(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 103685138 A
(43)申请公布日 2014.03.26

(21)申请号 201210315275.5
(22)申请日 2012.08.30
(71)申请人 卓望数码技术(深圳)有限公司
地址 518057 广东省深圳市南山区高新技术产业园南区深港产学研基地大楼西座六楼南翼
(72)发明人 王刚 刘志诚 吴勇 王有为 袁胜
(74)专利代理机构 深圳市顺天达专利商标代理有限公司 44217
代理人 李琴
(51)Int.Cl.
H04L 29/06(2006.01)
H04L 9/32(2006.01)
H04L 9/30(2006.01)

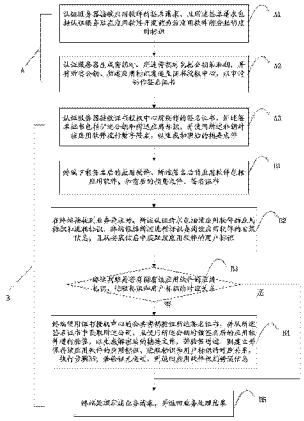
权利要求书2页 说明书5页 附图5页

(54)发明名称

移动互联网上的Android平台应用软件的认证方法和系统

(57)摘要

本发明公开了一种移动互联网上的Android平台应用软件的认证方法和系统，该认证方法包括：认证服务器接收应用软件的签名请求；生成密钥对，并将公钥、应用标识发送至证书授权中心；接收签名证书，并使用私钥对该应用软件进行数字签名；终端下载签名后的应用软件；根据进程标识查询该应用软件的安装信息，且从安装信息中获取用户标识；判断是否存储有该应用软件的应用标识、进程标识和用户标识的对应关系，若否，则验证签名证书，并使用公钥进行验签，在验签通过时建立并保存该应用软件的应用标识、进程标识和用户标识的对应关系，处理业务请求并返回业务处理结果。实施本发明的技术方案，能保证应用软件的来源真实性、完整性、防仿冒性。



CN 103685138 A

1. 一种移动互联网上的 Android 平台应用软件的认证方法，其特征在于，包括：

A. 签名步骤；及

B. 验签步骤；其中，

所述步骤 A 包括：

A1. 认证服务器接收应用软件的签名请求，且所述签名请求包括认证服务器在应用软件开发前为该应用软件所分配的应用标识；

A2. 认证服务器生成密钥对，所述密钥对包括公钥和私钥，并将所述公钥、所述应用标识发送至证书授权中心，以申请制作签名证书；

A3. 认证服务器接收证书授权中心所制作的签名证书，所述签名证书包括所述公钥和所述应用标识，并使用所述私钥对该应用软件进行数字签名，以生成加密后的摘要文件；

所述步骤 B 包括：

B1. 终端下载签名后的应用软件，所述签名后的应用软件包括应用软件、加密后的摘要文件、签名证书；

B2. 在终端接收到业务请求时，所述业务请求包括该应用软件的应用标识和进程标识，终端根据所述进程标识查询该应用软件的安装信息，且从安装信息中获取该应用软件的用户标识；

B3. 终端判断是否存储有该应用软件的应用标识、进程标识和用户标识的对应关系，若是，则执行步骤 B5；若否，则执行步骤 B4；

B4. 终端使用证书授权中心的公共密钥验证所述签名证书，并从所述签名证书中获取所述公钥，且使用所述公钥对该签名后的应用软件进行验签，以生成解密后的摘要文件，若验签通过，则建立并保存该应用软件的应用标识、进程标识和用户标识的对应关系，并执行步骤 B5；若验证无通过，则返回应用软件识别错误信息；

B5. 终端处理所述业务请求，并返回业务处理结果。

2. 根据权利要求 1 所述的移动互联网上的 Android 平台应用软件的认证方法，其特征在于，

在步骤 A3 中，使用 RSA 算法对该应用软件进行数字签名；

在步骤 B4 中，使用 RSA 算法对签名后的应用软件进行验签。

3. 一种移动互联网上的 Android 平台应用软件的认证系统，其特征在于，包括终端和认证服务器，且所述认证服务器包括：

签名请求接收单元，用于接收应用软件的签名请求，且所述签名请求包括认证服务器在应用软件开发前为该应用软件所分配的应用标识；

签名证书申请单元，用于生成密钥对，所述密钥对包括公钥和私钥，并将所述公钥、所述应用标识发送至证书授权中心，以申请制作签名证书；

签名单元，用于接收证书授权中心所制作的签名证书，所述签名证书包括所述公钥和所述应用标识，并使用所述私钥对该应用软件进行数字签名，以生成加密后的摘要文件；

所述终端包括：

下载单元，用于下载签名后的应用软件，所述签名后的应用软件包括应用软件、加密后的摘要文件、签名证书；

业务请求接收单元，用于接收业务请求，所述业务请求包括该应用软件的应用标识和

进程标识，终端根据所述进程标识查询该应用软件的安装信息，且从安装信息中获取该应用软件的用户标识；

判断单元，用于判断是否存储有该应用软件的应用标识、进程标识和用户标识的对应关系；

验签单元，用于在没有存储该应用软件的应用标识、进程标识和用户标识的对应关系时，使用证书授权中心的公共密钥验证所述签名证书，并从所述签名证书中获取所述公钥，且使用所述公钥对该签名后的应用软件进行验签，以生成解密后的摘要文件，若验签通过，则建立并保存该应用软件的应用标识、进程标识和用户标识的对应关系；若验证无通过，则返回应用软件识别错误信息；

