(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 104866739 A
(43)申请公布日 2015.08.26

(21)申请号 201510305704.4

(22)申请日 2015.06.04

(71)申请人 上海斐讯数据通信技术有限公司
地址 201616 上海市松江区思贤路 3666 号

(72)发明人 杨希锋 张莹莹

(74)专利代理机构 上海光华专利事务所 31219
代理人 高彦

(51)Int.Cl.
G06F 21/12(2013.01)

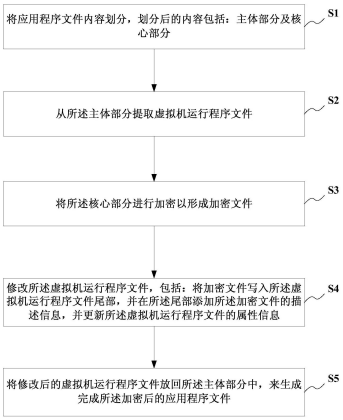
权利要求书2页 说明书5页 附图2页

(54)发明名称

安卓系统中应用程序加密方法及系统

(57)摘要

本发明提供的安卓系统中应用程序加密方法及系统，将应用程序文件内容划分，划分后的内容包括：主体部分及核心部分；从所述主体部分提取虚拟机运行程序文件；将所述核心部分进行加密以形成加密文件；修改所述虚拟机运行程序文件，包括：将加密文件写入所述虚拟机运行程序文件尾部，并在所述尾部添加所述加密文件的描述信息，并更新所述虚拟机运行程序文件的属性信息；将修改后的虚拟机运行程序文件放回所述主体部分中，来生成完成所述加密后的应用程序文件；通过本发明的技术方案，逆向工程所获得的代码中仍然是只有软件主体部分代码，核心功能部分的代码被隐藏可以有效对抗各种针对安卓系统中应用软件的逆向工程攻击，提高安全性。



CN 104866739 A

1. 一种安卓系统中应用程序加密方法，其特征在于，所述方法包括：

将应用程序文件内容划分，划分后的内容包括：主体部分及核心部分；

从所述主体部分提取虚拟机运行程序文件；

将所述核心部分进行加密以形成加密文件；

修改所述虚拟机运行程序文件，包括：将加密文件写入所述虚拟机运行程序文件尾部，并在所述尾部添加所述加密文件的描述信息，并更新所述虚拟机运行程序文件的属性信息；

将修改后的虚拟机运行程序文件放回所述主体部分中，来生成完成所述加密后的应用程序文件。

2. 根据权利要求 1 所述的安卓系统中应用程序加密方法，其特征在于，所述主体部分及核心部分均编译为独立的 .apk 文件。

3. 根据权利要求 1 所述的安卓系统中应用程序加密方法，其特征在于，所述虚拟机运行程序文件为 classes.dex 文件。

4. 根据权利要求 3 所述的安卓系统中应用程序加密方法，其特征在于，所述描述信息包括：加密文件的名称、位置及长度。

5. 根据权利要求 3 所述的安卓系统中应用程序加密方法，其特征在于，所述更新属性信息包括：重新计算所述虚拟机运行程序文件所包含的：校验和字段、签名字段和文件长度字段的值并替换原有值。

6. 根据权利要求 1 所述的安卓系统中应用程序加密方法，其特征在于，还包括：通过安卓软件开发包中提供的签名工具所述加密后的应用程序文件进行签名。

7. 一种安卓系统中应用程序加密系统，其特征在于，包括：

划分模块，用于将应用程序文件内容划分，划分后的内容包括：主体部分及核心部分；

提取模块，用于从所述主体部分提取虚拟机运行程序文件；

加密模块，用于将所述核心部分进行加密以形成加密文件；

修改模块，用于修改所述虚拟机运行程序文件，包括：将加密文件写入所述虚拟机运行程序文件尾部，并在所述尾部添加所述加密文件的描述信息，并更新所述虚拟机运行程序文件的属性信息；

文件生成模块，用于将修改后的虚拟机运行程序文件放回所述主体部分中，来生成完成所述加密后的应用程序文件。

8. 根据权利要求 7 所述的安卓系统中应用程序加密系统，其特征在于，所述主体部分及核心部分均编译为独立的 .apk 文件。

9. 根据权利要求 7 所述的安卓系统中应用程序加密系统，其特征在于，所述虚拟机运行程序文件为 classes.dex 文件。

10. 根据权利要求 9 所述的安卓系统中应用程序加密系统，其特征在于，所述描述信息包括：加密文件的名称、位置及长度。

11. 根据权利要求 9 所述的安卓系统中应用程序加密系统，其特征在于，所述更新属性信息包括：重新计算所述虚拟机运行程序文件所包含的：校验和字段、签名字段和文件大小字段的值并替换原有值。

