(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 107689934 B
(45)授权公告日 2020.12.04

(21)申请号 201610626636.6

(22)申请日 2016.08.03

(65)同一申请的已公布的文献号
申请公布号 CN 107689934 A

(43)申请公布日 2018.02.13

(73)专利权人 腾讯科技(深圳)有限公司
地址 518000 广东省深圳市福田区振兴路赛格科技园2栋东403室

(72)发明人 郭浩然

(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 蒋雅洁 张颖玲

(51)Int.Cl.
H04L 29/06(2006.01)

(56)对比文件
CN 103718531 A,2014.04.09
US 9292416 B2,2016.03.22

审查员 董莉

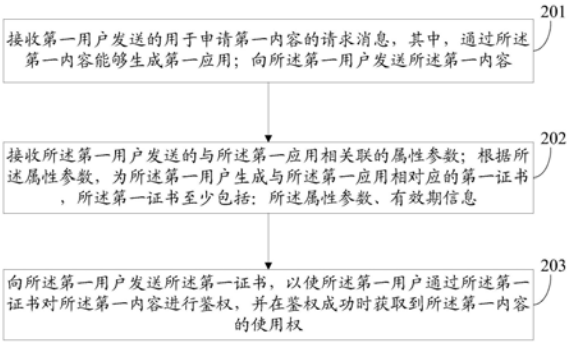
权利要求书3页 说明书13页 附图5页

(54)发明名称

一种保障信息安全的方法、服务器及客户端

(57)摘要

本发明公开了一种保障信息安全的方法、服务器及客户端，包括：接收第一用户发送的用于申请第一内容的请求消息，其中，通过所述第一内容能够生成第一应用；向所述第一用户发送所述第一内容；接收所述第一用户发送的与所述第一应用相关联的属性参数；根据所述属性参数，为所述第一用户生成与所述第一应用相对应的第一证书，所述第一证书至少包括：所述属性参数、有效期信息；向所述第一用户发送所述第一证书，以使所述第一用户通过所述第一证书对所述第一内容进行鉴权，并在鉴权成功时获取到所述第一内容的使用权。



CN 107689934 B

1.一种保障信息安全的方法,其特征在于,所述方法包括:

接收软件开发者发送的用于申请软件开发套件的请求消息,其中,通过所述软件开发套件能够生成第一应用;

向所述软件开发者发送所述软件开发套件;

接收所述软件开发者发送的与所述第一应用相关联的属性参数;

根据所述属性参数,为所述软件开发者生成与所述第一应用相对应的第一证书,所述第一证书至少包括:所述属性参数、有效期信息、各个功能模块对应的开关状态及签名指纹;

其中,所述软件开发者需要使用的功能模块对应开状态,所述软件开发者不需要使用的功能模块对应关状态;

向所述软件开发者发送所述第一证书,以使所述软件开发者通过所述第一证书对所述软件开发套件进行鉴权,并在鉴权成功时获取到所述软件开发套件中处于开状态的功能模块的使用权。

2.根据权利要求1所述的保障信息安全的方法,其特征在于,所述方法还包括:

对所述第一证书中的一个或多个内容进行更新,并向所述软件开发套件发送更新后的所述第一证书。

3.根据权利要求1所述的保障信息安全的方法,其特征在于,所述根据所述属性参数,为所述软件开发者生成与所述第一应用相对应的第一证书,包括:

根据所述属性参数,通过非对称加密算法为所述软件开发者生成与所述软件开发套件相对应的第一证书。

4.根据权利要求1至3任一项所述的保障信息安全的方法,其特征在于,所述第一证书还包括:各个功能模块对应的配置参数;

所述接收软件开发者发送的用于申请软件开发套件的请求消息后,所述方法还包括:

根据所述用于申请软件开发套件的请求消息,确定出处于开状态的各个功能模块;

将所述处于开状态的各个功能模块所对应的工具包添加至软件开发套件中。

5.一种保障信息安全的方法,其特征在于,所述方法包括:

向服务器发送用于申请软件开发套件的请求消息,其中,通过所述软件开发套件能够生成第一应用;

接收所述服务器发送的所述软件开发套件;

向所述服务器发送与所述第一应用相关联的属性参数;

接收所述服务器根据所述属性参数生成的与所述第一应用相对应的第一证书,所述第一证书至少包括:所述属性参数、有效期信息、各个功能模块对应的开关状态及签名指纹;其中,所述开状态对应软件开发者需要使用的功能模块,所述关状态对应所述软件开发者不需要使用的功能模块;

通过所述第一证书对所述软件开发套件进行鉴权,并在鉴权成功时获取到所述软件开发套件中处于开状态的功能模块的使用权。

6.根据权利要求5所述的保障信息安全的方法,其特征在于,所述接收所述服务器根据所述属性参数生成的与所述第一应用相对应的第一证书后,所述方法还包括:

将所述第一证书存储至目标目录下;

所述通过所述第一证书对所述软件开发套件进行鉴权，包括：

调用所述目标目录下的第一证书对所述软件开发套件进行鉴权。

7.根据权利要求6所述的保障信息安全的方法，其特征在于，与所述第一应用相对应的第一证书为一个或多个；

所述将所述第一证书存储至目标目录下，包括：

对所述一个或多个第一证书进行编号后，存储至目标目录下。

8.根据权利要求5所述的保障信息安全的方法，其特征在于，所述第一证书还包括：各个功能模块对应的配置参数；所述软件开发套件至少包括一个以上功能模块对应的工具包；所述方法还包括：

通过所述第一证书对所述各个功能模块进行鉴权；

当更新或者增加功能模块时，通过所述第一证书对所述更新或者增加功能模块进行鉴权。

9.一种服务器，其特征在于，所述服务器包括：

第一接收单元，用于接收软件开发者发送的用于申请软件开发套件的请求消息，其中，通过所述软件开发套件能够生成第一应用；

第一发送单元，用于向所述软件开发者发送所述软件开发套件；

第二接收单元，用于接收所述软件开发者发送的与所述第一应用相关联的属性参数；

生成单元，用于根据所述属性参数，为所述软件开发者生成与所述第一应用相对应的第一证书，所述第一证书至少包括：所述属性参数、有效期信息、各个功能模块对应的开关状态及签名指纹；

其中，所述软件开发者需要使用的功能模块对应开状态，所述软件开发者不需要使用的功能模块对应关状态；

第二发送单元，用于向所述软件开发者发送所述第一证书，以使所述软件开发者通过所述第一证书对所述软件开发套件进行鉴权，并在鉴权成功时获取到所述软件开发套件中处于开状态的功能模块的使用权。

10.根据权利要求9所述的服务器，其特征在于，所述服务器还包括：

更新单元，用于对所述第一证书中的一个或多个内容进行更新，并向所述软件开发者发送更新后的所述第一证书。

11.根据权利要求9所述的服务器，其特征在于，所述生成单元，还用于根据所述属性参数，通过非对称加密算法为所述软件开发者生成与所述第一应用相对应的第一证书。

12.根据权利要求9至11任一项所述的服务器，其特征在于，所述第一证书还包括：各个功能模块对应的配置参数；

所述服务器还包括：确定单元，用于根据所述用于申请软件开发套件的请求消息，确定出处于开状态的各个功能模块；

打包单元，用于将所述处于开状态的各个功能模块所对应的工具包添加至软件开发套件中。

13.一种客户端，其特征在于，所述客户端包括：

第一发送单元，用于向服务器发送用于申请软件开发套件的请求消息，其中，通过所述软件开发套件能够生成第一应用；

第一接收单元，用于接收所述服务器发送的所述软件开发套件；

第二发送单元，用于向所述服务器发送与所述第一应用相关联的属性参数；

第二接收单元，用于接收所述服务器根据所述属性参数生成的与所述第一应用相对应的第一证书，所述第一证书至少包括：所述属性参数、有效期信息、各个功能模块对应的开关状态及签名指纹；

