

(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
2017 年 6 月 1 日 (01.06.2017)



(10) 国际公布号
WO 2017/088682 A1

- (51) 国际专利分类号:
G06F 21/53 (2013.01)
- (21) 国际申请号: PCT/CN2016/105822
- (22) 国际申请日: 2016 年 11 月 15 日 (15.11.2016)
- (25) 申请语言: 中文
- (26) 公布语言: 中文
- (30) 优先权:
201510827728.6 2015 年 11 月 24 日 (24.11.2015) CN
- (71) 申请人: 阿里巴巴集团控股有限公司 (ALIBABA GROUP HOLDING LIMITED) [—/CN]; 开曼群岛大开曼资本大厦一座四层 847 号邮箱, Grand Cayman (KY)。
- (72) 发明人: 李雪峰 (LI, Xuefeng); 中国浙江省杭州市余杭区文一西路 969 号 3 号楼 5 楼阿里巴巴集团法务部, Zhejiang 311121 (CN)。 张雁冰 (ZHANG, Yanbing); 中国浙江省杭州市余杭区文一西路 969 号 3 号楼 5 楼阿里巴巴集团法务部, Zhejiang 311121 (CN)。
- (74) 代理人: 北京三友知识产权代理有限公司 (BEIJING SANYOU INTELLECTUAL PROPERTY

AGENCY LTD.); 中国北京市金融街 35 号国际企业大厦 A 座 16 层, Beijing 100033 (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, JP, KE, KG, 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, 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))。

(54) Title: METHOD AND DEVICE FOR IMPROVING SECURITY OF JAVA SANDBOX

(54) 发明名称: 用于提高 Java 沙箱安全性的方法及装置

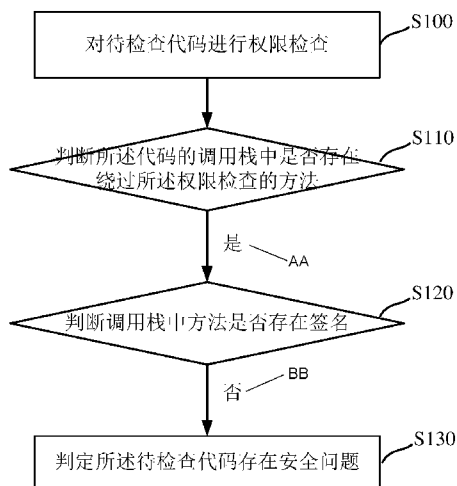


图 1

S100 Checking the authority of a code to be checked
S110 Judging whether a method for bypassing the authority check exists in a call stack of the code
S120 Judging whether there is a signature for the method in the call stack
S130 Determining that security problems occur in the code to be checked
AA Yes
BB No

(57) Abstract: A method and device for improving the security of a Java sandbox. The method for improving the security of a Java sandbox comprises: checking the authority of a code to be checked (S100); judging whether a method for bypassing the authority check exists in a call stack of the code (S110); if the method for bypassing the authority check exists, judging whether there is a signature for the method in the call stack (S120); and if there is no signature, determining that the code to be checked has a security problem (S130). The method may improve the security of a Java sandbox.

(57) 摘要: 一种用于提高 Java 沙箱安全性的方法及装置, 所述用于提高 Java 沙箱安全性的方法包括: 对待检查代码进行权限检查 (S100); 判断所述代码的调用栈中是否存在绕过所述权限检查的方法 (S110); 若存在绕过所述权限检查的方法, 则判断所述调用栈中方法是否存在签名 (S120); 若不存在签名, 则判定所述待检查代码存在安全问题 (S130)。该方法可以提高 Java 沙箱安全性。



WO 2017/088682 A1

用于提高 Java 沙箱安全性的方法及装置

本申请要求 2015 年 11 月 24 日递交的申请号为 201510827728.6、发明名称为“用于提高 Java 沙箱安全性的方法及装置”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。

技术领域

本申请涉及计算机安全技术领域，特别涉及一种用于提高 Java 沙箱安全性的方法及装置。

背景技术

在云计算的场景中，用户提供 Java 语言编写的代码，由云计算平台执行所述代码。由于所述代码往往存在很多的安全问题（如黑客的攻击代码），一般会在 Java 沙箱（Java Sandbox）中执行。

Java 沙箱提供了一种按照策略文件限制代码权限的执行环境。Java 沙箱运行代码过程中，当调用到方法 checkPermission 时，Java 沙箱的安全管理器（Security Manager）会对所述代码进行权限检查。所述权限检查用于判断所述代码运行过程中调用栈中每一层的方法是否具有策略文件中的权限，所述权限可以包括是否允许写、是否允许读或者是否允许使用网络等；如果所述调用栈中每一层的方法都具有权限，则将所述代码视为是安全的，如果所述调用栈中任一层的方法不具有权限，则将所述代码视为不安全的。

现有技术中，所述云计算平台会提供平台的特权方法，由于所述特权方法具有系统域（system domain），而安全管理器不会对具有系统域的方法进行权限检查，所以所述特权方法可以绕过所述安全管理器权限检查的步骤。一般的，用户的代码是不允许直接调用特权方法的，用户的代码只能通过可信任的方法（拥有签名的方法）间接的调用特权方法。但是恶意用户通过反射、提权等手段可以使代码直接调用特权方法，当代码直接调用特权方法时，会使得该代码也具有系统域，从而能绕过安全管理器的权限检查。如此，恶意用户的代码就突破了 Java 沙箱的防护。

综上所述，现有技术中存在 Java 沙箱安全性不高的问题。

发明内容

本申请实施例的目的是提供一种用于提高 Java 沙箱安全性的方法及装置，用以解决现有技术中 Java 沙箱安全性不高的问题。

为解决上述技术问题，本申请实施例提供的用于提高 Java 沙箱安全性的方法及装置是

这样实现的：

一种用于提高 Java 沙箱安全性的方法，包括：

对待检查代码进行权限检查；

判断所述代码的调用栈中是否存在绕过所述权限检查的方法；

5 若存在绕过所述权限检查的方法，则判断所述调用栈中方法是否存在签名；

若不存在签名，则判定所述待检查代码存在安全问题。

一种用于提高 Java 沙箱安全性的装置，包括：

检查模块，用于对待检查代码进行权限检查；

第一判断模块，用于判断所述代码的调用栈中是否存在绕过所述权限检查的方法；

10 第二判断模块，用于在所述代码的调用栈中存在绕过所述权限检查的方法时，判断所述调用栈中方法是否存在签名；

确定模块，用于在所述调用栈中方法不存在签名时，判定所述待检查代码存在安全问题。

15 由以上本申请实施例提供的技术方案可见，本申请实施例提供了一种用于提高 Java 沙箱安全性的方法及装置，可以克服待检查代码直接调用特权方法后绕过权限检查从而造成 Java 沙箱安全性低的问题，通过对存在绕过权限检查的方法所在调用栈中签名进行检查，若所述调用栈中方法不存在签名，则判定待检查代码存在安全问题。如此，提高了 Java 沙箱的安全性。

20 附图说明

为了更清楚地说明本申请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申请中记载的一些实施例，对于本领域普通技术人员来讲，在不付出创造性劳动性的前提下，还可以根据这些附图获得其他的附图。