12. 根据权利要求 9 所述的安卓系统中应用程序加密系统，其特征在于，还包括：签名

模块，用于通过安卓软件开发包中提供的签名工具对所述加密后的应用程序文件进行签名。

安卓系统中应用程序加密方法及系统

技术领域

[0001] 本发明涉及移动终端软件系统技术领域，特别是涉及安卓系统中应用程序加密方法及系统。

背景技术

[0002] 避免 Android 软件被破解和攻击是对开发者的技术方案的保护。目前，市场上有很多用于破解 Android 应用 APK 的工具，可以反编译出界面布局文件，甚至 Java 源码文件。这使得盗版软件遍地而出，也影响整个软件行业的发展。本发明提出了一种对 Android 应用软件进行加密的安全技术，通过动态加载 classes.dex 来实现对核心代码的加密，从而有效地保护软件的知识产权，通过加密后，反编译也难以看到原来的文件

[0003] 具体的，Android 平台使用 Java 编程语言，而 Java 源代码编译后的二进制文件极易被反编译，导致比其它的语言更容易被破解。目前有一些工具如 dex2jar、apktool 等便可以反编译出 Android 的源码文件。采用复杂的签名算法可以保护 Java 文件、jar 和 so 等链接库文件，这些被破解的难度很大；但是资源文件，主要是软件的 UI（图片、音频等文件）和界面布局（xml 文件）可以轻易的破解，这些资源也是 UI 工程师和前台工程师开发的作品，目前的技术还保护不了。

[0004] 申请号为 CN201310509543.1 的中国专利揭露了一种移动应用的软件加固技术，方法是为每个应用定制微型的虚拟机，在应用程序运行时先运行微型虚拟机，对应用程序整体进行安全性检查或验证，并在安全通过后再运行应用程序的功能模块，所述预设的安全要求包括应用程序的签名验证，应用程序中无效指令、非法指令的去除以及加密信息的解密。但是这一技术偏理论化，难以在真实的场景中普及。

[0005] 申请号为 CN201110429661.2 的中国专利揭露了 Android 系统中 Dalvik 虚拟机和 Linux 系统库增加接口，使 Android 具有从内存中直接加载 DEX 格式文件和 SO 格式文件的能力；将应用软件的核心代码存储在在线服务器中，加密并签名后发送给安装在客户端的应用软件；应用软件接收到核心代码后验证签名并解密，然后将明文存储在内存中，直接加载到系统中，然后调用其中的代码，最后释放内存。

[0006] 但是这种方法比较复杂，一是要在 Android 平台上修改系统代码，增加额外的接口；二是需要在线服务器，这无形中增加了开发者的压力。

发明内容

[0007] 鉴于以上所述现有技术的缺点，本发明的目的在于提供安卓系统中应用程序加密方法及系统，解决现有技术中的种种问题，提升软件安全性的同时降低开发成本。

[0008] 为实现上述目标及其他相关目标，本发明提供一种安卓系统中应用程序加密方法，所述方法包括：将应用程序文件内容划分，划分后的内容包括：主体部分及核心部分；从所述主体部分提取虚拟机运行程序文件；将所述核心部分进行加密以形成加密文件；修改所述虚拟机运行程序文件，包括：将加密文件写入所述虚拟机运行程序文件尾部，并在

