(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 113168476 B
(45)授权公告日 2024.11.01

(21)申请号 201980078025.4
(22)申请日 2019.11.26
(65)同一申请的已公布的文献号
申请公布号 CN 113168476 A
(43)申请公布日 2021.07.23
(30)优先权数据
62/773,524 2018.11.30 US
(85)PCT国际申请进入国家阶段日
2021.05.27
(86)PCT国际申请的申请数据
PCT/CA2019/051687 2019.11.26
(87)PCT国际申请的公布数据
WO2020/107104 EN 2020.06.04
(73)专利权人 百可德罗德公司
地址 加拿大安大略省彼得堡市米尔恩大道84号
专利权人 佰倬信息科技有限责任公司
(72)发明人 余祥 孟进 杨恩辉
(74)专利代理机构 无锡华源专利商标事务所(普通合伙) 32228
专利代理师 聂启新
(51)Int.Cl.
*G06F 21/60*(2013.01)
*G06F 21/51*(2013.01)
*G06F 21/74*(2013.01)
*G06F 9/445*(2018.01)
(56)对比文件
US 2011213971 A1,2011.09.01
US 2016253519 A1,2016.09.01
审查员 范园园

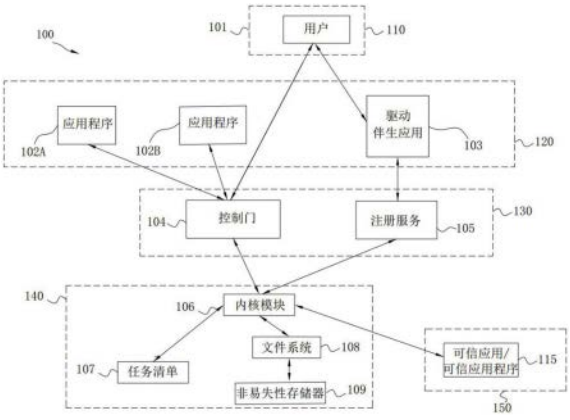
权利要求书9页 说明书38页 附图3页

(54)发明名称

操作系统中个性化密码学安全的访问控制

(57)摘要

本发明提出了一个访问控制系统,它包括一个被配置为提供与富执行环境隔离的可信执行环境的处理器。富操作系统在富执行环境中运行,而可信操作系统在可信执行环境中运行。多个受保护的数据文件被存储在非易失性存储器中。当一个进程请求访问一个受保护的数据文件时,仅当存在与请求进程相对应的验证的应用程序令牌时,计算机系统才允许该请求进程访问被请求数据文件。为关联应用程序生成一个应用程序令牌的过程如下:检测与所述关联应用程序相关联的第一进程的启动;确定在可信执行环境内有一个有效的用户码可用;在确定可信执行环境内有所述有效的用户码可用之后,使用该有效的用户码生成所述应用程序令牌。




CN 113168476 B

1. 一种用于计算设备的访问控制系统,所述访问控制系统包括:

(a)一个被配置为提供至少两个隔离执行环境的处理器,其中,所述至少两个隔离执行环境包括富执行环境和可信执行环境,所述处理器被配置为在所述富执行环境中操作富操作系统,在可信执行环境中操作可信操作系统;和

(b)被配置为存储多个数据文件和多个应用指令集的非易失性存储器,其中每个应用指令集对应于安装在所述计算设备上的一个应用,并且每个应用被配置为在富操作系统中运行,其中,所述多个数据文件包括多个受保护的数据文件;

(c)其中,该处理器被配置为:

(i)接收对存储在非易失性存储器中的被请求数据文件的一个文件请求,其中,所述被请求数据文件对应于所述多个受保护的数据文件中的一个受保护的数据文件,并且其中,该文件请求是从一个运行于富操作系统内的请求进程那里收到的;

(ii)确定一个与上述请求进程相关联的关联应用程序,其中,该关联应用程序对应于安装在所述计算设备上的应用程序中的一个应用程序;

(iii)确定上述关联应用程序是否存在一个验证的应用程序令牌;和

(iv)仅当存在上述验证的应用程序令牌时,才允许上述请求进程访问所述被请求数据文件,否则阻止该请求进程访问所述被请求数据文件;

其中, 上述处理器被配置为通过以下方式为所述关联应用程序生成所述验证的应用程序令牌:

检测与所述关联应用程序相关联的第一进程的启动;

确定在可信执行环境内有一个有效的用户码可用;和

在确定可信执行环境内有所述有效的用户码可用后,使用该有效的用户码生成应用程序令牌。

2.根据权利要求1所述的系统,其中:

(a)所述处理器被配置为:

(i)结合所述第一进程的启动,生成一个应用程序启动提示,其中,该应用程序启动提示被定义为提示用户输入一个用户码;

(ii)接收一个响应于所述应用程序启动提示的应用程序启动输入;

(iii)确定所述应用程序启动输入是否与所述有效的用户码相对应;和

(iv)在确定所述应用程序启动输入对应于所述有效的用户码后,使用所述用户码生成上述应用程序令牌。

3.根据权利要求2所述的系统,其中:

(a)确定所述应用程序启动输入是否对应于所述有效的用户码,包括以下步骤:

(i)从所述应用程序启动输入中,确定一个接收到的用户码;

(ii)通过对接收到的用户码进行哈希计算来确定一个哈希的用户码;

(iii)将该哈希的用户码与存储在可信执行环境的可信非易失性存储器中的存储的哈希的用户码进行比较;和

(iv)当该哈希的用户码与所述存储的哈希的用户码匹配时,确定接收到的用户码对应于所述有效的用户码,否则确定接收到的用户码无效。

4.根据权利要求1至3中任一权利要求所述的系统,其中:

(a)所述处理器被配置为通过以下方式生成所述应用程序令牌:

(i)确定与第一进程的启动相对应的时间戳;和

(ii)使用时间戳和所述有效的用户码生成所述应用程序令牌。

5. 根据权利要求1所述的系统,其中,所述处理器被配置为:

(a)将每个受保护的数据文件与一个文件特定的应用程序组相关联,其中,所述文件特定的应用程序组包括至少一个安装在所述计算设备上的应用程序;和

(b)仅在与所述请求进程相关联的关联应用程序是与所述被请求数据文件相关联的文件特定的应用程序组中的一个应用程序时,才允许所述请求进程访问所述被请求数据文件。

6. 根据权利要求5所述的系统,其中,所述处理器被配置为:

(a)定义一个受保护的应用程序组,其中所述受保护的应用程序组包括至少安装在所述计算设备上的应用程序中的之一;和

(b)仅在所述关联应用程序是所述受保护的应用程序之一时,才允许所述请求进程访问所述被请求数据文件。

7.根据权利要求6所述的系统,其中,所述处理器被配置为:对于由所述受保护的应用程序之一生成的每个新文件,自动地将该文件存储为一个所述受保护的数据文件之一。

8. 根据权利要求7所述的系统,其中:

(a)对于安装在所述计算设备上的每个应用程序,所述处理器被配置为,为该应用程序分配一个唯一的应用程序标识,其中该应用程序标识是通过使用一个唯一的由所述处理器定义的应用程序识别码和一个与该应用程序对应的应用程序证书来定义的;和

(b)所述处理器被配置为,通过使用所述关联应用程序的唯一应用程序标识和所述有效的用户码,来为所述关联应用程序生成所述应用程序令牌。

9.根据权利要求8所述的系统,其中:

(a)对于安装在所述计算设备上的每个应用程序,所述处理器被配置为通过以下方式定义所述应用程序标识:

(i)通过对与该应用程序对应的应用程序证书进行哈希计算,来生成一个哈希的应用程序证书;和

(ii)将所述应用程序识别码和上述哈希的应用程序证书组合在一起。

10.根据权利要求8和9中任一权利要求所述的系统,其中,所述处理器被配置为,在与所述关联应用程序相关联的所述第一进程启动时,

(a)验证与该关联应用程序相对应的应用程序证书;

(b)生成一个响应于验证所述应用程序证书的验证的应用程序启动提示,其中,所述验证的应用程序启动提示被定义为提示用户输入用户码;

(c)接收一个响应于所述验证的应用程序启动提示的验证的应用程序启动输入;

(d)确定上述验证的应用程序启动输入是否与所述有效的用户码相对应;和

(e)仅在所述应用程序证书已被验证并且所述验证的应用程序启动输入对应于所述有效的用户码之后,才允许所述关联应用程序执行。

11.根据权利要求1所述的系统,其中,所述处理器被配置为:

(a)对于安装在所述计算设备上的每个应用程序,为该应用程序分配一个唯一的应用

程序标识,其中该应用程序标识是通过使用一个唯一的由所述处理器定义的应用程序识别码和与该应用程序相对应的应用程序证书来定义的;

(b)定义一个受保护的应用程序组,其中所述受保护的应用程序组包括至少安装在所述计算设备上的应用程序中的之一;

(c)定义一个访问策略文件,其中,该访问策略文件包括所述受保护的应用程序组中每个受保护的应用程序的唯一应用程序标识;

(d)将上述访问策略文件存储在非易失性存储器中;和

(e)仅在上述访问策略文件包含所述关联应用程序的唯一应用程序标识时,才允许所述请求进程访问所述被请求数据文件。

12.根据权利要求11所述的系统,其中,所述处理器被配置为:

(a)用与对应于所述有效的用户码的哈希的码相关联的方式,将所述访问策略文件存储在非易失性存储器中;

(b)禁止对所述访问策略文件的修改,除与所述有效的用户码相对应的策略修改输入收到,并且同与所述有效的用户码相对应的所述哈希的码进行验证;和

(c)允许在没有策略修改输入的情况下对所述访问策略文件执行读取操作。

13. 根据权利要求12所述的系统,其中,所述处理器被配置为:

(a)在富操作系统内可访问的内核内存中定义一个访问策略数据结构;和

(b)将该访问策略数据结构与存储在非易失性存储器中的所述访问策略文件同步,使得所述访问策略数据结构包括所述受保护的应用程序组中每个受保护的应用程序的唯一应用程序标识。

14.根据权利要求13所述的系统,其中,所述处理器被配置为:在与所述关联应用程序相关联的所述第一进程启动时,

(a)验证与所述关联应用程序相对应的应用程序证书;

(b)在验证应用程序证书后,通过以下方式确定所述关联应用程序是否为受保护的应用程序之一:

(i)确定所述关联应用程序的唯一应用程序标识;和

(ii)确定所述关联应用程序的唯一应用程序标识是否包含在所述访问策略数据结构中;和

(iii)当所述访问策略数据结构包含所述关联应用程序的唯一应用程序标识时,确定所述关联应用程序是受保护的应用程序之一;否则,确定所述关联应用程序是不受保护的应用程序;

(c)在确定所述关联应用程序是受保护的应用程序之一之后,

(i)生成一个受保护的应用程序启动提示,其中,该受保护的应用程序启动提示被定义为提示用户输入一个用户码;

(ii)接收一个响应于所述受保护的应用程序启动提示的受保护的应用程序启动输入;

(iii)确定所述受保护的应用程序启动输入是否与所述有效的用户码相对应;和

(iv)仅在所述处理器确定所述受保护的应用程序启动输入对应于所述有效的用户码之后,才允许所述关联应用程序执行;和

(d)在确定所述关联应用程序是不受保护的应用程序后,允许所述关联应用程序执行。

15.根据权利要求14所述的系统,其中所述处理器被配置为,在所述请求进程启动时,

(a)确定与所述请求进程相关联的进程标识符;

(b)验证与所述关联应用程序相对应的应用程序证书;

(c)在验证应用程序证书后,确定所述关联应用程序是否为受保护的应用程序之一;

(d)在确定所述关联应用程序是受保护的应用程序之一时,生成一个进程启动提示,其中该进程启动提示被定义为提示用户输入一个用户码;

(e)接收一个响应于所述进程启动提示的进程启动输入;

(f)确定所述进程启动输入是否对应于所述有效的用户码;

(g)在确定所述关联应用程序是受保护的应用程序之一,并且所述进程启动输入对应于所述有效的用户码后,将所述请求进程的进程标识符存储在富操作系统内可访问的内核内存中的一个授权进程数据结构中;和

(h)仅在所述授权进程数据结构包含所述请求进程的进程标识符的情况下,才允许所述请求该进程访问所述被请求数据文件。

16. 根据权利要求15所述的系统,其中,所述处理器被配置为:

(a)接收来自运行于富操作系统内的特定进程的写操作请求,其中,该写操作请求与一个受保护的数据文件有关;和

(b)仅当所述授权进程数据结构包含上述特定进程的进程标识符时,才允许上述写操作请求发生。

17.根据权利要求15和16中任一权利要求所述的系统,其其中,所述处理器被配置为:

(a)检测在富操作系统中运行的行将终止进程的终止;

(b)确定所述授权进程数据结构包含上述终止进程的进程标识符;和

(c)更新所述授权进程数据结构,以删除上述终止进程的进程标识符。

18.根据权利要求11至16中任一权利要求所述的系统,其中,所述处理器被配置为:

(a)用与对应于所述有效的用户码的哈希的码相关联的方式,将所述访问策略文件存储在非易失性存储器中;

(b)接收一个策略更新输入,该输入指定要把一个特定应用程序添加到所述受保护的应用程序组中;

(c)生成一个响应于上述策略更新输入的更新验证提示,其中,该更新验证提示被定义为提示用户输入一个用户码;

(d)接收一个响应于上述更新验证提示的更新验证输入;

(e)使用用与所述访问策略文件关联的方式存储的,与所述有效的用户码相对应的哈希的码,确定上述更新验证输入是否与所述有效的用户码相对应;

(f)在确定上述更新验证输入对应于所述有效的用户码后,

(i)更新所述访问策略文件以包含上述特定应用程序;

(ii)识别存储在非易失性存储器中的,与该特定应用程序相对应的一组现有数据文件;和

(iii)将每个现有数据文件从不受保护的数据文件修改为受保护的数据文件。

19.一种控制对存储在计算设备的非易失性存储器中的多个数据文件的访问的方法,其中,所述多个数据文件包括多个受保护的数据文件,所述方法由一个处理器来执行,而该

处理器被配置为在一个富执行环境中运行一个富操作系统并且在一个可信执行环境中运行一个可信的操作系统,该方法包括:

(a)由上述处理器接收对存储在非易失性存储器中的被请求数据文件的一个文件请求,其中,所述被请求数据文件对应于所述多个受保护的数据文件中的一个受保护的数据文件,并且其中,该文件请求是从一个运行于富操作系统内请求进程那里收到的;

(b)由上述处理器确定一个与该请求进程相关联的关联应用程序,其中,该关联应用程序对应于安装在所述计算设备上的应用程序中的一个应用程序;

(c)由上述处理器确定所述关联应用程序是否存在一个验证的应用程序令牌;和

(d)仅当存在上述验证的应用程序令牌时,所述处理器才会允许上述请求进程访问所述被请求数据文件,否则阻止该请求进程访问所述被请求数据文件;

其中,上述处理器通过以下方式为所述关联应用程序生成所述验证的应用程序令牌:

检测与所述关联应用程序相关联的第一进程的启动;

确定在可信执行环境内存在一个有效的用户码可用;和

在确定在可信执行环境内有所述有效的用户码可用后,使用该有效的用户码生成应用程序令牌。

20.根据权利要求19所述的方法,还包括:

(i)由所述处理器结合所述第一进程的启动,生成一个应用程序启动提示,其中,该应用程序启动提示被定义为提示用户输入一个用户码;

(ii)由所述处理器接收一个响应于所述应用程序启动提示的应用程序启动输入;

(iii)由所述处理器确定所述应用程序启动输入是否与所述有效的用户码相对应;和

(iv)在确定所述应用程序启动输入对应于所述有效的用户码后,由所述处理器使用所述用户码生成上述应用程序令牌。

21.根据权利要求20所述的方法,其中:

(a)确定所述应用程序启动输入是否对应于所述有效的用户码,包括以下步骤:

(i)从所述应用程序启动输入中,确定一个接收到的用户码;

(ii)通过对接收到的用户码进行哈希计算来确定一个哈希的用户码;

(iii)将该哈希的用户码与存储在可信执行环境的可信非易失性存储器中的存储的哈希的用户码进行比较;和

(iv)当该哈希的用户码与所述存储的哈希的用户码匹配时,确定接收到的用户码对应于所述有效的用户码,否则确定接收到的用户码无效。

22.根据权利要求19至21中任一权利要求所述的方法,还包括:

(a)通过以下方式生成所述应用程序令牌:

(i)确定与第一进程的启动相对应的时间戳;和

(ii)使用时间戳和所述有效的用户码生成所述应用程序令牌。

23. 根据权利要求19所述的方法,还包括:

(a)将每个受保护的数据文件与一个文件特定的应用程序组相关联,其中,所述文件特定的应用程序组包括至少一个安装在所述计算设备上的应用程序;和

(b)仅在与所述请求进程相关联的关联应用程序是与所述被请求数据文件相关联的文件特定的应用程序组中的一个应用程序时,才允许所述请求进程访问所述被请求数据文

件。

24. 根据权利要求23所述的方法,还包括:

(a)定义一个受保护的应用程序组,其中所述受保护的应用程序组包括至少安装在所述计算设备上的应用程序中的之一;和

(b)仅在所述关联应用程序是所述受保护的应用程序之一时,才允许所述请求进程访问所述被请求数据文件。

25.根据权利要求23所述的方法,还包括:对于由所述受保护的应用程序之一生成的每个新文件,自动地将该文件存储为所述受保护的数据文件之一。

26. 根据权利要求25所述的方法,还包括:

(a)对于安装在所述计算设备上的每个应用程序,为该应用程序分配一个唯一的应用程序标识,其中,该应用程序标识是通过使用一个唯一的由所述处理器定义的应用程序识别码和一个与该应用程序相对应的应用程序证书来定义的;和

(b)通过使用所述关联应用程序的唯一应用程序标识和所述有效的用户码,来为所述关联应用程序生成所述应用程序令牌。

27.根据权利要求26所述的方法,还包括:

(a)对于安装在所述计算设备上的每个应用程序,通过以下方式定义所述应用程序标识:

(i)通过对与该应用程序对应的应用程序证书进行哈希计算,来生成一个哈希的应用程序证书;和

(ii)将所述应用程序识别码和上述哈希的应用程序证书组合在一起。

28.根据权利要求26和27中任一项权利要求所述的方法,还包括:在与所述关联应用程序相关联的所述第一进程启动时,

(a)验证与该关联应用程序相对应的应用程序证书;

(b)生成一个响应于验证所述应用程序证书的验证的应用程序启动提示,其中,所述验证的应用程序启动提示被定义为提示用户输入一个用户码;

(c)接收一个响应于所述验证的应用程序启动提示的验证的应用程序启动输入;

(d)确定上述验证的应用程序启动输入是否与所述有效的用户码相对应;和

(e)仅在所述应用程序证书已被验证并且所述验证的应用程序启动输入对应于所述有效的用户码之后,才允许所述关联应用程序执行。

29.根据权利要求19所述的方法,还包括:

(a)对于安装在所述计算设备上的每个应用程序,为该应用程序分配一个唯一的应用程序标识,其中,该应用程序标识是通过使用一个唯一的由所述处理器定义的应用程序识别码和与该应用程序相对应的应用程序证书来定义的;

(b)定义一个受保护的应用程序组,其中所述受保护的应用程序组包括至少安装在所述计算设备上的应用程序中的之一;

(c)定义一个访问策略文件,其中,该访问策略文件包括所述受保护的应用程序组中每个受保护的应用程序的唯一应用程序标识;

(d)将上述访问策略文件存储在非易失性存储器中;和

(e)仅在上述访问策略文件包含所述关联应用程序的唯一应用程序标识时,才允许所

述请求进程访问所述被请求数据文件。

30.根据权利要求29所述的方法,还包括:

(a)用与对应于所述有效的用户码的哈希的码相关联的方式,将所述访问策略文件存储在非易失性存储器中;

(b)禁止对所述访问策略文件进行修改,除与所述有效的用户码相对应的策略修改输入收到,并且同与所述有效的用户码相对应的所述哈希的码进行验证通;和

(c)允许在没有策略修改输入的情况下对所述访问策略文件执行读取操作。

31. 根据权利要求30所述的方法,还包括:

(a)在富操作系统内可访问的内核内存中定义一个访问策略数据结构;和

(b)将该访问策略数据结构与存储在非易失性存储器中的所述访问策略文件同步,使得所述访问策略数据结构包括所述受保护的应用程序组中每个受保护的应用程序的唯一应用程序标识。

32.根据权利要求31所述的方法,还包括:在与所述关联应用程序相关联的所述第一进程启动时,

(a)验证与所述关联应用程序相对应的应用程序证书;

(b)在验证应用程序证书后,通过以下方式确定所述关联应用程序是否为受保护的应用程序之一:

(i)确定所述关联应用程序的唯一应用程序标识;和

(ii)确定所述关联应用程序的唯一应用程序标识是否包含在所述访问策略数据结构中;和

(iii)当所述访问策略数据结构包含所述关联应用程序的唯一应用程序标识时,确定所述关联应用程序是受保护的应用程序之一;否则,确定所述关联应用程序是不受保护的应用程序;

(c)在确定所述关联应用程序是受保护的应用程序之一之后,

(i)生成一个受保护的应用程序启动提示,其中,该受保护的应用程序启动提示被定义为提示用户输入一个用户码;

(ii)接收一个响应于所述受保护的应用程序启动提示的受保护的应用程序启动输入;

(iii)确定所述受保护的应用程序启动输入是否与所述有效的用户码相对应;和

(iv)仅在所述处理器确定所述受保护的应用程序启动输入对应于所述有效的用户码之后,才允许所述关联应用程序执行;和

(d)在确定所述关联应用程序是不受保护的应用程序后,允许所述关联应用程序执行。

33.根据权利要求29所述的方法,还包括:在所述请求进程启动时,

(a)确定与所述请求进程相关的进程标识符;

(b)验证与所述关联应用程序相对应的应用程序证书;

(c)在验证应用程序证书后,确定所述关联应用程序是否为受保护的应用程序之一;

(d)在确定所述关联应用程序是受保护的应用程序之一时,生成一个进程启动提示,其中,该进程启动提示被定义为提示用户输入一个用户码;

(e)接收一个响应于所述进程启动提示的进程启动输入;

(f)确定所述进程启动输入是否对应于所述有效的用户码;

(g)在确定所述关联应用程序是受保护的应用程序之一,并且所述进程启动输入对应于所述有效的用户码后,将所述请求进程的进程标识符存储到富操作系统内可以访问的内核内存中的一个授权进程数据结构中;和

(h)仅在所述授权进程数据结构包含所述请求进程的进程标识符的情况下,才允许所述请求进程访问所述被请求数据文件。

34. 根据权利要求33所述的方法,还包括:

(a)接收来自运行于富操作系统内的特定进程的写操作请求,其中,该写操作请求与一个受保护的数据文件有关;和

(b)仅当所述授权进程数据结构包含上述特定进程的进程标识符时,才允许该写操作请求发生。

35.根据权利要求33和34中任一权利要求所述的方法,还包括:

(a)检测在富操作系统中运行的行将终止进程的终止;

(b)确定所述授权进程数据结构包含上述行将终止进程的进程标识符;和

(c)更新所述授权进程数据结构,以删除上述终止进程的进程标识符。

36.根据权利要求29所述的方法,还包括:

(a)用与对应于所述有效的用户码的哈希的码相关联的方式,将所述访问策略文件存储在非易失性存储器中;

(b)接收一个策略更新输入,该输入指定出要把一个特定应用程序添加到所述受保护的应用程序组中;

(c)生成一个响应于上述策略更新输入的更新验证提示,其中,该更新验证提示被定义为提示用户输入一个用户码;

(d)接收一个响应于上述更新验证提示的更新验证输入;

(e)使用用与所述访问策略文件相关联的方式存储的,与所述有效的用户码相对应的哈希的码,确定上述更新验证输入是否与所述有效的用户码相对应;

(f)在确定上述更新验证输入对应于所述有效的用户码后,

(i)更新所述访问策略文件以包含上述特定应用程序;

(ii)识别存储在非易失性存储器中的与该特定应用相对应的一组现有数据文件;和

(iii)将每个现有数据文件从不受保护的数据文件修改为受保护的数据文件。

37. 一种计算机可读介质,其上存储有用于控制对存储在计算设备的非易失性存储器中的多个数据文件的访问的非暂时性计算机可读指令,其中,所述计算设备包含一个处理器,所述处理器被配置为在富执行环境中运行富操作系统和在可信执行环境中运行可信操作系统,其中所述多个数据文件包括多个受保护的数据文件, 其中所述指令被定义为配置所述处理器做以下操作:

(a)接收对存储在非易失性存储器中的被请求数据文件的一个文件请求,其中,所述被请求数据文件对应于所述多个受保护的数据文件中一个受保护的数据文件,并且其中,该文件请求是从一个运行于富操作系统内的请求进程那里收到的;

(b)确定一个与上述请求进程相关联的关联应用程序,其中,该关联应用程序对应于安装在所述计算设备上的应用程序中的一个应用程序;

(c)确定上述关联应用程序是否存在一个验证的应用程序令牌;和

(d)仅当存在上述验证的应用程序令牌时,才允许该请求进程访问所述被请求数据文件,否则阻止该请求进程访问所述被请求数据文件;

其中,所述指令被定义为配置上述处理器通过以下方式为所述关联应用程序生成所述验证的应用程序令牌:

检测与所述关联应用程序相关联的第一进程的启动;

确定在可信执行环境内存在一个有效的用户码可用;和

在确定可信执行环境内有所述有效的用户码可用后,使用该有效的用户码生成应用程序令牌。

38.根据权利要求37所述的计算机可读介质,还包括被定义为配置所述处理器以执行权利要求20至36中的任一权利要求所述的方法的指令。

# 操作系统中个性化密码学安全的访问控制

[0001] 相关申请的交叉引用

[0002] 本申请要求于2018年11月30日提交的美国临时申请62/773,524的权益,其全部内容通过引用合并于此。

## 技术领域

[0003] 所描述的实施例涉及用于文件访问控制的系统和方法,并且尤其涉及用于管理对操作系统中受保护的数据文件的访问的系统和方法。

## 背景技术

[0004] 以下内容不表示下面讨论的任何内容是现有先进技术的一部分或本领域技术人员的公知常识的一部分。

[0005] 在日益数字化的世界中,数据是对人、公司和国家最重要的资产之一。结果就是,人们采取了广泛措施以确保敏感数据的安全性。但是,随着黑客技术越来越激进,数据安全挑战也日益严峻。

[0006] 在任何操作系统中,数据安全性都是至关重要的。但是,不同的操作系统可能会在提供数据安全性方面提出不同且独特的挑战。以最终用户的灵活性和硬件兼容性为目标而设计的操作系统可能很难确保数据安全性。例如,与其他移动操作系统(例如iOS)相比,通常认为更灵活的移动操作系统(例如Android™OS)最初是为给最终用户提供灵活性并与不同类型的硬件兼容而设计的,其数据安全性较差。同时,旨在提供强大数据安全性的操作系统可能需要复杂或困难的系统配置,甚至可能会影响系统性能。结果就是,用户可能会放松或禁用某些系统保护功能,以逃避复杂的系统配置和/或减轻对系统性能的影响。

## 发明内容

[0007] 下面介绍更详细的内容。该介绍无意于限制或定义任何要求保护的或尚未要求保护的发明。一个或多个发明可以存在于包括其权利要求和附图的本文的任何部分中公开的元素或处理步骤的任何组合或子组合。

[0008] 我们可以配置一个计算机系统去提供个性化的、密码学意义上安全的访问控制。该访问控制系统可以在数据访问的多个不同阶段实施密码技术。密码技术可以包括数据加密,以及建立从最终用户、应用程序、框架、内核到可信执行环境的个性化和密码安全的信任链。在类似Android操作系统的操作系统上实现并应用上述访问控制系统技术可以帮助保护应用程序数据免受已知和未知攻击,包括框架级的勒索软件、网络钓鱼攻击和/或通过滥用根特权来进行的攻击。

[0009] 该访问控制系统可以包括一个与计算机系统的各个组件交互以管理数据访问控制的内核模块。内核模块可以与框架实用程序进行交互,以从应用程序和/或用户收集属性。内核模块可以与在可信执行环境中运行的可信应用程序交互,以完成密钥管理。内核模块可以管理内核存储器中的可用于过程认证和授权的数据结构。内核模块还可以控制可用

于访问存储在计算机系统上的受保护和加密数据的文件系统操作。

[0010] 广泛地讲,这里提供了一种用于计算设备的访问控制系统,该访问控制系统包括:

[0011] (a)一个被配置为提供至少两个隔离执行环境的处理器,其中,所述至少两个隔离执行环境包括富执行环境和可信执行环境,所述处理器被配置为在所述富执行环境中运作富操作系统而在可信执行环境中运作可信操作系统;和

[0012] (b)被配置为存储多个数据文件和多个应用指令集的非易失性存储器,其中每个应用指令集对应于安装在所述计算设备上的一个应用,并且每个应用被配置为在富操作系统中运行,其中多个数据文件包括多个受保护的数据文件;

[0013] (c)其中,所述处理器被配置成为:

[0014] (i)接收对存储在非易失性存储器中的被请求数据文件的文件请求,其中,所述被请求数据文件对应于所述多个受保护的数据文件中的一个受保护的数据文件,并且其中,该文件请求是从一个运行于富操作系统内请求进程那里收到的;

[0015] (ii)确定一个与上述请求进程相关联的关联应用程序,其中,该关联应用程序对应于安装在所述计算设备上的应用程序中的一个应用程序;

[0016] (iii)确定上述关联应用程序是否存在一个验证的应用程序令牌;和

[0017] (iv)仅当存在上述验证的应用程序令牌时,才允许上述请求进程访问所述被请求数据文件,否则阻止该请求进程访问所述被请求数据文件;

[0018] 其中,上述处理器被配置为通过以下方式为所述关联应用程序生成所述验证的应用程序令牌:

[0019] 检测与所述关联应用程序相关联的第一进程的启动;

[0020] 确定在可信执行环境内有一个有效的用户码可用;和

[0021] 在确定可信执行环境内有所述有效的用户码可用后,使用该有效的用户码生成应用程序令牌。

[0022] 在任何实施例中,所述处理器可以被配置为,结合所述第一进程的启动,生成一个应用程序启动提示,其中,该应用程序启动提示被定义为提示用户输入用户码;接收一个响应于所述应用程序启动提示的应用程序启动输入;确定所述应用启动输入是否与所述有效的用户码相对应;在确定所述应用程序启动输入对应于所述有效的用户码后,使用所述用户码生成上述应用程序令牌。

[0023] 在任何实施例中,确定所述应用程序启动输入是否对应于所述有效的用户码可以包括:从所述应用程序启动输入中,确定一个接收到的用户码;通过对接收到的用户码的进行哈希计算来确定一个哈希的用户码;将该哈希的用户码与存储在可信执行环境的可信非易失性存储器中的存储的哈希的用户码进行比较;当所述哈希的用户码与所述存储的哈希的用户码匹配时,确定接收到的用户码与所述有效的用户码相对应;否则,确定接收到的用户码无效。

[0024] 在任何实施例中,所述处理器可以被配置为通过以下步骤来生成所述应用程序令牌:确定与第一进程的启动相对应的时间戳;并使用时间戳和所述有效的用户码生成所述应用程序令牌。

[0025] 在任何实施例中,所述处理器可以被配置为:将每个受保护的数据文件与一个文件特定的应用程序组相关联,其中所述文件特定的应用程序组包括至少一个安装在所述计

算设备上的应用程序;并且仅在与所述请求进程相关联的关联应用程序是与所述被请求数据文件相关联的文件特定的应用程序组中的一个应用程序时,才允许所述请求进程访问所述被请求数据文件。

[0026] 在任何实施例中,所述处理器可以被配置为:定义一个受保护的应用程序组,其中所述受保护的应用程序组包括至少安装在所述计算设备上的应用程序中的之一;并且仅在所述关联应用程序是所述受保护的应用程序之一时,才允许所述请求进程访问所述被请求数据文件。