其中，所述开状态对应软件开发者需要使用的功能模块，所述关状态对应所述软件开发者不需要使用的功能模块；

鉴权单元，用于通过所述第一证书对所述软件开发套件进行鉴权，并在鉴权成功时获取到所述软件开发套件中处于开状态的功能模块的使用权。

14.根据权利要求13所述的客户端，其特征在于，所述客户端还包括：

存储单元，用于将所述第一证书存储至目标目录下；

所述鉴权单元，还用于调用所述目标目录下的第一证书对所述软件开发套件进行鉴权。

15.根据权利要求14所述的客户端，其特征在于，与所述第一应用相对应的第一证书为一个或多个；

所述存储单元，还用于对所述一个或多个第一证书进行编号后，存储至目标目录下。

16.根据权利要求13所述的客户端，其特征在于，所述第一证书还包括：各个功能模块对应的配置参数；所述软件开发套件至少包括一个以上功能模块对应的工具包；

所述鉴权单元，还用于通过所述第一证书对所述各个功能模块进行鉴权；当更新或者增加功能模块时，通过所述第一证书对所述更新或者增加功能模块进行鉴权。

一种保障信息安全的方法、服务器及客户端

技术领域

[0001] 本发明涉及信息安全技术，尤其涉及一种保障信息安全的方法、服务器及客户端。

背景技术

[0002] 软件开发工具包(SDK，Software Development Ki)是软件开发工具的集合，软件开发者利用SDK能够为特定的软件包、软件框架、硬件平台、操作系统等建立应用软件。SDK的安全性设计中，一个重要的部分是防止被滥用，尤其对于基于Java的SDK，一般向软件开发者交付的文件(如.jar文件)极易被反编译和修改，不安全的鉴权机制导致SDK被破解使得SDK被滥用。

[0003] 传统的SDK鉴权机制通过发放不同的密码给不同的软件开发者，软件开发者利用密码对SDK进行鉴权。这种鉴权机制，SDK的密码颁发相对简单，通常会根据应用(APP)的签名指纹和/或包名，采用对称加密算法制作相应的密码(key)，然后把key交付给软件开发者。这种密码包含的信息较为单一，密码固定不变，极易被破解且可控性极差。

发明内容

[0004] 为解决上述技术问题，本发明实施例提供了一种保障信息安全的方法、服务器及客户端。

[0005] 本发明实施例提供的保障信息安全的方法，包括：

[0006] 接收第一用户发送的用于申请第一内容的请求消息，其中，通过所述第一内容能够生成第一应用；

[0007] 向所述第一用户发送所述第一内容；

[0008] 接收所述第一用户发送的与所述第一应用相关联的属性参数；

[0009] 根据所述属性参数，为所述第一用户生成与所述第一应用相对应的第一证书，所述第一证书至少包括：所述属性参数、有效期信息；

[0010] 向所述第一用户发送所述第一证书，以使所述第一用户通过所述第一证书对所述第一内容进行鉴权，并在鉴权成功时获取到所述第一内容的使用权。

[0011] 本发明实施例中，所述方法还包括：

[0012] 对所述第一证书中的一个或多个内容进行更新，并向所述第一用户发送更新后的所述第一证书。

[0013] 本发明实施例中，所述根据所述属性参数，为所述第一用户生成与所述第一应用相对应的第一证书，包括：

[0014] 根据所述属性参数，通过非对称加密算法为所述第一用户生成与所述第一应用相对应的第一证书。

[0015] 本发明实施例中，所述第一证书还包括：各个功能模块对应的状态及配置参数；

[0016] 所述接收第一用户发送的用于申请第一内容的请求消息后，所述方法还包括：

[0017] 根据所述用于申请第一内容的请求消息，确定出处于第一状态的各个功能模块；

[0018] 将所述处于第一状态的各个功能模块所对应的工具包添加至第一内容中。

[0019] 本发明另一实施例提供的保障信息安全的方法，包括：

[0020] 向服务器发送用于申请第一内容的请求消息，其中，通过所述第一内容能够生成第一应用；

[0021] 接收所述服务器发送的所述第一内容；

[0022] 向所述服务器发送与所述第一应用相关联的属性参数；

[0023] 接收所述服务器根据所述属性参数生成的与所述第一应用相对应的第一证书，所述第一证书至少包括：所述属性参数、有效期信息；

[0024] 通过所述第一证书对所述第一内容进行鉴权，并在鉴权成功时获取到所述第一内容的使用权。

[0025] 本发明实施例中，所述接收所述服务器根据所述属性参数生成的与所述第一应用相对应的第一证书后，所述方法还包括：

[0026] 将所述第一证书存储至目标目录下；

[0027] 所述通过所述第一证书对所述第一内容进行鉴权，包括：

[0028] 调用所述目标目录下的第一证书对所述第一内容进行鉴权。

[0029] 本发明实施例中，与所述第一应用相对应的第一证书为一个或多个；

[0030] 所述将所述第一证书存储至目标目录下，包括：

[0031] 对所述一个或多个第一证书进行编号后，存储至目标目录下。

[0032] 本发明实施例中，所述第一证书还包括：各个功能模块对应的状态及配置参数；所述第一内容至少包括一个以上功能模块对应的工具包；所述方法还包括：

[0033] 通过所述第一证书对所述各个功能模块进行鉴权；

[0034] 当更新或者增加功能模块时，通过所述第一证书对所述更新或者增加功能模块进行鉴权。

[0035] 本发明实施例提供的服务器，包括：

[0036] 第一接收单元，用于接收第一用户发送的用于申请第一内容的请求消息，其中，通过所述第一内容能够生成第一应用；

[0037] 第一发送单元，用于向所述第一用户发送所述第一内容；

[0038] 第二接收单元，用于接收所述第一用户发送的与所述第一应用相关联的属性参数；

[0039] 生成单元，用于根据所述属性参数，为所述第一用户生成与所述第一应用相对应的第一证书，所述第一证书至少包括：所述属性参数、有效期信息；

[0040] 第二发送单元，用于向所述第一用户发送所述第一证书，以使所述第一用户通过所述第一证书对所述第一内容进行鉴权，并在鉴权成功时获取到所述第一内容的使用权。

[0041] 本发明实施例中，所述服务器还包括：

[0042] 更新单元，用于对所述第一证书中的一个或多个内容进行更新，并向所述第一用户发送更新后的所述第一证书。

[0043] 本发明实施例中，所述生成单元，还用于根据所述属性参数，通过非对称加密算法为所述第一用户生成与所述第一应用相对应的第一证书。

[0044] 本发明实施例中，所述第一证书还包括：各个功能模块对应的状态及配置参数；

[0045] 所述服务器还包括:确定单元,用于根据所述用于申请第一内容的请求消息,确定出处于第一状态的各个功能模块;

[0046] 打包单元,用于将所述处于第一状态的各个功能模块所对应的工具包添加至第一内容中。

[0047] 本发明实施例提供的客户端,包括:

[0048] 第一发送单元,用于向服务器发送用于申请第一内容的请求消息,其中,通过所述第一内容能够生成第一应用;

[0049] 第一接收单元,用于接收所述服务器发送的所述第一内容;

[0050] 第二发送单元,用于向所述服务器发送与所述第一应用相关联的属性参数;

[0051] 第二接收单元,用于接收所述服务器根据所述属性参数生成的与所述第一应用相对应的第一证书,所述第一证书至少包括:所述属性参数、有效期信息;

[0052] 鉴权单元,用于通过所述第一证书对所述第一内容进行鉴权,并在鉴权成功时获取到所述第一内容的使用权。

[0053] 本发明实施例中,所述客户端还包括:

[0054] 存储单元,用于将所述第一证书存储至目标目录下;

[0055] 所述鉴权单元,还用于调用所述目标目录下的第一证书对所述第一内容进行鉴权。

[0056] 本发明实施例中,与所述第一应用相对应的第一证书为一个或多个;

[0057] 所述存储单元,还用于对所述一个或多个第一证书进行编号后,存储至目标目录下。

[0058] 本发明实施例中,所述第一证书还包括:各个功能模块对应的状态及配置参数;所述第一内容至少包括一个以上功能模块对应的工具包;

[0059] 所述鉴权单元,还用于通过所述第一证书对所述各个功能模块进行鉴权;当更新或者增加功能模块时,通过所述第一证书对所述更新或者增加功能模块进行鉴权。

[0060] 本发明实施例的技术方案中,接收第一用户发送的用于申请第一内容的请求消息,其中,通过所述第一内容能够生成第一应用;向所述第一用户发送所述第一内容;接收所述第一用户发送的与所述第一应用相关联的属性参数;根据所述属性参数,为所述第一用户生成与所述第一应用相对应的第一证书,所述第一证书至少包括:所述属性参数、有效期信息;向所述第一用户发送所述第一证书,以使所述第一用户通过所述第一证书对所述第一内容进行鉴权,并在鉴权成功时获取到所述第一内容的使用权。可见,本发明实施例采用证书的方式替代传统的密码对第一内容(也即SDK)进行鉴权,证书包含的信息较为丰富,证书不是固定不变的,对证书的灵活配置确保了证书的安全性并且证书的可控性较高。

附图说明

[0061] 图1为本发明实施例中进行信息交互的各方硬件实体的示意图;

[0062] 图2为本发明实施例的保障信息安全的方法的流程示意图一;

[0063] 图3为本发明实施例的证书的结构示意图;

[0064] 图4为本发明实施例的保障信息安全的方法的流程示意图二;

[0065] 图5为本发明实施例的保障信息安全的方法的流程示意图三;