处理单元，用于在存储有应用软件的应用标识、进程标识和用户标识的对应关系时时，处理业务请求，并返回业务处理结果。

4. 根据权利要求 3 所述的移动互联网上的 Android 平台应用软件的认证系统，其特征在于，

所述签名单元使用 RSA 算法对该应用软件进行数字签名；

所述验签单元使用 RSA 算法对该应用软件进行验签。

移动互联网上的 Android 平台应用软件的认证方法和系统

技术领域

[0001] 本发明涉及移动互联网信息安全领域，尤其涉及一种移动互联网上的 Android 平台应用软件的认证方法和系统。

背景技术

[0002] 随着基于 Android 系统的移动终端的日益普及，基于 Android 的应用软件数量增长迅速，能够为用户提供游戏、娱乐、社交、商务、旅行等各方面的业务功能。但可以看到，Android 系统本身是开放性的，对于应用软件的认证仅要求仅是自签名即可，并不对其来源的真实性、合法性提供认证，所以也为各种恶意软件的散布提供了可乘之机，对于用户的利益也会造成损失。所以为了保证终端应用和业务的安全，有必要对应用软件的真实性、合法性进行认证，保证只有被授权过的合法软件才能够进行业务访问。

[0003] 目前，通常使用应用 ID 对应用软件的认证，具体为：业务平台为应用软件分配一个代表应用软件身份的唯一字符串 ID，应用软件开发者会将此应用 ID 写入到应用软件包里。在用户使用该应用软件访问业务平台时候，该应用软件携带此应用 ID 向业务平台发起业务请求，业务平台校验应用 ID 是否有效，如果有效则认为该应用软件是真实的，且后续的业务请求都关联到此应用 ID。但是，采用分配应用 ID 进行应用软件认证的方法存在较大的安全隐患：在一种情况下，开发者将应用 ID 泄露给其他开发者，则其他开发者则可使用此应用 ID 来开发另一个假冒应用软件，而业务平台没有任何办法能够识别出应用 ID 已被转移使用；在另一种情况下，因为应用 ID 被内置在应用软件包里，存在被攻击者破解程序包后盗用的可能性，而对于持有应用 ID 的应用开发者来说则毫无察觉。

发明内容

[0004] 本发明要解决的技术问题在于，针对现有技术的上述应用软件的认证方法存在较大的安全隐患的缺陷，提供一种应用软件的认证方法，能保证应用软件的来源真实性、完整性、防仿冒性。

[0005] 本发明解决其技术问题所采用的技术方案是：构造一种移动互联网上的 Android 平台应用软件的认证方法，包括：

A. 签名步骤；及

B. 验签步骤；其中，

所述步骤 A 包括：

A1. 认证服务器接收应用软件的签名请求，且所述签名请求包括认证服务器在应用软件开发前为该应用软件所分配的应用标识；

A2. 认证服务器生成密钥对，所述密钥对包括公钥和私钥，并将所述公钥、所述应用标识发送至证书授权中心，以申请制作签名证书；

A3. 认证服务器接收证书授权中心所制作的签名证书，所述签名证书包括所述公钥和所述应用标识，并使用所述私钥对该应用软件进行数字签名，以生成加密后的摘要文件；

所述步骤 B 包括 ：

B1. 终端下载签名后的应用软件，所述签名后的应用软件包括应用软件、加密后的摘要文件、签名证书 ；

B2. 在终端接收到业务请求时，所述业务请求包括该应用软件的应用标识和进程标识，终端根据所述进程标识查询该应用软件的安装信息，且从安装信息中获取该应用软件的用户标识 ；

B3. 终端判断是否存储有该应用软件的应用标识、进程标识和用户标识的对应关系，若是，则执行步骤 B5 ；若否，则执行步骤 B4 ；

B4. 终端使用证书授权中心的公共密钥验证所述签名证书，并从所述签名证书中获取所述公钥，且使用所述公钥对该签名后的应用软件进行验签，以生成解密后的摘要文件，若验签通过，则建立并保存该应用软件的应用标识、进程标识和用户标识的对应关系，并执行步骤 B5 ；若验证无通过，则返回应用软件识别错误信息 ；

B5. 终端处理所述业务请求，并返回业务处理结果。

[0006] 在本发明所述的移动互联网上的 Android 平台应用软件的认证方法中，

在步骤 A3 中，使用 RSA 算法对该应用软件进行数字签名 ；

在步骤 B4 中，使用 RSA 算法对签名后的应用软件进行验签。

[0007] 本发明还构造一种移动互联网上的 Android 平台应用软件的认证系统，其特征在于，包括终端和认证服务器，且所述认证服务器包括 ：

签名请求接收单元，用于接收应用软件的签名请求，且所述签名请求包括认证服务器在应用软件开发前为该应用软件所分配的应用标识 ；

签名证书申请单元，用于生成密钥对，所述密钥对包括公钥和私钥，并将所述公钥、所述应用标识发送至证书授权中心，以申请制作签名证书 ；

签名单元，用于接收证书授权中心所制作的签名证书，所述签名证书包括所述公钥和所述应用标识，并使用所述私钥对该应用软件进行数字签名，以生成加密后的摘要文件 ；

所述终端包括 ：

下载单元，用于下载签名后的应用软件，所述签名后的应用软件包括应用软件、加密后的摘要文件、签名证书 ；

业务请求接收单元，用于接收业务请求，所述业务请求包括该应用软件的应用标识和进程标识，终端根据所述进程标识查询该应用软件的安装信息，且从安装信息中获取该应用软件的用户标识 ；

