(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 105512552 B
(45)授权公告日 2019.06.07

(21)申请号 201410505220.X

(22)申请日 2014.09.26

(65)同一申请的已公布的文献号
申请公布号 CN 105512552 A

(43)申请公布日 2016.04.20

(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 邱金涛 丁海峰

(74)专利代理机构 北京德琦知识产权代理有限公司 11018
代理人 谢安昆 宋志强

(51)Int.Cl.
G06F 21/55(2013.01)

(56)对比文件
US 2014047544 A1,2014.02.13,
US 2014181974 A1,2014.06.26,
CN 103559440 A,2014.02.05,
CN 103257881 A,2013.08.21,
CN 103778012 A,2014.05.07,
CN 104268472 A,2015.01.07,
Bill Cheng,知乎.“iOS如何判断用户是购买正版的用户还是越狱用户?”.《https://www.zhihu.com/question/20963339/answer/16738125》.2013,

审查员 于萍

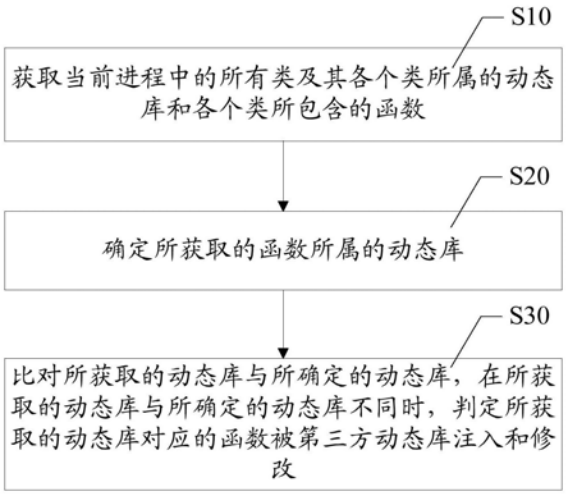
权利要求书3页 说明书9页 附图6页

(54)发明名称

参数检测方法及装置

(57)摘要

本发明公开了一种参数检测方法,所述参数检测方法包括步骤:获取当前进程中的所有类及各个类所属的动态库和各个类所包含的函数;确定所获取的函数所属的动态库;比对所获取的动态库与所确定的动态库,在所获取的动态库与所确定的动态库不同时,判定所获取的动态库对应的函数被第三方动态库注入和/或修改。本发明还公开一种参数检测装置,实现自动检测出当前进程中哪些原有函数被修改,进而保证支付类、账号类的应用处于安全运行环境,提高移动终端使用的安全性。



CN 105512552 B

1.一种参数检测方法,其特征在于,所述参数检测方法包括步骤:

获取当前进程中的所有类和各个类所包含的函数,并根据各个类获取各个类所属的动态库;

确定各个类所包含的函数所属的动态库;

比对所获取的各个类所属的动态库与所确定的各个类所包含的函数所属的动态库,在所获取的一个类所属的动态库与所确定的各个类所包含的函数所属的动态库不同时,判定所获取的该类所属的动态库对应的函数被第三方动态库注入和/或修改;

其中,所述确定各个类所包含的函数所属的动态库的步骤包括:

获取当前进程中加载的所有动态库及其对应的函数地址范围;

确定所获取的所有函数的函数地址;

将所确定的所有函数的函数地址与所获取的所有动态库对应的函数地址范围进行比对,确定所获取的所有函数所属的动态库;

所述获取当前进程中加载的所有动态库对应的函数地址范围的步骤包括:根据当前进程中加载的所有动态库对应的句柄信息得到所述所有动态库对应的函数地址范围。

2.如权利要求1所述的参数检测方法,其特征在于,所述获取当前进程中加载的所有动态库及其对应的函数地址范围的步骤包括:

获取当前进程中加载的所有动态库及所获取的动态库对应的句柄信息;

根据所述句柄信息,得到当前进程中加载的所有动态库的内存起始地址和内存占用信息;

根据所述内存起始地址和内存占用信息得到当前进程中加载的所有动态库对应函数地址范围。

3.如权利要求1所述的参数检测方法,其特征在于,所述根据各个类获取各个类所属的动态库的步骤包括:

获取各个类对应的NSBundle对象的可执行路径信息;

根据所获取的NSBundle对象的可执行路径信息,得到各个类所属的动态库。

4.如权利要求1所述的参数检测方法,其特征在于,所述获取当前进程中的所有类及其各个类所包含的函数的步骤包括:

初始化系统,根据objc_getClassList获取到当前进程中的所有类;

根据class_copyMethodList,获取到各个类所包含的函数。

5.如权利要求1至4任一项所述的参数检测方法,其特征在于,所述比对所获取的各个类所属的动态库与所确定的各个类所包含的函数所属的动态库,在所获取的一个类所属的动态库与所确定的各个类所包含的函数所属的动态库不同时,判定所获取的所述类所属的动态库对应的函数被第三方动态库注入和/或修改的步骤之后,还包括:

确定所获取的所述类所属的动态库中对应的函数;

发出提示信息。

6.一种参数检测装置,其特征在于,所述参数检测装置包括:

处理器,适于实现各指令;以及

存储设备,适于存储多条指令,所述指令适于由处理器加载并执行:

获取当前进程中的所有类和各个类所包含的函数,并根据各个类获取各个类所属的动

态库；

确定各个类所包含的函数所属的动态库；

比对所获取的各个类所属的动态库与所确定的各个类所包含的函数所属的动态库，在所获取的一个类所属的动态库与所确定的各个类所包含的函数所属的动态库不同时，判定所获取的该类所属的动态库对应的函数被第三方动态库注入和/或修改；

其中，所述确定各个类所包含的函数所属的动态库包括：

获取当前进程中加载的所有动态库及其对应的函数地址范围；

确定所获取的所有函数的函数地址；

将所确定的所有函数的函数地址与所获取的所有动态库对应的函数地址范围进行比对，确定所获取的所有函数所属的动态库；

所述获取当前进程中加载的所有动态库对应的函数地址范围的步骤包括：根据当前进程中加载的所有动态库对应的句柄信息得到所述所有动态库对应的函数地址范围。

7.如权利要求6所述的参数检测装置，其特征在于，所述获取当前进程中加载的所有动态库及其对应的函数地址范围包括：

获取当前进程中加载的所有动态库及所获取的动态库对应的句柄信息；

根据所述句柄信息，得到当前进程中加载的所有动态库的内存起始地址和内存占用信息；

根据所述内存起始地址和内存占用信息得到当前进程中加载的所有动态库对应函数地址范围。

8.如权利要求6所述的参数检测装置，其特征在于，所述根据各个类获取各个类所属的动态库包括：

获取各个类对应的NSBundle对象的可执行路径信息；

根据所获取的NSBundle对象的可执行路径信息，得到各个类所属的动态库。

9.如权利要求6所述的参数检测装置，其特征在于，所述获取当前进程中的所有类及其各个类所包含的函数包括：

初始化系统，根据objc_getClassList获取到当前进程中的所有类；

根据class_copyMethodList，获取到各个类所包含的函数。

10.如权利要求6至9任一项所述的参数检测装置，其特征在于，在比对所获取的各个类所属的动态库与所确定的各个类所包含的函数所属的动态库，在所获取的一个类所属的动态库与所确定的各个类所包含的函数所属的动态库不同时，判定所获取的所述类所属的动态库对应的函数被第三方动态库注入和/或修改之后，所述指令还由处理器加载并执行：

