

所述开发者测试终端用于测试所述应用软件;

所述应用开发终端还用于对通过所述测试的应用软件进行打包并提交给所述开发者社区服务器;

所述开发者社区服务器用于保存由应用开发终端提交的所述应用软件以供下载;

其中,所述系统还包括签名服务系统,其中:

所述应用开发终端包括:

加密模块,用于当所述应用开发终端接收到来自于开发者的注册申请后,根据约定的加密算法生成证书申请信息;

证书申请模块,用于向所述开发者社区服务器发送包含所述证书申请信息的证书申请;

存储模块,用于存储来自于所述开发者社区服务器的开发者证书,所述开发者证书包括第一硬件标识串;

结果反馈模块,用于向所述开发者社区服务器反馈注册申请处理结果;

所述开发者社区服务器包括:

证书生成模块,用于根据所述证书申请信息及与签名服务系统的约定生成开发者证书;

证书反馈模块,用于将所述签名服务系统签名后的开发者证书反馈给所述应用开发终端;

所述签名服务系统用于对证书生成模块生成的所述开发者证书进行签名;

所述应用开发终端还包括:

签名文件生成模块,用于根据所述应用软件生成签名文件;

安装包生成模块,用于根据签名文件生成模块生成的所述签名文件生成测试用安装包;

许可文件生成模块,用于根据安装包生成模块生成的所述测试用安装包生成测试用授权许可文件;

传输模块,用于将安装包生成模块生成的所述测试用安装包及许可文件生成模块生成的所述测试用授权许可文件传输给所述开发者测试终端;

所述开发者测试终端包括:

分离模块,用于从所述测试用安装包中分离出应用软件、能力文件和签名文件,并找到相应的安装包标识;

查找模块,用于根据分离模块分离出的所述安装包标识查找相应的测试用授权许可文件;

检查模块,用于对所述测试用授权许可文件进行检查;

标识串处理模块,用于当检查模块检查出所述测试用授权许可文件中的签名证书与所

述签名文件中的签名证书相同且均为开发者证书时,从所述开发者证书中分离出所述第一硬件标识串,并获取所述开发者测试终端的第二硬件标识串;

判断模块,用于判断所述第一硬件标识串与所述第二硬件标识串是否匹配;

安装模块,用于判断模块判断出所述第一硬件标识串与所述第二硬件标识串匹配时,安装所述应用软件及能力文件。

## 终端应用软件分发方法及系统

### 技术领域

[0001] 本发明涉及一种终端应用软件分发方法及系统,属于智能终端技术领域。

### 背景技术

[0002] 智能终端是指智能手机、电子书阅读器等终端设备。智能终端的安全问题主要包括用户数据(例如:联系人、账号、密码、照片等)的安全、终端资源(例如:摄像设备、录音设备、用户身份卡、网络连接设备、存储设备等)的安全、网络资源(例如:网上存储的联系人、照片等资源)的安全等。

[0003] 智能终端的发展离不开智能终端上的应用软件的发展壮大。应用软件由各种各样的软件提供商或软件设计人员设计,出于安全考虑,用户需要可信的应用软件下载途径,针对这种情况,多家终端设备商、系统制造商或运营商提供多种管控应用软件分发的技术方案。

[0004] 例如,美国苹果公司的软件商店技术方案是解决应用软件分发的方案之一。苹果公司终端软件的开发者把开发的应用软件上传给苹果公司,苹果公司审核成功后,把应用软件放在软件商店中,供苹果公司终端用户下载和使用。苹果公司的智能终端用户信任苹果公司的审核结果,可以较安心的从苹果公司的软件商店中下载应用软件。

[0005] 再例如,美国谷歌公司的软件商店技术方案也是解决应用软件分发的方案之一。与苹果公司的相关方案相比,谷歌公司不审核开发者的应用软件。因此,谷歌公司终端的用户并不能完全信任谷歌公司的软件商店上的应用软件。

[0006] 现有的应用软件分发方案虽然在一定程度上解决了应用软件的分发问题,但上述方案只能用于特定公司的特定终端产品,而其它公司不能使用,因此不具有通用性,其使用范围非常有限。

### 发明内容

[0007] 本发明提供一种终端应用软件分发方法及系统,用以提高软件分发的通用性和安全性。

[0008] 本发明一方面提供一种终端应用软件分发方法,其中包括:

[0009] 通过应用开发终端将开发者注册成为开发者社区的用户;

[0010] 由所述应用开发终端开发应用软件;

[0011] 由开发者测试终端测试所述应用软件;

[0012] 由所述应用开发终端对通过测试的应用软件进行打包并提交给所述开发者社区。

[0013] 本发明另一方面提供一种终端应用软件分发系统,其中包括:应用开发终端、开发者测试终端及开发者社区服务器,其中:

[0014] 所述应用开发终端用于将开发者注册成为开发者社区的用户,并开发应用软件;

[0015] 所述开发者测试终端用于测试所述应用软件;

[0016] 所述应用开发终端还用于对通过所述测试的应用软件进行打包并提交给所述开

发者社区服务器；

[0017] 所述开发者社区服务器用于保存由应用开发终端提交的所述应用软件以供下载。

[0018] 本发明无需限定于特定的终端产品，具有较高的通用性及安全性，可以由运营商搭建可管理、可运营、安全的应用软件可控分发体系；并且，该方法从开发者开发和测试应用软件的阶段便可以控制应用软件的分发，因此具有较高的可控性。

### 附图说明

[0019] 为了更清楚地说明本发明实施例或现有技术中的技术方案，下面将对实施例或现有技术描述中所需要使用的附图作一简单地介绍，显而易见地，下面描述中的附图是本发明的一些实施例，对于本领域普通技术人员来讲，在不付出创造性劳动的前提下，还可以根据这些附图获得其他的附图。

[0020] 图 1 为本发明所述四层可控软件分发体系结构的分层示意图；

[0021] 图 2 为本发明所述终端应用软件分发方法实施例的流程图；

[0022] 图 3 为图 2 所示步骤 100 的具体步骤信令图；

[0023] 图 4A 为图 2 所示步骤 300 的具体步骤流程图；

[0024] 图 4B 为图 4A 所示步骤 310 的具体步骤流程图；

[0025] 图 4C 为图 4A 所示步骤 330 的具体步骤流程图；

[0026] 图 4D 为图 4A 所示步骤 350 的具体步骤流程图；

[0027] 图 5A 为图 4A 所示步骤 320 中所述打包后形成的测试用安装包的数据格式示意图；

[0028] 图 5B 为图 5A 所示数据格式中相应的索引格式示意图；

[0029] 图 6 为本发明所述终端应用软件分发系统实施例的结构示意图；

[0030] 图 7 为图 6 所示应用开发终端 10 的一种可选结构示意图；

[0031] 图 8 为图 6 所示开发者社区服务器 30 的一种可选结构示意图；

[0032] 图 9 为图 6 所示应用开发终端 10 的另一种可选结构示意图；

[0033] 图 10 为图 6 所示开发者测试终端 20 的可选结构示意图。

### 具体实施方式

[0034] 为使本发明实施例的目的、技术方案和优点更加清楚，下面将结合本发明实施例中的附图，对本发明实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例是本发明一部分实施例，而不是全部的实施例。基于本发明中的实施例，本领域普通技术人员在没有作出创造性劳动前提下所获得的所有其他实施例，都属于本发明保护的范围。