[0066] 图6为本发明实施例的保障信息安全的方法的流程示意图四；

[0067] 图7为本发明实施例的服务器的结构组成示意图；

[0068] 图8为本发明实施例的客户端的结构组成示意图。

具体实施方式

[0069] 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实施例。

[0070] 图1为本发明实施例中进行信息交互的各方硬件实体的示意图，图1中包括：客户端11、服务器12。其中，客户端11通过有线网络或者无线网络与服务器12进行信息交互。客户端11所指的设备包括手机、台式机、PC机、一体机等类型。客户端11的使用者为第一用户，本发明实施例所提到的第一用户均可指代客户端11。一个示例中，第一用户为软件开发者，软件开者需要向SDK开发商(对应的硬件为服务器12)请求获取自己所需要的SDK来开发软件。为了防止SDK被滥用，服务器12向第一用户提供SDK的同时，还为第一用户制作属于该用户的证书，第一用户通过证书对SDK进行鉴权后方可获得SDK的使用权。

[0071] 上述图1的例子只是实现本发明实施例的一个系统架构实例，本发明实施例并不限于上述图1所述的系统结构，基于该系统架构，提出本发明各个实施例。

[0072] 图2为本发明实施例的保障信息安全的方法的流程示意图一，如图2所示，所述保障信息安全的方法包括以下步骤：

[0073] 步骤201：接收第一用户发送的用于申请第一内容的请求消息，其中，通过所述第一内容能够生成第一应用；向所述第一用户发送所述第一内容。

[0074] 本发明实施例中，服务器接收第一用户发送的用于申请第一内容的请求消息。这里，服务器指代第一内容的开发商；第一用户指代软件开发者。

[0075] 本发明实施例中，第一内容是指SDK，SDK是软件开发工具的集合，SDK的种类多样，例如杀毒引擎SDK、Root SDK等等。第一用户利用SDK提供的内容能够开发生成相关的应用，例如利用杀毒引擎SDK开发生成杀毒应用。

[0076] 当第一用户想要申请第一内容时，向服务器发送用于申请第一内容的请求消息。服务器接收到第一用户发送的用于申请第一内容的请求消息时，向第一用户发送为所述第一用户制作的第一内容。

[0077] 这里，第一内容包括：.jar文件、文档文件、示例文件。其中，.jar文件中包括各个功能模块对应的工具包。文档文件记录有各种配置参数等信息。示例文件中列举了使用第一内容对应用进行开发的若干个范例。

[0078] 本发明实施例中，服务器为第一用户提供的第一内容可以包括SDK的全部功能模块，也可以只包括SDK的部分功能模块。针对用户个性化的需求，为用户订制符合用户需求的SDK。具体地，SDK所包括的各个功能模块具有两种状态，分别为第一状态和第二状态。其中，第一状态是指开状态，代表该功能模块需要打包至第一内容中。第二状态是指关状态，代表该功能模块不需要打包至第一内容中。服务器根据用于申请第一内容的请求消息，确定出处于第一状态的各个功能模块；将所述处于第一状态的各个功能模块所对应的工具包添加至第一内容中。可见，对于功能模块的开关状态，服务器会按开关状态打包SDK中功能模块对应的jar工具包，功能模块处于开状态才打包至SDK中，功能模块处于关状态则不打

包至SDK中。

[0079] 步骤202:接收所述第一用户发送的与所述第一应用相关联的属性参数;根据所述属性参数,为所述第一用户生成与所述第一应用相对应的第一证书,所述第一证书至少包括:所述属性参数、有效期信息。

[0080] 本发明实施例中,第一用户为第一应用向服务器申请对应的第一证书,具体地,第一用户向服务器发送与所述第一应用相关联的属性参数,服务器接收所述第一用户发送的与所述第一应用相关联的属性参数。这里,属性参数至少包括以下之一:签名指纹、包名。这个属性参数代表了第一应用的APP信息。服务器根据属性参数,为所述第一用户生成与所述第一应用相对应的第一证书。

[0081] 本发明实施例中,服务器根据属性参数,通过非对称加密算法为所述第一用户生成与所述第一应用相对应的第一证书。

[0082] 这里,服务器对第一证书进行制作和签名时,使用的是基于非对称加密签名算法(如RSA-SHA1算法)进行签名,以确保向第一用户下发的证书不可被伪造。这里,SHA1为安全哈希算法(Secure Hash Algorithm),用于对数据进行签名;RSA是公钥加密算法,用于对数据进行加密。

[0083] 本发明实施例中,第一证书至少包括:属性参数、有效期信息;除此之外,第一证书还可以包括:各个功能模块对应的状态及配置参数。这里,第一证书是对应于第一应用而言的证书,除此之外,如果为第二应用申请证书,则对应的证书称为第二证书,同一个应用可以申请一个或多个证书。例如,可以为第一应用申请用于进行软件测试的证书,也可以为第一应用申请用于进行产品发布的证书,不同的证书对应同一第一内容(如SDK)的权限有所不同,例如,A部门需要的功能模块为A+B+C模块,B部门需要的功能模块为E+F模块,A部门对应的证书中,A、B、C模块的状态为开状态;而B部门对应的证书中,E、F模块对应状态为开状态。一个应用对应多个证书,方便了不同部门之间协作开发应用。

[0084] 参照图3,图3为本发明实施例的证书的结构示意图,证书内容至少包括:包名、签名指纹、有效期信息;其中,包名和签名指纹统称为与第一应用相关联的属性参数,这个属性参数唯一代表了软件开发者的身份信息。除此之外,证书内容还包括:各个功能模块的开关状态以及配置参数等。包名、签名指纹、有效期信息、各个功能模块的开关状态以及配置参数统称为证书的内容数据,可以通过非对称加密算法对内容数据进行签名加密。

[0085] 步骤203:向所述第一用户发送所述第一证书,以使所述第一用户通过所述第一证书对所述第一内容进行鉴权,并在鉴权成功时获取到所述第一内容的使用权。

[0086] 本发明实施例中,服务器向所述第一用户发送所述第一证书。第一用户通过所述第一证书对所述第一内容进行鉴权,并在鉴权成功时获取到所述第一内容的使用权。

[0087] 这里,以第一证书采用RSA-SHA1算法进行签名加密为例,第一证书的鉴权过程为校验以下两个条件是否同时成立:

[0088] 条件1:证书所报签名指纹==系统所报APP签名指纹

[0089] 条件2:RSA_pub_decrypt(签名)==SHA1(内容数据)

[0090] 其中,RSA_pub_decrypt基于第一用户使用的本地公钥解密确定,SHA1基于SHA128散列算法确定。

[0091] 本发明实施例中,第一用户通过第一证书对所述第一内容进行鉴权的鉴权入口包

括但不限于:SDK入口、各个功能模块入口、各个功能模块动态下发逻辑入口。以SDK入口鉴权为例,当第一用户进入SDK(或者使用SDK)时,需要利用第一证书对SDK进行鉴权,鉴权成功时则成功进入SDK获取到该SDK的使用权。以某个功能模块入口鉴权为例,当第一用户进一步进入功能模块(或者使用功能模块)时,需要利用第一证书对功能模块进行鉴权,鉴权成功时则成功进入该功能模块获取到该功能模块的使用权。以功能模块动态下发逻辑入口鉴权为例,功能模块动态下发逻辑指的是在需要服务器下发执行的地方,下载的功能模块都自带鉴权,先鉴权后再执行本体,以防下载的功能模块被滥用。这里,功能模块动态下发逻辑包括但不限于:1)更新或新增功能模块,这里,旧版本的功能模块是静态编译的,不再进行鉴权;新更新的功能模块是动态加载的,需要鉴权再执行。2)按机型适配的子逻辑,比如特殊病毒查杀库、漏洞检测功能、提权功能等。

[0092] 本发明实施例的技术方案,通过颁发证书而非密码的形式作为鉴权凭据,使用非对称算法进行鉴权,并且鉴权时对SDK以及各个功能模块进行多重鉴权,保障了SDK不被滥用。

[0093] 图4为本发明实施例的保障信息安全的方法的流程示意图二,如图4所示,所述保障信息安全的方法包括以下步骤:

[0094] 步骤401:接收第一用户发送的用于申请第一内容的请求消息,其中,通过所述第一内容能够生成第一应用;向所述第一用户发送所述第一内容。

[0095] 本发明实施例中,服务器接收第一用户发送的用于申请第一内容的请求消息。这里,服务器指代第一内容的开发商;第一用户指代软件开发者。

[0096] 本发明实施例中,第一内容是指SDK,SDK是软件开发工具的集合,SDK的种类多样,例如杀毒引擎SDK、Root SDK等等。第一用户利用SDK提供的内容能够开发生成相关的应用,例如利用杀毒引擎SDK开发生成杀毒应用。