[0027] 在任何实施例中,所述处理器可以被配置为,对于由所述受保护的应用程序之一生成的每个新文件,自动地将该文件存储为受保护的数据文件之一。

[0028] 在任何实施例中,对于安装在所述计算设备上的每个应用程序,所述处理器可以被配置为,为该应用程序分配一个唯一的应用程序标识,其中该应用程序标识是通过使用一个唯一的由所述处理器定义的应用程序识别码和与该应用程序相对应的应用程序证书来定义的,并且所述处理器可以配置为,通过使用所述关联应用程序的唯一应用程序标识和所述有效的用户码,来为所述关联应用程序生成所述应用程序令牌。

[0029] 在任何实施例中,对于安装在所述计算设备上的每个应用程序,所述处理器可以被配置为,通过以下方式定义所述应用程序标识:通过对与该应用程序对应的应用程序证书进行哈希计算,来生成一个哈希的应用程序证书;将所述应用程序识别码和所述哈希的应用程序证书组合在一起。

[0030] 在任何实施例中,所述处理器可以被配置为,在与所述关联应用程序相关联的所述第一进程启动时,验证与该关联应用程序相对应的应用程序证书;生成一个响应于验证所述应用程序证书的验证的应用程序启动提示,其中,所述验证的应用程序启动提示被定义为提示用户输入用户码;接收一个响应于所述验证的应用程序启动提示的验证的应用程序启动输入;确定所述验证的应用程序启动输入是否与所述有效的用户码相对应;并且仅在所述应用程序证书已被验证并且所述验证的应用程序启动输入对应于所述有效的用户码之后,才允许所述关联应用程序执行。

[0031] 在任何实施例中,所述处理器可以被配置为:对于安装在所述计算设备上的每个应用程序,为该应用程序分配一个唯一的应用程序标识,其中,其中该应用程序标识是通过使用一个唯一的由所述处理器定义的应用程序识别码和与该应用程序相对应的应用程序证书来定义的;定义一个受保护的应用程序组,其中所述受保护的应用程序组包括至少安装在所述计算设备上的应用程序中的之一;定义一个访问策略文件,其中,该访问策略文件包括所述受保护的应用程序组中每个受保护的应用程序的唯一应用程序标识;将上述访问策略文件存储在非易失性存储器中;并且仅在上述访问策略文件包含所述关联应用程序的唯一应用程序标识时,才允许所述请求进程访问所述被请求数据文件。

[0032] 在任何实施例中,所述处理器可以被配置为:用与对应于所述有效的用户码的哈希的码相关联的方式,将所述访问策略文件存储在非易失性存储器中;禁止对所述访问策略文件的修改,除与所述有效的用户码相对应的策略修改输入收到,并且同与所述有效的用户码相对应的所述哈希的码进行验证;并且允许在没有策略修改输入的情况下对所述访问策略文件执行读取操作。

[0033] 在任何实施例中,所述处理器可以被配置为:在富操作系统内可访问的内核内存

中定义一个访问策略数据结构;并且,将该访问策略数据结构与存储在非易失性存储器中的所述访问策略文件同步,使得所述访问策略数据结构包括所述受保护的应用程序组中每个受保护的应用程序的唯一应用程序标识。

[0034] 在任何实施例中,所述处理器可以被配置成:在与所述关联应用程序相关联的所述第一进程启动时,验证与所述关联应用程序相对应的应用程序证书;在验证应用程序证书后,通过以下方式确定所述关联应用程序是否为受保护的应用程序之一:确定所述关联应用程序的唯一应用程序标识;和确定所述关联应用程序的唯一应用程序标识是否包含在所述访问策略数据结构中;和当所述访问策略数据结构包含所述关联应用程序的唯一应用程序标识时,确定所述关联应用程序是受保护的应用程序之一;否则,确定所述关联应用程序是不受保护的应用程序;在确定所述关联应用程序是受保护的应用程序之一之后,生成一个受保护的应用程序启动提示,其中,该受保护的应用程序启动提示被定义为提示用户输入一个用户码;接收一个响应于所述受保护的应用程序启动提示的受保护的应用程序启动输入;确定所述受保护的应用程序启动输入是否与所述有效的用户码相对应;和仅在所述处理器确定所述受保护的应用程序启动输入对应于所述有效的用户码之后,才允许所述关联应用程序执行;并且在确定所述关联应用程序是不受保护的应用程序后,允许所述关联应用程序执行。所述处理器可以被配置为,在确定所述关联应用程序是不受保护的应用程序之后,即使没有用户码的情况下也允许所述关联应用程序运行。

[0035] 在任何实施例中,所述处理器可以被配置为:在所述请求进程启动时,确定与所述请求进程相关联的进程标识符;验证与所述关联应用程序相对应的应用程序证书;在验证应用程序证书后,确定所述关联应用程序是否为受保护的应用程序之一;在确定所述关联应用程序是受保护的应用程序之一时,生成一个进程启动提示,其中该进程启动提示被定义为提示用户输入一个用户码;接收一个响应于所述进程启动提示的进程启动输入;确定所述进程启动输入是否对应于所述有效的用户码;在确定所述关联应用程序是受保护的应用程序之一,并且所述进程启动输入对应于所述有效的用户码后,将所述请求进程的进程标识符存储在富操作系统内可访问的内核内存中的一个授权进程数据结构中; 并且仅在所述授权进程数据结构包含所述请求进程的进程标识符的情况下,才允许所述请求该进程访问所述被请求数据文件。

[0036] 在任何实施例中,所述处理器可以被配置为:接收来自运行于富操作系统内的特定进程的写操作请求,其中,该写操作请求与一个受保护的数据文件有关;并且仅当所述授权进程数据结构包含上述特定进程的进程标识符时,才允许所述写操作请求发生。

[0037] 在任何实施例中,所述处理器可以被配置为:检测在富操作系统中运行的行将终止进程的终止;确定所述授权进程数据结构包含所述终止进程的进程标识符;并且更新所述授权进程数据结构,以删除所述终止进程的进程标识符。

[0038] 在任何实施例中,上述处理器可以被配置为:用与对应于所述有效的用户码的哈希的码相关联的方式,将所述访问策略文件存储在非易失性存储器中;接收一个策略更新输入,该输入指定要把一个特定应用程序添加到所述受保护的应用程序组中;生成一个响应于所述策略更新输入的更新验证提示,其中,该更新验证提示被定义为提示用户输入一个用户码;接收一个响应于上述更新验证提示的更新验证输入;使用用与所述访问策略文件关联的方式存储的,与所述有效的用户码相对应的哈希的码,确定上述更新验证输入是

否与所述有效的用户码相对应;在确定上述更新验证输入对应于所述有效的用户码后,更新所述访问策略文件以包含上述特定应用程序;识别存储在非易失性存储器中的,与该特定应用程序相对应的一组现有数据文件;并且将每个现有数据文件从不受保护的数据文件修改为受保护的数据文件。

[0039] 在广义上,这里提供了一种控制对存储在计算设备的非易失性存储器中的多个数据文件的访问的方法,其中,所述多个数据文件包括多个受保护的数据文件,所述方法由一个处理器来执行,而该处理器被配置为在一个富执行环境中运行一个富操作系统并且在一个可信执行环境中运行一个可信的操作系统,该方法包括:

[0040] (a)由所述处理器接收对存储在非易失性存储器中的被请求数据文件的一个文件请求,其中,所述被请求数据文件对应于所述多个受保护的数据文件中的一个受保护的数据文件,并且其中,该文件请求是从一个运行于富操作系统内请求进程那里收到的;

[0041] (b)由所述处理器确定一个与该请求进程相关联的关联应用程序,其中,该关联应用程序对应于安装在所述计算设备上的应用程序中的一个应用程序;

[0042] (c)由所述处理器确定所述关联应用程序是否存在一个验证的应用程序令牌;和

[0043] (d)仅当存在上述验证的应用程序令牌时,所述处理器才会允许上述请求进程访问所述被请求数据文件,否则阻止该请求进程访问所述被请求数据文件;

[0044] 其中,所述处理器通过以下方式为所述关联应用程序生成所述验证的应用程序令牌:

[0045] 检测与所述关联应用程序相关联的第一进程的启动;

[0046] 确定在可信执行环境内存在一个有效的用户码可用;和

[0047] 在确定在可信执行环境内有所述有效的用户码可用后,使用该有效的用户码生成应用程序令牌。

[0048] 在任何实施例中,所述方法可以包括:由所述处理器结合所述第一进程的启动,生成一个应用程序启动提示,其中,该应用程序启动提示被定义为提示用户输入一个用户码;由所述处理器接收一个响应于所述应用程序启动提示的应用程序启动输入;由所述处理器确定所述应用程序启动输入是否与所述有效的用户码相对应;并且在确定所述应用程序启动输入对应于所述有效的用户码后,由所述处理器使用所述用户码生成所述应用程序令牌。

[0049] 在任何实施例中,确定所述应用程序启动输入是否对应于所述有效的用户码,可以包括以下步骤:从所述应用程序启动输入中,确定一个接收到的用户码;通过对接收到的用户码进行哈希计算来确定一个哈希的用户码;将该哈希的用户码与存储在可信执行环境的可信非易失性存储器中的存储的哈希的用户码进行比较;并且,当该哈希的用户码与所述存储的哈希的用户码匹配时,确定接收到的用户码对应于所述有效的用户码,否则确定接收到的用户码无效。

[0050] 在任何实施例中,所述方法可以包括通过以下步骤来生成所述应用程序令牌:确定与第一进程的启动相对应的一个时间戳;并使用该时间戳和所述有效的用户码生成所述应用程序令牌。

[0051] 在任何实施例中,所述方法可以包括:将每个受保护的数据文件与一个文件特定的应用程序组相关联,其中,所述文件特定的应用程序组包括至少一个安装在所述计算设

备上的应用程序;和仅在与所述请求进程相关联的关联应用程序是与所述被请求数据文件相关联的文件特定的应用程序组中的一个应用程序时,才允许所述请求进程访问所述被请求数据文件。

[0052] 在任何实施例中,所述方法可以包括:定义一个受保护的应用程序组,其中所述受保护的应用程序组包括至少安装在所述计算设备上的应用程序中的之一;和仅在所述关联应用程序是所述受保护的应用程序之一时,才允许所述请求进程访问所述被请求数据文件。

[0053] 在任何实施例中,所述方法可以包括:对于由所述受保护的应用程序之一生成的每个新文件,自动地将该文件存储为所述受保护的数据文件之一。

[0054] 在任何实施例中,所述方法可以包括:对于安装在所述计算设备上的每个应用程序,为该应用程序分配一个唯一的应用程序标识,其中,该应用程序标识是通过使用一个唯一的由所述处理器定义的应用程序识别码和一个与该应用程序相对应的应用程序证书来定义的;和通过使用所述关联应用程序的唯一应用程序标识和所述有效的用户码,来为所述关联应用程序生成所述应用程序令牌。

[0055] 在任何实施例中,所述方法可以包括:对于安装在所述计算设备上的每个应用程序,通过以下方式定义所述应用程序标识:通过对与该应用程序对应的应用程序证书进行哈希计算,来生成一个哈希的应用程序证书;和将所述应用程序识别码和所述哈希的应用程序证书组合在一起。

[0056] 在任何实施例中,所述方法可以包括:在与所述关联应用程序相关联的所述第一进程启动时,验证与该关联应用程序相对应的应用程序证书;生成一个响应于验证所述应用程序证书的验证的应用程序启动提示,其中,所述验证的应用程序启动提示被定义为提示用户输入一个用户码;接收一个响应于所述验证的应用程序启动提示的验证的应用程序启动输入;确定所述验证的应用程序启动输入是否与所述有效的用户码相对应;和仅在所述应用程序证书已被验证并且所述验证的应用程序启动输入对应于所述有效的用户码之后,才允许所述关联应用程序执行。

[0057] 在任何实施例中,所述方法可以包括:对于安装在所述计算设备上的每个应用程序,为该应用程序分配一个唯一的应用程序标识,其中,该应用程序标识是通过使用一个唯一的由所述处理器定义的应用程序识别码和与该应用程序相对应的应用程序证书来定义的;定义一个受保护的应用程序组,其中所述受保护的应用程序组包括至少安装在所述计算设备上的应用程序中的之一;定义一个访问策略文件,其中,该访问策略文件包括所述受保护的应用程序组中每个受保护的应用程序的唯一应用程序标识;将上述访问策略文件存储在非易失性存储器中;和仅在上述访问策略文件包含所述关联应用程序的唯一应用程序标识时,才允许所述请求进程访问所述被请求数据文件。

[0058] 在任何实施例中,所述方法可以包括:用与对应于所述有效的用户码的哈希的码相关联的方式,将所述访问策略文件存储在非易失性存储器中;禁止对所述访问策略文件进行修改,除与所述有效的用户码相对应的策略修改输入收到,并且同与所述有效的用户码相对应的所述哈希的码进行验证通;和允许在没有策略修改输入的情况下对所述访问策略文件执行读取操作。

[0059] 在任何实施例中,所述方法可以包括:在富操作系统内可访问的内核内存中定义

一个访问策略数据结构;和将该访问策略数据结构与存储在非易失性存储器中的所述访问策略文件同步,使得所述访问策略数据结构包括所述受保护的应用程序组中每个受保护的应用程序的唯一应用程序标识。

[0060] 在任何实施例中,所述方法可以包括:在与所述关联应用程序相关联的所述第一进程启动时,验证与所述关联应用程序相对应的应用程序证书;在验证应用程序证书后,通过以下方式确定所述关联应用程序是否为受保护的应用程序之一:确定所述关联应用程序的唯一应用程序标识;和确定所述关联应用程序的唯一应用程序标识是否包含在所述访问策略数据结构中;和当所述访问策略数据结构包含所述关联应用程序的唯一应用程序标识时,确定所述关联应用程序是受保护的应用程序之一;否则,确定所述关联应用程序是不受保护的应用程序;在确定所述关联应用程序是受保护的应用程序之一之后,生成一个受保护的应用程序启动提示,其中,该受保护的应用程序启动提示被定义为提示用户输入一个用户码;接收一个响应于所述受保护的应用程序启动提示的受保护的应用程序启动输入;确定所述受保护的应用程序启动输入是否与所述有效的用户码相对应;和仅在所述处理器确定所述受保护的应用程序启动输入对应于所述有效的用户码之后,才允许所述关联应用程序执行;和在确定所述关联应用程序是不受保护的应用程序后,允许所述关联应用程序执行。所述处理器可以被配置为,在确定所述关联应用程序是不受保护的应用程序之后,即使没有用户码的情况下也允许所述关联应用程序运行。

[0061] 在任何实施例中,所述方法可以包括:在所述请求进程启动时,确定与所述请求进程相关的进程标识符;验证与所述关联应用程序相对应的应用程序证书;在验证应用程序证书后,确定所述关联应用程序是否为受保护的应用程序之一;在确定所述关联应用程序是受保护的应用程序之一时,生成一个进程启动提示,其中,该进程启动提示被定义为提示用户输入一个用户码;接收一个响应于所述进程启动提示的进程启动输入;确定所述进程启动输入是否对应于所述有效的用户码;在确定所述关联应用程序是受保护的应用程序之一,并且所述进程启动输入对应于所述有效的用户码后,将所述请求进程的进程标识符存储到富操作系统内可以访问的内核内存中的一个授权进程数据结构中;和仅在所述授权进程数据结构包含所述请求进程的进程标识符的情况下,才允许所述请求进程访问所述被请求数据文件。

[0062] 在任何实施例中,该方法可以包括:从在富操作系统中运行的特定进程接收写操作请求,其中,写操作请求与受保护的数据文件有关;仅当授权的进程数据结构包括特定进程的进程标识时,才允许发生写操作请求。

[0063] 在任何实施例中,所述方法可以包括:检测在富操作系统中运行的行将终止进程的终止;确定所述授权进程数据结构包含上述行将终止进程的进程标识符;和更新所述授权进程数据结构,以删除上述终止进程的进程标识符。

[0064] 在任何实施例中,所述方法可以包括:用与对应于所述有效的用户码的哈希的码相关联的方式,将所述访问策略文件存储在非易失性存储器中;接收一个策略更新输入,该输入指定出要把一个特定应用程序添加到所述受保护的应用程序组中;生成一个响应于上述策略更新输入的更新验证提示,其中,该更新验证提示被定义为提示用户输入一个用户码;接收一个响应于所述更新验证提示的更新验证输入;使用用与所述访问策略文件相关联的方式存储的,与所述有效的用户码相对应的哈希的码,确定上述更新验证输入是否与

