(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 106295371 B
(45)授权公告日 2019.10.25

(21)申请号 201610698778.3

(22)申请日 2016.08.22

(65)同一申请的已公布的文献号
申请公布号 CN 106295371 A

(43)申请公布日 2017.01.04

(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 张帆 陈春荣 周玲玲 张洪睿

(74)专利代理机构 北京三高永信知识产权代理有限责任公司 11138
代理人 朱雅男

(51)Int.Cl.
G06F 21/60(2013.01)

(56)对比文件
CN 104021321 A,2014.09.03,
CN 103914637 A,2014.07.09,
CN 102831338 A,2012.12.19,

审查员 张剑峰

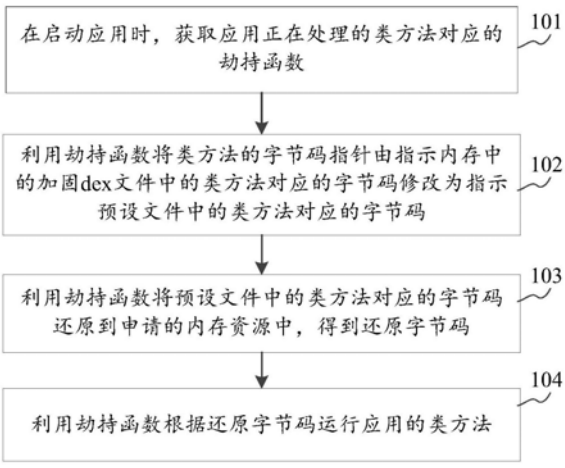
权利要求书5页 说明书12页 附图6页

(54)发明名称

应用运行方法、文件加固方法和装置

(57)摘要

本发明公开了一种应用运行类方法、文件加固类方法和装置，属于计算机技术领域。该方法包括：在启动应用时，获取应用正在处理的类方法对应的劫持函数；利用劫持函数将类方法的字节码指针由指示内存中的加固dex文件中的类方法对应的字节码修改为指示预设文件中的类方法对应的字节码；利用劫持函数将预设文件中的类方法对应的字节码还原到申请的内存资源中，得到还原字节码；利用劫持函数根据还原字节码运行应用的类方法，解决了在终端将还原字节码还原到内存中的加固dex文件中时，恶意人员可以从该加固dex文件中获取到还原字节码，从而恶意篡改该还原字节码，导致终端运行应用不安全的问题，达到了提高终端运行应用的安全性的效果。



CN 106295371 B

1.一种应用运行方法,其特征在于,所述方法包括:

在启动应用时,获取所述应用正在处理的类方法对应的劫持函数,所述类方法是所述应用包括的一段代码,所述劫持函数用于修改所述类方法的字节码指针,所述字节码指针用于指示所述类方法对应的字节码;

利用所述劫持函数将所述类方法的字节码指针由指示内存中的加固dex文件中的所述类方法对应的字节码修改为指示预设文件中的所述类方法对应的字节码,所述内存中的加固dex文件是终端运行所述应用的安卓安装包APK中的加固dex文件得到的,所述预设文件是所述APK中的文件,且所述预设文件中的所述类方法对应的字节码是从所述APK中的加固dex文件中转移得到的;

利用所述劫持函数将所述预设文件中的所述类方法对应的字节码还原到申请的内存资源中,得到还原字节码;

利用所述劫持函数根据所述还原字节码运行所述应用的所述类方法。

2.根据权利要求1所述的方法,其特征在于,所述在启动应用时,获取所述应用正在处理的类方法对应的劫持函数,包括:

在启动应用时,检测系统支持的安全策略,根据所述系统支持的安全策略调用n种劫持函数,所述安全策略用于从所述n种劫持函数中确定所述类方法所对应的一个劫持函数,所述n为正整数;

对于所述n种劫持函数中的第m种劫持函数,当所述第m种劫持函数获取到所述类方法时,获取所述类方法对应的安全策略;根据预设的劫持函数与安全策略的对应关系,检测所述第m种劫持函数对应的安全策略是否与所述类方法对应的安全策略相同;在所述第m种劫持函数对应的安全策略与所述类方法对应的安全策略相同时,确定所述第m种劫持函数为所述类方法对应的劫持函数,触发执行所述利用所述劫持函数将所述类方法的字节码指针由指示内存中的加固dex文件中的所述类方法对应的字节码修改为指示预设文件中的所述类方法对应的字节码的步骤;在所述第m种劫持函数对应的安全策略与所述类方法对应的安全策略相同时,将m更新为m+1,继续执行所述当所述第m种劫持函数获取到所述类方法时,获取所述类方法对应的安全策略的步骤,其中,所述第m种劫持函数在第m+1种劫持函数之前获取到所述类方法,所述m为小于所述n的正整数。

3.根据权利要求2所述的方法,其特征在于,所述安全策略包括一级安全策略、二级安全策略和三级安全策略,所述劫持函数包括加载劫持函数、预执行劫持函数和执行劫持函数,所述加载劫持函数在所述预执行劫持函数之前获取到所述类方法,所述预执行函数在所述执行劫持函数之前获取到所述类方法,所述检测系统支持的安全策略,根据所述系统支持的安全策略调用n种劫持函数,包括:

检测所述系统是否支持所述一级安全策略、所述二级安全策略和所述三级安全策略中的至少一种;

当所述系统支持所述一级安全策略时,调用所述一级安全策略对应的所述加载劫持函数,所述加载劫持函数用于在加载所述类方法时修改所述类方法的字节码指针;

当所述系统支持所述二级安全策略时,调用所述二级安全策略对应的所述预执行劫持函数,所述预执行劫持函数用于在预执行所述类方法时修改所述类方法的字节码指针;

当所述系统支持所述三级安全策略时,调用所述三级安全策略对应的所述执行劫持函

数,所述执行劫持函数用于在执行所述类方法时修改所述类方法的字节码指针。

4.根据权利要求3所述的方法,其特征在于,当所述类方法对应的安全策略为所述三级安全策略时,在所述利用所述劫持函数根据所述还原字节码运行所述应用的所述类方法之后,所述方法还包括:

利用所述劫持函数将所述类方法的字节码指针由指示所述预设文件中的所述类方法对应的字节码修改为指示所述内存中的加固dex文件中的所述类方法对应的字节码;

利用所述劫持函数删除所述内存资源中的所述还原字节码。

5.根据权利要求3所述的方法,其特征在于,当所述系统不支持所述一级安全策略、所述二级安全策略和所述三级安全策略中的任意一种时,所述方法还包括:

调用类劫持函数,所述类劫持函数用于获取所述应用正在处理的类,所述类包括至少一个类方法;

利用所述类劫持函数将所述预设文件中所述类的所有类方法对应的字节码复制到所述内存中的加固dex文件中,还原所述内存中的加固dex文件中的字节码得到所述还原字节码;根据所述还原字节码运行所述应用的所述类。

6.根据权利要求3所述的方法,其特征在于,所述获取所述类方法对应的安全策略,包括:

获取所述应用的安全策略文件,所述安全策略文件包括所述应用中的每个类和每个类对应的安全策略,每个类包括至少一个类方法;

当所述系统支持所述一级安全策略、所述二级安全策略和所述三级安全策略中的至少一种,且所述系统支持的安全策略包括所述安全策略文件指示的所述类方法所属的类对应的安全策略时,将所述类方法所属的类对应的安全策略确定为所述类方法对应的安全策略;或者,

当所述系统支持所述一级安全策略、所述二级安全策略和所述三级安全策略中的任意一种或两种,且所述系统支持的安全策略不包括所述安全策略文件指示的所述类方法所属的类对应的安全策略时,将系统支持的安全策略中的一种确定为所述类方法对应的安全策略。

7.根据权利要求6所述的方法,其特征在于,所述获取所述应用的安全策略文件,包括:

在启动所述应用时向服务器发送安全策略请求指令,所述服务器根据所述安全策略请求指令下发所述安全策略文件;

获取所述服务器下发的所述安全策略文件。

8.根据权利要求6所述的方法,其特征在于,所述获取所述应用的安全策略文件,包括:

从所述APK中读取所述安全策略文件。

9.一种文件加固方法,其特征在于,所述方法包括:

根据应用包括的每个类对应的安全策略生成安全策略文件,所述安全策略用于供终端从至少一个劫持函数中确定所述终端正在处理的类方法所对应的一个劫持函数;

将所述应用的安卓安装包APK中的加固dex文件中的字节码转移至预设文件;

将所述安全策略文件和所述预设文件插入所述应用的所述APK中,以使所述终端在启动所述应用时执行权利要求1-8中任一项所述的方法。

10.根据权利要求9所述的方法,其特征在于,所述方法还包括:

接收终端发送的安全策略请求指令；

根据所述安全策略请求指令向所述终端发送所述安全策略文件。

11.一种应用运行装置，其特征在于，所述装置包括：

获取模块，用于在启动应用时，获取所述应用正在处理的类方法对应的劫持函数，所述类方法是所述应用包括的一段代码，所述劫持函数用于修改所述类方法的字节码指针，所述字节码指针用于指示所述类方法对应的字节码；