所述尾部添加所述加密文件的描述信息，并更新所述虚拟机运行程序文件的属性信息；将修改后的虚拟机运行程序文件放回所述主体部分中，来生成完成所述加密后的应用程序文件。

[0009] 可选的，所述主体部分及核心部分均编译为独立的 .apk 文件。

[0010] 可选的，所述虚拟机运行程序文件为 classes.dex 文件。

[0011] 可选的，所述描述信息包括：加密文件的名称、位置及长度。

[0012] 可选的，所述更新属性信息包括：重新计算所述虚拟机运行程序文件所包含的：校验和字段、签名字段和文件长度字段的值并替换原有值。

[0013] 可选的，所述的安卓系统中应用程序加密方法，还包括：通过安卓软件开发包中提供的签名工具所述加密后的应用程序文件进行签名。

[0014] 为实现上述目标及其他相关目标，本发明提供一种安卓系统中应用程序加密系统，包括：划分模块，用于将应用程序文件内容划分，划分后的内容包括：主体部分及核心部分；提取模块，用于从所述主体部分提取虚拟机运行程序文件；加密模块，用于将所述核心部分进行加密以形成加密文件；修改模块，用于修改所述虚拟机运行程序文件，包括：将加密文件写入所述虚拟机运行程序文件尾部，并在所述尾部添加所述加密文件的描述信息，并更新所述虚拟机运行程序文件的属性信息；文件生成模块，用于将修改后的虚拟机运行程序文件放回所述主体部分中，来生成完成所述加密后的应用程序文件。

[0015] 可选的，所述主体部分及核心部分均编译为独立的 .apk 文件。

[0016] 可选的，所述虚拟机运行程序文件为 classes.dex 文件。

[0017] 可选的，所述描述信息包括：加密文件的名称、位置及长度。

[0018] 可选的，所述更新属性信息包括：重新计算所述虚拟机运行程序文件所包含的：校验和字段、签名字段和文件大小字段的值并替换原有值。

[0019] 可选的，所述的安卓系统中应用程序加密系统，还包括：签名模块，用于通过安卓软件开发包中提供的签名工具对所述加密后的应用程序文件进行签名。

[0020] 如上所述，本发明提供的安卓系统中应用程序加密方法及系统，将应用程序文件内容划分，划分后的内容包括：主体部分及核心部分；从所述主体部分提取虚拟机运行程序文件；将所述核心部分进行加密以形成加密文件；修改所述虚拟机运行程序文件，包括：将加密文件写入所述虚拟机运行程序文件尾部，并在所述尾部添加所述加密文件的描述信息，并更新所述虚拟机运行程序文件的属性信息；将修改后的虚拟机运行程序文件放回所述主体部分中，来生成完成所述加密后的应用程序文件；通过本发明的技术方案，逆向工程所获得的代码中仍然是只有软件主体部分代码，核心功能部分的代码被隐藏可以有效对抗各种针对安卓系统中应用软件的逆向工程攻击，提高安全性。

附图说明

[0021] 图 1 显示为本发明一实施例中的安卓系统中应用程序加密方法的流程示意图。

[0022] 图 2 显示为本发明一具体实例中安卓系统中应用程序加密方法的流程示意图。

[0023] 图 3 显示为本发明一实施例中的安卓系统中应用程序加密系统的结构示意图。

[0024] 元件标号说明

[0025] 1 应用程序加密系统

[0026] 11 划分模块

[0027] 12 提取模块

[0028] 13 加密模块

[0029] 14 修改模块

[0030] 15 文件生成模块

[0031] S1 ～ S5 方法步骤

具体实施方式

[0032] 以下通过特定的具体实例说明本发明的实施方式，本领域技术人员可由本说明书所揭露的内容轻易地了解本发明的其他优点与功效。本发明还可以通过另外不同的具体实施方式加以实施或应用，本说明书中的各项细节也可以基于不同观点与应用，在没有背离本发明的精神下进行各种修饰或改变。需说明的是，在不冲突的情况下，本申请中的实施例及实施例中的特征可以相互组合。

[0033] 本发明的技术方案应用于安卓 (Android) 系统中，安卓系统 (Android) 是一种基于 Linux 的自由及开放源代码的操作系统，主要使用于移动设备，如智能手机和平板电脑，由 Google 公司和开放手机联盟领导及开发，因此，在以下实施例中，涉及的专业性词汇可在 Android 相关的在先技术中查找，故不作详细赘述。

[0034] 如图 1 所示，本发明提供一种安卓系统中应用程序加密方法，所述方法包括 ：

[0035] 步骤 S1 ：将应用程序文件内容划分，划分后的内容包括 ：主体部分及核心部分。

[0036] 在一实施例中，所述应用程序文件在安卓系统中一般是以 .apk 后缀名的形式出现的，所述软件主体部分是实际安装在 Android 系统中的部分，核心功能部分是软件主体部分调用的需要重点保护的功能代码，由于该内容本领域技术人员均可根据本发明的教示来结合实际需求加以变化，因此此处不作赘述 ；优选的，所述主体部分及核心部分皆编译为独立的 .apk 格式。