[0097] 当第一用户想要申请第一内容时,向服务器发送用于申请第一内容的请求消息。服务器接收到第一用户发送的用于申请第一内容的请求消息时,向第一用户发送为所述第一用户制作的第一内容。

[0098] 这里,第一内容包括:.jar文件、文档文件、示例文件。其中,.jar文件中包括各个功能模块对应的工具包。文档文件记录有各种配置参数等信息。示例文件中列举了使用第一内容对应用进行开发的若干个范例。

[0099] 本发明实施例中,服务器为第一用户提供的第一内容可以包括SDK的全部功能模块,也可以只包括SDK的部分功能模块。针对用户个性化的需求,为用户订制符合用户需求的SDK。具体地,SDK所包括的各个功能模块具有两种状态,分别为第一状态和第二状态。其中,第一状态是指开状态,代表该功能模块需要打包至第一内容中。第二状态是指关状态,代表该功能模块不需要打包至第一内容中。服务器根据用于申请第一内容的请求消息,确定出处于第一状态的各个功能模块;将所述处于第一状态的各个功能模块所对应的工具包添加至第一内容中。可见,对于功能模块的开关状态,服务器会按开关状态打包SDK中功能模块对应的jar工具包,功能模块处于开状态才打包至SDK中,功能模块处于关状态则不打包至SDK中。

[0100] 步骤402:接收所述第一用户发送的与所述第一应用相关联的属性参数;根据所述属性参数,为所述第一用户生成与所述第一应用相对应的第一证书,所述第一证书至少包

括:所述属性参数、有效期信息。

[0101] 本发明实施例中,第一用户为第一应用向服务器申请对应的第一证书,具体地,第一用户向服务器发送与所述第一应用相关联的属性参数,服务器接收所述第一用户发送的与所述第一应用相关联的属性参数。这里,属性参数至少包括以下之一:签名指纹、包名。这个属性参数代表了第一应用的APP信息。服务器根据属性参数,为所述第一用户生成与所述第一应用相对应的第一证书。

[0102] 本发明实施例中,服务器根据属性参数,通过非对称加密算法为所述第一用户生成与所述第一应用相对应的第一证书。

[0103] 这里,服务器对第一证书进行制作和签名时,使用的是基于非对称加密签名算法(如RSA-SHA1算法)进行签名,以确保向第一用户下发的证书不可被伪造。这里,SHA1为安全哈希算法(Secure Hash Algorithm),用于对数据进行签名;RSA是公钥加密算法,用于对数据进行加密。

[0104] 本发明实施例中,第一证书至少包括:属性参数、有效期信息;除此之外,第一证书还可以包括:各个功能模块对应的状态及配置参数。这里,第一证书是对应于第一应用而言的证书,除此之外,如果为第二应用申请证书,则对应的证书称为第二证书,同一个应用可以申请一个或多个证书。例如,可以为第一应用申请用于进行软件测试的证书,也可以为第一应用申请用于进行产品发布的证书,不同的证书对应同一第一内容(如SDK)的权限有所不同,例如,A部门需要的功能模块为A+B+C模块,B部门需要的功能模块为E+F模块,A部门对应的证书中,A、B、C模块的状态为开状态;而B部门对应的证书中,E、F模块对应状态为开状态。一个应用对应多个证书,方便了不同部门之间协作开发应用。

[0105] 参照图3,图3为本发明实施例的证书的结构示意图,证书内容至少包括:包名、签名指纹、有效期信息;其中,包名和签名指纹统称为与第一应用相关联的属性参数,这个属性参数唯一代表了软件开发者的身份信息。除此之外,证书内容还包括:各个功能模块的开关状态以及配置参数等。包名、签名指纹、有效期信息、各个功能模块的开关状态以及配置参数统称为证书的内容数据,可以通过非对称加密算法对内容数据进行签名加密。

[0106] 步骤403:向所述第一用户发送所述第一证书,以使所述第一用户通过所述第一证书对所述第一内容进行鉴权,并在鉴权成功时获取到所述第一内容的使用权。

[0107] 本发明实施例中,服务器向所述第一用户发送所述第一证书。第一用户通过所述第一证书对所述第一内容进行鉴权,并在鉴权成功时获取到所述第一内容的使用权。

[0108] 这里,以第一证书采用RSA-SHA1算法进行签名加密为例,第一证书的鉴权过程为校验以下两个条件是否同时成立:

[0109] 条件1:证书所报签名指纹＝＝系统所报APP签名指纹

[0110] 条件2:RSA_pub_decrypt(签名)＝＝SHA1(内容数据)

[0111] 其中,RSA_pub_decrypt基于第一用户使用的本地公钥解密确定,SHA1基于SHA128散列算法确定。

[0112] 本发明实施例中,第一用户通过第一证书对所述第一内容进行鉴权的鉴权入口包括但不限于:SDK入口、各个功能模块入口、各个功能模块动态下发逻辑入口。以SDK入口鉴权为例,当第一用户进入SDK(或者使用SDK)时,需要利用第一证书对SDK进行鉴权,鉴权成功时则成功进入SDK获取到该SDK的使用权。以某个功能模块入口鉴权为例,当第一用户进

一步进入功能模块(或者使用功能模块)时,需要利用第一证书对功能模块进行鉴权,鉴权成功时则成功进入该功能模块获取到该功能模块的使用权。以功能模块动态下发逻辑入口鉴权为例,功能模块动态下发逻辑指的是在需要服务器下发执行的地方,下载的功能模块都自带鉴权,先鉴权后再执行本体,以防下载的功能模块被滥用。这里,功能模块动态下发逻辑包括但不限于:1)更新或新增功能模块,这里,旧版本的功能模块是静态编译的,不再进行鉴权;新更新的功能模块是动态加载的,需要鉴权再执行。2)按机型适配的子逻辑,比如特殊病毒查杀库、漏洞检测功能、提权功能等。

[0113] 本发明实施例的技术方案,通过颁发证书而非密码的形式作为鉴权凭据,使用非对称算法进行鉴权,并且鉴权时对SDK以及各个功能模块进行多重鉴权,保障了SDK不被滥用。

[0114] 步骤404:对所述第一证书中的一个或多个内容进行更新,并向所述第一用户发送更新后的所述第一证书。

[0115] 本发明实施例中,服务器获取所述第一证书的有效期信息;服务器对所述有效期信息进行更新,并向所述第一用户发送更新后的所述第一证书。

[0116] 这里,第一证书具有有效期信息,软件开发使用者并不能够无限期的使用SDK,降低了SDK被破解的可能性。此外,如果用户想要续签使用SDK,则服务器可以对第一证书中的有效期信息进行更新,从而对第一证书进行灵活的控制。

[0117] 本发明实施例中,服务器除了能够对第一证书的有效期信息进行更新之外,还可以对第一证书中的其他内容进行灵活调整,例如调整各个功能模块的开关状态等。这样,第一用户使用SDK以及各个功能模块的权限达到了灵活调整。

[0118] 图5为本发明实施例的保障信息安全的方法的流程示意图三,如图5所示,所述保障信息安全的方法包括以下步骤:

[0119] 步骤501:向服务器发送用于申请第一内容的请求消息,其中,通过所述第一内容能够生成第一应用;接收所述服务器发送的所述第一内容。

[0120] 本发明实施例中,第一用户向服务器发送用于申请第一内容的请求消息。这里,第一用户指代软件开发者;服务器指代第一内容的开发商。

[0121] 本发明实施例中,第一内容是指SDK,SDK是软件开发工具的集合,SDK的种类多样,例如杀毒引擎SDK、Root SDK等等。第一用户利用SDK提供的内容能够开发生成相关的应用,例如利用杀毒引擎SDK开发生成杀毒应用。

[0122] 当第一用户想要申请第一内容时,向服务器发送用于申请第一内容的请求消息。而后,第一用户接收所述服务器发送的所述第一内容。

[0123] 这里,第一内容包括:.jar文件、文档文件、示例文件。其中,.jar文件中包括各个功能模块对应的工具包。文档文件记录有各种配置参数等信息。示例文件中列举了使用第一内容对应用进行开发的若干个范例。

[0124] 本发明实施例中,服务器为第一用户提供的第一内容可以包括SDK的全部功能模块,也可以只包括SDK的部分功能模块。针对用户个性化的需求,为用户订制符合用户需求的SDK。具体地,SDK所包括的各个功能模块具有两种状态,分别为第一状态和第二状态。其中,第一状态是指开状态,代表该功能模块需要打包至第一内容中。第二状态是指关状态,代表该功能模块不需要打包至第一内容中。服务器根据用于申请第一内容的请求消息,确