[0035] 首先介绍为本实施例所述方法构建的四层可控软件分发体系结构，如图 1 所示，包括：

[0036] 1、安全系统层

[0037] 该层主要包括签名服务系统。签名服务系统是应用可控分发安全体系中的基础安全设施。签名服务系统为开发者社区提供证书与密钥管理服务、开发者证书签名和验证服务、应用软件签名与验证服务。

[0038] 2、业务系统层

[0039] 该层主要包括开发者社区。在业务系统层,可以有多个开发者社区。一个签名服务系统可以同时为多个开发者社区提供安全控制服务。开发者社区管理开发者的信息,包括:开发者描述信息、开发者证书、开发者密钥、开发者终端信息、开发者级别信息等。

### [0040] 3、终端层

[0041] 该层主要包括开发者测试终端和应用开发终端。开发者使用应用开发终端来开发终端应用软件,使用开发者测试终端测试开发的终端应用软件。开发者社区为开发者的应用开发终端提供应用安全控制服务。

### [0042] 4、用户层

[0043] 该层主要包括终端应用软件的开发者 and 使用者(用户)。开发者可以有多个开发者测试终端或多个应用开发终端。使用者可以有多个用户终端。用户终端与开发者测试终端可以相同。开发者使用开发者应用开发终端开发终端应用软件,使用开发者测试终端测试应用软件。使用者在用户终端上使用终端应用软件。

[0044] 在上述四层可控软件分发体系结构中,使用证书标识各个功能实体。使用签名服务系统证书(也称为根证书)标识签名服务系统。使用开发者社区证书标识开发者社区。使用开发者证书标识终端应用软件的开发者,使用用户证书标识终端应用软件的使用者。使用根证书签名开发者社区证书。使用开发者社区证书签名开发者证书。用户证书可以由根证书签名,也可以由开发者社区证书签名。具体可以使用常见的证书格式,例如 X509。与证书对应的是,相关功能实体的公钥和私钥。签名服务系统生成和管理自己的公钥和私钥。开发者社区、开发者、用户的证书对应的公钥和私钥可以由签名服务系统生成,然后通过安全途径分发给相应的功能实体。其中,所有证书使用相同的密钥和摘要算法。在证书中,需要标识使用的密钥算法和摘要算法。

[0045] 图 2 为本发明所述终端应用软件分发方法实施例的流程图,如图所示,该方法包括如下步骤:

[0046] 步骤 100,开发者通过应用开发终端注册成为开发者社区的用户。

[0047] 通过本步骤,所述开发者可以得到开发者证书和密钥。

[0048] 步骤 200,由应用开发终端开发应用软件。

[0049] 具体地,开发者可以使用应用开发终端的应用编程工具编辑、编译、链接和测试应用软件,并使用应用开发终端编辑和测试能力文件。

[0050] 步骤 300,由开发者测试终端测试应用软件。

[0051] 具体地,开发者在开发者测试终端上测试应用软件之前,或者向开发者社区提交应用软件之前,可以先对应用软件打包生成测试用安装包,然后把测试用安装包推送到开发者测试终端或者开发者社区。该测试用安装包中包含应用软件、能力文件和签名文件。

[0052] 步骤 400,由应用开发终端对通过测试的应用软件进行打包并提交给开发者社区。

[0053] 此后,用户终端可以到相应的开发者社区下载获取应用软件,以实现终端应用软件的分发。

[0054] 本实施例所述终端应用软件的分发方法无需限定于特定的终端产品,具有较高的通用性及安全性,可以由运营商搭建可管理、可运营、安全的应用软件可控分发体系;并且,该方法从开发者开发和测试应用软件的阶段便可以控制应用软件的分发,因此具有较高的可控性。

[0055] 如图 3 所示,上述步骤 100 可以具体包括如下步骤:

[0056] 步骤 101,开发者向应用开发终端发送注册申请。

[0057] 步骤 102,应用开发终端根据约定的加密算法生成证书申请信息。

[0058] 其中,所述证书申请信息包括开发者名称、开发者描述、公钥、私钥、开发者测试终端的硬件标识等信息。所述开发者测试终端的硬件标识可以是相关终端的 CPU 序列号、硬盘序列号、网络设备号、用户身份卡设备号等,或者由这些硬件设备号生成的摘要等,用于在开发者测试终端测试时,应用安装引擎识别被测试应用是否可以安装到开发者测试终端的依据。因此,要求应用开发终端与开发者测试终端的应用安装引擎使用相同的算法生成硬件标识号。开发者证书对应的公钥与私钥可以由签名服务系统生成,然后通过安全途径分发给开发者应用开发终端。所述约定的加密算法可以采用椭圆曲线密码编码 (Elliptic Curves Cryptography,简称:ECC) 算法及公钥加密算法 (RSA) 等算法,这些算法可以由签名服务系统约定。

[0059] 步骤 103,应用开发终端向开发者社区发送包含上述证书申请信息的证书申请。

[0060] 步骤 104,开发者社区根据所述证书申请判断是否接受开发者的注册申请,如果接受,则根据所述证书申请信息及与签名服务系统的约定生成开发者证书,否则转至步骤 108。

[0061] 其中,所述开发者证书的内容至少包括:

[0062] 1) 证书格式与版本,可以采用 X.509 格式;

[0063] 2) 证书编码方法,可以使用 BASE64 编码方式;

[0064] 3) 签名算法,可以使用无线局域网鉴别和保密基础结构 (Wireless LAN Authentication and Privacy Infrastructure,简称:WAPI) ECC 算法;

[0065] 4) 摘要算法,可以采用缩微图算法 (SHA-1);

[0066] 5) 证书序列号,由签名服务系统生成,可以是随机数;

[0067] 6) 证书主题,可以包括国家标识、开发者类型、开发者测试终端的硬件标识串(可以多个)、开发者的安全级别、开发者在开发者社区的账号等,为了便于说明,在本实施例中将开发者证书中包含的上述开发者测试终端的硬件标识串简称为第一硬件标识串;

[0068] 7) 证书的签名机构标识,也即开发者社区的标识;

[0069] 8) 证书摘要,用于检测开发者证书。

[0070] 开发者证书对应的公钥存储在开发者证书中。开发者证书对应的私钥存储在应用开发终端的安全存储地点,并可以以加密的方式存储。应用开发终端提供安全存储和访问开发者证书对应的私钥的方法和设施。

[0071] 步骤 105,开发者社区请求签名服务系统签名开发者证书。

[0072] 步骤 106,签名服务系统对开发者证书进行签名,并将签名后的开发者证书反馈给开发者社区。

[0073] 其中,签名服务系统可以使用开发者社区证书及对应的私钥签名开发者证书,也可以使用根证书及对应的私钥签名开发者证书。如果由签名服务系统生成开发者证书对应的公钥和私钥,则可以一并把所述公钥和私钥反馈给开发者社区。

