(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
2023 年 3 月 16 日 (16.03.2023)

WIPO | PCT

(10) 国际公布号
WO 2023/035751 A1

(51) 国际专利分类号:
*G06F 21/56* (2013.01)

(21) 国际申请号: PCT/CN2022/104296

(22) 国际申请日: 2022 年 7 月 7 日 (07.07.2022)

(25) 申请语言: 中文

(26) 公布语言: 中文

(30) 优先权:
202111067984.1 2021年9月13日 (13.09.2021) CN

(71) 申请人: 支付宝（杭州）信息技术有限公司 (**ALIPAY (HANGZHOU) INFORMATION TECHNOLOGY CO., LTD.**) [CN/CN]; 中国浙江省杭州市西湖区西溪路 556 号 8 层 B 段 801-11, Zhejiang 310000 (CN)。

(72) 发明人: **曹世杰(CAO, Shijie)**; 中国浙江省杭州市西湖区西溪路556号8层B段801-11, Zhejiang 310000 (CN)。 **张建涛(ZHANG, Jiantao)**; 中国浙江省杭州市西湖区西溪路556号8层B段801-11, Zhejiang 310000 (CN)。 **张宁(ZHANG, Ning)**; 中国浙江省杭州市西湖区西溪路556号8层B段801-11, Zhejiang 310000 (CN)。 **姚尧(YAO, Yao)**; 中国浙江省杭州市西湖区西溪路556号8层B段801-11, Zhejiang 310000 (CN)。

(74) 代理人: 北京博思佳知识产权代理有限公司 (**BEIJING BESTIPR INTELLECTUAL PROPERTY LAW CORPORATION**); 中国北京市海淀区上地三街 9 号嘉华大厦 B 座 409, Beijing 100085 (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, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ,

(54) **Title:** INTELLIGENT CONFUSION FOR MOBILE TERMINAL APPLICATION

(54) 发明名称: 移动端应用的智能混淆

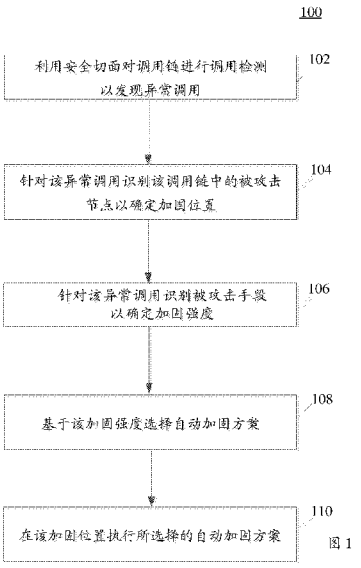


图 1

102 Perform call detection on a call chain by using a security aspect, so as to discover an anomalous call
104 For the anomalous call, identify an attacked node in the call chain to determine a reinforcement position
106 For the anomalous call, identify an attacked means to determine reinforcement strength
108 Select an automatic reinforcement scheme on the basis of the reinforcement strength
110 Execute the selected automatic reinforcement scheme at the reinforcement position

(57) **Abstract:** The present disclosure provides an intelligent confusion method for a mobile terminal application. The method comprises: performing call detection on a call chain by using a security aspect, so as to discover an anomalous call; for the anomalous call, identifying an attacked node in the call chain to determine a reinforcement position; for the anomalous call, identifying an attacked means to determine reinforcement strength; selecting an automatic reinforcement scheme on the basis of the reinforcement strength; and executing the selected automatic reinforcement scheme at the reinforcement position to control the influence of automatic reinforcement on the performance to be within an acceptable range.

WO 2023/035751 A1

[见续页]

IR, IS, IT, JM, 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, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, 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**：本公开提供了一种移动端应用的智能混淆方法。其中，所述方法包括：利用安全切面对调用链进行调用检测，以发现异常调用；针对该异常调用识别该调用链中的被攻击节点以确定加固位置；针对该异常调用识别被攻击手段以确定加固强度；基于该加固强度选择自动加固方案；以及在该加固位置执行所选择的自动加固方案，以将该自动加固对性能的影响控制在可接受范围内。

# 移动端应用的智能混淆

## 技术领域

[01]本公开主要涉及软件混淆，尤其涉及移动端应用的智能混淆。

## 背景技术

[02]目前业界对与 App 的混淆加固，大都是对程序编译阶段生成的中间文件进行混淆。举例而言，ollvm 混淆加固技术主要是依赖人工对于特定的代码逻辑，应用 ollvm 在代码、符号层进行加固。又如，基于分析代码的性能与复杂度，计算混淆强度和混淆弹性，从而来自动输出混淆。

[03]面对日益增多的网络安全问题，需要保护移动端应用系统，以防外部通过逆向手法对逻辑进行跟踪分析、并进一步对移动端应用进行攻击。

[04]因此，本领域需要能够一种能够快速高效地对移动端应用中的敏感逻辑进行自动化保护加固的方案。

## 发明内容

[05]为解决上述技术问题，本公开提供了一种移动端应用的智能混淆方案。该方案能够利用安全切面体系对移动端应用的内视能力，对资源调用进行检测，针对异常调用识别出被攻击节点和被攻击手段，从而针对性地自动选择加固方案并进行加固。

[06]在本公开一实施例中，提供了一种移动端应用的智能混淆方法，包括：利用安全切面对调用链进行调用检测，以发现异常调用；针对该异常调用识别该调用链中的被攻击节点以确定加固位置；针对该异常调用识别被攻击手段以确定加固强度；基于该加固强度选择自动加固方案；以及在该加固位置执行所选择的自动加固方案，以将该自动加固对性能的影响控制在可接受范围内。

[07]在本公开另一实施例中，该异常调用基于动态代理的 Hook 机制。

[08]在本公开又一实施例中，该异常调用与静态分析或动态分析相对应。

[09]在本公开另一实施例中，该安全切面是和业务相交织且平行的安全层。

[10]在本公开又一实施例中，该调用链中的被攻击节点包括反射/动态代理 hook 点、JNI 接口调用 hook 点、类加载路径 hook 点、JNI 函数 hook 点、动态链接库 hook 点、inline hook 点中的一个或多个。

[11]在本公开另一实施例中，该被攻击手段包括对应于静态分析的被攻击手段、或对应于动态分析的被攻击手段、或对应于静态分析和动态分析两者的被攻击手段。

[12]在本公开又一实施例中，该加固强度可在弱加固强度、中加固强度和强加固强度之中选择。

[13]在本公开另一实施例中，基于该弱加固强度选择的自动加固方案采用静态混淆。

[14]在本公开又一实施例中，基于该中加固强度选择的自动加固方案采用动态混淆。

[15]在本公开再一实施例中，基于该强加固强度选择的自动加固方案采用动态混淆或者静态混淆和动态混淆的组合。

[16]在本公开一实施例中，提供了一种移动端应用的智能混淆系统，包括：检测模块，利用安全切面对调用链进行调用检测，以发现异常调用；识别模块，针对该异常调用识别该调用链中的被攻击节点以确定加固位置，并针对该异常调用识别被攻击手段以确定加固强度；加固选择模块，基于该加固强度选择自动加固方案；以及加固执行模块，在该加固位置执行所选择的自动加固方案，以将该自动加固对性能的影响控制在可接受范围内。