定出处于第一状态的各个功能模块;将所述处于第一状态的各个功能模块所对应的工具包添加至第一内容中。可见,对于功能模块的开关状态,服务器会按开关状态打包SDK中功能模块对应的jar工具包,功能模块处于开状态才打包至SDK中,功能模块处于关状态则不打包至SDK中。

[0125] 步骤502:向所述服务器发送与所述第一应用相关联的属性参数;接收所述服务器根据所述属性参数生成的与所述第一应用相对应的第一证书,所述第一证书至少包括:所述属性参数、有效期信息。

[0126] 本发明实施例中,属性参数至少包括以下之一:签名指纹、包名。这个属性参数代表了第一应用的APP信息。服务器根据属性参数,为所述第一用户生成与所述第一应用相对应的第一证书。

[0127] 本发明实施例中,服务器根据属性参数,通过非对称加密算法为所述第一用户生成与所述第一应用相对应的第一证书。

[0128] 本发明实施例中,第一证书至少包括:属性参数、有效期信息;除此之外,第一证书还可以包括:各个功能模块对应的状态及配置参数。这里,第一证书是对应于第一应用而言的证书,除此之外,如果为第二应用申请证书,则对应的证书称为第二证书,同一个应用可以申请一个或多个证书,可见,与所述第一应用相对应的第一证书为一个或多个。例如,可以为第一应用申请用于进行软件测试的证书,也可以为第一应用申请用于进行产品发布的证书,不同的证书对应同一第一内容(如SDK)的权限有所不同,例如,A部门需要的功能模块为A+B+C模块,B部门需要的功能模块为E+F模块,A部门对应的证书中,A、B、C模块的状态为开状态;而B部门对应的证书中,E、F模块对应状态为开状态。一个应用对应多个证书,方便了不同部门之间协作开发应用。

[0129] 参照图3,图3为本发明实施例的证书的结构示意图,证书内容至少包括:包名、签名指纹、有效期信息;其中,包名和签名指纹统称为与第一应用相关联的属性参数,这个属性参数唯一代表了软件开发者的身份信息。除此之外,证书内容还包括:各个功能模块的开关状态以及配置参数等。包名、签名指纹、有效期信息、各个功能模块的开关状态以及配置参数统称为证书的内容数据,可以通过非对称加密算法对内容数据进行签名加密。

[0130] 本发明实施例中,第一用户接收所述服务器根据所述属性参数生成的与所述第一应用相对应的第一证书后,将所述第一证书存储至目标目录下。

[0131] 这里,软件开发者获得证书后要求将证书放到指定的目录下,以Android的APK为例,指定的目录为assets目录,这样可提高APK的标识度:

[0132] 1)以手机管家SDK、KingRoot SDK为例,其证书分别为tmsdk.cert和krsdk.cert,若发现assets目录中有该文件名,软件开发者可获知与证书对应的SDK的存在,增加软件开发者使用SDK的可能性。

[0133] 2)如果通过SDK内部上报机制发现软件开发者的APP有与SDK相关的功能,但assets目录中证书文件不存在,则表明SDK很可能已经被破解。

[0134] 由于与所述第一应用相对应的第一证书为一个或多个,这里,对所述一个或多个第一证书进行编号后,存储至目标目录下。编号后,各个部门可以依据编号去获取对应的第一证书。

[0135] 步骤503:通过所述第一证书对所述第一内容进行鉴权,并在鉴权成功时获取到所

述第一内容的使用权。

[0136] 本发明实施例中，调用在指定目标目录下的第一证书对所述第一内容进行鉴权。

[0137] 这里，以第一证书采用RSA-SHA1算法进行签名加密为例，第一证书的鉴权过程为校验以下两个条件是否同时成立：

[0138] 条件1：证书所报签名指纹＝＝系统所报APP签名指纹

[0139] 条件2：RSA_pub_decrypt(签名)＝＝SHA1(内容数据)

[0140] 其中，RSA_pub_decrypt基于第一用户使用的本地公钥解密确定，SHA1基于SHA128散列算法确定。

[0141] 本发明实施例中，第一用户通过第一证书对所述第一内容进行鉴权，除此之外，还对所述各个功能模块进行鉴权；当更新或者增加功能模块时，通过所述第一证书对所述更新或者增加功能模块进行鉴权。

[0142] 具体地，第一用户通过第一证书对所述第一内容进行鉴权的鉴权入口包括但不限于：SDK入口、各个功能模块入口、各个功能模块动态下发逻辑入口。以SDK入口鉴权为例，当第一用户进入SDK(或者使用SDK)时，需要利用第一证书对SDK进行鉴权，鉴权成功时则成功进入SDK获取到该SDK的使用权。以某个功能模块入口鉴权为例，当第一用户进一步进入功能模块(或者使用功能模块)时，需要利用第一证书对功能模块进行鉴权，鉴权成功时则成功进入该功能模块获取到该功能模块的使用权。以功能模块动态下发逻辑入口鉴权为例，功能模块动态下发逻辑指的是在需要服务器下发执行的地方，下载的功能模块都自带鉴权，先鉴权后再执行本体，以防下载的功能模块被滥用。这里，功能模块动态下发逻辑包括但不限于：1)更新或新增功能模块，这里，旧版本的功能模块是静态编译的，不再进行鉴权；新更新的功能模块是动态加载的，需要鉴权再执行。2)按机型适配的子逻辑，比如特殊病毒查杀库、漏洞检测功能、提权功能等。

[0143] 本发明实施例的技术方案，通过颁发证书而非密码的形式作为鉴权凭据，使用非对称算法进行鉴权，并且鉴权时对SDK以及各个功能模块进行多重鉴权，保障了SDK不被滥用。SDK能够做到在无网络时可用、有网络时可控、且具有较高的安全性的要求。

[0144] 图6为本发明实施例的保障信息安全的方法的流程示意图四，如图6所示，所述保障信息安全的方法包括两个部分：

[0145] 第一部分：软件开发者通过SDK接口人申请记录软件开发者信息，并获取SDK开发包(包含.jar文件、文档文件、示例文件等)。

[0146] 第二部分：软件开发者给每个APP信息(包括指纹签名、包名)申请一个至多个证书。当申请多个证书时，软件开发者可以将多个证书按指定的编号方法进行编号后同时放到指定的目录中。

[0147] 如图6所示，所述流程具体包括如下步骤：

[0148] 步骤601：软件开发者向SDK接口人发送用于申请SDK开发包的请求消息。

[0149] 步骤602：SDK接口人对软件开发者的身份进行审核。

[0150] 步骤603：SDK接口人向SDK服务器发送软件开发者的信息。

[0151] 步骤604、605：SDK服务器经SDK接口人向软件开发者发送SDK开发包。

[0152] 步骤606：软件开发者向SDK接口人发送第一应用的属性参数，具体为APP1信息。

[0153] 步骤607：SDK接口人将APP1信息以及相关的控制参数1发送给SDK服务器。

[0154] 这里,控制参数1包括但限不限于各个功能模块对应开关状态以及配置参数等。

[0155] 步骤608:SDK服务器为APP1制作证书1。

[0156] 步骤609、610:SDK服务器将证书1经SDK接口人发送给软件开发者。

[0157] 步骤611-615:软件开发者向SDK服务器为APP2信息申请证书2;SDK服务器将制作的证书2发送给软件开发者。

[0158] 本发明实施例的技术方案提高了通过逆向工程破解SDK的门槛,这里,逆向工程是指反编译。具体地,

[0159] 1)SDK破解后仍然可有效鉴权

[0160] 由于SDK是.jar文件或其它可链接文件,较易破解或修改,甚至出现冒充系统接口导致签名判断不可靠。

[0161] 在上面所述的动态下发逻辑进行鉴权中,如果使用本地(native)层的方式实现,可以提高被破解的门槛,即使在.jar文件被破解或系统接口被冒充后仍然可以进行有效的鉴权。

[0162] 2)多次Java/native间跳转提高反编译难度

[0163] 为了提高破解SDK的难度,可以在鉴权流程上多次进行Java层与native层间的跳转,其中,native层还包括JNI接口和一般的native进程。以上面所述的“证书所报签名指纹==系统所报APP签名指纹”为例,可有如下的实现方式:

[0164] 证书的内容数据解密通常需要多个步骤,一般需要5步骤。其中,第1、3、5步骤使用native层来处理,第2、4步骤使用Java层来处理。同样,提取签名指纹字段也分多个步骤,可以按上述类似步骤进行处理。

[0165] 3)使用APK加固服务

[0166] 有较多的加固服务可以使已经生成.dex文件和.elf文件加固。针对SDK是交付.jar文件而非.dex文件的情况,仅对.elf文件(通过native层处理)进行加固。

