



(12)发明专利

(10)授权公告号 CN 105357204 B

(45)授权公告日 2020.06.16

(21)申请号 201510788423.9

H04W 8/18(2009.01)

(22)申请日 2015.11.17

(56)对比文件

(65)同一申请的已公布的文献号
申请公布号 CN 105357204 A

CN 103593465 A,2014.02.19,

CN 104408336 A,2015.03.11,

CN 104811931 A,2015.07.29,

CN 104394022 A,2015.03.04,

CN 104348624 A,2015.02.11,

(43)申请公布日 2016.02.24

(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

火星十一郎.基于局部敏感哈希的协同过滤
算法之simHash算法.《博客园,https://
www.cnblogs.com/hxsyl/p/4456001.html》
.2015,

(72)发明人 贺海军 孔蓓蓓 熊健 熊焰
杨剑鸣

weixin_33727510.常见hash算法的原理.

《CSDN,https://blog.csdn.net/weixin_
33727510/article/details/85661840》.2014,

(74)专利代理机构 广州三环专利商标代理有限
公司 44202

代理人 郝传鑫 熊永强

审查员 李浩

(51)Int.Cl.

H04L 29/06(2006.01)

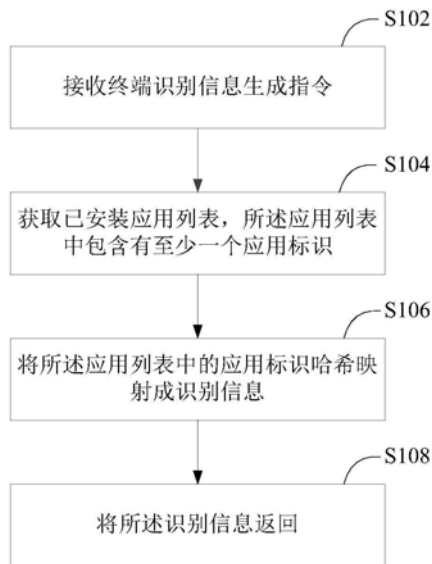
权利要求书2页 说明书7页 附图4页

(54)发明名称

生成终端识别信息的方法及装置

(57)摘要

本发明实施例公开了一种生成终端识别信息的方法,包括:接收终端识别信息生成指令;获取已安装应用列表,所述应用列表中至少包含有一个应用标识;将所述应用列表中的应用标识哈希映射成识别信息;将所述识别信息返回。本发明还相应地公开了一种生成终端识别信息的装置。上述生成终端识别信息的方法和装置生成的识别信息不方便伪造或篡改。



1. 一种生成终端识别信息的方法,其特征在于,包括:

接收终端识别信息生成指令;

获取已安装应用列表,所述应用列表中包含有至少一个应用标识;

将所述应用列表中的应用标识归类为多个应用标识子集,所述应用标识子集中包含至少一个应用标识;

将所述多个应用标识子集哈希映射成预设位数的识别信息;所述识别信息是通过所述多个应用标识子集分别对应的二进制码和权重系数所生成的;一个应用标识子集对应的二进制码是由该应用标识子集中至少一个应用标识进行拼接得到的字符串所生成的;所述权重系数为用于表示所述多个应用标识在哈希映射过程中的重要性的参数;所述权重系数是基于所述应用标识子集中包含的应用标识的个数所设置的;

将所述识别信息返回。

2. 根据权利要求1所述的生成终端识别信息的方法,其特征在于,所述将所述多个应用标识子集哈希映射成预设位数的识别信息的步骤为:

为归类得到的每个所述应用标识子集设置权重参数;

将归类得到的每个所述应用标识子集通过预设的哈希算法映射成所述预设位数的二进制码;

根据每个所述应用标识子集的二进制码和权重系数生成每个所述应用标识子集对应的权重参数序列,且在权重参数序列中,序列值的绝对值均为对应该权重参数序列的应用标识子集的权重参数,对应该权重参数序列的二进制码在每一位上的值确定该权重参数序列中相应位置的序列值的正负符号;

生成初始化的所述预设位数的识别信息;

遍历所述识别信息的位,获取每个所述应用标识子集的权重参数序列在所述遍历到的位对应的序列位置的序列值,计算获取到的序列值的和,根据所述计算得到的和的符号设置所述识别信息在该遍历到的位的数值。

3. 根据权利要求2所述的生成终端识别信息的方法,其特征在于,所述将归类得到的每个所述应用标识子集通过预设的哈希算法映射成所述预设位数的二进制码的步骤为:

遍历归类得到的应用标识子集,将遍历到的应用标识子集中包含的应用标识的字符串进行拼接,根据预设的哈希算法将所述拼接得到的字符串映射成所述预设位数的二进制码。

4. 根据权利要求1所述的生成终端识别信息的方法,其特征在于,所述将所述应用列表中的应用标识归类为多个应用标识子集的步骤之前包括:

将所述应用列表中的应用标识排序。

5. 一种生成终端识别信息的装置,其特征在于,包括:

指令接收模块,用于接收终端识别信息生成指令;

列表获取模块,用于获取已安装应用列表,所述应用列表中包含有至少一个应用标识;