第一修改模块，用于利用所述获取模块获取的所述劫持函数将所述类方法的字节码指针由指示内存中的加固dex文件中的所述类方法对应的字节码修改为指示预设文件中的所述类方法对应的字节码，所述内存中的加固dex文件是终端运行所述应用的安卓安装包APK中的加固dex文件得到的，所述预设文件是所述APK中的文件，且所述预设文件中的所述类方法对应的字节码是从所述APK中的加固dex文件中转移得到的；

还原模块，用于利用所述劫持函数将所述第一修改模块得到的所述预设文件中的所述类方法对应的字节码还原到申请的内存资源中，得到还原字节码；

第一运行模块，用于利用所述劫持函数根据所述还原模块得到的所述还原字节码运行所述应用的所述类方法。

12.根据权利要求11所述的装置，其特征在于，所述获取模块，包括：

调用单元，用于在启动应用时，检测系统支持的安全策略，根据所述系统支持的安全策略调用n种劫持函数，所述安全策略用于从所述n种劫持函数中确定所述类方法所对应的一个劫持函数，所述n为正整数；

获取单元，用于对于所述n种劫持函数中的第m种劫持函数，当所述第m种劫持函数获取到所述类方法时，获取所述类方法对应的安全策略；

检测单元，用于根据预设的劫持函数与安全策略的对应关系，检测所述调用单元调用的所述第m种劫持函数对应的安全策略是否与所述获取单元获取的所述类方法对应的安全策略相同；

确定单元，用于在所述第m种劫持函数对应的安全策略与所述类方法对应的安全策略相同时，确定所述第m种劫持函数为所述类方法对应的劫持函数，触发执行所述利用所述劫持函数将所述类方法的字节码指针由指示内存中的加固dex文件中的所述类方法对应的字节码修改为指示预设文件中的所述类方法对应的字节码的步骤；在所述第m种劫持函数对应的安全策略与所述类方法对应的安全策略相同时，将m更新为m+1，继续执行所述当所述第m种劫持函数获取到所述类方法时，获取所述类方法对应的安全策略的步骤，其中，所述第m种劫持函数在第m+1种劫持函数之前获取到所述类方法，所述m为小于所述n的正整数。

13.根据权利要求12所述的装置，其特征在于，所述安全策略包括一级安全策略、二级安全策略和三级安全策略，所述劫持函数包括加载劫持函数、预执行劫持函数和执行劫持函数，所述加载劫持函数在所述预执行劫持函数之前获取到所述类方法，所述预执行函数在所述执行劫持函数之前获取到所述类方法，所述调用单元，包括：

检测子单元，用于检测所述系统是否支持所述一级安全策略、所述二级安全策略和所述三级安全策略中的至少一种；

第一调用子单元，用于当所述检测子单元检测出所述系统支持所述一级安全策略时，调用所述一级安全策略对应的所述加载劫持函数，所述加载劫持函数用于在加载所述类方

法时修改所述类方法的字节码指针；

第二调用子单元，用于当所述检测子单元检测出所述系统支持所述二级安全策略时，调用所述二级安全策略对应的所述预执行劫持函数，所述预执行劫持函数用于在预执行所述类方法时修改所述类方法的字节码指针；

第三调用子单元，用于当所述检测子单元检测出所述系统支持所述三级安全策略时，调用所述三级安全策略对应的所述执行劫持函数，所述执行劫持函数用于在执行所述类方法时修改所述类方法的字节码指针。

14.根据权利要求13所述的装置，其特征在于，当所述类方法对应的安全策略为所述三级安全策略时，在所述利用所述劫持函数根据所述还原字节码运行所述应用的所述类方法之后，所述装置还包括：

第二修改模块，用于利用所述劫持函数将所述类方法的字节码指针由指示所述预设文件中的所述类方法对应的字节码修改为指示所述内存中的加固dex文件中的所述类方法对应的字节码；

删除模块，用于利用所述劫持函数删除所述内存资源中的所述还原字节码。

15.根据权利要求13所述的装置，其特征在于，当所述系统不支持所述一级安全策略、所述二级安全策略和所述三级安全策略中的任意一种时，所述装置还包括：

调用模块，用于调用类劫持函数，所述类劫持函数用于获取所述应用正在处理的类，所述类包括至少一个类方法；

第二运行模块，用于利用所述调用模块调用的所述类劫持函数将所述预设文件中所述类的所有类方法对应的字节码复制到所述内存中的加固dex文件中，还原所述内存中的加固dex文件中的字节码得到所述还原字节码；根据所述还原字节码运行所述应用的所述类。

16.根据权利要求13所述的装置，其特征在于，所述获取单元，包括：

第一获取子单元，用于获取所述应用的安全策略文件，所述安全策略文件包括所述应用中的每个类和每个类对应的安全策略，每个类包括至少一个类方法；

第一确定子单元，用于当所述系统支持所述一级安全策略、所述二级安全策略和所述三级安全策略中的至少一种，且所述系统支持的安全策略包括所述获取子单元获取的所述安全策略文件指示的所述类方法所属的类对应的安全策略时，将所述类方法所属的类对应的安全策略确定为所述类方法对应的安全策略；或者，

第二确定子单元，用于当所述系统支持所述一级安全策略、所述二级安全策略和所述三级安全策略中的任意一种或两种，且所述系统支持的安全策略不包括所述获取子单元获取的所述安全策略文件指示的所述类方法所属的类对应的安全策略时，将系统支持的安全策略中的一种确定为所述类方法对应的安全策略。

17.根据权利要求16所述的装置，其特征在于，所述第一获取子单元，包括：

发送子单元，用于在启动所述应用时向服务器发送安全策略请求指令，所述服务器根据所述安全策略请求指令下发所述安全策略文件；

第二获取子单元，用于获取所述服务器下发的所述安全策略文件。

18.根据权利要求16所述的装置，其特征在于，所述第一获取子单元，包括：

读取子单元，用于从所述APK中读取所述安全策略文件。

19.一种文件加固装置，其特征在于，所述装置包括：

生成模块，用于根据应用包括的每个类对应的安全策略生成安全策略文件，所述安全策略用于供终端从至少一个劫持函数中确定所述终端正在处理的类方法所对应的一个劫持函数；

转移模块，用于将所述应用的安卓安装包APK中的加固dex文件中的字节码转移至预设文件；

插入模块，用于将所述生成模块生成的所述安全策略文件和所述转移模块得到的所述预设文件插入所述应用的所述APK中，以使所述终端在启动所述应用时执行权利要求1-8中任一项所述的方法。

20.根据权利要求19所述的装置，其特征在于，所述装置还包括：

接收模块，用于接收终端发送的安全策略请求指令；

发送模块，用于根据所述接收模块接收的所述安全策略请求指令向所述终端发送所述安全策略文件。

应用运行方法、文件加固方法和装置

技术领域

[0001] 本发明涉及计算机技术领域，特别涉及一种应用运行方法、文件加固方法和装置。

背景技术

[0002] 安装有Android(安卓)系统的终端通过安装应用的加固APK(Android Package，安卓安装包)来执行应用。加固APK括加固dex文件，该加固dex文件(Dalvik Executable File，Dalvik虚拟机的可执行文件)是指后缀名为dex的文件，且该加固dex文件包括该应用包括的每个类对应的字节码。终端在应用启动时，在内存中运行该加固dex文件以还原该字节码得到还原字节码，通过执行该还原字节码来运行该应用。

[0003] 终端在启动应用时，会加载运行该应用时所需的类，每个类包括至少一种类方法。终端劫持该应用正在加载的类，将加固dex文件中该类包括的所有类方法对应的字节码还原到内存中的加固dex文件中，得到该类包括的所有类方法对应的还原字节码，继续加载该类，在加载完成后根据该类包括的所有类方法对应的还原字节码运行该应用。

[0004] 由于终端将类包括的所有类方法对应的字节码还原到内存中的加固dex文件中，这样，在终端从加载应用运行时所需的类开始，所有类方法对应的还原字节码一直暴露在内存中的加固dex文件中，此时，所有类方法对应的还原字节码很容易被获取和篡改，导致终端运行该应用的安全性不高。

发明内容

[0005] 为了解决相关技术中终端运行应用的安全性不高的问题，本发明实施例提供了一种应用运行方法、文件加固方法和装置。技术方案如下：

[0006] 第一方面，提供了一种应用运行方法，该方法包括：

[0007] 在启动应用时，获取应用正在处理的类方法对应的劫持函数，类方法是应用包括的一段代码，劫持函数用于修改类方法的字节码指针，字节码指针用于指示类方法对应的字节码；

[0008] 利用劫持函数将类方法的字节码指针由指示内存中的加固dex文件中的类方法对应的字节码修改为指示预设文件中的类方法对应的字节码，内存中的加固dex文件是终端运行应用的APK中的加固dex文件得到的，预设文件是APK中的文件，且预设文件中的类方法对应的字节码是从APK中的加固dex文件中转移得到的；