确定所获取的所述类所属的动态库中对应的函数；

发出提示信息。

11.一种存储设备，其中存储有多条指令，所述指令适于由处理器加载并执行：

获取当前进程中的所有类和各个类所包含的函数，并根据各个类获取各个类所属的动态库；

确定各个类所包含的函数所属的动态库；

比对所获取的各个类所属的动态库与所确定的各个类所包含的函数所属的动态库，在所获取的一个类所属的动态库与所确定的各个类所包含的函数所属的动态库不同时，判定

所获取的该类所属的动态库对应的函数被第三方动态库注入和/或修改；

其中，所述确定各个类所包含的函数所属的动态库的步骤包括：

获取当前进程中加载的所有动态库及其对应的函数地址范围；

确定所获取的所有函数的函数地址；

将所确定的所有函数的函数地址与所获取的所有动态库对应的函数地址范围进行比对，确定所获取的所有函数所属的动态库；

所述获取当前进程中加载的所有动态库对应的函数地址范围的步骤包括：根据当前进程中加载的所有动态库对应的句柄信息得到所述所有动态库对应的函数地址范围。

参数检测方法及装置

技术领域

[0001] 本发明涉及到移动终端数据处理技术领域,特别涉及到参数检测方法及装置。

背景技术

[0002] 随着智能技术的不断发展,移动终端由于其便携性逐渐成为主要的网络信息终端,同时伴随着移动终端的发展越来越多的智能系统被开发使用在移动终端上,例如,ios系统、安卓系统等。然而ios系统因其系统使用的局限性,导致可以下载使用的应用程序较少,降低了用户对ios系统移动终端的使用。因此,为了突破ios系统的限制,越来越多的ios系统用户选择将移动终端的ios系统进行越狱,即通过ios系统越狱以使移动终端能满足更多应用程序的要求,使得ios系统的终端能使用更多的应用程序。

[0003] 在ios越狱环境下,进程注入是一种普遍的技术,绝大多数的第三方插件(插件实质是动态库)都是通过进程注入实现的,实现的原理是修改进程中原有的函数,把自己的代码加入到进程中,进而实现ios系统的越狱。

[0004] 现有技术中,无法检测出ios越狱环境下哪些原有函数被修改,因此,会导致用户在使用支付类、账号类的应用时,无法确定是否处于安全运行环境,降低了移动终端使用的安全性。

发明内容

[0005] 本发明实施例提供一种参数检测方法及装置,旨在解决导致用户在使用支付类、账号类的应用时,无法确定是否处于安全运行环境,降低了移动终端使用的安全性的问题。

[0006] 本发明实施例提出一种参数检测方法,所述参数检测方法包括步骤:

[0007] 获取当前进程中的所有类和各个类所包含的函数,并根据各个类获取各个类所属的动态库;

[0008] 确定各个类所包含的函数所属的动态库;

[0009] 比对所获取的各个类所属的动态库与所确定的各个类所包含的函数所属的动态库,在所获取的一个类所属的动态库与所确定的各个类所包含的函数所属的动态库不同时,判定所获取的该类所属的动态库对应的函数被第三方动态库注入和/或修改。

[0010] 本发明还提出一种参数检测装置,所述参数检测装置包括:

[0011] 处理器,适于实现各指令;以及

[0012] 存储设备,适于存储多条指令,所述指令适于由处理器加载并执行:

[0013] 获取当前进程中的所有类和各个类所包含的函数,并根据各个类获取各个类所属的动态库;

[0014] 确定各个类所包含的函数所属的动态库;

[0015] 比对所获取的各个类所属的动态库与所确定的各个类所包含的函数所属的动态库,在所获取的一个类所属的动态库与所确定的各个类所包含的函数所属的动态库不同时,判定所获取的该类所属的动态库对应的函数被第三方动态库注入和/或修改。

[0016] 本发明还提出一种存储设备，其中存储有多条指令，所述指令适于由处理器加载并执行：

[0017] 获取当前进程中的所有类和各个类所包含的函数，并根据各个类获取各个类所属的动态库；

[0018] 确定各个类所包含的函数所属的动态库；

[0019] 比对所获取的各个类所属的动态库与所确定的各个类所包含的函数所属的动态库，在所获取的一个类所属的动态库与所确定的各个类所包含的函数所属的动态库不同时，判定所获取的该类所属的动态库对应的函数被第三方动态库注入和/或修改。

[0020] 本发明实施例通过检测当前运行程序中所有类所属的动态库和所有类对应函数所属的动态库，在两者有不同时，判定有函数被第三方动态库注入和/或修改。实现自动检测出当前进程中哪些原有函数被修改，进而保证支付类、账号类的应用处于安全运行环境，提高移动终端使用的安全性。

附图说明

[0021] 图1为本发明参数检测方法的第一实施例的流程示意图；

[0022] 图2为图1中步骤S10一实施例的细化流程示意图；

[0023] 图3为图1中步骤S10另一实施例的细化流程示意图；

[0024] 图4为图1中步骤S20一实施例的细化流程示意图；

[0025] 图5为图4中步骤S21的细化流程示意图；

[0026] 图6为本发明参数检测方法的第二实施例的流程图；

[0027] 图7为本发明参数检测装置的第一实施例的功能模块示意图；

[0028] 图8为图7中获取模块的细化功能模块示意图；

[0029] 图9为本发明参数检测装置的第二实施例的功能模块示意图；

[0030] 图10为本发明参数检测装置所在终端的硬件结构示意图。

[0031] 本发明目的的实现、功能特点及优点将结合实施例，参照附图做进一步说明。

具体实施方式

[0032] 应当理解，此处所描述的具体实施例仅仅用以解释本发明，并不用于限定本发明。

[0033] 如图1所示，图1为本发明参数检测方法的第一实施例的流程示意图。本实施例提到的参数检测方法包括以下步骤：

[0034] 步骤S10，获取当前进程中的所有类及其各个类所属的动态库和各个类所包含的函数；

[0035] 随着ios系统的移动终端越来越被人们所喜欢，与ios系统对应的应用程序被开发以供用户加载在ios系统的移动终端。然而，与安卓、windows phone系统不同的是，ios系统为封闭式系统，第三方应用不能完全访问ios系统终端的所有目录，因此，导致很多应用程序无法成功加载到ios系统的终端中。为了突破ios系统的限制，越来越多的用户选择对ios系统进行越狱操作，即针对ios系统限制用户存储读写权限的破解操作，经过越狱的ios终端拥有对系统底层的读写权限，能够让ios系统终端免费使用破解后的App Store应用程序。对ios系统进行越狱操作主要是通过注入的范式修改进程中原有的函数，把需要的代码