判断单元，用于判断是否存储有该应用软件的应用标识、进程标识和用户标识的对应关系 ；

验签单元，用于在没有存储该应用软件的应用标识、进程标识和用户标识的对应关系时，使用证书授权中心的公共密钥验证所述签名证书，并从所述签名证书中获取所述公钥，且使用所述公钥对该签名后的应用软件进行验签，以生成解密后的摘要文件，若验签通过，则建立并保存该应用软件的应用标识、进程标识和用户标识的对应关系 ；若验证无通过，则返回应用软件识别错误信息 ；

处理单元，用于在存储有应用软件的应用标识、进程标识和用户标识的对应关系时时，处理业务请求，并返回业务处理结果。

[0008] 在本发明所述的应用软件的认证系统中，

所述签名单元使用 RSA 算法对该应用软件进行数字签名；

所述验签单元使用 RSA 算法对该应用软件进行验签。

[0009] 实施本发明的技术方案，通过应用软件发布之前对应用软件颁发签名证书并对应用软件进行签名，应用软件访问业务平台时进行验签的方式来识别应用软件的真实身份，从而保证应用软件的来源真实性、完整性、防仿冒性。另外，本方案基于 PKI 密钥和签名验签的密码算法，再结合操作系统对进程标识、用户标识分配管理的安全机制，不但具有很高的安全性，同时除首次认证对应用软件进行验证处理耗时外，后续认证只是查询应用标识、进程标识、用户标识组关系是否存在，从而具有很高的处理性能。进一步地，本方案由于能够为基于移动互联网上的业务运营提供安全保障，从而为各种移动电子商务以及各类增值服务提供有力的支撑。

附图说明

[0010] 下面将结合附图及实施例对本发明作进一步说明，附图中：

图 1 是本发明移动互联网上的 Android 平台应用软件的认证方法实施例一的流程图；

图 2 是本发明移动互联网上的 Android 平台应用软件的认证方法中签名步骤实施例一的流程图；

图 3 是本发明移动互联网上的 Android 平台应用软件的认证方法中首次认证时的验签步骤实施例一的流程图；

图 4 是本发明移动互联网上的 Android 平台应用软件的认证方法中后续认证时的验签步骤实施例一的流程图；

图 5 是本发明移动互联网上的 Android 平台应用软件的认证系统实施例一的逻辑图。

具体实施方式

[0011] 如图 1 所示的本发明移动互联网上的 Android 平台应用软件的认证方法实施例一的流程图，该应用软件的认证方法包括

A. 签名步骤；及

B. 验签步骤；其中，

签名步骤包括：

A1. 认证服务器接收应用软件的签名请求，且签名请求包括 APPID（application identification，应用标识），应当说明是，该 APPID 是认证服务器在应用软件开发前为该应用软件所分配的唯一标识，在程序开发时，该 APPID 被植入到应用软件的程序包中；

A2. 认证服务器生成密钥对，密钥对包括公钥和私钥，并将公钥、APPID 发送至 CA（Certificate Authority 证书授权）中心，以申请制作签名证书。关于 CA 中心，应说明的是，CA 中心是一家能向用户签发签名证书以确认用户身份的第三方管理机构。为了防止数字凭证的伪造，CA 中心的公共密钥必须是可靠的，CA 中心必须公布其公共密钥或由更高级别的认证中心提供一个电子凭证来证明其公共密钥的有效性。CA 中心在颁发签名证书时，把 APPID 和公钥封装成签名证书，签名证书的尾部必须有 CA 中心的数字签名。由于 CA 中心的数字签名是不可伪造的，因此该应用软件的签名证书不可伪造。CA 中心对该应用软件

的身份资格审查通过后，才对申请者颁发签名证书，将该应用软件的身份与签名证书对应起来；

A3. 认证服务器接收证书授权中心所制作的签名证书，签名证书包括公钥和 APPID，并使用私钥对该应用软件进行数字签名，以生成加密后的摘要文件。关于数字签名，应当说明的是，首先从应用软件中生成一个 128 位的散列值(即摘要)。接着，用密钥对中的私钥对这个摘要进行加密来形成数字签名。然后，这个数字签名将作为应用软件的附件和应用软件一起发送给应用软件的开发者；

验签步骤包括：

B1. 终端下载签名后的应用软件，签名后的应用软件包括应用软件、加密后的摘要文件、签名证书；

B2. 在终端接收到业务请求时，业务请求包括该应用软件的 APPID 和 PID（Process Identifier，进程标识），终端根据 PID 查询该应用软件的安装信息，且从安装信息中获取该应用软件的 UID（user identifier，用户标识）；

B3. 终端判断是否存储有该应用软件的 APPID、PID 和 UID 的对应关系，若是，则执行步骤 B5；若否，则执行步骤 B4；

B4. 终端使用 CA 中心的公共密钥验证签名证书，并从签名证书中获取公钥，且使用公钥对该签名后的应用软件进行验签，以生成解密后的摘要文件，若验签通过，则建立并保存该应用软件的 APPID、PID 和 UID 的对应关系，执行步骤 B5；若验证无通过，则返回应用软件识别错误信息。关于验签是否通过，应说明的是，终端首先从接收到的应用软件中计算出 128 位的散列值(即摘要)，接着再用公钥来对加密后的摘要文件进行解密。如果两个散列值相同，那么就能确认该数字签名是认证服务器的；相反，如果两个散列值不相同，那么就能确认该数字签名不是认证服务器的；