[0037] 步骤 S2 ：从所述主体部分提取虚拟机运行程序文件。

[0038] 在一实施例中，所述虚拟机运行程序文件为 classes.dex 文件，Android 系统中的应用都运行在虚拟机上，因此 classes.dex 就是包含其的应用程序在虚拟机运行的执行文件。

[0039] 步骤 S3 ：将所述核心部分进行加密以形成加密文件。

[0040] 在一实施例中，对核心部分 .apk 文件的加密方式可为现有的加密算法或通过现有的反编译工具进行，故此处不作赘述 ；优选的，所述加密方式中可加入签名的字段值作为特征值。

[0041] 步骤 S4 ：修改所述虚拟机运行程序文件，包括 ：将加密文件写入所述虚拟机运行程序文件尾部，并在所述尾部添加所述加密文件的描述信息，并更新所述虚拟机运行程序文件的属性信息。

[0042] 在一实施例中，所述描述信息包括 ：加密文件的名称、位置及长度，用以解密时找到核心部分代码的起始位置 ；而所述更新属性信息包括 ：重新计算所述虚拟机运行程序文件所包含的 ：校验和字段、签名字段和文件长度字段的值并替换原有值，即 classes.dex 文件的 checksum、signature 和 file_size 字段值。

[0043] 步骤 S5 :将修改后的虚拟机运行程序文件放回所述主体部分中,来生成完成所述加密后的应用程序文件。

[0044] 本发明之所以利用该文件的原因在于,Android 系统在解析执行 classes.dex 文件的时候,如果发现其文件头的校验码字段与 SHA-1 签名字段无误,则认为此文件未损坏或未被篡改,是可以执行的。因此,可以猜想,如果在 classes.dex 文件后面增加一些内容,同时在增加这些内容后,修改 classes.dex 文件的校验码、SHA-1 签名及文件大小(修改文件后,该字段也会相应发生变化)字段,classes.dex 仍然能够正确执行。

[0045] 基于这个原理,本方案把部分软件的核心部分代码编译成一个独立的文件,添加到 classes.dex 文件后面,然后在程序运行时动态地分离出这些代码,再通过反射机制对这部分关键代码进行动态加载。同时,可以对这些核心部分代码进行加密处理,需要执行时再进行解密。经过这样处理后,逆向工程逆向出来的代码便只有软件的主体部分,而核心部分被隐藏。而且,即使核心部分被发现了,也会由于代码被人为加密而无法得到解密后的源码,从而很好地保护了核心代码。

[0046] 在一实施例中,所述的安卓系统中应用程序加密方法,还包括 :通过安卓软件开发包 (Android SDK) 中提供的签名工具所述加密后的应用程序文件进行签名。

[0047] 具体的,以下介绍签名的具体内容 :

[0048] apk 签名的作用 :

[0049] 在 Android 系统中,所有安装到系统的应用程序都必有一个数字证书,此数字证书用于标识应用程序的作者和在应用程序之间建立信任关系 , 如果一个 permission 的 protectionLevel 为 signature,那么就只有那些跟该 permission 所在的程序拥有同一个数字证书的应用程序才能取得该权限。Android 使用 Java 的数字证书相关的机制来给 apk 加盖数字证书,要理解 android 的数字证书,需要先了解以下数字证书的概念和 java 的数字证书机制。Android 系统要求每一个安装进系统的应用程序都是经过数字证书签名的,数字证书的私钥则保存在程序开发者的手中。Android 将数字证书用来标识应用程序的作者和在应用程序之间建立信任关系,不是用来决定最终用户可以安装哪些应用程序。这个数字证书并不需要权威的数字证书签名机构认证,它只是用来让应用程序包自我认证的 ;也是用来判断该应用是否被别人破解,二次打包的一个标准,但是签名并不能防止被破解。

[0050] 如图 2 所示,提供一个前述加密方法的具体应用的实施例,从该实施例可更加直观了解本发明方法具体实施中的一种方式。

[0051] 进一步的,根据本发明前述的加密方法的原理,可以推得解密方法 :关于所述核心部分的动态加载,经过加密(或称加固)的应用程序软件安装到 Android 系统中后,需要调用被隐藏的核心部分的时候,需要对其进行动态分离、解密和加载,这个过程是与前述加密过程一一对应的。整个动态加载的实现流程如下 :