加入到进程中，突破ios系统底层读写权限。在本发明实施例中以ios系统的移动终端为例进行描述，但本发明实施例并不仅仅局限于应用在ios系统移动终端的参数检测。

[0036] 在ios系统越狱后，侦测用户触发的参数检测指令或者通过系统自动设置，在设置的时间达到(在开机或系统重新启动，或一段时间间隔5min/10min等)时，自动开启参数检测指令，在通过自动设置触发的参数检测指令后，发出提示信息，以提示用户开了参数检测，在接收到停止参数检测指令时，不开启参数检测指令，在接收到进行参数检测的指令时，或者在预设时间达到(20s或30s等)时，继续参数检测过程。在侦测到参数检测指令后，获取当前进程中的所有类及各个类所属的动态库和各个类所包含的函数。可以理解的是：在侦测到参数检测指令后，判断当前系统是否越狱，在当前系统越狱时，获取当前进程中的所有类及各个类所属的动态库和各个类所包含的函数；在当前系统未越狱时，提示用户无需进行参数检测。所述的越狱为系统的破解，即完成系统底层读写权限的破解，不局限于ios系统的越狱，也可是其他需要破解底层读写权限的系统。

[0037] 可以理解的是：在参数检测触发时，获取当前系统的内存使用信息，在当前系统的内存使用值大于或等于预设使用阈值(90％或80％等)时，停止参数检测，在停止参数检测后，实时或者定时(5min或8min时间间隔后)获取当前系统的内存使用值，以判定是否可以恢复参数检测，或者在停止参数检测后，提示用户关闭一些应用程序的运行，以恢复参数检测过程。在当前系统的内存使用之小于预设使用阈值时，继续参数检测过程。通过获取系统的内存使用信息，在前系统的内存使用之小于预设使用阈值时，才进行参数检测过程，有效保证其他应用程序的运行，并提高系统性能。

[0038] 具体的，参考图2，所述获取当前进程中所有类及各个类所包含的函数的过程可以是：步骤S11，初始化系统，根据objc_getClassList获取到当前进程中的所有类；步骤S12，根据class_copyMethodList，获取到各个类所包含的函数。

[0039] 初始化系统，即开启ios移动终端，使得移动终端的ios系统处于开启状态，获取objc_getClassList，根据objc_getClassList获取到当前进程中的所有类；获取到class_copyMethodList，根据class_copyMethodList，获取到各个类所包含的函数，即得到当前进程中所有类的所包含的所有函数。所获取的函数包括ios系统原有的函数，以及越狱后被第三方动态库注入和/或修改的函数。

[0040] 具体的，参考图3，所述获取各个类所属的动态库的过程可以包括：步骤S13，获取各个类对应的NSBundle对象的可执行路径信息；步骤S14，根据所获取的NSBundle对象的可执行路径信息，得到各个类所属的动态库。

[0041] 在初始化系统，获取到当前进程中的所有类之后，获取各个类对应的NSBundle对象及其可执行路径信息，根据获取的可执行路径信息，得到各个类所属的动态库，即得到各个类所属的动态库名称。所获取的动态库只包括ios系统越狱前的原动态库名称。

[0042] 步骤S20，确定所获取的函数所属的动态库；

[0043] 在获取到当前进程中所有类及各个类所包含的函数后，确定所获取的函数所属的动态库。

[0044] 具体的，参考图4，所述确定所获取的函数所属的动态库的过程可以包括：

[0045] 步骤S21，获取当前进程中加载的所有动态库及其对应的函数地址范围；

[0046] 通过dyld的API(Application Programming Interface，应用程序编程接口)，获

取当前进程中加载的所有动态库，所获取的动态库包括ios系统的原有动态库和第三方的动态库，在获取到进程中加载的所有动态库后，获取当前进程中加载的所有动态库对应的函数地址范围。具体的，参考图5，所述获取当前进程中加载的所有动态库对应的函数地址范围的过程可以包括：

[0047] 步骤S211，获取当前进程中加载的所有动态库对应的句柄信息；

[0048] 步骤S212，根据所述句柄信息，得到当前进程中加载的所有动态库的内存起始地址和内存占用信息；

[0049] 步骤S213，根据所述内存起始地址和内存占用信息得到当前进程中加载的所有动态库对应函数地址范围。

[0050] 所述句柄信息用来标识应用程序中的不同对象和同类对象中的不同的实例，且通过句柄信息来记录数据地址信息，所记录的数据地址信息包括数据地址变更信息及数据在内存中的地址信息。根据句柄信息得到当前进程中加载的所有动态库的内存起始地址和内存占用信息，根据内存其实地址和内存占用信息得到当前进程中加载的所有动态库的内存结束地址，进而通过起始地址和结束地址得到当前进程中加载的所有动态库对应函数地址范围。所获取到的地址范围可以是一个或者多个，每个地址范围对应一个动态库，且每个动态库下包含至少一个函数。

[0051] 步骤S22，确定所获取的所有函数的函数地址；

[0052] 步骤S23，将所确定的函数地址与所获取的函数地址范围进行比对，确定所获取的函数所属的动态库。

[0053] 检测所获取的函数，即检测所有当前进程中的所有类所包含的所有函数，得到每个函数的函数地址，将所得到的函数地址与所获取的函数地址范围进行比对，即获取到所得到的函数地址具体落入哪个函数地址范围内，进而通过函数地址范围与动态库的映射关系，确定所获取的函数所属的动态库。每个函数或者多个函数对应一个动态库，因此，所确定的动态库可以是一个或者多个动态库。