识别信息映射模块,用于将所述应用列表中的应用标识归类为多个应用标识子集,所述应用标识子集中包含至少一个应用标识,将所述多个应用标识子集哈希映射成预设位数的识别信息;所述识别信息是通过所述多个应用标识子集分别对应的二进制码和权重系数所生成的;一个应用标识子集对应的二进制码是由该应用标识子集中至少一个应用标识进

行拼接得到的字符串所生成的;所述权重系数为用于表示所述多个应用标识在哈希映射过程中的重要性的参数;所述权重系数是基于所述应用标识子集中包含的应用标识的个数所设置的;

识别信息返回模块,用于将所述识别信息返回。

6. 根据权利要求5所述的生成终端识别信息的装置,其特征在于,所述识别信息映射模块还用于:

为归类得到的每个所述应用标识子集设置权重参数;

将归类得到的每个所述应用标识子集通过预设的哈希算法映射成所述预设位数的二进制码;

根据每个所述应用标识子集的二进制码和权重系数生成每个所述应用标识子集对应的权重参数序列,且在权重参数序列中,序列值的绝对值均为对应该权重参数序列的应用标识子集的权重参数,对应该权重参数序列的二进制码在每一位上的值确定该权重参数序列中相应位置的序列值的正负符号;

生成初始化的所述预设位数的识别信息;

遍历所述识别信息的位,获取每个所述应用标识子集的权重参数序列在所述遍历到的位对应的序列位置的序列值,计算获取到的序列值的和,根据所述计算得到的和的符号设置所述识别信息在该遍历到的位的数值。

7. 根据权利要求6所述的生成终端识别信息的装置,其特征在于,所述识别信息映射模块还用于:遍历归类得到的应用标识子集,将遍历到的应用标识子集中包含的应用标识的字符串进行拼接,根据预设的哈希算法将所述拼接得到的字符串映射成所述预设位数的二进制码。

8. 根据权利要求5所述的生成终端识别信息的装置,其特征在于,所述识别信息映射模块还用于:将所述应用列表中的应用标识排序。

9. 一种终端,其特征在于,包括:处理器和存储器;

所述存储器存储有计算机程序,所述计算机程序被所述处理器执行时,使得所述处理器执行如权利要求1至4中任一项所述方法的步骤。

生成终端识别信息的方法及装置

技术领域

[0001] 本发明涉及计算机技术领域,尤其涉及一种生成终端识别信息的方法及装置。

背景技术

[0002] 传统技术中,为了区分不同的终端,终端上的操作系统通常需要根据终端的软硬件环境生成识别信息。例如,较常见的手机终端上,通常使用国际移动设备身份码(英文:International Mobile Equipment Identity,简称:IMEI,存储在手机固件中的一段识别信息)或者国际移动用户识别码(英文:International Mobile Subscriber Identification Number,简称:IMSI,存储在手机SIM卡中的一段识别信息)来区分不同的终端,而对于接入互联网的网络终端,例如个人电脑、笔记本电脑等,则通常使用网卡的物理地址(英文:Media Access Control,简称:MAC)来进行区分。

[0003] 然而,传统技术中,上述使用IMEI、IMSI或MAC作为识别信息的方式存在一定的问题。现有的操作系统中的一些应用框架和插件,例如Xpose框架或者008神器等软件可对手机终端上的IMEI、IMSI或MAC等标识信息进行修改,修改的方式可以是直接修改在各个特定的存储区域存储的IMEI、IMSI或MAC值,或者是修改读取IMEI、IMSI或MAC等标识信息的函数值,从而导致可通过修改某些终端的IMEI、IMSI或MAC来将其伪装成其他终端的身份进行相应的业务活动,也就是说,现有的用于区分终端的识别信息较易被篡改。

发明内容

[0004] 基于此,为解决上述提到的传统技术中以IMEI、IMSI或MAC等标识作为终端识别信息的方式容易被修改伪造而导致的安全性不足的技术问题,特提供了一种生成终端识别信息的方法。

[0005] 一种生成终端识别信息的方法,包括:

[0006] 接收终端识别信息生成指令;

[0007] 获取已安装应用列表,所述应用列表中包含有至少一个应用标识;

[0008] 将所述应用列表中的应用标识哈希映射成识别信息;

[0009] 将所述识别信息返回。

[0010] 此外,为解决上述提到的传统技术中以IMEI、IMSI或MAC等标识作为终端识别信息的方式容易被修改伪造而导致的安全性不足的技术问题,特提供了生成终端识别信息的装置。

[0011] 一种生成终端识别信息的装置,包括:

[0012] 指令接收模块,用于接收终端识别信息生成指令;

[0013] 列表获取模块,用于获取已安装应用列表,所述应用列表中包含有至少一个应用标识;

[0014] 识别信息映射模块,用于将所述应用列表中的应用标识哈希映射成识别信息;

[0015] 识别信息返回模块,用于将所述识别信息返回。

[0016] 实施本发明实施例,将具有如下有益效果:

[0017] 采用了上述生成终端识别信息的方法及装置之后,终端的识别信息不在依赖单一的容易遭到篡改的IMEI、IMSI或MAC,而是根据终端上已安装的应用的应用标识生成。对于不同的终端,若伪造者希望对终端的识别信息进行篡改,则需要对终端上的应用进行安装或卸载,与传统技术中简单的修改机器识别码的方式相比,耗时更多,从而更加不方便被伪造,从而使得在基于识别信息的终端数量统计的应用场景中,可有效的防止不法分子伪造统计数据。