25 图 1 为本申请一实施例中提供的用于提高 Java 沙箱安全性的方法的流程图；

图 2 为本申请一实施例中提供的用于提高 Java 沙箱安全性的方法的流程图；

图 3 为本申请一实施例中提供的用于提高 Java 沙箱安全性的方法的流程图；

图 4 为本申请一实施例中提供的用于提高 Java 沙箱安全性的方法的流程图；

图 5 为本申请一实施例中提供的用于提高 Java 沙箱安全性的方法的流程图；

30 图 6 为本申请一实施例中提供的用于提高 Java 沙箱安全性的装置的模块示意图。

具体实施方式

为了使本技术领域的人员更好地理解本申请中的技术方案，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清楚、完整地描述，显然，所描述的实施例仅仅是本申请一部分实施例，而不是全部的实施例。基于本申请中的实施例，本领域普通技术人员在没有付出创造性劳动前提下所获得的所有其他实施例，都应当属于本申请保护的

5 的范围。

图 1 为本申请一实施例中提供的提高 Java 沙箱安全性的方法的流程图。本实施例中，所述方法包括如下步骤：

S100：对待检查代码进行权限检查；

在本申请实施例中，待检查代码是用户通过 Java 语言编写的代码，该代码中包含有用

10 户方法。

如上所述，Java 沙箱运行待检查代码过程中，当调用到方法 `checkPermission` 时，安全管理器会对所述待检查代码进行权限检查。所述权限检查用于判断所述代码运行过程中调用栈中每一层的方法是否具有策略文件中的权限，所述权限可以包括是否允许写、是否允许读或者是否允许使用网络等；如果所述调用栈中每一层的方法都具有权限，则将所述

15 代码视为是安全的，如果所述调用栈中任一层的方法不具有权限，则将所述代码视为不安全的。

以下例子 1 为云计算场景中用户使用 `main()` 方法打开文件名为 `filename` 操作的调用栈：

com.alibaba.apsara.sandking.SecurityManage.checkPermission(FILE_READ_ACTION)	第 0 层
com.alibaba.apsara.sandking.SecurityManage.checkRead(filename)	第 1 层
java.io.FileInputStream.FileInputStream(filename)	第 2 层
com.readwrapper(filename)	第 3 层
usercode.main()	第 4 层

20 N)

如上所示，该调用栈的调用过程为：首先第 4 层方法 `main()` 调用第 3 层方法 `readwrapper(filename)`，再调用第 2 层方法 `FileInputStream(filename)`，再调用第 1 层方法 `checkRead(filename)`，然后调用第 0 层方法 `checkPermission(FILE_READ_ACTION)`。该调用栈中，第 4 层方法 `main()` 属于用户方法，第 3 层方法 `readwrapper()` 属于云计算平台的公有方法（具有签名“alibaba”的可信任方法），第 2 层方法 `FileInputStream()` 属于

25 Java 系统方法，第 1 层方法 `checkRead()` 属于 Java 沙箱的方法，第 0 层方法 `checkPermission()` 属于 Java 沙箱的方法。在第 0 层 `checkPermission(FILE_READ_ACTION)`

30

触发后，进入权限检查。该权限检查会从策略文件中获取是否允许读的权限，从而对调用栈中每一层的方法进行检查。由于第 0 层和第 1 层的方法为 Java 沙箱的方法，具有读的权限，第 2 层的方法为 Java 系统方法，也具有读的权限，第 3 层的方法为公有方法，也具有读的权限，第 4 层的方法为用户方法，由于打开文件需要获取读的权限，若所述策略文件中允许读，则第 4 层的方法可以获取读的权限，即第 4 层的方法具有读的权限。这样，因为调用栈中每一层的方法都具有权限，所以待检查代码是安全的，用户可以打开云计算平台中文件名为 filename 的文件。反之，若所述策略文件不允许读，则第 4 层的方法不能获取读的权限，即第 4 层的方法不具有读的权限。这样，因为调用栈中存在不具有权限的方法，所以待检查代码是不安全的，用户不能打开云计算平台中文件名为 filename 的文件。

以下例子 2 为在云计算平台提供了特权方法 doPrivileged() 时，该云计算场景中用户使用 main() 方法打开文件名为 filename 操作的调用栈：

com.alibaba.apsara.sandking.SecurityManage.checkPermission(FILE_READ_ACTION)	第 0 层
com.alibaba.apsara.sandking.SecurityManage.checkRead(filename)	第 1 层
java.io.FileInputStream.FileInputStream(filename)	第 2 层
com.readwrapper(filename)	第 3 层
java.security.AccessController.doPrivileged()	第 4 层
com.alibaba.privilegedwrapper()	第 5 层
usercode.main()	第 6 层

该调用栈与上一例子 1 中的调用栈不同之处在于，在方法 main() 调用方法 readwrapper(filename) 之间，多了第 5 层方法 privilegedwrapper() 和第 4 层方法 doPrivileged()，即第 6 层方法 main() 调用第 5 层方法 privilegedwrapper()，再调用第 4 层方法 doPrivileged()，再调用第 3 层方法 readwrapper(filename)。该调用栈中，第 6 层方法 main() 属于用户方法，第 5 层方法 privilegedwrapper() 和第 3 层方法 readwrapper(filename) 属于云计算平台的公有方法，第 4 层方法 doPrivileged() 属于云计算平台的特权方法，第 2 层方法 FileInputStream(filename) 属于 Java 系统方法，第 1 层方法 checkRead() 属于 Java 沙箱的方法，第 0 层方法 checkPermission() 属于 Java 沙箱的方法。在第 0 层 checkPermission(FILE_READ_ACTION) 触发后，进入权限检查。该权限检查与上一例子中的权限检查不同之处在于，第 4 层的方法 doPrivileged() 为特权方法，可以绕过权限检查，第 5 层的方法 privilegedwrapper() 调用了第 4 层的特权方法，也可以绕过权限检查。由于第 5 层的方法 privilegedwrapper() 属于云计算平台的公有方法，本身就具有

读的权限，所以即使其绕过权限检查，并不影响对第 6 层的方法 `main()` 的检查。

以下例 3 为在云计算平台提供了特权方法 `doPrivileged()`，并且该云计算场景中用户使用 `main()` 方法直接调用该特权方法 `doPrivileged()`，打开文件名为 `filename` 操作的调用栈。

5	<code>com.alibaba.apsara.sandking.SecurityManage.checkPermission(FILE_READ_ACTION)</code>	第 0 层
	<code>com.alibaba.apsara.sandking.SecurityManage.checkRead(filename)</code>	第 1 层
	<code>java.io.FileInputStream.FileInputStream(filename)</code>	第 2 层
	<code>com.readwrapper(filename)</code>	第 3 层
10	<code>java.security.AccessController.doPrivileged()</code>	第 4 层
	<code>usercode.main()</code>	第 5 层

该调用栈与上一例子 2 中的调用栈不同之处在于，方法 `main()` 直接调用了特权方法 `doPrivileged()`，即第 5 层方法 `main()` 调用第 4 层方法 `doPrivileged()`。在权限检查时，由于第 4 层的方法 `doPrivileged()` 为特权方法可以绕过权限检查，第 5 层的方法 `main()` 调用了特权方法也可以绕过权限检查。如此，第 0 层和第 1 层的方法为 Java 沙箱的方法，具有读的权限，第 2 层的方法为 Java 系统方法，也具有读的权限，第 3 层的方法为公有方法，也具有读的权限，第 4 层和第 5 层绕过了权限检查，即使在策略文件不允许读的情况下，该权限检查始终是都是通过的，从而安全管理器得出待检查代码是安全的。这样，对于用户方法 `main` 即使不具有读的权限，也可以打开云计算平台中文件名为 `filename` 的文件。

