(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 110727925 A
(43)申请公布日 2020.01.24

(21)申请号 201910785003.3

(22)申请日 2019.08.23

(71)申请人 北京邮电大学
地址 100876 北京市海淀区西土城路10号

(72)发明人 徐国爱 郭燕慧 阚泽亮

(74)专利代理机构 北京风雅颂专利代理有限公司 11403
代理人 李弘

(51)Int.Cl.
G06F 21/14(2013.01)
G06F 21/56(2013.01)

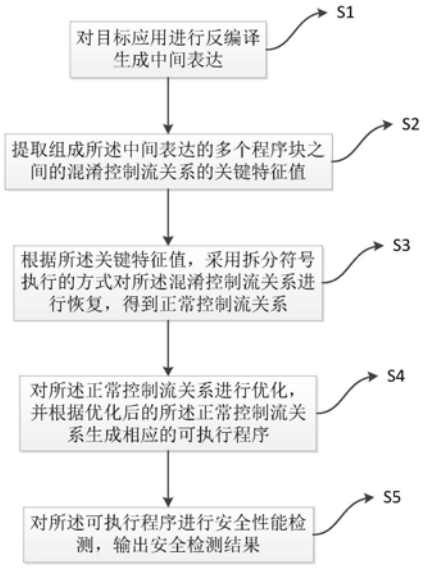
权利要求书2页 说明书9页 附图8页

(54)发明名称

一种目标应用安全检测方法、装置与电子设备

(57)摘要

本发明公开了一种目标应用安全检测方法、装置与电子设备，能够对目标应用进行准确反混淆处理与有效安全检测。所述方法包括：对目标应用进行反编译，生成中间表达；提取组成中间表达的多个程序块之间的混淆控制流关系的关键特征值；根据关键特征值，采用拆分符号执行的方式对混淆控制流关系进行恢复，得到正常控制流关系；对正常控制流关系进行优化，并根据优化后的正常控制流关系生成相应的可执行程序；对可执行程序进行安全性能检测，输出安全检测结果。所述装置包括反编译模块、特征提取模块、恢复模块、优化模块与检测模块。所述电子设备包括存储器、处理器及存储在存储器上并可在处理器上运行实现目标应用安全检测方法的计算机程序。



CN 110727925 A

1.一种目标应用安全检测方法,其特征在于,包括:

对目标应用进行反编译,生成中间表达;

提取组成所述中间表达的多个程序块之间的混淆控制流关系的关键特征值;

根据所述关键特征值,采用拆分符号执行的方式对所述混淆控制流关系进行恢复,得到正常控制流关系;

对所述正常控制流关系进行优化,并根据优化后的所述正常控制流关系生成相应的可执行程序;

对所述可执行程序进行安全性能检测,输出安全检测结果。

2.根据权利要求1所述的方法,其特征在于,所述提取组成所述中间表达的多个程序块之间的混淆控制流关系的关键特征值,包括:

根据代码指令内容,从所述多个程序块中选取出调度程序块;

根据所述调度程序块中的跳转指令确定所述多个程序块之间的Switch调度结构;

根据所述调度程序块中的比较指令确定跳转路由变量;

所述关键特征值包括所述Switch调度结构与所述跳转路由变量。

3.根据权利要求2所述的方法,其特征在于,所述根据所述调度程序块中的比较指令确定跳转路由变量,包括:

V=立即数+ins.address+n

其中,V表示跳转路由变量,ins.address表示所述比较指令的地址,n表示偏移量,在ARM指令集模式下,所述偏移量n=8,在Thumb指令集模式下,所述偏移量n=4。

4.根据权利要求1所述的方法,其特征在于,所述根据所述关键特征值,采用拆分符号执行的方式对所述混淆控制流关系进行恢复,得到正常控制流关系,包括:

所述多个程序块包括序幕程序块、调度程序块、相关程序块与返回程序块,为所述序幕程序块、所述相关程序块与所述返回程序块创建乱序块队列,并将所述序幕程序块设置在队首;

从所述乱序块队列中的队首程序块开始,根据所述关键特征值采用所述拆分符号执行的方式确定所述乱序块序列中每个所述程序块的后继程序块,对所述乱序块队列中的多个所述程序块的顺序进行调整,得到正序块队列;

根据所述正序块队列确定所述正常控制流关系。

5.根据权利要求4所述的方法,其特征在于,所述根据所述关键特征值采用所述拆分符号执行的方式确定所述乱序块序列中每个所述程序块的后继程序块,对所述乱序块队列中的多个所述程序块的顺序进行调整,包括:

A:设置执行指针指向所述乱序块队列的第一位,设置交换指针指向所述乱序块队列的第二位;

B:若所述执行指针当前所指向程序块位于队尾,结束顺序调整;

反之,对所述执行指针所指向的所述程序块的跳转类型进行判定,若为条件跳转,执行步骤C,若为无条件跳转,执行步骤G;

C:根据所述关键特征值,确定所述执行指针当前所指向程序块的True跳转分支所对应的True分支后继程序块;

D:将所述True分支后继程序块与所述交换指针当前所指向程序块的位置进行交换,并

将所述交换指针后移一位；

E:根据所述关键特征值,确定所述执行指针当前所指向程序块的False跳转分支所对应的False分支后继程序块；

F:将所述False分支后继程序块与所述交换指针当前所指向程序块的位置进行交换,并将所述交换指针后移一位,执行步骤I；

G:根据所述关键特征值,确定所述执行指针当前所指向程序块的唯一跳转分支所对应的Only分支后继程序块；

H:将所述Only分支后继程序块与所述交换指针当前所指向程序块的位置进行交换,并将所述交换指针后移一位,执行步骤I。

I:将所述执行指针后移一位,返回步骤B。

6.根据权利要求5所述的方法,其特征在于,在对所述乱序块队列中的多个所述程序块的顺序进行调整的过程中,在将所述True分支后继程序块、所述False分支后继程序块或所述Only分支后继程序块与所述交换指针当前所指向程序块的位置进行交换之前,都将相应所述后继程序块的执行状态进行保存；

在所述执行指针后移一位之后,都对其所指向程序块的执行状态进行恢复。

7.根据权利要求1所述的方法,其特征在于,所述对所述正常控制流关系进行优化,包括:

对于通过无条件跳转相连接的两个所述程序块,若父节点的所述程序块的度不大于1,则将这两个所述程序块合并。

8.根据权利要求1所述的方法,其特征在于,所述对所述正常控制流关系进行优化,包括:

对于具有两个跳转分支的多个所述程序块,若这些所述程序块通过一个分支依次串联连接,并且这些所述程序块的另一个分支都指向同一个后继程序块,则将这些所述程序块间的跳转结构优化作为循环结构。

9.一种目标应用的安全检测装置,其特征在于,包括：

反编译模块,被配置为对目标应用进行反编译,生成中间表达；

特征提取模块,被配置为提取组成所述中间表达的多个程序块之间的混淆控制流关系的关键特征值；

恢复模块,被配置为根据所述关键特征值,采用拆分符号执行的方式对所述混淆控制流关系进行恢复,得到正常控制流关系；

优化模块,被配置为对所述正常控制流关系进行优化,并根据优化后的所述正常控制流关系生成相应的可执行程序；

检测模块,被配置为对所述可执行程序进行安全性能检测,输出安全检测结果。

10.一种电子设备,包括存储器、处理器及存储在存储器上并可在处理器上运行的计算机程序,其特征在于,所述处理器执行所述程序时实现如权利要求1至8任意一项所述的方法。

一种目标应用安全检测方法、装置与电子设备

技术领域

[0001] 本发明涉及智能终端软件安全技术领域，特别是指一种目标应用安全检测方法、装置与电子设备。

背景技术

[0002] 随着智能终端设备的快速普及，适用于智能终端的应用软件的数量也呈现爆炸式增长。移动应用快速增长的同时，恶意软件的数量与种类也开始不断增多，随之而来的安全隐患问题也造成严重的影响。经研究发现，越来越多的恶意软件开始利用软件混淆技术来逃脱现有的安全检测手段的检测，其中大部分都采用控制流平坦化方法为蓝本，对本身软件的程序控制流进行改变或复杂化，大大提高了破译难度。