[0052] (1) 主体部分从自身的 apk 文件中读取 classes.dex 文件,在 classes.dex 文件尾部得到加密数据的长度,根据加密数据长度计算出加密数据的起始位置,从而读取得到加密数据。

[0053] (2) 运行解密方法,解密得到核心功能部分 apk。

[0054] (3) 通过 Android API 提供的 DexClassLoader 类,对核心部分代码进行反射调用,从而实现核心功能部分代码的动态加载。

[0055] (4) 调用完成后，删除核心功能部分 apk 文件，从而避免核心功能部分代码暴露在系统内部存储之中，被攻击者得到。

[0056] 如图 3 所示，与前述方法原理类似的，本发明提供一种安卓系统中应用程序加密系统 1，由于前述加密方法中的技术特征均可应用于此系统实施例中，因此不做重复赘述；所述系统 1 包括：划分模块 11，用于将应用程序文件内容划分，划分后的内容包括：主体部分及核心部分；提取模块 12，用于从所述主体部分提取虚拟机运行程序文件；加密模块 13，用于将所述核心部分进行加密以形成加密文件；修改模块 14，用于修改所述虚拟机运行程序文件，包括：将加密文件写入所述虚拟机运行程序文件尾部，并在所述尾部添加所述加密文件的描述信息，并更新所述虚拟机运行程序文件的属性信息；文件生成模块 15，用于将修改后的虚拟机运行程序文件放回所述主体部分中，来生成完成所述加密后的应用程序文件。

[0057] 在一实施例中，所述主体部分及核心部分均编译为独立的 apk 文件。

[0058] 在一实施例中，所述虚拟机运行程序文件为 classes.dex 文件。

[0059] 在一实施例中，所述描述信息包括：加密文件的名称、位置及长度。

[0060] 在一实施例中，所述更新属性信息包括：重新计算所述虚拟机运行程序文件所包含的：校验和字段、签名字段和文件大小字段的值并替换原有值。

[0061] 在一实施例中，所述的安卓系统中应用程序加密系统，还包括：签名模块，用于通过安卓软件开发包中提供的签名工具对所述加密后的应用程序文件进行签名。

[0062] 如上所述，本发明提供的安卓系统中应用程序加密方法及系统，将应用程序文件内容划分，划分后的内容包括：主体部分及核心部分；从所述主体部分提取虚拟机运行程序文件；将所述核心部分进行加密以形成加密文件；修改所述虚拟机运行程序文件，包括：将加密文件写入所述虚拟机运行程序文件尾部，并在所述尾部添加所述加密文件的描述信息，并更新所述虚拟机运行程序文件的属性信息；将修改后的虚拟机运行程序文件放回所述主体部分中，来生成完成所述加密后的应用程序文件；通过本发明的技术方案，逆向工程所获得的代码中仍然是只有软件主体部分代码，核心功能部分的代码被隐藏可以有效对抗各种针对安卓系统中应用软件的逆向工程攻击，提高安全性。

[0063] 上述实施例仅例示性说明本发明的原理及其功效，而非用于限制本发明。任何熟悉此技术的人士皆可在不违背本发明的精神及范畴下，对上述实施例进行修饰或改变。因此，举凡所属技术领域中具有通常知识者在未脱离本发明所揭示的精神与技术思想下所完成的一切等效修饰或改变，仍应由本发明的权利要求所涵盖。