[17]在本公开另一实施例中，该异常调用与静态分析或动态分析相对应。

[18]在本公开又一实施例中，该被攻击手段包括对应于静态分析的被攻击手段、或对应于动态分析的被攻击手段、或对应于静态分析和动态分析两者的被攻击手段。

[19]在本公开另一实施例中，该加固强度可在弱加固强度、中加固强度和强加固强度之中选择。

[20]在本公开又一实施例中，该加固选择模块基于弱加固强度选择的自动加固方案采用静态混淆。

[21]在本公开另一实施例中，该加固选择模块基于中加固强度选择的自动加固方案采用动态混淆。

[22]在本公开又一实施例中，该加固选择模块基于强加固强度选择的自动加固方案采用动态混淆或者静态混淆和动态混淆的组合。

[23]在本公开一实施例中，提供了一种存储有指令的计算机可读存储介质，当这些指令被执行时使得机器执行如前所述的方法。

[24]提供本概述以便以简化的形式介绍以下在详细描述中进一步描述的一些概念。本概

述并不旨在标识所要求保护主题的关键特征或必要特征，也不旨在用于限制所要求保护主题的范围。

## 附图说明

[25]本公开的以上发明内容以及下面的具体实施方式在结合附图阅读时会得到更好的理解。需要说明的是，附图仅作为所请求保护的发明的示例。在附图中，相同的附图标记代表相同或类似的元素。

[26]图 1 是示出根据本公开一实施例的移动端应用的智能混淆方法的流程图；

[27]图 2 是示出根据本公开一实施例的安全切面系统的示意图；

[28]图 3 是示出根据本公开一实施例的基于安全切面进行智能混淆的过程的示意图；

[29]图 4 是示出根据本公开一实施例的基于调用链异常判断不同被攻击节点的过程的示意图；

[30]图 5 是示出根据本公开一实施例的移动端应用的智能混淆系统的框图。

## 具体实施方式

[31]为使得本公开的上述目的、特征和优点能更加明显易懂，以下结合附图对本公开的具体实施方式作详细说明。

[32]在下面的描述中阐述了很多具体细节以便于充分理解本公开，但是本公开还可以采用其它不同于在此描述的其它方式来实施，因此本公开不受下文公开的具体实施例的限制。

[33]在当今数字时代，移动应用的数量呈爆炸性增长，涵盖金融、电子商务、社区、医疗、房地产、工业等各行各业。在给人类带来便利的同时，网络攻击也从传统的 PC 网站转移到移动互联网。

[34]网络攻击针对移动端应用可以是反译 APP，修改重要代码和服务器的连接方式，重新包装和签字，生成与原创相同或相似的应用，由此谋取利益。还可以是在破解应用程序后添加恶意代码，比如获取相册数据、获取地址簿和短信数据，以及监控银行账户等敏感信息等等。

[35]为保护移动端应用，本领域采用的 App 混淆技术基本上通过程序编译阶段生成的中间文件进行混淆，主要是基于程序的复杂性、性能，生成一定混淆强度的方案。但是在

移动端应用程序中，除了考虑性能的影响以外，还需要关注哪些逻辑是要保护的，此时需要在外部通过逆向手法对逻辑进行跟踪分析。由此，本领域通用的对中间文件进行混淆并不适合对逻辑进行针对性保护。

[36]本公开所揭示的移动端应用的智能混淆方案纳入了安全切面体系，通过安全切面发现移动应用程序真实被攻击的风险点，从而针对性进行加固。本公开的移动端应用的智能混淆方案通过对调用链的还原，实现安全内视和追溯，具有强大的感知能力；并且通过对调用链节点信息的分析，来实现高效针对性加固。

[37]本公开将主要以安卓 Android 移动端应用为例进行移动端应用的智能混淆方案的具体描述，但本领域技术人员可以理解，本公开的技术方案同样适用于 iOS 移动端应用，在下文中将不再赘述。

[38]图 1 是示出根据本公开一实施例的移动端应用的智能混淆方法 100 的流程图。

[39]在 102，利用安全切面对调用链进行调用检测，以发现异常调用。

[40]安全切面体系建立了和业务相交织且平行的安全层，使得安全能够深入业务逻辑，实现细致的观测；同时又保证业务和安全的解耦合，使得业务和安全各自独立，各自高效运行。安全切面体系采用的是面向切面编程（AOP，Aspect-oriented Programming），具体是通过预编译、运行时动态代理或者注入的方式，实现在不修改源代码的情况下向程序动态添加功能。AOP 实现调用者和被调用者之间的解耦，能够提供跨模块的功能汇聚。AOP 适合解决程序中涉及横切面（cross-cut）的系统功能。

[41]安全切面体系由于具备了对 App 运行时对核心资源访问的内视能力，通过对调用链的检测，可以发现有风险的调用，并推导加固方案的生成。

[42]在 104，针对该异常调用识别该调用链中的被攻击节点以确定加固位置。

[43]在 Android 移动端应用中，AOP 编程的具体实现用到了动态代理的 Hook 机制。hook 机制可以实现针对调用链的不同节点和不同粒度的精确切入。根据本公开一实施例的基于调用链异常判断不同被攻击节点的过程将在以下结合图 4 具体描述。在本公开一实施例中，调用链中的被攻击节点包括反射/动态代理 hook 点、JNI 接口调用 hook 点、类加载路径 hook 点、JNI 函数 hook 点、动态链接库 hook 点、inline hook 点中的一个或多个

[44]当针对异常调用识别出调用链的被攻击节点时，即可确定加固位置。

[45]本领域技术人员可以理解，被攻击节点可涉及一个或多个，可针对性地逐个在各个

加固位置进行加固。

[46]在 106，针对该异常调用识别被攻击手段以确定加固强度。

[47]由于 hook 机制可以实现针对调用链的不同节点和不同粒度的精确切入，因此其对应的调用链的被攻击手段通常对应于静态分析（例如反射调用）和动态分析（例如代码注入、栈跟踪、木马劫持等）。

[48]静态分析通常并不运行代码，而对代码的静态特征和功能模块进行分析。静态分析能描绘程序的轮廓，包括控制流和数据结构。与静态分析攻击相对应的保护方法是静态混淆。静态混淆主要可以分为混淆控制流和混淆数据结构。举例而言，静态控制流混淆方法包括花指令、控制流平面化等，而数据结构混淆方法包括合并和拆分类和结构体，隐藏虚表等。

[49]动态分析则通过在可控环境中运行代码，全程监控代码的所有操作，观察其状态和执行流程的变化，从而获得执行过程中的各种数据。其包括调试、剖分、跟踪、模拟器等。与动态分析攻击相对应的保护方法是动态混淆。动态混淆主要包括自修改代码和虚拟机保护等混淆手段。自修改代码是程序运行期间修改或产生代码的机制，其将指令和数据存储在同一个内存空间中，因此指令可以被视作数据被其他指令读取和修改。程序在运行时向代码段中写数据，并且写入的数据被作为指令执行，达到自我修改的效果。自修改保护机制可以有效抵御静态逆向分析而且由于代码仅在需要时才以明文的形式出现，可以在一定程度上阻碍逆向工具获取程序所有的明文代码，从而抵抗动态分析。虚拟机保护是在虚拟机增加了一层自定义的指令集，而且其难以逆向，破解者想要获得程序源码，首先必须弄懂虚拟机的指令集，这极大增加了逆向的开销。虚拟机保护技术通过增加复杂度和在时间和空间上开销，获取了更高的保护强度。