[0167] 4)使用o-llvm在编译时进行加固

[0168] 将o-llvm混淆编译器移植到Android集成环境中,通过参数设置使之在编译期进行混淆。

[0169] 本发明实施例的证书具有可控性:一般来说,纯云端的SDK(如社交OpenAPI)具有极大的可控性,但由于实际软件开发者多为偏传统的合作者,要求SDK在没有联网时已经包括可运行的整个业务逻辑。使业务逻辑和数据格式较容易通过反编译等方式获得。本发明实施例加强这种情况下的可控性,包括对使用期限的控制、对各个功能模块开关的控制。这里,功能模块的开关除了第一次向软件开发者交付时具有的包含/不包含某功能的开关外,在功能模块动态更新后还指是否允许某个功能模块的功能。加入使用期限可在过期时通过联网更新证书内容,达到对证书可控的目的。一但证书更新,则使用更新的证书进行鉴权。这里,使用期限和功能模块开关的更新,极大地限制了SDK被滥用的情况。

[0170] 下面通过各种特殊场景说明SDK的安全性:

[0171] 1)软件开发者尝试通过删除动态下发的证书来防止模块被禁用

[0172] 因为证书有使用期限,SDK只要发现到达期限就会禁止各个功能模块的使用,直到证书被更新为止。

[0173] 2)软件开发者尝试通过修改系统时间来防止功能模块过期

[0174] 实际上经验证，修改时间会导致其它APP有问题，随着时间差距越大，越不可能修改时间。

[0175] 3)软件开发者尝试通过破解或冒充系统功能来绕过鉴权，扩大SDK使用范围

[0176] 动态下发的业务逻辑使这种方式失效。

[0177] 图7为本发明实施例的服务器的结构组成示意图，如图7所示，所述服务器包括：

[0178] 第一接收单元71，用于接收第一用户发送的用于申请第一内容的请求消息，其中，通过所述第一内容能够生成第一应用；

[0179] 第一发送单元72，用于向所述第一用户发送所述第一内容；

[0180] 第二接收单元73，用于接收所述第一用户发送的与所述第一应用相关联的属性参数；

[0181] 生成单元74，用于根据所述属性参数，为所述第一用户生成与所述第一应用相对应的第一证书，所述第一证书至少包括：所述属性参数、有效期信息；

[0182] 第二发送单元75，用于向所述第一用户发送所述第一证书，以使所述第一用户通过所述第一证书对所述第一内容进行鉴权，并在鉴权成功时获取到所述第一内容的使用权。

[0183] 所述服务器还包括：

[0184] 更新单元76，用于对所述第一证书中的一个或多个内容进行更新，并向所述第一用户发送更新后的所述第一证书。

[0185] 所述生成单元74，还用于根据所述属性参数，通过非对称加密算法为所述第一用户生成与所述第一应用相对应的第一证书。

[0186] 所述第一证书还包括：各个功能模块对应的状态及配置参数；

[0187] 所述服务器还包括：确定单元77，用于根据所述用于申请第一内容的请求消息，确定出处于第一状态的各个功能模块；

[0188] 打包单元78，用于将所述处于第一状态的各个功能模块所对应的工具包添加至第一内容中。

[0189] 本领域技术人员应当理解，图7所示的服务器中的各单元的实现功能可参照前述保障信息安全的方法的相关描述而理解。图7所示的服务器中的各单元的功能可通过运行于处理器上的程序而实现，也可通过具体的逻辑电路而实现。

[0190] 图8为本发明实施例的客户端的结构组成示意图，如图8所示，所述客户端包括：

[0191] 第一发送单元81，用于向服务器发送用于申请第一内容的请求消息，其中，通过所述第一内容能够生成第一应用；

[0192] 第一接收单元82，用于接收所述服务器发送的所述第一内容；

[0193] 第二发送单元83，用于向所述服务器发送与所述第一应用相关联的属性参数；

[0194] 第二接收单元84，用于接收所述服务器根据所述属性参数生成的与所述第一应用相对应的第一证书，所述第一证书至少包括：所述属性参数、有效期信息；

[0195] 鉴权单元85，用于通过所述第一证书对所述第一内容进行鉴权，并在鉴权成功时获取到所述第一内容的使用权。

[0196] 所述客户端还包括：

[0197] 存储单元86，用于将所述第一证书存储至目标目录下；

[0198] 所述鉴权单元85，还用于调用所述目标目录下的第一证书对所述第一内容进行鉴权。

[0199] 与所述第一应用相对应的第一证书为一个或多个；

[0200] 所述存储单元86，还用于对所述一个或多个第一证书进行编号后，存储至目标目录下。

[0201] 所述第一证书还包括：各个功能模块对应的状态及配置参数；所述第一内容至少包括一个以上功能模块对应的工具包；

[0202] 所述鉴权单元85，还用于通过所述第一证书对所述各个功能模块进行鉴权；当更新或者增加功能模块时，通过所述第一证书对所述更新或者增加功能模块进行鉴权。

[0203] 本领域技术人员应当理解，图8所示的客户端中的各单元的实现功能可参照前述保障信息安全的方法的相关描述而理解。图8所示的客户端中的各单元的功能可通过运行于处理器上的程序而实现，也可通过具体的逻辑电路而实现。

[0204] 本发明实施例所记载的技术方案之间，在不冲突的情况下，可以任意组合。

[0205] 在本发明所提供的几个实施例中，应该理解到，所揭露的方法和智能设备，可以通过其它的方式实现。以上所描述的设备实施例仅仅是示意性的，例如，所述单元的划分，仅仅为一种逻辑功能划分，实际实现时可以有另外的划分方式，如：多个单元或组件可以结合，或可以集成到另一个系统，或一些特征可以忽略，或不执行。另外，所显示或讨论的各组成部分相互之间的耦合、或直接耦合、或通信连接可以是通过一些接口，设备或单元的间接耦合或通信连接，可以是电性的、机械的或其它形式的。

[0206] 上述作为分离部件说明的单元可以是、或也可以不是物理上分开的，作为单元显示的部件可以是、或也可以不是物理单元，即可以位于一个地方，也可以分布到多个网络单元上；可以根据实际的需要选择其中的部分或全部单元来实现本实施例方案的目的。

[0207] 另外，在本发明各实施例中的各功能单元可以全部集成在一个第二处理单元中，也可以是各单元分别单独作为一个单元，也可以两个或两个以上单元集成在一个单元中；上述集成的单元既可以采用硬件的形式实现，也可以采用硬件加软件功能单元的形式实现。

[0208] 以上所述，仅为本发明的具体实施方式，但本发明的保护范围并不局限于此，任何熟悉本技术领域的技术人员在本发明揭露的技术范围内，可轻易想到变化或替换，都应涵盖在本发明的保护范围之内。