[0074] 具体的签名过程可以包括:签名服务系统根据开发者社区提供的证书申请信息生成开发者证书 A;签名服务系统把开发者证书 A 的内容作为输入源,按约定的摘要算法(例



如, SHA-1) 计算开发者证书的摘要, 得到摘要 A; 签名服务系统使用开发者社区证书对应的私钥 (或者根证书对应的私钥) 按约定的摘要签名算法 (例如, ECC) 加密摘要 A 得到摘要 B; 签名服务系统把摘要 B 加入到开发者证书 A 的约定的地方, 得到开发者证书 B。此时, 开发者证书 B 即是签名后的开发者证书。

[0075] 步骤 107, 开发者社区存储签名后的开发者证书、开发者公钥等信息。

[0076] 如果由签名服务系统生成开发者证书对应的公钥和私钥, 则开发者社区也需要存储所述开发者私钥。

[0077] 步骤 108, 开发者社区反馈证书申请处理结果给应用开发终端。

[0078] 具体地, 如果开发者已注册并已有开发者证书, 则在步骤 104 中的开发者社区拒绝开发者的证书申请, 相应地在本步骤中的证书申请处理结果则表明证书申请失败; 如果通过步骤 105 ~ 107 成功申请到证书, 则在本步骤中的证书申请处理结果则表明证书申请成功。

[0079] 步骤 109, 应用开发终端存储开发者社区的证书申请处理结果。

[0080] 如果开发者社区接受开发者的证书申请, 则存储开发者证书以及相应的公钥与私钥等信息。

[0081] 步骤 110, 应用开发终端向开发者反馈注册申请处理结果。

[0082] 具体地, 如果证书申请成功, 则该注册申请处理结果为注册申请成功; 如果证书申请失败, 则该注册申请处理结果为注册申请失败。

[0083] 如图 4A 所示, 上述步骤 300 可以具体包括如下步骤:

[0084] 步骤 310, 应用开发终端根据所述应用软件生成签名文件。

[0085] 具体地, 可以通过应用开发终端的签名打包工具根据约定的规则, 以及应用软件、能力文件、开发者证书、开发者私钥等信息, 生成签名文件。签名文件的内容至少包括:

[0086] 1) 开发者证书相关的内容: 开发者证书的类型、开发者证书的编码方式、开发者证书内容, 开发者证书作为签名证书;

[0087] 2) 开发者社区证书相关的内容: 开发者社区证书的类型、开发者社区证书的编码方式、开发者社区证书内容, 开发者社区证书作为可信任的证书;

[0088] 3) 应用软件摘要相关的内容: 应用软件摘要的编码方式、标识与摘要内容;

[0089] 4) 能力文件摘要相关的内容: 能力文件摘要的编码方式、标识与摘要内容;

[0090] 5) 摘要算法相关的内容: 摘要算法标识, 签名文件中使用的摘要算法相同;

[0091] 6) 签名文件摘要相关的内容: 签名文件摘要的编码方式、摘要内容。

[0092] 签名文件可以使用可扩展标记语言 (Extensible Markup Language, 简称: XML) 文档格式。在具体实施例中, 可以使用下表中的描述方法, 如下所示:

[0093] 1) 签名文件使用 XML 文档格式, UTF-8 编码;

[0094] 2) 证书使用 X509 格式, BASE64 编码;

[0095] 3) 摘要算法使用 WAPI-SHA1 算法, BASE64 编码;

[0096] 4) 签名的加密算法使用 ECC 算法, BASE 64 编码。

[0097] 具体编码内容如下:

[0098] <? xml version = " 1.0" encoding = " utf-8" ? >

[0099] < ! -- 开发者证书 -- >

```

[0100] <SignCert type = " x509" encoding = " base64" >.....</SignCert>
[0101] <! -- 开发者社区证书 -->
[0102] <TrustCert type = " x509" encoding = " base64" >.....</TrustCert>
[0103] <! -- 应用软件摘要与能力文件摘要 -->
[0104] <Digests encoding = " base64" >
[0105] <DigestValue name = " application" >.....</DigestValue>
[0106] <DigestValue name = " manifest" >.....</DigestValue>
[0107] </Digests>
[0108] <! -- 摘要算法 -->
[0109] <Algorithm name = " WAPI-SHA1" />
[0110] <! -- 签名文件摘要 -->
[0111] <Signature encoding = " base64" algorithm = "ECC  "
[0112] >.....</Signature>
[0113] </Signed>

```

[0114] 步骤 320, 根据所述签名文件生成测试用安装包。

[0115] 具体地, 应用开发终端的签名打包工具把应用软件、能力文件、签名文件按约定的规则组合成一个文件, 组合的文件称为应用软件包, 在测试时, 也称为测试用安装包。该测试用安装包存储的数据可以依次为: 应用软件数据包、能力文件数据包、签名文件数据包、应用软件数据包索引、能力文件数据包索引、签名文件数据包索引、索引数量、版本号。

[0116] 打包后形成的测试用安装包的数据格式如图 5A 所示, 其相应的索引格式如图 5B 所示。该数据格式既适用于打包应用软件, 也适用于打包授权许可文件。

[0117] 如图 5A 所示, 数据被打包后, 整体数据包分为四个部分: 数据区、索引区、索引数量与版本号。其中: 数据区依次存储数据包, 例如, 在打包应用软件时, 数据区存储应用软件数据、能力文件数据、签名文件数据。这些数据可以压缩, 也可以不压缩。数据区内的数据包不分先后顺序; 索引区依次存储数据区内数据包的索引信息, 每个索引由 16 个字节构成, 如图 5B 所示, 依次存储数据包的类型 (4 字节)、数据包距整体数据包的文件的字节偏移量 (4 字节)、数据包的字节长度 (4 字节)、保留字节 (4 字节)。数据包的类型可以根据业务需要定义, 例如, 数据包可以是应用软件、能力文件、签名文件、购买信息等; 索引数量存储整体数据包中包含的索引的个数; 版本号存储整体数据包的版本号。

[0118] 步骤 330, 根据所述测试用安装包生成测试用授权许可文件。

[0119] 具体地, 可以由应用开发终端的测试授权工具生成上述测试用授权许可文件。该测试用授权许可文件的内容至少包括:

[0120] 1) 开发者证书相关的内容: 开发者证书的类型、开发者证书的编码方式、开发者证书内容, 开发者证书作为签名证书;

[0121] 2) 购买信息的摘要签名相关的内容: 购买信息摘要的编码方式、标识与摘要内容;

[0122] 3) 摘要算法相关的内容: 摘要算法标识, 签名文件中使用的摘要算法相同。

[0123] 授权许可文件可以使用 XML 文档格式。在具体实施例中, 可以使用下表中的描述方法, 如下所示:

[0124] 1) 授权许可文件使用 XML 文档格式, UTF-8 编码;