[0054] 步骤S30，比对所获取的动态库与所确定的动态库，在所获取的动态库与所确定的动态库不同时，判定所获取的动态库对应的函数被第三方动态库注入和修改。

[0055] 在获取到所有类所属的动态库和确定所有类对应函数的动态库后，比对所获取的动态库和所确定的动态库，以确定所获取的动态库是否与所确定的动态库相同，当所获取的动态库存在多个时，将所获取的动态库逐个与所确定的动态库比对，以确定所获取的所有动态库是否均与所确定的动态库相同。在所获取的动态库与所确定的动态库不同时，判定所获取的动态库对应的函数被第三方动态库注入和修改；在所获取的动态库与所确定的动态库相同时，判定所获取的动态库对应的函数未被第三方动态库注入和修改。

[0056] 本发明实施例通过检测当前运行程序中所有类所属的动态库和所有类对应函数所属的动态库，在两者有不同时，判定有函数被第三方动态库注入和/或修改。实现自动检测出当前进程中哪些原有函数被修改，进而保证支付类、账号类的应用处于安全运行环境，提高移动终端使用的安全性。

[0057] 进一步地，基于上述参数检测方法第一实施例，提出本发明参数检测方法的第二实施例。如图6所示，在步骤S30之后还包括步骤：

[0058] 步骤S40，确定所获取的动态库中与所确定的动态库不同的动态库对应的函数；

[0059] 步骤S50,发出提示信息。

[0060] 在所获取的动态库与所确定的动态库不同时,确定所获取的动态库中与所确定的动态库所有不同的动态库,并确定所有不同动态库对应的函数,发出提示信息。所述提示信息包括但不限于当前进程中的函数被第三方注入和/或修改、被注入和/或修改的函数的信息等,通过提示信息提示用户哪些函数和哪些动态库被注入和/或修改。所述发出提示信息的方式可以是文字、图片、语音、视频等。本发明实施例通过在当前进程中有函数被第三方动态库注入和/或修改时,显示所被注入和/或修改的函数信息和发出提示信息,以提示移动终端用户有哪些函数被注入和/或修改,进而使得移动终端用户能及时根据提示信息注意移动终端应用程序的加载和使用,提高移动终端应用程序使用的安全性,进而提高移动终端应用程序的使用体验。

[0061] 上述第一至第二实施例的参数检测的方法的执行主体均可以为移动终端。更进一步地,该方法可以由安装在移动设备上的客户端应用程序(如检测软件等)实现,其中,该移动终端可以包括但不限于手机、平板电脑或者PDA(Personal Digital Assistant,个人数字助理)等电子设备。

[0062] 进一步地,提出本发明的参数检测装置的第一实施例。如图7所示,所述参数检测装置包括:获取模块10、处理模块20及检测模块30。

[0063] 所述获取模块10,用于获取当前进程中的所有类及其各个类所属的动态库和各个类所包含的函数;

[0064] 随着ios系统的移动终端越来越被人们所喜欢,与ios系统对应的应用程序被开发以供用户加载在ios系统的移动终端。然而,与安卓、windows phone系统不同的是,ios系统为封闭式系统,第三方应用不能访问ios系统终端的所有目录,因此,导致很多应用程序无法成功加载到ios系统的终端中。为了突破ios系统的限制,越来越多的用户选择对ios系统进行越狱操作,即针对ios系统限制用户存储读写权限的破解操作,经过越狱的ios终端拥有对系统底层的读写权限,能够让ios系统终端免费使用破解后的App Store应用程序。对ios系统进行越狱操作主要是通过注入的范式修改进程中原有的函数,把需要的代码加入到进程中,突破ios系统底层读写权限。在本发明实施例中以ios系统的移动终端为例进行描述,但本发明实施例并不仅仅局限于应用在ios系统移动终端的参数检测。

[0065] 在ios系统越狱后,侦测用户触发的参数检测指令或者通过系统自动设置,在设置的时间达到(在开机或系统重新启动,或一段时间间隔5min/10min等)时,自动开启参数检测指令,在通过自动设置触发的参数检测指令后,发出提示信息,以提示用户开了参数检测,在接收到停止参数检测指令时,不开启参数检测指令,在接收到进行参数检测的指令时,或者在预设时间达到(20s或30s等)时,继续参数检测过程。在侦测到参数检测指令后,获取当前进程中的所有类及各个类所属的动态库和各个类所包含的函数。可以理解的是:在侦测到参数检测指令后,判断当前系统是否越狱,在当前系统越狱时,获取当前进程中的所有类及各个类所属的动态库和各个类所包含的函数;在当前系统未越狱时,提示用户无需进行参数检测。所述的越狱为系统的破解,即完成系统底层读写权限的破解,不局限于ios系统的越狱,也可是其他需要破解底层读写权限的系统。

[0066] 可以理解的是:在参数检测触发时,获取当前系统的内存使用信息,在当前系统的内存使用值大于或等于预设使用阈值(90%或80%等)时,停止参数检测,在停止参数检测

后,实时或者定时(5min或8min时间间隔后)获取当前系统的内存使用值,以判定是否可以恢复参数检测,或者在停止参数检测后,提示用户关闭一些应用程序的运行,以恢复参数检测过程。在当前系统的内存使用之小于预设使用阈值时,继续参数检测过程。通过获取系统的内存使用信息,在前系统的内存使用之小于预设使用阈值时,才进行参数检测过程,有效保证其他应用程序的运行,并提高系统性能。

[0067] 具体的,参考图9,所述获取模块10包括获取单元11和确定单元12,其中,所述确定单元12,用于初始化系统,根据objc_getClassList获取到当前进程中的所有类;所述获取单元11,用于根据class_copyMethodList,获取到各个类所包含的函数。

[0068] 初始化系统,即开启ios移动终端,使得移动终端的ios系统处于开启状态,获取objc_getClassList,根据objc_getClassList获取到当前进程中的所有类;获取到class_copyMethodList,根据class_copyMethodList,获取到各个类所包含的函数,即得到当前进程中所有类的所包含的所有函数。所获取的函数包括ios系统原有的函数,以及越狱后被第三方动态库注入和/或修改的函数。

[0069] 进一步地,所述获取单元11,还用于获取各个类对应的NSBundle对象的可执行路径信息;所述确定单元12,还用于根据所获取的NSBundle对象的可执行路径信息,得到各个类所属的动态库。

[0070] 在初始化系统,获取到当前进程中的所有类之后,获取各个类对应的NSBundle对象及其可执行路径信息,根据获取的可执行路径信息,得到各个类所属的动态库,即得到各个类所属的动态库名称。所获取的动态库只包括ios系统越狱前的原动态库名称。

[0071] 所述处理模块20,还用于确定所获取的函数所属的动态库;

[0072] 在获取到当前进程中所有类及各个类所包含的函数后,确定所获取的函数所属的动态库。

[0073] 进一步地,所述获取单元11,还用于获取当前进程中加载的所有动态库及其对应的函数地址范围;