B5. 终端处理业务请求，并返回业务处理结果。

[0012] 在本发明应用软件的认证方法的一个优选实施例中，在步骤 A3 中，可使用 RSA 算法对该应用软件进行数字签名；在步骤 B4 中，使用 RSA 算法对签名后的应用软件进行验签。

[0013] 图 2 是本发明移动互联网上的 Android 平台应用软件的认证方法中签名步骤实施例一的流程图，在应用软件开发环节中，应用软件开发者向认证服务器上传应用软件，并请求对该应用软件进行签名，此应用软件的程序包里已经植入认证服务器事先所分配的 APPID；认证服务器生成 PKI 密钥对，并向 CA 中心提交 APPID、应用软件程序包名、公钥等信息，以向 CA 中心申请签名证书。认证服务器向 CA 中心申请签名证书成功后，使用私钥对应用软件进行数字签名，按照 Android 程序包签名证书格式替换到原来的开发用 Debug 证书，应用程序签名完成，并通知应用软件开发者签名完成，开发者从平台下载签名后的应用软件，签名后的应用软件可以发布到用户终端上进行使用。

[0014] 图 3 是本发明移动互联网上的 Android 平台应用软件的认证方法中首次认证时的验签步骤实施例一的流程图，首先说明的是，认证代理是安装运行在用户移动终端上软件，对于终端设备上的应用软件进行识别和访问控制。在首次认证环节中，终端应首先将签名后的应用软件下载到终端中。然后，终端内的应用软件向认证代理(Agent)发起业务请求，且携带有应用软件自身的 PID 和 APPID。接着，认证代理根据 PID 查询操作系统进程信息和安装信息，获得 PID 对应的应用软件的程序包信息，这些信息里包括程序包文件路径、应用

软件的 UID 信息。随后，认证代理使用内置的平台 CA 中心的公钥对签名证书进行验证，确保证书的真实性和里面包含的 APPID 的正确性，并对应用软件进行签名验签，验签如果通过则表示应用程序包未被篡改、是真实的，否则直接返回身份识别错误。另外，验证如果通过，认证代理还建立并保存 PID、UID、APPID 三元组信息。最后，在验签通过时，认证代理根据应用软件的授权处理业务请求并返回业务处理结果。

[0015] 图 4 是本发明移动互联网上的 Android 平台应用软件的认证方法中后续认证时的验签步骤实施例一的流程图，在后续认证环节，应用软件向认证代理发起业务请求，携带应用自身的 PID 和 APPID。然后，认证代理根据 PID 查询操作系统进程信息和应用安装信息，获得 PID 对应的应用软件的 APPID、UID，接着判断是否已存在 PID、UID、APPID 三元组信息，如果三元组不存在，则表示该应用软件尚未通过应用认证，转入首次认证环节里的验签名处理流程，对应用软件重新进行认证；如果三元组存在，则表示应用软件已经通过应用认证、是真实的，则继续业务处理，认证代理根据应用软件的授权处理业务请求并返回业务处理结果。

[0016] 图 5 是本发明移动互联网上的 Android 平台应用软件的认证系统实施例一的逻辑图，该应用软件的认证系统包括认证服务器 10 和终端 20。而且，认证服务器 10 包括签名请求接收单元 11、签名证书申请单元 12 和签名单元 13；终端 20 包括下载单元 21、业务请求接收单元 22、判断单元 23、验签单元 24 和处理单元 25。其中，在软件开发环节，签名请求接收单元 11 用于接收应用软件的签名请求，且签名请求包括认证服务器 10 在应用软件开发前为该应用软件所分配的 APPID；签名证书申请单元 12 用于生成密钥对，密钥对包括公钥和私钥，并将公钥、APPID 发送至 CA 中心，以申请制作签名证书；签名单元 13 用于接收 CA 中心所制作的签名证书，签名证书包括公钥和 APPID，并使用私钥对该应用软件进行数字签名，以生成加密后的摘要文件。在认证环节，下载单元 21 用于下载签名后的应用软件，签名后的应用软件包括应用软件、加密后的摘要文件、签名证书；业务请求接收单元 22 用于接收业务请求，业务请求包括该应用软件的 APPID 和 PID，终端根据 PID 查询该应用软件的安装信息，且从安装信息中获取该应用软件的 UID；判断单元 23 用于判断是否存储有该应用软件的 APPID、PID 和 UID 的对应关系；验签单元 24 用于在没有存储该应用软件的 APPID、PID 和 UID 的对应关系时，使用 CA 中心的公共密钥验证签名证书，并从签名证书中获取公钥，且使用公钥对该签名后的应用软件进行验签，以生成解密后的摘要文件，若验签通过，则建立并保存该应用软件的 APPID、PID 和 UID 的对应关系；若验证无通过，则返回应用软件识别错误信息；处理单元 25 用于在存储有应用软件的 APPID、PID 和 UID 的对应关系时时，处理业务请求，并返回业务处理结果。

[0017] 在本发明应用软件的认证系统的一个优选实施例中，签名单元 13 可使用 RSA 算法对该应用软件进行数字签名；验签单元 24 可使用 RSA 算法对该应用软件进行验签。