所述有效的用户码相对应;在确定上述更新验证输入对应于所述有效的用户码后,更新所述访问策略文件以包含上述特定应用程序;识别存储在非易失性存储器中的与该特定应用相对应的一组现有数据文件;和将每个现有数据文件从不受保护的数据文件修改为受保护的数据文件。

[0065] 在广义上,这里提供了一种计算机可读介质,其上存储有用于控制对存储在非易失性存储器中的多个数据文件的访问的非暂时性计算机可读指令,其中,所述计算设备包含一个处理器,所述处理器被配置为在富执行环境中运行富操作系统和在可信执行环境中运行可信操作系统,其中所述多个数据文件包括多个受保护的数据文件, 其中所述指令被定义为配置所述处理器做以下操作:

[0066] (a)接收对存储在非易失性存储器中的被请求数据文件的一个文件请求,其中,所述被请求数据文件对应于所述多个受保护的数据文件中一个受保护的数据文件,并且其中,该文件请求是从一个运行于富操作系统内的请求进程那里收到的;

[0067] (b)确定一个与上述请求进程相关联的关联应用程序,其中,该关联应用程序对应于安装在所述计算设备上的应用程序中的一个应用程序;

[0068] (c)确定上述关联应用程序是否存在一个验证的应用程序令牌;和

[0069] (d)仅当存在上述验证的应用程序令牌时,才允许该请求进程访问所述被请求数据文件,否则阻止该请求进程访问所述被请求数据文件;

[0070] 其中,所述指令被定义为配置上述处理器通过以下方式为所述关联应用程序生成所述验证的应用程序令牌:

[0071] 检测与所述关联应用程序相关联的第一进程的启动;

[0072] 确定在可信执行环境内存在一个有效的用户码可用;和

[0073] 在确定可信执行环境内有所述有效的用户码可用后,使用该有效的用户码生成应用程序令牌。

[0074] 在任何实施例中,所述计算机可读介质还可包括被定义为配置所述处理器以执行本文描述的任何方法的指令。

[0075] 本领域技术人员将理解,本文所公开的系统、方法或计算机可读介质可以体现本文所包含的任何一个或多个特征,并且这些特征可以以任何特定的组合或子组合形式使用。

[0076] 下面将更详细地描述各种实施例的这些和其他方面的特征。

## 附图说明

[0077] 随附的附图用于说明本说明书所教导的文献、方法和设备的各种示例,并且无意于以任何方式限制所教导的内容范围。

[0078] 图1是一个示出根据实施例的计算机系统的框图。

[0079] 图2是一个示出根据实施例可在计算机系统图1中使用的用于对进程授权的示例流程的流程图。

[0080] 图3是一个示出根据实施例可在计算机系统图1中使用的用于生成应用程序令牌的示例流程的流程图。

## 具体实施方式

[0081] 以下描述的附图是为了说明,而非限制,本文描述的实施例的各个示例的各方面和特征。为了图示的简单和清楚起见,附图中所示的元件未必按比例绘制。为了清楚起见,一些元件的尺寸可能相对于其他元件被放大。需要理解的是,为了图示的简单和清楚,在适当的情况下,为指示对应或相似的元件或步骤,参考标号可能在附图中被多次重复。

[0082] 下面为提供所要求保护的主题的实施例的示例,将描述各种系统或方法。以下描述的实施例没有限制任何要求保护的主题,并且任何要求保护的主题可以覆盖与以下描述的方法或系统不同的方法或系统。所要求保护的主题不限于具有以下所述的任何一种系统或方法的所有特征的系统或方法,也不限于以下所述的多个或所有装置或方法所共有的特征。以下描述的系统或方法有可能不是任何要求保护的主题中记载的实施例。下文描述的系统或方法中披露的,未在本文档中要求保护的任何主题可能是另一种保护性文书的主题,例如,连续专利申请。申请人、发明人或所有者无意通过本文档的公开去放弃或取消任何此类主题的权益。

[0083] 此外,需要理解的是,为了图示的简单和清楚,在认为适当的情况下,可以在附图之间重复参考标号以指示对应或相似的元件。另外,本文阐述了许多具体细节以便提供对本文描述的实施例的透彻理解。然而,本领域普通技术人员将理解,可以在没有这些具体细节的情况下实践本文描述的实施例。在其他情况下,本文对公知的方法、过程和组件没有详细描述,以免使本文所述的实施例不清楚。而且,任何描述不应被认为是限制本文描述的实施例的范围。

[0084] 术语“实施例”, “一个或多个实施例”,“一些实施例”和“一个实施例”的意思是“一个或多个”,但不表明意指全部实施例。,除非另有明确说明。

[0085] 应该注意的是,本文所用的程度术语,例如“基本上”,“大约”和“大概”,是指修饰术语的合理偏离量,使得最终结果不会显着改变。如果该程度的术语不会抵消其所修饰的术语的含义,则这些程度的术语也可以被解释为包括一个对该术语的偏离。

[0086] 此外,本文中端点的数值范围的任何列举均包括该范围内所包含的所有数字和分数(例如1至5包括1、1.5、2、2.75、3、3.90、4和5)。还应理解,所有数字及其分数均假定被术语“约”修饰,这意味着如果最终结果没有显着变化,则最多可有一定数量的变化。

[0087] 本文描述的系统和方法的示例实施例可以被实现为硬件或软件的组合。在某些情况下,可以至少部分地通过使用一个或多个计算机程序,在包括至少一个处理元件和数据存储元件(包括易失性存储器,非易失性存储器,存储元件或其任意组合)。这些设备还可以具有至少一个输入设备(例如,按钮键盘,鼠标,触摸屏等),以及至少一个输出设备(例如,显示屏,打印机,无线电等),具体取决于设备的性质。

[0088] 还应注意,可能存在一些用于实现本文描述的实施例之一的至少一部分的元件,这些元件可以通过以高级计算机编程语言(例如面向对象)编写的软件来实现。因此,程序代码可以用C,C ++或任何其他合适的编程语言编写,并且可以包括模块或类,如面向对象编程的技术人员所知。另外或者还有,可以根据需要用汇编语言、机器语言或固件来编写软件,来实现这些元件中的一些。无论哪种情况,编程语言可以是编译语言或解释语言。

[0089] 这些软件程序中的至少一些可以存储在存储介质(例如,但不限于ROM,磁盘,光盘的计算机可读介质)或通用或专用可读的设备上用途可编程设备。当由可编程设备读取时,

软件程序代码将可编程设备配置为以新的、特定的和预定义的方式操作，以便执行本文所述的方法中的至少一种。

[0090] 此外，与本文描述的实施例的系统和方法相关联的程序中的至少一些程序能够被分发在包括计算机可读介质的计算机程序产品中，该计算机可读介质承载用于一个或多个处理器的计算机可用指令。可以以各种形式提供介质，包括非临时形式，例如但不限于一个或多个磁盘，光盘，磁带，芯片以及磁性和电子存储。

[0091] 计算机程序是可以由计算机(即，由处理器)执行的一组指令。进程是程序的一个实例，即计算机内存中已准备好由计算机的中央处理器(CPU)执行的程序的副本。在下面的讨论中，参考计算机系统的处理器和由计算机系统的处理器执行的操作。应当理解，这样的参考包括一个或多个处理元件以及使用一个或多个处理元件来执行操作，例如一个或多个CPU内的一个或多个处理核心。

[0092] 操作系统(OS)是控制计算机可用硬件和软件资源使用的软件(包括多个程序和库)。操作系统的核心组件(内核)是管理所有计算机硬件设备的程序。在运行时，从内核实例化的内核进程还管理从除了内核之外的程序(例如，由用户启动的程序)实例化的进程，并为那些非内核进程访问硬件设备提供统一的接口(称为系统调用)。

[0093] 处理器可以被配置为区分来自内核的指令和来自除内核之外的程序的指令，并以单独的模式(即，内核模式和用户模式)执行指令。在操作系统内从除内核之外的程序(即，内核外部)实例化的进程在本文中可以被称为用户模式进程。相应的程序或应用程序可以称为用户模式应用程序。

[0094] 在内核内，文件系统是必不可少的模块，其向用户模式进程提供对诸如磁盘，闪存等非易失性存储的访问。从用户模式进程的角度来看，该进程访问并运行(例如，打开通过向OS内核发出系统调用，在非易失性存储上进行读 /写 /关闭文件等操作)。收到这些系统调用后，OS内核将管理请求进程是否被允许以及如何访问请求的文件。OS内核可以根据有关发起者(请求进程)和目标文件的信息，通过仲裁是否允许请求进程(发出系统调用的进程)来访问目标文件或目录，从而执行访问控制，使用文件系统模块如果允许访问，则执行系统调用，并通过系统调用的返回值通知用户模式进程。

[0095] 通常，数据文件用于存储信息。当信息以直接可读/可理解的方式存储时(即，没有被混淆或以其他方式编码以防止对信息的直接理解)，则数据文件可以被称为明文。在某些情况下，可以修改(即加密)数据文件以防止对由该数据文件存储的明文信息的未授权访问。

[0096] 加密是使用秘密(加密密钥)将信息(即，明文)转换为混淆形式(可以称为密文)的过程。解密是加密的反向过程，并使用秘密(例如，取决于使用的加密方法的加密密钥或不同的解密密钥)将密文转换为明文。

[0097] 在某些情况下，文件系统(例如ext4文件系统和NTFS文件系统)可能支持对存储在计算机系统中的数据文件的文件系统级加密。在文件系统级加密中，单个数据文件中数据的加密/解密可以由文件系统执行。该系统可以基于个人(即，基于每个文件)定义与所存储的文件有关的加密策略(包括文件是否已经被加密或者应该被加密以及加密密钥如何被导出)。当创建或存储文件时，文件系统可以采用加密策略来确定是否应在存储之前对文件进行加密，然后文件系统可以根据需要对数据文件进行加密和存储。当访问文件时，文件系统

可以使用加密策略并对数据文件进行解密,然后再将解密后的数据文件提供给请求进程。因为加密/解密是由文件系统执行的,所以文件系统层级的加密对于用户模式进程是透明的,用户模式进程正常访问文件并通过系统调用对文件执行操作。

[0098] 许多操作系统涉及最终用户的可用性与数据安全性之间的折衷。通常认为,为最终用户提供灵活性和兼容性而设计的操作系统缺乏数据安全性。结果,私人或机密数据可能会面临被披露或截获的风险。

[0099] 设计为提供增强的数据安全性的操作系统通常被认为是不灵活、复杂和/或降低了系统性能。这可能会阻止广泛采用和/或导致用户禁用安全功能,以提高可用性或性能。结果就是,私密或机密数据可能仍会暴露。

[0100] Android是主要设计用于与触摸屏移动设备(例如智能手机和平板电脑)一起使用的移动操作系统。Android操作系统基于被修改的Linux内核。

[0101] Android是操作系统的示例,通常被认为以牺牲数据安全性为代价提供灵活性和硬件兼容性。为了提高Android操作系统的安全性,人们提出了许多不同的方法,例如沙箱,屏幕短码锁和智能锁,基于文件的加密以及诸如SELinux等强制性访问控制。此外,鼓励用户采用最佳做法,例如及时更新和执行用户权限控制。但是,这些技术对于某些类型的攻击(例如对根特权的滥用探索)可能仍然无效。

[0102] 沙箱和强制访问控制是可用于防止对受保护数据进行未经授权访问的技术。在强制访问控制过程中,每个用户模式进程都可以标记有关联的进程标签,每个文件都可以有一个关联的文件标签。可以预定义访问策略并将其加载到内核内存中。

[0103] 访问策略可以指定具有特定进程标签的用户模式进程是否可以访问具有特定文件标签的文件。在某些情况下,访问策略可以包括针对每个进程标签和文件标签的访问标准(例如,针对进程标签和文件标签的每种可能的组合)。当用户模式进程向文件发出操作请求时,内核可以根据预定义的策略检查用户模式进程的标签和文件的标签。然后,内核可以相应地授予或拒绝该操作请求。

[0104] 沙箱可以限制在计算系统上运行的特定应用程序可访问的资源集。通过在安装时为每个应用程序创建唯一的Linux UID,可以基于自由访问控制来定义沙箱。即使在Android 4.3之前,此过程也已用于Android操作系统的实现中。

[0105] 移动操作系统也可以被配置为实现基于文件的加密方法。基于文件的加密可保护静态数据免遭破坏。通过加密,可以将明文数据文件编码为无法识别的模糊形式,除非使用相应的解密密钥解密。自Android 7.0起,Android操作系统已包含基于文件的加密。

[0106] 增强安全性的Linux(SELinux)对Linux内核提供了一组修改,以促进强制性访问控制策略的实施。SELinux还可以增强操作系统中沙箱的操作。自Android版本4.3起,SELinux已包含在Android操作系统中。

[0107] SELinux涉及使用已定义的访问策略来控制数据访问。访问策略往往依赖于请求访问数据的进程的应用程序识别码(UID或AppID)和进程ID(PID)。从理论上讲,SELinux是提供高级别数据安全性的强大工具。但是,实现SELinux的操作系统仍然容易受到root特权的滥用的攻击。

[0108] 一个给定的应用程序和进程的应用程序识别码和进程标识是由操作系统生成的,这个生成过程中没有来自用户或外部系统的任何输入。结果就是,root用户(例如,当富操

作系统为root时)可以修改应用程序识别码。例如,root用户或具有root特权的进程可能会禁用SELinux。超级用户也可以利用系统调用和/或系统实用程序来修改现有的SELinux策略和/或创建新的SELinux策略。这些操作中的任何一项都可能危害数据安全性。

[0109] SELinux还因其复杂性和配置难度以及对系统性能的干扰而臭名昭著。实际上,这通常会导致SELinux保护的放松或禁用,以避免复杂的配置或减少对系统性能的干扰。结果,可能会损害SELinux提供的数据安全性。

[0110] SELinux还通常在其配置策略中缺乏精细的粒度,并且通常不能按需和/或即时配置访问策略。SELinux可能也无法为其分配给数据、应用程序、进程的标签提供足够的保护。另外,分配给计算设备中的数据、应用程序和进程的标签通常与最终用户完全断开连接(例如,标签的生成和分配是在没有任何用户输入的情况下进行的)。

[0111] 尽管上述数据安全技术为用户数据和应用程序提供了某种保护,但它们通常仍然容易受到复杂攻击的攻击。在eSecurityPlanet发布的2017年报告中,接受调查的安全专家中有64%质疑他们防止移动数据泄露的组织级能力。该报告还发现,超过三分之一的公司未能充分保护移动设备的安全。因此,对于许多组织而言,移动数据安全性仍然是一个严重而切实的问题。开发可以解决上述技术的某些不足的新数据安全方法可以帮助提高组织级数据安全性,并有助于防止移动数据泄露。

[0112] 本文描述的实施例提供了可以控制对存储在计算机系统上的数据的访问的系统、方法和计算机程序产品。本文描述的实施例可以控制应用程序或进程是否有权访问存储在计算机系统中的某些数据文件(在本文中称为受保护的数据文件)。

[0113] 计算机系统可以被认为具有一系列或一堆的概念层级。例如,与Android操作系统一起运行的移动计算系统可以包括概念层级,例如最终用户层级,应用程序层级,框架层级和内核层级。在某些情况下,计算系统可以包括其他概念层级,例如可信执行环境(TEE)。

[0114] 本文描述的实施例可以定义跨计算系统的各个概念层级运行的访问控制系统和方法。所述访问控制系统和方法可以配置为,建立跨计算系统的概念层级的安全信任链,例如覆盖从最终用户到应用程序,框架和内核层级,再到可信执行环境。

[0115] 本文描述的实施例可以把包括数据加密在内的密码技术应用到数据访问的各种不同阶段。本文描述的实施例可以提供个性化和密码学意义上安全的访问控制(PACSAC),以在整个计算机系统中(例如,跨越所有概念性系统层级)建立个性化和密码安全的信任链。

[0116] 本文所述的访问控制系统和方法可以被实现为保护诸如Android操作系统之类的操作系统中的应用程序数据。对于每个要保护的应用程序,可以使用用户密码(例如,用户码),来密码保护用于标识特定应用程序的应用程序标识的创建。该应用程序标识还可以用于标识与相应应用程序关联的进程和/或数据/文件。用户秘密也可以用于为给定应用程序生成文件访问数据(例如密钥材料)。