[0009] 利用劫持函数将预设文件中的类方法对应的字节码还原到申请的内存资源中，得到还原字节码；

[0010] 利用劫持函数根据还原字节码运行应用的类方法。

[0011] 第二方面，提供了一种文件加固方法，该方法包括：

[0012] 根据应用包括的每个类对应的安全策略生成安全策略文件，该安全策略用于供终端从至少一个劫持函数中确定终端正在处理的类方法所对应的一个劫持函数；

[0013] 将应用的APK中的加固dex文件中的字节码转移至预设文件；

[0014] 将安全策略文件和预设文件插入应用的APK中。

[0015] 第三方面，提供了一种应用运行装置，该装置包括：

[0016] 获取模块，用于在启动应用时，获取应用正在处理的类方法对应的劫持函数，类方法是应用包括的一段代码，该劫持函数用于修改类方法的字节码指针，该字节码指针用于指示类方法对应的字节码；

[0017] 第一修改模块，用于利用获取模块获取的劫持函数将类方法的字节码指针由指示内存中的加固dex文件中的类方法对应的字节码修改为指示预设文件中的类方法对应的字节码，内存中的加固dex文件是终端运行应用的APK中的加固dex文件得到的，预设文件是APK中的文件，且预设文件中的类方法对应的字节码是从APK中的加固dex文件中转移得到的；

[0018] 还原模块，用于利用劫持函数将第一修改模块得到的预设文件中的类方法对应的字节码还原到申请的内存资源中，得到还原字节码；

[0019] 第一运行模块，用于利用劫持函数根据还原模块得到的还原字节码运行应用的类方法。

[0020] 第四方面，提供了一种文件加固装置，该装置包括：

[0021] 生成模块，用于根据应用包括的每个类对应的安全策略生成安全策略文件，该安全策略用于供终端从至少一个劫持函数中确定终端正在处理的类方法所对应的一个劫持函数；

[0022] 转移模块，用于将应用的APK中的加固dex文件中的字节码转移至预设文件；

[0023] 插入模块，用于将生成模块生成的安全策略文件和转移模块得到的预设文件插入应用的APK中。

[0024] 本发明实施例提供的技术方案带来的有益效果是：通过利用劫持函数将终端正在处理的类方法的字节码指针由内存中的加固dex文件中的该类方法对应的字节码修改为指示预设文件中的该类方法对应的字节码，将该预设文件中的该类方法的对应的字节码还原到申请的内存资源中，而不是还原到内存中的加固dex文件中，使得恶意人员无法从内存中的加固dex文件中获得还原字节码，也无法从该应用的APK中的加固dex文件获取还原字节码，解决了在终端将还原字节码还原到内存中的加固dex文件中时，恶意人员可以从该加固dex文件中获取到还原字节码，从而恶意篡改该还原字节码，导致终端运行应用不安全的问题，达到了提高终端运行应用的安全性的效果。

[0025] 另外，通过将APK中的加固dex文件中的字节码转移至预设文件，使得恶意人员无法从该加固dex文件中获取到字节码，从而对字节码进行反编译得到还原字节码，提高了应用中字节码的安全性。

附图说明

[0026] 为了更清楚地说明本发明实施例中的技术方案，下面将对实施例描述中所需要使用的附图作简单地介绍，显而易见地，下面描述中的附图仅仅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。

[0027] 图1A是本发明一个实施例中提供的应用运行方法的方法流程图；

[0028] 图1B是本发明一个实施例中提供的文件加固方法的方法流程图；

[0029] 图2是本发明另一个实施例中提供的应用运行方法的方法流程图；

[0030] 图3是本发明一个实施例中提供的根据一级安全策略运行应用的示意图；

[0031] 图4是本发明一个实施例中提供的根据二级安全策略运行应用的示意图；

[0032] 图5A是本发明一个实施例中提供的根据三级安全策略运行应用的示意图；

[0033] 图5B是本发明一个实施例中提供的另一种根据三级安全策略运行应用的示意图；

[0034] 图6是本发明另一个实施例中提供的文件加固方法的方法流程图；

[0035] 图7是本发明一个实施例中提供的应用运行装置的框图；

[0036] 图8是本发明一个实施例中提供的文件加固装置的框图。

具体实施方式

[0037] 为使本发明的目的、技术方案和优点更加清楚，下面将结合附图对本发明实施方式作进一步地详细描述。

[0038] 请参见图1A，其示出了本发明实施例提供的一种应用运行方法的流程图。本实施例以该应用运行方法用于终端中为例进行说明，该方法至少包括以下步骤。

[0039] 步骤101，在启动应用时，获取应用正在处理的类方法对应的劫持函数。

[0040] 类方法是应用包括的一段代码，劫持函数用于修改类方法的字节码指针，字节码指针用于指示类方法对应的字节码。

[0041] 步骤102，利用劫持函数将类方法的字节码指针由指示内存中的加固dex文件中的类方法对应的字节码修改为指示预设文件中的类方法对应的字节码。

[0042] 内存中的加固dex文件是终端运行应用的APK中的加固dex文件得到的，预设文件是APK中的文件，且预设文件中的类方法对应的字节码是从APK中的加固dex文件中转移得到的。

[0043] 需要说明的，应用的APK可以为加固APK，也可以为未加固的APK，本实施例不作限定。本文中以该APK为加固APK为例进行说明。

[0044] 步骤103，利用劫持函数将预设文件中的类方法对应的字节码还原到申请的内存资源中，得到还原字节码，。

[0045] 步骤104，利用劫持函数根据还原字节码运行应用的类方法。

[0046] 综上所述，本发明实施例提供的应用运行方法，通过利用劫持函数将终端正在处理的类方法的字节码指针由内存中的加固dex文件中的该类方法对应的字节码修改为指示预设文件中的该类方法对应的字节码，将该预设文件中的该类方法的对应的字节码还原到申请的内存资源中，而不是还原到内存中的加固dex文件中，使得恶意人员无法从内存中的加固dex文件中获得还原字节码，也无法从该应用的APK中的加固dex文件获取还原字节码，解决了在终端将还原字节码还原到内存中的加固dex文件中时，恶意人员可以从该加固dex文件中获取到还原字节码，从而恶意篡改该还原字节码，导致终端运行应用不安全的问题，达到了提高终端运行应用的安全性的效果。

[0047] 在终端执行图1A所述的应用运行方法之前，需要从服务器下载该应用的APK，该应用的APK会由服务器进行加固，以使得终端可以安全运行该应用。请参见图1B，其示出了本发明实施例提供的一种文件加固方法的流程图。本实施例以该应用运行方法用于服务器中

为例进行说明，该方法至少包括以下步骤。

[0048] 步骤106，根据应用包括的每个类对应的安全策略生成安全策略文件。

[0049] 该安全策略用于供终端从至少一个劫持函数中确定该终端正在处理的类方法所对应的一个劫持函数。

[0050] 步骤107，将应用的APK中的加固dex文件中的字节码转移至预设文件。

[0051] 步骤108，将安全策略文件和预设文件插入应用的APK中。

[0052] 综上所述，本发明实施例提供的应用运行方法，通过将APK中的加固dex文件中的字节码转移至预设文件，使得恶意人员无法从该加固dex文件中获取到字节码，从而对字节码进行反编译得到还原字节码，提高了应用中字节码的安全性。

[0053] 请参见图2，其示出了本发明实施例提供的一种应用运行方法的方法流程图。本实施例以该应用运行方法用于终端中为例进行说明，该方法至少包括以下步骤。

[0054] 步骤201，在启动应用时，检测系统支持的安全策略，根据系统支持的安全策略调用n种劫持函数，n为正整数。

[0055] 在启动应用时，终端会从服务器下载的该应用的APK中获取并执行代理dex文件，通过该代理dex文件启动终端中的安全动态库，再通过该安全动态库来确定系统支持的安全策略。其中，安全动态库是指APK中后缀名为so的文件，且该安全动态库包括n种劫持函数和每种劫持函数对应的安全策略。

[0056] n种劫持函数至少包括加载劫持函数、预执行劫持函数和执行劫持函数，加载劫持函数用于在加载类方法时修改该类方法的字节码指针，预执行劫持函数用于在预执行该类方法时修改该类方法的字节码指针，执行劫持函数用于在执行该类方法时修改该类方法的字节码指针，字节码指针用于指示类方法对应的字节码。当终端调用了该n种劫持函数时，加载劫持函数会在预执行劫持函数之前获取到终端正在处理的类方法，预执行函数会在执行劫持函数之前获取到该终端正在处理的类方法。其中，类方法是指应用的APK中的运行该应用所需的每个类中的一段代码，此处的类方法也可以被称为函数，本实施不对类方法的名称作限定。

[0057] 安全策略用于从终端调用的n种劫持函数中确定该终端正在处理的类方法所对应的一个劫持函数。其中，该安全策略至少包括一级安全策略、二级安全策略和三级安全策略。

[0058] 本实施例中，一级安全策略与加载劫持函数对应，二级安全策略与预执行劫持函数对应，三级安全策略与执行劫持函数对应。由于安全动态库中的n种劫持函数获取到终端正在处理的类方法的时机越早，该类方法对应的还原字节码暴露在内存中的时间越长，该还原字节码越不安全，因此，对于保证还原字节码的安全性来说，一级安全策略的安全等级低于二级安全策略的安全等级，二级安全策略的安全等级低于三级安全策略的安全等级。其中，还原字节码是将待处理类方法对应的字节码还原到内存后得到的，且应用在运行时需要使用该还原字节码。