[0125] 2) 证书使用 X509 格式, BASE64 编码;

[0126] 3) 摘要算法使用 WAPI-SHA1 算法, BASE64 编码;

[0127] 4) 签名的加密算法使用 ECC 算法, BASE64 编码。

[0128] 具体编码内容如下:

[0129] <? xml version = " 1.0" encoding = " utf-8" ? >

[0130] <! -- 开发者证书 -->

[0131] <SignCert type = " x509" encoding = " base64" >.....</SignCert>

[0132] <! -- 购买信息签名 -->

[0133] <Digests encoding = " base64" algorithm = "ECC">

[0134] <DigestValue name = " license" >.....</DigestValue>

[0135] </Digests>

[0136] <! -- 摘要算法 -->

[0137] <Algorithm name = " WAPI-SHA1" />

[0138] </Signed>

[0139] 步骤 340, 将所述测试用安装包及所述测试用授权许可文件传输给所述开发者测试终端。

[0140] 具体地, 可以通过推送或复制等方式传输给开发者测试终端。

[0141] 步骤 350, 当确认所述授权许可文件的合法性及有效性后, 在所述开发者测试终端上安装并测试所述测试用安装包。

[0142] 具体地, 可以由开发者测试终端的应用安装引擎安装所述测试用安装包并验证开发者证书。其中, 应用安装引擎通过授权许可文件中的签名证书判断授权许可文件是由开发者签名, 还是由其它功能实体签名。签名证书中包含证书的类型。如果签名证书不由开发者签名, 则不认为是测试用安装。测试用授权许可文件中的签名证书与测试用安装包签名文件中的签名证书应该相同。在签名证书中, 包含证书签发机构的信息。应用安装引擎通过所述签发机构验证所述开发者证书。具体可以通过验证开发者证书中的签名来确认所述授权许可文件的合法性及有效性。

[0143] 如图 4B 所示, 上述步骤 310 可以包括:

[0144] 步骤 311, 生成应用程序的摘要。

[0145] 具体可以把整个或部分应用程序的内容作为输入源, 按约定的摘要算法, 生成应用程序摘要。

[0146] 步骤 312, 生成所述应用程序相应能力文件的摘要。

[0147] 具体可以把整个或部分能力文件的内容作为输入源, 按约定的摘要算法, 生成能力文件摘要。

[0148] 步骤 313, 计算所述应用程序的摘要与所述能力文件的摘要的签名信息。

[0149] 具体可以把应用程序的摘要与能力文件的摘要串连作为输入源, 按约定的摘要算法, 生成签名文件摘要, 然后, 按约定的加密算法, 使用开发者的私钥加密签名文件摘要, 并把加密后的摘要作为新的签名文件摘要。在整个应用可控分发体系中, 摘要算法一致, 可以使用 SHA-1 算法。在整个应用可控分发体系中, 加密算法一致, 可以使用 ECC 算法。

- [0150] 步骤 314,根据所述签名信息生成签名文件。
- [0151] 如图 4C 所示,上述步骤 330 可以包括:
- [0152] 步骤 331,根据所述测试用安装包生成购买信息。
- [0153] 其中,该购买信息也可称为使用信息。
- [0154] 步骤 332,按约定的摘要算法及摘要加密算法,生成加密后的所述购买信息的摘要。
- [0155] 具体地,根据购买信息的部分或全部内容作为输入源,按约定的摘要算法,生成购买信息摘要,然后按约定的摘要加密算法,使用开发者的私钥加密购买信息摘要,并把加密后的购买信息摘要作为购买信息的摘要。
- [0156] 步骤 333,根据所述购买信息的摘要按约定的规则生成测试用授权许可文件。
- [0157] 如图 4D 所示,上述步骤 350 可以包括:
- [0158] 步骤 351,从所述测试用安装包中分离出应用软件、能力文件和签名文件,并找到相应的安装包标识。
- [0159] 其中,检查测试用安装包以及分离测试用安装包的内容,与前述签名应用软件,以及打包应用软件的过程相同,但顺序相反。所述安装包标识由应用开发终端的应用编程工具生成,可以采用全局用户标识 (GUI) 方式生成,以保证安装包标识的唯一性。
- [0160] 步骤 352,根据所述安装包标识查找相应的测试用授权许可文件。
- [0161] 其中,测试用安装包和测试用授权许可文件可以放在同一个目录,并使用相同的名字,但具有不同的扩展名。应用安装引擎在查找测试用授权许可文件时,可以直接在测试用安装包所在的目录查找相同名字的测试用授权许可文件。
- [0162] 步骤 353,对所述测试用授权许可文件进行检查,当该测试用授权许可文件中的签名证书与所述签名文件中的签名证书相同且均为开发者证书时,则继续执行步骤 354;否则执行步骤 357。
- [0163] 具体地,可以检查该测试用授权许可文件是否完整、一致和有效,以及是否合法。其中,检查该测试用授权许可文件的方法,与签名该测试用授权许可文件的过程相同,但执行顺序相反。
- [0164] 步骤 354,从所述开发者证书中分离出第一硬件标识串,并获取所述开发者测试终端的第二硬件标识串。
- [0165] 其中,具体获取步骤与图 3 所示过程相似,此处不再赘述。
- [0166] 步骤 355,判断所述第一硬件标识串与所述第二硬件标识串是否匹配,当不匹配时,则表明该测试用安装包不能在该开发者测试终端中安装,导致安装失败,执行步骤 357;否则继续执行步骤 356。
- [0167] 步骤 356,安装所述应用软件及能力文件。
- [0168] 其中,安装应用软件的具体方法此处不做限定。安装能力文件时,要将能力文件或其容复制到约定的地方。
- [0169] 步骤 357,显示安装结果。
- [0170] 如果安装成功,则显示测试成功的安装结果;如果安装失败,则显示测试失败的安装结果。
- [0171] 图 6 为本发明所述终端应用软件分发系统实施例的结构示意图,该系统能够实现

上述各方法实施例所述的方法。如图所示,该系统至少包括:应用开发终端 10、开发者测试终端 20 及开发者社区服务器 30,其工作原理如下:

[0172] 开发者使用通过所述应用开发终端 10 将开发者注册成为开发者社区的用户,并开发应用软件;通过本步骤,可以得到开发者证书和密钥,具体地,开发者可以使用应用开发终端 10 的应用编程工具编辑、编译、链接和测试应用软件,并使用应用开发终端编辑和测试能力文件。

[0173] 开发者通过开发者测试终端 20 测试所述应用软件。具体地,开发者在开发者测试终端上测试应用软件之前,或者向开发者社区提交应用软件之前,可以先对应用软件打包生成测试用安装包,然后把测试用安装包推送到开发者测试终端或者开发者社区。该测试用安装包中包含应用软件、能力文件和签名文件。

[0174] 此后,所述开发者还通过应用开发终端 10 对通过所述测试的应用软件进行打包并提交给所述开发者社区服务器 30,由该开发者社区服务器 30 保存由应用开发终端 10 提交的所述应用软件以供用户终端下载。从而实现终端应用软件的分发。