[0117] 所述应用程序标识和文件访问数据可以用于对与给定应用程序相对应的进程进行认证和授权。所述访问控制系统可以配置为,仅在最终用户(或至少通过使用用户码)为受保护的数据文件对进程验证并授权后,才允许该进程访问受保护的数据文件。所述验证和授权过程可以使用用户密码进行加密保护。这样,对于每个受保护的应用程序,相应的受保护的数据可以得以保护,以免遭各种尝试的入侵,例如对根特权的滥用,勒索软件攻击和

数据泄露。

[0118] 本文描述的实施例可以克服一些现有数据安全技术,例如SELinux,的缺点。SELinux依靠应用程序识别码(UID或AppID)和进程ID以及访问策略来提供访问控制。在SELinux中,应用程序识别码和进程ID是由计算系统生成的,而无需来自最终用户的输入,也无需使用用户密码。结果,应用程序识别码可以被root用户修改(例如,当富操作系统为root时。)本文所述的实施例可以通过在进程将要运行时进一步对进程进行密码认证来帮助抵抗此类根特权滥用攻击 。例如,可以基于一个有效的用户码来认证该进程。该有效的用户码可以是从用户接收的,或已经存储在计算系统中(例如存储在可信执行环境内)的存储的用户码,或者甚至存储在于计算系统外部的一个代码。

[0119] 例如,用于解锁电话的PIN码或密码(一旦从用户接收到)可以用作用户码。在某些情况下,计算系统可以提示用户提供用户码以执行各种认证和/或授权过程(例如,当启动进程时)。替代地或附加地,用户码可以在最初被接收之后被存储在系统中,并且可以被重新用于随后的授权和/或认证过程(例如,当后续进程被启动时)。

[0120] 替代地,一个用户码可以被存储在设备中,而无需要求来自最终用户的直接输入。例如,预定义的用户码可以与预先安装在计算设备上的应用程序相关联。可以将预定义的用户码存储在计算系统上,例如存储在可信执行环境的非易失性存储器中。如此,可以使用预定义的用户码来保护、认证和授权所述预安装的应用程序。

[0121] 系统架构

[0122] 以下描述的计算机系统体系结构可以在任何访问控制系统或方法中独自使用,或者与其他特征或披露的特征一起进行任何组合或子组合使用。所述特征包括:用户码、应用程序标识,应用程序令牌,文件访问数据,将应用程序添加到受保护区域的方法,对进程进行身份验证的方法,授权进程访问数据文件的方法,受保护的数据文件,启动应用程序的方法,生成新的受保护的数据文件的方法,以及访问受保护的数据文件的方法。

[0123] 图1示出了可以根据本文描述的实施例使用的计算机系统100的示例。图1所示的计算机系统可以是任何类型的计算机系统,例如移动设备(例如,智能电话或平板电脑),膝上型计算机,台式计算机,服务器等。图1所示的计算机系统100是一个可用来提供个性化密码安全访问控制的计算机系统的示例。

[0124] 在图1所示的示例中,计算机系统100包括多个系统层级。如图所示,系统层级包括最终用户层级110,应用程序层级120,和操作系统层级。操作系统层级可以包括多个操作系统层级,例如操作系统框架130,内核140和可信执行环境(TEE)150。

[0125] 通常,计算机系统100可以包括处理器(例如,具有一个或多个CPU内核的微处理器)和存储。处理器可以通过计算机数据总线和存储耦合关联。

[0126] 存储器可以包括易失性存储器和诸如非易失性存储器109之类的非易失性存储器。非易失性存储器存储由计算机可执行指令组成的计算机程序,其可以被按需加载到易失性存储器中以便由处理器执行。本领域技术人员将理解的是,本文中将计算机系统100称为执行某功能或以某特定方式进行动作意味着处理器的一个或多个核正在执行存储在存储器中的指令(例如,软件程序),并可能通过一个或多个接口发送或接收输入和输出。

[0127] 例如,非易失性存储器109可以存储多个应用程序指令集。每个应用程序指令集可以对应于安装在计算设备100上的应用程序102之一。应用程序指令集可以定义被配置为在

富操作系统内运行的应用程序102的操作。

[0128] 存储器还可以在执行计算机可执行指令的过程中存储进出处理器的数据。例如，非易失性存储存储器109可以被配置为存储多个数据文件。

[0129] 在内核140内操作的文件系统模块108可以被配置为，使用数据库结构将数据文件存储在非易失性存储器109中。例如，可以使用关系数据库或非关系数据库，例如一个键-值对数据库，NoSQL数据库等来存储数据文件。

[0130] 多个数据文件可以包括多个受保护的数据文件。可以将计算机系统配置为控制对受保护的数据文件的访问。例如，对受保护的数据文件的访问可能仅限于为受保护的数据文件授权的进程。

[0131] 多个数据文件可以包括多个加密的数据文件。例如，某些或所有受保护的数据文件可以存储为加密的数据文件。在某些情况下，其他数据文件也可以存储为加密数据文件。例如，在某些系统中，存储在非易失性存储器109上的所有数据文件可以被加密。

[0132] 文件标签可以与为其分配了数据文件相关联地存储在非易失性存储器109中。例如，文件标签可以由文件系统当成文件属性来存储。然后，在访问关联的数据文件时，可以将文件属性读入内核内存。

[0133] 计算机系统100可以实施访问控制过程以保护受保护的数据文件免于未经授权的访问。例如，计算机系统100可以被配置为，当验证的应用程序令牌当前，对特定应用程序，不存在时，阻止特定应用程序访问相关联的受保护的文件。仅当关联应用程序存在验证的应用程序令牌时，才允许与该应用程序相关联的进程访问受保护的数据文件。

[0134] 所述处理器可以被配置为提供至少两个隔离的执行环境。隔离的执行环境可以包括富执行环境(REE)和可信执行环境(TEE)150。富执行环境可以支持计算设备操作系统的各个组件，例如框架130和内核140。所述处理器可以定义REE和可信环境150以彼此隔离地操作，使得运行于REE中的组件不能访问加载到可信环境150中的代码和数据。 可信环境150可以包含在处理器的更安全的部分内，从而为可信环境150提供更高的安全性。计算机内存可以分为两部分，安全内存和非安全内存。安全存储器只能由可信环境150访问。

[0135] 在一些示例中，处理器可以被配置为在REE和TEE 102中提供单独的操作系统。处理器可以在REE中操作该富操作系统。可以在富执行环境中运行的富操作系统的示例包括有Android™操作系统、Linux操作系统、Microsoft Windows™操作系统等。富操作系统可以包括内核140，该内核有助于帮助在富操作系统 上运行的和应用程序102相关联的进程来访问计算机系统100的资源。应用程序可以调用内核140，以访问非易失性存储存储器109中存储的文件并对其执行操作。在正常操作期间，用户101与计算机系统100只能与富操作系统直接交互。

[0136] 处理器还可以被配置为在可信执行环境150中运行可信操作系统。可信操作系统可以被加载到可信执行环境150中并在可信执行环境150中运行。例如，使用Android富操作系统的计算设备可以被配置为在可信执行环境内实现可信安全操作系统。

[0137] 通常，可信操作系统可以由根据定义的一组操作标准而配置的操作系统来实现，该操作标准具有比富操作系统更高的安全级别。例如，可信OS可能比功能丰富的OS具有更少的容量，但具有更高的安全性。可以根据诸如信息技术安全性评估的通用标准(例如ISO / IEC 15408)和指定的安全功能要求之类的标准来定义可信操作系统。

[0138] 例如,可信操作系统可以配置有“安全启动序列”。在安全启动序列中,可将可信操作系统的指令划分为几个(即多个)不同的操作系统指令。每个操作系统指令都可以进行数字签名。引导顺序中需要加载到计算机内存中的第一条指令可以存储在只读存储器(ROM)中。每个指令段可以包括指令,以验证要按照引导顺序加载到计算机存储器中的后续指令的数字签名。可选地,引导序列可能需要验证每个指令的数字签名,以便完成引导序列。例如,如果无法验证指令之一的数字签名,则可以中止引导序列。

[0139] 如图所示,计算机系统100还可以包括在可信执行环境150内的可信操作系统上运行的可信应用程序115。在可信OS上运行的可信进程可以称为可信应用程序(TA)115。这些可信应用程序可以提供额外的安全性和验证功能,以确保可信OS的更高安全性。例如,可以用可信应用程序供应商的公共密钥对可信应用程序115进行数字签名,使得入侵可信应用程序变得更加复杂。

[0140] 如图1所示,内核140可以包括内核模块106。内核模块106可以与可信应用程序115通信以便处理敏感信息。内核140还可以包括任务列表107。任务列表107可以提供在计算机系统100上运行的进程的列表。内核模块106还可以基于从任务列表107获得的进程信息来控制文件系统108的操作。

[0141] 特别地,内核140以及在内核140内操作的内核模块106和文件系统模块108可以使应用程序102能够访问非易失性存储装置109。内核模块106和文件系统模块108可以使能应用程序102去执行与存储在非易失性存储器109中的文件有关的操作。

[0142] 对应于应用程序102的进程可以将系统调用传输到内核140,以便执行与非易失性存储109中的数据文件有关的操作。文件系统108可以被配置为一些系统调用实现。文件系统108可以向进程提供对存储的数据文件的访问。文件系统108还可以执行操作来存储由进程创建或修改的附加数据文件。

[0143] 内核模块106可以被配置为拦截来自与应用程序102相对应的进程的系统调用。当进程向富操作系统内核140提交请求(或系统调用)时,内核模块106可以在请求到达文件系统108之前将其拦截。

[0144] 内核模块106可以被配置为监视和拦截从与在富操作系统内运行的应用程序102相对应的用户模式进程接收到的数据文件请求。在某些情况下,内核模块106可以在请求到达文件系统108之前拦截来自与应用程序102相对应的进程的每个文件请求。

[0145] 可信应用程序115可以被配置为执行与发布给富操作系统的内核140的文件请求有关的访问控制操作。可信应用程序115可以与内核模块106通信。内核模块106和可信应用程序115可以管理对存储在非易失性存储器109中的受保护的数据文件的访问。

[0146] 可信应用程序115在可信执行环境150中操作,以响应于内核模块106接收到的数据请求来管理对数据文件的访问。在从与应用程序102之一相对应的进程接收到请求后,内核模块106可以(例如通过安全信道)与可信应用115通信。内核模块106可以将与所拦截的请求相对应的文件请求数据发送到可信应用115。

[0147] 在某些情况下,可信应用程序115可以评估文件请求数据,以确定是否应允许该请求。可信应用程序115可以向内核模块106提供一个指示是否满足上述请求的响应。

[0148] 可信应用程序115还可以向内核模块106提供用于完成请求所需的数据,例如解密加密文件所需的密钥。然后,内核模块106可以基于从可信应用程序115接收的数据与请求

进程和文件系统108进行交互。

[0149] 可信应用程序115和内核模块106可以安全地通信,以通过在富操作系统中运行的进程来管理数据访问。例如,可信应用程序115可以提供可用于为与运行进程相关联的应用程序生成应用程序令牌的数据。可信应用程序115还可以,例如说通过验证应用程序令牌,向内核模块106提供关于当前可用的应用程序令牌的反馈。

[0150] 可信应用程序115可以通过监视是否向计算机系统100提供了有效的用户码来协调有效应用程序令牌的生成。例如,在确定可信执行环境150内有有效的用户码可用之后,才可以生成应用程序令牌。内核模块106可以与可信应用程序115通信,以确认所述有效的用户码是否已被提供。

[0151] 一个有效应用程序令牌可以通过使用所述有效的用户码或其某种修改形式来生成。这可以让可信应用程序115在随后验证应用程序令牌,例如说基于存储在可信执行环境150中的有效的用户码(或其修改形式)。

[0152] 每个加密文件可以具有关联的文件访问数据(也称为密钥属性或密钥材料)。与特定加密文件相对应的文件访问数据可以被在可信执行环境150中运行的可信应用程序115用来规范对该加密文件的访问。

[0153] 如图1所示,应用程序102可以通过框架层130与内核140交互。框架层130可以包括各种程序,例如代码库、编译器以及用来协调应用程序102在特定操作系统内操作的其他程序。

[0154] 可以在框架层130中提供控制门104。控制门104可以协调对尝试通过对内核140的调用访问存储在非易失性数据存储器109中的文件的应用程序102进行认证和授权。

[0155] 在展示的实例中,应用程序层级120可包含一个驱动伴生应用程序103。驱动伴生应用程序103可协调用户101与内核140之间的通信。例如,驱动伴生应用程序103可与由框架层130提供的注册服务105进行通信。用户101可以与注册服务105交互(通过驱动伴生应用程序103)以指定一个或多个应用程序102是受保护的应用程序,例如通过将应用程序102A和102B注册到内核模块106存储的访问策略中。

[0156] 计算机系统100的配置可以提供从用户101到应用程序102A,102B,到框架130,到内核140,到可信执行环境150,最后到存储在非易失性存储器109上的应用程序数据的信任链。因此,计算机系统100可以在整个计算机系统100的各层中向用户提供个性化和密码安全的访问控制。

[0157] 计算机系统100还可以包括图1中未示出的各种其他组件。例如,计算机系统100可以包括一个或多个通信接口,输入设备和输出设备。例如,输出设备可以包括合适的显示器,用于根据各种计算机程序的需要输出信息和数据。特别地,显示器可以显示丰富操作系统的图形用户界面(GUI)。还可以提供各种其他类型的输出设备,例如扬声器和/或触觉反馈传感器。

[0158] 输入设备可以使用户能够与富操作系统进行交互。例如,计算机系统100可以包括一个或多个输入设备,诸如键盘,鼠标,触控板和各种其他输入设备。

[0159] 还应理解,可以使用硬件和软件资源的组合来实现计算机系统100的某些元件,例如存储器和/或处理器中的一些或全部,例如使用虚拟机和/或或容器。

[0160] 可以提供诸如一个或多个有线或无线数据网络接口的通信接口,以使得能够与外

部系统或设备进行通信和/或通过网络进行通信,例如通用串行总线,Bluetooth TM或以太网连接。计算机系统100有时可以经由互联网连接到外部计算机或服务器。例如,计算机系统100可以连接到软件更新服务器以获得软件应用程序或固件的最新版本。

[0161] 用户码

[0162] 以下描述的用户码可以在任何访问控制系统或方法中独自使用,或者与其他特征或披露的特征一起进行任何组合或子组合使用。所述特征包括:系统体系结构,应用程序标识,应用程序令牌,文件访问数据,将应用程序添加到受保护区域的方法,对进程进行认证的方法,授权进程访问数据文件的方法,受保护的数据文件,启动应用程序的方法,生成新的受保护数据文件的方法,以及访问受保护的数据文件的方法。在本文中,用户码也可以被称为Qcode。

[0163] 在这里的实施例中,访问控制系统和方法可以依赖于用户码来控制对受保护的数据文件的访问。用户码可以形成用户101与计算机系统100的组件(例如应用程序102,内核140和可信执行环境150)之间的信任链的基础。

[0164] 计算机系统100可以被配置为防止访问受保护的数据文件,除非在计算机系统100内有有效的用户码可用或已经使之可用。例如,对受保护的数据文件的访问可被阻止,除非用户码在可信执行环境中可用(尽管在某些情况下,用户码只能以被修改和/或加密的形式存储,例如用户码的哈希版本)。即使有效的用户码可用,对受保护的数据文件的访问仍可能仅限于一组授权的应用程序。

[0165] 在一些示例中,可以由最终用户101将用户码提供给计算机系统100。最终用户101可以通过以各种输入形式来输入用户码。例如,用户101可以直接使用诸如鼠标,键盘或触摸屏之类的输入设备向计算机系统100提供用户码输入,该用户码输入定义了用户码(Qcode)。替代地或附加地,用户101可以间接地提供Qcode。例如,用户可以提供可以由计算机系统100测量的生物特征用户码输入,例如指纹或面部图像。生物特征输入可以用于导出用户码(Qcode)。

[0166] 计算机系统100可以存储用户码数据,该用户码数据可用于确定由用户提供的(例如,由用户键入的密码,指纹,面部图像等)声称的访问代码(即,用户码输入)是否对应到真实的用户码。在某些情况下,用户码可以直接存储在非易失性存储器109中。例如,用户码可以以加密或其他方式隐藏的形式(例如,用户码的哈希版本)存储。然后,计算机系统100可以将从用户101接收的声称的用户码与用户码的存储版本进行比较。