20 S110：判断所述代码的调用栈中是否存在绕过所述权限检查的方法；若存在绕过所述权限检查的方法，则执行 S120。

安全管理器判断所述代码的调用栈中是否存在绕过所述权限检查的方法。

如上述例子 1 中所示，由于调用栈中不存在绕过所述权限检查的方法，所以安全管理器可以根据该权限检查的结果判断待检查的代码是否安全，即如果权限检查通过，则安全管理器判断待检查的代码是安全的；如果权限检查不通过，则安全管理器判断待检查的代码存在安全问题。

如上述例子 2 中所示，由于调用栈中存在绕过所述权限检查的方法（`doPrivileged()` 和 `privilegedwrapper()`），所以执行 S120 步骤。

如上述例子 3 中所示，由于调用栈中存在绕过所述权限检查的方法（`doPrivileged()` 和 `main()`），所以执行 S120 步骤。

S120：判断所述调用栈中方法是否存在签名；若不存在签名，则执行 S130 步骤。

沿用例子 2，第 6 层没有签名，第 5 层的签名为“alibaba”，第 4 层没有签名，第 3 层没有签名，第 2 层没有签名，第 1 层没有签名，第 0 层没有签名。由于所述调用栈中第 5 层的方法存在签名“alibaba”，所以安全管理器判断所述待检查的代码通过权限检查，即待检查的代码是安全的。

5 沿用例子 3，第 5 层没有签名，第 4 层没有签名，第 3 层没有签名，第 2 层没有签名，第 1 层没有签名，第 0 层没有签名。安全管理器判断所述调用栈中方法是否存在签名时，由于所述调用栈中不存在签名，所以执行 S130 步骤。

S130：判定所述待检查代码存在安全问题。

安全管理器判断所述待检查代码存在安全问题，将抛出安全异常，终止待检查代码的
10 调用，在例子 3 中，则用户无法打开云计算平台中文件名为 filename 的文件。

本实施例中，通过对存在绕过权限检查的方法所在调用栈中签名进行检查，若所述调用栈中方法不存在签名，则判定待检查代码存在安全问题。如此，就可以避免调用特权方法绕过权限检查，从而提高了 Java 沙箱的安全性。

在例子 3 中，如果第 3 层的方法 readwrapper() 也具有签名（在实际应用中云计算平台
15 提供的公有方法可能存在签名），那么安全管理器可以得出所述待检查的代码通过权限检查，即待检查的代码是安全的。然而例子 3 中存在第 5 层的用户方法 main() 直接调用第 4 层的特权方法 doPrivileged()，得出待检查的代码是安全的这显然是不合理的。

为了解决以上调用栈中公有方法存在签名时，Java 沙箱不能检查出用户方法直接调用特权方法的问题，如图 2 所示，在判断所述调用栈中方法是否存在签名之后，本方法还可
20 以包括如下步骤：

S121：若存在签名，则判断所述签名对应的方法的深度是否大于特权方法的深度；若不大于特权方法的深度，则执行 S130。

本实施例中，所述深度为方法在调用栈中的层数，例如一方法位于调用栈中的第 3 层，则该方法的深度即为 3。

25 在上述例子 2 中的基础上，如果第 3 层的方法 readwrapper() 也具有签名“alibaba”，则调用栈为：

com.alibaba.apsara.sandking.SecurityManage.checkPermission(FILE_READ_ACTION)	第 0 层
com.alibaba.apsara.sandking.SecurityManage.checkRead(filename)	第 1 层
30 java.io.FileInputStream.FileInputStream(filename)	第 2 层
com.alibaba.readwrapper(file name)	第 3 层

java.security.AccessController.doPrivileged()	第 4 层
com.alibaba.privilegedwrapper()	第 5 层
usercode.main()	第 6 层

该调用栈中，签名对应的方法 readwrapper() 的深度为 3；签名对应的方法 privilegedwrapper() 的深度为 5，而特权方法 doPrivileged() 的深度为 4。很明显的方法 privilegedwrapper() 的深度大于特权方法 doPrivileged() 的深度，所以安全管理器判定所述待检查的代码通过权限检查，即待检查的代码是安全的。

在上述例子 3 中的基础上，如果第 3 层的方法 readwrapper() 也具有签名 “alibaba”，则调用栈为：

com.alibaba.apsara.sandking.SecurityManage.checkPermission(FILE_READ_ACTION)	第 0 层
com.alibaba.apsara.sandking.SecurityManage.checkRead(filename)	第 1 层
java.io.FileInputStream.FileInputStream(filename)	第 2 层
com.alibaba.readwrapper(filename)	第 3 层
java.security.AccessController.doPrivileged()	第 4 层
usercode.main()	第 5 层

该调用栈中，签名对应的方法 readwrapper() 的深度为 3；而特权方法 doPrivileged() 的深度为 4。很明显的方法 privilegedwrapper() 的深度小于特权方法 doPrivileged() 的深度，所以安全管理器执行 S130 步骤，即判定所述待检查代码存在安全问题。

在本申请的一个具体的实施例中，如图 3 所示，在判断所述调用栈中方法是否存在签名之后，所述方法还可以包括如下步骤：

S122：若存在签名，从策略文件中获取预设签名；

S123：判断所述调用栈中方法的签名是否符合所述预设签名；若不符合所述预设签名，则执行 S130 步骤。

在实际的应用中，云计算平台一般只接受其平台提供的签名，而不接受别的平台提供的签名。或者所述云计算平台只接受其平台允许的签名，而不接受其它的签名。

本实施例中，所述预设签名可以是人为预先在策略文件中配置的。假设例子 2 所示的调用栈中第 5 层的方法 privilegedwrapper() 的签名为 “AAA”，而所述预设签名为 “alibaba”。那么，由于所述签名 “AAA” 不符合预设签名 “alibaba”，则执行 S130 步骤，即判定所述待检查代码存在安全问题。

在本申请的一个具体的实施例中，如图 4 所示，在判断所述调用栈中方法是否存在签

名之后，所述方法还可以包括如下步骤：

S124：若存在签名，获取所述调用栈中方法的类名；

S125：从策略文件中获取预设类名；

S126：判断所述调用栈中方法的类名是否符合所述预设类名；若不符合所述预设类名，

5 则执行 S130 步骤。

在实际的应用中，与上一实施例类似，云计算平台一般只接受其平台允许的类名，而不接受所述允许的类名以外的其它类名。

本实施例中，所述预设类名可以是人为预先在策略文件中配置的。

例如调用栈中方法的类名为：第 0 层的类名为 classA，第 1 层的类名为 classB，第 2
10 层的类名为 classC；如果从策略文件中获取的预设类名为 classA，classB，classC，classD，
则所述调用栈中方法的类名均符合所述预设类名，所以安全管理器可以判定所述待检查代
码是安全的。如果从策略文件中获取的预设类名为 classA，classC，classD，则所述调用
栈中方法的类名 classB 不符合所述预设类名，所以执行 S130 步骤，即判定所述待检查代
码存在安全问题。

15 在本申请的一个具体的实施例中，如图 5 所示，在判断所述调用栈中方法是否存在签
名之后，所述方法还可以包括如下步骤：

S127：若存在签名，从策略文件中获取预设调用顺序；

S128：判断所述调用栈中方法的调用顺序是否符合所述预设调用顺序；若不符合所述
预设调用顺序，则执行 S130 步骤。

20 本实施例中，所述预设调用顺序可以是人为预先在策略文件中配置的。

例如待检查代码的调用栈中方法的调用顺序为：A 调用 B，再调用 C；而从策略文件中
获取的预设调用顺序为：A 调用 D，再调用 C。那么，由于所述调用栈中方法的调用顺序不
符合所述预设调用顺序，服务器执行 S130 步骤，即判定所述待检查代码存在安全问题。