[0059] 其中，安全动态库确定系统支持的安全策略，根据系统支持的安全策略调用n种劫持函数，包括：检测系统是否支持一级安全策略、二级安全策略和三级安全策略中的至少一种；当系统支持一级安全策略时，调用一级安全策略对应的加载劫持函数；当系统支持二级安全策略时，调用二级安全策略对应的预执行劫持函数；当系统支持三级安全策略时，调用

三级安全策略对应的执行劫持函数。

[0060] 安全动态库在检测系统是否支持一级安全策略、二级安全策略和三级安全策略时采用的检测方法相同，下文以安全动态库检测系统是否支持一级安全策略为例进行说明。安全动态库检测当前系统是否包括执行该一级安全策略所需使用数据，在当前系统包括执行该一级安全策略所需使用的数据时，确定当前系统支持该一级安全策略；在当前系统不包括执行该一级安全策略所需使用的数据时，确定当前系统不支持该一级安全策略。其中，该数据可以包括执行对应的级别的安全策略所需使用的变量、符号、函数等，本实施例不作限定。

[0061] 需要说明的是，不同级安全策略所需使用的数据的参数类型或者参数值不同。比如，一级安全策略所需使用的数据包括变量和符号；二级安全策略所需使用的数据包括函数；或者，一级安全策略和二级安全策略所需使用的数据都包括变量，且一级安全策略对应的数据的变量名称和二级安全策略对应的数据的变量名称不同。

[0062] 当检测出该系统不支持一级安全策略、二级安全策略和三级安全策略中的任意一种时，执行步骤202；当检测出当前的系统支持一级安全策略、二级安全策略和三级安全策略中的至少一种时，执行步骤204。

[0063] 步骤202，当检测出系统不支持一级安全策略、二级安全策略和三级安全策略中的任意一种时，调用类劫持函数。

[0064] 类劫持函数用于获取应用正在处理的类，该类包括至少一个类方法。

[0065] 当安全动态库检测出系统不支持一级安全策略、二级安全策略和三级安全策略中的任意一种时，通过安全动态库将虚拟机中的加载类函数修改为加载类劫持指令，该加载类劫持指令使得终端中的虚拟机在利用加载类函数加载正在处理的类时，调用加载类劫持函数获取该类。虚拟机可以为Dalvik虚拟机、也可以为ART(Android Runtime)虚拟机，本实施例不作限定。

[0066] 步骤203，利用类劫持函数将预设文件中类的所有类方法对应的字节码复制到内存中的加固dex文件中，还原内存中的加固dex文件中的字节码得到还原字节码；根据还原字节码运行应用的类，流程结束。

[0067] 其中，预设文件是开发者预设在服务器中的，用于存储加固dex文件中的字节码，且预设文件中的字节码是从APK中的加固dex文件中转移得到的，本实施例不对该预设文件的类型作限定。

[0068] 步骤204，当检测出当前的系统支持一级安全策略、二级安全策略和三级安全策略中的至少一种时，获取应用正在处理的类方法对应的劫持函数。

[0069] 本实施例中，虚拟机处理类方法包括以下处理阶段：加载类方法、预执行类方法、执行类方法。在加载类方法时利用加载函数加载该类方法，在预执行类方法时利用预执行函数预执行该类方法，在执行类方法时利用执行函数执行该类方法。

[0070] 安全动态库在检测出系统支持一级安全策略、二级安全策略和三级安全策略中的至少一种时，会根据每种系统支持的安全策略对应的劫持函数，将虚拟机中对应的函数修改为劫持指令，该劫持指令使得对应的劫持函数获取到终端正在处理的类方法。

[0071] 假设系统支持一级安全策略，则安全动态库会将虚拟机中的加载函数修改为与加载劫持函数对应的加载劫持指令，虚拟机在利用加载劫持指令函数加载类方法时，加载劫

持指令将该类方法劫持到加载劫持函数，使得加载劫持函数获取到该类方法。

[0072] 又假设系统支持二级安全策略，则安全动态库会将虚拟机中的预执行函数修改为与预执行劫持函数对应的预执行劫持指令，虚拟机在利用预执行劫持指令函数预执行类方法时，预执行劫持指令将该类方法劫持到预执行劫持函数，使得预执行劫持函数获取到该类方法。

[0073] 又假设系统支持三级安全策略，则安全动态库会将虚拟机中的执行函数修改为与执行劫持函数对应的执行劫持指令，虚拟机在利用执行劫持指令函数执行类方法时，执行劫持指令将该类方法劫持到执行劫持函数，使得执行劫持函数获取到该类方法。

[0074] 对于n种劫持函数中的第m种劫持函数，当第m种劫持函数获取到类方法时，获取类方法对应的安全策略；根据预设的劫持函数与安全策略的对应关系，检测第m种劫持函数对应的安全策略是否与类方法对应的安全策略相同；在第m种劫持函数对应的安全策略与类方法对应的安全策略相同时，确定第m种劫持函数为类方法对应的劫持函数；在第m种劫持函数对应的安全策略与类方法对应的安全策略相同时，将m更新为m+1，继续执行当第m种劫持函数获取到类方法时，获取类方法对应的安全策略的步骤，直至确定出类方法对应的劫持函数时停止。其中，第m种劫持函数在第m+1种劫持函数之前获取到类方法，m为小于n的正整数。

[0075] 假设终端支持一级安全策略、二级安全策略和三级安全策略，调用了加载劫持函数、预执行劫持函数和执行劫持函数，那么加载劫持函数会优先接收到终端正在处理的类方法，并检测该类方法对应的安全策略是否为一级安全策略，在该类方法对应的安全策略是一级安全策略时，确定该加载劫持函数为该类方法对应的劫持函数；在该类方法对应的安全策略不是一级安全策略时，该加载劫持函数将该类方法返回虚拟机，使得该虚拟机继续处理该类方法，在该虚拟机预执行该类方法时，预执行劫持函数会获取到该类方法，检测该类方法对应的安全策略是否为二级安全策略，在该类方法对应的安全策略是二级安全策略时，确定该预执行劫持函数为该类方法对应的劫持函数；在该类方法对应的安全策略不是二级安全策略时，继续执行上述检测步骤。

[0076] 安全动态库获取类方法对应的安全策略，包括：获取应用的安全策略文件，该安全策略文件包括应用中的每个类和每个类对应的安全策略，每个类包括至少一个类方法；当系统支持一级安全策略、二级安全策略和三级安全策略中的至少一种，且系统支持的安全策略包括安全策略文件指示的类方法所属的类对应的安全策略时，将类方法所属的类对应的安全策略确定为类方法对应的安全策略；或者，当系统支持一级安全策略、二级安全策略和三级安全策略中的任意一种或两种，且系统支持的安全策略不包括安全策略文件指示的类方法所属的类对应的安全策略时，将系统支持的安全策略中的一种确定为类方法对应的安全策略。

[0077] 在安全策略文件中，不同的类可以对应相同的安全策略，不同的类也可以对应不同的安全策略，本实施例不作限定。

[0078] 可选的，当系统支持的安全策略不包括安全策略文件指示的类方法所属的类对应的安全策略时，安全动态库将该类方法对应的安全策略确定为系统支持的最高安全等级的安全策略。

[0079] 假设安全动态库获取到的安全策略文件如下表一所示，终端正在处理的类方法所

属的类为Class3，则根据表一所示的安全策略文件可知，该待处理类方法对应的安全策略为三级安全策略，当系统支持三级安全策略时，该类方法对应的安全策略即为三级安全策略，当系统不支持三级安全策略时，该类方法对应的安全策略即为二级安全策略。

[0080] 表一：

[0081]

类	安全策略
Class1	一级安全策略
Class2	一级安全策略
Class3	三级安全策略
Class4	二级安全策略

[0082] 安全动态库获取应用的安全策略文件包括但不限于下述实现方式。

[0083] 在一种实现方式中，安全动态库直接从应用的APK中读取安全策略文件。此时，由于APK中的安全策略文件不会改变，因此，即使某个类对应的最优的安全策略不是该安全策略文件中该类所对应的安全策略时，终端也不会更改该类对应的安全策略，从而无法确定出最优的该类中的类方法对应的劫持函数。

[0084] 在另一种实现方式中，终端在启动应用时，安全动态库向服务器发送安全策略请求指令，该服务器根据安全策略请求指令下发安全策略文件；该安全动态库获取服务器下发的安全策略文件。

[0085] 其中，服务器中的安全策略文件是由开发者在获得每个类当前对应的最优的安全策略后，对安全策略文件进行更新后上传到服务器中的。

[0086] 由于在每次启动应用时，安全动态库都从服务器处重新获取一份安全策略文件，使得应用包括的每个类对应的安全策略都是最优的，提高了终端确定正在处理的类方法对应的劫持函数的准确性。

[0087] 步骤205，利用类方法对应的劫持函数将该类方法的字节码指针由指示内存中的加固dex文件中的类方法对应的字节码修改为指示预设文件中的类方法对应的字节码。