[50]在识别出调用链所受攻击的攻击手段是对应于静态分析、动态分析、还是静态分析和动态分析两者的组合之后，确定对应的加固强度。

[51]在本公开一实施例中，如果被攻击手段仅对应于静态分析，则确定的加固强度为弱加固强度；如果被攻击手段对应于动态分析，则确定的加固强度为中加固强度；以及如果被攻击手段对应于静态分析和动态分析两者的组合，则确定的加固强度为强加固强度。

[52]本领域技术人员可以理解，在不同的网络应用场景中，攻击手段纷繁复杂，加固强度可按需确定，在此不做赘述。

[53]在 108，基于该加固强度选择自动加固方案。

[54]基于在 106 确定的加固强度，可选择自动加固方案。该自动加固方案可依场景进行不同的设置。

[55]在本公开一实施例中，基于弱加固强度选择的自动加固方案采用静态混淆；基于中加固强度选择的自动加固方案采用动态混淆；而基于强加固强度选择的自动加固方案采用静态混淆和动态混淆的组合。

[56]同样，本领域技术人员可以理解，在不同的网络应用场景中，攻击手段纷繁复杂，自动加固方案可按需设置。

[57]举例而言，静态混淆手法可包括：将 App 的源代码中敏感字符串做随机加密处理、在运行时进行对字符串动态解密的字符串加密；对 App 应用中的类名称、函数名称进行混淆操作的符号混淆；对源代码中的变量名称进行做混淆操作的局部变量名称混淆；将执行控制逻辑变换为平坦的控制逻辑，从抽象语法树层面进行深度混淆的控制流平坦化；对代码中的运算表达式进行等效转换的指令替换；在混淆过程中引入随机性技术的混淆多样化；以及将代码中分支跳转判断条件的不透明谓词，等等。

[58]由此，通过多样化静态防护手段，实现控制流混淆、符号混淆等来全方位保护代码，大大的提高代码静态逆向能力。

[59]举例而言，动态混淆手法可包括：对 App 应用进行的防动态调试，当检测到配置防动态调试功能的类、方法、函数被逆向工具进行动态调试时，App 应用进行自动退出运行操作；当检测到针对 App 应用的注入操作时，App 应用进行自动退出运行操作的防动态注入；当检测到动态 hook 时，App 进行自动退出操作的防 HOOK 保护；当发现 App 应用中的代码段被篡改时，App 应用进行自动退出运行的代码段校验；以及对 App 中指定的函数进行的完整性校验，当应用被重新签名和代码的完整性遭到破坏时，其触发 App 程序闪退。

[60]在本公开一实施例中，自动加固方案可按需确定对以上的静态混淆手法和动态混淆手法的采用规则。本领域技术人员可以理解，本公开的自动加固方案不限于采用以上所列举的静态混淆手法和动态混淆手法，并可随技术的进步纳入其他混淆手段。

[61]在 110，在该加固位置执行所选择的自动加固方案，以将该自动加固对性能的影响控制在可接受范围内。

[62]将在 108 选择的自动加固方案在 104 确定的加固位置执行。该自动加固方案执行后，对混淆的强度和系统的稳定性进行自动评估。如果性能超过规则所预订的膨胀最大值，

则进行加固方案回滚，也就是不执行加固；如果属于性能膨胀最大值范围内，则进行加固。

[63]在本公开一实施例中，通过编译参数的逐步收敛，将混淆的性能影响控制在可以接受的范围内，以使得最终选定的自动加固方案是强度和稳定并重的加固方案。

[64]本公开所揭示的移动端应用的智能混淆方法纳入了安全切面体系，通过安全切面发现移动应用程序真实被攻击的风险点，从而针对性进行加固。本公开的移动端应用的智能混淆方法通过对调用链的还原，实现安全内视和追溯，具有强大的感知能力；并且通过对调用链节点信息的分析，来实现高效针对性加固。

[65]图 2 是示出根据本公开一实施例的安全切面系统的示意图。

[66]在图 2 的安全切面系统中，安全切面防御层嵌入到了整个 APP 的框架中，同时保持独立运行的规则引擎和升级能力。APP 中所有重点安全相关的接口，都有切面进行保护。实际上就是为直接的业务关注点，即直切关注点，提供安全服务，就是横切关注点。横切关注点的模块化即为切面。

[67]采用面向切面的软件安全架构来进行安全切入，实现在不修改源代码的情况下给程序动态添加安全功能。安全切面判断某个调用是否合理时，会追溯查看调用的链路信息，对链路的调用风险进行透视，以做出获得准确全面的判断。

[68]具体而言，如图 2 所示的安全切面系统采用切面（Aspect）来定义安全的切入点和通知对象，采用连接点（Joint Point）来感知安全事件、对象、服务、业务数据等，还采用切入点（Point Cut）对连接点进行全部操作的定义，对需要安全看护的点进行环绕增强，此时需要对数据的透视能力。

[69]如图 2 所示，举例而言，核心逻辑、关键行为、隐私信息等在根据本公开一实施例的场景中为需要安全看护的点，为进行环绕增强可各自定义一个或多个切入点。如图 2 所示的安全切面系统在本实施例中可通过隐私保护、安全防御、异常检测、风险隔离等多种技术手段来对上述安全切入点进行环绕增强。

[70]如图 2 所示的安全切面系统利用通知（Advice）来描述切面要完成的工作，并描述何时执行这个工作。在根据本公开一实施例的场景中，可应用的通知类型有：Before — 在方法调用之前调用通知，After — 在方法完成之后调用通知，无论方法执行成功与否，After-returning — 在方法执行成功之后调用通知，以及 After-throwing — 在方法抛出异常后进行通知。

[71]如图 2 所示的安全切面系统还利用代理（Proxy）对目前服务进行安全能力的增强，诸如日志记录，性能监控，异常处理、熔断限流这样的非核心功能被单独抽取出来，与业务代码分离，横切在核心业务代码之上。

[72]获取全链路信息是安全切面系统的核心。由于融入到了整个系统内部，因此安全切面系统可以获得大量的这种真实的、完整的、细致的数据来建设异常检测能力和攻击阻断能力。也就是通过对调用链路的还原，实现安全内视和追溯，并通过代码节点信息分析和定位攻击风险点，来进行针对性加固。

[73]图 3 是示出根据本公开一实施例的基于安全切面进行智能混淆的过程的示意图。

[74]在安全切面层，由于对系统资源进行接管，可获取到每个业务对于系统资源的调用情况。例如，从堆栈中，通过每个符号所对应的模块名、路径，来判断是否属于原生程序的调用来源，否则上报到云端进行分析。

[75]在本公开一实施例中，可判断调用链中是否存在 Hook 框架（如 Xpose）以及外部动态库的调用地址或者符号信息。

[76]在本公开一实施例中，可在调用链中判断调用来源地址是否存在 Hook 风险，举例而言，函数地址对应的起始地址未跳转汇编指令，或系统函数的地址属于外部第三方的一模块。

[77]云端数据分析服务器，会将不同的堆栈和被攻击节点（在本实施例中为符号）进行去重，以及分析出对应的被攻击符号、被攻击手段（例如，反射调用、代码注入、木马劫持等等）。被攻击符号会通过云端服务器进行包相关信息关联，找到对应的模块、代码仓库，以及对应被攻击符号所对应的代码位置。

[78]智能加固服务器接收到加固指令后，在函数开头以及编译器，根据对应攻击手法生成加固规则，并将混淆参数在编译器进行配置，并在对应函数头部进行插入相关符号用于 ollvm 编译器识别。攻击手法和对应的加固方案可人工配制成规则，并输入对应强度所能容忍的性能膨胀最大值范围，自动化执行到后面的流程中。在本实施例中，加固手段包括 3 类：常量保护、代码混淆、符号混淆。