[0074] 通过dyld的API(Application Programming Interface,应用程序编程接口),获取当前进程中加载的所有动态库,所获取的动态库包括ios系统的原有动态库和第三方的动态库,在获取到进程中加载的所有动态库后,获取当前进程中加载的所有动态库对应的函数地址范围。

[0075] 具体的,所述获取单元11获取当前进程中加载的所有动态库对应的函数地址范围的过程可以包括:获取当前进程中加载的所有动态库对应的句柄信息;根据所述句柄信息,得到当前进程中加载的所有动态库的内存起始地址和内存占用信息;根据所述内存起始地址和内存占用信息得到当前进程中加载的所有动态库对应函数地址范围。

[0076] 所述句柄信息用来标识应用程序中的不同对象和同类对象中的不同的实例,且通过句柄信息来记录数据地址信息,所记录的数据地址信息包括数据地址变更信息及数据在内存中的地址信息。根据句柄信息得到当前进程中加载的所有动态库的内存起始地址和内存占用信息,根据内存其实地址和内存占用信息得到当前进程中加载的所有动态库的内存结束地址,进而通过起始地址和结束地址得到当前进程中加载的所有动态库对应函数地址范围。所获取到的地址范围可以是一个或者多个,每个地址范围对应一个动态库,且每个动态库下包含至少一个函数。

[0077] 所述确定单元12，还用于确定所获取的所有函数的函数地址；将所确定的函数地址与所获取的函数地址范围进行比对，确定所获取的函数所属的动态库。

[0078] 检测所获取的函数，即检测所有当前进程中的所有类所包含的所有函数，得到每个函数的函数地址，将所得到的函数地址与所获取的函数地址范围进行比对，即获取到所得到的函数地址具体落入哪个函数地址范围内，进而通过函数地址范围与动态库的映射关系，确定所获取的函数所属的动态库。每个函数或者多个函数对应一个动态库，因此，所确定的动态库可以是一个或者多个动态库。

[0079] 所述检测模块30，用于比对所获取的动态库与所确定的动态库，在所获取的动态库与所确定的动态库不同时，判定所获取的动态库对应的函数被第三方动态库注入和修改。

[0080] 在获取到所有类所属的动态库和确定所有类对应函数的动态库后，比对所获取的动态库和所确定的动态库，以确定所获取的动态库是否与所确定的动态库相同，当所获取的动态库存在多个时，将所获取的动态库逐个与所确定的动态库比对，以确定所获取的所有动态库是否均与所确定的动态库相同。在所获取的动态库与所确定的动态库不同时，判定所获取的动态库对应的函数被第三方动态库注入和修改；在所获取的动态库与所确定的动态库相同时，判定所获取的动态库对应的函数未被第三方动态库注入和修改。

[0081] 本发明实施例通过检测当前运行程序中所有类所属的动态库和所有类对应函数所属的动态库，在两者有不同时，判定有函数被第三方动态库注入和/或修改。实现自动检测出当前进程中哪些原有函数被修改，进而保证支付类、账号类的应用处于安全运行环境，提高移动终端使用的安全性。

[0082] 进一步地，提出本发明参数检测装置的第二实施例。如图9所示，所述参数检测装置还包括：提示模块40。

[0083] 所述处理模块20，还用于确定所获取的动态库中与所确定的动态库不同的动态库对应的函数；

[0084] 所述提示模块50，用于发出提示信息。

[0085] 在所获取的动态库与所确定的动态库不同时，确定所获取的动态库中与所确定的动态库所有不同的动态库，并确定所有不同动态库对应的函数，发出提示信息。所述提示信息包括但不限于当前进程中的函数被第三方注入和/或修改、被注入和/或修改的函数的信息等，通过提示信息提示用户哪些函数和哪些动态库被注入和/或修改。所述发出提示信息的方式可以是文字、图片、语音、视频等。本发明实施例通过在当前进程中有函数被第三方动态库注入和/或修改时，显示所被注入和/或修改的函数信息和发出提示信息，以提示移动终端用户有哪些函数被注入和/或修改，进而使得移动终端用户能及时根据提示信息注意移动终端应用程序的加载和使用，提高移动终端应用程序使用的安全性，进而提高移动终端应用程序的使用体验。

[0086] 如图10所示，图10为本发明实施例中的装置所在终端的总线图。该终端可以包括：至少一个处理器301，例如CPU，至少一个网络接口304，用户接口303，存储器305，至少一个通信总线302。其中，通信总线302用于实现这些组件之间的连接通信。其中，用户接口303可以包括显示屏(Display)、键盘(Keyboard)，还可以包括标准的有线接口、无线接口。网络接口304可以包括标准的有线接口、无线接口(如无线网络接口)。存储器305可以是高速RAM存

储器，也可以是非不稳定的存储器(non-volatile memory)，例如至少一个磁盘存储器。存储器305还可以是至少一个位于远离前述处理器301的存储装置。作为一种计算机存储介质的存储器305中可以包括操作系统、网络通信模块、用户接口模块以及参数检测的程序。

[0087] 在图10所示的参数检测的装置所在终端中，网络接口304主要用于连接服务器，与服务器进行数据通信；而用户接口303主要用于接收用户指令，并与用户进行交互；而处理器301可以用于调用存储器305中存储的参数检测的程序，并执行以下操作：

[0088] 通过用户接口303接收用户触发的或者系统自动触发的参数检测指令；在用户接口303接收到参数检测指令后，获取当前进程中的所有类及各个类所属的动态库和各个类所包含的函数；确定所获取的函数所属的动态库；比对所获取的动态库与所确定的动态库，在所获取的动态库与所确定的动态库不同时，判定所获取的动态库对应的函数被第三方动态库注入和/或修改。

