

# (12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织  
国际局

(43) 国际公布日  
2020 年 1 月 30 日 (30.01.2020)



(10) 国际公布号  
**WO 2020/019482 A1**

(51) 国际专利分类号:  
**G06F 21/52** (2013.01)

(21) 国际申请号: PCT/CN2018/107745

(22) 国际申请日: 2018 年 9 月 26 日 (26.09.2018)

(25) 申请语言: 中文

(26) 公布语言: 中文

(30) 优先权:  
201810841834.3 2018 年 7 月 27 日 (27.07.2018) CN

(71) 申请人: 平安科技(深圳)有限公司(PING AN TECHNOLOGY (SHENZHEN) CO., LTD.) [CN/CN];  
中国广东省深圳市福田区福田街道福安社区益田路 5033 号平安金融中心 23 楼, Guangdong 518000 (CN)。

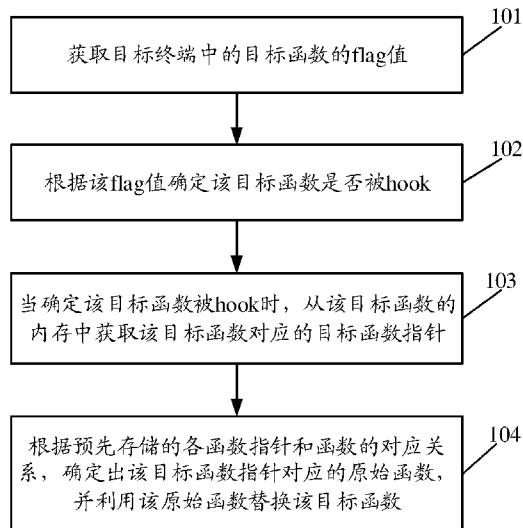
(72) 发明人: 刘瑞恺(LIU, Ruikai); 中国广东省深圳市福田区福田街道福安社区益田路 5033 号平安金融中心 23 楼, Guangdong 518000 (CN)。

(74) 代理人: 广州三环专利商标代理有限公司(SCIHEAD IP LAW FIRM); 中国广东省广州市越秀区先烈中路 80 号汇华商贸大厦 1508 室, Guangdong 510070 (CN)。

(81) 指定国(除另有指明, 要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL,

(54) Title: FUNCTION HOOK DETECTION METHOD, FUNCTION HOOK DETECTION DEVICE, AND COMPUTER-READABLE MEDIUM

(54) 发明名称: 一种函数hook检测方法、检测设备及计算机可读介质



- 101 Acquire a flag value of an objective function in a target terminal  
102 Determine whether the objective function has been hooked according to the flag value  
103 When it is determined that the objective function has been hooked, acquire, from a memory of the objective function, an objective function pointer corresponding to the objective function  
104 Determine, according to pre-stored correlations between various function pointers and functions, an original function corresponding to the objective function pointer, and use the original function to replace the objective function

图 1

(57) Abstract: A function hook detection method, a function hook detection device, and a computer-readable medium. The method comprises: when it is detected that a target terminal has an Xposed plug-in installed, acquiring a flag value of an objective function in the target terminal (101); determining whether the objective function has been hooked according to the flag value (102); when it is determined that the objective function has been hooked, acquiring, from a memory of the objective function, an objective function pointer corresponding to the objective function (103); and determining, according to pre-stored correlations between various function pointers and functions, an original function corresponding to the objective function pointer, and using the original function to replace the objective function (104). The method facilitates avoiding a false positive of a hook behavior, and enhancing the accuracy of hook detection.

SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG,  
US, UZ, VC, VN, ZA, ZM, ZW。

- (84) 指定国(除另有指明, 要求每一种可提供的地区  
保护): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ,  
NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM,  
AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG,  
CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU,  
IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT,  
RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI,  
CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。

本国际公布:

- 包括国际检索报告(条约第21条(3))。

(57) 摘要: 一种函数hook检测方法、检测设备及计算机可读介质, 其中, 该方法包括: 当检测到目标终端安装有Xposed插件时, 获取所述目标终端中的目标函数的flag值(101); 根据该flag值确定该目标函数是否被hook(102); 当确定该目标函数被hook时, 从该目标函数的内存中获取该目标函数对应的目标函数指针(103); 根据预先存储的各函数指针和函数的对应关系, 确定出该目标函数指针对应的原始函数, 并利用该原始函数替换该目标函数(104)。该方法有助于避免hook行为的误报, 提升hook检测的准确性。

## 一种函数 hook 检测方法、检测设备及计算机可读介质

本申请要求于 2018 年 07 月 27 日提交中国专利局、申请号为 201810841834.3、申请名称为“一种函数 hook 检测方法、检测设备及计算机可读介质”的中国专利申请的优先权，其全部内容通过引用结合在本申请中。

5

### 技术领域

本申请涉及通信技术领域，尤其涉及一种函数 hook 检测方法、检测设备及计算机可读介质。

### 10 背景技术

Xposed 是一种开源框架，其提供了一种可以在不修改 Android 安装包（Android Package, APK）源码的情况下，通过自己编写的模块或代码来影响程序运行的框架服务。一些非法分子会利用 Xposed 插件篡改设备信息，而目前市面上的设备风险识别类型的产品对上述的篡改行为普遍无法识别，一般仅  
15 是通过检测终端中是否安装 Xposed 插件来确定是否存在 hook 行为。该仅检测 Xposed 插件是否安装来确定 hook 行为的方式，误报率较高。

### 发明内容

本申请提供一种函数 hook 检测方法、检测设备及计算机可读介质，有助于  
20 于避免 hook 行为的误报，提升 hook 检测的准确性。

第一方面，本申请提供了一种函数 hook 检测方法，包括：

当检测到目标终端安装有 Xposed 插件时，获取所述目标终端中的目标函数的 flag 值，所述 flag 值用于标记所述目标函数的状态；

根据所述 flag 值确定所述目标函数是否被 hook；

25 当确定所述目标函数被 hook 时，从所述目标函数的内存中获取所述目标函数对应的目标函数指针；

根据预先存储的各函数指针和函数的对应关系，确定出所述目标函数指针对应的原始函数，并利用所述原始函数替换所述目标函数。

30 第二方面，本申请提供了一种检测设备，该检测设备包括用于执行上述第一方面的方法的单元。

第三方面，本申请提供了另一种检测设备，包括处理器、用户接口、通信接口和存储器，所述处理器、用户接口、通信接口和存储器相互连接，其中，所述存储器用于存储支持检测设备执行上述方法的计算机程序，所述计算机程序包括程序指令，所述处理器被配置用于调用所述程序指令，执行上述第一方

面的方法。

第四方面，本申请提供了一种计算机可读存储介质，所述计算机存储介质存储有计算机程序，所述计算机程序包括程序指令，所述程序指令当被处理器执行时使所述处理器执行上述第一方面的方法。

- 5       本申请实施例能够通过获取终端中待进行 hook 检测的函数的 flag 值，以根据该 flag 值确定该函数是否被 hook，并在确定该函数被 hook 时，能够通过获取该函数对应的函数指针，以根据该函数指针确定出真实的原始函数并替换该被 hook 的函数，这就有助于避免 hook 行为的误报，提升 hook 检测的准确性，并能够通过还原真实函数，提升终端安全性。

10

#### 附图说明

为了更清楚地说明本申请实施例技术方案，下面将对实施例描述中所需要使用的附图进行说明。

图 1 是本申请实施例提供的一种函数 hook 检测方法的流程示意图；

- 15       图 2 是本申请实施例提供的另一种函数 hook 检测方法的流程示意图；

图 3 是本申请实施例提供的又一种函数 hook 检测方法的流程示意图；

图 4 是本申请实施例提供的一种检测设备的结构示意图；

图 5 是本申请实施例提供的另一种检测设备的结构示意图。

#### 20   具体实施方式

下面将结合本申请实施例中的附图，对本申请实施例中的技术方案进行描述。

- 25       本申请的技术方案可应用于检测设备中，该检测设备可包括各种终端、服务器或与终端连接的风险识别产品（设备）等等，用于对终端中的 hook 行为进行检测。本申请涉及的终端可以是手机、电脑、平板、智能手表等，本申请不做限定。该终端中部署有 Xposed 开源框架（或称为“安装有 Xposed 插件”），或其他开源框架，该开源框架中经常存在 hook 行为。其中，hook 即“挂钩”或“钩子”，其可以改变函数执行的结果，比如应用程序编程接口（Application Programming Interface, API）函数的执行结果，由此，非法分子可通过 hook  
30       技术修改设备信息，以达到非法目的。通过本申请能够及时发现该 hook 行为。

在本申请中，函数还可以叫做不同的名称，比如其还可以叫做 Method、ArtMethod、方法、结构、Method 结构、ArtMethod 结构或者其余名称。

具体的，本申请可通过获取终端中的函数的 flag 值，以根据该 flag 值确定

该函数是否被 hook，这就有助于避免 hook 行为的误报，提升 hook 检测的准确性。在获取函数的 flag 值之前，可以通过检测该终端是否安装有 Xposed 或其他开源框架或插件，并在确定安装有该 Xposed 或其他开源框架或插件时，再获取该函数的 flag 值，以减小设备开销。进一步的，在确定某一函数被 hook 时，能够通过获取该函数对应的函数指针，并根据该函数指针确定出其对应的真实的原始函数以替换该被 hook 的函数，从而能够实现还原真实函数，提升终端安全性。以下分别详细说明。