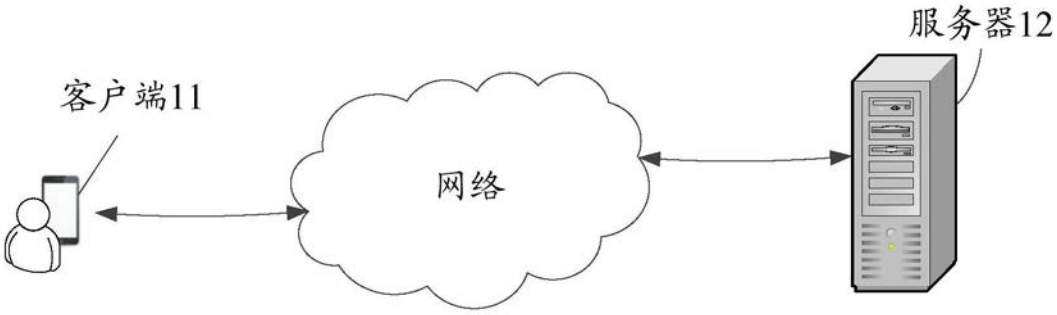

服务器12
客户端11
网络

图1

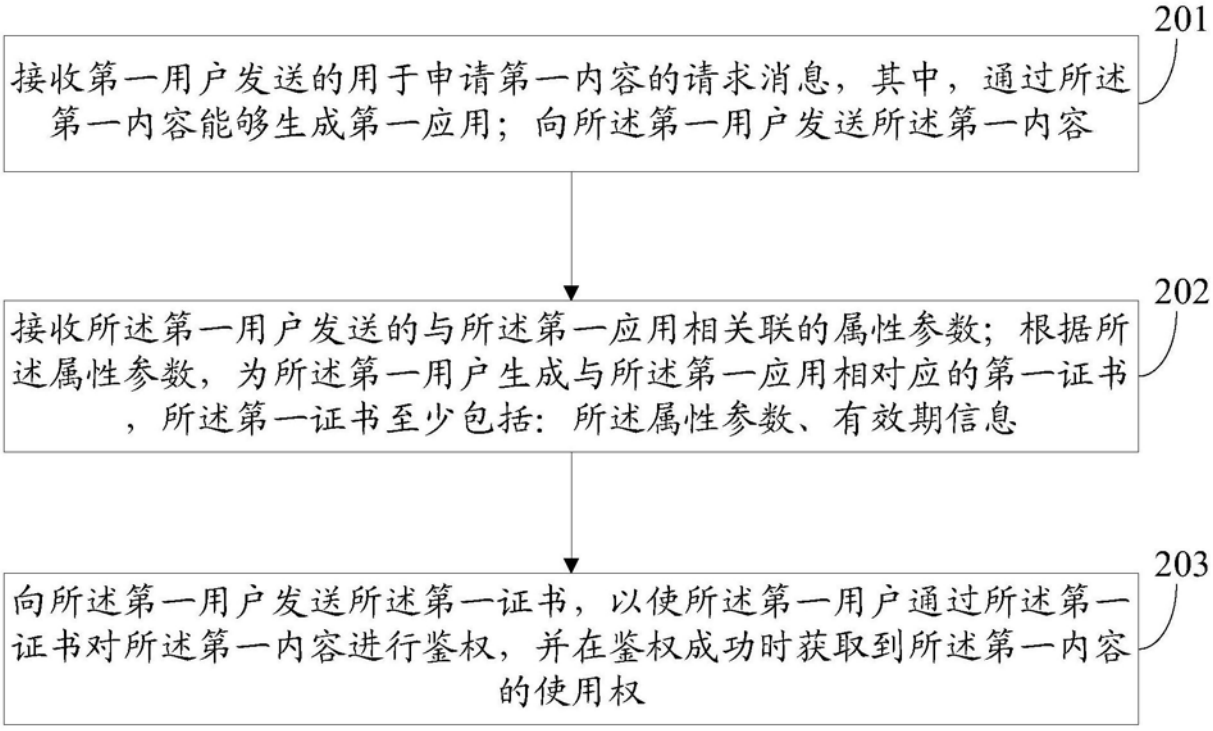

201
接收第一用户发送的用于申请第一内容的请求消息，其中，通过所述第一内容能够生成第一应用；向所述第一用户发送所述第一内容
202
接收所述第一用户发送的与所述第一应用相关联的属性参数；根据所述属性参数，为所述第一用户生成与所述第一应用相对应的第一证书，所述第一证书至少包括：所述属性参数、有效期信息
203
向所述第一用户发送所述第一证书，以使所述第一用户通过所述第一证书对所述第一内容进行鉴权，并在鉴权成功时获取到所述第一内容的使用权

图2

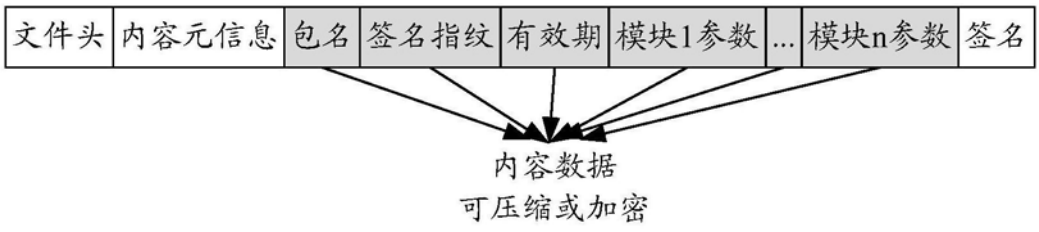

文件头
内容元信息
包名
签名指纹
有效期
模块1参数
...
模块n参数
签名
内容数据
可压缩或加密

图3

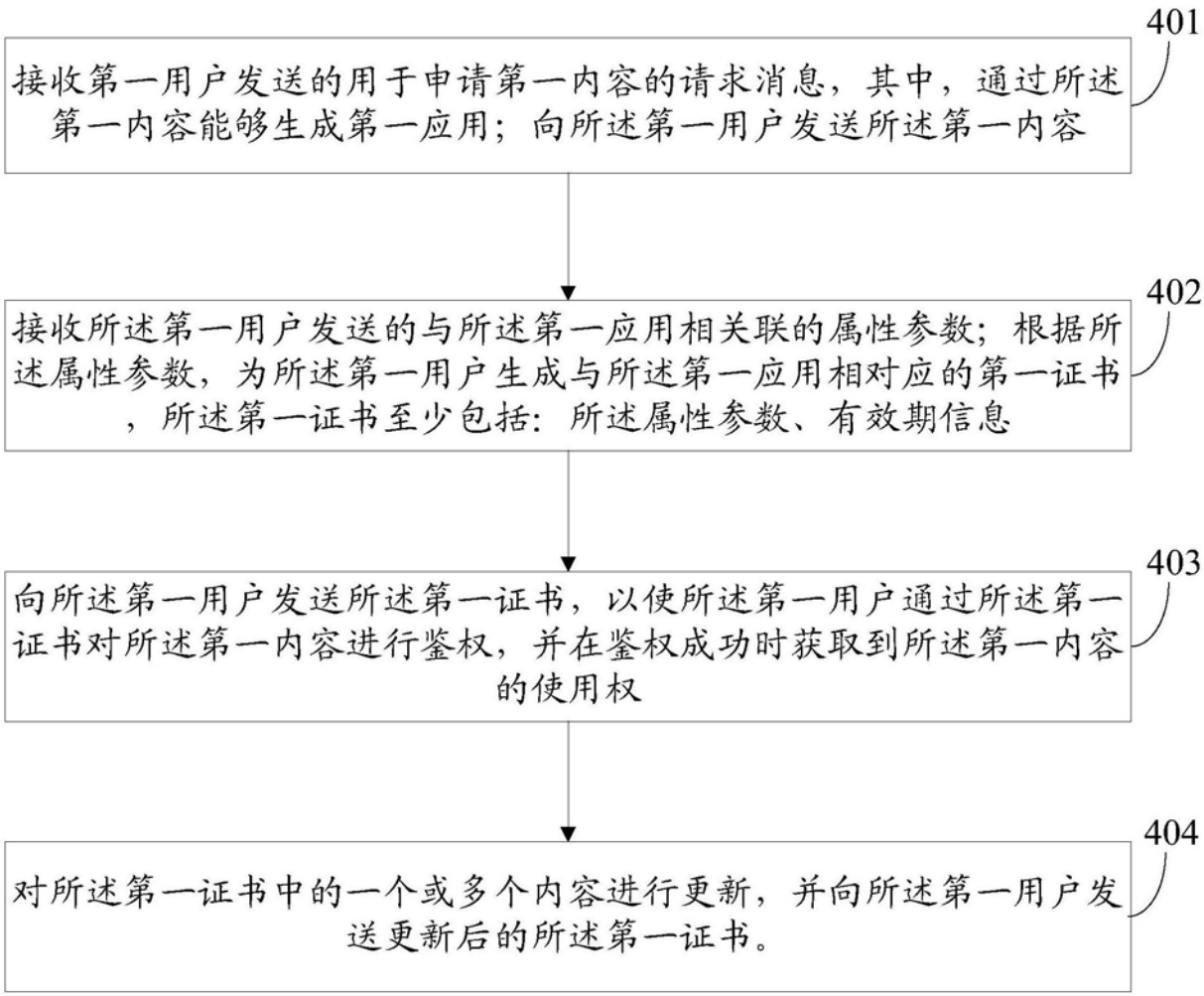

401
接收第一用户发送的用于申请第一内容的请求消息，其中，通过所述第一内容能够生成第一应用；向所述第一用户发送所述第一内容
402
接收所述第一用户发送的与所述第一应用相关联的属性参数；根据所述属性参数，为所述第一用户生成与所述第一应用相对应的第一证书，所述第一证书至少包括：所述属性参数、有效期信息
403
向所述第一用户发送所述第一证书，以使所述第一用户通过所述第一证书对所述第一内容进行鉴权，并在鉴权成功时获取到所述第一内容的使用权
404
对所述第一证书中的一个或多个内容进行更新，并向所述第一用户发送更新后的所述第一证书。