[0167] 替代地或附加地,用户码可以不直接存储在非易失性存储器109中。在某些情况下,计算机系统100可以被配置为从来都不将用户码直接存储在非易失性存储器109中。例如,在最终用户101提供间接定义所声称的用户码的用户码输入的情况下(例如,生物特征输入),可以将代码输入的修改版存储在非易失性存储器109中。计算机系统100通过使用代码输入修改器来修改用户码输入从而来确定所声称的用户码。例如,代码输入修改器可以定义用户码和用户码输入之间的差异。

[0168] 一旦用户101提供了用户码输入,计算机系统100就可以确定用户码。例如,计算机系统100可以通过将代码输入修改器应用于一个生物特征输入来确定用户码。在没有用户码输入的情况下,从代码输入修改版确定用户码可能仅仅只能是随机猜测。

[0169] 在一些示例中,用户码和/或代码输入修改版可以被存储在非易失性存储器109内

可被富操作系统访问到的部分。或者,用户码和/或代码输入修改版可以存储在仅可信执行环境150可访问的安全存储器中。即,仅可信操作系统和/或可信应用115可以访问存储有用户码和/或代码输入修改版的存储空间。

[0170] 当用户101输入包括所声称的用户码的用户码输入时,计算机系统100可以将所声称的用户码提供给可信执行环境150,以使用存储在可信执行环境150中的用户码的变化版本进行验证。用户101因此可以在计算机系统100内提供信任链的一个锚点,而存储在可信执行环境150中的用户码的变化版本可以提供信任链的另一锚点。

[0171] 在一些示例中,一个模糊形式的用户码可以存储在可信执行环境150中。例如,可信执行环境150可以存储用户码的哈希版本。可以使用相同的哈希函数对声称的用户码进行哈希处理,以生成哈希的声称的用户码。可以将哈希的声称的用户码与可信环境150中存储的用户码的哈希的版本进行比较,以验证所声称的用户码是有效的用户码。

[0172] 在计算机系统100的操作过程中的一个或多个时间点,计算机系统100可以为用户101生成用户码提示。用户码提示可以向用户指示应该输入用户码。然后,计算机系统100可以基于从用户101接收到的输入来确定用户码。在某些情况下,可以使用用于解锁手机的PIN码或密码(一旦从用户接收到)作为用户码。替代地或附加地,当应用程序和/或进程启动时,计算系统100可以提示用户提供用户码。

[0173] 计算机系统100可以基于从用户101接收到的用户码输入来确定接收到的用户码。计算机系统100然后可以通过对接收到的用户码进行哈希处理来确定哈希的用户码。可以将哈希的用户码与存储在可信执行环境150中的可信非易失性存储器内的存储的哈希的用户码进行比较。如果哈希的用户码与存储的哈希的用户码匹配,则计算机系统100可以确定接收到的用户码对应于有效的用户码。否则,计算机系统100可以确定所接收的用户码无效。

[0174] 在本文描述的实施例中,用户码可以用于生成各种其他类型的数据,例如应用程序令牌和/或文件访问数据。在一些示例中,用户码可以直接用于这样的计算中。备选地,用户码的哈希版本可以用来进一步保护该有效的用户码,防止它被确定。

[0175] 应用程序标识

[0176] 以下描述的应用程序标识可以在任何访问控制系统或方法中独自使用,或者与其他特征或披露的特征一起进行任何组合或子组合使用。所述特征包括:系统体系结构,用户码,应用程序令牌,文件访问数据,将应用程序添加到受保护区域的方法,对进出进行认证的方法,授权仅程访问数据文件的方法,受保护的数据文件,启动应用程序的方法,生成新的受保护数据文件的方法,以及访问受保护数据文件的方法。本文中,应用程序标识也可以被称为appID。

[0177] 如上所述,计算机系统100可以为在计算机系统100中运行或安装的每个应用程序102定义一个唯一的应用程序识别码(例如,由操作系统生成的UID或AppID)。操作系统可以为每个应用程序分配其自己的应用程序识别码。可以在没有用户输入的情况下生成应用程序识别码。

[0178] 每个应用程序可以仅被分配一个应用程序识别码。可以将每个应用程序的应用程序识别码在系统100上安装后分配给该应用程序102(即,对于每个应用程序,当该应用程序安装在系统100上时,可以将应用程序识别码分配给该应用程序)。除非直到该应用程序被

卸载,否则特定应用程序102的应用程序识别码可以保持不变。

[0179] 在某些操作系统(例如,膝上型计算机,台式计算机和服务器计算机的操作系统)中,UID可以定义用户标识代码或用户ID。可以为计算机的每个用户分配一个UID。UID可用于指示哪个用户已启动特定程序/应用程序(以及相关进程)。

[0180] 或者,在包括诸如Android之类的移动操作系统的其他操作系统中,可以不同地实现UID的概念。可以为每个应用程序分配其自己的UID(即应用程序识别码)。例如,操作系统可以默认为每个应用程序分配一个UID。

[0181] 例如,对于运行Android操作系统的设备,可以将设备上安装的每个Android软件包文件分配其自己的统一的Linux用户ID(上述的UID /应用程序识别码)。应用程序识别码通常是在将应用程序安装到设备中时分配的。在卸载该应用程序之前,该应用程序的应用程序识别码可以在设备中永久保留。操作系统可以为该应用程序定义沙箱。可以将沙箱配置为防止其他应用程序影响该特定应用程序。

[0182] 在本文描述的实施例中,计算机系统100可以被配置为,为在计算机系统100上操作或安装的每个应用程序102(以及存储在非易失性存储器109中的相应应用程序指令集)生成应用程序标识(appID)。可以为每个应用程序102(和相应的应用程序指令集)分配一个唯一的应用程序标识。除了操作系统定义的应用程序识别码之外,每个应用程序102的应用程序标识可以被定义。

[0183] 每个应用程序标识可以为相应的应用程序及其相关进程提供唯一且持久的身份。该应用程序标识可以用于认证和授权在计算机系统100中运行的进程。应用程序标识还可以用于提供访问控制系统和方法。例如,应用程序标识可以用于指定允许哪些应用程序访问存储在存储器109中的受保护的数据文件。每个受保护的数据文件可以与一个或多个被允许访问该受保护的数据文件的应用程序的应用程序标识相关联。

[0184] 对于每个应用程序102,可以使用为该应用程序102生成的应用程序识别码来定义应用程序标识。如上所述,对于操作系统中的每个应用程序,应用程序识别码可以是唯一的。但是,在卸载应用程序后,可以重新使用上述应用程序识别码。结果,应用程序识别码可能不会在设备中存储的数据的整个生命周期中保持不变。即,可以在设备上的数据的使用期限内将应用程序识别码分配给不同的应用程序(即使特定的应用程序识别码只能在给定的时间分配给一个应用程序)。因此,在仅仅只依赖于应用程序识别码的访问控制系统中,新应用程序可能会意外继承先前安装的应用程序的数据特权。

[0185] 通常,应用程序识别码的分配是由root特权管理的。同样,UID验证由root特权管理。结果,仅依赖于应用程序识别码的访问控制系统可能会遭受根特权滥用的攻击。

[0186] 在此描述的实施例可以使用附加的识别特征来为每个应用生成应用程序标识。例如,可以使用与特定应用相对应的应用程序证书来生成应用程序标识。应用程序证书可以是由特定应用程序的开发者定义的数据(例如,证书字符串)。在某些情况下,可以为来自同一开发人员的多个应用程序定义相同的应用程序证书。或者,可以为单个应用程序定义应用程序证书。在某些情况下,可以为特定版本的应用程序定义应用程序证书。或者,对于不同版本或应用程序更新,应用程序证书可以保持不变。使用应用程序证书可以确保应用程序标识专用于该应用程序,或者至少特定于该应用程序的开发人员。

[0187] 在某些操作系统中,例如Android,可以使用相应的应用程序证书对每个应用程序

包进行数字签名。这可能有助于防止篡改应用程序包。对于发布包,发布包的开发人员可以指定使用特定证书对发布包进行签名。证书可以由开发人员针对其组织开发的应用程序创建。然后,当生成应用程序标识时,计算机系统100可以使用该证书。

[0188] 在一些示例中,如果未为应用指定特定证书,则计算机系统100可以指定可用于生成应用程序标识的系统证书。例如,可以在计算机系统上存储默认系统证书(例如,参见C:\Users \ admin.android \ debug.keystore)。

[0189] 证书文件可以包含一对公钥和私钥。私钥可用于对应用程序包进行签名。在安装期间,系统可以使用与证书中的私钥相对应的公钥来验证应用程序包是否已被修改。如果证明该应用程序包未修改,则可以安装该应用程序包。如果对已签名的应用程序包的评估指示该应用程序包已被修改,则计算机系统100可能会阻止应用程序包的安装。

[0190] 可以使用应用程序识别码和应用程序证书来定义应用程序标识。基于应用程序识别码的应用程序标识数据可以与基于应用程序证书的证书数据组合以生成应用程序标识。例如,可以通过将应用程序标识数据和证书数据进行级联来定义应用程序标识。

[0191] 在一些示例中,证书数据可以包括一个应用程序证书的哈希。可以通过对与该应用程序相对应的应用程序证书进行哈希处理来生成哈希的应用程序证书。可以通过组合应用程序识别码和哈希的应用程序证书来生成应用程序标识。

[0192] 例如,可以通过将应用程序识别码和哈希的应用程序证书连接为以下内容来定义应用程序的应用程序标识(appID):

[0193] UID | sha256hash(应用程序证书),

[0194] 其中,UID表示应用程序识别码(例如,将64位整数分配给应用程序),“ |”表示用于将两个字符串连接为一个的连接操作(例如,x | y表示将x和y连接在一起),证书可以是字符串,sha256hash()是使用sha256算法的示例哈希函数。

[0195] 可以用各种形式定义应用程序标识。例如,应用程序标识可以定义为字节缓冲区。字节缓冲器可以包括与应用识别码相对应的部分和与应用程序证书相对应的部分。例如,字节缓冲器可以被设置为长度为40。字节缓冲器可以包括与应用识别码(UID)相对应的8字节部分和与应用程序证书(例如,证书的哈希版本)相对应的32字节部分。

[0196] 在某些情况下,应用程序标识可以以加密形式存储在非易失性存储器109中。例如,可以使用存储在可信执行环境150中的加密密钥来加密应用程序标识。

[0197] 可以基于用户码定义应用程序标识加密密钥。因此,可能需要来自用户的输入来定义应用程序标识。类似地,可能需要用户输入来评估现有的应用程序标识。这可以确保由系统100提供的访问控制对于用户101是个性化的。

[0198] 应用程序标识可以被诸如应用程序102A和/或102B之类的应用程序用来向用户101和计算机系统100认证自己。系统100还可以使用应用程序标识来授权应用程序执行某些操作。(例如,访问数据、生成受保护的数据等)。内核模块106可以使用应用程序标识来控制文件系统108,以访问非易失性存储器109上的应用程序102A或102B数据。

[0199] 特定应用程序的应用程序标识也可以与对应于该应用程序的进程相关联。由与特定应用程序相对应的进程生成的文件也可以与应用程序标识相关联。例如,应用程序标识可以与对应于特定应用程序的每个存储的数据文件相关联。这可以允许内核模块106识别相应的应用并控制请求进程是否可以访问存储在非易失性存储器109中的数据。

[0200] 计算机系统100可以被配置为在多个不同的阶段中提供访问控制。计算机系统100的当前操作阶段可以根据用户码是否可用而变化。所应用的安全级别可以根据计算机系统100的当前阶段而变化。例如,从最终用户101到可信环境150的信任链可以在两个阶段中进行操作。第一安全阶段可以提供比第二安全阶段更高的安全级别。

[0201] 例如,当用户码不可用时,计算机系统100可以在第一安全阶段中操作。当用户码可用时,计算机系统100然后可以在第二安全阶段中操作。

[0202] 在第一安全阶段中,计算机系统100可以被配置为防止任何进程访问非易失性存储器109中的受保护数据。也就是说,在第一个安全阶段,所有运行在计算机系统100上的进程都不可以访问存储在非易失性存储中受保护的数据。在该第一安全阶段中,用户码在计算机系统100中不可用(例如,在可信环境150中不可用)。在某些情况下,在第一阶段,由于解密数据所需的数据(例如,用户码)不可用,因此进程无法访问计算机系统100中的加密数据。

[0203] 当用户码可用时,计算机系统100可以在第二安全阶段中操作。在第二安全阶段,计算机系统100可以被配置为仅允许授权的应用程序进程访问受保护的数据。在第二安全阶段,计算机系统100可以被配置为防止未授权的应用程序访问受保护的数据。计算机系统100可以防止未经授权的应用程序访问受保护的数据,而同时支持授权的应用程序访问受保护的数据。

[0204] 计算机系统100可以基于何时向系统100提供用户码来区分在第一安全阶段和第二安全阶段中的操作。可以使用与输入用户码的时间相对应的时间戳来区分两个阶段。在本文所述的实施例中,令牌(本文称为appToken)可用于标识何时提供了用户码。

[0205] 应用程序令牌

[0206] 以下描述的应用程序令牌可以在任何访问控制系统或方法中独自使用,或者与其他特征或披露的特征一起进行任何组合或子组合使用。所述特征包括:系统体系结构、用户码、应用程序标识、文件访问数据、将应用程序添加到受保护区域的方法、进行进程认证的方法、授权进程访问数据文件的方法、受保护的数据文件、启动应用程序的方法、生成新的受保护数据文件的方法以及访问受保护数据文件的方法。如本文所使用的,应用程序令牌也可以被称为appToken。

[0207] 在本文描述的实施例中,应用程序令牌可以用于评估是否允许应用程序访问被请求数据文件。图2示出了示例进程200,其可以用于评估是否允许该进程访问被请求数据文件。进程200是可以由诸如计算机系统100之类的计算机系统实现的进程的示例。

[0208] 在210处,内核模块106可以接收对存储在非易失性存储存储器109中的被请求数据文件的文件请求。所述被请求数据文件可以对应于所述多个受保护的数据文件中的一个受保护的数据文件。在一些示例中,内核模块106可以基于与所述被请求数据文件相关联的属性数据来确定所述被请求数据文件是否是受保护的数据文件之一。

[0209] 所述文件请求可以从在富操作系统内运行的请求进程那里收到。例如,所述请求进程可以向内核140发出针对所述被请求数据文件的系统调用。内核模块106可以在所述文件请求到达文件系统108之前截获所述文件请求。

[0210] 在220处,计算机系统100可以被配置为确定与所述请求进程相关联的关联应用程序。例如,计算机系统100可以响应于对所述被请求数据文件的系统调用来确定安装在计算

设备100上的应用程序102中的哪个与所述请求进程相关联。内核模块106可以识别与所述请求进程相关联的应用程序标识,以便识别所述关联应用程序。

[0211] 在230处,响应于所述文件请求,计算机系统100可以确定所述关联应用程序是否存在一个验证的应用程序令牌。内核模块106可以使用所述关联应用程序的应用程序标识来确定所述应用程序令牌是否存在于计算机系统100中。内核模块106还可以验证所述应用程序令牌,例如通过与可信应用程序115的通信。

[0212] 在240处,在确定存在有效的应用程序令牌之后,计算系统100可以向所述请求进程提供对所述被请求数据文件的访问。仅当存在上述验证的应用程序令牌时,计算机系统100才可以允许所述请求进程访问所述被请求数据文件。否则,计算机系统100可以禁止所述请求进程访问所述被请求数据文件。

[0213] 计算机系统100可以被配置为防止访问(例如,读/写)存储在非易失性存储器109上的受保护的文件的明文/密文,除非与所述请求进程相对应的应用程序的应用程序令牌在计算机系统100内是可用的。替代地或附加地,每个受保护的文件可以与一个或多个指定的应用程序102A相关联。这个关联可以通过使用应用程序标识来完成。计算机系统100会禁止对所述受保护的文件的读取和/或写入操作, 除非与这些文件关联的应用程序的应用程序令牌存在并被验证。

[0214] 在确定计算机系统100内有有效的用户码可用后,计算机系统100可为一个特定应用程序生成应用程序令牌。仅当所述有效的用户码可用时,才会生成所述应用程序令牌。