本申请实施例提供还提供一种装置，可以实现上述的方法步骤，且该装置可以通过软
25 件实现，也可以通过硬件或者软硬件结合的方式实现。以软件实现为例，作为逻辑意义上
的装置，是通过服务器的 CPU（Central Process Unit，中央处理器）将对应的计算机程序
指令读取到内存中运行形成的。

图 6 为本申请一实施例中提供的用于提高 Java 沙箱安全性的装置的模块示意图。本实
施例中，所述装置包括：

30 检查模块 200，用于对待检查代码进行权限检查；

第一判断模块 210，用于判断所述代码的调用栈中是否存在绕过所述权限检查的方法；

第二判断模块 220，用于在所述代码的调用栈中存在绕过所述权限检查的方法时，判断所述调用栈中方法是否存在签名；

确定模块 230，用于在所述调用栈中方法不存在签名时，判定所述待检查代码存在安全问题。

5 优选地，在所述第二判断模块 220 之后，还可以包括：

第一判断子模块，用于在所述调用栈中方法存在签名时，判断所述签名对应的方法的深度是否大于特权方法的深度；

相应地，所述确定模块 230，还用于在所述签名对应的方法的深度不大于特权方法的深度时，判定所述待检查代码存在安全问题。

10 优选地，在所述第二判断模块 220 之后，还可以包括：

第一获取子模块，用于在所述调用栈中方法存在签名时，从策略文件中获取预设签名；

第二判断子模块，用于判断所述调用栈中方法的签名是否符合所述预设签名；

相应地，所述确定模块 230，还用于在所述调用栈中方法的签名不符合所述预设签名时，判定所述待检查代码存在安全问题。

15 优选地，在所述第二判断模块 220 之后，还可以包括：

第二获取子模块，用于在所述调用栈中方法存在签名时，获取所述调用栈中方法的类名；

第三获取子模块，用于从策略文件中获取预设类名；

第三判断子模块，用于判断所述调用栈中方法的类名是否符合所述预设类名；

20 相应地，所述确定模块 230，还用于在所述调用栈中方法的类名不符合所述预设类名时，判定所述待检查代码存在安全问题。

优选地，在所述第二判断模块 220 之后，还可以包括：

第四获取子模块，用于在所述调用栈中方法存在签名时，从策略文件中获取预设调用顺序；

25 第四判断子模块，用于判断所述调用栈中方法的调用顺序是否符合所述预设调用顺序；

相应地，所述确定模块 230，还用于在所述调用栈中方法的调用顺序不符合所述预设调用顺序时，判定所述待检查代码存在安全问题。

本申请实施例所提供的提高 Java 沙箱安全性的方法通过对待检查代码进行权限检查；判断所述代码的调用栈中是否存在绕过所述权限检查的方法若是，则判断所述调用栈中方法是否存在签名；在所述调用栈中方法不存在签名时，判定所述待检查代码存在安全问题，从而实现提高 Java 沙箱安全性的目的。

30

在 20 世纪 90 年代，对于一个技术的改进可以很明显地区分是硬件上的改进（例如，对二极管、晶体管、开关等电路结构的改进）还是软件上的改进（对于方法流程的改进）。然而，随着技术的发展，当今的很多方法流程的改进已经可以视为硬件电路结构的直接改进。设计人员几乎都通过将改进的方法流程编程到硬件电路中来得到相应的硬件电路结构。

5 因此，不能说一个方法流程的改进就不能用硬件实体模块来实现。例如，可编程逻辑器件（Programmable Logic Device, PLD）（例如现场可编程门阵列（Field Programmable Gate Array, FPGA））就是这样一种集成电路，其逻辑功能由用户对器件编程来确定。由设计人员自行编程来把一个数字系统“集成”在一片 PLD 上，而不需要请芯片制造厂商来设计和制作专用的集成电路芯片。而且，如今，取代手工地制作集成电路芯片，这种编程也多半
10 改用“逻辑编译器（logic compiler）”软件来实现，它与程序开发撰写时所用的软件编译器相类似，而要编译之前的原始代码也得用特定的编程语言来撰写，此称之为硬件描述语言（Hardware Description Language, HDL），而 HDL 也并非仅有一种，而是有许多种，如 ABEL（Advanced Boolean Expression Language）、AHDL（Altera Hardware Description Language）、Confluence、CUPL（Cornell University Programming Language）、HDCal、
15 JHDL（Java Hardware Description Language）、Lava、Lola、MyHDL、PALASM、RHDH（Ruby Hardware Description Language）等，目前最普遍使用的是 VHDL（Very-High-Speed Integrated Circuit Hardware Description Language）与 Verilog。本领域技术人员也应该清楚，只需要将方法流程用上述几种硬件描述语言稍作逻辑编程并编程到集成电路中，就可以很容易得到实现该逻辑方法流程的硬件电路。

20 控制器可以按任何适当的方式实现，例如，控制器可以采取例如微处理器或处理器以及存储可由该（微）处理器执行的计算机可读程序代码（例如软件或固件）的计算机可读介质、逻辑门、开关、专用集成电路（Application Specific Integrated Circuit, ASIC）、可编程逻辑控制器和嵌入微控制器的形式，控制器的例子包括但不限于以下微控制器：ARC 625D、Atmel AT91SAM、Microchip PIC18F26K20 以及 Silicone Labs C8051F320，存储器
25 控制器还可以被实现为存储器的控制逻辑的一部分。本领域技术人员也知道，除了以纯计算机可读程序代码方式实现控制器以外，完全可以通过将方法步骤进行逻辑编程来使得控制器以逻辑门、开关、专用集成电路、可编程逻辑控制器和嵌入微控制器等的形式来实现相同功能。因此这种控制器可以被认为是一种硬件部件，而对其内包括的用于实现各种功能的装置也可以视为硬件部件内的结构。或者甚至，可以将用于实现各种功能的装置视为
30 既可以是实现方法的软件模块又可以是硬件部件内的结构。

上述实施例阐明的系统、装置、模块或单元，具体可以由计算机芯片或实体实现，或

者由具有某种功能的产品来实现。

为了描述的方便，描述以上装置时以功能分为各种单元分别描述。当然，在实施本申请时可以把各单元的功能在同一个或多个软件和/或硬件中实现。

本领域内的技术人员应明白，本发明的实施例可提供为方法、系统、或计算机程序产品。因此，本发明可采用完全硬件实施例、完全软件实施例、或结合软件和硬件方面的实施例的形式。而且，本发明可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质（包括但不限于磁盘存储器、CD-ROM、光学存储器等）上实施的计算机程序产品的形式。

本发明是参照根据本发明实施例的方法、设备（系统）、和计算机程序产品的流程图和/或方框图来描述的。应理解可由计算机程序指令实现流程图和/或方框图中的每一流程和/或方框、以及流程图和/或方框图中的流程和/或方框的结合。可提供这些计算机程序指令到通用计算机、专用计算机、嵌入式处理机或其他可编程数据处理设备的处理器以产生一个机器，使得通过计算机或其他可编程数据处理设备的处理器执行的指令产生用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的装置。

这些计算机程序指令也可存储在能引导计算机或其他可编程数据处理设备以特定方式工作的计算机可读存储器中，使得存储在该计算机可读存储器中的指令产生包括指令装置的制造品，该指令装置实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能。

这些计算机程序指令也可装载到计算机或其他可编程数据处理设备上，使得在计算机或其他可编程设备上执行一系列操作步骤以产生计算机实现的处理，从而在计算机或其他可编程设备上执行的指令提供用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的步骤。

在一个典型的配置中，计算设备包括一个或多个处理器（CPU）、输入/输出接口、网络接口和内存。

内存可能包括计算机可读介质中的非永久性存储器，随机存取存储器（RAM）和/或非易失性内存等形式，如只读存储器（ROM）或闪存（flash RAM）。内存是计算机可读介质的示例。