[0003] 现有的一些可行的反混淆技术方法分为PC通用程序反混淆和安卓程序反混淆：

[0004] 在PC通用程序反混淆技术中，Francis Gabriel与El-Faramaw等人提出的反混淆技术视图将混淆的函数恢复为控制流图，但是未能解决基本块拆分和指令优化带来的挑战，且因为未考虑上下文继承与子函数调用的问题，在分析大型程序时成功率较低；Yadegari等人提出的在x86上自动反混淆二进制代码的通用方法并不适用于分析Android项目程序。

[0005] 在安卓程序反混淆技术中，现有的方法大多数都是处理Java层的反混淆。能够在一定程度上解决布局混淆的问题，但是面对控制流平坦化混淆所带来的程序逻辑的改变仍无能为力。

发明内容

[0006] 有鉴于此，本发明的目的在于提出一种能够对经控制流平坦化方法混淆的恶意软件进行准确反混淆处理以实现对所述恶意软件进行有效安全检测的目标应用安全检测方法、装置与电子设备。

[0007] 基于上述目的，本发明提供了一种目标应用安全检测方法，包括：

[0008] 对目标应用进行反编译，生成中间表达；

[0009] 提取组成所述中间表达的多个程序块之间的混淆控制流关系的关键特征值；

[0010] 根据所述关键特征值，采用拆分符号执行的方式对所述混淆控制流关系进行恢复，得到正常控制流关系；

[0011] 对所述正常控制流关系进行优化，并根据优化后的所述正常控制流关系生成相应的可执行程序；

[0012] 对所述可执行程序进行安全性能检测，输出安全检测结果。

[0013] 可选的，所述提取组成所述中间表达的多个程序块之间的混淆控制流关系的关键特征值，包括：

[0014] 根据代码指令内容，从所述多个程序块中选取出调度程序块；

[0015] 根据所述调度程序块中的跳转指令确定所述多个程序块之间的Switch调度结构；

[0016] 根据所述调度程序块中的比较指令确定跳转路由变量；

[0017] 所述关键特征值包括所述Switch调度结构与所述跳转路由变量。

[0018] 可选的，所述根据所述调度程序块中的比较指令确定跳转路由变量，包括：

[0019] V＝立即数+ins.address+n

[0020] 其中，V表示跳转路由变量，ins.address表示所述比较指令的地址，n表示偏移量，在ARM指令集模式下，所述偏移量n＝8，在Thumb指令集模式下，所述偏移量n＝4。

[0021] 可选的，所述根据所述关键特征值，采用拆分符号执行的方式对所述混淆控制流关系进行恢复，得到正常控制流关系，包括：

[0022] 所述多个程序块包括序幕程序块、调度程序块、相关程序块与返回程序块，为所述序幕程序块、所述相关程序块与所述返回程序块创建乱序块队列，并将所述序幕程序块设置在队首；

[0023] 从所述乱序块队列中的队首程序块开始，根据所述关键特征值采用所述拆分符号执行的方式确定所述乱序块序列中每个所述程序块的后继程序块，对所述乱序块队列中的多个所述程序块的顺序进行调整，得到正序块队列；

[0024] 根据所述正序块队列确定所述正常控制流关系。

[0025] 可选的，所述根据所述关键特征值采用所述拆分符号执行的方式确定所述乱序块序列中每个所述程序块的后继程序块，对所述乱序块队列中的多个所述程序块的顺序进行调整，包括：

[0026] A:设置执行指针指向所述乱序块队列的第一位，设置交换指针指向所述乱序块队列的第二位；

[0027] B:若所述执行指针当前所指向程序块位于队尾，结束顺序调整；

[0028] 反之，对所述执行指针所指向的所述程序块的跳转类型进行判定，若为条件跳转，执行步骤C，若为无条件跳转，执行步骤G；

[0029] C:根据所述关键特征值，确定所述执行指针当前所指向程序块的True跳转分支所对应的True分支后继程序块；

[0030] D:将所述True分支后继程序块与所述交换指针当前所指向程序块的位置进行交换，并将所述交换指针后移一位；

[0031] E:根据所述关键特征值，确定所述执行指针当前所指向程序块的False跳转分支所对应的False分支后继程序块；

[0032] F:将所述False分支后继程序块与所述交换指针当前所指向程序块的位置进行交换，并将所述交换指针后移一位，执行步骤I；

[0033] G:根据所述关键特征值，确定所述执行指针当前所指向程序块的唯一跳转分支所对应的Only分支后继程序块；

[0034] H:将所述Only分支后继程序块与所述交换指针当前所指向程序块的位置进行交换，并将所述交换指针后移一位，执行步骤I。

[0035] I:将所述执行指针后移一位，返回步骤B。

[0036] 可选的，在对所述乱序块队列中的多个所述程序块的顺序进行调整的过程中，在将所述True分支后继程序块、所述False分支后继程序块或所述Only分支后继程序块与所述交换指针当前所指向程序块的位置进行交换之前，都将相应所述后继程序块的执行状态

进行保存；

[0037] 在所述执行指针后移一位之后，都对其所指向程序块的执行状态进行恢复。

[0038] 可选的，所述对所述正常控制流关系进行优化，包括：

[0039] 对于通过无条件跳转相连接的两个所述程序块，若父节点的所述程序块的度不大于1，则将这两个所述程序块合并。

[0040] 可选的，所述对所述正常控制流关系进行优化，包括：

[0041] 对于具有两个跳转分支的多个所述程序块，若这些所述程序块通过一个分支依次串联连接，并且这些所述程序块的另一个分支都指向同一个后继程序块，则将这些所述程序块间的跳转结构优化作为循环结构。

[0042] 基于上述目的，本发明还提供了一种目标应用的安全检测装置，包括：

[0043] 反编译模块，被配置为对目标应用进行反编译，生成中间表达；

[0044] 特征提取模块，被配置为提取组成所述中间表达的多个程序块之间的混淆控制流关系的关键特征值；

[0045] 恢复模块，被配置为根据所述关键特征值，采用拆分符号执行的方式对所述混淆控制流关系进行恢复，得到正常控制流关系；

[0046] 优化模块，被配置为对所述正常控制流关系进行优化，并根据优化后的所述正常控制流关系生成相应的可执行程序；

[0047] 检测模块，被配置为对所述可执行程序进行安全性能检测，输出安全检测结果。

[0048] 基于上述目的，本发明还提供了一种电子设备，包括存储器、处理器及存储在存储器上并可在处理器上运行的计算机程序，所述处理器执行所述程序时实现所述目标应用安全检测方法。

[0049] 从上面所述可以看出，本发明提供的一种目标应用安全检测方法，对目标应用反编译得到中间表达，利用提取得到的原始控制流的关键特征值通过静态特征切割原始控制流，能够克服基本块分裂带来的问题，之后以每个基本块为分析目标，动态的调整分析目标序列以最大化保护并还原上下文继承关系，动态调整之后再继续对指令进行优化，从而对目标应用进行准确反混淆，由此，能够实现对所述目标应用的有效安全检测。