[79]在尝试编译的过程中，会根据编译报错，自动进行相关编译配置的优化，进行重试，在重试一定次数之后若编译仍然不成功则放弃加固。如果编译成功，则生成未加固产物（原产物）、加固产物。

[80]最后对加固产物通过模拟执行对应加固函数，进行性能比较，如果性能超过规则所

预订的膨胀最大值，则进行加固方案回滚，也就是不执行加固；如果属于性能膨胀最大值范围内，则进行加固。

[81]图 4 是示出根据本公开一实施例的基于调用链异常判断不同被攻击节点的过程的示意图。该实施例利用动态代理的 Hook 机制来实现针对调用链的不同节点和不同粒度的精确切入。

[82]如图 4 所示的 Android 进程实质上是 Linux 进程，其动态链接了一堆动态链接库 so。该 Linux 进程中主要运行的是虚拟机，该虚拟机包含类加载器、对象/内存管理、线程调度等，为 JAVA 提供运行时环境。虚拟机中包含 JAVA FrameWork 和应用 JAVA 代码两部分。移动端应用通常还包含 navtive 代码，该 native 代码会和虚拟机一起在 Linux 进程中运行。Android 进程交换数据依赖于 linux 内核提供进程通信接口，例如：Binder 通信、Socket 通信等等。

[83]Hook 机制可以通过修改 Android 进程中的这些组件来达到预期目的。在本实施例中，本公开的安全切面系统可通过检测 Hook 调用链的不同节点，来判断可能的被攻击位置和被攻击手段。

[84]如图 4 中节点 1，Hook 机制作用于 Java 层。虚拟机提供了作为标准编程接口的反射/动态代理。反射 API 可以帮助访问到 private 属性并修改，动态代理可以直接从 Interface 中动态地构造出代理对象，并去监控该对象。即，在节点 1（即，反射/动态代理 hook 节点），Hook 机制通常用动态代理构造出代理对象，然后用反射 API 去替换进程中的对象，从而达到 hook 的目的。针对该节点 1 的调用如果判断为异常，则可针对性地采用针对静态分析的混淆，例如变量名称混淆、数据流混淆等。

[85]如图 4 中节点 2，java 代码和 native 之间的调用是通过 JNI 接口调用的，所有 JNI 接口的函数指针都会被保存在虚拟机的表中。所以，作用于节点 2（即，JNI 接口调用 hook 节点）的 Hook 机制可以通过修改函数指针实现。针对该节点 2 的调用如果判断为异常，则可针对性地采用针对静态分析的数据流混淆等，以模糊函数的调用关系。

[86]如图 4 中节点 3，java 代码的执行依靠虚拟机的类加载器 ClassLoader 去加载。所以，作用于节点 3（即，类加载路径 hook 节点）的 Hook 机制通过修改 ClassLoader 加载 java class 的 Path 路径达到目的。针对该节点 3 的调用如果判断为异常，则可针对性地采用针对静态分析的路径混淆等。

[87]如图 4 中节点 4，Hook 机制作用于虚拟机。虚拟机为 java 提供运行时环境，所有的

java method 都保存在虚拟机的 Map 中维护，每个 Java Method 都有个是否是 JNI 函数的标志位，如果是 JNI 函数则去查找对应的 native 函数。所以，作用于节点 4（即，JNI 函数 hook 节点）的 Hook 机制通过将要 hook 的函数修改为 JNI 函数，实现对应的 native 函数。针对该节点 4 的调用如果判断为异常，则可针对性地采用针对动态分析的混淆，例如虚拟机保护等。

[88]如图 4 中节点 5，Android 进程加载动态链接库时都通过 dlopen()函数去把 SO 读入到当前进程中的内存区域中。当调用 so 代码时，直接跳转到 so 的内存区域去执行。so 对外提供的函数表及函数地址也都在这块内存中。所以，作用于节点 5（即，动态链接库 hook 节点）的 Hook 机制会修改该函数地址。针对该节点 5 的调用如果判断为异常，则可针对性地采用针对动态分析的混淆，例如防动态 hook 等。

[89]如图 4 中节点 6，有些场景需要 hook so 内部函数，此时要用到 inline hook 技术。基本原理是在目标函数执行区域中插入 Jump 指令，使得 cpu 跳转到 hook 函数（shellcode）中。inline hook 的实现和指令平台强相关。所以，作用于节点 6（即，inline hook 节点）的 Hook 机制会需要针对性地实现指令集。针对该节点 6 的调用如果判断为异常，则可针对性地采用针对动态分析的混淆，例如防动态调试等。

[90]如图 4 所示，基于调用链的不同节点的检测，可识别出该调用链中的被攻击节点，从而进一步确定加固位置。本领域技术人员可以理解，被攻击节点可以是一个或多个，加固方案可针对性地逐个在各个加固位置执行加固。

[91]图 5 是示出根据本公开一实施例的移动端应用的智能混淆系统 500 的框图。

[92]该移动端应用的智能混淆系统 500 包括检测模块 502、识别模块 504、加固选择模块 506 和加固执行模块 508。

[93]检测模块 502 利用安全切面对调用链进行调用检测，以发现异常调用。

[94]如上所述，安全切面体系由于具备了对 App 运行时对核心资源访问的内视能力，通过对调用链的检测，可以发现有风险的调用，并推导加固方案的生成。

[95]识别模块 504 针对该异常调用识别该调用链中的被攻击节点以确定加固位置。

[96]在 Android 移动端应用中，AOP 编程的具体实现用到了动态代理的 Hook 机制。hook 机制可以实现针对调用链的不同节点和不同粒度的精确切入。当针对异常调用识别出调用链的被攻击节点时，即可确定加固位置。本领域技术人员可以理解，被攻击节点可涉及一个或多个，可针对性地逐个在各个加固位置进行加固。

[97]进一步地，识别模块 504 针对该异常调用识别被攻击手段以确定加固强度。

[98]由于 hook 机制可以实现针对调用链的不同节点和不同粒度的精确切入，因此其对应的调用链的被攻击手段通常对应于静态分析（例如反射调用）和动态分析（例如代码注入、栈跟踪、木马劫持等）。

[99]与静态分析攻击相对应的保护方法是静态混淆。静态混淆主要可以分为混淆控制流和混淆数据结构。举例而言，静态控制流混淆方法包括花指令、控制流平面化等，而数据结构混淆方法包括合并和拆分类和结构体，隐藏虚表等。与动态分析攻击相对应的保护方法是动态混淆。动态混淆主要包括自修改代码和虚拟机保护等混淆手段。

[100]在识别出调用链所受攻击的攻击手段是对应于静态分析、动态分析、还是静态分析和动态分析两者的组合之后，识别模块 504 确定对应的加固强度。

[101]在本公开一实施例中，如果被攻击手段仅对应于静态分析，则确定对应的加固强度为静态混淆；如果被攻击手段仅对应于动态分析，则确定对应的加固强度为动态混淆；而如果被攻击手段对应于静态分析和动态分析两者的组合，则确定对应的加固强度为静态混淆和动态混淆的组合。