计算机可读介质包括永久性和非永久性、可移动和非可移动媒体可以由任何方法或技术来实现信息存储。信息可以是计算机可读指令、数据结构、程序的模块或其他数据。计算机的存储介质的例子包括，但不限于相变内存（PRAM）、静态随机存取存储器（SRAM）、

动态随机存取存储器 (DRAM)、其他类型的随机存取存储器 (RAM)、只读存储器 (ROM)、电可擦除可编程只读存储器 (EEPROM)、快闪记忆体或其他内存技术、只读光盘只读存储器 (CD-ROM)、数字多功能光盘 (DVD) 或其他光学存储、磁盒式磁带, 磁带磁磁盘存储或其他磁性存储设备或任何其他非传输介质, 可用于存储可以被计算设备访问的信息。按照本文
5 中的界定, 计算机可读介质不包括暂存电脑可读媒体 (transitory media), 如调制的数据信号和载波。

还需要说明的是, 术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含, 从而使得包括一系列要素的过程、方法、商品或者设备不仅包括那些要素, 而且还包括没有明确列出的其他要素, 或者是还包括为这种过程、方法、商品或者设备所固有的
10 要素。在没有更多限制的情况下, 由语句“包括一个……”限定的要素, 并不排除在包括所述要素的过程、方法、商品或者设备中还存在另外的相同要素。

本领域技术人员应明白, 本申请的实施例可提供为方法、系统或计算机程序产品。因此, 本申请可采用完全硬件实施例、完全软件实施例或结合软件和硬件方面的实施例的形式。而且, 本申请可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储
15 介质 (包括但不限于磁盘存储器、CD-ROM、光学存储器等) 上实施的计算机程序产品的形式。

本申请可以在由计算机执行的计算机可执行指令的一般上下文中描述, 例如程序模块。一般地, 程序模块包括执行特定任务或实现特定抽象数据类型的例程、程序、对象、组件、数据结构等等。也可以在分布式计算环境中实践本申请, 在这些分布式计算环境中, 由通
20 过通信网络而被连接的远程处理设备来执行任务。在分布式计算环境中, 程序模块可以位于包括存储设备在内的本地和远程计算机存储介质中。

本说明书中的各个实施例均采用递进的方式描述, 各个实施例之间相同相似的部分互相参见即可, 每个实施例重点说明的都是与其他实施例的不同之处。尤其, 对于系统实施例而言, 由于其基本相似于方法实施例, 所以描述的比较简单, 相关之处参见方法实施例
25 的部分说明即可。

以上所述仅为本申请的实施例而已, 并不用于限制本申请。对于本领域技术人员来说, 本申请可以有各种更改和变化。凡在本申请的精神和原理之内所作的任何修改、等同替换、改进等, 均应包含在本申请的权利要求范围之内。

权 利 要 求 书

1、一种用于提高 Java 沙箱安全性的方法，其特征在于，包括：

对待检查代码进行权限检查；

判断所述代码的调用栈中是否存在绕过所述权限检查的方法；

- 5 若存在绕过所述权限检查的方法，则判断所述调用栈中方法是否存在签名；
 若不存在签名，则判定所述待检查代码存在安全问题。

2、如权利要求 1 所述的方法，其特征在于，还包括：

若存在签名，则判断所述签名对应的方法的深度是否大于特权方法的深度；

若不大于特权方法的深度，则判定所述待检查代码存在安全问题。

- 10 3、如权利要求 1 所述的方法，其特征在于，还包括：

若存在签名，从策略文件中获取预设签名；

判断所述调用栈中方法的签名是否符合所述预设签名；

若不符合所述预设签名，则判定所述待检查代码存在安全问题。

4、如权利要求 1 所述的方法，其特征在于，还包括：

- 15 若存在签名，获取所述调用栈中方法的类名；

从策略文件中获取预设类名；

判断所述调用栈中方法的类名是否符合所述预设类名；

若不符合所述预设类名，则判定所述待检查代码存在安全问题。

5、如权利要求 1 所述的方法，其特征在于，还包括：

- 20 若存在签名，从策略文件中获取预设调用顺序；

判断所述调用栈中方法的调用顺序是否符合所述预设调用顺序；

若不符合所述预设调用顺序，则判定所述待检查代码存在安全问题。

6、一种用于提高 Java 沙箱安全性的装置，其特征在于，包括：

检查模块，用于对待检查代码进行权限检查；

- 25 第一判断模块，用于判断所述代码的调用栈中是否存在绕过所述权限检查的方法；
 第二判断模块，用于在所述代码的调用栈中存在绕过所述权限检查的方法时，判断
 所述调用栈中方法是否存在签名；