请参见图 1，图 1 是本申请实施例提供的一种函数 hook 检测方法的流程示意图。具体的，如图 1 所示，该函数 hook 检测方法可以包括以下步骤：

10 101、获取目标终端中的目标函数的 flag 值。

其中，该目标终端可以是指需要进行 hook 检测的任一终端，比如与风险检测设备连接的终端，或者处于特定风控场景下的终端，或者触发（比如通过预设按键或手势或预设的其他触发方式）了 hook 检测的终端，等等，本申请不做限定。该风控场景可以包括登录场景、交易场景、APP 优惠领取场景等等。

15 该目标函数可以是指任一 API 函数，在本申请中，检测设备可对终端中的所有函数如所有 API 函数进行 hook 检测；或者可以仅对该终端中的部分指定函数，比如预设的函数列表中的函数，或者一些容易被 hook 的函数等等进行 hook 检测，以减小设备的检测开销，提升 hook 检测效率。该指定函数通常是特定需求或风控场景所需的信息函数。例如，该指定函数可以为定位函数，即  
20 定位信息对应的函数，为了防止打车软件伪造定位，可对定位函数如 GPS 函数进行 hook 检测；又如，该指定函数可以为设备 ID 的函数，为了获取设备的真实的设备 ID，可对获取该设备 ID 的函数进行 hook 检测；又如，该指定函数可以为终端环境信息如电池信息、Wi-Fi 信息、4G 网络基站信息、已经安装的 App 列表、通讯录内容、开机时间等的函数，为了判断目标终端是否运行  
25 在模拟器环境，可以对目标终端的环境信息，例如对电池信息、Wi-Fi 信息、4G 网络基站信息、已经安装的 App 列表、通讯录内容、开机时间等的函数进行 hook 检测，等等，此处不一一列举。

可选的，在获取目标终端中的目标函数即待进行 hook 检测函数（需要进行 hook 检测的函数）的 flag 值之前，检测设备还可检测该目标终端是否安装  
30 有 Xposed 插件（部署有 Xposed 框架），并在确定安装有该 Xposed 插件时，再获取该目标函数的 flag 值。从而能够实现特定环境下的 hook 检测，提升 hook 检测的可靠性，降低设备开销。

102、根据该 flag 值确定该目标函数是否被 hook。

其中，该 flag 值可用于标记所述目标函数的状态，该状态可以是指是否被

篡改的状态，或者可以是指读写状态、阻塞与非阻塞状态、退出进程或程序的状态和/或更改文件的内容的状态等等，从而能够根据该 flag 值确定出该目标函数是否被 hook。具体的，每一个函数都有对应的 flag，该 flag 为一个变量，当某一函数被篡改时，该函数对应的 flag 会发生改变。由此，检测设备可通过检测函数的 flag 是否发生改变，来确定该函数是否被 hook，比如可将获取的 flag（或按照预设逻辑算法处理后的 flag）值与默认值（未被篡改时的固定值）进行比较，如果 flag（或按照预设逻辑算法处理后的 flag）的值发生改变，即与该默认值不同，则表明该函数被 hook，也即该函数对应的设备信息被篡改。其中，该 flag 的值可以是存储于该目标函数对应的内存中。

- 5 检测函数的 flag 是否发生改变，来确定该函数是否被 hook，比如可将获取的 flag（或按照预设逻辑算法处理后的 flag）值与默认值（未被篡改时的固定值）进行比较，如果 flag（或按照预设逻辑算法处理后的 flag）的值发生改变，即与该默认值不同，则表明该函数被 hook，也即该函数对应的设备信息被篡改。其中，该 flag 的值可以是存储于该目标函数对应的内存中。
- 10 103、当确定该目标函数被 hook 时，从该目标函数的内存中获取该目标函数对应的目标函数指针。

其中，该函数指针和被 hook 的函数是存储于同一块内存的不同字段中的，且不同函数指针和原始函数存在映射关系，或者说不同函数指针和原始函数的存储地址存在映射关系。

- 15 可选的，在确定该目标函数被 hook 之后，检测设备还可对该被 hook 的目标函数进行还原，以便于确定出该目标函数对应的真实设备信息。具体的，在确定某一函数如该目标函数被 hook 之后，可从其内存中快速获取该目标函数对应的函数指针，即上述的目标函数指针，以便于根据该目标函数指针确定出该目标函数对应的原始函数如原生 API，即未被 hook 的真实函数。

- 20 104、根据预先存储的各函数指针和函数的对应关系，确定出该目标函数指针对应的原始函数，并利用该原始函数替换该目标函数。

- 在确定出该目标函数对应的内存中的目标函数指针之后，即可进一步确定出该目标函数指针对应的原始函数，即真实的 Method。进而可通过该原始函数替换该目标函数，实现对被 hook 的函数的还原。从而检测设备可通过该原始函数确定出该目标终端的真实设备信息，以基于真实的设备信息对该目标终端的操作进行控制，比如指示目标终端输出提示，要求用户输入验证信息；禁止用户在对应的风控场景的一切请求访问操作，等等，此处不一一列举。
- 25

- 应理解，该内存中存储的原始函数指针是不会被篡改的，根据 Xposed 插件的工作原理，在篡改目标函数之前，会将函数的原始信息备份下来，并保存在内存中的特定地址，即该目标函数指针指向的地址。而一旦这些备份信息也被篡改，那 Xposed 插件将无法正常工作。因此，在该目标函数指针指向的特定地址获取到的原始函数，一定是正确的函数，其不会被篡改。
- 30

在本申请实施例，检测设备可通过获取终端中的函数的 flag 值，以根据该 flag 值确定该函数是否被 hook，并在确定该函数被 hook 时，能够通过获取

该函数对应的函数指针,以根据该函数指针确定出真实的原始函数并替换该被 hook 的函数,这就有助于避免 hook 行为的误报,提升 hook 检测的准确性,并能够通过还原真实函数,提升终端安全性。

请参见图 2,图 2 是本申请实施例提供的另一种函数 hook 检测方法的流程示意图。具体的,如图 2 所示,该函数 hook 检测方法可以包括以下步骤:

201、获取目标终端的当前设备信息。

可选的,该当前设备信息可包括该目标终端的型号、品牌、系统版本、风控场景信息等设备信息中的一项或多项。其中,该风控场景信息可以包括风控场景标识和/或场景描述信息等等,用于指示终端所在的风控场景。例如,该  
10 风控场景标识可以包括登录标识、交易标识等等;又如,该场景描述信息可以包括终端处于登录状态的描述信息、终端处于交易状态的描述信息等等,此处不一一列举。

202、根据预置的不同设备信息和函数列表的对应关系,确定出该当前设备信息对应的函数列表。

其中,该设备信息与上述的当前设备信息相对应,其可包括终端的型号、品牌、系统版本、风控场景信息等信息中的一项或多项。每一个函数列表包括对应的设备信息下被 hook 的频率较高的至少一个函数,比如被 hook 的频率最高的前 M (M 为大于 0 的整数,如取 8) 个函数,或者被 hook 的频率大于预设频率阈值的函数;和/或,包括对应的设备信息下被 hook 的次数较多的函数,  
20 比如被 hook 的次数最多的前 N (N 为大于 0 的整数,如取 10) 个函数,或者被 hook 的次数大于预设数目阈值的函数,等等。

203、获取该目标终端中的目标函数的 flag 值,该目标函数为该当前设备信息对应的函数列表中的函数。

具体的,可预先根据不同设备信息对历史记录中被 hook 的函数进行分组,确定出每组设备信息下对应的函数分组,并根据该函数分组预先设置得到多个函数列表,即每个函数列表包括一组设备信息对应的函数分组,以预置得到不同设备信息和函数列表的对应关系。进而检测设备可根据当前设备信息查找出其对应的函数列表后,对该函数列表中的函数进行 hook 检测,而无需针对终端中的所有函数进行 hook 检测,这就提升了 hook 检测的效率,减小了设备开  
30 销。

例如,该设备信息为终端的系统版本,并预先配置有不同系统版本对应的多个函数列表。则检测设备在对目标终端进行 hook 检测时,可以通过获取该目标终端的当前系统版本,进而查找出该当前系统版本对应的函数列表,该函数列表中的函数即为需要进行 hook 函数的函数。

又如,该设备信息为终端的风控场景信息,并预先配置有不同风控场景信息对应的多个函数列表。则检测设备在对目标终端进行 hook 检测时,可以通过获取该目标终端的当前风控场景信息,进而查找出该当前风控场景信息对应的函数列表,该函数列表中的函数即为需要进行 hook 函数的函数。

5 又如,该设备信息为终端的型号和风控场景信息,并预先配置有不同型号和风控场景信息对应的多个函数列表。则检测设备在对目标终端进行 hook 检测时,可以通过获取该目标终端的型号和当前风控场景信息,进而查找出与该目标终端的型号和当前风控场景信息对应的函数列表,该函数列表中的函数即为需要进行 hook 函数的函数。