附图说明

[0018] 为了更清楚地说明本发明实施例或现有技术中的技术方案,下面将对实施例或现有技术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。

[0019] 其中:

[0020] 图1为一个实施例中一种生成终端识别信息的方法的流程图;

[0021] 图2为一个实施例中一种将多个应用标识子集哈希映射成预设位数的识别信息的方法的流程图;

[0022] 图3为一个实施例中将多个应用标识子集哈希映射成预设位数的识别信息的过程原理示意图;

[0023] 图4为一个实施例中一种生成终端识别信息的装置的示意图;

[0024] 图5为一个实施例中运行前述第一种或第二种读取智能卡的方法的计算机设备的结构示意图。

具体实施方式

[0025] 下面将结合本发明实施例中的附图,对本发明实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例仅仅是本发明一部分实施例,而不是全部的实施例。基于本发明中的实施例,本领域普通技术人员在没有作出创造性劳动前提下所获得的所有其他实施例,都属于本发明保护的范围。

[0026] 如前所述,传统技术中通常使用移动设备的IMEI、IMSI、MAC等硬件信息作为识别信息来区分移动设备,但是这些硬件信息都是可以修改的。例如,Xposed框架是一款可以在不修改APK的情况下影响程序运行(修改系统)的框架服务,基于它可以制作出许多功能强大的模块,且在功能不冲突的情况下同时运作。基于Xposed框架可以向指定的程序提供虚假的IMEI、IMSI、MAC、屏幕分辨率、机型、SIM卡号、手机号码、地理位置、运营商编号或名称、手机OS版本等各种硬件接入环境参数。

[0027] 基于Xposed框架的典型工具有008K神器。在渠道推广中,每个不同的IMEI指纹(或者由IMEI+IMSI+MAC等信息联合构成的指纹)代表一个新用户,类似008K神器这样的工具被广泛地用于制造虚假用户(即渠道刷量)。现有的手机刷量工具可以屏蔽工具本身的APP包名或某些敏感的APP包名,避免该APP包名出现在手机的应用列表中,从而避免被检测到。

[0028] 例如,在统计应用的注册用户数时,通常需要通过使用IMEI、IMSI或MAC区分终端,

从而避免用户重复注册。而刷量用户可以在少量的手机上通过008K神器写入不同的IMEI、IMSI或MAC,即可对终端的识别信息进行篡改,从而可使用少量的手机注册较多次数,进而伪造注册用户的数量。

[0029] 为解决上述提到的传统技术中以IMEI、IMSI或MAC等标识作为终端识别信息的方式容易被修改伪造而导致的安全性不足的技术问题,特提供了一种生成终端识别信息的方法,该方法的实现可依赖于计算机程序,该计算机程序可运行于基于冯诺依曼体系的计算机系统之上,该计算机系统可以是智能手机、平板电脑、笔记本电脑、个人PC等计算机设备。

[0030] 具体的,如图1所示,生成终端识别信息的方法包括如下步骤:

[0031] 步骤S102:接收终端识别信息生成指令。

[0032] 终端识别信息生成指令用于通知终端生成终端识别信息。该终端识别信息可以由外部服务器发送的,也可以是由终端根据程序运行自行生成的。

[0033] 例如,若多个终端需要与服务器保持连接,服务器需要同时且异步地为该多个终端进行业务处理服务,则服务器需要对该多个终端进行区分。因此,在每个终端与服务器建立连接或在服务器注册时,服务器可以向该终端发送终端识别信息生成指令,通知终端返回终端识别信息(或者终端可在生成连接请求或注册请求时,触发生成终端识别信息生成指令,然后在连接请求中或注册请求中附加识别信息)。然后服务器创建与该识别信息对应的会话,当该终端掉线重连时,服务器可根据相同的识别信息查找到该会话,从而恢复掉线之前的业务处理进程。

[0034] 步骤S104:获取已安装应用列表,所述应用列表中包含有至少一个应用标识。

[0035] 已安装应用列表即为终端上已安装的应用程序列表或app列表。已安装应用列表中包含有已安装的应用程序或app的应用标识。

[0036] 例如,在Android(安卓)系统中,提供了PackageManger管理类,可通过调用PackageManger的接口函数获取安装了Android系统的终端上已安装的应用的包名(例如,日程表应用的报名即为com.android.calendar),该报名即可作为已安装应用的应用标识。

[0037] 而在IOS系统中,提供了LSApplicationWorkspace_class的运行时runtime获取已安装应用列表的类,或者也可通过读取com.apple.mobile.installation.plist来获取已安装应用列表。

[0038] 步骤S106:将所述应用列表中的应用标识哈希映射成识别信息。

[0039] 步骤S108:将所述识别信息返回。

[0040] 如前所述,若获取到的已安装应用列表中应用的应用标识(应用的包名)如下所示:

[0041] android;

[0042] com.android.backupconfirm;

[0043] com.android.bluetooth;

[0044] com.android.calendar;

[0045] com.android.certinstaller;

[0046] com.android.contacts;

[0047] com.android.defcontainer”

[0048]