[0175] 另外,如图 6 所示,所述系统还可以进一步包括签名服务系统 40;如图 7 所示,所述应用开发终端 10 可以具体包括:加密模块 11、证书申请模块 12、存储模块 13 及结果反馈模块 14;如图 8 所示,所述开发者社区服务器 30 包括:证书生成模块 31 及证书反馈模块 32,其注册过程的工作原理说明如下:

[0176] 当所述应用开发终端 10 接收到来自于开发者的注册申请后,应用开发终端 10 的加密模块 11 根据约定的加密算法生成证书申请信息,有关证书申请信息的说明可参见上述步骤 102 的相关说明,此处不再赘述;证书申请模块 12 向所述开发者社区服务器 30 发送包含所述证书申请信息的证书申请;所述开发者社区服务器 30 中的证书生成模块 31 根据所述证书申请信息及与签名服务系统 40 的约定生成开发者证书,通过所述签名服务系统 40 对证书生成模块 31 的所述开发者证书进行签名后,由证书反馈模块 32 将所述签名服务系统 40 签名后的开发者证书反馈给所述应用开发终端 10。

[0177] 此后,应用开发终端 10 的存储模块 13 存储来自于所述开发者社区服务器的开发者证书,并通过结果反馈模块 14 向所述开发者社区服务器 30 反馈注册申请处理结果,从而完成注册过程。

[0178] 如图 9 所示,所述应用开发终端 10 可以具体包括:签名文件生成模块 15、安装包生成模块 16、许可文件生成模块 17 及传输模块 18;如图 10 所示,所述开发者测试终端 20 包括:分离模块 21、查找模块 22、检查模块 23、标识串处理模块 24、判断模块 25 和安装模块 26。对应用软件的测试过程说明如下:

[0179] 应用开发终端 10 中的签名文件生成模块 15 根据所述应用软件生成签名文件。具体地,可以通过应用开发终端的签名打包工具根据约定的规则,以及应用软件、能力文件、开发者证书、开发者私钥等信息,生成签名文件。有关签名文件的内容可参见上述步骤 310 的相关说明,此处不再赘述。

[0180] 安装包生成模块 16 根据签名文件生成模块 15 生成的所述签名文件生成测试用安装包。有关该测试用安装包可以参考上述步骤 320 的相关说明,此处不再赘述。许可文件生成模块 17 根据安装包生成模块生成的所述测试用安装包生成测试用授权许可文件。有关该测试用授权许可文件可以参考上述步骤 330 的相关说明,此处不再赘述。

[0181] 传输模块 18 将安装包生成模块 16 生成的所述测试用安装包及许可文件生成模块 17 生成的所述测试用授权许可文件传输给所述开发者测试终端 20。

[0182] 此后,该开发者测试终端 20 中的分离模块 21 从所述测试用安装包中分离出应用软件、能力文件和签名文件,并找到相应的安装包标识,其中,检查测试用安装包以及分离测试用安装包的内容,与前述签名应用软件,以及打包应用软件的过程相同,但顺序相反。所述安装包标识由应用开发终端的应用编程工具生成,可以采用全局用户标识 (GUI) 方式生成,以保证安装包标识的唯一性。

[0183] 查找模块 22 根据分离模块 21 分离出的所述安装包标识查找相应的测试用授权许可文件。其中,测试用安装包和测试用授权许可文件可以放在同一个目录,并使用相同的名字,但具有不同的扩展名。应用安装引擎在查找测试用授权许可文件时,可以直接在测试用安装包所在的目录查找相同名字的测试用授权许可文件。

[0184] 检查模块 23 对所述测试用授权许可文件进行检查;当检查模块 23 检查出所述测试用授权许可文件中的签名证书与所述签名文件中的签名证书相同且均为开发者证书时,由标识串处理模块 24 从所述开发者证书中分离出第一硬件标识串,并获取所述开发者测试终端的第二硬件标识串。具体地,可以检查该测试用授权许可文件是否完整、一致和有效,以及是否合法。其中,检查该测试用授权许可文件的方法,与签名该测试用授权许可文件的过程相同,但执行顺序相反。

[0185] 判断模块 25 判断所述第一硬件标识串与所述第二硬件标识串是否匹配,当判断模块 25 判断出所述第一硬件标识串与所述第二硬件标识串匹配时,由安装模块 26 在开发者测试终端 20 上安装所述应用软件及能力文件,以便进行测试。

[0186] 本实施例所述终端应用软件的分发系统无需限定于特定的终端产品,具有较高的通用性及安全性,可以由运营商搭建可管理、可运营、安全的应用软件可控分发体系;并且,该方法从开发者开发和测试应用软件的阶段便可以控制应用软件的分发,因此具有较高的可控性。

[0187] 本领域普通技术人员可以理解:实现上述方法实施例的全部或部分步骤可以通过程序指令相关的硬件来完成,前述的程序可以存储于一计算机可读取存储介质中,该程序在执行时,执行包括上述方法实施例的步骤;而前述的存储介质包括:ROM、RAM、磁碟或者光盘等各种可以存储程序代码的介质。

[0188] 最后应说明的是:以上实施例仅用以说明本发明的技术方案,而非对其限制;尽管参照前述实施例对本发明进行了详细的说明,本领域的普通技术人员应当理解:其依然可以对前述各实施例所记载的技术方案进行修改,或者对其中部分技术特征进行等同替换;而这些修改或者替换,并不使相应技术方案的本质脱离本发明各实施例技术方案的精神和范围。