10 204、根据该 flag 值确定该目标函数是否被 hook。

可选的,在根据该 flag 值确定该目标函数是否被 hook 时,检测设备可以将该 flag 值中的预设位置处的字符与预设的固定字符进行比较;当比较得到该预设位置处的字符与该固定字符不同时,确定该目标函数被 hook。其中,该预设位置处的字符的字符数与该固定字符的字符数相同,以便于匹配比较。也

15 就是说,该 flag 的值发生改变可以是指该 flag 的一位或多位发生改变,且该一位或多位可以是指 flag 的预设位置处的一位或多位。从而检测设备可以通过将获取的 flag 值预设位置处的一位或多位与未被篡改时的固定字符进行比较,如果 flag 值的该一位或多位发生改变,即 flag 值的一位或多位与该固定字符不同,则表明该函数被 hook,即该函数对应的设备信息被篡改。

20 例如,针对 Android 版本在 4.4 以上及 5.0 以下的系统或其他系统,有的 Xposed 插件对某函数进行 hook 时,会将该函数的 flag 值的固定位置处的 1 位(bit)设置为 1;而正常未被篡改的函数,flag 值的该位是 0(即上述的固定字符)。因此,可通过检测函数的 flag 值的该固定位是否是 0,就可以知道该函数是否被 Xposed 插件进行了 hook。也即,如果该测函数的 flag 值的该固定位不为 0,

25 即可表明该函数被 hook,该函数被篡改。

可选的,在根据该 flag 值确定该目标函数是否被 hook 时,检测设备还可以按照预设的逻辑算法对该 flag 值进行逻辑运算,以得到运算结果值,其中,该逻辑算法是根据预设字符串和系统中的原生函数执行时的跳转地址确定的;当该运算结果值为正整数时,确定该目标函数被 hook。也就是说,还可将按

30 照预设逻辑算法对 flag 处理后的值与未被篡改时的固定字符如 0 进行比较,如果处理后的该值发生改变,即不为 0,比如为某一正整数时,则表明该函数被 hook。

例如,针对 Android 版本在 5.0 及其以上的系统或其他系统,如果按照逻辑算法如逻辑算式 `EntryPointFromJni && AccessFlags & 0x10000000` 结果等于

正整数,则可表明该函数被篡改;如果该逻辑算式结果等于 0(即为固定字符),则可表明该函数未被篡改。其中,该 EntryPointFromJni 可以是指原生函数如 native 函数执行时的跳转地址,AccessFlags 即为上述的 flag。

5 进一步可选的,检测设备在根据该 flag 值确定该目标函数是否被 hook 之前,还可确定该目标终端当前使用的系统版本,进而根据该目标终端的系统版本去选择根据该 flag 值确定该目标函数是否被 hook 的方式,以提升 hook 检测的效率。该系统版本和 hook 检测的方式的对应关系可预先设置得到。例如,在上述的 Android 版本 4.4 以上及 5.0 以下系统,检测设备可根据 flag 值中的预设位置处的字符与预设的固定字符的比较来进行 hook 检测;又如,在上述  
10 的 Android 版本在 5.0 及其以上的系统,检测设备可按照预设的逻辑算法对该 flag 值进行逻辑运算后进行 hook 检测。或者,可选的,检测设备可以通过上述两种 hook 检测方式进行函数 hook 检测,满足任一种方式对应的条件时,即可确定该函数被 hook。

15 205、当确定该目标函数被 hook 时,从该目标函数的内存中获取该目标函数对应的目标函数指针。

206、根据预先存储的各函数指针和函数的对应关系,确定出该目标函数指针对应的原始函数,并利用该原始函数替换该目标函数。

具体的,该步骤 205-206 的描述请参照上述图 1 所示实施例中的相关描述,此处不赘述。

20 207、确定目标终端的目标风险等级,根据预设的不同风险等级与控制策略的对应关系,确定与该目标风险等级对应的控制策略,并按照确定出的该控制策略对该目标终端的操作进行控制。

25 在一些可选的实施例中,通过原始函数获得真实信息之后,可以结合风控规则识别得到函数被篡改的风险等级,进而对于不同风险等级,可以采取不同的控制策略。

30 可选的,在该利用该原始函数替换该目标函数之后,检测设备还可获取该原始函数对应的设备信息,并确定该设备信息的优先级,进而根据该设备信息的优先级确定该目标函数被 hook 的目标风险等级,从而能够根据预设的不同风险等级与控制策略的对应关系,确定与该目标风险等级对应的控制策略,并按照确定出的该控制策略对该目标终端的操作进行控制。也就是说,可预置得到各设备信息的优先级(重要等级),以及各优先级分别对应的风险等级,以及各风险等级分别对应的控制策略。其中,设备信息的优先级越高,风险等级越高。从而检测设备可根据被篡改的设备信息的优先级来确定对该目标终端的控制策略,灵活性较高。

可选的，检测设备还可统计预设时间范围内该目标终端被 hook 的函数的数目，并根据该数目确定该目标终端的目标风险等级，进而根据预设的不同风险等级与控制策略的对应关系，确定与该目标风险等级对应的控制策略，并按照确定出的该控制策略对该目标终端的操作进行控制。也就是说，该风险等级还可以根据该终端预设时间范围内被 hook 的函数的数目确定出，如被 hook 的函数的数目越多，风险等级越高。或者，可选的，该风险等级还可根据预设时间范围内发生函数被 hook 的次数确定出，如该预设时间范围内被 hook 的函数的次数越多，风险等级越高，等等。从而检测设备可根据被 hook 的函数的数目或函数被 hook 的发生次数来确定对该目标终端的控制策略，灵活性较高。

举例来说，该风险等级可以分为低危、中危、高危，或者可以分为一级、二级、三级等等。对于风险等级的划分和确定方式，本申请不做限定。控制策略可包括指示终端输出提示，要求用户输入验证信息（风险等级较低时，如低危场景）；指示终端禁止用户在特定的风控场景（例如登录，APP 优惠领取如领取红包、兑换优惠券，或交易如消费、转账等等）下的请求访问操作（风险等级较高时，如中危场景）；指示终端禁止用户一切请求访问操作（风险等级很高时，如高危场景），等等，此处不一一列举。

非法分子通过篡改设备信息，影响的主要是 App 的服务提供方。例如，许多“薅羊毛”的攻击者通过 hook 框架修改手机信息，从而只需少量手机就可以模拟出许多伪造的手机，绕过设备领取次数的限制，大量领取并转卖服务方提供的活动福利，例如满减券、话费券等。这样，原本用于宣传推广的福利，就被少数攻击者领取获利了，而没有起到应有的效果。通过本申请的函数 hook 检测方案，能够及时地发现终端中的 hook 行为，并可对真实设备信息进行还原，以减小篡改行为带来的影响。

例如，针对 hook 终端 GPS 信息，将终端定位在特定地点的场景。假设司机使用打车 APP 并伪造在热点地区附近（即 hook GPS 到该热点区域）抢单；或者实际上没有在行驶，但通过伪造行驶路径刷单冒领补贴；又或者伪造运动步数信息，冒领活动奖励等等。通过本申请的 hook 检测方式，能够及时检测到该 GPS 函数被 hook，终端 GPS 信息被篡改，进而还原真实 GPS 信息后，如果发现不符，可禁止其领取奖励，或取消其领取的奖励，甚至及时的禁止其类似操作行为。

又如，针对 hook 设备 ID 的场景，为防止恶意用户大量注册账号刷单领活动福利，有些 App 根据设备 ID 来识别唯一设备，并限制每台手机只能参加一次活动。通过 hook 修改设备 ID 信息，就可以在不更换手机的情况下，伪造

成其他手机，多次领取福利。通过本申请的 hook 检测方式，能够及时检测到该设备 ID 的函数被 hook，该设备 ID 被篡改，还原真实的设备 ID 之后，可及时的禁止其操作行为。比如用户使用某一账号登录了一 APP 重复领取活动福利，检测设备识别出设备 ID 被篡改并获取到真实的设备 ID 之后，可以将该真实的设备 ID 和该登录账号都返回服务端，并与已有数据（包括各设备 ID、登录账号等等）进行比较，从而识别出哪些账号实际是运行在同一台设备上（比如设备 ID 相同的设备为同一设备）的。接下来，就可以对这些账号领取的活动福利进行取消、提现请求进行人工审核、或者封禁这批账号，以实现

对设备操作的控制。

又如，针对 hook 其他环境信息的场景。目前许多 App 风控规则会考虑到当前 App 运行的环境。例如，正常用户不会通过终端上的 Android 模拟器来运行 APP，使用模拟器的通常是恶意用户，进而禁止在模拟器环境运行 APP。而系统会明确指出当前是真的终端还是模拟器，由此恶意用户可能会 hook 系统信息；或者，由于在正常终端上看到的 Wi-Fi 信息也会和在模拟器中看到的

不一样，由此，用户可能会 hook Wi-Fi 信息。通过以上篡改方式可以将 APP 运行环境篡改为模拟器，使得难以区分正常用户还是可能的恶意用户。通过本申请的 hook 检测方式，能够及时检测到该运行环境被篡改，并及时地还原真实的运行环境，进而可及时地发现并禁止其操作行为，比如禁止该 APP 运行，或者禁止该 APP 上的一切访问请求。