[0049] 则,在本实施例中,可通过哈希映射,将上述获取到的多个应用标识通过预设的哈希算法映射成哈希值,然后将该哈希值作为识别信息输出。终端将应用列表中的应用标识归类为多个应用标识子集的步骤之前还可将所述应用列表中的应用标识排序。避免获取应用标识顺序不一致而导致的同一设备产生不同的识别信息。

[0050] 例如,可将上述多个应用标识连接成字符串,然后使用MD5算法将连接后得到的字符串映射为哈希序列值,然后将该哈希序列值作为终端的识别信息输出。

[0051] 再例如,可设置识别信息的预设长度,例如,可定义64位为识别信息的长度,然后通过预设的哈希算法,遍历已安装应用列表中的应用标识,将遍历到的应用标识依次映射到该识别信息的某一位或某些位上。然后输出遍历结束后得到的识别信息。

[0052] 优选的,终端将所述应用列表中的应用标识哈希映射成识别信息的步骤包括:

[0053] 将所述应用列表中的应用标识归类为多个应用标识子集,所述应用标识子集中包含至少一个应用标识;将所述多个应用标识子集哈希映射成预设位数的识别信息。

[0054] 如上例中,可将获取到的已安装应用列表中的应用标识分组,分组后的应用标识为

[0055] Android,com.android.backupconfirm;

[0056] com.android.backupconfirm,com.android.bluetooth;

[0057] com.android.bluetooth,com.android.calendar;

[0058] com.android.calendar,com.android.certinstaller;

[0059] com.android.certinstaller,com.android.contacts;

[0060] com.android.contacts,com.android.defcontainer”;

[0061] 终端可遍历归类得到的应用标识子集,将遍历到的应用标识子集中包含的应用标识的字符串进行拼接,根据预设的哈希算法将所述拼接得到的字符串映射成所述预设位数的二进制码。

[0062] 如前所述,可使用MD5码对拼接后得到的字符串进行哈希映射,也可遍历拼接应用标识得到的字符串,将遍历到的字符串依次映射到预设长度的识别信息的某一位或某些位上,然后输出遍历结束后得到的识别信息。

[0063] 进一步的,本实施例中,还提出了一种将多个应用标识子集哈希映射成预设位数的识别信息的方法,具体的,如图2所示,包括:

[0064] 步骤S202:为归类得到的每个所述应用标识子集设置权重参数。

[0065] 权重系数为用于表示某个应用标识在哈希映射过程中的重要性的参数,在本实施例中,可全部设置为相同的值;在其他实施例中,还可根据应用标识子集对应的应用类型或者应用标识子集中包含的应用标识的个数设置权重系数。

[0066] 例如,若应用分为系统自带应用和用户安装的应用,则可将系统自带应用的应用标识的权重系数设置为1,将用户安装的应用的权重系数设置为2。若预设的应用标识子集的个数为2,则若某个应用标识子集中包含两个系统自带应用的应用标识,则其权重系数为1;若某个应用标识子集中包含两个用户安装应用的应用标识,则其权重系数为2;若某个应用标识子集中包含一个系统自带应用的应用标识和一个用户安装应用的应用标识,则其权重系数为1.5。

[0067] 步骤S204:将归类得到的每个所述应用标识子集通过预设的哈希算法映射成所述

预设位数的二进制码。

[0068] 也就是说,在本实施例中,哈希映射包括两部分的映射过程,在第一部分映射过程中,需要先使用预设的哈希算法,将每个应用标识子集映射成一个预设位数的二进制码,从而得到多个二进制码的序列,然后再将该二进制码的序列映射成最终输出的识别信息。

[0069] 如前所述,在步骤S204中完成了本实施例中第一部分的哈希映射过程,得到了与每个应用标识子集对应的二进制码构成的序列,已下结合图3所示的哈希映射过程详细阐述第二部分的哈希映射过程,即将得到的二进制码的序列映射成最终输出的识别信息。

[0070] 步骤S206:根据每个所述应用标识子集的二进制码和权重系数生成每个所述应用标识子集对应的权重参数序列,且在权重参数序列中,序列值的绝对值均为对应该权重参数序列的应用标识子集的权重参数,对应该权重参数序列的二进制码在每一位上的值确定该权重参数序列中相应位置的序列值的正负符号。

[0071] 在图3中, W_1 至 W_n 为每个所述应用标识子集对应的权重系数; W_1 对应的100110、 W_2 对应的110000...、 W_n 对应的001001即为每个应用标识子集对应的二进制码构成的序列。在该例中预设的识别信息的长度为6位。

[0072] 也就是说,编号为1的应用标识子集对应的权重系数为 W_1 ,且编号为1的应用标识子集通过步骤S204进行哈希映射后得到的预设长度的哈希值为100110,则得到的编号为1的应用标识子集对应的权重参数序列为:

[0073] $+W_1, -W_1, -W_1, +W_1, +W_1, -W_1$;

[0074] 也就是说,编号为1的应用标识子集对应的权重系数决定了相应的权重参数序列中各个序列值的绝对值为 W_1 ;编号为1的应用标识子集对应的二进制码100110决定了相应的权重参数序列中各个序列值的符号为+,-,-,+,+,-;