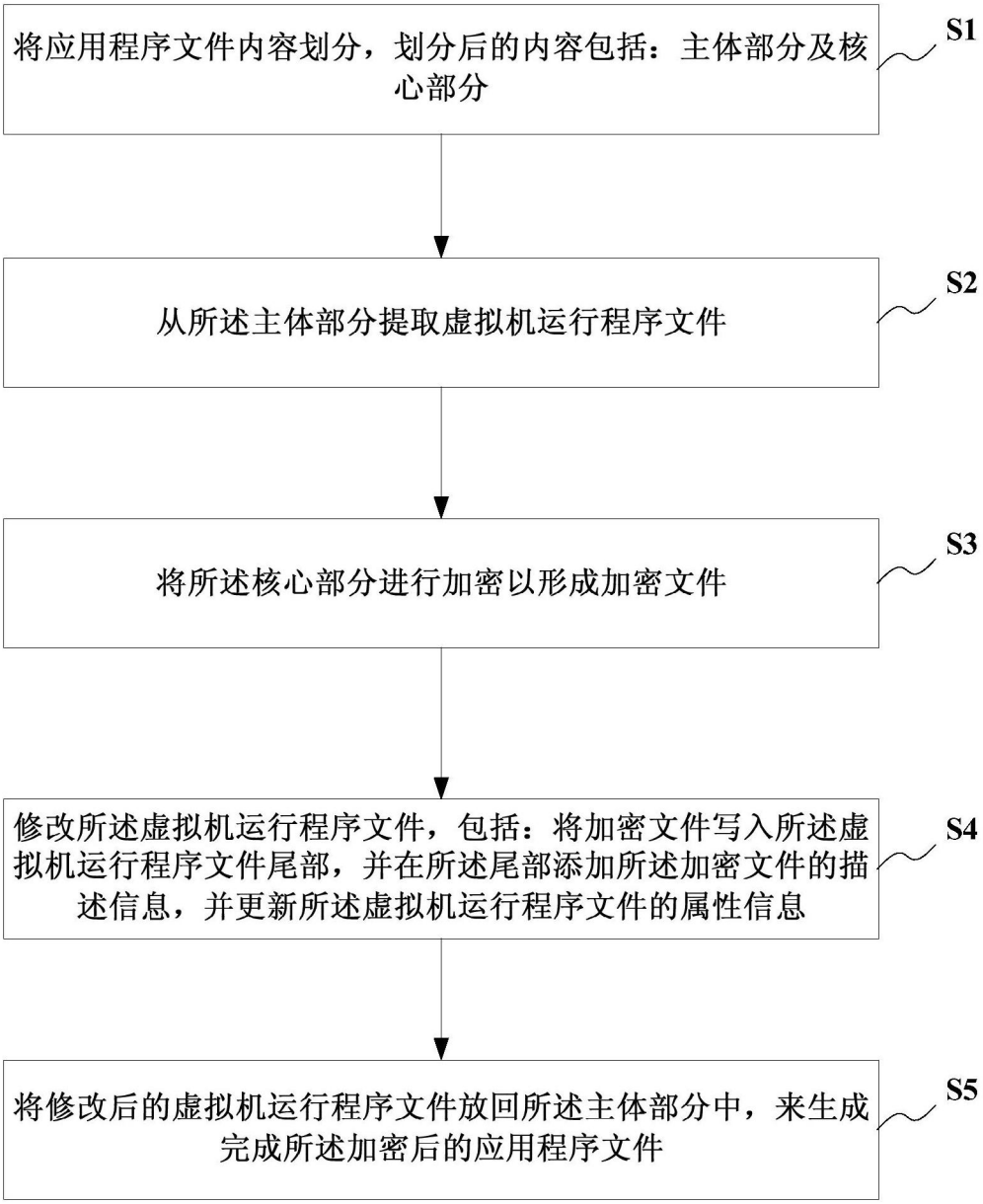

将应用程序文件内容划分，划分后的内容包括：主体部分及核心部分
S1
从所述主体部分提取虚拟机运行程序文件
S2
将所述核心部分进行加密以形成加密文件
S3
修改所述虚拟机运行程序文件，包括：将加密文件写入所述虚拟机运行程序文件尾部，并在所述尾部添加所述加密文件的描述信息，并更新所述虚拟机运行程序文件的属性信息
S4
将修改后的虚拟机运行程序文件放回所述主体部分中，来生成完成所述加密后的应用程序文件
S5