附图说明

[0050] 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。

[0051] 图1为本发明实施例所提供的一种目标应用安全检测方法示意图；

[0052] 图2为控制流平坦化混淆方法原理示意图；

[0053] 图3为本发明实施例所提供的一种目标应用安全检测方法中提取关键特征值的方法示意图；

[0054] 图4为本发明实施例所提供的一种目标应用安全检测方法中反编译得到的所述中间表达示意图；

[0055] 图5为本发明实施例所提供的一种目标应用安全检测方法中对混淆控制流关系进

行恢复的方法示意图。

[0056] 图6为本发明实施例所提供的一种目标应用安全检测方法中对乱序块队列中程序块顺序进行调整的方法示意图；

[0057] 图7-1为本发明实施例所提供的一种目标应用安全检测方法中当执行指针所指为条件跳转程序块时对程序块顺序进行调整的方法示意图；

[0058] 图7-2为本发明实施例所提供的一种目标应用安全检测方法中当执行指针所指为非条件跳转程序块时对程序块顺序进行调整的方法示意图；

[0059] 图8为本发明实施例所提供的一种目标应用安全检测方法中对正常控制流关系进行优化的跳转结构示意图。

[0060] 图9为本发明实施例所提供的一种目标应用安全检测方法中对正常控制流关系进行优化的跳转结构示意图；

[0061] 图10为本发明实施例所提供的一种目标应用安全检测装置示意图；

[0062] 图11为本发明实施例所提供的一种目标应用安全检测电子设备示意图。

具体实施方式

[0063] 为使本发明的目的、技术方案和优点更加清楚明白，以下结合具体实施例，并参照附图，对本发明进一步详细说明。

[0064] 需要说明的是，本发明实施例中所有使用“第一”和“第二”的表述均是为了区分两个相同名称非相同的实体或者非相同的参量，可见“第一”“第二”仅为了表述的方便，不应理解为对本发明实施例的限定，后续实施例对此不再一一说明。

[0065] 在一方面，本发明提供了一种目标应用安全检测方法。

[0066] 如图1所示，本发明的一些可选实施例所提供的一种目标应用安全检测方法，包括：

[0067] S1：对目标应用进行反编译，生成中间表达；

[0068] S2：提取组成所述中间表达的多个程序块之间的混淆控制流关系的关键特征值；

[0069] S3：根据所述关键特征值，采用拆分符号执行的方式对所述混淆控制流关系进行恢复，得到正常控制流关系；

[0070] S4：对所述正常控制流关系进行优化，并根据优化后的所述正常控制流关系生成相应的可执行程序；

[0071] S5：对所述可执行程序进行安全性能检测，输出安全检测结果。

[0072] 如图2所示，为控制流平坦化混淆原理示意图，其基本思想是将原始代码程序中的所有易于识别的条件跳转以及循环结构等移除，并将这些条件跳转以及循环结构替换为一个大的Switch结构，相应的以此大Switch结构来控制引导代码的控制流执行流入一个个基本块。如图所示，控制流平坦化混淆方法将图中左半部分所示的原始代码程序中的简单跳转结构进行替换，得到右半部分所示的复杂控制流跳转结构，使得控制流关系难以辨识的同时，将原始代码程序中的完成的执行代码拆分成基本块，基本块之间的上下文集成关系也被打乱。前述的这些操作对控制流平坦化混淆后的软件的反混淆带来困难。

[0073] 所述目标应用安全检测方法，对目标应用反编译得到中间表达，利用提取得到的原始控制流的关键特征值通过静态特征切割原始控制流，能够克服基本块分裂带来的问

题,之后以每个基本块为分析目标,动态的调整分析目标序列以最大化保护并还原上下文继承关系,动态调整之后再继续对指令进行优化,从而对目标应用进行准确反混淆,由此,能够实现对所述目标应用的有效安全检测。

[0074] 如图3所示,在本发明的一些可选实施例所提供的一种目标应用安全检测方法中,所述提取组成所述中间表达的多个程序块之间的混淆控制流关系的关键特征值S2,包括:

[0075] S21:根据代码指令内容,从所述多个程序块中选取出调度程序块;

[0076] S22:根据所述调度程序块中的跳转指令确定所述多个程序块之间的Switch调度结构;

[0077] S23:根据所述调度程序块中的比较指令确定跳转路由变量;

[0078] 所述关键特征值包括所述Switch调度结构与所述跳转路由变量。

[0079] 如图4所示,为控制流平坦化混淆后的所述中间表达示意图。组成所述中间表达的多个程序块包括序幕程序块Prologue、调度程序块Dispatcher、相关程序块Relevant与返回程序块Return。根据每个所述程序块的代码指令内容可以从中挑选出所述调度程序块Dispatcher,以条件跳转指令结束的所述程序块由三条指令组成,这样的程序块即为所述调度程序块Dispatcher。根据多个所述调度程序块Dispatcher中的跳转指令能够确定所述目标应用的大的Switch跳转结构;根据所述调度程序块Dispatcher中的比较指令能够确定所述跳转路由变量。

[0080] 所述目标应用安全检测方法中,在对所述目标应用反编译得到中间表达之后,通过每个程序块的代码指令确定所述关键特征值,从而实现对所述目标应用的原始控制流进行静态切割,将多个所述程序块独立,便于之后多个所述程序块进行动态调整,能够克服基本块拆分所带来的问题。

[0081] 在本发明的一些可选实施例所提供的一种目标应用安全检测方法中,所述根据所述调度程序块中的比较指令确定跳转路由变量S23,包括:

[0082] V=立即数+ins.address+n

[0083] 其中,V表示跳转路由变量,ins.address表示所述比较指令的地址,n表示偏移量,在ARM指令集模式下,所述偏移量n=8,在Thumb指令集模式下,所述偏移量n=4。图4中所示,每个所述程序块的代码指令为汇编指令,采用ARM指令集中指令表示,此时所述偏移量n=8,所述程序块的代码指令也有可能采用Thumb指令集中指令表示,若用Thumb指令集中指令表示,所述偏移量n=4。

[0084] 如图5所示,在本发明的一些可选实施例所提供的一种目标应用安全检测方法中,所述根据所述关键特征值,采用拆分符号执行的方式对所述混淆控制流关系进行恢复,得到正常控制流关系S3,包括:

[0085] S31:所述多个程序块包括序幕程序块Prologue、调度程序块Dispatcher、相关程序块Relevant与返回程序块Return,为所述序幕程序块Prologue、所述相关程序块Relevant与所述返回程序块Return创建乱序块队列,并将所述序幕程序块Prologue设置在队首;

[0086] 所述序幕程序块Prologue是程序运行时第一个执行的块,因此也将所述序幕程序块Prologue设置在队首,作为第一个分析的块。

[0087] S32:从所述乱序块队列中的队首程序块开始,根据所述关键特征值采用所述拆分

符号执行的方式确定所述乱序块序列中每个所述程序块的后继程序块，对所述乱序块队列中的多个所述程序块的顺序进行调整，得到正序块队列；

[0088] S33：根据所述正序块队列确定所述正常控制流关系。

[0089] 所述目标应用安全检测方法中，为除调度程序块Dispatcher之外的其他程序块创建队列，此时所创建的队列中程序块之间的顺序是乱序的，即所创建的是乱序块队列，之后利用所述关键特征值，对所述乱序块队列中的程序块的顺序进行调整，最后得到正序块队列，采用这样的动态调整方式，恢复正常控制流关系的同时还最大化地保护并还原上下文继承关系，从而能够保证最后反混淆结果的准确性。

[0090] 如图6所示，在本发明的一些可选实施例所提供的一种目标应用安全检测方法中，所述根据所述关键特征值采用所述拆分符号执行的方式确定所述乱序块序列中每个所述程序块的后继程序块，对所述乱序块队列中的多个所述程序块的顺序进行调整S32，包括：