[0075] 步骤S208:生成初始化的所述预设位数的识别信息。

[0076] 如前例中,预设的识别信息的长度为6位,则预设的识别信息可以为000000。

[0077] 步骤S210:遍历所述识别信息的位,获取每个所述应用标识子集的权重参数序列在所述遍历到的位对应的序列位置的序列值,计算获取到的序列值的和,根据所述计算得到的和的符号设置所述识别信息在该遍历到的位的数值。

[0078] 再次参考图3,对于6位的识别信息的第1位,可先获取每个应用标识子集对应的权重参数序列中的第1位。

[0079] 例如,对于编号为1的应用标识子集对应的权重参数序列:

[0080] $+W_1, -W_1, -W_1, +W_1, +W_1, -W_1$,第一位为 $+W_1$;

[0081] 对于编号为2的应用标识子集对应的权重参数序列:

[0082] $+W_2, +W_2, -W_1, -W_1, -W_1, -W_1$,第一位为 $+W_2$;

[0083] 对于编号为n的应用标识子集对应的权重参数序列:

[0084] $-W_n, -W_n, +W_n, -W_n, -W_n, +W_n$,第一位为 $-W_n$;

[0085] 则最终得到的识别信息的第1位即对应 $+W_1+W_2\cdots-W_n$ 的正负号,也就是说,若 $+W_1+W_2\cdots-W_n$ 的值大于0,则最终得到的识别信息的第1位为1;若 $+W_1+W_2\cdots-W_n$ 的值小于或等于0,则最终得到的识别信息的第1位为0。

[0086] 以此类推,对于6位的识别信息的第2位对应 $-W_1+W_2\cdots-W_n$ 的正负号。也就是说,若 $-W_1+W_2\cdots-W_n$ 的值大于0,则最终得到的识别信息的第2位为1;若 $-W_1+W_2\cdots-W_n$ 的值小于或等于

0,则最终得到的识别信息的第2位为0。

[0087] 以此类推,对于6位的识别信息的第6位对应 $-W_1-W_2\cdots+W_n$ 的正负号。也就是说,若 $-W_1-W_2\cdots+W_n$ 的值大于0,则最终得到的识别信息的第6位为1;若 $-W_1-W_2\cdots+W_n$ 的值小于或等于0,则最终得到的识别信息的第6位为0。

[0088] 若在本实施例中,最终得到的序列值的和构成的序列为如图3所示的13、108、-22、-5、32、55,则相应的正负号序列为正、正、负、负、正、正,那么最终输出的识别信息即为110011。

[0089] 以下将结合一个具体的应用场景来阐述本方法在实际应用中所起的作用。

[0090] 在一个应用平台(例如应用宝、应用市场、软件管家等)的场景中,应用平台中注册有多款应用程序。该应用程序可以由第三方厂家发布,用户访问应用平台之后即可浏览到该发布的应用。应用平台对上述注册的应用具有管理和统计功能。

[0091] 例如,应用平台可统计注册的应用的下载安装数量,且该下载安装数量与终端对应,也就是说,应用平台的服务器统计的是下载或安装了某个应用的终端的个数,因此需要终端提供识别信息使得应用平台的服务器能够对终端加以区分从而可方便地进行统计。

[0092] 在传统技术中,某些第三方应用的发布者,为了营造其发布的应用下载量较大或安装量较大的假象,使用了刷码的方式,即使用少量手机,在短时间内通过写入不同的IMEI码并重复下载来使得应用平台的服务器统计的下载安装次数不准确。

[0093] 而在本应用场景中,应用平台的服务器在用户下载或安装应用时,通知终端上传终端的识别信息,而该识别信息是根据终端的已安装应用列表生成的,第三方应用的发布者无法通过写入IMEI码来伪造终端的识别信息,从而使得应用平台的服务器统计的下载安装数据更加真实。也就是说,对于已统计过的终端,伪造统计信息的用户必须要对该终端进行安装应用或卸载应用的操作(根据哈希算法的不同可能需要安装和卸载的应用不止一个),因此,对于短时间内需要伪造较大下载安装数据量的伪造需求而言,根据终端的已安装应用列表生成的识别信息有效地防止了伪造统计数据的情况,从而使得统计数据更加真实。

[0094] 为解决上述提到的传统技术中以IMEI、IMSI或MAC等标识作为终端识别信息的方式容易被修改伪造而导致的安全性不足的技术问题,特提供了一种生成终端识别信息的装置,该装置包括指令接收模块102、列表获取模块104、识别信息映射模块106以及识别信息返回模块108,其中:

[0095] 指令接收模块102,用于接收终端识别信息生成指令;

[0096] 列表获取模块104,用于获取已安装应用列表,所述应用列表中包含有至少一个应用标识;

[0097] 识别信息映射模块106,用于将所述应用列表中的应用标识哈希映射成识别信息;

[0098] 识别信息返回模块108,用于将所述识别信息返回。

[0099] 可选的,在本实施例中,识别信息映射模块106还用于:将所述应用列表中的应用标识归类为多个应用标识子集,所述应用标识子集中包含至少一个应用标识;将所述多个应用标识子集哈希映射成预设位数的识别信息。

[0100] 可选的,在本实施例中,识别信息映射模块106还用于:为归类得到的每个所述应用标识子集设置权重参数;将归类得到的每个所述应用标识子集通过预设的哈希算法映射