[0088] 劫持函数在处理类方法时，会生成该类方法的方法结构体，该方法结构体包括该类方法对应的字节码指针。

[0089] 假设类方法对应的劫持函数为加载加持函数时，请参考图3，虚拟机中的加载劫持指令使得类方法被劫持到安全动态库中的加载劫持函数中，安全动态库中的加载劫持函数将该类方法的字节码指针的指示对象由内存中的加固dex文件中的类方法对应的字节码修改为预设文件中的类方法对应的字节码。

[0090] 假设类方法对应的劫持函数为预执行劫持函数时，请参考图4，虚拟机中的预执行劫持指令使得类方法被劫持到安全动态库中的预执行劫持函数中，安全动态库中的预执行劫持函数将该类方法的字节码指针的指示对象由内存中的加固dex文件中的类方法对应的字节码修改为预设文件中的类方法对应的字节码。

[0091] 假设类方法对应的劫持函数为执行劫持函数时，请参考图5A，虚拟机中的执行劫持指令使得类方法被劫持到安全动态库中的预执行劫持函数中，安全动态库中的执行劫持函数将该类方法的字节码指针的指示对象由内存中的加固dex文件中的类方法对应的字节码修改为预设文件中的类方法对应的字节码。

[0092] 步骤206，利用劫持函数将预设文件中的类方法对应的字节码还原到申请的内存资源中，得到还原字节码。

[0093] 其中，终端将还原字节码还原到劫持函数申请的内存资源中，使得恶意人员无法从内存中的加固dex文件中获取还原字节码，提高了应用运行过程中的安全性。

[0094] 步骤207，利用劫持函数根据该还原字节码运行应用的类方法。

[0095] 可选的，当类方法对应的劫持函数为执行劫持函数时，执行劫持函数在运行该类方法后，会将该类方法的字节码指针的指示对象由预设文件中的类方法对应的字节码恢复为内存中的加固dex文件中的类方法对应的字节码，并删除内存资源中的还原字节码。这样，还原字节码只有在安全动态库真正执行该类方法期间才会暴露在内存中，提高了应用运行的安全性，请参考图5B。

[0096] 综上所述，本发明实施例提供的应用运行方法，通过利用劫持函数将终端正在处理的类方法的字节码指针由内存中的加固dex文件中的该类方法对应的字节码修改为指示预设文件中的该类方法对应的字节码，将该预设文件中的该类方法的对应的字节码还原到申请的内存资源中，而不是还原到内存中的加固dex文件中，使得恶意人员无法从内存中的加固dex文件中获得还原字节码，也无法从该应用的APK中的加固dex文件获取还原字节码，解决了在终端将还原字节码还原到内存中的加固dex文件中时，恶意人员可以从该加固dex文件中获取到还原字节码，从而恶意篡改该还原字节码，导致终端运行应用不安全的问题，达到了提高终端运行应用的安全性的效果。

[0097] 另外，通过在终端每次启动应用时，都从服务器处重新获取一份安全策略文件，使得应用包括的每个类对应的安全策略可以根据开发者上传到服务器中的更新的安全策略文件而变化，由于更新的安全策略文件通常为每个类对应的最优的安全策略，提高了终端中的安全动态库确定每个正在处理的类方法对应的劫持函数的准确性。

[0098] 另外，通过检测系统支持的安全策略，使得终端在系统不支持待处理类方法所属的类对应的安全策略时，修改该类对应的安全策略，提高了终端执行每个类的灵活性。

[0099] 在终端执行图2所述的应用运行类方法之前，需要从服务器下载该应用的APK，该应用的APK会由服务器进行加固，以使得终端可以安全运行该应用。请参见图6，其示出了本发明实施例提供的一种文件加固类方法的类方法流程图。本实施例以该文件加固类方法用于服务器中为例进行说明，该类方法至少包括以下步骤。

[0100] 步骤601，根据应用包括的每个类对应的安全策略生成安全策略文件。

[0101] 该安全策略用于供终端从至少一个劫持函数中确定该终端正在处理的类方法所对应的一个劫持函数。

[0102] 每个类对应的安全策略是由开发者上传到服务器中的，也可以是服务器中默认的，本实施例不作限定。

[0103] 步骤602，将应用的APK中的加固dex文件中的字节码转移至预设文件。

[0104] 通过将加固dex文件中的字节码转移至预设文件，这样加固dex文件不包括该应用对应的字节码，使得恶意人员无法从该加固dex文件中获取到字节码，从而对字节码进行反编译得到还原字节码，提高了应用中字节码的安全性。

[0105] 步骤603，将安全策略文件和预设文件插入应用的APK中。

[0106] 可选的，当终端首次启动应用之前，会向服务器发送用于请求该应用的APK的请求

指令，服务器根据该请求指令向终端发送该APK。终端从该APK中获取安全策略文件和预设文件，根据安全策略文件和预设文件执行图2所示的应用运行类方法。

[0107] 步骤604，接收终端发送的安全策略请求指令。

[0108] 可选的，在本步骤之前，服务器还会接收到开发者上传的更新的安全策略文件。

[0109] 步骤605，根据安全策略请求指令向终端发送安全策略文件。

[0110] 可选的，终端每次启动应用时，都会向服务器发送安全策略请求指令，服务器根据该安全策略请求指令向终端发送安全策略文件。当服务器在上次接收到安全策略请求指令之后，本次接收到安全策略请求指令之前，接收到开发者上传的更新的安全策略文件，则服务器根据该安全策略请求指令向终端发送该更新的安全策略文件。

[0111] 综上所述，本实施例提供的文件加固类方法，通过将APK中的加固dex文件中的字节码转移至预设文件，使得恶意人员无法从该加固dex文件中获取到字节码，从而对字节码进行反编译得到还原字节码，提高了应用中字节码的安全性。

[0112] 请参见图7，其示出了本发明实施例提供的一种应用运行装置的框图。本实施例以该应用运行装置用于终端中为例进行说明，该装置包括：

[0113] 获取模块710，用于在启动应用时，获取应用正在处理的类方法对应的劫持函数，类方法是应用包括的一段代码，劫持函数用于修改类方法的字节码指针，字节码指针用于指示类方法对应的字节码；

[0114] 第一修改模块720，用于利用获取模块710获取的劫持函数将类方法的字节码指针由指示内存中的加固dex文件中的类方法对应的字节码修改为指示预设文件中的类方法对应的字节码，内存中的加固dex文件是终端运行应用的安卓安装包APK中的加固dex文件得到的，预设文件是APK中的文件，且预设文件中的类方法对应的字节码是从APK中的加固dex文件中转移得到的；

[0115] 还原模块730，用于利用劫持函数将第一修改模块720得到的预设文件中的类方法对应的字节码还原到申请的内存资源中，得到还原字节码；

[0116] 第一运行模块740，用于利用劫持函数根据还原模块730得到的还原字节码运行应用的类方法。

[0117] 可选的，获取模块，包括：

[0118] 调用单元，用于在启动应用时，检测系统支持的安全策略，根据系统支持的安全策略调用n种劫持函数，安全策略用于从至少一个劫持函数中确定类方法所对应的一个劫持函数，n为正整数；

[0119] 获取单元，用于对于n种劫持函数中的第m种劫持函数，当第m种劫持函数获取到类方法时，获取类方法对应的安全策略；

[0120] 检测单元，用于根据预设的劫持函数与安全策略的对应关系，检测调用单元调用的第m种劫持函数对应的安全策略是否与获取单元获取的类方法对应的安全策略相同；

[0121] 确定单元，用于在第m种劫持函数对应的安全策略与类方法对应的安全策略相同时，确定第m种劫持函数为类方法对应的劫持函数，触发执行利用劫持函数将类方法的字节码指针由指示内存中的加固dex文件中的类方法对应的字节码修改为指示预设文件中的类方法对应的字节码的步骤；在第m种劫持函数对应的安全策略与类方法对应的安全策略相同时，将m更新为m+1，继续执行当第m种劫持函数获取到类方法时，获取类方法对应的安全

策略的步骤，其中，第m种劫持函数在第m+1种劫持函数之前获取到类方法，m为小于n的正整数。

[0122] 可选的，安全策略包括一级安全策略、二级安全策略和三级安全策略，劫持函数包括加载劫持函数、预执行劫持函数和执行劫持函数，加载劫持函数在预执行劫持函数之前获取到类方法，预执行函数在执行劫持函数之前获取到类方法，调用单元，包括：

[0123] 检测子单元，用于检测系统是否支持一级安全策略、二级安全策略和三级安全策略中的至少一种；

[0124] 第一调用子单元，用于当检测子单元检测出系统支持一级安全策略时，调用一级安全策略对应的加载劫持函数，加载劫持函数用于在加载类方法时修改类方法的字节码指针；

[0125] 第二调用子单元，用于当检测子单元检测出系统支持二级安全策略时，调用二级安全策略对应的预执行劫持函数，预执行劫持函数用于在预执行类方法时修改类方法的字节码指针；