确定模块，用于在所述调用栈中方法不存在签名时，判定所述待检查代码存在安全问题。

- 30 7、如权利要求 6 所述的装置，其特征在于，还包括：

第一判断子模块，用于在所述调用栈中方法存在签名时，判断所述签名对应的方法的深度是否大于特权方法的深度；

相应地，所述确定模块，还用于在所述签名对应的方法的深度不大于特权方法的深度时，判定所述待检查代码存在安全问题。

5 8、如权利要求 6 所述的装置，其特征在于，还包括：

第一获取子模块，用于在所述调用栈中方法存在签名时，从策略文件中获取预设签名；

第二判断子模块，用于判断所述调用栈中方法的签名是否符合所述预设签名；

10 相应地，所述确定模块，还用于在所述调用栈中方法的签名不符合所述预设签名时，判定所述待检查代码存在安全问题。

9、如权利要求 6 所述的装置，其特征在于，还包括：

第二获取子模块，用于在所述调用栈中方法存在签名时，获取所述调用栈中方法的类名；

第三获取子模块，用于从策略文件中获取预设类名；

15 第三判断子模块，用于判断所述调用栈中方法的类名是否符合所述预设类名；

相应地，所述确定模块，还用于在所述调用栈中方法的类名不符合所述预设类名时，判定所述待检查代码存在安全问题。

10、如权利要求 6 所述的装置，其特征在于，还包括：

20 第四获取子模块，用于在所述调用栈中方法存在签名时，从策略文件中获取预设调用顺序；

第四判断子模块，用于判断所述调用栈中方法的调用顺序是否符合所述预设调用顺序；

相应地，所述确定模块，还用于在所述调用栈中方法的调用顺序不符合所述预设调用顺序时，判定所述待检查代码存在安全问题。

25

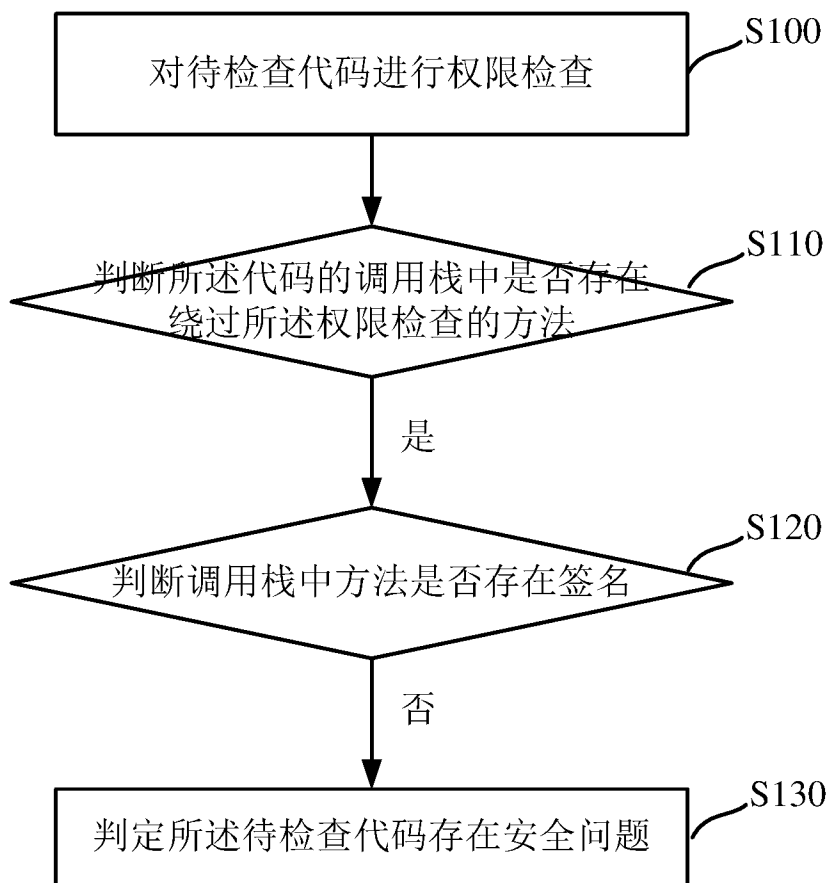


图 1

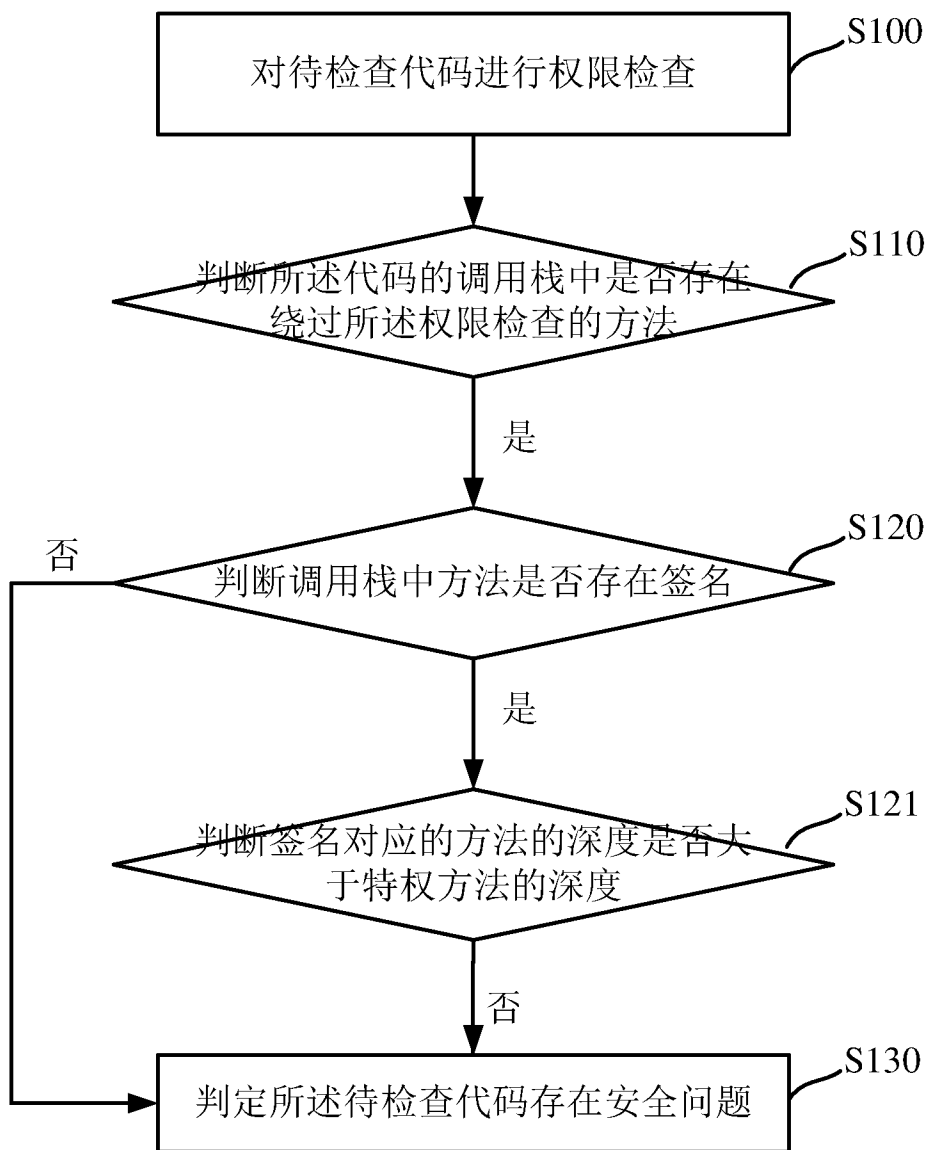


图 2

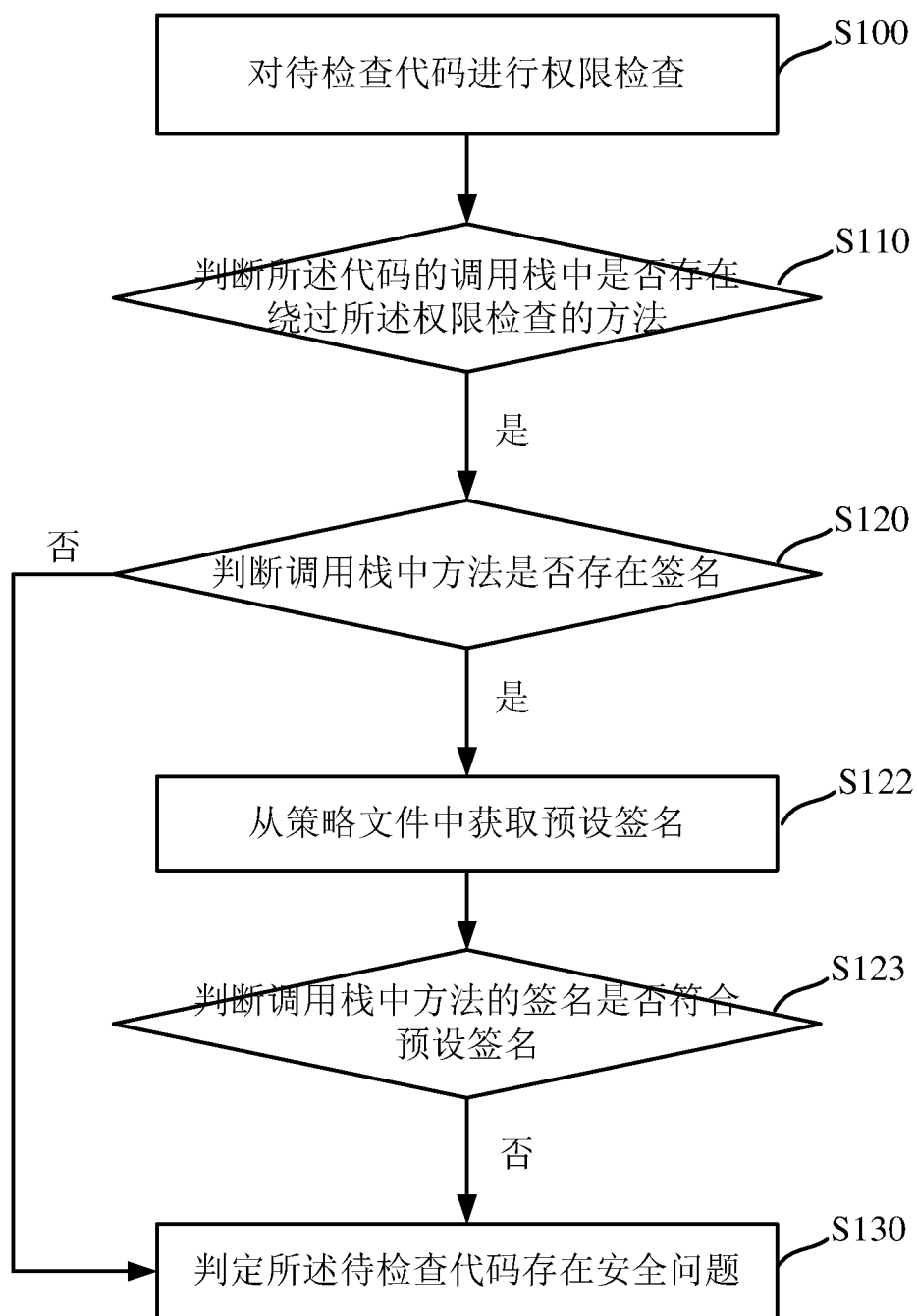


图 3

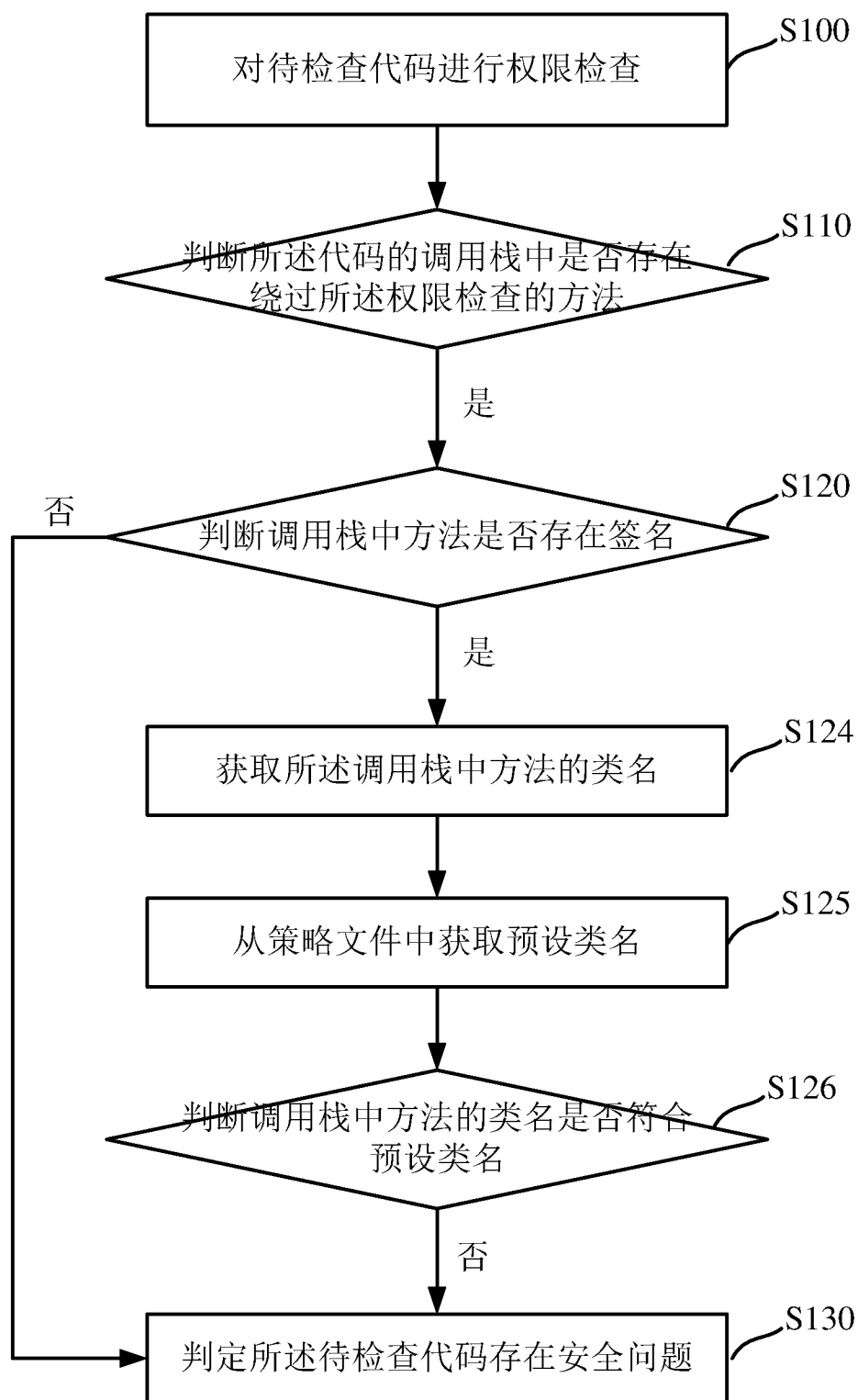


图4

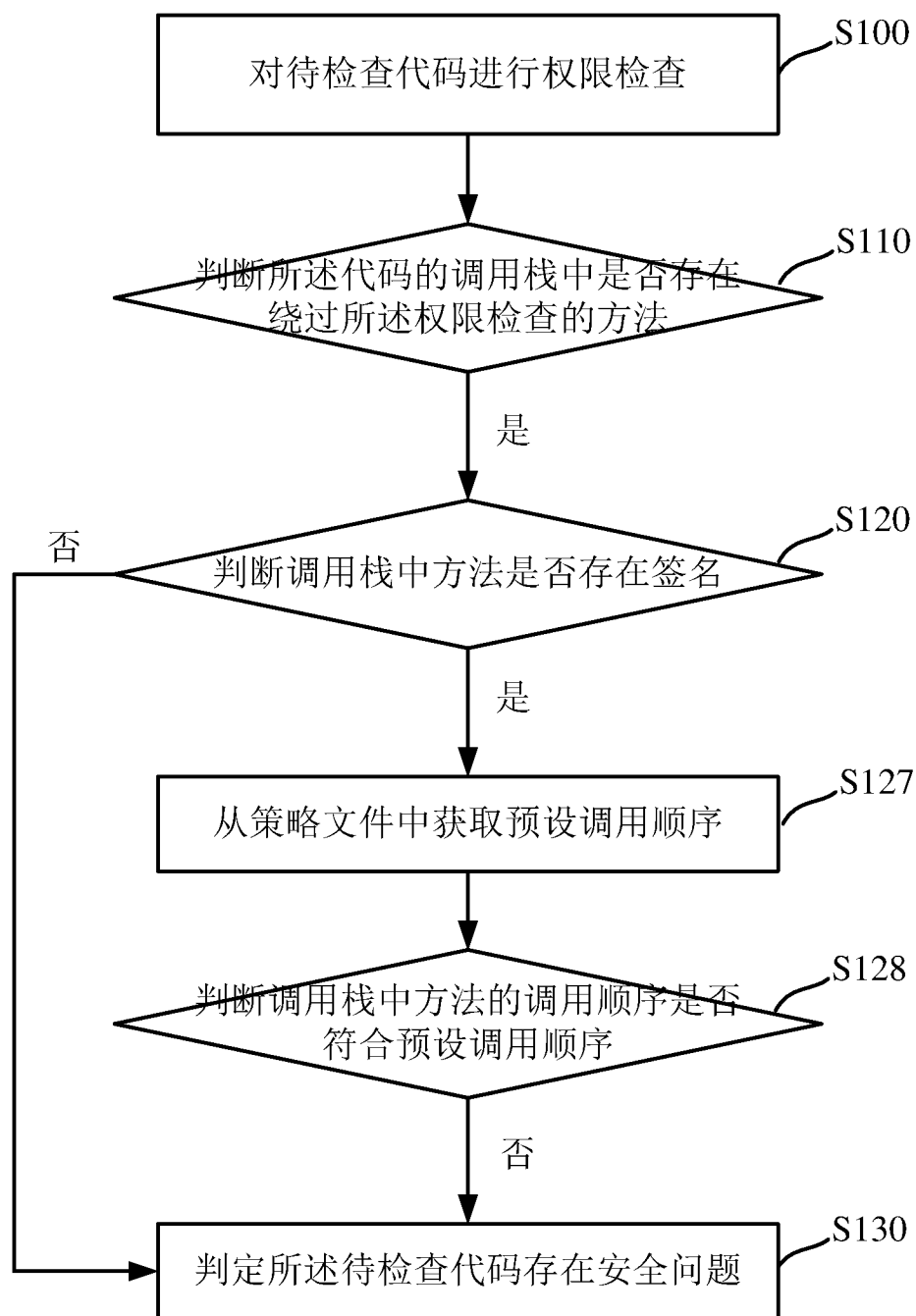


图 5

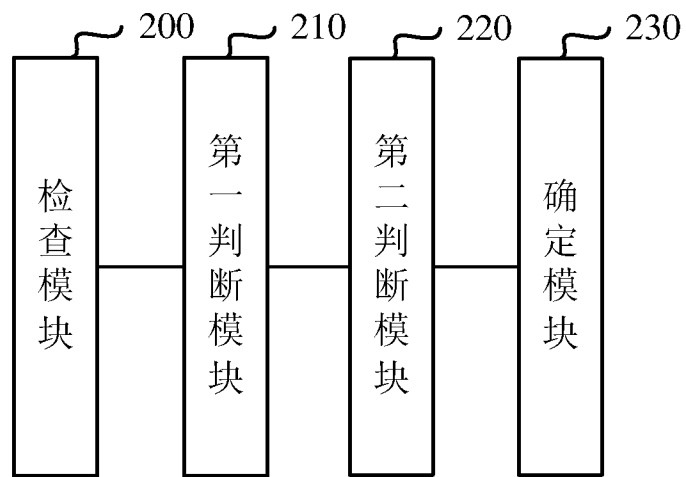


图 6

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2016/105822**A. CLASSIFICATION OF SUBJECT MATTER**

G06F 21/53 (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)

CNPAT, WPI, EPODOC, CNKI, IEEE: sandbox, security, authority, code, stack, signature, round, stop, check, depth, class name

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	CN 103136470 A (JIANGNAN INSTITUTE OF COMPUTING TECHNOLOGY), 05 June 2013 (05.06.2013), description, paragraphs [0003]-[0005]	1-10
A	CN 104216741 A (HUAZHONG UNIVERSITY OF SCIENCE AND TECHNOLOGY), 17 December 2014 (17.12.2014), the whole document	1-10
A	CN 103593605 A (FUDAN UNIVERSITY), 19 February 2014 (19.02.2014), the whole document	1-10
A	US 6836888 B1 (LUCENT TECHNOLOGIES INC.), 28 December 2004 (28.12.2004), the whole document	1-10