成所述预设位数的二进制码；根据每个所述应用标识子集的二进制码和权重系数生成每个所述应用标识子集对应的权重参数序列，且在权重参数序列中，序列值的绝对值均为对应该权重参数序列的应用标识子集的权重参数，对应该权重参数序列的二进制码在每一位上的值确定该权重参数序列中相应位置的序列值的正负符号；生成初始化的所述预设位数的识别信息；遍历所述识别信息的位，获取每个所述应用标识子集的权重参数序列在所述遍历到的位对应的序列位置的序列值，计算获取到的序列值的和，根据所述计算得到的和的符号设置所述识别信息在该遍历到的位的数值。

[0101] 可选的，在本实施例中，识别信息映射模块106还用于：遍历归类得到的应用标识子集，将遍历到的应用标识子集中包含的应用标识的字符串进行拼接，根据预设的哈希算法将所述拼接得到的字符串映射成所述预设位数的二进制码。

[0102] 可选的，在本实施例中，识别信息映射模块106还用于：将所述应用列表中的应用标识排序。

[0103] 实施本发明实施例，将具有如下有益效果：

[0104] 采用了上述生成终端识别信息的方法及装置之后，终端的识别信息不在依赖单一的容易遭到篡改的IMEI、IMSI或MAC，而是根据终端上已安装的应用的应用标识生成。对于不同的终端，若伪造者希望对终端的识别信息进行篡改，则需要对终端上的应用进行安装或卸载，与传统技术中简单的修改机器识别码的方式相比，耗时更多，从而更加不方便被伪造，从而使得在基于识别信息的终端数量统计的应用场景中，可有效的防止不法分子伪造统计数据。

[0105] 在一个实施例中，如图5所示，图5展示了一种运行上述生成终端识别信息的方法的基于冯诺依曼体系的计算机系统的终端10。该计算机系统可以是智能手机、平板电脑、掌上电脑，笔记本电脑或个人电脑等终端设备。具体的，可包括通过系统总线连接的外部输入接口1001、处理器1002、存储器1003和输出接口1004。其中，外部输入接口1001可选的可包括鼠标10012、键盘10014或触摸板10016中的至少一个。存储器1003可包括外存储器10032（例如硬盘、光盘或软盘等）和内存储器10034。输出接口1004可包括显示屏10042和音响/喇叭10044等设备。

[0106] 在本实施例中，本方法的运行基于计算机程序，该计算机程序的程序文件存储于前述基于冯诺依曼体系的计算机系统10的外存储器10032中，在运行时被加载到内存储器10034中，然后被编译为机器码之后传递至处理器1002中执行，从而使得基于冯诺依曼体系的计算机系统10中形成逻辑上的指令接收模块102、列表获取模块104、识别信息映射模块106以及识别信息返回模块108。且在上述生成终端识别信息的方法执行过程中，输入的参数均通过外部输入接口1001接收，并传递至存储器1003中缓存，然后输入到处理器1002中进行处理，处理的结果数据或缓存于存储器1003中进行后续地处理，或被传递至输出接口1004进行输出。