[0091] A：设置执行指针指向所述乱序块队列的第一位，设置交换指针指向所述乱序块队列的第二位；

[0092] B：若所述执行指针当前所指向程序块位于队尾，结束顺序调整；

[0093] 反之，对所述执行指针所指向的所述程序块的跳转类型进行判定，若为条件跳转，执行步骤C，若为无条件跳转，执行步骤G；

[0094] C：根据所述关键特征值，确定所述执行指针当前所指向程序块的True跳转分支所对应的True分支后继程序块；

[0095] D：将所述True分支后继程序块与所述交换指针当前所指向程序块的位置进行交换，并将所述交换指针后移一位；

[0096] E：根据所述关键特征值，确定所述执行指针当前所指向程序块的False跳转分支所对应的False分支后继程序块；

[0097] F：将所述False分支后继程序块与所述交换指针当前所指向程序块的位置进行交换，并将所述交换指针后移一位，执行步骤I；

[0098] G：根据所述关键特征值，确定所述执行指针当前所指向程序块的唯一跳转分支所对应的Only分支后继程序块；

[0099] H：将所述Only分支后继程序块与所述交换指针当前所指向程序块的位置进行交换，并将所述交换指针后移一位，执行步骤I。

[0100] I：将所述执行指针后移一位，返回步骤B。

[0101] 所述目标应用安全检测方法中，在对所述乱序块队列中的多个所述程序块进行顺序调整前，首先对指针进行初始化，并设定好调整结束的条件，即当所述执行指针Execution Pointer指向队尾的程序块时，说明完成对队列中全部程序块的顺序调整工作。在调整时，始终以所述执行指针Execution Pointer所指向的所述程序块为目标，根据所指向的所述程序块的跳转类型不同采取不同的后续执行操作。

[0102] 如图7-1所示，当所述执行指针Execution Pointer所指向的所述程序块为条件跳转时，以所述序幕程序块Prologue为例，一般情况下所述序幕程序块Prologue总是有两个后续分支，利用符号拆分执行的方法，根据所述关键特征值能够分别确定两个后续分支路径。先确定所述序幕程序块Prologue的True分支后继程序块，如图所指为程序块Block5，之后将程序块Block5与所述交换指针Swap Pointer所指向的程序块Block1的位置进行交换，

交换之后程序块Block5位于所述乱序块队列中第二位，程序块Block1位于所述乱序块队列中第六位，并将所述交换指针Swap Pointer后移一位，指向程序块Block2；然后，确定所述序幕程序块Prologue的False分支后继程序块，如图所指为程序块Block3，之后将程序块Block3与所述交换指针Swap Pointer所指向的程序块Block2的位置进行交换，交换之后程序块Block3位于所述乱序块队列中第三位，程序块Block2位于所述乱序块队列中第四位，并将所述交换指针Swap Pointer后移一位，指向程序块Block3；两个分支的后继块都确定并交换完成之后，所述执行指针Execution Pointer后移一位，指向程序块Block5。

[0103] 如图7-2所示，当所述执行指针Execution Pointer所指向的所述程序块为无条件跳转时，以程序块Block5为例，只有唯一的跳转分支，利用符号拆分执行的方法，根据所述关键特征值能够确定唯一跳转分支路径。确定程序块Block5的Only分支后继程序块，如图7-2所指为程序块Block1，之后将程序块Block1与所述交换指针Swap Pointer所指向的程序块Block2的位置进行交换，交换之后程序块Block1位于所述乱序块队列中第四位，并将所述交换指针Swap Pointer后移一位，指向程序块Block4，之后将所述执行指针Execution Pointer后移一位，指向程序块Block3。

[0104] 所述目标应用安全检测方法中，利用所述关键特征值，对所述乱序块队列中的程序块的顺序进行调整，最后得到正序块队列，采用这样的动态调整方式，恢复正常控制流关系的同时还最大化地保护并还原上下文继承关系，从而能够保证最后反混淆结果的准确性。

[0105] 在本发明的一些可选实施例所提供的一种目标应用安全检测方法中，在对所述乱序块队列中的多个所述程序块的顺序进行调整的过程中，在将所述True分支后继程序块、所述False分支后继程序块或所述Only分支后继程序块与所述交换指针当前所指向程序块的位置进行交换之前，都将相应所述后继程序块的执行状态进行保存；

[0106] 在所述执行指针后移一位之后，都对其所指向程序块的执行状态进行恢复。

[0107] 所述目标应用安全检测方法中，在对所述乱序块队列中程序块进行位置交换前都保存其执行状态，在以所述程序块为目标确定后继程序块时都恢复期执行状态，采用这样的方式能够保证对所述目标应用进行反混淆的恢复强度高，使得反混淆之后所得到的可执行程序更加准确，更贴近于目标应用混淆处理前的程序。

[0108] 如图8所示，在本发明的一些可选实施例所提供的一种目标应用安全检测方法中，所述对所述正常控制流关系进行优化S4，包括：

[0109] 对于通过无条件跳转相连接的两个所述程序块，若父节点的所述程序块的度不大于1，则将这两个所述程序块合并。

[0110] 如图8所示，表示节点间跳转连接关系，其中节点3、节点4与节点5所代表的程序块都通过无条件跳转连接。对于节点3与节点4，节点3的父节点1的度为2，对于节点4与节点5，节点4的父节点3的度为1，于是可以将节点4与节点5所代表的程序块进行合并。

[0111] 所述目标应用安全检测方法中，针对已经恢复得到的所述正常控制流关系，其中仍有可能存在一些较为冗余的跳转结构，对于上述的跳转结构将相连接两个节点程序块合并，能够精简并优化最终得到的可执行程序代码，便于之后对所述可执行程序进行安全性能检测。

[0112] 如图9所示，在本发明的一些可选实施例所提供的一种目标应用安全检测方法中，

所述对所述正常控制流关系进行优化S4，包括：

[0113] 对于具有两个跳转分支的多个所述程序块，若这些所述程序块通过一个分支依次串联连接，并且这些所述程序块的另一个分支都指向同一个后继程序块，则将这些所述程序块间的跳转结构优化为循环结构。

[0114] 如图9所示，表示节点间跳转连接关系。其中节点1、节点2、节点3与节点4所代表的程序块都具有两个跳转分支，节点1到节点4都通过一个分支依次串联连接，且另一个分支都指向节点6。于是可以将节点1至节点4所代表的程序块间的跳转结构优化为循环结构。

[0115] 所述目标应用安全检测方法中，针对已经恢复得到的所述正常控制流关系，其中仍有可能存在一些较为冗余的跳转结构，对于上述的跳转结构将串联连接的几个节点程序块优化为循环结构，能够精简并优化最终得到的可执行程序代码，便于之后对所述可执行程序进行安全性能检测。

[0116] 在另一方面，本发明还提供了一种目标应用安全检测装置。

[0117] 如图1所示，本发明的一些可选实施例所提供的一种目标应用的安全检测装置，包括：

[0118] 反编译模块1，被配置为对目标应用进行反编译，生成中间表达；

[0119] 特征提取模块2，被配置为提取组成所述中间表达的多个程序块之间的混淆控制流关系的关键特征值；

[0120] 恢复模块3，被配置为根据所述关键特征值，采用拆分符号执行的方式对所述混淆控制流关系进行恢复，得到正常控制流关系；

[0121] 优化模块4，被配置为对所述正常控制流关系进行优化，并根据优化后的所述正常控制流关系生成相应的可执行程序；

[0122] 检测模块5，被配置为对所述可执行程序进行安全性能检测，输出安全检测结果。

[0123] 在另一方面，本发明还提供了一种执行所述目标应用安全检测方法的电子设备。

[0124] 如图11所示，所述电子设备包括：

[0125] 一个或多个处理器601以及存储器602，图6中以一个处理器601为例。

[0126] 所述执行所述目标应用安全检测方法的电子设备还可以包括：输入装置603和输出装置603。

[0127] 处理器601、存储器602、输入装置603和输出装置603可以通过总线或者其他方式连接，图7中以通过总线连接为例。

[0128] 存储器602作为一种非易失性计算机可读存储介质，可用于存储非易失性软件程序、非易失性计算机可执行程序以及模块，如本申请实施例中的所述目标应用安全检测方法对应的程序指令/模块。处理器601通过运行存储在存储器602中的非易失性软件程序、指令以及模块，从而执行服务器的各种功能应用以及数据处理，即实现上述方法实施例的目标应用安全检测方法。