[0089] 在一个实施例中，处理器301调用存储器305中存储的参数检测的程序还可以执行以下操作：

[0090] 获取当前进程中加载的所有动态库及其对应的函数地址范围；

[0091] 确定所获取的所有函数的函数地址；

[0092] 将所确定的函数地址与所获取的函数地址范围进行比对，确定所获取的函数所属的动态库。

[0093] 在一个实施例中，处理器301调用存储器305中存储的参数检测的程序还可以执行以下操作：

[0094] 获取当前进程中加载的所有动态库及所获取的动态库对应的句柄信息；

[0095] 根据所述句柄信息，得到当前进程中加载的所有动态库的内存起始地址和内存占用信息；

[0096] 根据所述内存起始地址和内存占用信息得到当前进程中加载的所有动态库对应函数地址范围。

[0097] 在一个实施例中，处理器301调用存储器305中存储的参数检测的程序还可以执行以下操作：

[0098] 获取各个类对应的NSBundle对象的可执行路径信息；

[0099] 根据所获取的NSBundle对象的可执行路径信息，得到各个类所属的动态库。

[0100] 在一个实施例中，处理器301调用存储器305中存储的参数检测的程序还可以执行以下操作：

[0101] 初始化系统，根据objc_getClassList获取到当前进程中的所有类；

[0102] 根据class_copyMethodList，获取到各个类所包含的函数。

[0103] 在一个实施例中，处理器301调用存储器305中存储的参数检测的程序还可以执行以下操作：

[0104] 确定所获取的动态库中与所确定的动态库不同的动态库对应的函数；

[0105] 发出提示信息。

[0106] 需要说明的是，在本文中，术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含，从而使得包括一系列要素的过程、方法、物品或者装置不仅包括那些要素，而且还包括没有明确列出的其他要素，或者是还包括为这种过程、方法、物品或者装置所固有

的要素。在没有更多限制的情况下，由语句“包括一个……”限定的要素，并不排除在包括该要素的过程、方法、物品或者装置中还存在另外的相同要素。

[0107] 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。

[0108] 通过以上的实施方式的描述，本领域的技术人员可以清楚地了解到上述实施例方法可借助软件加必需的通用硬件平台的方式来实现，当然也可以通过硬件，但很多情况下前者是更佳的实施方式。基于这样的理解，本发明的技术方案本质上或者说对现有技术做出贡献的部分可以以软件产品的形式体现出来，该计算机软件产品存储在一个存储介质(如ROM/RAM、磁碟、光盘)中，包括若干指令用以使得一台终端设备(可以是手机，计算机，服务器，或者网络设备等)执行本发明各个实施例所述的方法。

[0109] 以上所述仅为本发明的优选实施例，并非因此限制本发明的专利范围，凡是利用本发明说明书及附图内容所作的等效结构或等效流程变换，或直接或间接运用在其他相关的技术领域，均同理包括在本发明的专利保护范围内。

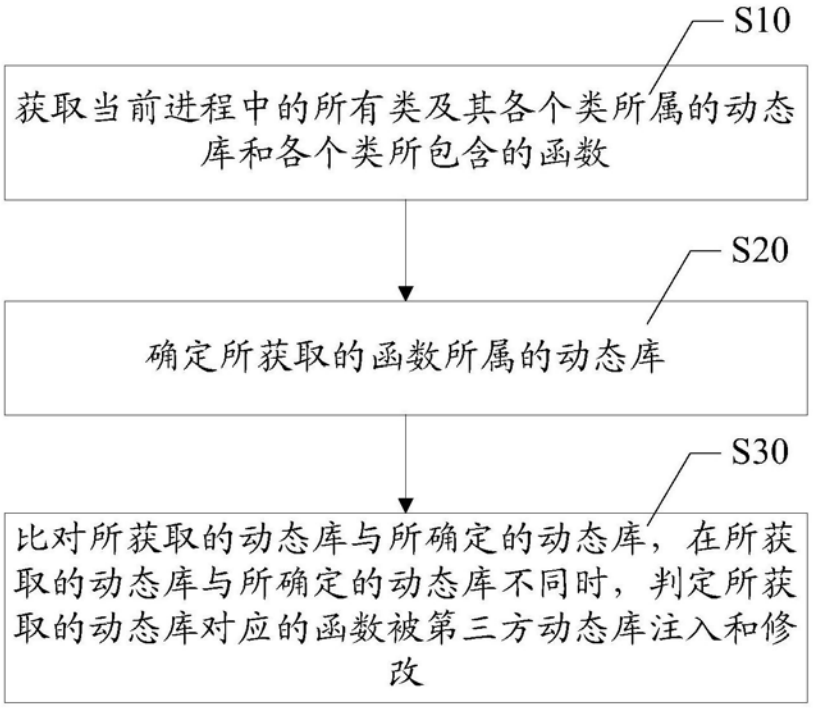

S10
获取当前进程中的所有类及其各个类所属的动态库和各个类所包含的函数
S20
确定所获取的函数所属的动态库
S30
比对所获取的动态库与所确定的动态库，在所获取的动态库与所确定的动态库不同时，判定所获取的动态库对应的函数被第三方动态库注入和修改

图1

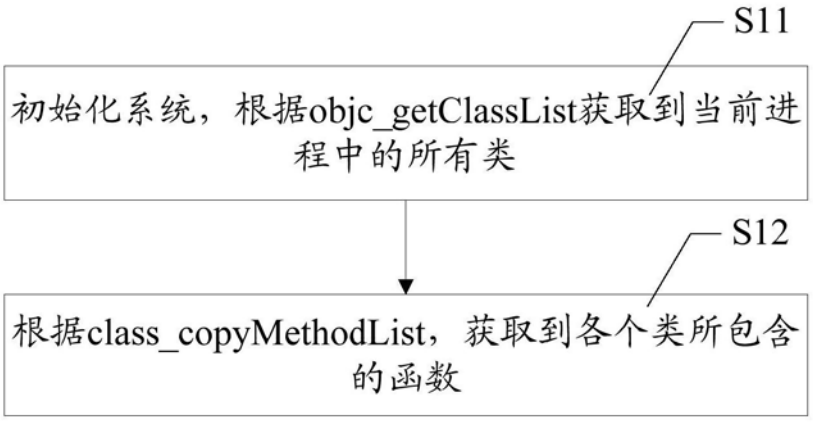

S11
初始化系统，根据objc_getClassList获取到当前进程中的所有类
S12
根据class_copyMethodList，获取到各个类所包含的函数