[0107] 以上所揭露的仅为本发明较佳实施例而已，当然不能以此来限定本发明之权利范围，因此依本发明权利要求所作的等同变化，仍属本发明所涵盖的范围。

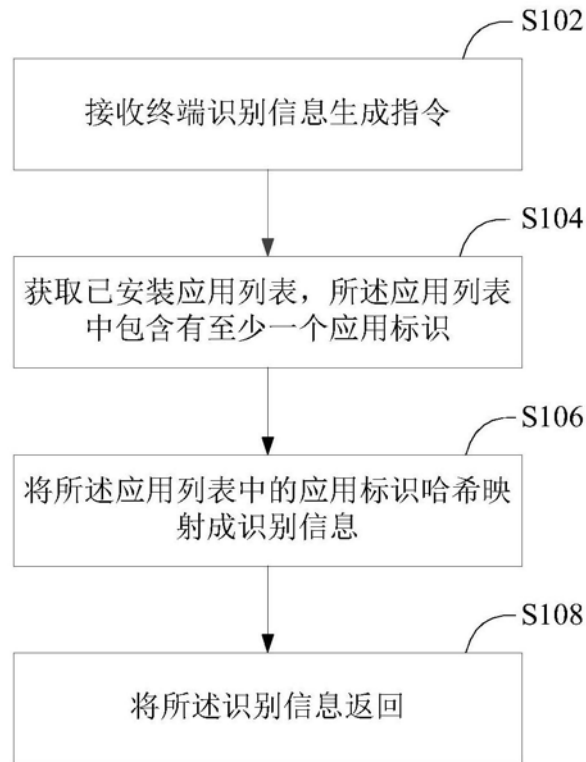


图1

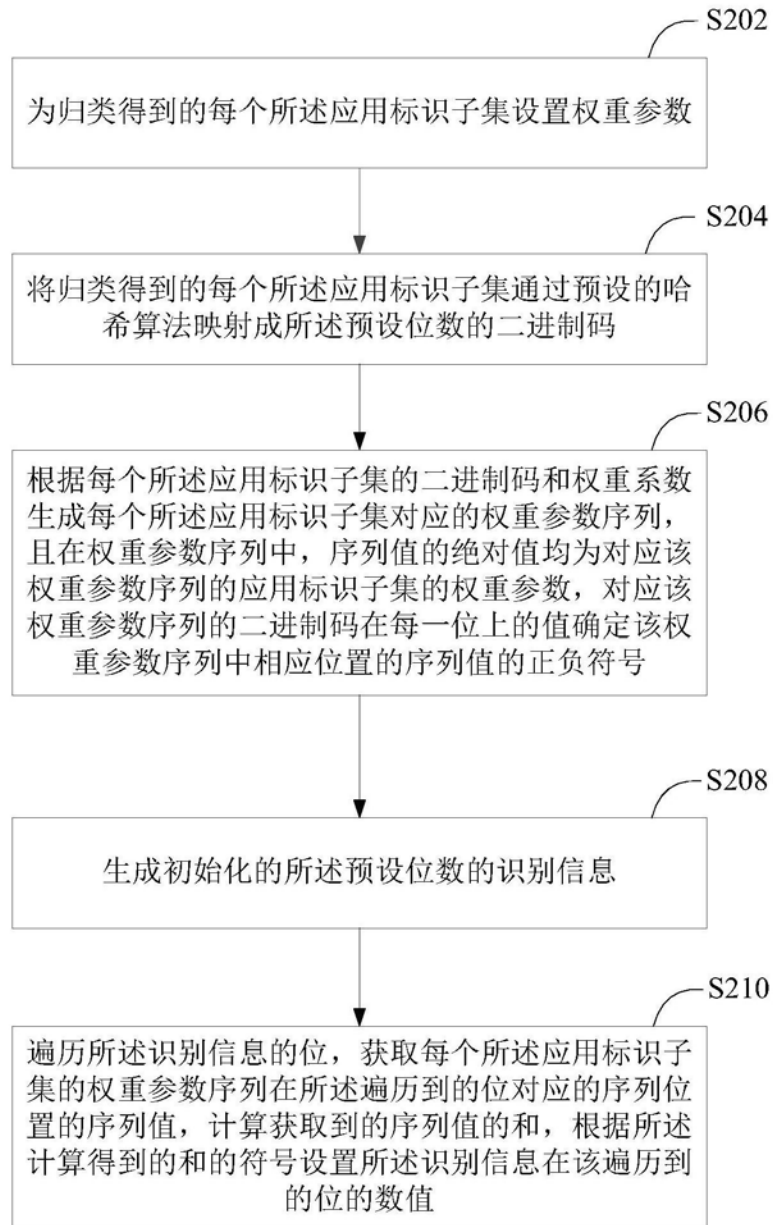