[0018] 以上仅为本发明的优选实施例而已，并不用于限制本发明，对于本领域的技术人员来说，本发明可以有各种更改和变化。凡在本发明的精神和原则之内，所作的任何修改、等同替换、改进等，均应包含在本发明的权利要求范围之内。

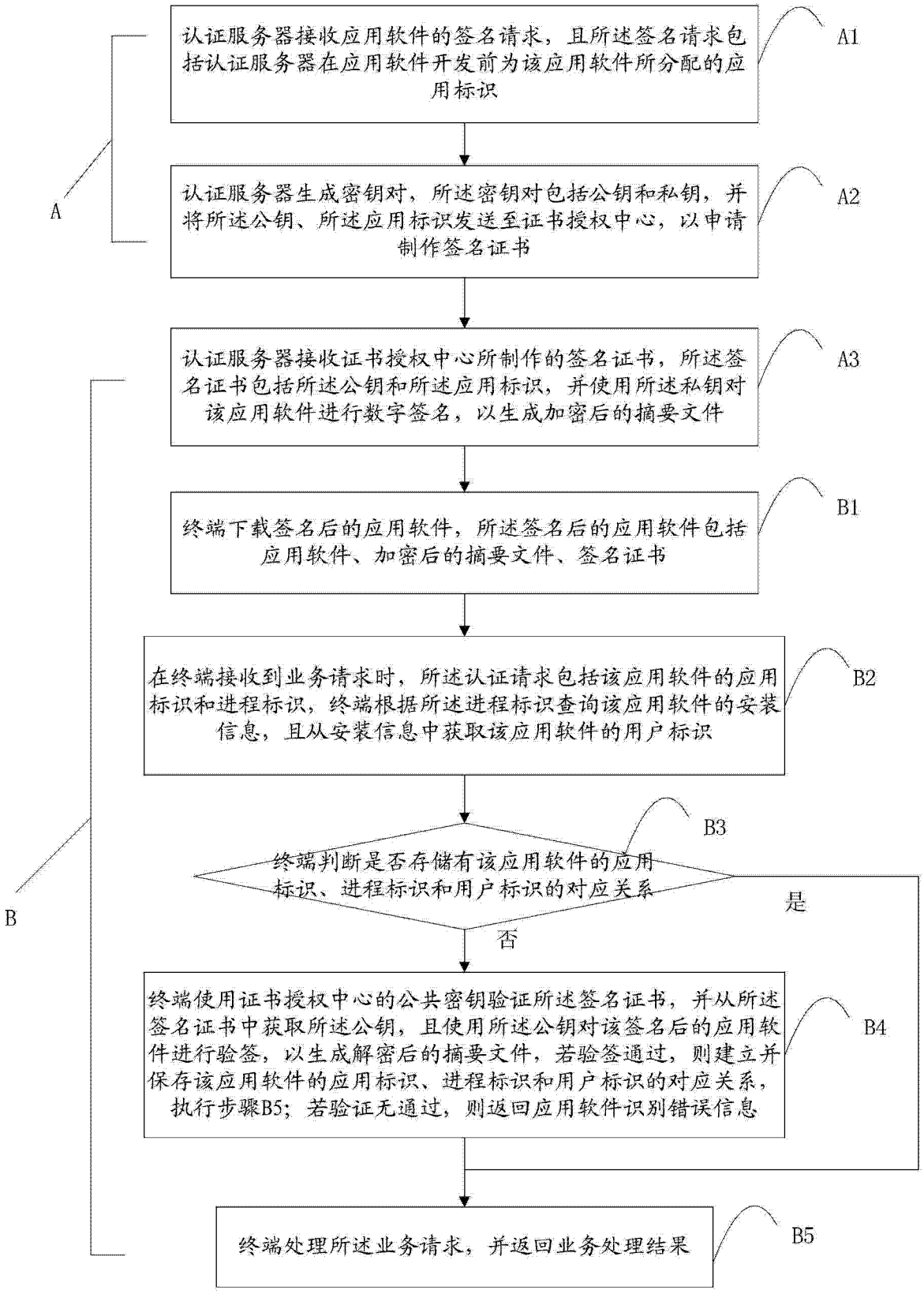

认证服务器接收应用软件的签名请求，且所述签名请求包括认证服务器在应用软件开发前为该应用软件所分配的应用标识
A1
A
认证服务器生成密钥对，所述密钥对包括公钥和私钥，并将所述公钥、所述应用标识发送至证书授权中心，以申请制作签名证书
A2
认证服务器接收证书授权中心所制作的签名证书，所述签名证书包括所述公钥和所述应用标识，并使用所述私钥对该应用软件进行数字签名，以生成加密后的摘要文件
A3
终端下载签名后的应用软件，所述签名后的应用软件包括应用软件、加密后的摘要文件、签名证书
B1
在终端接收到业务请求时，所述认证请求包括该应用软件的应用标识和进程标识，终端根据所述进程标识查询该应用软件的安装信息，且从安装信息中获取该应用软件的用户标识
B2
B
终端判断是否存储有该应用软件的应用标识、进程标识和用户标识的对应关系
B3
是
否
终端使用证书授权中心的公共密钥验证所述签名证书，并从所述签名证书中获取所述公钥，且使用所述公钥对该签名后的应用软件进行验签，以生成解密后的摘要文件，若验签通过，则建立并保存该应用软件的应用标识、进程标识和用户标识的对应关系，执行步骤B5；若验证无通过，则返回应用软件识别错误信息
B4
终端处理所述业务请求，并返回业务处理结果
B5