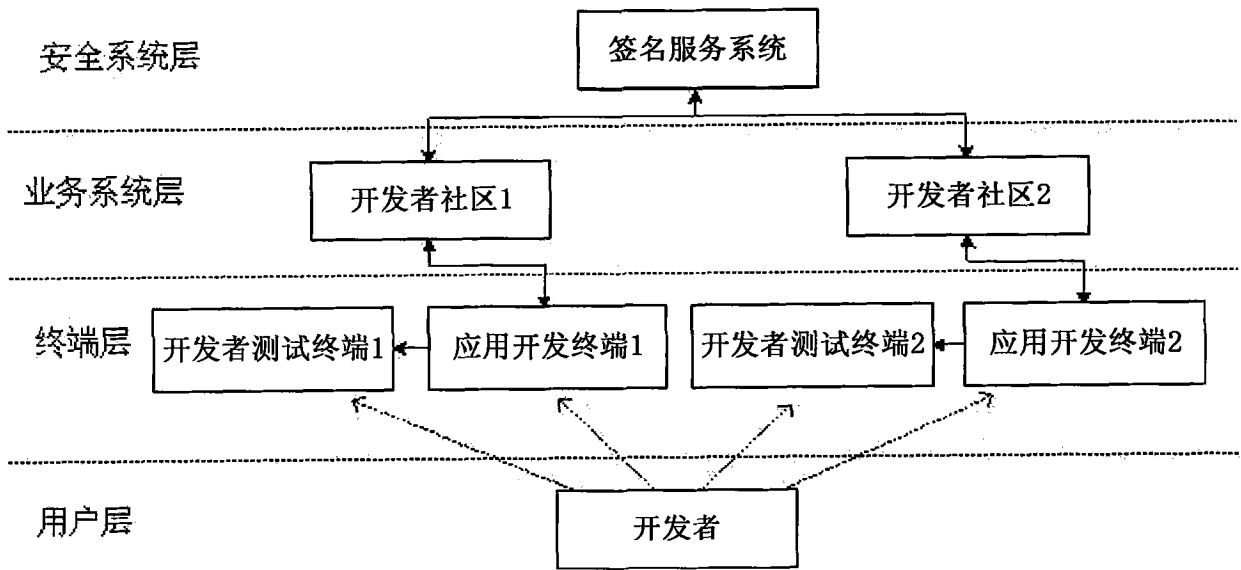


图 1

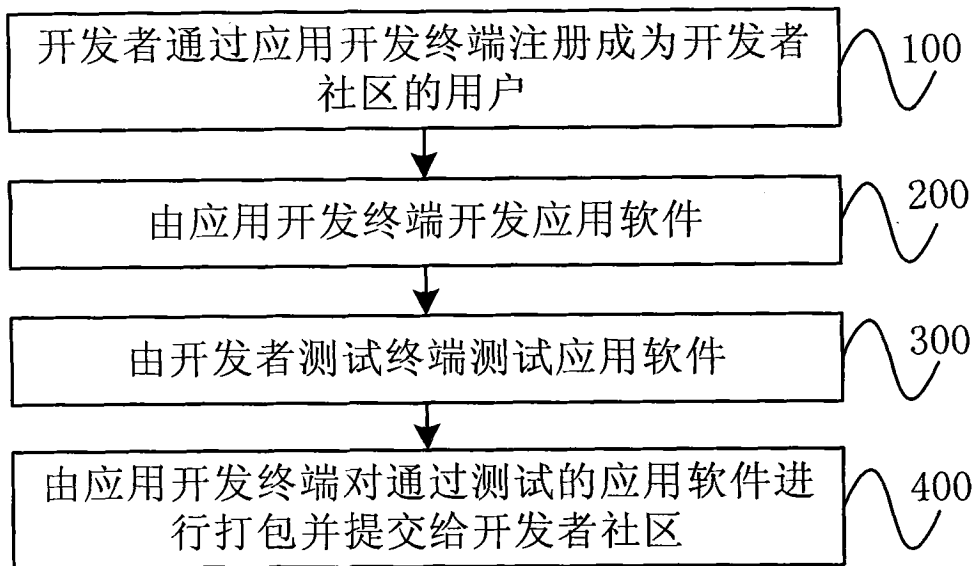


图 2

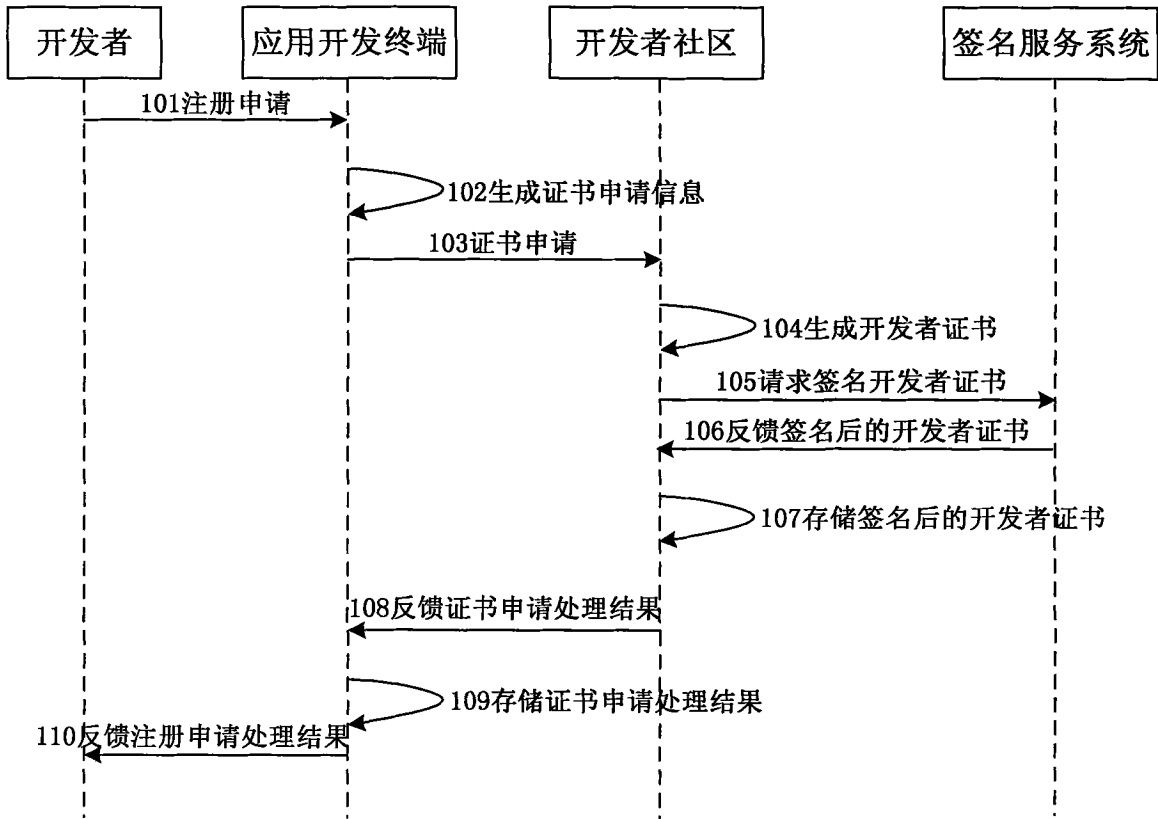


图 3

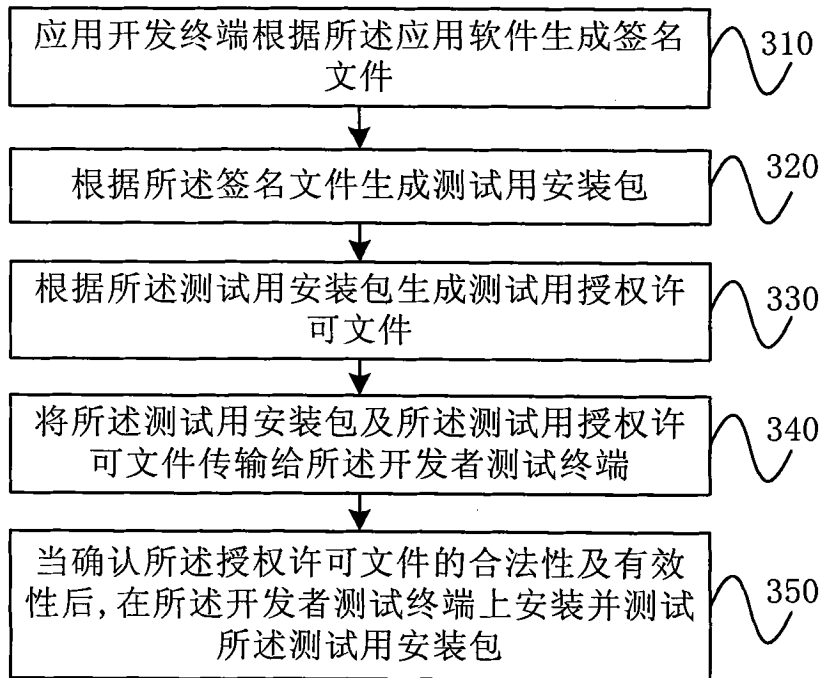


图 4A