[0129] 存储器602可以包括存储程序区和存储数据区，其中，存储程序区可存储操作系统、至少一个功能所需要的应用程序；存储数据区可存储根据执行所述目标应用安全检测方法的装置的使用所创建的数据等。此外，存储器602可以包括高速随机存取存储器，还可以包括非易失性存储器，例如至少一个磁盘存储器件、闪存器件、或其他非易失性固态存储器件。在一些实施例中，存储器602可选包括相对于处理器601远程设置的存储器，这些远程

存储器可以通过网络连接至会员用户行为监控装置。上述网络的实例包括但不限于互联网、企业内部网、局域网、移动通信网及其组合。

[0130] 输入装置603可接收输入的数字或字符信息，以及产生与执行所述目标应用安全检测方法装置的用户设置以及功能控制有关的键信号输入。输出装置603可包括显示屏等显示设备。

[0131] 所述一个或者多个模块存储在所述存储器602中，当被所述一个或者多个处理器601执行时，执行上述任意方法实施例中的目标应用安全检测方法。所述执行所述目标应用安全检测方法的装置的实施例，其技术效果与前述任意方法实施例相同或者类似。

[0132] 上述实施例的装置用于实现前述实施例中相应的方法，并且具有相应的方法实施例的有益效果，在此不再赘述。

[0133] 所属领域的普通技术人员应当理解：以上任何实施例的讨论仅为示例性的，并非旨在暗示本公开的范围(包括权利要求)被限于这些例子；在本发明的思路下，以上实施例或者不同实施例中的技术特征之间也可以进行组合，步骤可以以任意顺序实现，并存在如上所述的本发明的不同方面的许多其它变化，为了简明它们没有在细节中提供。

[0134] 另外，为简化说明和讨论，并且为了不会使本发明难以理解，在所提供的附图中可以示出或可以不示出与集成电路(IC)芯片和其它部件的公知的电源/接地连接。此外，可以以框图的形式示出装置，以便避免使本发明难以理解，并且这也考虑了以下事实，即关于这些框图装置的实施方式的细节是高度取决于将要实施本发明的平台的(即，这些细节应当完全处于本领域技术人员的理解范围内)。在阐述了具体细节(例如，电路)以描述本发明的示例性实施例的情况下，对本领域技术人员来说显而易见的是，可以在没有这些具体细节的情况下或者这些具体细节有变化的情况下实施本发明。因此，这些描述应被认为是说明性的而不是限制性的。

[0135] 尽管已经结合了本发明的具体实施例对本发明进行了描述，但是根据前面的描述，这些实施例的很多替换、修改和变型对本领域普通技术人员来说将是显而易见的。例如，其它存储器架构(例如，动态RAM(DRAM))可以使用所讨论的实施例。

[0136] 本发明的实施例旨在涵盖落入所附权利要求的宽泛范围之内的所有这样的替换、修改和变型。因此，凡在本发明的精神和原则之内，所做的任何省略、修改、等同替换、改进等，均应包含在本发明的保护范围之内。

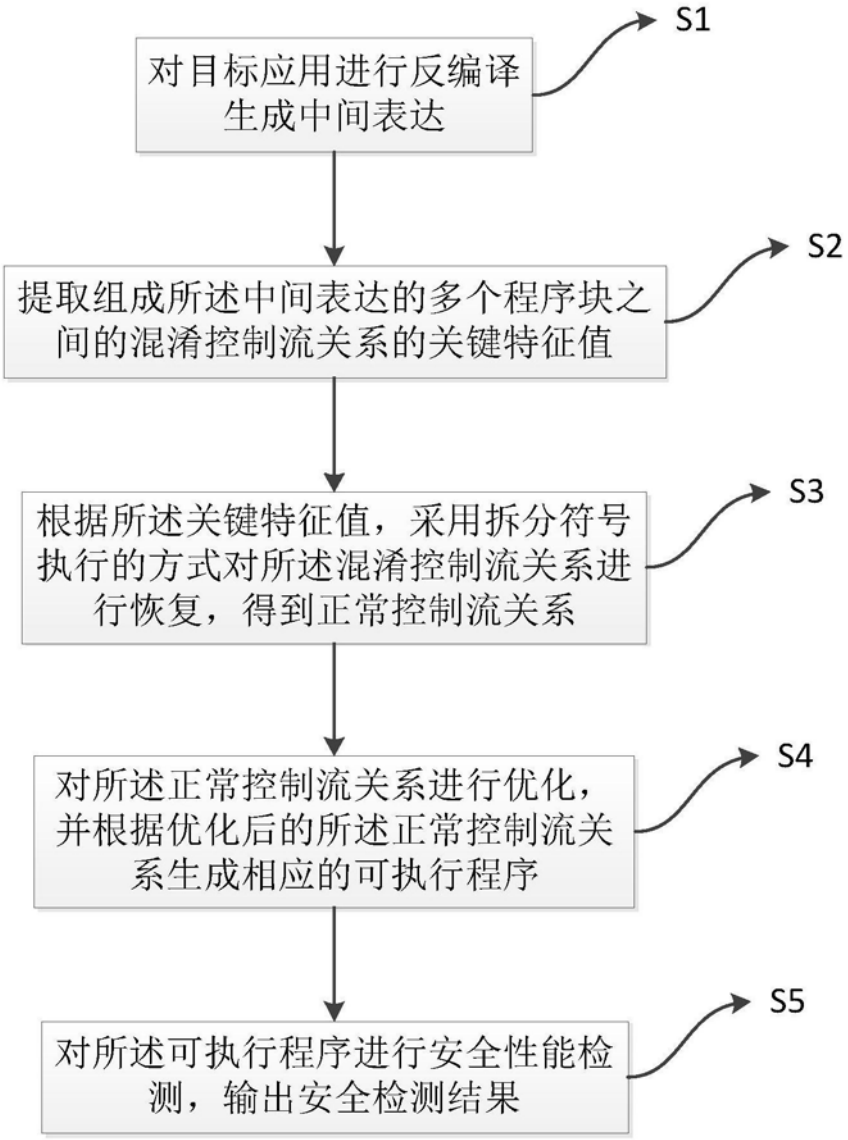

S1
对目标应用进行反编译生成中间表达
S2
提取组成所述中间表达的多个程序块之间的混淆控制流关系的关键特征值
S3
根据所述关键特征值，采用拆分符号执行的方式对所述混淆控制流关系进行恢复，得到正常控制流关系
S4
对所述正常控制流关系进行优化，并根据优化后的所述正常控制流关系生成相应的可执行程序
S5
对所述可执行程序进行安全性能检测，输出安全检测结果