进一步可选的，检测设备还可以将该被 hook 的函数和该真实的原始函数发送到服务器，或者可以将该原始函数的返回值和 hook 篡改后的目标函数的返回值发送给服务器，以便于帮助业务风控人员检测设备风险。进一步的，检测设备或服务器还可关联存储该被 hook 的函数（的返回值）和该原始函数（的返回值）的信息，使得后续能够快速锁定被 hook 的函数以及根据该关联关系快速找回正确的原始函数，并可提升函数还原的效率。

在本申请实施例

中，检测设备可通过获取目标终端的当前设备信息，对预置的与当前设备信息对应的函数列表中的函数进行 hook 检测，而不必对该目标终端中的所有函数进行 hook 检测，这就提升了 hook 检测的效率，节省了设备开销。进而通过分别获取这些函数的 flag 值，以根据该 flag 值确定各函数是否被 hook，并在确定该函数被 hook 时，能够通过获取该函数对应的函数指针，快速确定出真实的原始函数并还原，这就有助于避免 hook 行为的误报，提升 hook 检测的准确性，并能够通过还原真实函数，来提升终端安全性。

请参见图 3，图 3 是本申请实施例提供的又一种函数 hook 检测方法的流程示意图。具体的，如图 3 所示，该函数 hook 检测方法可以包括以下步骤：

301、根据统计的历史数据中不同设备信息下被 hook 的函数，建立函数检测模型。

可选的，该设备信息可包括设备型号、品牌、系统版本、风控场景信息中的一项或多项。

5       具体的，检测设备可预先收集历史记录中被 hook 的函数，并根据各被 hook 的函数对应终端的设备信息进行分组，确定出不同设备信息下对应的函数分组。进而可以以不同的设备信息为输入，其对应的函数分组中的函数为输出，训练得到函数检测模型。从而能够通过大数据分析不同型号、不同品牌、不同系统、不同风控场景的终端容易被 hook 的函数，来建立函数检测模型，使得  
10       后续能够通过将设备信息输入该函数检测模型，获取到需要进行 hook 检测的目标函数。

302、获取该目标终端的当前设备信息，并将该当前设备信息输入该函数检测模型，以得到需要进行 hook 检测的目标函数。

在建立得到函数检测模型之后，在需要对某一终端进行 hook 检测时，即  
15       可通过获取该终端的当前设备信息，如该终端的型号、品牌、当前系统版本、当前风控场景信息等信息，并将该当前设备信息输入该函数检测模型，以得到该终端中的目标函数，这就提升了 hook 检测的效率，针对性较强，且节省了设备开销。

可选的，检测设备在不同检测场景下，比如目标终端运行了不同类型的  
20       APP 时，可以根据不同的检测场景获取不同的设备信息，并根据上述的函数检测模型获取到该设备信息对应的目标函数，以进一步提升 hook 检测的效率和可靠性。

例如，检测到目标终端运行某一需要进行风控的 APP 时，检测设备可获取该目标终端的当前风控场景信息，并可输入该当前风控场景信息到该函数检测模型，以获取得到该当前风控场景信息对应的目标函数以进行 hook 检测。  
25       其中，该需要进行风控的 APP 可预先标记得到，比如通过预置一个包括各需要风控的 APP 的应用列表，通过检测运行的 APP 是否为该应用列表中的 APP 来确定该 APP 是否为需要进行风控的 APP。进一步的，该当前风控场景信息可以是针对该 APP 下的风控场景的信息。

30       又如，检测到目标终端运行打车 APP 时，检测设备可获取该目标终端的型号和系统版本，并可将该型号和当前系统版本输入到该函数检测模型，以获取得到该型号和当前系统版本对应的目标函数以进行 hook 检测。

303、获取该目标函数的 flag 值。

304、根据该 flag 值确定该目标函数是否被 hook。

305、当确定该目标函数被 hook 时，从该目标函数的内存中获取该目标函数对应的目标函数指针。

306、根据预先存储的各函数指针和函数的对应关系，确定出该目标函数指针对应的原始函数，并利用该原始函数替换该目标函数。

5       307、确定目标终端的目标风险等级，根据预设的不同风险等级与控制策略的对应关系，确定与该目标风险等级对应的控制策略，并按照确定出的该控制策略对该目标终端的操作进行控制。

具体的，该步骤 304-307 的描述请参照上述图 2 所示实施例中步骤 204-207 的相关描述，此处不赘述。

10       在本申请实施例中，检测设备可通过统计历史数据中不同设备信息对应的被 hook 的函数建立函数检测模型，进而在进行 hook 检测时，能够通过获取目标终端的当前设备信息并输入该函数检测模型后得到对应的输出函数之后，针对该输出的目标函数进行 hook 检测，而不必对该目标终端中的所有函数进行 hook 检测，这就提升了 hook 检测的效率，节省了设备开销。进而通过分别获  
15       取这些函数的 flag 值，以根据该 flag 值确定各函数是否被 hook，并在确定该函数被 hook 时，能够通过获取该函数对应的函数指针，快速确定出真实的原始函数并还原，这就有助于避免 hook 行为的误报，提升 hook 检测的准确性，并能够通过还原真实函数，来提升终端安全性。

上述方法实施例都是对本申请的函数 hook 检测方法的举例说明，对各个  
20       实施例的描述都各有侧重，某个实施例中未详述的部分，可以参见其他实施例的相关描述。

请参见图 4，图 4 是本申请实施例提供的一种检测设备的结构示意图。本申请实施例的检测设备包括用于执行上述函数 hook 检测方法的单元。具体的，本实施例的检测设备 400 可包括：获取单元 401 和处理单元 402。其中，

25       获取单元 401，用于获取所述目标终端中的目标函数的 flag 值，所述 flag 值用于标记所述目标函数的状态；

处理单元 402，用于根据所述 flag 值确定所述目标函数是否被 hook；

所述获取单元 401，还用于当确定所述目标函数被 hook 时，从所述目标函数的内存中获取所述目标函数对应的目标函数指针；

30       所述处理单元 402，还用于根据预先存储的各函数指针和函数的对应关系，确定出所述目标函数指针对应的原始函数，并利用所述原始函数替换所述目标函数。

可选的，获取单元 401 可在检测到目标终端安装有 Xposed 插件时，再获取该目标终端中的目标函数的 flag 值，此处不赘述。

可选的，所述获取单元 401，还用于获取所述目标终端的当前设备信息，所述当前设备信息包括所述目标终端的型号、系统版本、风控场景信息中的一项或多项；

5 所述处理单元 402，还用于根据预置的不同设备信息和函数列表的对应关系，确定出所述当前设备信息对应的函数列表，将所述当前设备信息对应的函数列表中的函数作为所述目标函数。

其中，每一个函数列表可包括对应的设备信息下被 hook 的频率大于预设频率阈值的函数和/或被 hook 的次数大于预设数目阈值的函数。

可选的，所述检测设备还包括：模型建立单元 403；

10 所述模型建立单元 403，用于根据统计的历史数据中不同设备信息下被 hook 的函数，建立函数检测模型，所述设备信息包括设备型号、系统版本、风控场景信息中的一项或多项；

所述获取单元 401，还用于获取所述目标终端的当前设备信息，并将所述当前设备信息输入所述函数检测模型，以得到所述目标函数。

15 可选的，所述处理单元 402，具体用于将所述 flag 值中的预设位置处的字符与预设的固定字符进行比较，所述预设位置处的字符的字符数与所述固定字符的字符数相同；当比较得到所述预设位置处的字符与所述固定字符不同时，确定所述目标函数被 hook。

20 可选的，所述处理单元 402，具体用于按照预设的逻辑算法对所述 flag 值进行逻辑运算，以得到运算结果值，其中，所述逻辑算法是根据预设字符串和系统中的原生函数执行时的跳转地址确定的；当所述运算结果值为正整数时，确定所述目标函数被 hook。

25 可选的，所述处理单元 402，还用于根据所述原始函数获取所述原始函数对应的设备信息，并确定所述设备信息的优先级；根据所述设备信息的优先级确定所述目标函数被 hook 的目标风险等级；根据预设的不同风险等级与控制策略的对应关系，确定与所述目标风险等级对应的控制策略，并按照确定出的所述控制策略对所述目标终端的操作进行控制。

30 可选的，所述处理单元 402，还用于统计预设时间范围内所述目标终端被 hook 的函数的数目，并根据所述数目确定所述目标终端的目标风险等级；根据预设的不同风险等级与控制策略的对应关系，确定与所述目标风险等级对应的控制策略，并按照确定出的所述控制策略对所述目标终端的操作进行控制。

具体的，该检测设备可通过上述单元实现上述图 1 至图 3 所示实施例中的函数 hook 检测方法中的部分或全部步骤。应理解，本申请实施例是对应方法实施例的装置实施例，对方法实施例的描述，也适用于本申请实施例。

请参见图 5，图 5 是本申请实施例提供的另一种检测设备的结构示意图。该检测设备用于执行上述的方法。如图 5 所示，本实施例中的检测设备 500 可以包括：一个或多个处理器 501 和存储器 502。可选的，该检测设备还可包  
5 括一个或多个用户接口 503，和/或，一个或多个通信接口 504。上述处理器 501、用户接口 503、通信接口 504 和存储器 502 可通过总线 505 连接，或者可以通过其他方式连接，图 5 中以总线方式进行示例说明。其中，存储器 502 用于存储计算机程序，所述计算机程序包括程序指令，处理器 501 用于执行存储器 502 存储的程序指令。