图2

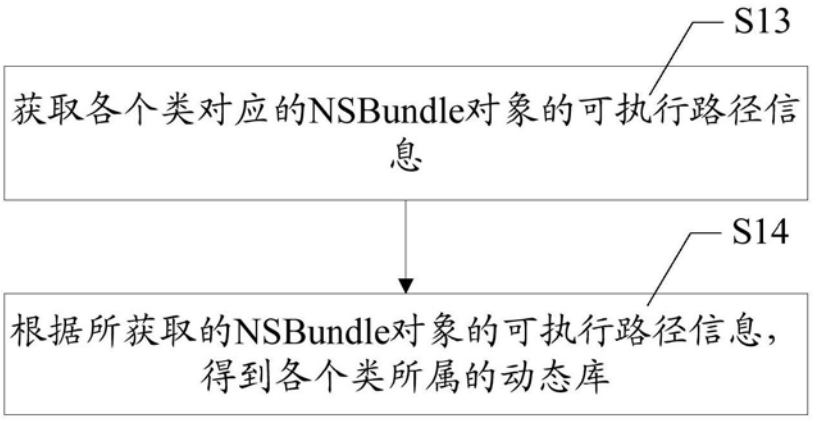

S13
获取各个类对应的NSBundle对象的可执行路径信息
S14
根据所获取的NSBundle对象的可执行路径信息，得到各个类所属的动态库

图3

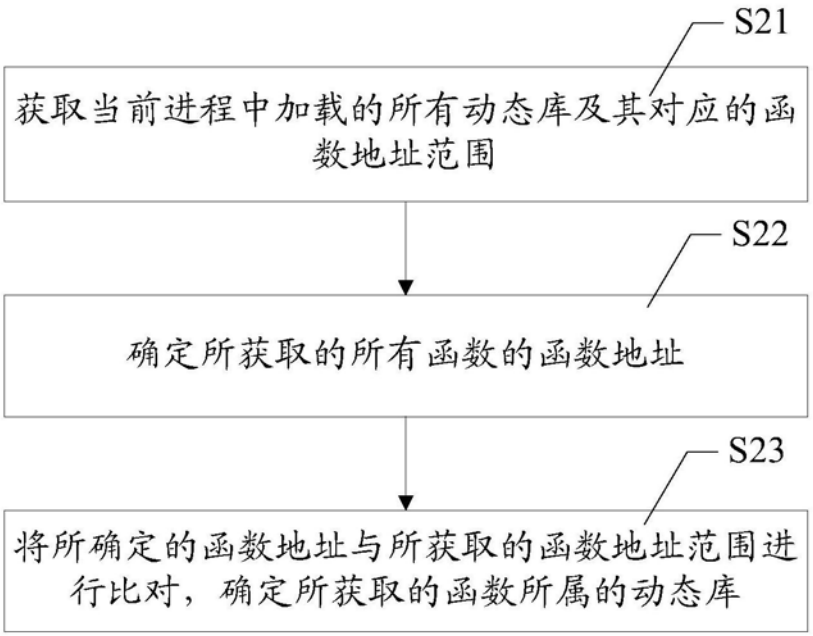

S21
获取当前进程中加载的所有动态库及其对应的函数地址范围
S22
确定所获取的所有函数的函数地址
S23
将所确定的函数地址与所获取的函数地址范围进行比对，确定所获取的函数所属的动态库

图4

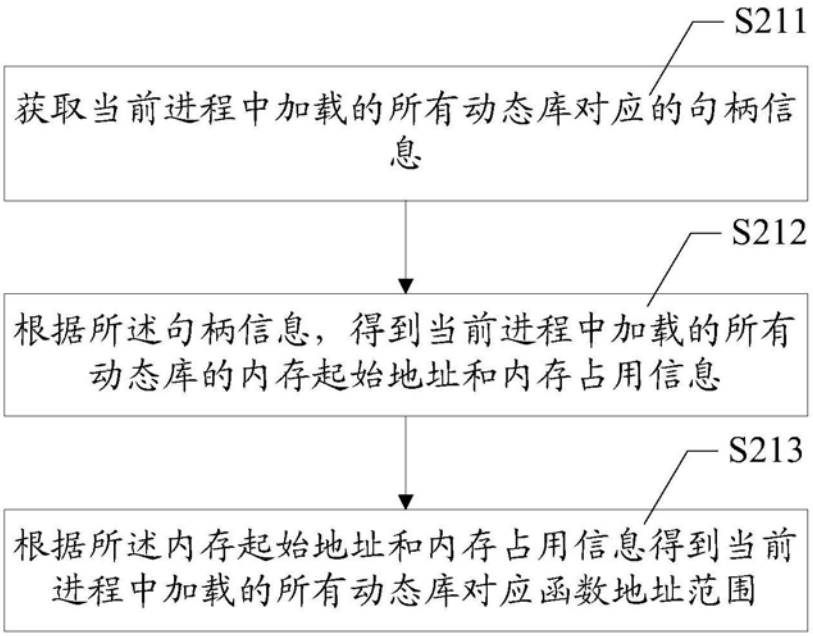

S211
获取当前进程中加载的所有动态库对应的句柄信息
S212
根据所述句柄信息，得到当前进程中加载的所有动态库的内存起始地址和内存占用信息
S213
根据所述内存起始地址和内存占用信息得到当前进程中加载的所有动态库对应函数地址范围