图1

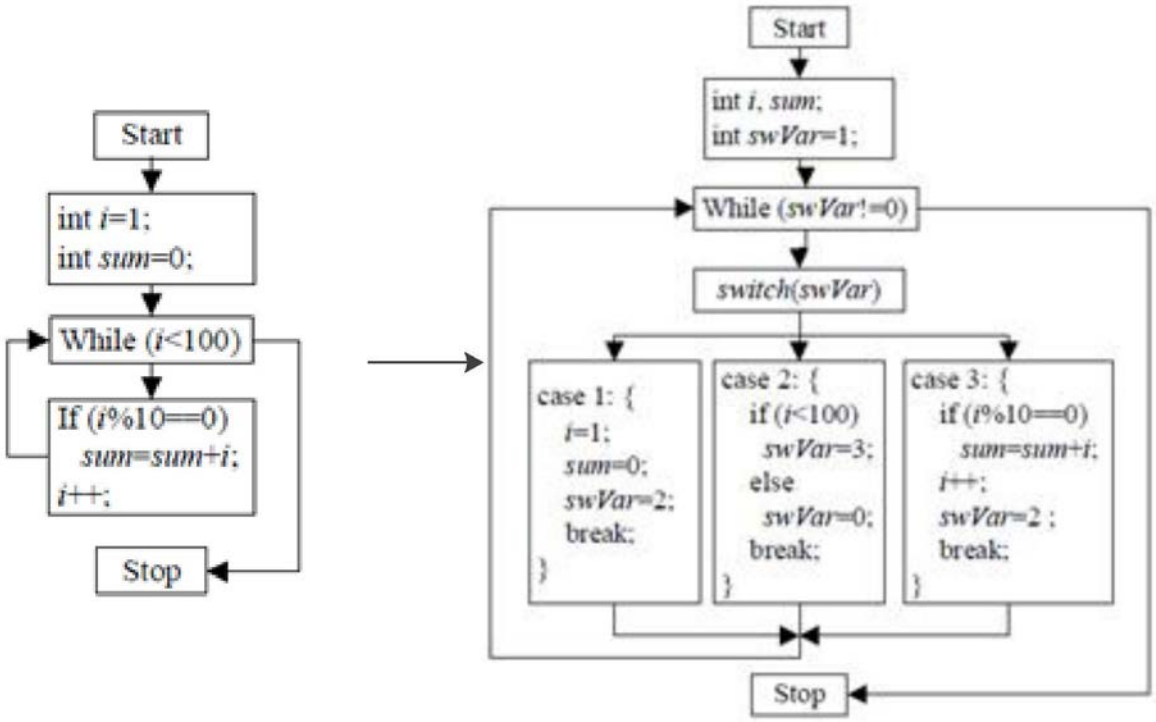

Start
int i=1;
int sum=0;
While (i<100)
If (i%10==0)
sum=sum+i;
i++;
Stop
Start
int i, sum;
int swVar=1;
While (swVar!=0)
switch(swVar)
case 1: {
i=1;
sum=0;
swVar=2;
break;
}
case 2: {
if (i<100)
swVar=3;
else
swVar=0;
break;
}
case 3: {
if (i%10==0)
sum=sum+i;
i++;
swVar=2 ;
break;
}
Stop

图2

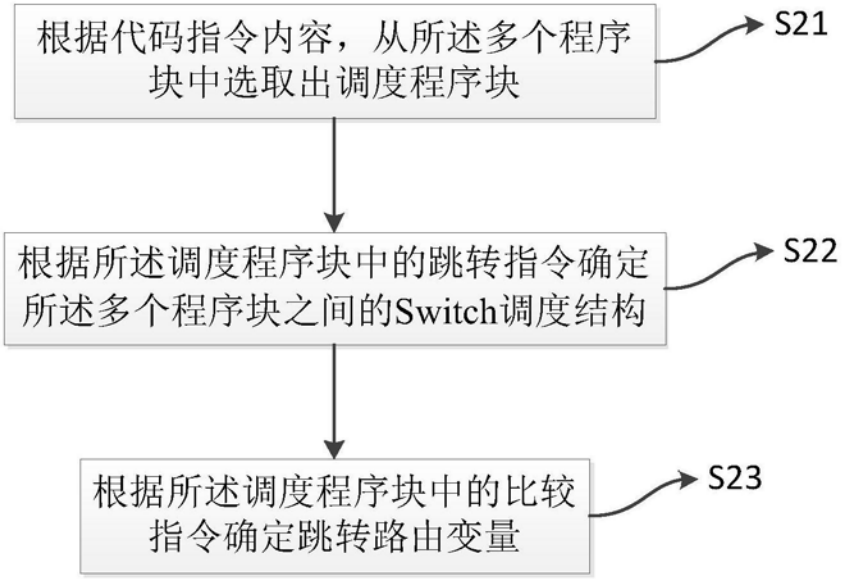

根据代码指令内容，从所述多个程序块中选取出调度程序块
S21
根据所述调度程序块中的跳转指令确定所述多个程序块之间的Switch调度结构
S22
根据所述调度程序块中的比较指令确定跳转路由变量
S23

图3

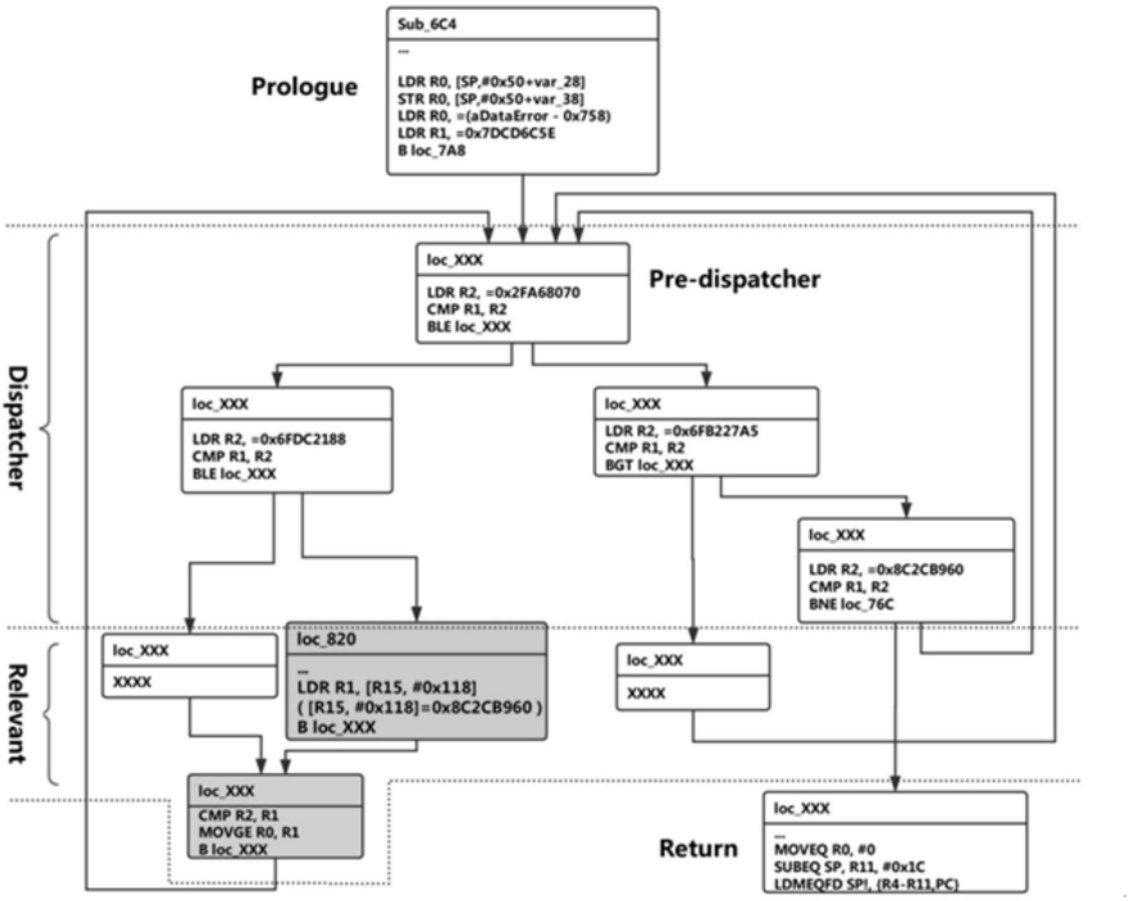

Prologue
Sub_6C4
...
LDR R0, [SP,#0x50+var_28]
STR R0, [SP,#0x50+var_38]
LDR R0, =(aDataError - 0x758)
LDR R1, =0x7DCD6C5E
B loc_7A8
Dispatcher
Relevant
loc_XXX
LDR R2, =0x2FA68070
CMP R1, R2
BLE loc_XXX
Pre-dispatcher
loc_XXX
LDR R2, =0x6FDC2188
CMP R1, R2
BLE loc_XXX
loc_XXX
LDR R2, =0x6FB227A5
CMP R1, R2
BGT loc_XXX
loc_XXX
LDR R2, =0x8C2CB960
CMP R1, R2
BNE loc_76C
loc_XXX
XXXX
loc_820
...
LDR R1, [R15, #0x118]
([R15, #0x118]=0x8C2CB960)
B loc_XXX
loc_XXX
XXXX
loc_XXX
CMP R2, R1
MOVGE R0, R1
B loc_XXX
Return
loc_XXX
...
MOVEQ R0, #0
SUBEQ SP, R11, #0x1C
LDMEQFD SP!, {R4-R11,PC}