[0215] 现在参考图3,其中示出了示例进程300,其可以用于生成应用程序令牌。方法300是可以由诸如计算机系统100之类的计算机系统实现的令牌生成过程的一个示例。在310,计算机系统100可以检测与特定应用程序相关联的第一进程的启动。

[0216] 在320处,计算机系统100可以确定所述有效的用户码是否可用。仅在所述有效的用户码在计算机系统100中可用之后,才会去定义所述应用程序令牌。

[0217] 例如,计算机系统100可以确定所述有效的用户码在可信执行环境中是否可用。在确定所述有效的用户码可用之后,计算机系统100可以在360处使用所述有效的用户码来生成所述应用程序令牌。

[0218] 计算机系统100可以确定与所述第一进程的启动相对应的时间戳。然后可以使用时间戳生成所述应用程序令牌。所述应用程序令牌可以通过使用时间戳来定义,该时间戳指明与所述应用程序相关联的所述第一个进程是什么时候启动的。因此,应用程序令牌可以提供有关该应用程序何时首次启动的指示。

[0219] 所述应用程序令牌可以通过使用所述关联应用程序的唯一应用程序标识来生成。这样可以确保每个应用程序都有一个唯一的应用程序令牌,后者并可以在向关联的进程提供对所述被请求数据文件的访问之前被验证。

[0220] 使用可通过与可信执行环境150进行通信来验证的数据,可以生成所述应用程序令牌。用于生成所述应用程序令牌的部分数据可能仅在可信执行环境150中可用(或仅可从存储在可信执行环境中的数据派生出来)。例如,存储在可信执行环境150中的数据或从存储在可信执行环境150中的其他数据生成的数据可用于生成所述应用程序令牌。这可以确保在验证应用程序令牌的过程中包括(并且必须征询)可信执行环境150。

[0221] 在某些情况下,可以使用所述应用程序标识和所述用户码的组合来定义所述应用

程序令牌。这可以允许随后使用用户码来验证应用程序令牌。例如,用户码的哈希版本可以用于生成应用程序令牌。这样可以确保用户码不能从所述应用程序令牌中确定。

[0222] 作为示例,可以使用时间戳、应用程序标识和用户码来定义应用程序令牌。例如,计算机系统100可以将应用程序令牌(appToken)定义为:

[0223] enc(timestamp | appID_length | appID,Qcode_hash)|

[0224] hmac(timestamp | appID_length | appID,Qcode_hash)

[0225] 其中时间戳表示与所述关联的应用程序相对应的所述第一个进程启动时的时间戳,enc(,)表示加密函数,其中enc(x,y)表示使用y字符串加密字符串x产生的输出,hmac(,)表示哈希函数,其中hmac(x,y)表示用秘密密钥y对x的字符串进行哈希处理的输出字符串;“|”表示串联操作,appID_length表示应用程序标识(appID)的字节长度,而Qcode_hash表示用户码(Qcode)的哈希值。

[0226] 可以响应于用户101输入与所述有效的用户码相对应的用户码输入来生成应用程序令牌。例如,如上所述,计算机系统100可以在320处检测与所述关联应用相关联的第一进程的启动。

[0227] 在330处,计算机系统100可以结合所述第一进程的启动来产生应用程序启动提示。所述应用程序启动提示被定义为提示用户输入用户码或相应的用户码输入。

[0228] 例如,在320处,计算机系统100可以确定用户码在计算机系统100内是否可用。可以响应于确定用户码在计算机系统100内不可用而生成应用程序启动提示。替代地或附加地,可以响应于每个在启动的应用程序来生成应用程序启动提示。在某些情况下,可以响应于每个早启动的用户模式应用程序来生成应用程序启动提示,而对于系统应用程序可以省略该应用程序启动提示。

[0229] 在340处,计算机系统100可以接收应用启动输入(例如,用户码输入)。用户可以响应于应用程序启动提示而提供应用程序启动输入。应用程序启动输入可以对应于预期的用户码输入,该预期的用户码输入(直接或间接)定义了所声称的用户码。

[0230] 在350,计算机系统100可以确定所声称的用户码是否对应于所述有效的用户码。在360处,一旦在350处确定应用程序启动输入对应于所述有效的用户码(即,所声称的用户码与所述有效的用户码匹配),则计算机系统100可以生成应用程序令牌。

[0231] 如上所述,计算机系统100可以确定是否存在验证的应用程序令牌以便响应文件请求。可以通过与可信执行环境150的通信来验证与特定应用相关联的应用程序令牌。例如,可以通过内核模块106与在可信执行环境150中运行的可信应用115之间的通信来执行对应用程序令牌的验证。

[0232] 如上所述,可以使用需要访问可信执行环境150的数据(例如哈希的用户码)来生成应用程序令牌。该可信数据可能仅在可信执行环境150中或通过最终用户101的输入才可用于计算机系统100。这可以确保对appToken的验证只能在可行环境150中进行。

[0233] 如上所述,可以响应于来自用户101的直接输入来生成应用程序令牌,并在可行环境 150中对其进行验证。因此,应用程序令牌可以用作一个软件锁,可以为存储在非易失性存储器109中受保护的数据提供高级别的安全性。即,当应用程序令牌对于与请求进程相对应的应用程序不可用时,可以防止对被保护的文件的访问。这可能有助于保护受保护的文件免受任何软件级别的黑客攻击。

[0234] 例如,受保护的数据文件可以与特定应用程序相关联。因此,当对应于特定应用程序102A的应用程序令牌不可用时,计算机系统100可以防止请求进程访问该受保护的数据文件。这提供了针对属于该应用程序102A的数据的软件级黑客/攻击的保护。

[0235] 当对应于与受保护数据相关联的应用程序的应用程序令牌可用时,存在一个访问受保护的数据的授权应用程序102A。在这种情况下,计算机系统100仍可以被配置为防止对受保护的数据文件的未授权访问。例如,如果数据文件不与请求进程的应用程序相关联,则仍可以阻止来自计算机系统100中运行的其他进程的对被请求数据文件的访问请求。

[0236] 在一些示例中,受保护的数据文件可以被加密。可以使用在可信环境150内生成的加密密钥来加密受保护的数据文件。这可以进一步防止对受保护的数据文件的未授权访问。

[0237] 文件访问数据

[0238] 以下描述的文件访问数据可以在任何访问控制系统或方法中独自使用,或者与其他特征或披露的特征一起进行任何组合或子组合使用。所述特征包括:系统体系结构、用户码、应用程序标识、应用程序令牌、将应用程序添加到受保护区域的方法、对进程进行认证的方法、授权进程访问数据文件的方法、受保护的数据文件、启动应用程序的方法、生成新的受保护数据文件的方法以及,访问受保护数据文件的方法。

[0239] 在本文描述的实施例中,存储在非易失性存储器109中的一些或全部数据文件可以以加密形式存储。例如,可以对存储在非易失性存储器109中的受保护数据文件进行加密。另外还有,可以对存储在非易失性存储器109中的所有数据文件进行加密。

[0240] 另外,与一个或多个特定应用相关联的数据文件可以被加密。例如,计算机系统100可以包括需要更高级别的安全性的应用程序102A。因此,与应用程序102A相对应的所有数据文件可以以加密形式存储。

[0241] 可以使用加密密钥对每个加密的数据文件进行加密。在某些情况下,每个加密文件的加密密钥可能特定于该文件(一个文件特定的加密密钥)。因此,可能需要文件特定的解密密钥,以便访问每个加密数据文件的明文数据。取决于所使用的特定加密技术,解密密钥可以与加密密钥相同,解密密钥也可以与加密密钥不同。

[0242] 每个加密的数据文件可以与文件访问数据相关联。例如,文件访问数据可以被存储为加密数据文件的扩展属性。与加密数据文件关联的文件访问数据可以包括密钥材料。密钥材料可用于确定与该加密数据文件相对应的解密密钥。文件访问数据可以允许在可信执行环境150内确定解密密钥。如上所述,加密/解密密钥可以对于每个加密文件是特定的。因此,每个加密文件的密钥材料可以是文件特定的。

[0243] 可以基于在可信执行环境150内存储的数据来生成文件访问数据。例如,可以使用与用户码相对应的访问代码秘密来定义每个加密文件的密钥材料。只能在可信环境 150内访问该访问代码秘密。这可以确保对加密数据文件的解密需要与可信执行环境150进行通信。结果就是,在可信执行环境中运行的可信应用程序115可以判断系统是否有为被请求的加密数据文件提供了加密密钥。

[0244] 非易失性存储器109存储的多个加密数据文件中的每个加密数据文件可以使用文件特定的加密密钥来加密。可信应用程序115可以被配置为针对每个加密的数据文件独立地确定文件特定的加密密钥。例如,可信应用程序115可以独立地(例如,在文件创建时)为

每个加密文件确定随机密钥信息。然后可以使用随机密钥信息来导出对应的文件加密密钥。该密钥信息或可用于导出该密钥信息的数据可以包含在对应文件的文件访问数据中。

[0245] 在某些情况下,加密的数据文件可能与特定的应用程序相关联。例如,关联的应用程序可以是用于生成特定的加密数据文件的应用程序。

[0246] 与特定的加密数据文件相关联的文件访问数据可以包括专用访问数据。可以基于与相关联的应用程序相关联的识别数据来定义应用程序特定的访问数据。例如,可以使用关联的应用程序的应用程序标识来定义与特定数据文件相对应的应用程序特定的访问数据。

[0247] 在某些情况下,文件访问数据可能以不可理解的形式定义。例如,可以使用存储在可信执行环境150中的用户码(或用户码数据)对文件访问数据进行加密。结果就是,从文件访问数据中导出有用的数据可能需要用户码的知识。由于用户码可能仅在可信执行环境150中可用(或通过用户输入),因此这可以确保可信执行环境150或用户101参与对明文文件访问数据的导出。

[0248] 计算机系统100可以将文件访问数据生成为一个字符串。例如,对于与与特定应用程序标识(appID)对应的应用程序相关联的数据文件,文件访问数据可以用以下格式定义:

[0249] class_id|profile_id|reserved_len|appID_length|appID|enc(keyIndex,Qcode_secret)|enc(keyLength|fileKey,Qcode_secret)|

[0250] hmac(class_id|profile_id|reserved_len|appID_length|appID|keyIndex|keyLength|fileKey,Qcode_secret)

[0251] 其中class_id代表一个常数,指示加密密钥的类别,profile_id代表一个64位整数,可用于区分与文件关联的配置文件(该配置文件可能对应于该设备的一个特定用户),fileKey代表文件加密密钥,enc( ,)代表加密函数,其中enc(x,y)代表使用y密钥加密x字符串的输出字符串,hmac(,)代表哈希函数,hmac(x,y)代表使用秘密密钥y计算字符串x的哈希得到的字符串输出,“ |”代表级联操作,并且Qcode_secret代表仅在可信环境 150内可访问的访问代码秘密。仅在可信环境 150内可访问的常量的示例在标题为“在可信执行环境中的个性化、密码学意义安全的访问控制技术”的美国专利申请第16 / 521,945号中进行了详细说明,该专利全部内容本文纳入引用。

[0252] 文件访问数据字符串可以被存储为加密数据文件的扩展属性。随后,计算机系统100可以使用文件访问数据字符串来识别和/或生成用于加密数据文件的解密密钥。

[0253] 在一些实施例中,可以在可信执行环境150内生成文件访问数据。如上所述,可以使用仅在可信环境 150内可用的数据(例如访问代码秘密)来生成文件访问数据。因此可以随后在可信环境 150内分析文件访问数据,以便辨识适当的解密密钥。

[0254] 可以将文件访问数据从可信环境150中发送到富操作系统。如上所述,文件访问数据可以作为受保护文件的扩展属性存储在非易失性存储器109上。另外,文件访问数据可以存储为可以通过文件系统108读取然后发送回可信环境 150的文件属性。

[0255] 将应用程序添加到保护区的方法

[0256] 以下是对将应用程序添加到受保护区域的方法的描述,该方法可以由其自身在任何访问控制系统或方法中使用,或者与所公开的下列任何其他一个或多个特征进行任何组合或子组合使用,这包括系统体系结构、用户码、应用程序标识、应用程序令牌、文件访问数

据、进行进程认证的方法、授权进程访问数据文件的方法、受保护的数据文件、启动应用程序的方法、生成新的受保护数据文件的方法以及访问受保护数据文件的方法。

[0257] 在本文描述的实施例中,计算机系统100可以将一个或多个应用指定为一组受保护的应用。可能允许该组受保护的应用程序中的应用程序访问受保护的数据文件。受保护的应用程序也可能生成新的受保护数据文件。

[0258] 其他应用程序对受保护的应用程序存储的受保护数据的访问可能会受到限制。在某些情况下,可能会阻止其他应用程序访问受保护的应用程序存储的受保护数据。

[0259] 在某些情况下,可能主要针对用户模式进程来限制对受保护数据的访问。例如,可以禁止未受保护的用户模式应用程序访问受保护的应用程序存储的受保护的数据。在某些情况下,假设访问控制系统的其他要求(例如,用户码、应用程序令牌和/或解密密钥的可用性)得到满足,仍然可以允许系统应用程序访问受保护的数据。

[0260] 计算机系统100可以提供用于将应用程序标识为受保护的应用程序的方法(例如,将应用程序注册在受保护的应用程序区域中)。例如,计算机系统100可以定义访问策略文件,该访问策略文件标识一个或多个受保护的应用程序。访问策略文件可以由计算机系统100用于确定是否允许特定应用程序(或与该特定应用程序关联的进程)访问被请求的文件。

[0261] 可以将访问策略文件定义为在该组受保护应用程序中包括每个受保护应用程序的一个唯一的应用程序标识。当请求进程向内核模块106发出文件请求以获取受保护的数据文件时,内核模块106可以检查访问策略文件以确定关联应用程序的应用程序标识是否包括在访问策略文件中。仅当关联应用程序的唯一应用程序标识包含在访问策略文件中时,内核模块106才允许请求进程访问被请求数据文件。否则,内核模块106会禁止请求应用程序访问被请求数据文件。

[0262] 访问策略文件可以存储在计算机系统100的非易失性存储器109中。另外,访问策略文件可以存储在可信执行环境150的非易失性存储器中。

[0263] 计算机系统100可以提供一个系统功能,该系统功能可以用于保护存储在非易失性存储存储器中的数据文件。例如,运行Android操作系统的计算机系统100可以提供系统功能来保护任何Android应用程序的文件。如图1所示,计算机系统100可以包括一个注册服务105,该注册服务105可以用于将应用注册到受保护区域(即,将应用程序识别为受保护的应用)。

[0264] 在某些情况下,计算机系统可以允许用户101将一个或多个应用程序指定为受保护的应用程序。用户101可以将应用程序添加到“受保护区域”,以确保来自应用程序的数据作为计算机系统100中的受保护数据进行存储和管理。用户101也可以从受保护区域中删除应用程序。

[0265] 用户101可以选择要添加到受保护区域的应用程序。在用户101选择应用程序之后,计算机系统100可以生成策略修改提示以请求或提示用户101提供用户码(Qcode)。然后,计算机系统100可以接收策略修改输入(例如,用户码输入)。计算机系统100可以通过与可信应用程序115通信来验证输入的用户码是否与有效的用户码相对应。类似地,在用户101选择要从受保护区域中删除的应用程序的情况下,计算机系统100可以根据用户码验证该请求。

[0266] 一旦在可信环境 150中成功验证了接收到的用户码输入,计算机系统100可以允许注册服务105执行操作,例如将某应用程序添加到受保护区域和/或从受保护区域中删除某应用程序。

[0267] 要将应用程序添加到受保护区域(即,将应用程序标识为受保护的应用程序),可以先确定该应用程序的应用程序标识。然后可以将确定的应用程序标识添加到存储在非易失性存储器109和/或可信环境 150中的访问策略文件中。

[0268] 可选地,计算机系统100可以在内核140的内核内存中包括内核访问策略数据结构(也称为appID_list)。应用程序标识也可以添加到内核访问策略数据结构中。

[0269] 要从受保护区域中删除应用程序(即,指定该应用程序不再是受保护的应用程序),可以先确定该应用程序的应用程序标识。该应用程序标识可以用于搜索存储在计算机系统100上(例如,在非易失性存储器109中)的访问策略文件。另外,应用程序标识可以用于搜索存储在内核内存中的内核访问策略数据结构。