[102]加固选择模块 506 基于该加固强度选择自动加固方案。

[103]基于识别模块 504 所确定的加固强度，加固选择模块 506 可选择自动加固方案。该自动加固方案可依场景进行不同的设置。

[104]举例而言，静态混淆手法可包括：字符串加密、符号混淆、局部变量名称混淆、控制流平坦化、指令替换、混淆多样化；以及不透明谓词，等等；而动态混淆手法可包括：防动态调试、防动态注入、防 HOOK 保护、代码段校验、完整性校验，等等。

[105]由此，通过多样化静态防护手段，实现控制流混淆、符号混淆等来全方位保护代码，大大的提高代码静态逆向能力。

[106]本公开的自动加固方案可按需确定对以上的静态混淆手法和动态混淆手法的采用规则。

[107]加固执行模块 508 在该加固位置执行所选择的自动加固方案，以将该自动加固对性能的影响控制在可接受范围内。

[108]将加固选择模块 506 所选择的自动加固方案在识别模块 504 所确定的加固位置执行。该自动加固方案执行后，对混淆的强度和系统的稳定性进行自动评估。

[109]在本公开一实施例中，通过编译参数的逐步收敛，将混淆的性能影响控制在可以接受的范围内，以使得最终选定的自动加固方案是强度和稳定并重的加固方案。

[110]本公开所揭示的移动端应用的智能混淆系统纳入了安全切面体系，通过安全切面发现移动应用程序真实被攻击的风险点，从而针对性进行加固。本公开的移动端应用的智能混淆系统通过对调用链的还原，实现安全内视和追溯，具有强大的感知能力；并且通过对调用链节点信息的分析，来实现高效针对性加固。

[111]以上描述的移动端应用的智能混淆方法和系统的各个步骤和模块可以用硬件、软件、或其组合来实现。如果在硬件中实现，结合本发明描述的各种说明性步骤、模块、以及电路可用通用处理器、数字信号处理器（DSP）、专用集成电路（ASIC）、现场可编程门阵列（FPGA）、或其他可编程逻辑组件、硬件组件、或其任何组合来实现或执行。通用处理器可以是处理器、微处理器、控制器、微控制器、或状态机等。如果在软件中实现，则结合本发明描述的各种说明性步骤、模块可以作为一条或多条指令或代码存储在计算机可读介质上或进行传送。实现本发明的各种操作的软件模块可驻留在存储介质中，如RAM、闪存、ROM、EPROM、EEPROM、寄存器、硬盘、可移动盘、CD-ROM、云存储等。存储介质可耦合到处理器以使得该处理器能从/向该存储介质读写信息，并执行相应的程序模块以实现本发明的各个步骤。而且，基于软件的实施例可以通过适当的通信手段被上载、下载或远程地访问。这种适当的通信手段包括例如互联网、万维网、内联网、软件应用、电缆（包括光纤电缆）、磁通信、电磁通信（包括 RF、微波和红外通信）、电子通信或者其他这样的通信手段。

[112]还应注意，这些实施例可能是作为被描绘为流程图、流图、结构图、或框图的过程来描述的。尽管流程图可能会把诸操作描述为顺序过程，但是这些操作中有许多操作能够并行或并发地执行。另外，这些操作的次序可被重新安排。

[113]所公开的方法、装置和系统不应以任何方式被限制。相反，本发明涵盖各种所公开的实施例（单独和彼此的各种组合和子组合）的所有新颖和非显而易见的特征和方面。所公开的方法、装置和系统不限于任何具体方面或特征或它们的组合，所公开的任何实施例也不要求存在任一个或多个具体优点或者解决特定或所有技术问题。

[114]上面结合附图对本发明的实施例进行了描述，但是本发明并不局限于上述的具体实施方式，上述的具体实施方式仅仅是示意性的，而不是限制性的，本领域的普通技术人员在本发明的启示下，在不脱离本发明宗旨和权利要求所保护的范围情况下，还可做出很多更改，这些均落在本发明的保护范围之内。

# 权利要求书

1.一种移动端应用的智能混淆方法，包括：

利用安全切面对调用链进行调用检测，以发现异常调用；

针对所述异常调用识别所述调用链中的被攻击节点以确定加固位置；

针对所述异常调用识别被攻击手段以确定加固强度；

基于所述加固强度选择自动加固方案；以及

在所述加固位置执行所选择的自动加固方案，以将所述自动加固对性能的影响控制在可接受范围内。

2.如权利要求 1 所述的方法，所述异常调用基于动态代理的 Hook 机制。

3.如权利要求 1 所述的方法，所述异常调用与静态分析或动态分析相对应。

4.如权利要求 1 所述的方法，所述安全切面是和业务相交织且平行的安全层。

5.如权利要求 1 所述的方法，所述调用链中的被攻击节点包括反射/动态代理 hook 点、JNI 接口调用 hook 点、类加载路径 hook 点、JNI 函数 hook 点、动态链接库 hook 点、inline hook 点中的一个或多个。