10 其中，处理器 501 可用于调用所述程序指令执行以下步骤：当检测到目标终端安装有 Xposed 插件时，获取所述目标终端中的目标函数的 flag 值，所述 flag 值用于标记所述目标函数的状态；根据所述 flag 值确定所述目标函数是否被 hook；当确定所述目标函数被 hook 时，从所述目标函数的内存中获取所述  
15 目标函数对应的目标函数指针；根据预先存储的各函数指针和函数的对应关系，确定出所述目标函数指针对应的原始函数，并利用所述原始函数替换所述目标函数。

可选的，处理器 501 还可调用所述程序指令执行以下步骤：获取所述目标终端的当前设备信息，所述当前设备信息包括所述目标终端的型号、系统版本、  
20 风控场景信息中的一项或多项；根据预置的不同设备信息和函数列表的对应关系，确定出所述当前设备信息对应的函数列表，其中，每一个函数列表包括对应的设备信息下被 hook 的频率大于预设频率阈值的函数和/或被 hook 的次数大于预设数目阈值的函数；将所述当前设备信息对应的函数列表中的函数作为所述目标函数。

25 可选的，处理器 501 还可调用所述程序指令执行以下步骤：根据统计的历史数据中不同设备信息下被 hook 的函数，建立函数检测模型，所述设备信息包括设备型号、系统版本、风控场景信息中的一项或多项；获取所述目标终端的当前设备信息，并将所述当前设备信息输入所述函数检测模型，以得到所述目标函数。

30 可选的，处理器 501 在调用所述程序指令执行所述根据所述 flag 值确定所述目标函数是否被 hook 时，具体执行以下步骤：将所述 flag 值中的预设位置处的字符与预设的固定字符进行比较，所述预设位置处的字符的字符数与所述固定字符的字符数相同；当比较得到所述预设位置处的字符与所述固定字符不同时，确定所述目标函数被 hook。

可选的，处理器 501 在调用所述程序指令执行所述根据所述 flag 值确定所

述目标函数是否被 hook 时，具体执行以下步骤：按照预设的逻辑算法对所述 flag 值进行逻辑运算，以得到运算结果值，其中，所述逻辑算法是根据预设字符串和系统中的原生函数执行时的跳转地址确定的；当所述运算结果值为正整数时，确定所述目标函数被 hook。

- 5 可选的，处理器 501 还可调用所述程序指令执行所述利用所述原始函数替换所述目标函数之后，还可执行以下步骤：根据所述原始函数获取所述原始函数对应的设备信息，并确定所述设备信息的优先级；根据所述设备信息的优先级确定所述目标函数被 hook 的目标风险等级；根据预设的不同风险等级与控制策略的对应关系，确定与所述目标风险等级对应的控制策略，并按照确定出的所述控制策略对所述目标终端的操作进行控制。

- 10 可选的，处理器 501 还可调用所述程序指令执行以下步骤：统计预设时间范围内所述目标终端被 hook 的函数的数目，并根据所述数目确定所述目标终端的目标风险等级；根据预设的不同风险等级与控制策略的对应关系，确定与所述目标风险等级对应的控制策略，并按照确定出的所述控制策略对所述目标终端的操作进行控制。

- 15 其中，所述处理器 501 可以是中央处理单元 (Central Processing Unit, CPU)，该处理器还可以是其他通用处理器、数字信号处理器 (Digital Signal Processor, DSP)、专用集成电路 (Application Specific Integrated Circuit, ASIC)、现成可编程门阵列 (Field-Programmable Gate Array, FPGA) 或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件等。通用处理器可以是微处理器或者该处理器也可以是任何常规的处理器等。

用户接口 503 可包括输入设备和输出设备，输入设备可以包括触控板、麦克风等，输出设备可以包括显示器 (LCD 等)、扬声器等。

通信接口 504 可包括接收器和发射器，用于与其他设备进行通信。

- 25 存储器 502 可以包括只读存储器和随机存取存储器，并向处理器 501 提供指令和数据。存储器 502 的一部分还可以包括非易失性随机存取存储器。例如，存储器 502 还可以存储上述的函数指针和函数的对应关系等等。

- 30 具体实现中，本申请实施例中所描述的处理器 501 等可执行上述图 1 至图 3 所示的方法实施例中所描述的实现方式，也可执行本申请实施例图 4 所描述的各单元的实现方式，此处不赘述。

本申请实施例还提供了一种计算机可读存储介质，所述计算机可读存储介质存储有计算机程序，所述计算机程序被处理器执行时可实现图 1 至图 3 所对应实施例中描述的函数 hook 检测方法中的部分或全部步骤，也可实现本申请图 4 或图 5 所示实施例的检测设备的功能，此处不赘述。

本申请实施例还提供了一种包含指令的计算机程序产品，当其在计算机上运行时，使得计算机执行上述方法中的部分或全部步骤。

5 所述计算机可读存储介质可以是前述任一实施例所述的检测设备的内部存储单元，例如检测设备的硬盘或内存。所述计算机可读存储介质也可以是所述检测设备的外部存储设备，例如所述检测设备上配备的插接式硬盘，智能存储卡(Smart Media Card, SMC)，安全数字(Secure Digital, SD)卡，闪存卡(Flash Card)等。

10 在本申请中，术语“和/或”，仅仅是一种描述关联对象的关联关系，表示可以存在三种关系，例如，A 和/或 B，可以表示：单独存在 A，同时存在 A 和 B，单独存在 B 这三种情况。另外，本文中字符“/”，一般表示前后关联对象是一种“或”的关系。

在本申请的各种实施例中，上述各过程的序号的大小并不意味着执行顺序的先后，各过程的执行顺序应以其功能和内在逻辑确定，而不应对本申请实施例的实施过程构成任何限定。

15 以上所述，仅为本申请的部分实施方式，但本申请的保护范围并不局限于此，任何熟悉本技术领域的技术人员在本申请揭露的技术范围内，可轻易想到各种等效的修改或替换，这些修改或替换都应涵盖在本申请的保护范围之内。

## 权 利 要 求

1、一种函数 hook 检测方法，其特征在于，包括：

5 当检测到目标终端安装有 Xposed 插件时，获取所述目标终端中的目标函数的 flag 值，所述 flag 值用于标记所述目标函数的状态；

根据所述 flag 值确定所述目标函数是否被 hook；

当确定所述目标函数被 hook 时，从所述目标函数的内存中获取所述目标函数对应的目标函数指针；

10 根据预先存储的各函数指针和函数的对应关系，确定出所述目标函数指针对应的原始函数，并利用所述原始函数替换所述目标函数。

2、根据权利要求 1 所述的方法，其特征在于，所述方法还包括：

获取所述目标终端的当前设备信息，所述当前设备信息包括所述目标终端的型号、系统版本、风控场景信息中的一项或多项；

15 根据预置的设备信息和函数列表的对应关系，确定出所述当前设备信息对应的函数列表，其中，每一个函数列表包括对应的设备信息下被 hook 的频率大于预设频率阈值的函数和/或被 hook 的次数大于预设数目阈值的函数；