图4

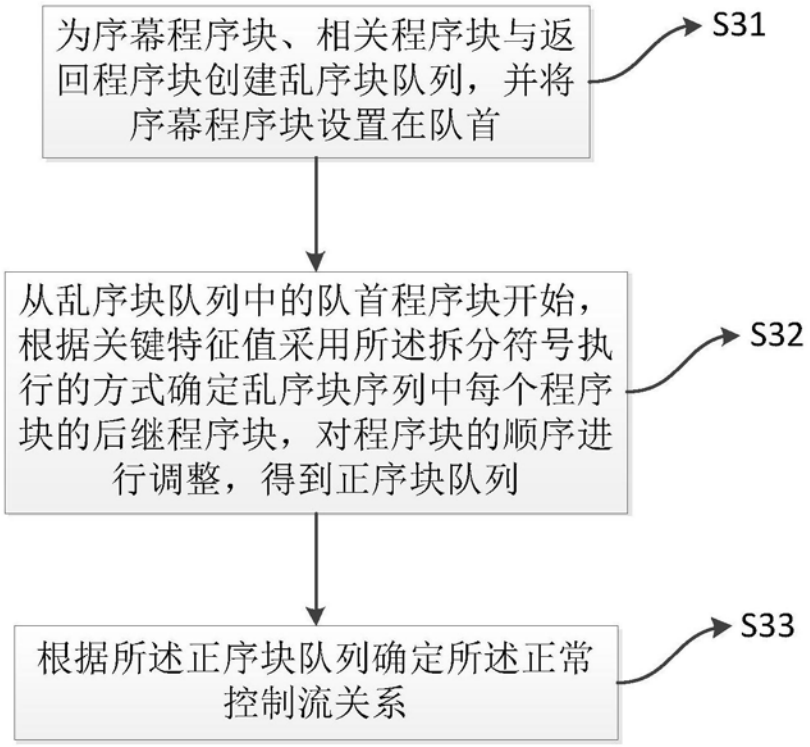

为序幕程序块、相关程序块与返回程序块创建乱序块队列，并将序幕程序块设置在队首
S31
从乱序块队列中的队首程序块开始，根据关键特征值采用所述拆分符号执行的方式确定乱序块序列中每个程序块的后继程序块，对程序块的顺序进行调整，得到正序块队列
S32
根据所述正序块队列确定所述正常控制流关系
S33

图5

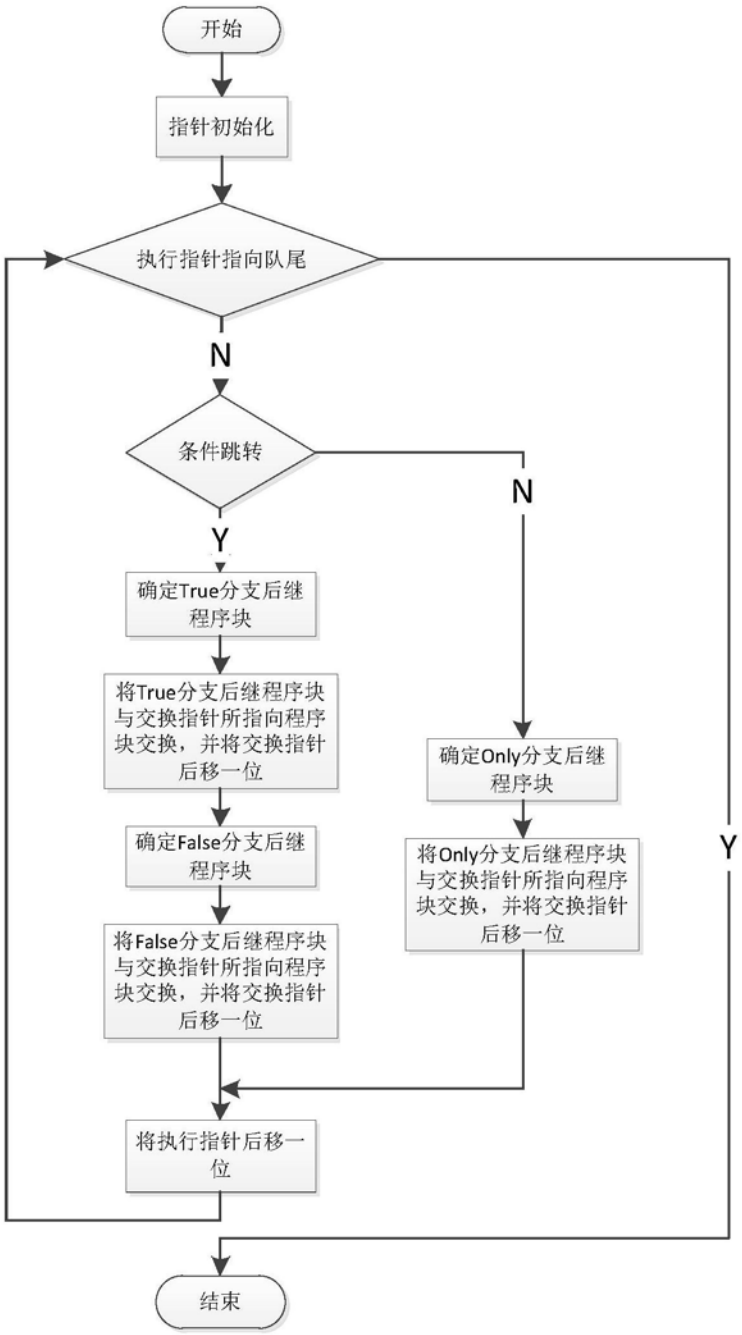

开始
指针初始化
执行指针指向队尾
N
条件跳转
N
Y
确定True分支后继程序块
将True分支后继程序块与交换指针所指向程序块交换，并将交换指针后移一位
确定False分支后继程序块
将False分支后继程序块与交换指针所指向程序块交换，并将交换指针后移一位
确定Only分支后继程序块
将Only分支后继程序块与交换指针所指向程序块交换，并将交换指针后移一位
Y
将执行指针后移一位
结束

图6

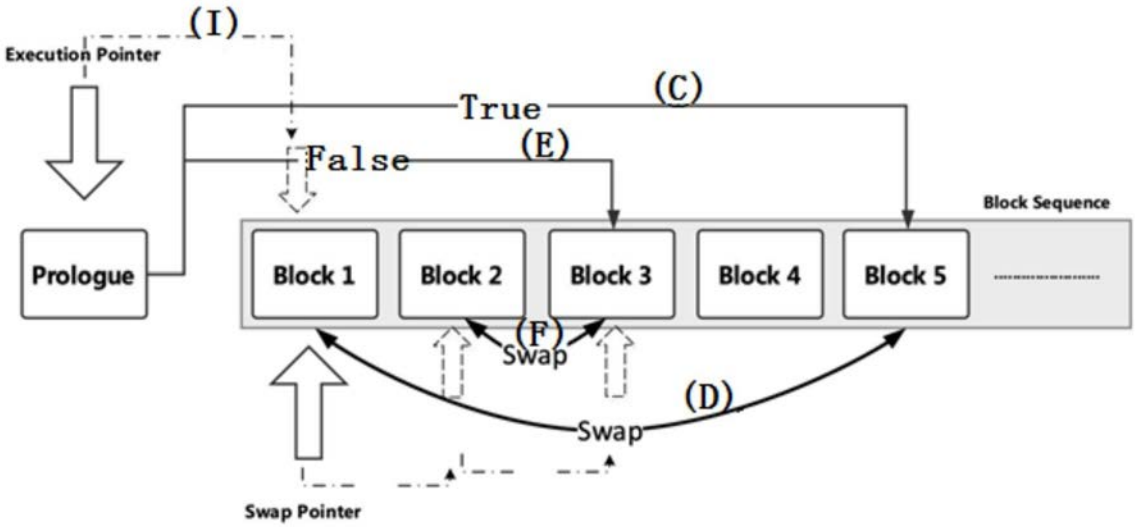

(I)
Execution Pointer
True
(C)
False
(E)
Block Sequence
Prologue
Block 1
Block 2
Block 3
Block 4
Block 5
(F)
Swap
(D)
Swap
Swap Pointer