☐ Further documents are listed in the continuation of Box C.☒ See patent family annex.

* Special categories of cited documents:	“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
“A” document defining the general state of the art which is not considered to be of particular relevance	“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
“E” earlier application or patent but published on or after the international filing date	“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
“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)	“&” document member of the same patent family
“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	

Date of the actual completion of the international search 09 January 2017 (09.01.2017)	Date of mailing of the international search report 06 February 2017 (06.02.2017)
Name and mailing address of the ISA/CN: State Intellectual Property Office of the P. R. China No. 6, Xitucheng Road, Jimenqiao Haidian District, Beijing 100088, China Facsimile No.: (86-10) 62019451	Authorized officer LI, Yukun Telephone No.: (86-10) 62414421

International application No.
PCT/CN2016/105822

Form PCT/ISA/210 (patent family annex) (July 2009)

A. 主题的分类 G06F 21/53 (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)) CNPAT, WPI, EPODOC, CNKI, IEEE: 沙箱, 安全, 权限, 代码, 栈, 签名, 绕过, 停止, 检查, 深度, 类名, sandbox, security, authority, code, stack, signature, round, stop, check, depth, class name																	
C. 相关文件 <table border="1"> <thead> <tr> <th>类 型*</th> <th>引用文件, 必要时, 指明相关段落</th> <th>相关的权利要求</th> </tr> </thead> <tbody> <tr> <td>A</td> <td>CN 103136470 A (无锡江南计算技术研究所) 2013年 6月 5日 (2013 - 06 - 05) 说明书第[0003]-[0005]段</td> <td>1-10</td> </tr> <tr> <td>A</td> <td>CN 104216741 