[0126] 第三调用子单元，用于当检测子单元检测出系统支持三级安全策略时，调用三级安全策略对应的执行劫持函数，执行劫持函数用于在执行类方法时修改类方法的字节码指针。

[0127] 可选的，当类方法对应的安全策略为三级安全策略时，类方法对应的劫持函数为执行劫持函数，在利用劫持函数根据还原字节码运行应用的类方法之后，装置还包括：

[0128] 第二修改模块，用于利用劫持函数将类方法的字节码指针由指示预设文件中的类方法对应的字节码修改为指示内存中的加固dex文件中的类方法对应的字节码；

[0129] 删除模块，用于利用劫持函数删除内存资源中的还原字节码。

[0130] 可选的，当系统不支持一级安全策略、二级安全策略和三级安全策略中的任意一种时，该装置还包括：

[0131] 调用模块，用于调用类劫持函数，类劫持函数用于获取应用正在处理的类，类包括至少一个类方法；

[0132] 第二运行模块，用于利用调用模块调用的类劫持函数将预设文件中类的所有类方法对应的字节码复制到内存中的加固dex文件中，还原内存中的加固dex文件中的字节码得到还原字节码；根据还原字节码运行应用的类。

[0133] 可选的，获取单元，包括：

[0134] 第一获取子单元，用于获取应用的安全策略文件，安全策略文件包括应用中的每个类和每个类对应的安全策略，每个类包括至少一个类方法；

[0135] 第一确定子单元，用于当系统支持一级安全策略、二级安全策略和三级安全策略中的至少一种，且系统支持的安全策略包括获取子单元获取的安全策略文件指示的类方法所属的类对应的安全策略时，将类方法所属的类对应的安全策略确定为类方法对应的安全策略；或者，

[0136] 第二确定子单元，用于当系统支持一级安全策略、二级安全策略和三级安全策略中的任意一种或两种，且系统支持的安全策略不包括获取子单元获取的安全策略文件指示的类方法所属的类对应的安全策略时，将系统支持的安全策略中的一种确定为类方法对应的安全策略。

[0137] 可选的，第一获取子单元，包括：

[0138] 发送子单元，用于在启动应用时向服务器发送安全策略请求指令，服务器根据安全策略请求指令下发安全策略文件；

[0139] 第二获取子单元，用于获取服务器下发的安全策略文件。

[0140] 可选的，第一获取子单元，包括：

[0141] 读取子单元，用于从APK中读取安全策略文件。

[0142] 综上所述，本发明实施例提供的应用运行装置，通过利用劫持函数将终端正在处理的类方法的字节码指针由内存中的加固dex文件中的该类方法对应的字节码修改为指示预设文件中的该类方法对应的字节码，将该预设文件中的该类方法的对应的字节码还原到申请的内存资源中，而不是还原到内存中的加固dex文件中，使得恶意人员无法从内存中的加固dex文件中获得还原字节码，也无法从该应用的APK中的加固dex文件获取还原字节码，解决了在终端将还原字节码还原到内存中的加固dex文件中时，恶意人员可以从该加固dex文件中获取到还原字节码，从而恶意篡改该还原字节码，导致终端运行应用不安全的问题，达到了提高终端运行应用的安全性的效果。

[0143] 另外，通过在终端每次启动应用时，都从服务器处重新获取一份安全策略文件，使得应用包括的每个类对应的安全策略可以根据开发者上传到服务器中的更新的安全策略文件而变化，由于更新的安全策略文件通常为每个类对应的最优的安全策略，提高了终端中的安全动态库确定每个正在处理的类方法对应的劫持函数的准确性。

[0144] 另外，通过检测系统支持的安全策略，使得终端在系统不支持待处理类方法所属的类对应的安全策略时，修改该类对应的安全策略，提高了终端执行每个类的灵活性。

[0145] 请参见图8，其示出了本发明实施例提供的一种文件加固装置的框图。本实施例以该文件加固装置用于终端中为例进行说明，该装置包括：

[0146] 生成模块810，用于根据应用包括的每个类对应的安全策略生成安全策略文件，安全策略用于供终端从至少一个劫持函数中确定终端正在处理的类方法所对应的一个劫持函数；

[0147] 转移模块820，用于将应用的安卓安装包APK中的加固dex文件中的字节码转移至预设文件；

[0148] 插入模块830，用于将生成模块810生成的安全策略文件和转移模块820得到的预设文件插入应用的APK中。

[0149] 可选的，该装置还包括：

[0150] 接收模块，用于接收终端发送的安全策略请求指令；

[0151] 发送模块，用于根据接收模块接收到的安全策略请求指令向终端发送安全策略文件。

[0152] 综上所述，本发明实施例提供的文件加固装置，通过将APK中的加固dex文件中的字节码转移至预设文件，使得恶意人员无法从该加固dex文件中获取到字节码，从而对字节码进行反编译得到还原字节码，提高了应用中字节码的安全性。

[0153] 上述本发明实施例序号仅仅为了描述，不代表实施例的优劣。

[0154] 本领域普通技术人员可以理解实现上述实施例的全部或部分步骤可以通过硬件来完成，也可以通过程序来指令相关的硬件完成，所述的程序可以存储于一种计算机可读

存储介质中，上述提到的存储介质可以是只读存储器，磁盘或光盘等。

[0155] 以上所述仅为本发明的较佳实施例，并不用以限制本发明，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保护范围之内。

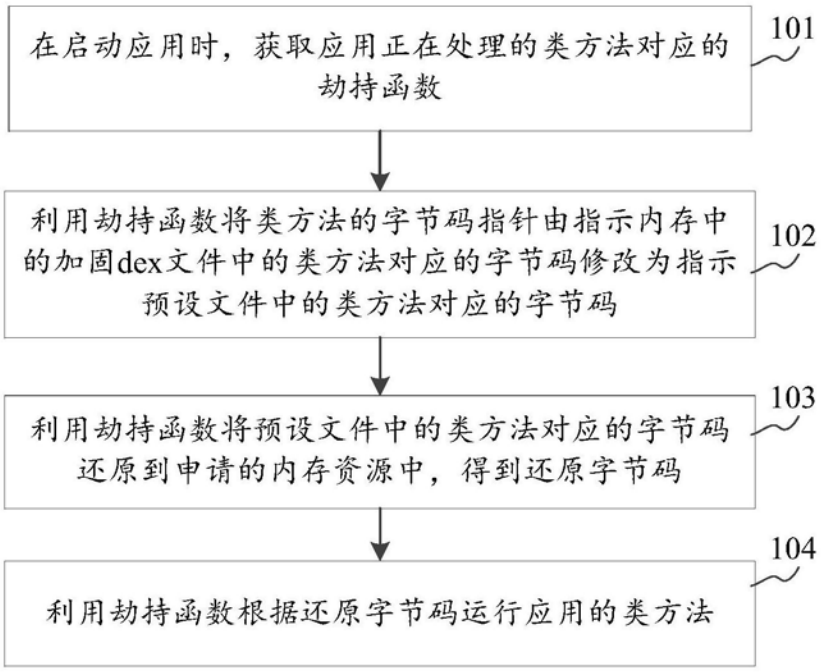

在启动应用时，获取应用正在处理的类方法对应的劫持函数
101
利用劫持函数将类方法的字节码指针由指示内存中的加固dex文件中的类方法对应的字节码修改为指示预设文件中的类方法对应的字节码
102
利用劫持函数将预设文件中的类方法对应的字节码还原到申请的内存资源中，得到还原字节码
103
利用劫持函数根据还原字节码运行应用的类方法
104

图1A

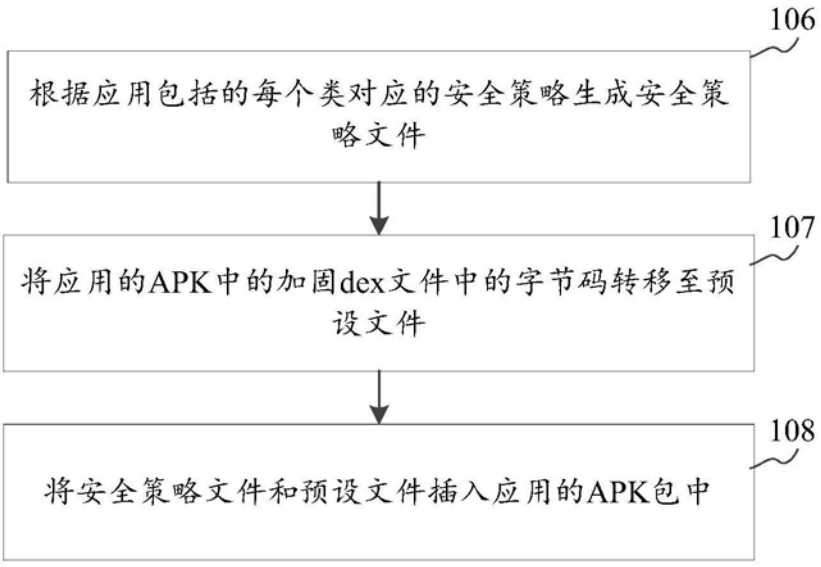

根据应用包括的每个类对应的安全策略生成安全策略文件
106
将应用的APK中的加固dex文件中的字节码转移至预设文件
107
将安全策略文件和预设文件插入应用的APK包中
108