将所述当前设备信息对应的函数列表中的函数作为所述目标函数。

3、根据权利要求 1 所述的方法，其特征在于，所述方法还包括：

20 根据统计的历史数据中不同设备信息下被 hook 的函数，建立函数检测模型，所述设备信息包括设备型号、系统版本、风控场景信息中的一项或多项；

获取所述目标终端的当前设备信息，并将所述当前设备信息输入所述函数检测模型，以得到所述目标函数。

4、根据权利要求 1-3 任一项所述的方法，其特征在于，所述根据所述 flag 值确定所述目标函数是否被 hook，包括：

25 将所述 flag 值中的预设位置处的字符与预设的固定字符进行比较，所述预设位置处的字符的字符数与所述固定字符的字符数相同；

当比较得到所述预设位置处的字符与所述固定字符不同时，确定所述目标函数被 hook。

30 5、根据权利要求 1-3 任一项所述的方法，其特征在于，所述根据所述 flag 值确定所述目标函数是否被 hook，包括：

按照预设的逻辑算法对所述 flag 值进行逻辑运算，以得到运算结果值，其中，所述逻辑算法是根据预设字符串和系统中的原生函数执行时的跳转地址确定的；

当所述运算结果值为正整数时，确定所述目标函数被 hook。

6、根据权利要求1所述的方法，其特征在于，在所述利用所述原始函数替换所述目标函数之后，所述方法还包括：

根据所述原始函数获取所述原始函数对应的设备信息，并确定所述设备信息的优先级；

5 根据所述设备信息的优先级确定所述目标函数被 hook 的目标风险等级；

根据预设的不同风险等级与控制策略的对应关系，确定与所述目标风险等级对应的控制策略，并按照确定出的所述控制策略对所述目标终端的操作进行控制。

7、根据权利要求1所述的方法，其特征在于，所述方法还包括：

10 统计预设时间范围内所述目标终端被 hook 的函数的数目，并根据所述数目确定所述目标终端的目标风险等级；

根据预设的不同风险等级与控制策略的对应关系，确定与所述目标风险等级对应的控制策略，并按照确定出的所述控制策略对所述目标终端的操作进行控制。

15 8、一种检测设备，其特征在于，包括：获取单元和处理单元；

所述获取单元，用于获取目标终端中的目标函数的 flag 值，所述 flag 值用于标记所述目标函数的状态；

所述处理单元，用于根据所述 flag 值确定所述目标函数是否被 hook；

20 所述获取单元，还用于当确定所述目标函数被 hook 时，从所述目标函数的内存中获取所述目标函数对应的目标函数指针；

所述处理单元，还用于根据预先存储的各函数指针和函数的对应关系，确定出所述目标函数指针对应的原始函数，并利用所述原始函数替换所述目标函数。

9、根据权利要求8所述的检测设备，其特征在于，

25 所述获取单元，还用于获取所述目标终端的当前设备信息，所述当前设备信息包括所述目标终端的型号、系统版本、风控场景信息中的一项或多项；

所述处理单元，还用于根据预置的设备信息和函数列表的对应关系，确定出所述当前设备信息对应的函数列表，将所述当前设备信息对应的函数列表中的函数作为所述目标函数；

30 其中，每一个函数列表包括对应的设备信息下被 hook 的频率大于预设频率阈值的函数和/或被 hook 的次数大于预设数目阈值的函数。

10、根据权利要求8所述的检测设备，其特征在于，所述检测设备还包括：模型建立单元；

所述模型建立单元，用于根据统计的历史数据中不同设备信息下被 hook

的函数，建立函数检测模型，所述设备信息包括设备型号、系统版本、风控场景信息中的一项或多项；

所述获取单元，还用于获取所述目标终端的当前设备信息，并将所述当前设备信息输入所述函数检测模型，以得到所述目标函数。

5 11、根据权利要求 8-10 任一项所述的检测设备，其特征在于，

所述处理单元，具体用于将所述 flag 值中的预设位置处的字符与预设的固定字符进行比较，所述预设位置处的字符的字符数与所述固定字符的字符数相同；当比较得到所述预设位置处的字符与所述固定字符不同时，确定所述目标函数被 hook。

10 12、根据权利要求 8-10 任一项所述的检测设备，其特征在于，

所述处理单元，具体用于按照预设的逻辑算法对所述 flag 值进行逻辑运算，以得到运算结果值，其中，所述逻辑算法是根据预设字符串和系统中的原生函数执行时的跳转地址确定的；当所述运算结果值为正整数时，确定所述目标函数被 hook。

15 13、根据权利要求 8 所述的检测设备，其特征在于，

所述处理单元，还用于根据所述原始函数获取所述原始函数对应的设备信息，并确定所述设备信息的优先级；根据所述设备信息的优先级确定所述目标函数被 hook 的目标风险等级；根据预设的不同风险等级与控制策略的对应关系，确定与所述目标风险等级对应的控制策略，并按照确定出的所述控制策略

20 对所述目标终端的操作进行控制。

14、根据权利要求 8 所述的检测设备，其特征在于，

所述处理单元，还用于统计预设时间范围内所述目标终端被 hook 的函数的数目，并根据所述数目确定所述目标终端的目标风险等级；根据预设的不同风险等级与控制策略的对应关系，确定与所述目标风险等级对应的控制策略，

25 并按照确定出的所述控制策略对所述目标终端的操作进行控制。

15、一种检测设备，其特征在于，包括处理器和存储器，所述处理器和存储器相互连接，其中，所述存储器用于存储计算机程序，所述计算机程序包括程序指令，所述处理器被配置用于调用所述程序指令，执行以下步骤：

30 当检测到目标终端安装有 Xposed 插件时，获取所述目标终端中的目标函数的 flag 值，所述 flag 值用于标记所述目标函数的状态；根据所述 flag 值确定所述目标函数是否被 hook；当确定所述目标函数被 hook 时，从所述目标函数的内存中获取所述目标函数对应的目标函数指针；根据预先存储的各函数指针和函数的对应关系，确定出所述目标函数指针对应的原始函数，并利用所述原始函数替换所述目标函数。

16、根据权利要求 15 所述的检测设备，其特征在于，所述处理器还调用所述程序指令执行以下步骤：

5 获取所述目标终端的当前设备信息，所述当前设备信息包括所述目标终端的型号、系统版本、风控场景信息中的一项或多项；根据预置的设备信息和函数列表的对应关系，确定出所述当前设备信息对应的函数列表，其中，每一个函数列表包括对应的设备信息下被 hook 的频率大于预设频率阈值的函数和/或被 hook 的次数大于预设数目阈值的函数；将所述当前设备信息对应的函数列表中的函数作为所述目标函数。

10 17、根据权利要求 15 所述的检测设备，其特征在于，所述处理器还调用所述程序指令执行以下步骤：

根据统计的历史数据中不同设备信息下被 hook 的函数，建立函数检测模型，所述设备信息包括设备型号、系统版本、风控场景信息中的一项或多项；获取所述目标终端的当前设备信息，并将所述当前设备信息输入所述函数检测模型，以得到所述目标函数。

15 18、根据权利要求 15-17 任一项所述的检测设备，其特征在于，所述处理器在调用所述程序指令执行所述根据所述 flag 值确定所述目标函数是否被 hook 时，具体执行以下步骤：

20 将所述 flag 值中的预设位置处的字符与预设的固定字符进行比较，所述预设位置处的字符的字符数与所述固定字符的字符数相同；当比较得到所述预设位置处的字符与所述固定字符不同时，确定所述目标函数被 hook。

19、根据权利要求 15-17 任一项所述的检测设备，其特征在于，所述处理器在调用所述程序指令执行所述根据所述 flag 值确定所述目标函数是否被 hook 时，具体执行以下步骤：

25 按照预设的逻辑算法对所述 flag 值进行逻辑运算，以得到运算结果值，其中，所述逻辑算法是根据预设字符串和系统中的原生函数执行时的跳转地址确定的；当所述运算结果值为正整数时，确定所述目标函数被 hook。

20、一种计算机可读存储介质，其特征在于，所述计算机可读存储介质存储有计算机程序，所述计算机程序包括程序指令，所述程序指令当被处理器执行时使所述处理器执行如权利要求 1-7 任一项所述的方法。