图2

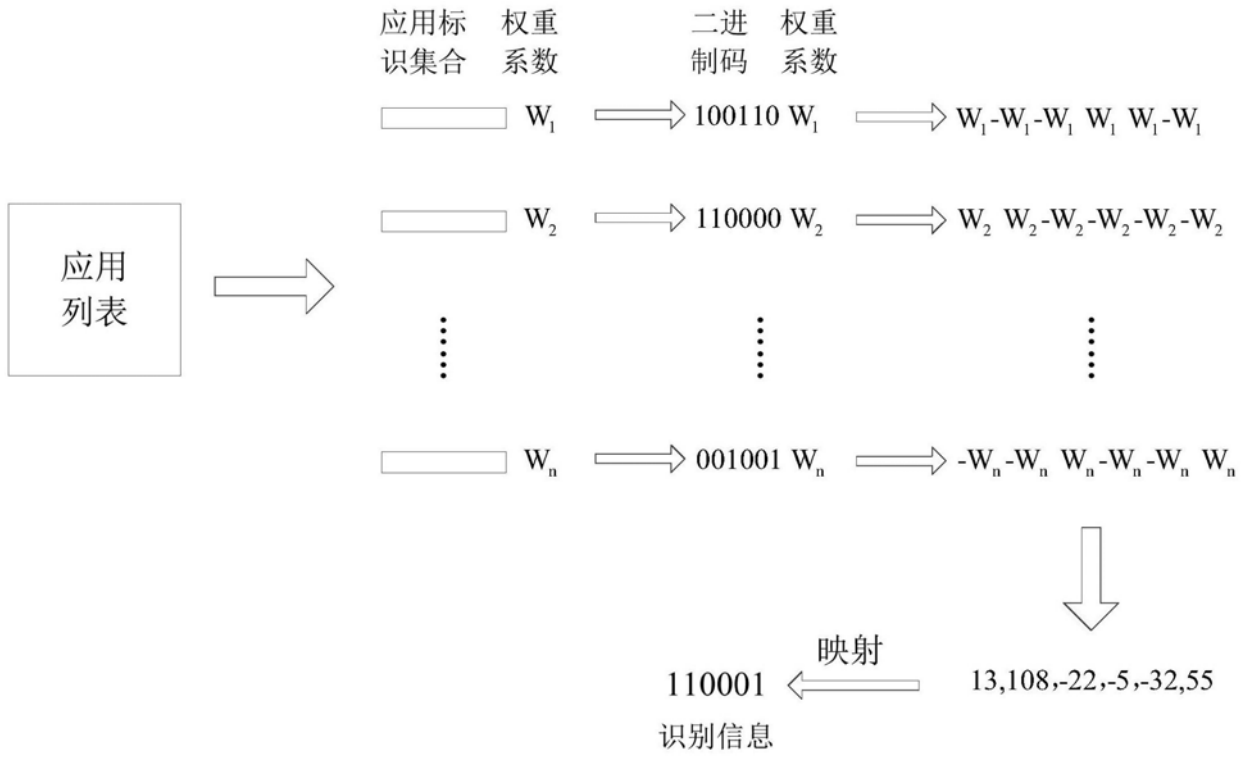


图3

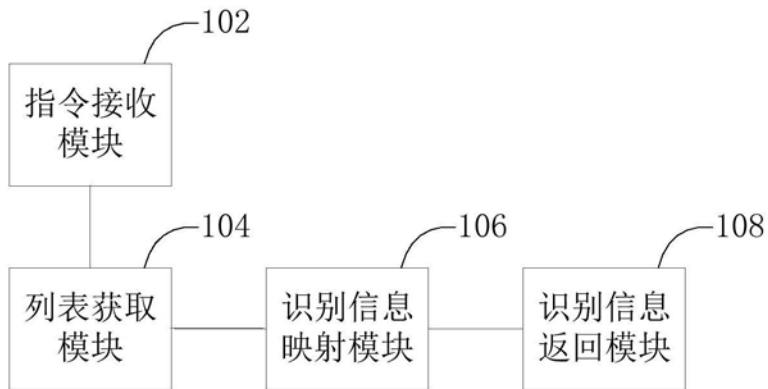


图4

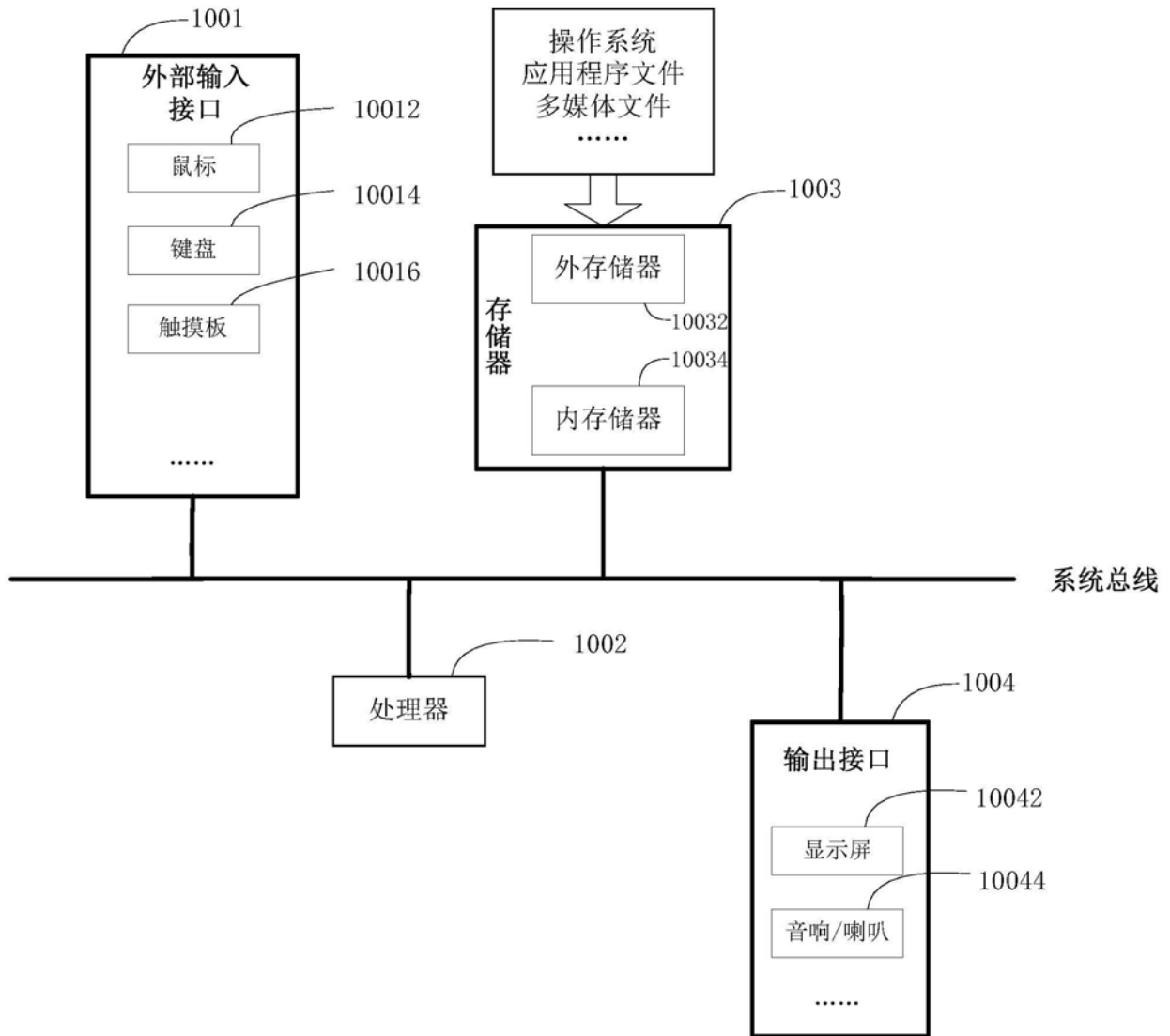


图5