图5

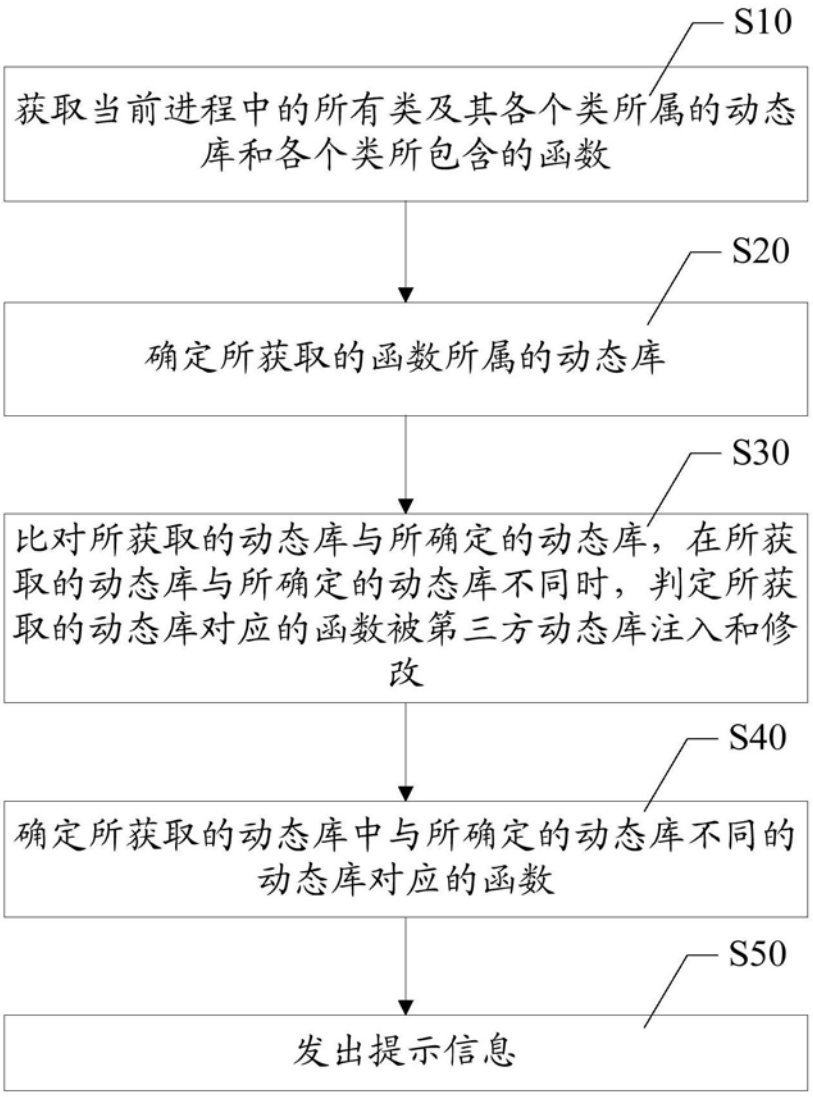

S10
获取当前进程中的所有类及其各个类所属的动态库和各个类所包含的函数
S20
确定所获取的函数所属的动态库
S30
比对所获取的动态库与所确定的动态库，在所获取的动态库与所确定的动态库不同时，判定所获取的动态库对应的函数被第三方动态库注入和修改
S40
确定所获取的动态库中与所确定的动态库不同的动态库对应的函数
S50
发出提示信息

图6

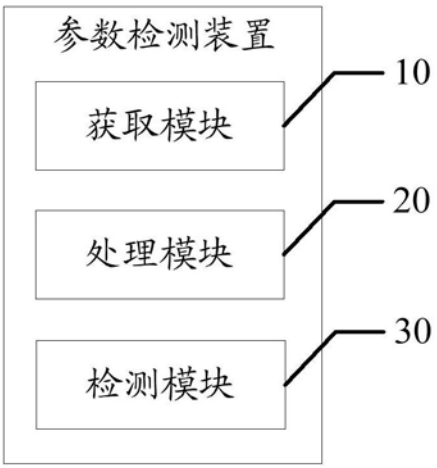

参数检测装置
获取模块
10
处理模块
20
检测模块
30

图7

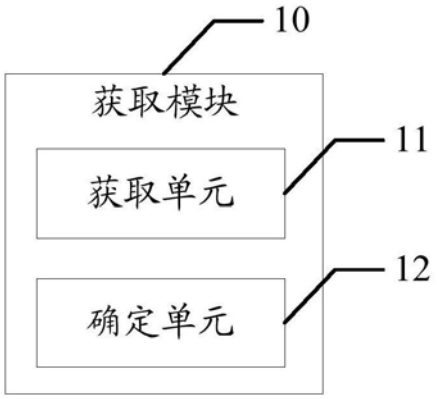


图8

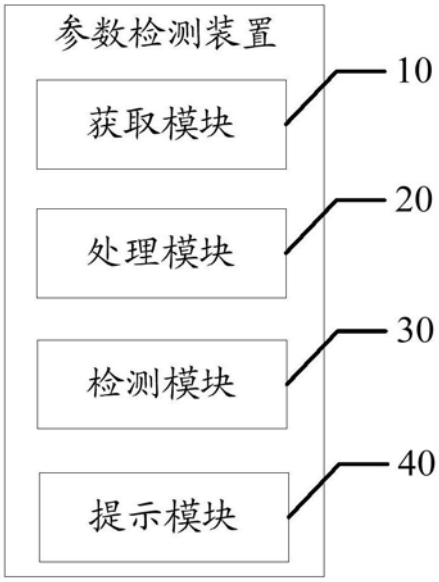


图9

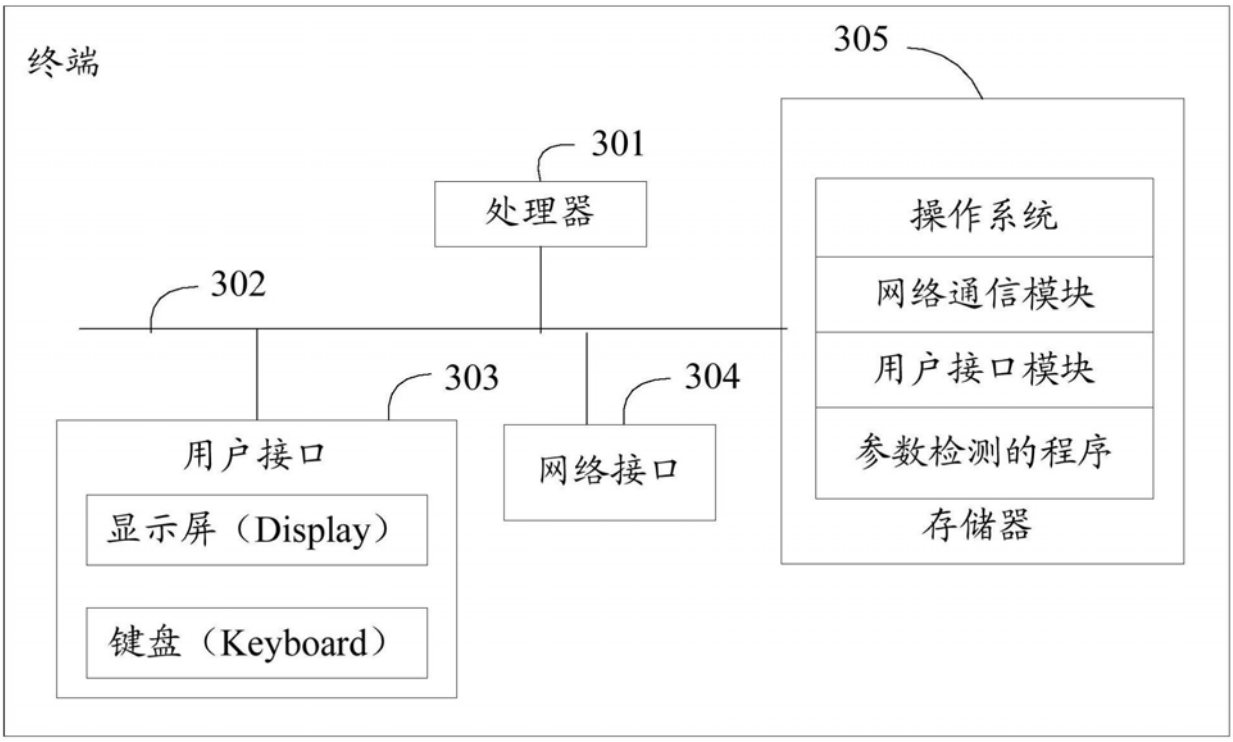

终端
305
301
处理器
操作系统
网络通信模块
用户接口模块
参数检测的程序
存储器
302
303
304
用户接口
显示屏（Display）
键盘（Keyboard）
网络接口

图10