图4

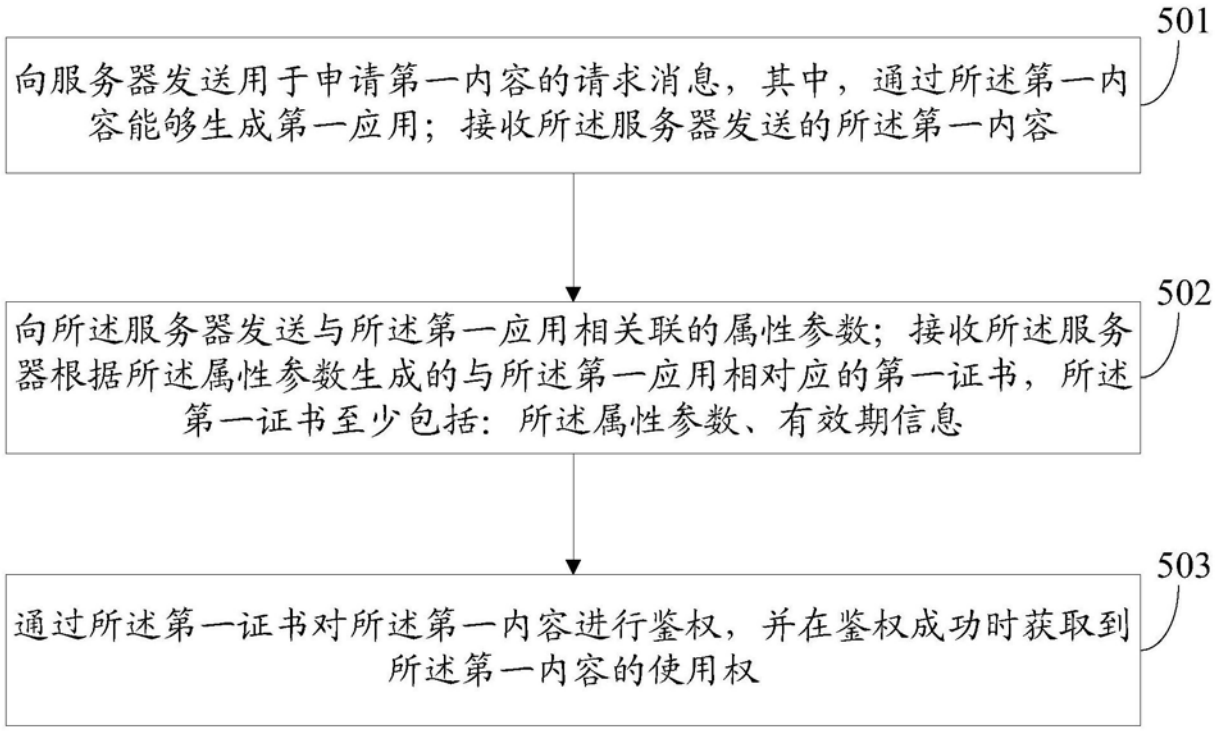

501
向服务器发送用于申请第一内容的请求消息，其中，通过所述第一内容能够生成第一应用；接收所述服务器发送的所述第一内容
502
向所述服务器发送与所述第一应用相关联的属性参数；接收所述服务器根据所述属性参数生成的与所述第一应用相对应的第一证书，所述第一证书至少包括：所述属性参数、有效期信息
503
通过所述第一证书对所述第一内容进行鉴权，并在鉴权成功时获取到所述第一内容的使用权

图5

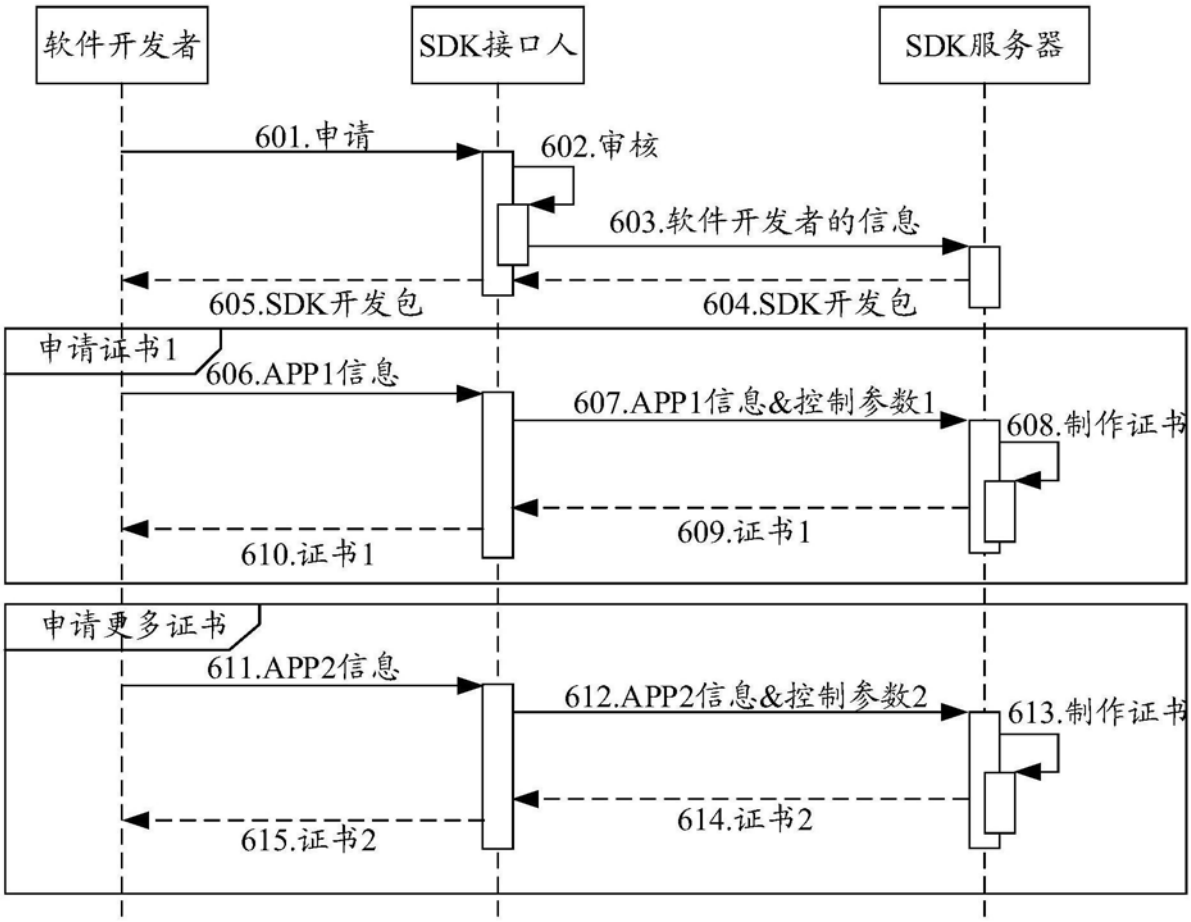

软件开发者
SDK接口人
SDK服务器
601.申请
602.审核
603.软件开发者的信息
605.SDK开发包
604.SDK开发包
申请证书1
606.APP1信息
607.APP1信息&控制参数1
608.制作证书
609.证书1
610.证书1
申请更多证书
611.APP2信息
612.APP2信息&控制参数2
613.制作证书
614.证书2
615.证书2

图6


第二接收单元73
第一接收单元71
生成单元74
确定单元77
第二发送单元75
更新单元76
打包单元78
第一发送单元72

图7

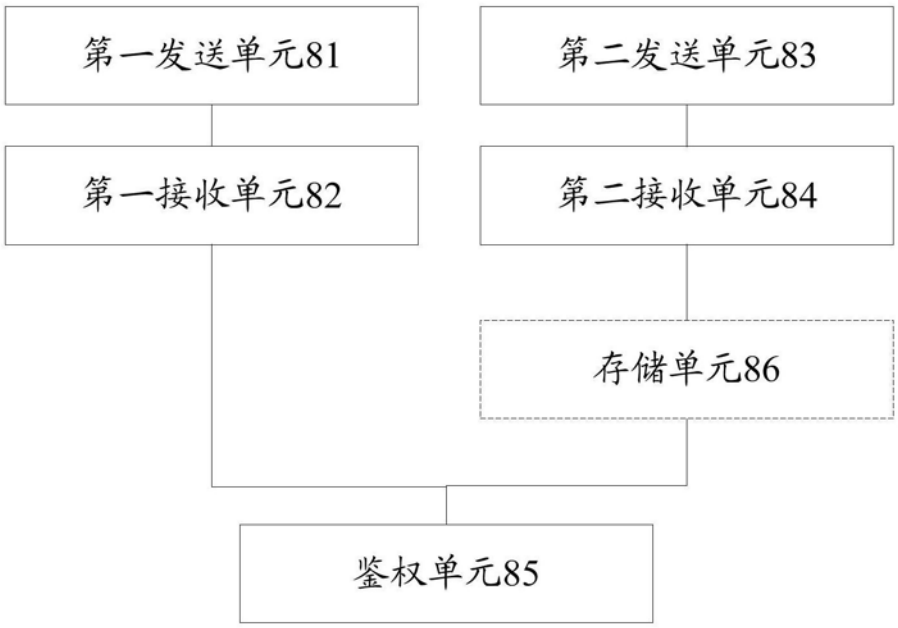


图8