图1B

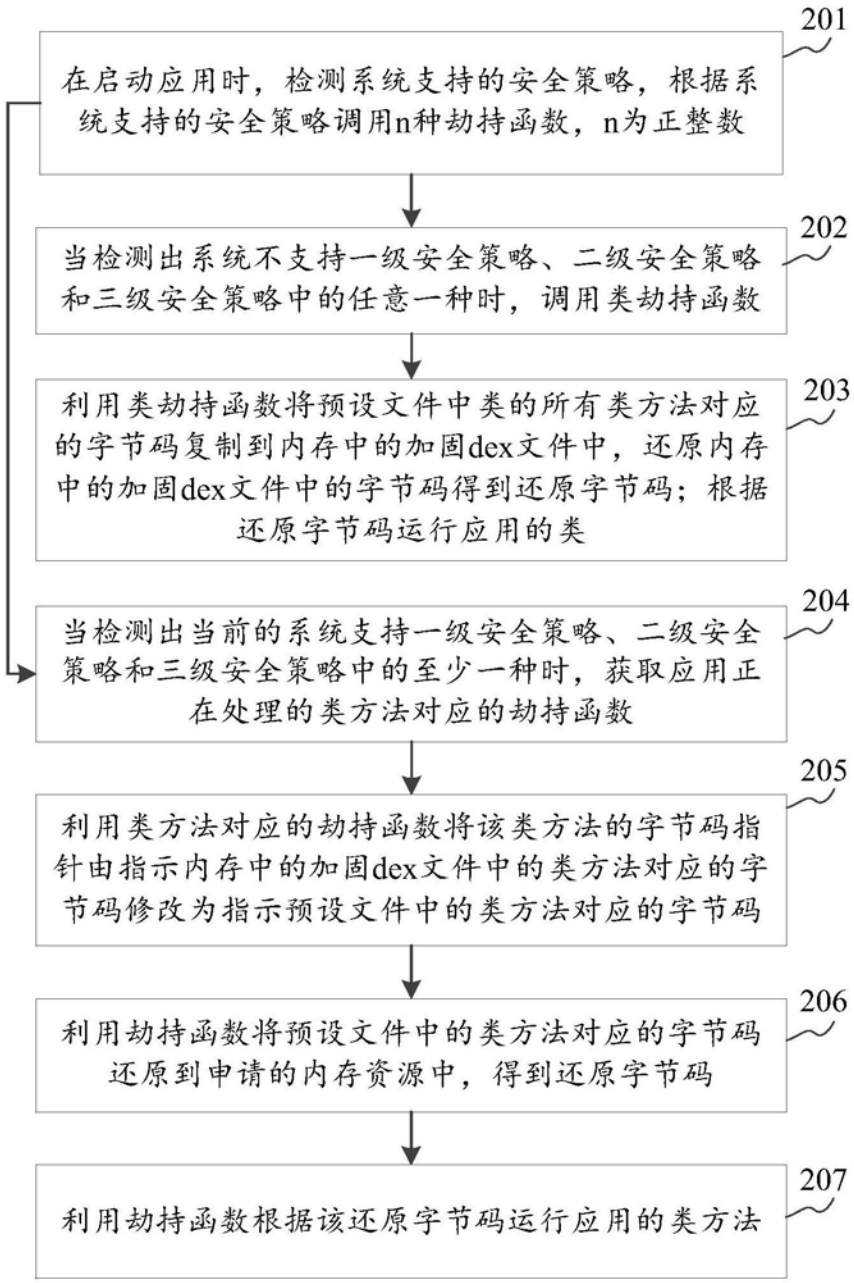

201
在启动应用时，检测系统支持的安全策略，根据系统支持的安全策略调用n种劫持函数，n为正整数
202
当检测出系统不支持一级安全策略、二级安全策略和三级安全策略中的任意一种时，调用类劫持函数
203
利用类劫持函数将预设文件中类的所有类方法对应的字节码复制到内存中的加固dex文件中，还原内存中的加固dex文件中的字节码得到还原字节码；根据还原字节码运行应用的类
204
当检测出当前的系统支持一级安全策略、二级安全策略和三级安全策略中的至少一种时，获取应用正在处理的类方法对应的劫持函数
205
利用类方法对应的劫持函数将该类方法的字节码指针由指示内存中的加固dex文件中的类方法对应的字节码修改为指示预设文件中的类方法对应的字节码
206
利用劫持函数将预设文件中的类方法对应的字节码还原到申请的内存资源中，得到还原字节码
207
利用劫持函数根据该还原字节码运行应用的类方法