A (华中科技大学) 2014年 12月 17日 (2014 - 12 - 17) 全文</td> <td>1-10</td> </tr> <tr> <td>A</td> <td>CN 103593605 A (复旦大学) 2014年 2月 19日 (2014 - 02 - 19) 全文</td> <td>1-10</td> </tr> <tr> <td>A</td> <td>US 6836888 B1 (LUCENT TECHNOLOGIES INC.) 2004年 12月 28日 (2004 - 12 - 28) 全文</td> <td>1-10</td> </tr> </tbody> </table>			类 型*	引用文件, 必要时, 指明相关段落	相关的权利要求	A	CN 103136470 A (无锡江南计算技术研究所) 2013年 6月 5日 (2013 - 06 - 05) 说明书第[0003]-[0005]段	1-10	A	CN 104216741 A (华中科技大学) 2014年 12月 17日 (2014 - 12 - 17) 全文	1-10	A	CN 103593605 A (复旦大学) 2014年 2月 19日 (2014 - 02 - 19) 全文	1-10	A	US 6836888 B1 (LUCENT TECHNOLOGIES INC.) 2004年 12月 28日 (2004 - 12 - 28) 全文	1-10
类 型*	引用文件, 必要时, 指明相关段落	相关的权利要求															
A	CN 103136470 A (无锡江南计算技术研究所) 2013年 6月 5日 (2013 - 06 - 05) 说明书第[0003]-[0005]段	1-10															
A	CN 104216741 A (华中科技大学) 2014年 12月 17日 (2014 - 12 - 17) 全文	1-10															
A	CN 103593605 A (复旦大学) 2014年 2月 19日 (2014 - 02 - 19) 全文	1-10															
A	US 6836888 B1 (LUCENT TECHNOLOGIES INC.) 2004年 12月 28日 (2004 - 12 - 28) 全文	1-10															
☐ 其余文件在C栏的续页中列出。 ☒ 见同族专利附件。 <table border="0"> <tr> <td> * 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 </td> <td> “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 </td> </tr> </table>			* 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													
* 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 2017年 1月 9日		国际检索报告邮寄日期 2017年 2月 6日															
ISA/CN的名称和邮寄地址 中华人民共和国国家知识产权局 (ISA/CN) 中国北京市海淀区蓟门桥西土城路6号 100088 传真号 (86-10) 62019451		受权官员 李玉坤 电话号码 (86-10) 62414421															

国际检索报告
关于同族专利的信息

国际申请号

PCT/CN2016/105822

检索报告引用的专利文件			公布日 (年/月/日)	同族专利	公布日 (年/月/日)
CN	103136470	A	2013年 6月 5日	无	
CN	104216741	A	2014年 12月 17日	无	
CN	103593605	A	2014年 2月 19日	无	
US	6836888	B1	2004年 12月 28日	无	