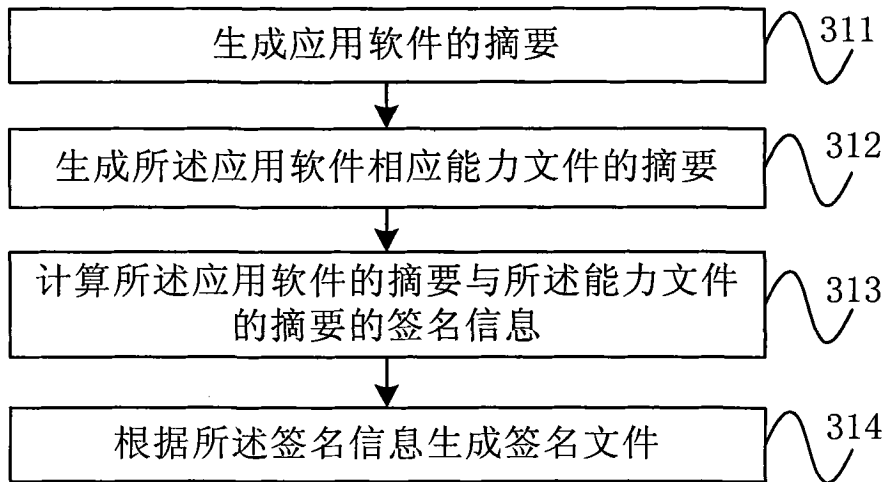


图 4B

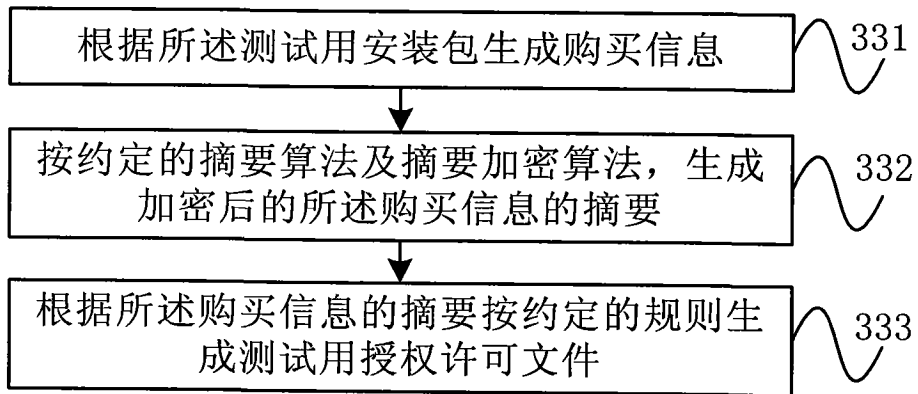


图 4C

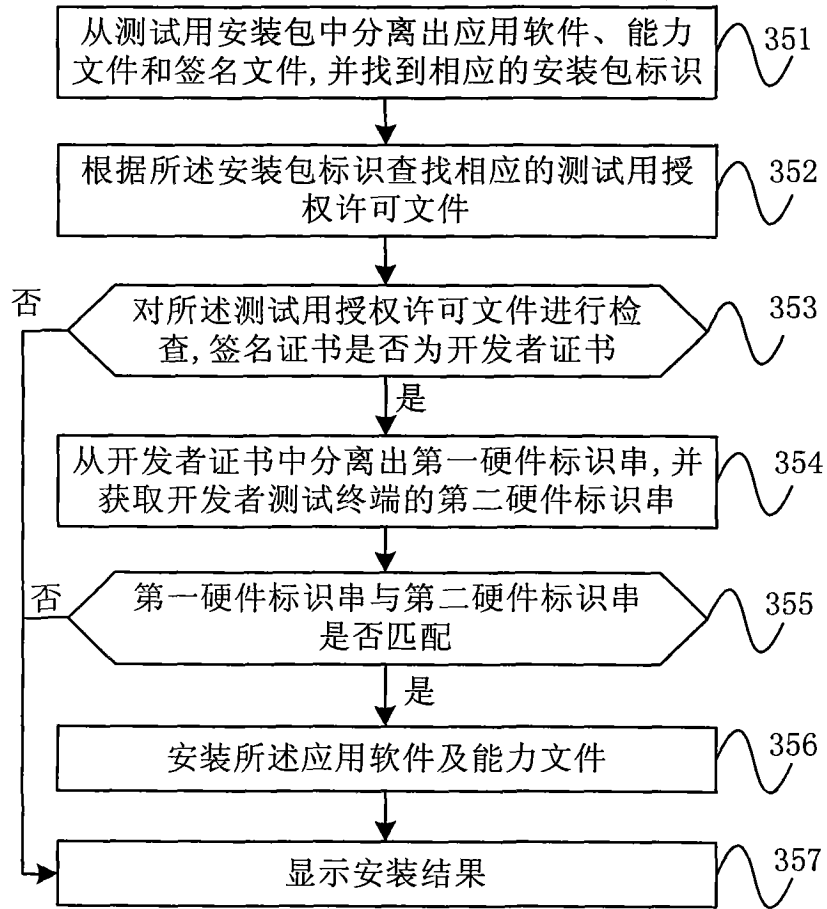


图 4D

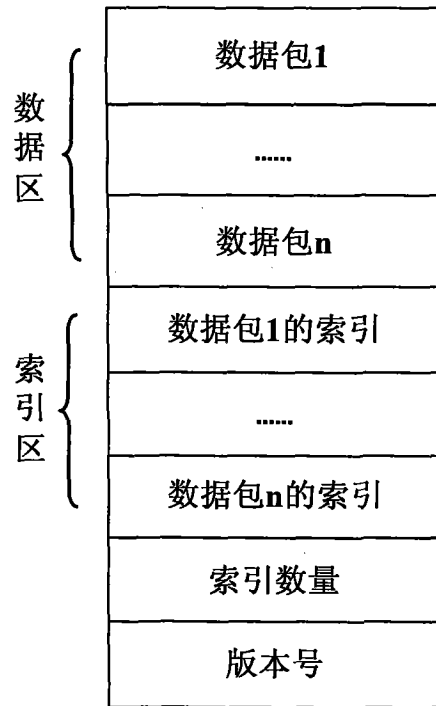


图 5A

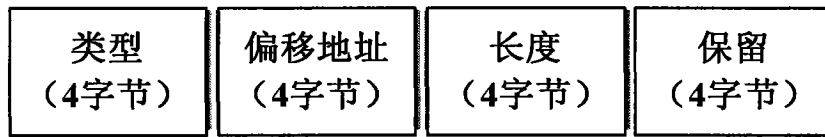