图 1

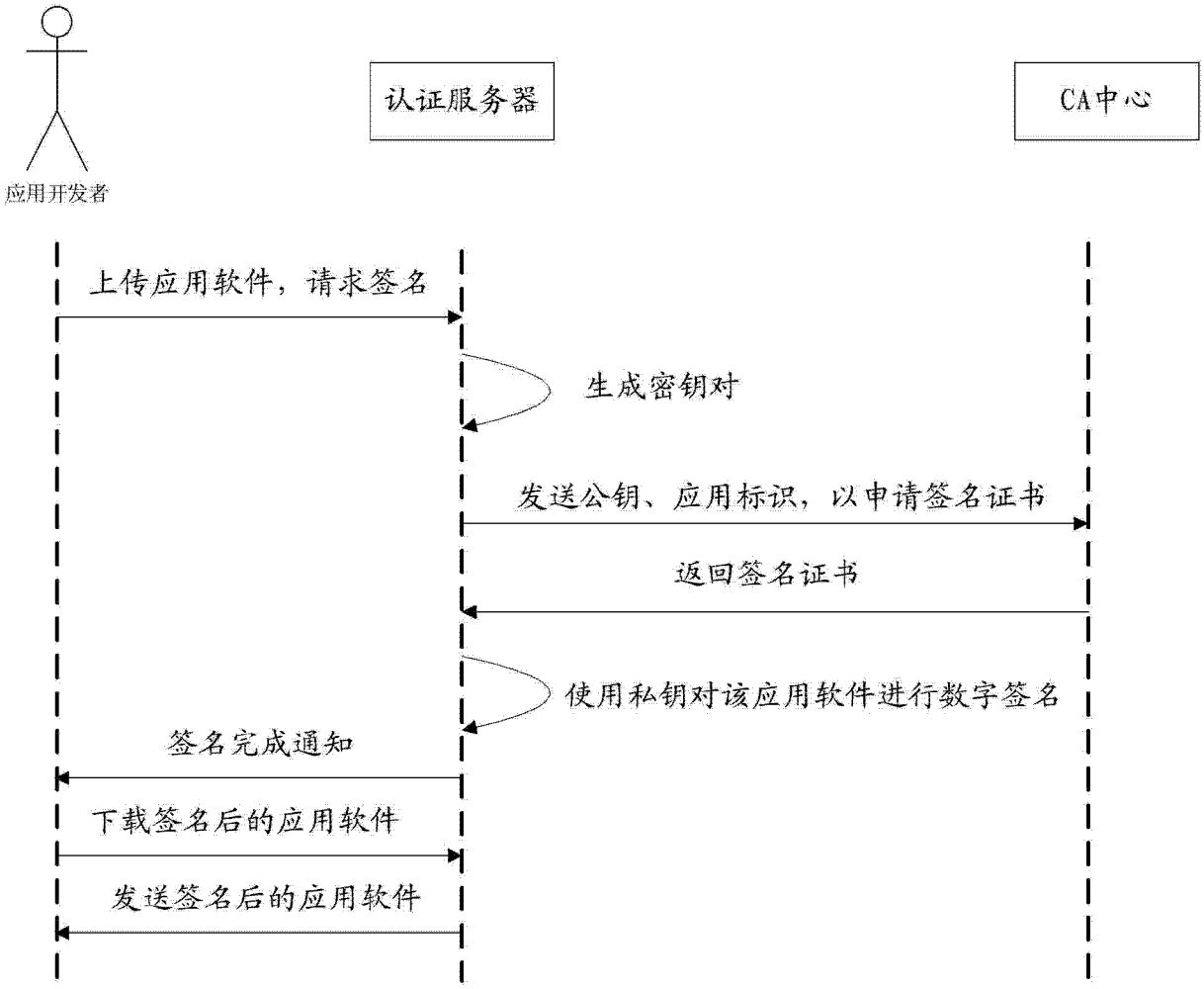

应用开发者
认证服务器
CA中心
上传应用软件，请求签名
生成密钥对
发送公钥、应用标识，以申请签名证书
返回签名证书
使用私钥对该应用软件进行数字签名
签名完成通知
下载签名后的应用软件
发送签名后的应用软件