图7-1

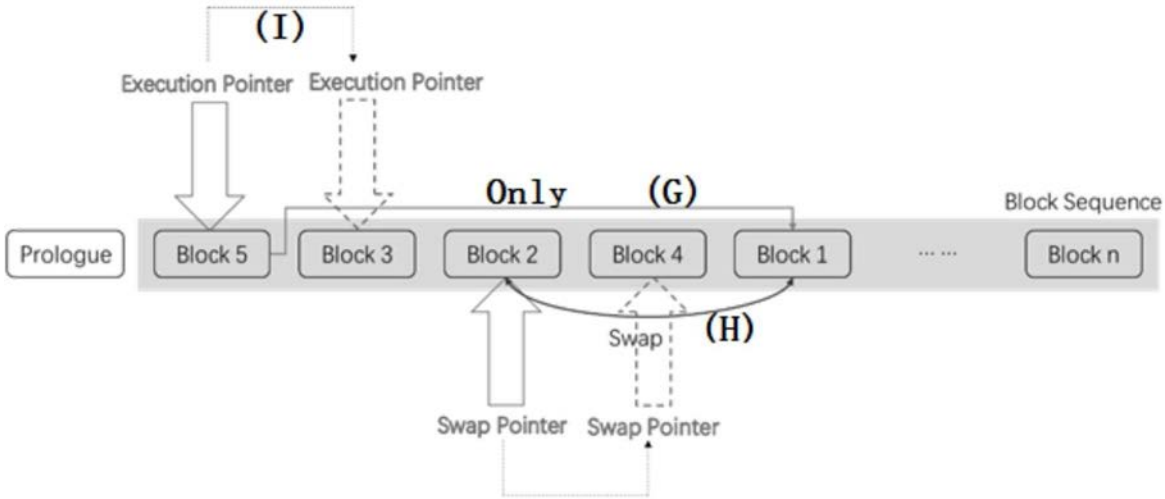

(I)
Execution Pointer
Execution Pointer
Only
(G)
Block Sequence
Prologue
Block 5
Block 3
Block 2
Block 4
Block 1
Block n
Swap
(H)
Swap Pointer
Swap Pointer

图7-2

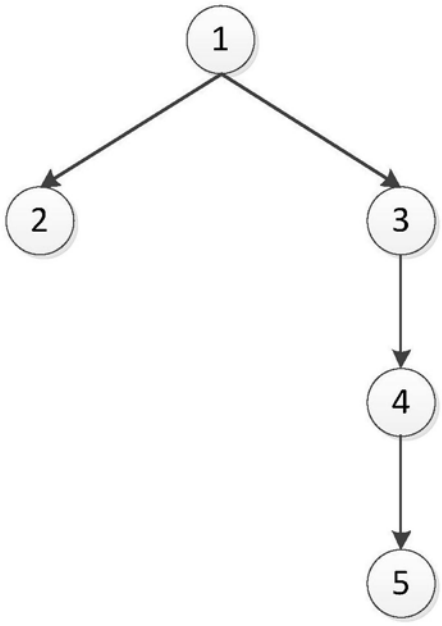


图8

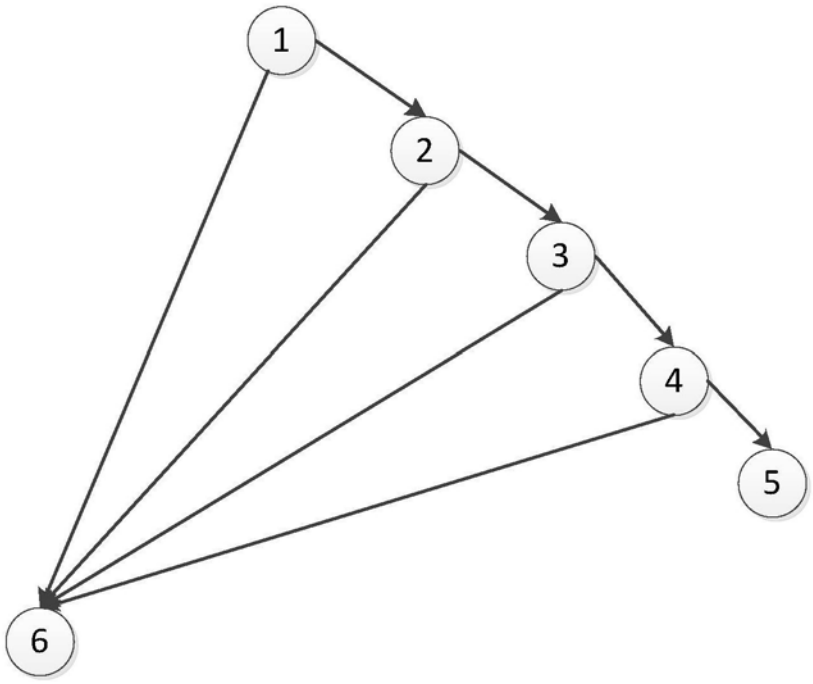


图9

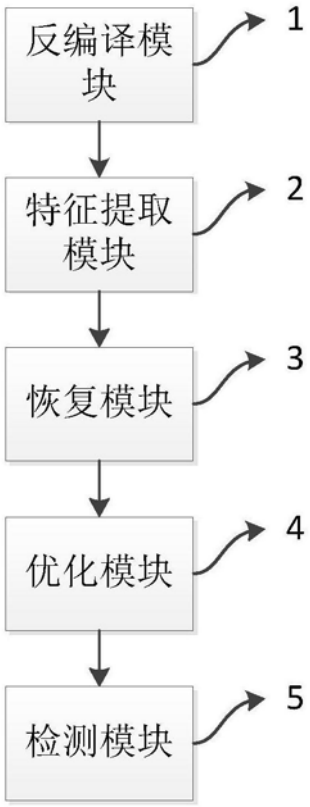

反编译模块
1
特征提取模块
2
恢复模块
3
优化模块
4
检测模块
5

图10

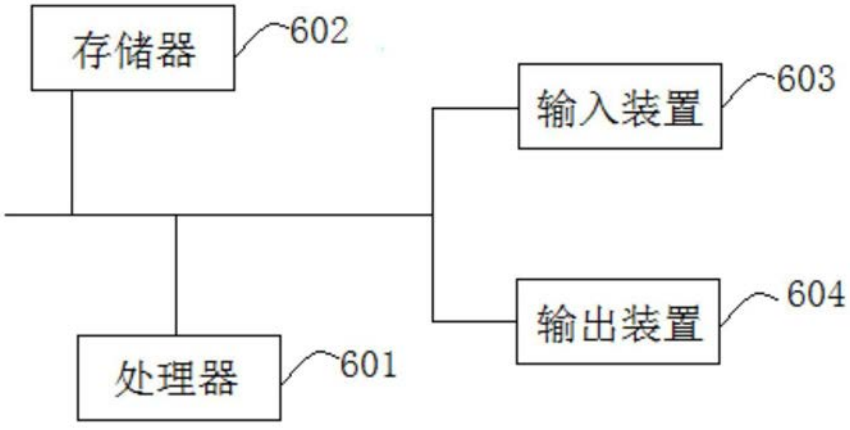

存储器
602
输入装置
603
输出装置
604
处理器
601

图11