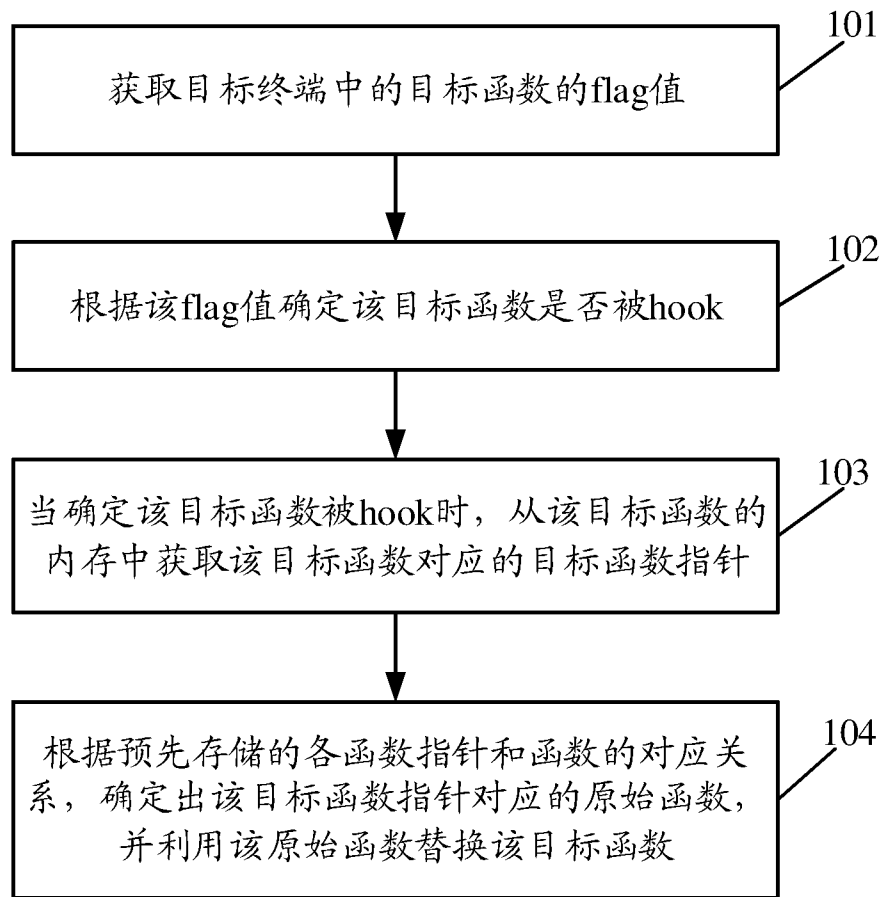


图 1

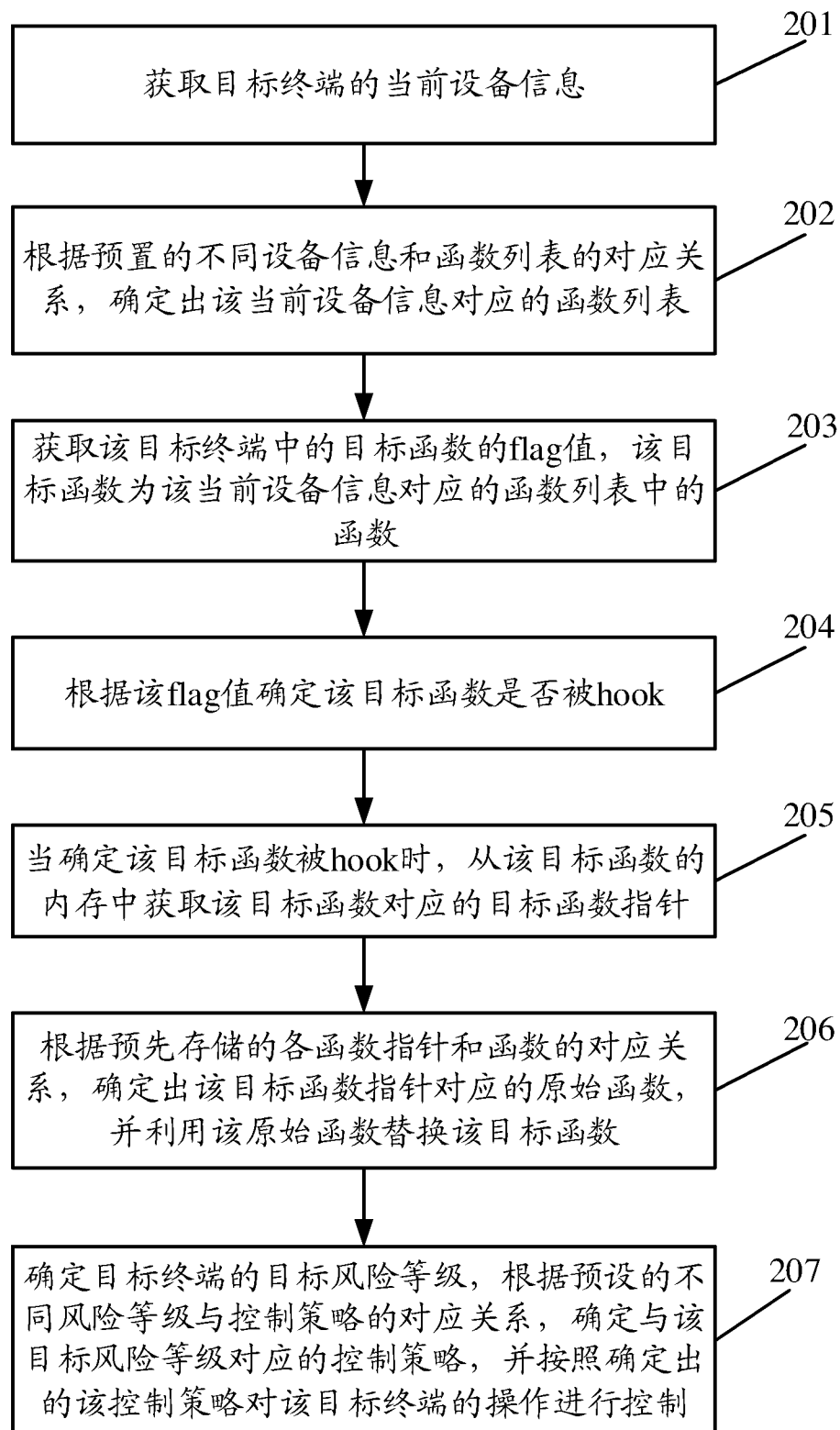


图 2

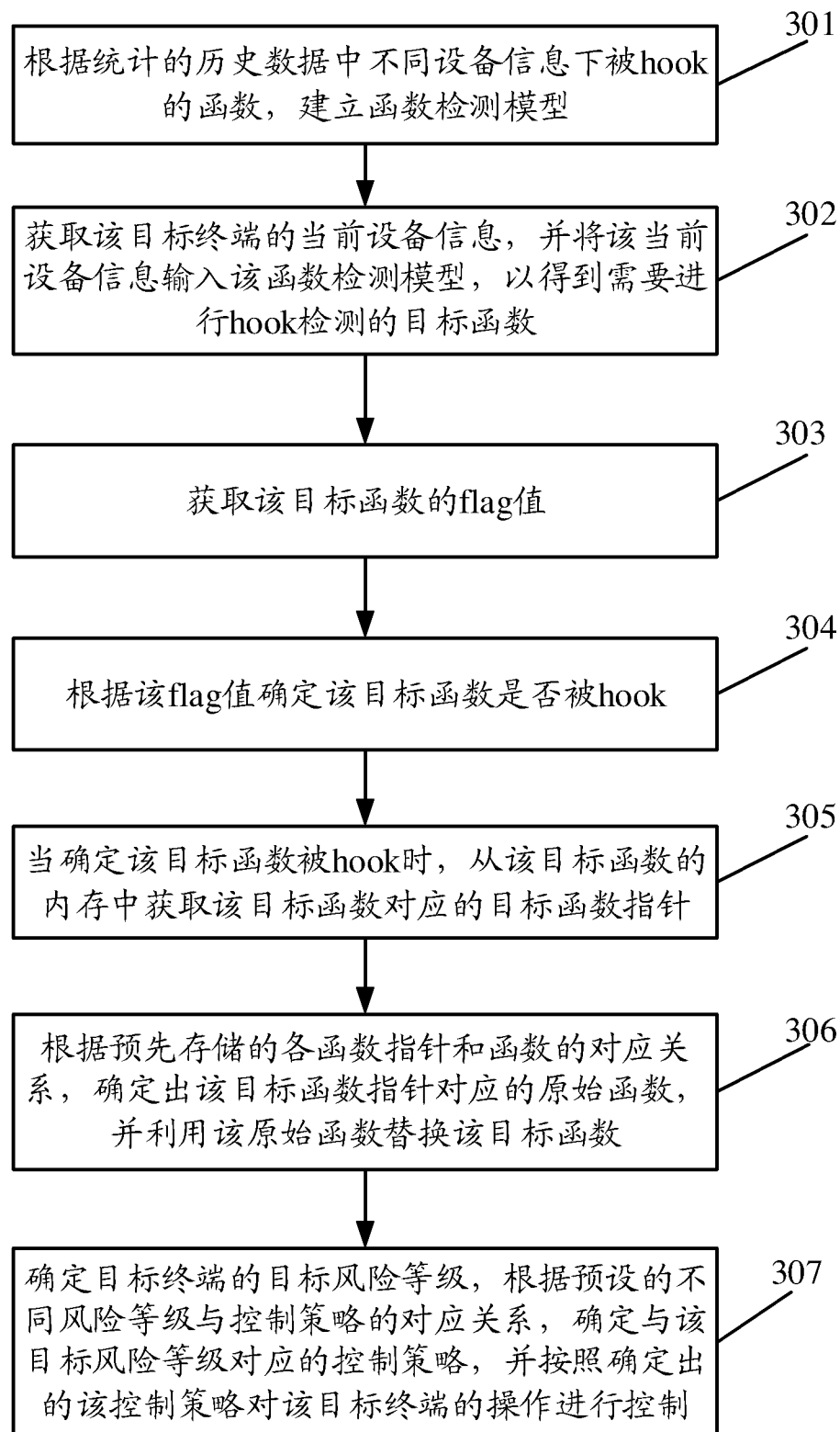


图 3

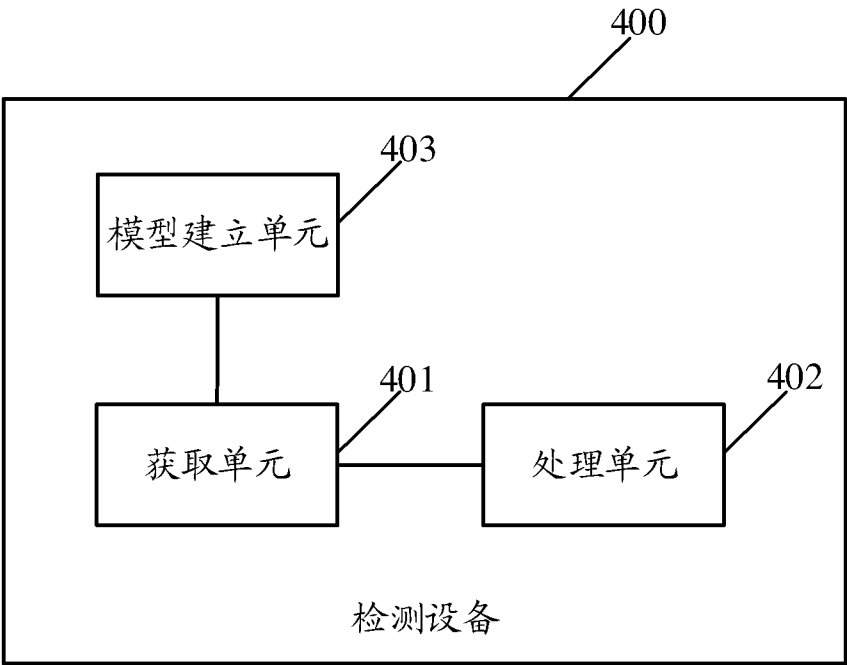


图 4

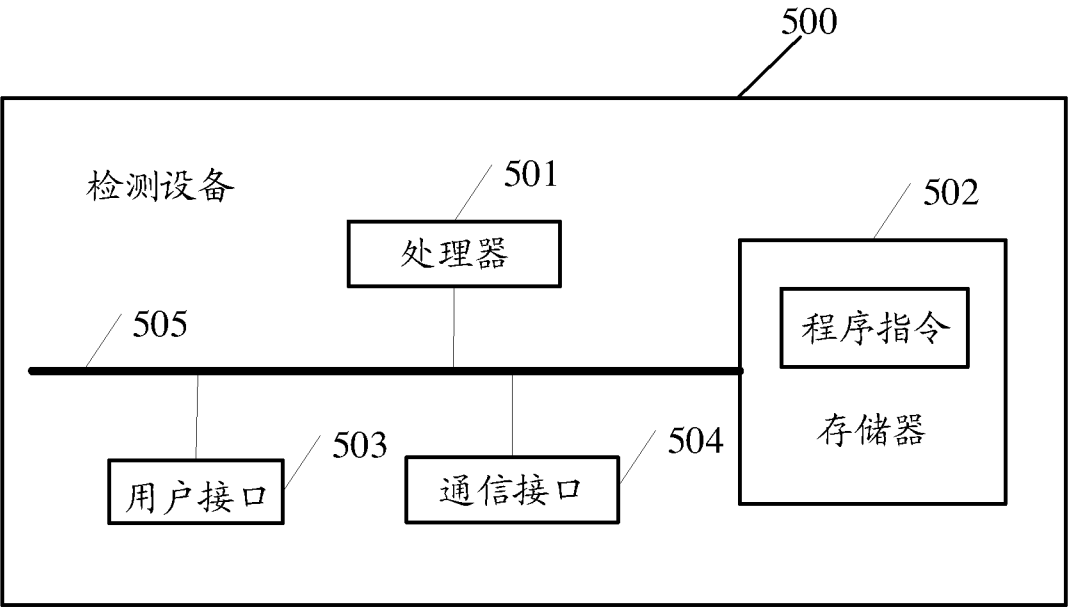


图 5

## INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2018/107745

**A. CLASSIFICATION OF SUBJECT MATTER**

G06F 21/52(2013.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

**B. FIELDS SEARCHED**

Minimum documentation searched (classification system followed by classification symbols)

G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

WPI, EPODOC, CNPAT, CNKI, IEEE: 挂钩, 钩子, 函数, 修改, 篡改, 替换, 原始函数, 解除, 恢复, 指针, hook, function, Xposed, tamper, protection, point, original, table, list, replace

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

| Category* | Citation of document, with indication, where appropriate, of the relevant passages                                                                                                                                                                       | Relevant to claim No. |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|
| Y         | CN 107808096 A (XIAMEN ANSHENG NETWORK TECHNOLOGY CO., LTD.) 16 March 2018 (2018-03-16)<br>description, paragraphs [0033]-[0044] and [0048]                                                                                                              | 1-20                  |
| Y         | "Xposed Hook行为的检测 (Non-official translation: Detectio of Xposed Hook Behaviour)"<br><a href="https://goodoak.github.io/2017/01/18/xposed-detect/">https://goodoak.github.io/2017/01/18/xposed-detect/</a> , 18 January 2017 (2017-01-18),<br>pp. 1 and 2 | 1-20                  |
| Y         | CN 106325927 A (BEIJING KINGSOFT SECURITY MANAGEMENT SYSTEM TECHNOLOGY CO., LTD.) 11 January 2017 (2017-01-11)<br>description, paragraphs [0048]-[0050]                                                                                                  | 1-20                  |
| A         | CN 106096391 A (BEIJING KINGSOFT CORPORATION LIMITED) 09 November 2016 (2016-11-09)<br>entire document                                                                                                                                                   | 1-20                  |
| A         | CN 101620658 A (PEKING UNIVERSITY) 06 January 2010 (2010-01-06)<br>entire document                                                                                                                                                                       | 1-20                  |
| A         | US 2005203904 A1 (INTERNATIONAL BUSINESS MACHINES CORPORATION) 15 September 2005 (2005-09-15)<br>entire document                                                                                                                                         | 1-20                  |



Further documents are listed in the continuation of Box C.



See patent family annex.

\* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&amp;" document member of the same patent family

Date of the actual completion of the international search

09 April 2019

Date of mailing of the international search report

29 April 2019

Name and mailing address of the ISA/CN

State Intellectual Property Office of the P. R. China (ISA/  
CN)  
No. 6, Xitucheng Road, Jimenqiao Haidian District, Beijing  
100088  
China

Facsimile No. (86-10)62019451

Authorized officer

Telephone No.

**INTERNATIONAL SEARCH REPORT**  
**Information on patent family members**

International application No.

**PCT/CN2018/107745**

| Patent document<br>cited in search report |            |    | Publication date<br>(day/month/year) | Patent family member(s) |            |    | Publication date<br>(day/month/year) |
|-------------------------------------------|------------|----|--------------------------------------|-------------------------|------------|----|--------------------------------------|