6.如权利要求 1 所述的方法，所述被攻击手段包括对应于静态分析的被攻击手段、或对应于动态分析的被攻击手段、或对应于静态分析和动态分析两者的被攻击手段。

7.如权利要求 1 所述的方法，所述加固强度可在弱加固强度、中加固强度和强加固强度之中选择。

8.如权利要求 7 所述的方法，基于所述弱加固强度选择的自动加固方案采用静态混淆。

9.如权利要求 7 所述的方法，基于所述中加固强度选择的自动加固方案采用动态混淆。

10.如权利要求 7 所述的方法，基于所述强加固强度选择的自动加固方案采用动态混淆或者静态混淆和动态混淆的组合。

11.一种移动端应用的智能混淆系统，包括：

检测模块，利用安全切面对调用链进行调用检测，以发现异常调用；

识别模块，针对所述异常调用识别所述调用链中的被攻击节点以确定加固位置，并针对所述异常调用识别被攻击手段以确定加固强度；

加固选择模块，基于所述加固强度选择自动加固方案；以及

加固执行模块，在所述加固位置执行所选择的自动加固方案，以将所述自动加固对性能的影响控制在可接受范围内。

12.如权利要求 11 所述的系统，所述异常调用与静态分析或动态分析相对应。

13.如权利要求 11 所述的系统，所述被攻击手段包括对应于静态分析的被攻击手段、或对应于动态分析的被攻击手段、或对应于静态分析和动态分析两者的被攻击手段。

14.如权利要求 11 所述的系统，所述加固强度可在弱加固强度、中加固强度和强加固强度之中选择。

15.如权利要求 14 所述的系统，所述加固选择模块基于所述弱加固强度选择的自动加固方案采用静态混淆。

16.如权利要求 15 所述的系统，所述加固选择模块基于所述中加固强度选择的自动加固方案采用动态混淆。

17.如权利要求 15 所述的系统，所述加固选择模块基于所述强加固强度选择的自动加固方案采用动态混淆或者静态混淆和动态混淆的组合。

18. 一种存储有指令的计算机可读存储介质，当所述指令被执行时使得机器执行如权利要求 1-10 中任一项所述的方法。

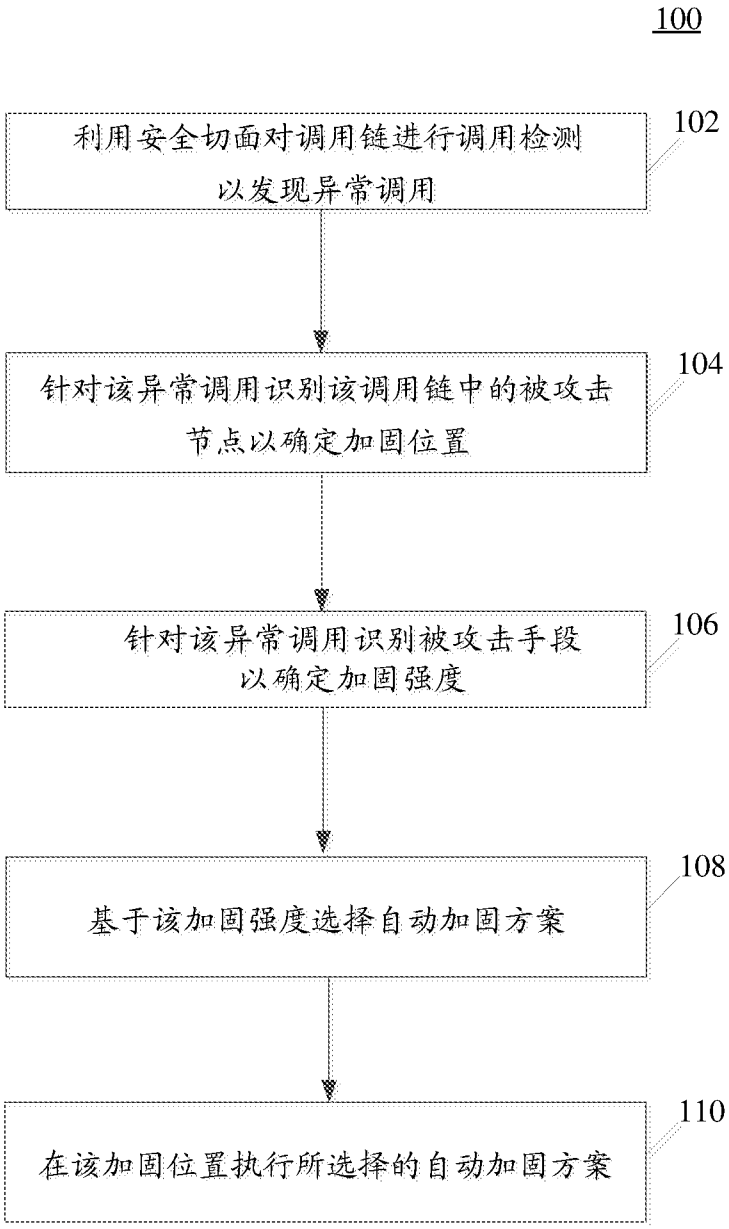
100
利用安全切面对调用链进行调用检测以发现异常调用
102
针对该异常调用识别该调用链中的被攻击节点以确定加固位置
104
针对该异常调用识别被攻击手段以确定加固强度
106
基于该加固强度选择自动加固方案
108
在该加固位置执行所选择的自动加固方案
110

图 1

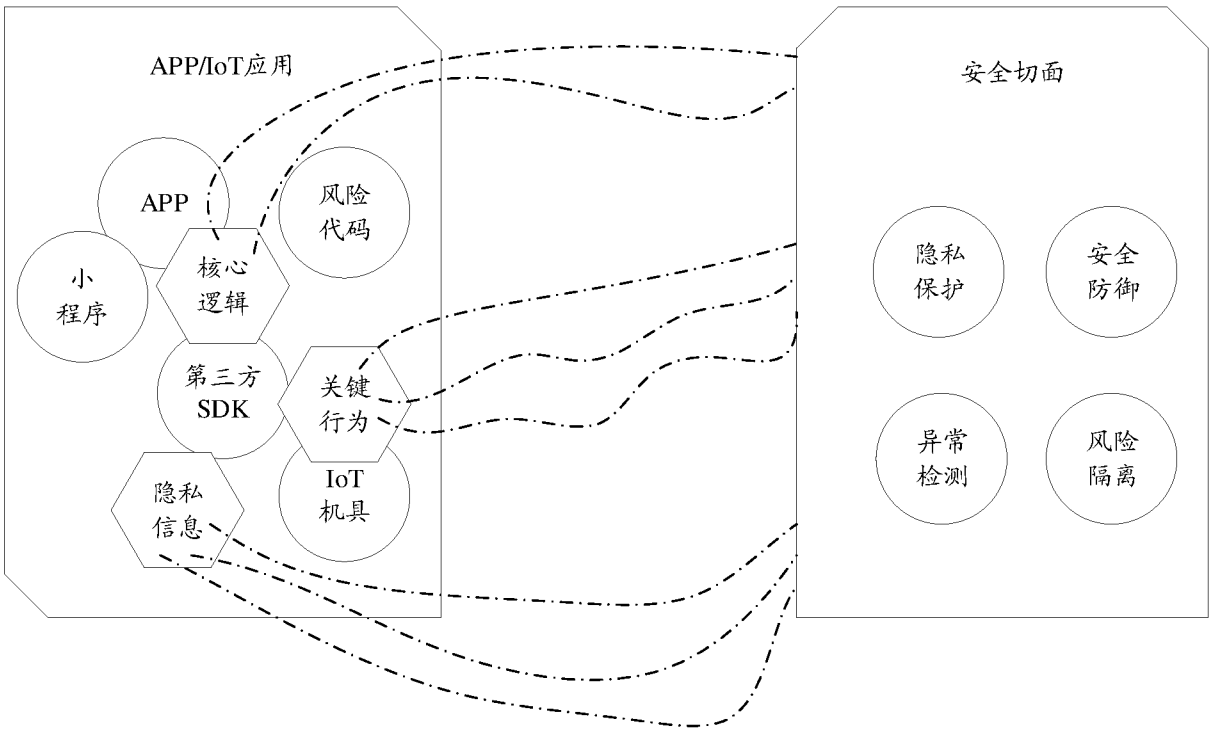
APP/IoT应用
APP
风险代码
小程序
核心逻辑
第三方SDK
关键行为
隐私信息
IoT机具
安全切面
隐私保护
安全防御
异常检测
风险隔离