图 2

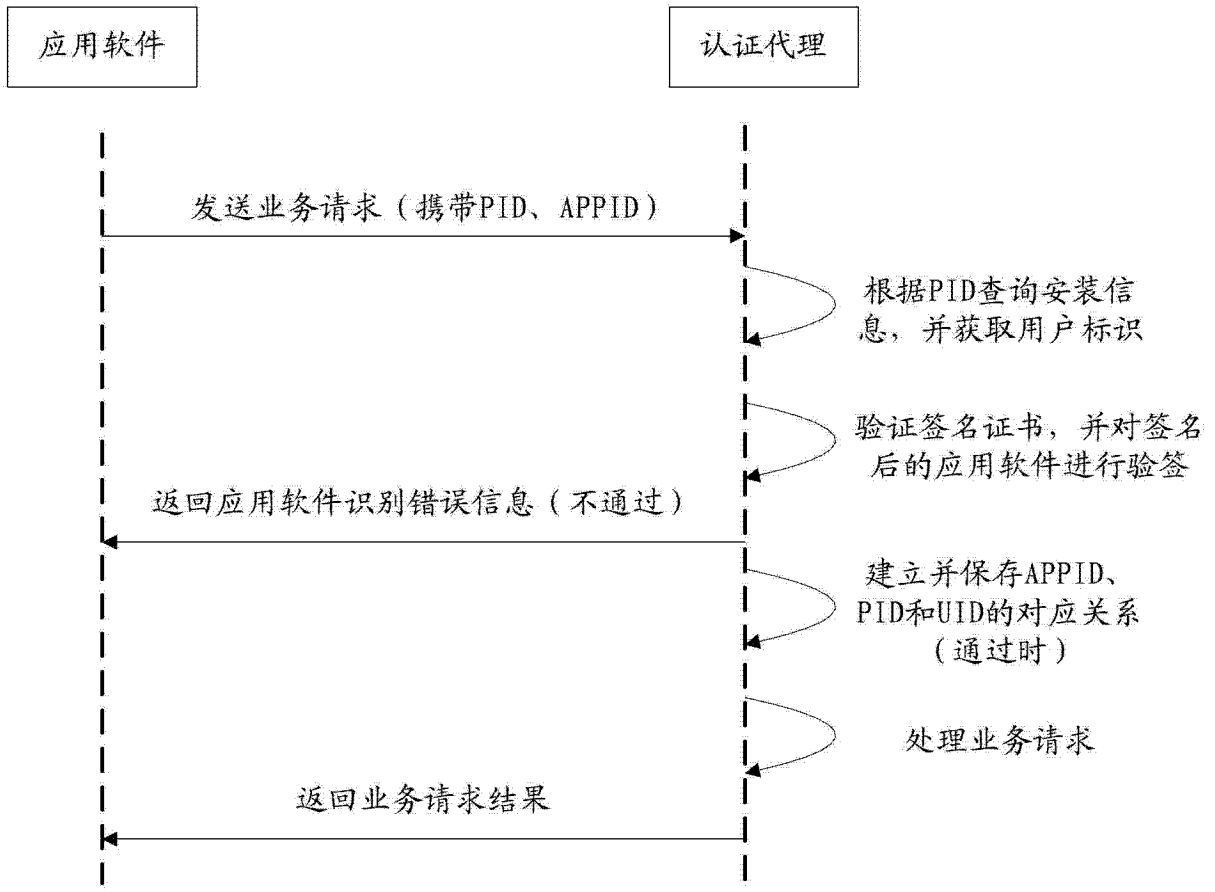

应用软件
认证代理
发送业务请求（携带PID、APPID）
根据PID查询安装信息，并获取用户标识
验证签名证书，并对签名后的应用软件进行验签
返回应用软件识别错误信息（不通过）
建立并保存APPID、PID和UID的对应关系（通过时）
处理业务请求
返回业务请求结果