| CN                                        | 107808096  | A  | 16 March 2018                        | None                    |            |    |                                      |
| CN                                        | 106325927  | A  | 11 January 2017                      | None                    |            |    |                                      |
| CN                                        | 106096391  | A  | 09 November 2016                     | None                    |            |    |                                      |
| CN                                        | 101620658  | A  | 06 January 2010                      | None                    |            |    |                                      |
| US                                        | 2005203904 | A1 | 15 September 2005                    | US                      | 2008059673 | A1 | 06 March 2008                        |

## 国际检索报告

国际申请号

PCT/CN2018/107745

## A. 主题的分类

G06F 21/52 (2013.01) i

按照国际专利分类 (IPC) 或者同时按照国家分类和 IPC 两种分类

## B. 检索领域

检索的最低限度文献 (标明分类系统和分类号)

G06F

包含在检索领域中的除最低限度文献以外的检索文献

在国际检索时查阅的电子数据库 (数据库的名称, 和使用的检索词 (如使用))

WPI, EPDOC, CNPAT, CNKI, IEEE: 挂钩, 钩子, 函数, 修改, 篡改, 替换, 原始函数, 解除, 恢复, 指针, hook, function, Xposed, tamper, protection, point, original, table, list, replace

## C. 相关文件

| 类 型* | 引用文件, 必要时, 指明相关段落                                                                                                                                                                        | 相关的权利要求 |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| Y    | CN 107808096 A (厦门安胜网络科技有限公司) 2018年 3月 16日 (2018 - 03 - 16)<br>说明书第[0033]-[0044]段, 第[0048]段                                                                                              | 1-20    |
| Y    | "Xposed Hook行为的检测"<br><a href="https://goodoak.github.io/2017/01/18/xposed-detect/">https://goodoak.github.io/2017/01/18/xposed-detect/</a> , 2017年 1月<br>18日 (2017 - 01 - 18),<br>第1-2页 | 1-20    |
| Y    | CN 106325927 A (北京金山安全管理系统技术有限公司) 2017年 1月 11日 (2017 -<br>01 - 11)<br>说明书第[0048]-[0050]段                                                                                                 | 1-20    |
| A    | CN 106096391 A (北京金山安全软件有限公司) 2016年 11月 9日 (2016 - 11 - 09)<br>全文                                                                                                                        | 1-20    |
| A    | CN 101620658 A (北京大学) 2010年 1月 6日 (2010 - 01 - 06)<br>全文                                                                                                                                 | 1-20    |
| A    | US 2005203904 A1 (INTERNATIONAL BUSINESS MACHINES CORPORATION) 2005年 9月 15<br>日 (2005 - 09 - 15)<br>全文                                                                                   | 1-20    |

☐ 其余文件在C栏的续页中列出。☒ 见同族专利附件。

\* 引用文件的具体类型:

“A” 认为不特别相关的表示了现有技术一般状态的文件

“E” 在国际申请日的当天或之后公布的在先申请或专利

“L” 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件 (如具体说明的)

“O” 涉及口头公开、使用、展览或其他方式公开的文件

“P” 公布日先于国际申请日但迟于所要求的优先权日的文件

“T” 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件

“X” 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性

“Y” 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性

“&amp;” 同族专利的文件

国际检索实际完成的日期

2019年 4月 9日

国际检索报告邮寄日期

2019年 4月 29日

ISA/CN的名称和邮寄地址

中国国家知识产权局 (ISA/CN)  
中国北京市海淀区蓟门桥西土城路6号 100088

受权官员

曾璇

传真号 (86-10) 62019451

电话号码 86-(10)-53961373

国际检索报告  
关于同族专利的信息

国际申请号

PCT/CN2018/107745

| 检索报告引用的专利文件 |            |    | 公布日<br>(年/月/日) | 同族专利 |            |    | 公布日<br>(年/月/日) |
|-------------|------------|----|----------------|------|------------|----|----------------|
| CN          | 107808096  | A  | 2018年 3月 16日   | 无    |            |    |                |
| CN          | 106325927  | A  | 2017年 1月 11日   | 无    |            |    |                |
| CN          | 106096391  | A  | 2016年 11月 9日   | 无    |            |    |                |
| CN          | 101620658  | A  | 2010年 1月 6日    | 无    |            |    |                |
| US          | 2005203904 | A1 | 2005年 9月 15日   | US   | 2008059673 | A1 | 2008年 3月 6日    |