图 2

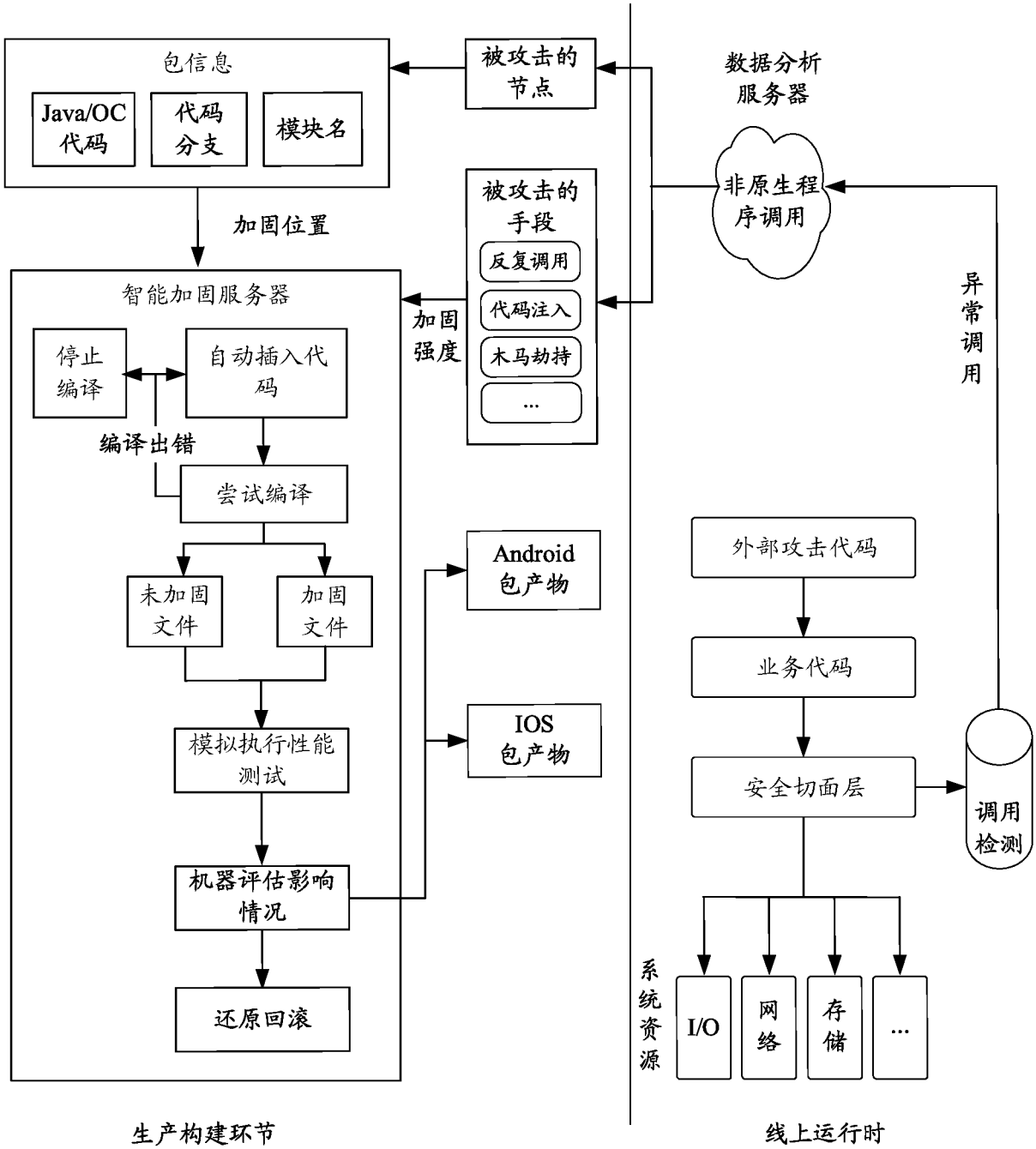
包信息
Java/OC 代码
代码分支
模块名
被攻击的节点
数据分析服务器
非原生程序调用
被攻击的手段
反复调用
代码注入
木马劫持
...
加固位置
智能加固服务器
加固强度
异常调用
停止编译
自动插入代码
编译出错
尝试编译
未加固文件
加固文件
模拟执行性能测试
机器评估影响情况
还原回滚
Android 包产物
IOS 包产物
外部攻击代码
业务代码
安全切面层
调用检测
系统资源
I/O
网络
存储
...
生产构建环节
线上运行时

图 3

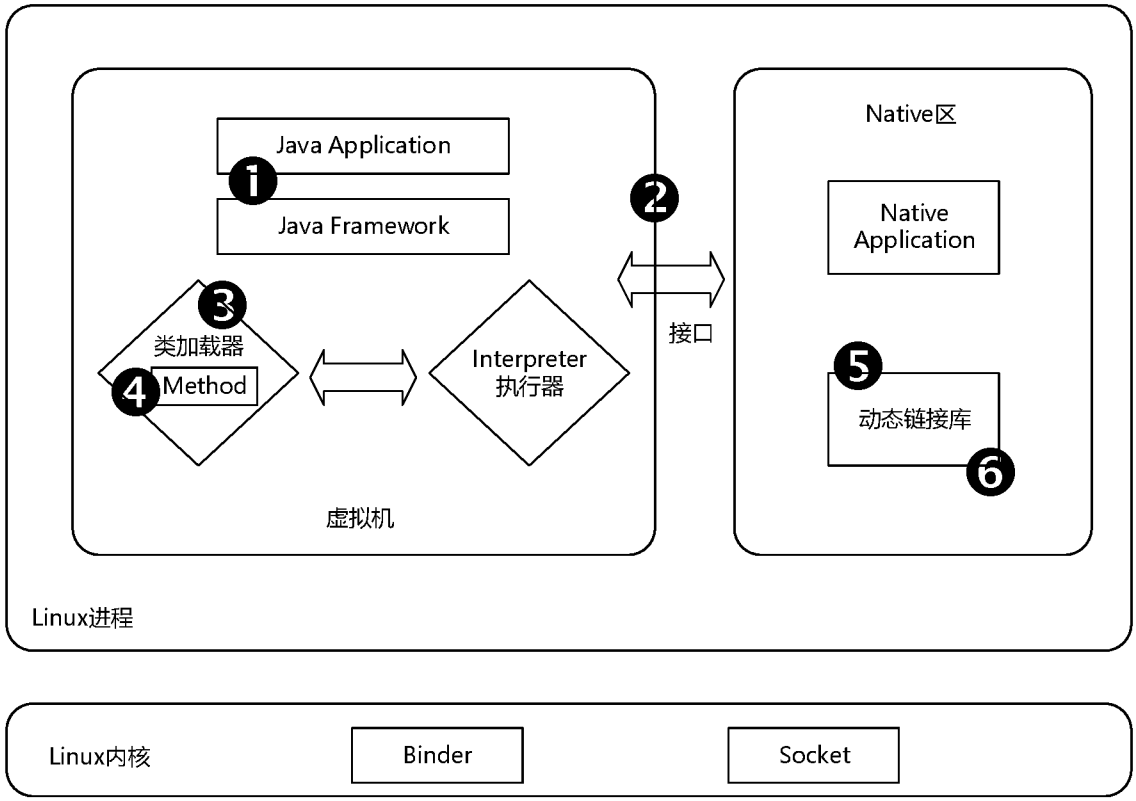
Java Application
1
Java Framework
2
接口
Native区
Native Application
3
类加载器
4
Method
Interpreter 执行器
5
动态链接库
6
虚拟机
Linux进程
Linux内核
Binder
Socket