图2

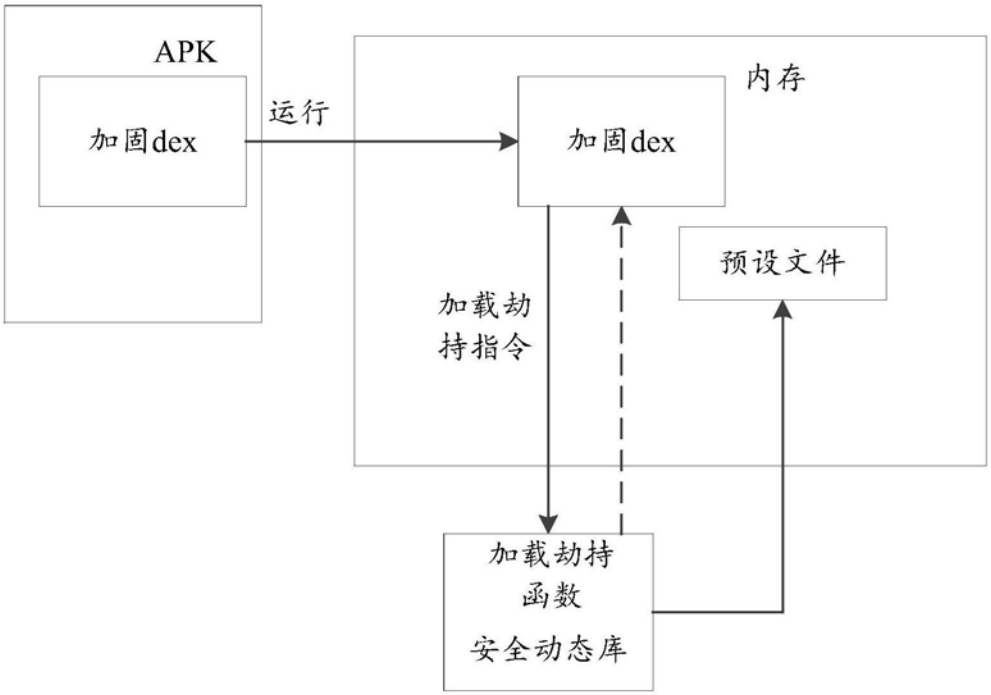

APK
加固dex
运行
内存
加固dex
预设文件
加载劫持指令
加载劫持函数
安全动态库

图3

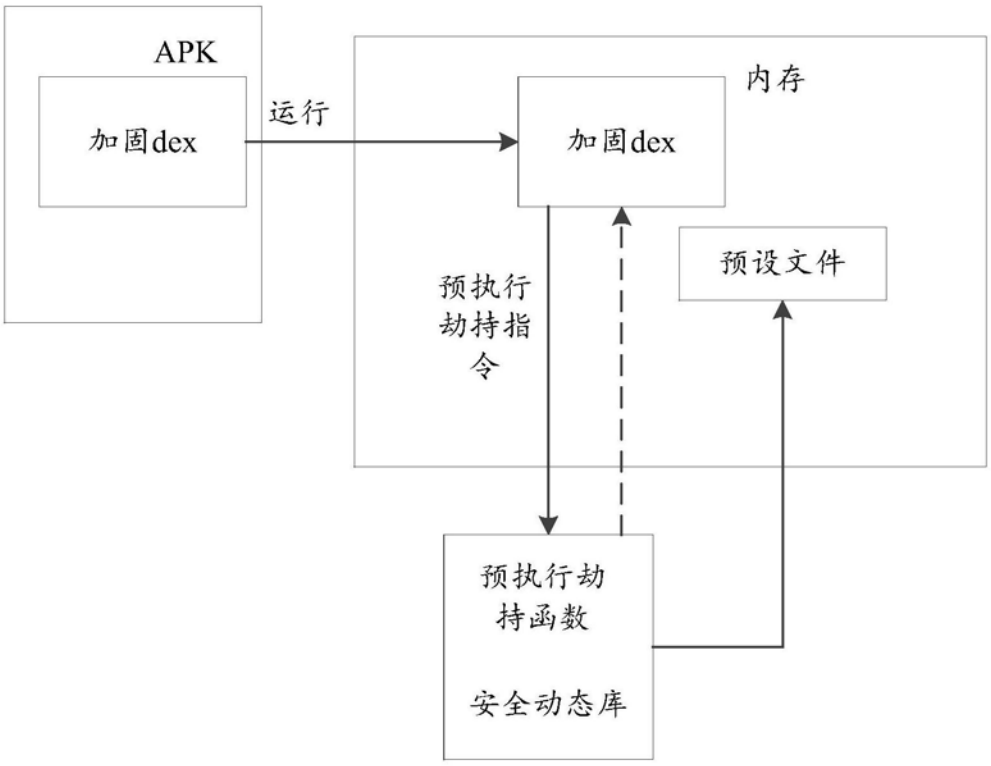

APK
加固dex
运行
内存
加固dex
预设文件
预执行劫持指令
预执行劫持函数
安全动态库

图4

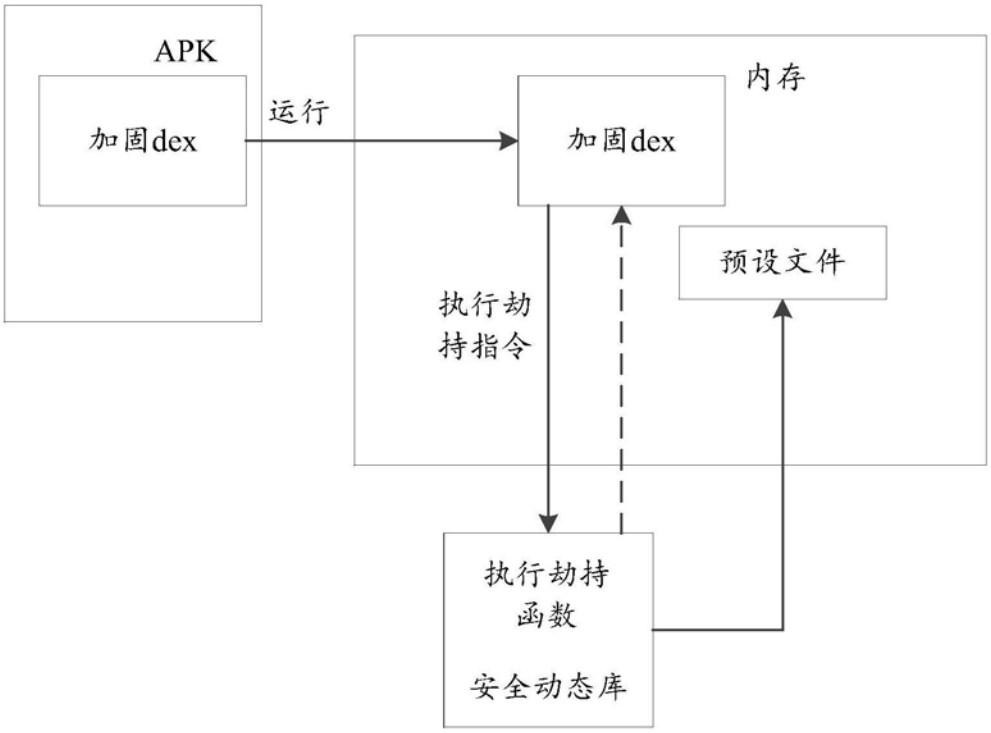

APK
加固dex
运行
内存
加固dex
预设文件
执行劫持指令
执行劫持函数
安全动态库

图5A

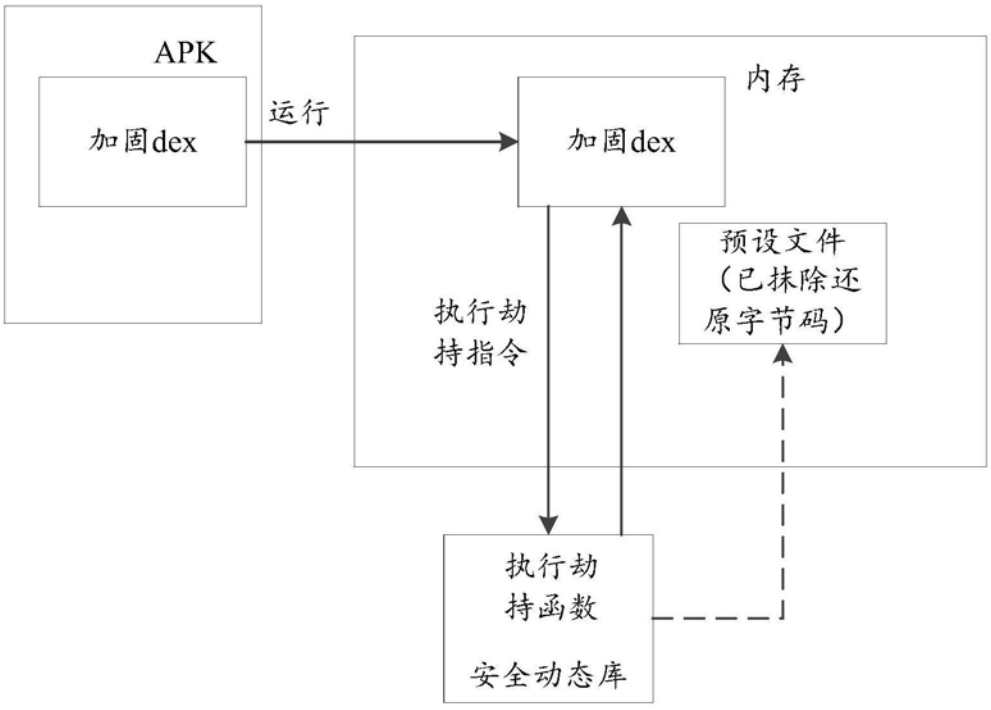

APK
加固dex
运行
内存
加固dex
预设文件（已抹除还原字节码）
执行劫持指令
执行劫持函数
安全动态库

图5B

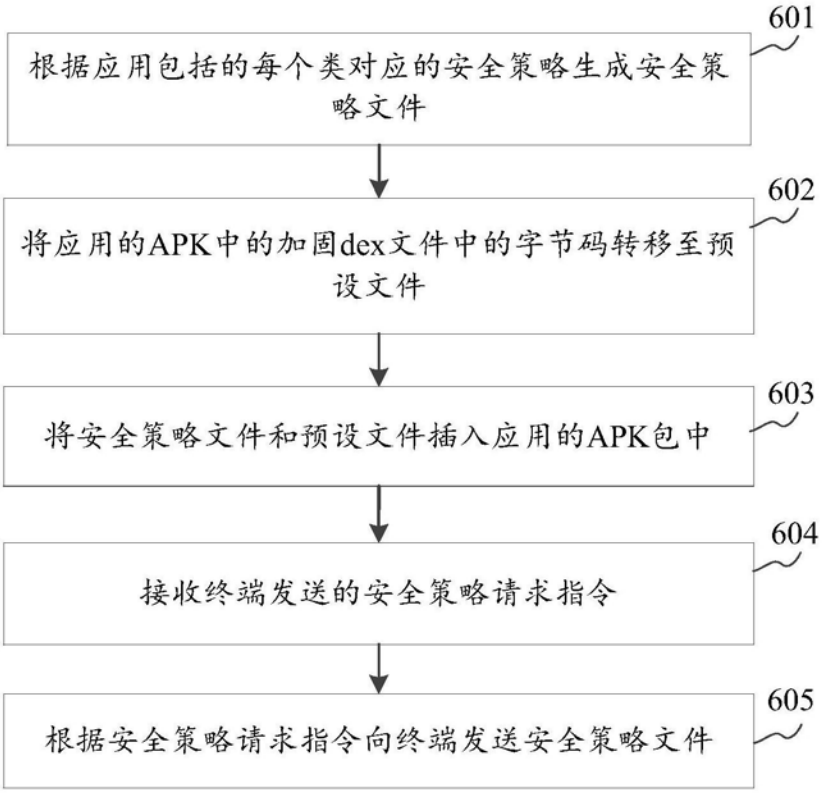

601
根据应用包括的每个类对应的安全策略生成安全策略文件
602
将应用的APK中的加固dex文件中的字节码转移至预设文件
603
将安全策略文件和预设文件插入应用的APK包中
604
接收终端发送的安全策略请求指令
605
根据安全策略请求指令向终端发送安全策略文件

图6

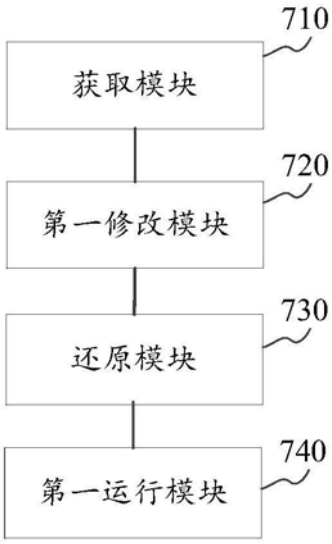

710
获取模块
720
第一修改模块
730
还原模块
740
第一运行模块

图7

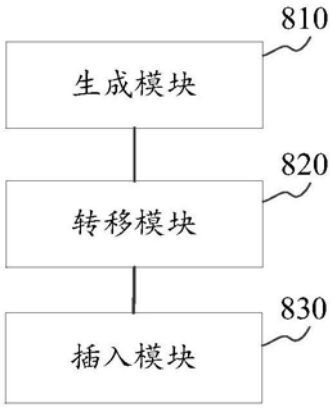


图8