图 3

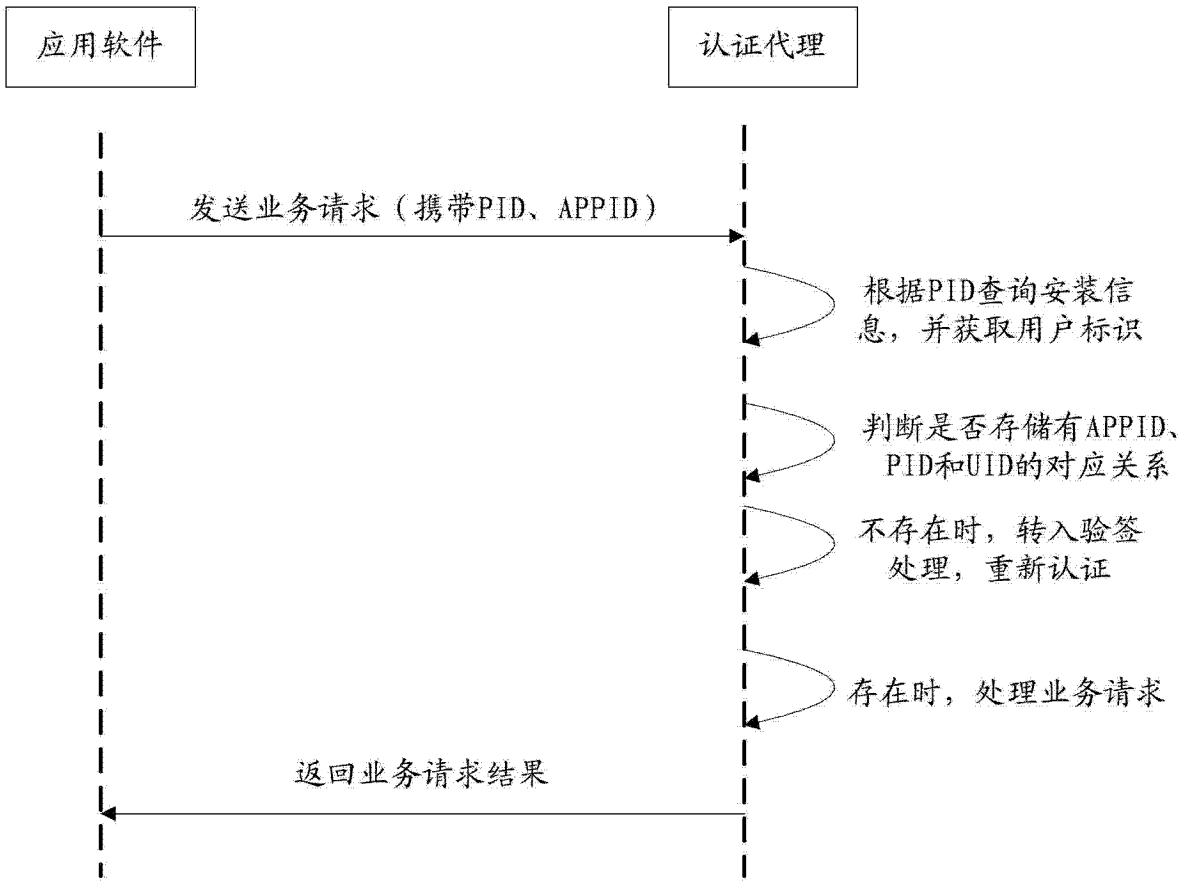

应用软件
认证代理
发送业务请求（携带PID、APPID）
根据PID查询安装信息，并获取用户标识
判断是否存储有APPID、PID和UID的对应关系
不存在时，转入验签处理，重新认证
存在时，处理业务请求
返回业务请求结果

图 4

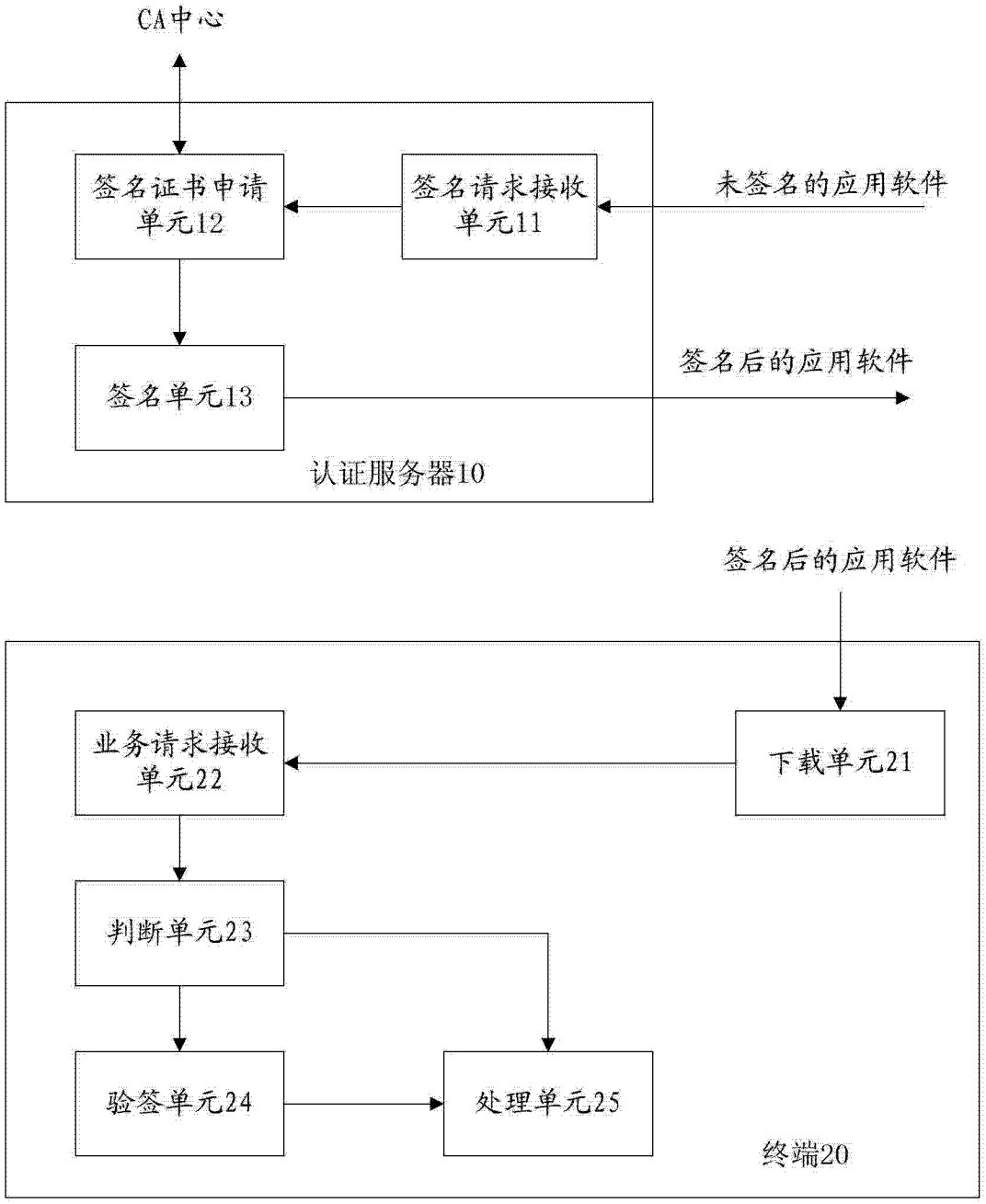

CA中心
签名证书申请单元12
签名请求接收单元11
未签名的应用软件
签名单元13
签名后的应用软件
认证服务器10
签名后的应用软件
业务请求接收单元22
下载单元21
判断单元23
验签单元24
处理单元25
终端20

图 5