图 4

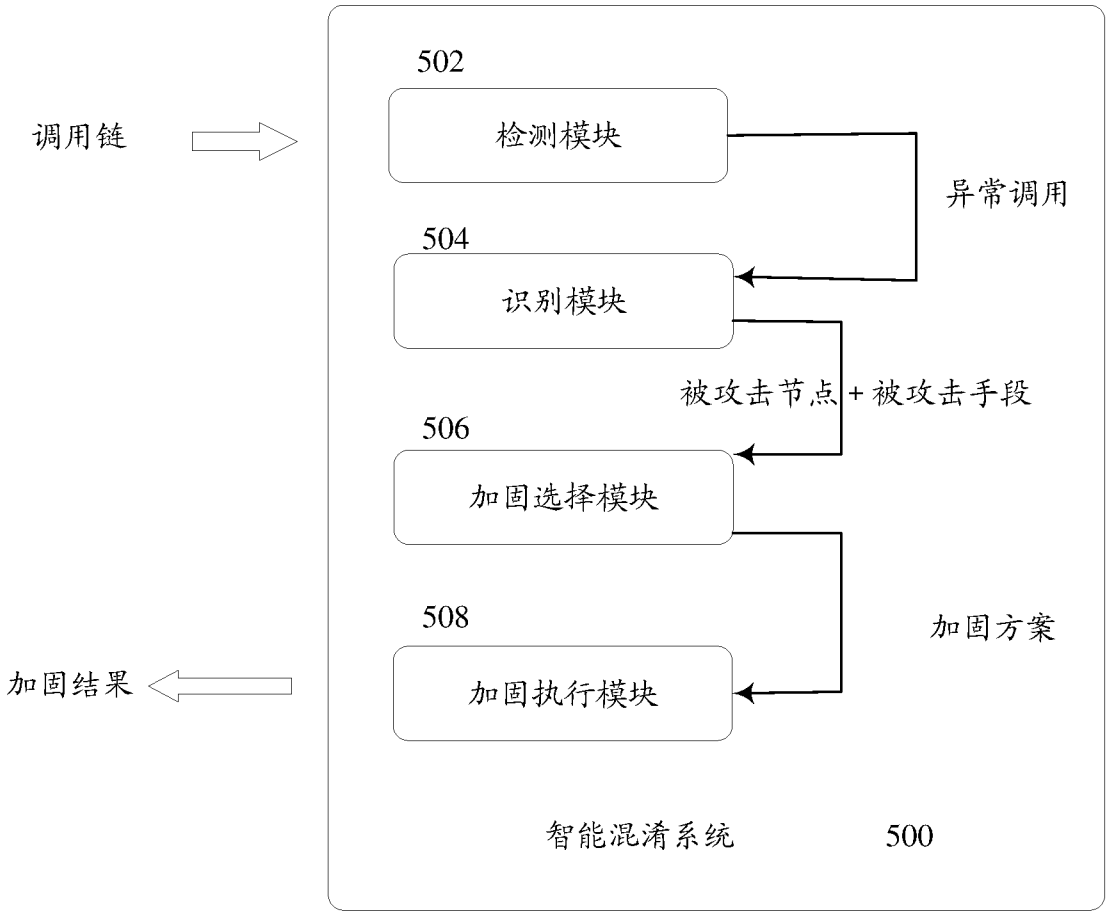
调用链
502
检测模块
异常调用
504
识别模块
被攻击节点+被攻击手段
506
加固选择模块
加固方案
508
加固结果
加固执行模块
智能混淆系统
500

图 5

# INTERNATIONAL SEARCH REPORT

International application No.
**PCT/CN2022/104296**

## A. CLASSIFICATION OF SUBJECT MATTER

G06F 21/56(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)

G06F21/-

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)

CNTXT; ENTXT; ENTXTC; DWPI; CNKI: 移动, 应用, 混淆, 智能, 链, 调用, 攻击, 加固, mobile, application, confusion, intelligent, chain, call, attack, reinforcement

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

| Category* | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|---|---|---|
| PX | CN 113779578 A (ALIPAY (HANGZHOU) INFORMATION TECHNOLOGY CO., LTD.) 10 December 2021 (2021-12-10)<br>claims 1-18 | 1-18 |
| Y | CN 111614624 A (ALIPAY (HANGZHOU) INFORMATION TECHNOLOGY CO., LTD.) 01 September 2020 (2020-09-01)<br>claims 1-14, and description, paragraphs [0020]-[0095] | 1-18 |
| Y | CN 112804184 A (ALIBABA GROUP HOLDING LIMITED) 14 May 2021 (2021-05-14)<br>claims 1-29, and description, paragraphs [0087]-[0377] | 1-18 |
| Y | CN 111428237 A (ALIPAY (HANGZHOU) INFORMATION TECHNOLOGY CO., LTD.) 17 July 2020 (2020-07-17)<br>claims 1-18, and description, paragraphs [0062]-[0270] | 1-18 |
| Y | CN 107992724 A (SICHUAN UNIVERSITY) 04 May 2018 (2018-05-04)<br>claims 1-4, and description, paragraphs [0021]-[0029] | 1-18 |
| A | US 2015363580 A1 (APPLE INC.) 17 December 2015 (2015-12-17)<br>entire document | 1-18 |

☐ 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
"&" document member of the same patent family

| Date of the actual completion of the international search | Date of mailing of the international search report |
|---|---|
| **19 August 2022** | **07 September 2022** |
| Name and mailing address of the ISA/CN<br>**China National Intellectual Property Administration (ISA/CN)**<br>**No. 6, Xitucheng Road, Jimenqiao, Haidian District, Beijing 100088, China**<br>Facsimile No. **(86-10)62019451** | Authorized officer<br><br>Telephone No. |

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

International application No.
**PCT/CN2022/104296**

| Patent document cited in search report | Publication date (day/month/year) | Patent family member(s) | Publication date (day/month/year) |
|---|---|---|---|
| CN 113779578 A | 10 December 2021 | None | |
| CN 111614624 A | 01 September 2020 | None | |
| CN 112804184 A | 14 May 2021 | None | |
| CN 111428237 A | 17 July 2020 | None | |
| CN 107992724 A | 04 May 2018 | None | |
| US 2015363580 A1 | 17 December 2015 | US 9639673 B2 | 02 May 2017 |

# 国际检索报告

国际申请号
PCT/CN2022/104296

A. 主题的分类

G06F 21/56(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)

G06F21/-

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））

CNTXT;ENTXT;ENTXTC;DWPI;CNKI:移动,应用,混淆,智能,链,调用,攻击,加固, mobile,application,confusion,intelligent,chain,call,attack,reinforcement

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 |
|---|---|---|
| PX | CN 113779578 A (支付宝杭州信息技术有限公司) 2021年12月10日 (2021-12-10)<br>权利要求1-18 | 1-18 |
| Y | CN 111614624 A (支付宝杭州信息技术有限公司) 2020年9月1日 (2020-09-01)<br>权利要求1-14，说明书第[0020]-[0095] | 1-18 |
| Y | CN 112804184 A (阿里巴巴集团控股有限公司) 2021年5月14日 (2021-05-14)<br>权利要求1-29，说明书第[0087]-[0377]段 | 1-18 |
| Y | CN 111428237 A (支付宝杭州信息技术有限公司) 2020年7月17日 (2020-07-17)<br>权利要求1-18，说明书第[0062]-[0270]段 | 1-18 |
| Y | CN 107992724 A (四川大学) 2018年5月4日 (2018-05-04)<br>权利要求1-4，说明书第[0021]-[0029]段 | 1-18 |
| A | US 2015363580 A1 (APPLE INC) 2015年12月17日 (2015-12-17)<br>全文 | 1-18 |

☐ 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

“&” 同族专利的文件

| 国际检索实际完成的日期 | 国际检索报告邮寄日期 |
|---|---|
| 2022年8月19日 | 2022年9月7日 |
| ISA/CN的名称和邮寄地址<br>中国国家知识产权局(ISA/CN)<br>中国北京市海淀区蓟门桥西土城路6号 100088<br>传真号 (86-10)62019451 | 受权官员<br>邢爽<br>电话号码 (86-10)62411289 |

# 国际检索报告
## 关于同族专利的信息

国际申请号
PCT/CN2022/104296

| 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 113779578 A | 2021年12月10日 | 无 | |
| CN 111614624 A | 2020年9月1日 | 无 | |
| CN 112804184 A | 2021年5月14日 | 无 | |
| CN 111428237 A | 2020年7月17日 | 无 | |
| CN 107992724 A | 2018年5月4日 | 无 | |
| US 2015363580 A1 | 2015年12月17日 | US 9639673 B2 | 2017年5月2日 |