图 1

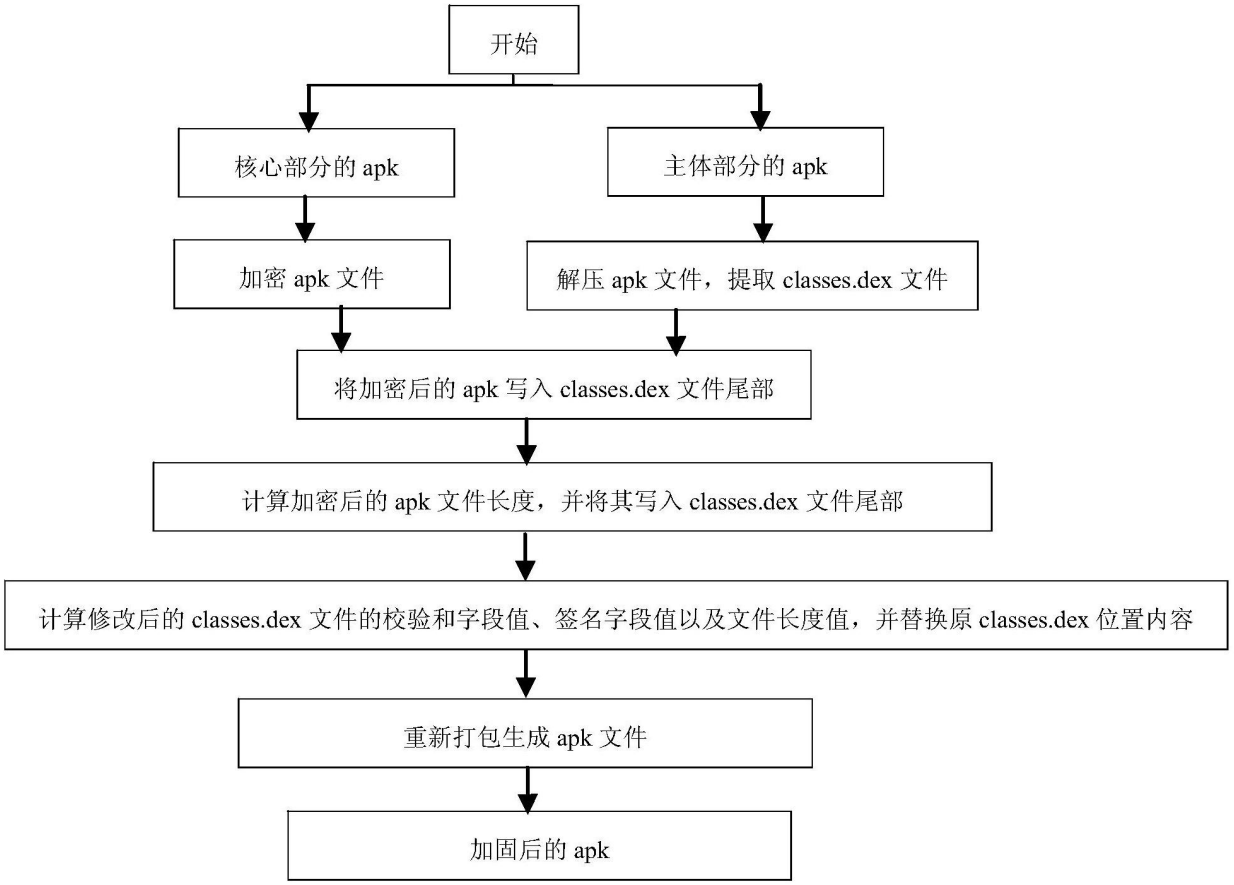

开始
核心部分的 apk
主体部分的 apk
加密 apk 文件
解压 apk 文件，提取 classes.dex 文件
将加密后的 apk 写入 classes.dex 文件尾部
计算加密后的 apk 文件长度，并将其写入 classes.dex 文件尾部
计算修改后的 classes.dex 文件的校验和字段值、签名字段值以及文件长度值，并替换原 classes.dex 位置内容
重新打包生成 apk 文件
加固后的 apk

图 2

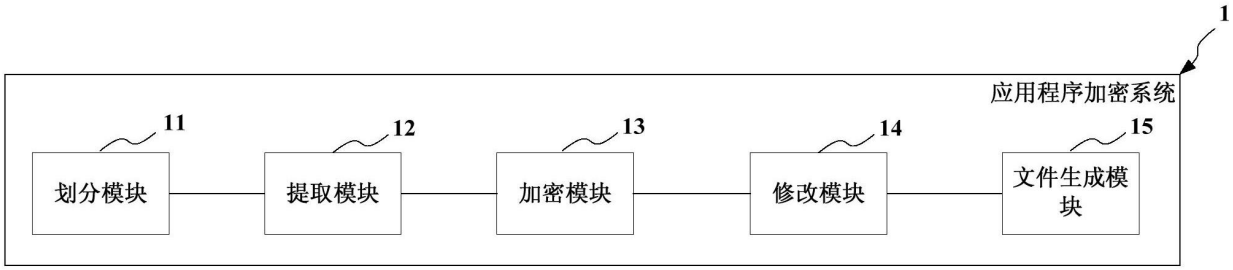

1
应用程序加密系统
11
12
13
14
15
划分模块
提取模块
加密模块
修改模块
文件生成模块

图 3