图 5B

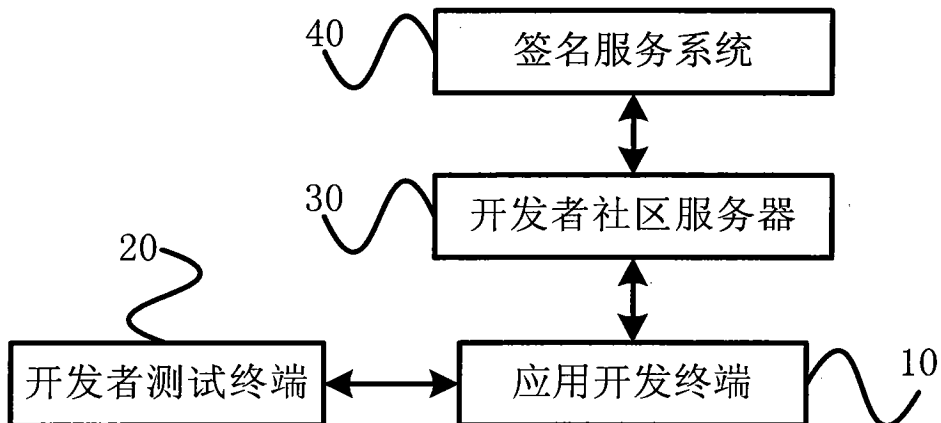


图 6

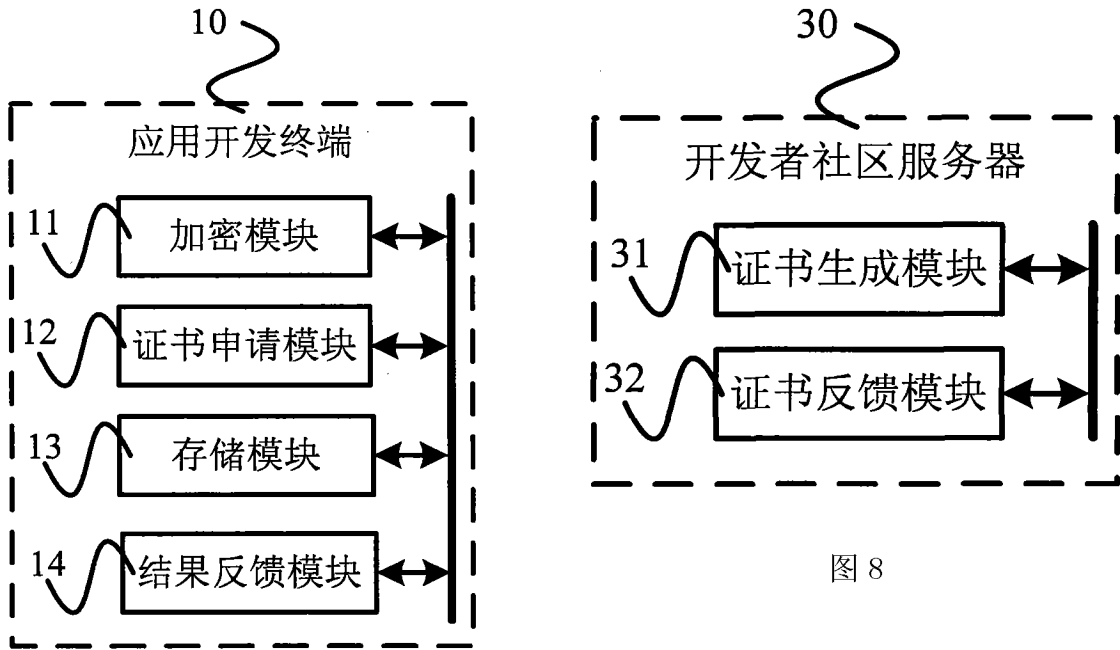


图 8

图 7

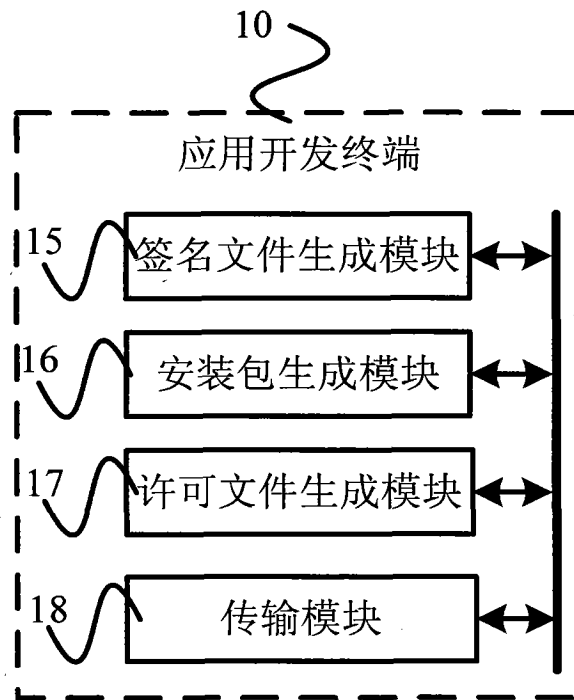


图 9

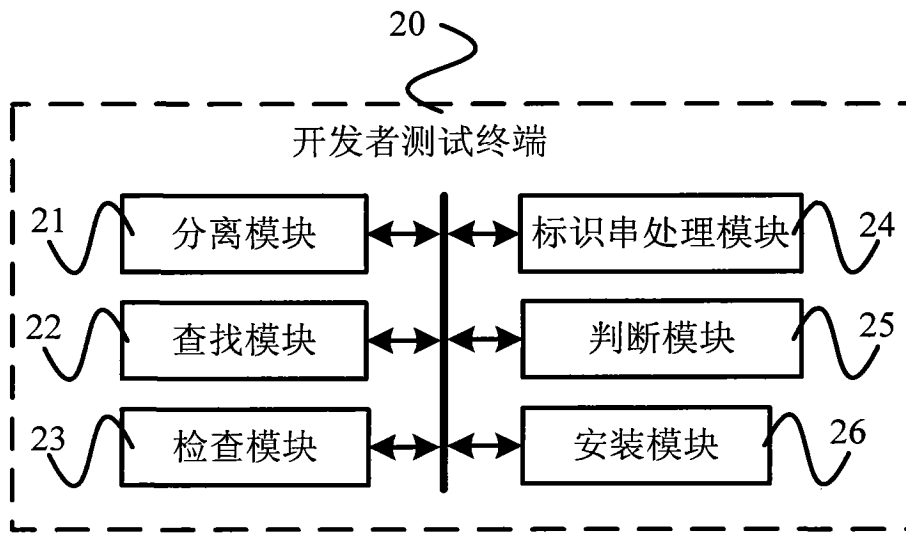


图 10