[0270] 如果在访问策略文件(或内核访问策略数据结构)中标识了应用程序标识,则可以从存储在计算机系统100上的访问策略文件中删除该应用程序标识。内核访问策略数据结构存储在内核内存中。

[0271] 计算机系统100可以将访问策略文件与从用户码生成的用户码数据(例如,用户码的哈希版本)相关联地存储在非易失性存储器109中。计算机系统100可以将从根据输入的接收到的用户码确定的接收到的用户码数据与用户码数据进行比较,该用户码数据是与访问策略文件相关联地存储的。

[0272] 计算机系统100可以防止对访问策略文件的修改(例如,写操作、删除操作),除非接收到与用户码相对应的策略修改输入并针对与访问策略相关联存储的用户码数据进行了验证。这可以帮助防止对访问策略文件进行未经授权的修改(即篡改)。

[0273] 例如,计算机系统100可以与用户码的哈希版本相关联地将访问策略文件存储在非易失性存储器109中。可以使用用户码的哈希值来保护数据访问策略文件。因此,计算机系统100可以在没有用户码的情况下防止对策略文件的写操作。

[0274] 在某些情况下,即使在没有用户码的情况下(即,在没有与用户码相对应的策略修改输入的情况下),也可能允许对访问策略文件的读取操作。因此,即使当用户码不可用时,系统100也可以访问访问策略文件以便确定知道应用程序是否受到保护。

[0275] 在实际操作中,计算机系统100可以接收一个策略更新输入,该策略更新输入指定要添加到访问策略文件中包括的一组受保护应用程序的特定应用程序。在一些示例中,计算机系统可以响应于策略更新输入而生成更新验证提示。可以定义更新验证提示,以提示用户输入用户码。这可以确保用户101已经适当地授权了对访问策略文件的修改。

[0276] 计算机系统100可以响应于更新验证提示而接收更新验证输入(例如,用户码输入)。然后,计算机系统100可以使用与访问策略文件相关联地存储的用户码数据(例如,用户码的哈希版本)来确定更新验证输入是否对应于有效用户码。计算机系统100可以响应于确定更新验证输入对应于有效用户码来更新访问策略文件以包括特定应用。如果更新的验证输入与有效的用户码不对应,则可以拒绝/阻止对访问策略文件的修改。

[0277] 如上所述,内核140可以在富操作系统内可访问的内核内存中存储访问策略数据结构。计算机系统100可以被配置为将访问策略数据结构与存储在非易失性存储器109中的

访问策略文件同步。访问策略数据结构(即,appID_list数据结构)和访问策略文件可以被同步,使得访问策略数据结构包括访问策略文件中定义的一组受保护应用程序中每个受保护应用程序的唯一应用程序标识。

[0278] 在某些情况下,访问策略文件和访问策略数据结构的更新也可以同步。例如,可以通过最初更新内核内存访问策略数据结构来更新访问策略文件。然后,可以使用更新的内核内存数据结构覆盖现有的访问策略文件。

[0279] 在一些实施例中,可以响应于来自用户101的策略更新输入(跟随验证),即时更新内核存储器中的访问策略数据结构。可以先更新内核存储器中的访问策略数据结构,然后进行刷新操作以覆盖非易失性存储中的相应策略文件。如果不希望经常对访问策略文件进行更新,则最好使用此过程。

[0280] 如上所述,可以通过用户码来控制访问策略文件和对访问策略文件的更新。访问策略文件因此可以基于用户码(可以从用户101接收并且存储在可信环境 150中的信任链的硬件锚中)来实现密码保护。

[0281] 进程认证方法

[0282] 以下是对进程进行认证的方法的描述,该方法本身可以在任何访问控制系统或方法中使用,或者与所公开的下列任何其他一个或多个特征进行任何组合或子组合使用,这包括系统体系结构、用户码、应用程序标识、应用程序令牌、文件访问数据、将应用程序添加到受保护区域的方法、授权进程访问数据文件的方法、受保护数据文件以及启动应用程序的方法、生成新的受保护数据文件的方法以及访问受保护数据文件的方法。

[0283] 在许多计算机系统中,可以基于相关联的应用程序的应用程序识别码来进行进程认证。当启动进程时,可以验证与该进程相对应的应用程序的应用程序证书。该进程还可以与进程标识相关联。

[0284] 一旦已经验证了应用程序证书,就可以使用相关联的应用程序的应用程序识别码来标记进程标识。此过程可能会将潜在的安全漏洞引入计算机系统。例如,应用程序识别码可以用于标记与该应用相对应的数据文件。当应用程序识别码被重新使用时(例如,当应用程序被卸载并且应用程序识别码被分配给其他应用程序时),数据泄漏可能导致不同的应用程序能够访问由先前的具有相同的识别码的应用程序存储的数据文件。

[0285] 另外,有可能使用系统调用(例如,setUID)来修改进程标签,这可能给数据安全性带来漏洞。这些系统调用通常由root特权保护。但是,对于需要数据安全性以防止滥用根特权的系统,应用程序识别码不是一个好的进行进程认证的选择。

[0286] 本文描述的实施例可以提供一个密码学意义上安全的用于认证应用程序的进程的过程。在本文描述的实施例中,用于特定应用的应用程序标识可以与对应于该应用的每个进程相关联。该应用程序标识可以通过用相应的应用程序标识来标记该进程标识,从而与该进程标识相关联。

[0287] 如图1所示,可以在内核140中定义一个进程映射数据结构。可以将进程映射数据结构存储在内核存储器中。可以将进程映射数据结构定义为包括在任务列表107中维护的每个进程标识与对应应用程序的应用程序标识之间的映射。特定进程标识(PID)和对应的应用程序标识之间的映射可以表示为PID_appID。可以在创建每个新过程时(例如,在对相应证书进行验证之后)定义映射,并将其存储在进程映射数据结构中。当启动每个进程时,

可以更新进程映射数据结构。当每个进程终止时,进程映射数据结构也会被更新。

[0288] 在计算机系统100中,可以在框架层级130上执行密码学意义上安全的进程认证。例如,可以由控制门104执行该进程认证。系统可以由控制门104生成给定进程标识和相应进程标识之间的映射(PID_appID)。当创建一个新进程以生成映射时,控制门104可以从应用程序102A / 102B和用户101接收输入。在某些情况下,控制门104可能仅在验证了应用程序证书之后生成映射(PID_appID)。然后该映射可以被存储在进程映射数据结构中。

[0289] 一旦已经存储了映射(PID_appID),则可以使用进程映射数据结构将与进程标识相对应的进程认证为具有对应应用程序标识的与该应用相对应的进程。

[0290] 可以使用用户访问代码来保护上述进程映射。例如,可以使用用户码来保护进程映射数据结构。对进程映射数据结构的修改可能需要与用户码相对应的用户码数据才能实现。

[0291] 结果就是,可以通过内核模块106与可信应用程序115之间的交互来管理和保护进程映射数据结构。进程映射数据结构可以确保进程的身份(以及与特定应用程序的关联)可以通过使用可信环境150进行身份验证,从而实现在其操作寿命内安全地进行身份认证和维护。

[0292] 如上所述,进程映射数据结构可以存储在内核存储器中,以记录进程标识和应用程序标识之间的关联。这可以提供进程与相应应用程序之间的集中映射。

[0293] 此外还有,系统可以使用分布式映射方案。例如,对应的应用程序标识可以存储在任务列表107中存储的进程的任务结构中。当应用程序标识包括在进程的任务结构中时,这可以提供一种简单的方法来指示该进程具有已通过身份认证。在某些情况下,计算机系统100可能需要附加安全性,以防止模仿任务结构中的应用程序标识和/或防止合法过程的任务列表107中存储的任务结构中的应用程序标识项被篡改。

[0294] 授权进程访问数据文件的方法

[0295] 下面是对授权过程访问数据文件的方法的描述,该方法可以由其自身在任何访问控制系统或方法中使用,或者与所公开的下列任何其他一个或多个特征以任何组合或子组合方式的使用,这包括系统架构、用户码、应用程序标识、应用程序令牌、文件访问数据、将应用程序添加到受保护区域的方法、对进程进行身份认证的方法、受保护的数据文件、启动应用程序的方法、生成新的受保护数据文件的方法以及访问受保护数据文件的方法。

[0296] 计算机系统100可以被配置为确定请求进程是否被授权访问所被请求数据文件。计算机系统100可以被配置为提供一个个性化和密码学意义上安全的方法,来授权请求进程访问被请求数据文件。

[0297] 上述进程授权过程可以包括多个授权阶段。例如,进程授权过程可以包括三个授权阶段。在第一授权阶段,可以生成与特定应用程序相对应的应用程序令牌(appToken)。例如,可以使用本文上面描述的令牌生成过程来生成令牌(例如,方法300)。当用户101提供输入以启动计算机系统100内的该应用102时,譬如通过选择应用程序图标或语音命令等方式,系统可以启动该令牌生成过程。

[0298] 在计算机系统100的示例中,可以由控制门104生成应用程序令牌。例如,可以通过来自用户101和应用程序102的输入来生成与应用程序102相对应的应用程序令牌104,如上面所述。当控制门104生成应用程序令牌时,它可以被转发到内核模块106。内核模块106可

以依次将应用程序令牌发送到可信环境150。在可信环境 150中运行的可信应用115可以被配置为验证从内核模块106接收到的令牌。

[0299] 当启动一个进程时,计算机系统100可以确定与该进程相关联的进程标识。然后,计算机系统100可以验证用于该进程的进程证书。例如,可以将进程证书与应用程序证书进行比较,以确保该进程具有有效和可靠的证书。如果该进程是针对应用发起的第一进程,则计算机系统100可以进一步验证和认证该应用程序证书。

[0300] 在某些情况下,计算机系统100可以被配置为在允许应用程序执行之前提示用户进行授权。在创建一个进程并成功验证证书之后,系统可以提示最终用户101提供用户码,以授权该应用程序运行。在该进程开始加载要执行的应用程序代码之前,计算机系统100可以提示最终用户101提供用户码。

[0301] 例如,在验证进程证书后,计算机系统100可以确定相关联的应用程序是否是受保护的应用程序之一。如果关联的应用程序是受保护的应用程序之一,则计算机系统100可以配置为使用用户码来管理对进程映射数据结构的更新。计算机系统100可以为用户101生成进程启动提示。进程启动提示被定义为提示用户输入用户码(例如,直接地或通过相关的用户码输入)。然后,计算机系统100可以响应于进程启动提示而接收进程启动输入(即,用户码输入)。计算机系统100可以确定进程启动输入是否对应于有效用户码。

[0302] 可以将用户码输入(例如,响应于进程启动提示而在进程启动输入中接收到的用户码)提供给控制门104。控制门104可以基于输入给用户的用户码来转发声称的用户码数据到可信应用程序115(通过内核模块106),从而在可信执行环境150中来完成验证。

[0303] 例如,控制门104可以计算用户码输入的哈希。然后可以将用户码输入的哈希版本提供给内核模块106。内核模块106可以依次将用户码输入的哈希版本发送到可信环境 150中的可信应用程序(TA)115。如上所述,用户码的哈希版本可以存储在可信环境 150内。可信应用程序115可以将输入的用户码的哈希版本与可信环境 150中存储的用户码的哈希版本进行比较。在标题为“可信执行环境中的个性化、密码学意义安全的访问控制”的美国专利申请16/521,945中进一步详细描述了一个用于验证用户码的进程示例,其全部内容通过引用合并于此。

[0304] 如果用户码未由可信应用程序115验证,则控制门104可以阻止进程和应用程序执行。另外还有,控制门104可以产生另一提示,以允许用户101具有多一次提供用户码的机会。

[0305] 如果用户码由可信应用程序115验证,则控制门104可以随后为相应的应用程序生成应用程序令牌。如上所述,控制门104可以使用当前时间戳和对应应用程序的应用程序标识来生成应用程序令牌。然后,控制门104可以将应用程序令牌提供给可信环境 150中的可信的应用115(例如,经由内核模块106)。该应用程序令牌可以用于向可信应用115指示应用102已经被启动。

[0306] 可信应用程序115可以被配置为管理上述存储在非易失性存储器109中的数据文件的加密密钥和/或加密密钥材料。可信应用程序115可以控制何时将加密密钥提供给内核模块106以对被请求的存储在非易失性存储器109中数据进行解密。

[0307] 可信执行环境150内的可信应用115可以被配置为仅在合适的应用程序令牌在可信执行环境150内可用之后才生成和/或提供加密密钥数据。对于可信环境150,在富操作系

统上运行的相应进程可以从可信环境150获取文件加密密钥,以便访问存储在非易失性存储器109上的受加密保护的数据文件。在应用程序令牌被提供给可信环境150(例如,来自控制门104)之前,在富操作系统内运行的进程将无法访问存储在非易失性存储器109上的加密数据文件。

[0308] 一旦计算机系统100已经确定进程启动输入对应于有效用户码(并且应用程序令牌在可信执行环境中可用),则计算机系统100可以进行进程授权方法的第二阶段。第二授权阶段可以在已经从最终用户101获得用户码输入并在可信环境150中成功验证之后进行。在第二阶段,计算机系统100可以确定是否允许请求进程访问被请求的受保护数据文件。

[0309] 在第二授权阶段,计算机系统100(例如控制门104)可以确定与请求进程相对应的应用程序标识。然后,计算机系统100可以使用存储的访问策略文件和/或内核访问策略数据结构来确定应用程序标识是否对应于受保护的应用程序之一。

[0310] 例如,可以将应用程序标识提供给内核模块106。内核模块106可以使用接收到的应用程序标识来确定相关的应用是否包括在内核访问策略数据结构中。

[0311] 如果应用程序标识对应于一个在内核访问策略数据结构和/或访问策略文件中列出的被保护应用程序清单,则计算机系统100可以确定该请求进程是受保护进程。然后可以允许该请求进程访问存储在非易失性存储器109中的受保护数据。

[0312] 请求进程的进程标识也可以存储在内核内存中的授权进程数据结构(也称为Trusted_PID数据结构或PID_list数据结构)中。可以定义授权进程数据结构以提供所有授权进程的白名单。授权的进程数据结构可以在富操作系统中访问。授权进程数据结构可以提供集中式数据结构,随后系统可以对该集中式数据结构进行评估,以确定是否允许请求进程访问被请求数据文件。

[0313] 如上所述,第二授权阶段可以仅在从用户101接收到用户码输入并由可信应用程序115成功验证了用户码输入之后才发生。这可以确保系统可以使用用户码来保证对授权进程数据结构的修改具有个性化和密码学意义上的安全性。

[0314] 计算机系统100也可以在第三授权阶段中操作。第三授权阶段可以在进程已经被认证并且相应的进程标识已经被添加到授权进程数据结构之后发生。内核模块106可以被配置为以进行中的模式访问和评估授权的进程数据结构。这可以允许内核模块106管理由文件系统108针对存储在非易失性存储器109中的受保护文件执行的文件操作。

[0315] 例如,当请求进程通过文件系统108发送关于受保护文件的文件读取请求和/或文件写入请求时,系统100可以检查请求进程的进程标识是否存储在文件系统中。内核模块106在内核内存中维护的授权进程数据结构。系统100可以被配置为仅在确定请求进程的进程标识被存储在授权进程数据结构中之后才允许请求进程执行所请求的操作。

[0316] 被保护的数据文件

[0317] 以下是对被保护的数据文件的描述,该数据文件本身可以在任何访问控制系统或方法中使用,或者与所公开的下列任何其他一个或多个特征以任何组合或子组合方式的使用,这包括系统体系结构、用户码、应用程序标识、应用程序令牌、文件访问数据、将应用程序添加到受保护区域的方法、对进程进行认证的方法、授权进程访问数据文件的方法、启动应用程序的方法、生成新的受保护数据文件的方法以及访问受保护数据文件的方法。

[0318] 计算机系统100可以被配置为限制对存储在非易失性存储器109中的特定受保护

的数据文件的访问。所述受保护的数据文件可以用于提供密码学安全的数据。在某些情况下,所述受保护的数据文件可能包括应用程序特定的受保护的数据文件。每个应用程序特定的受保护数据文件都可以与一组允许访问该受保护的数据文件的授权应用程序相关联。

[0319] 每个应用程序特定的受保护的数据文件可以与一组文件特定的应用程序相关联。所述文件特定的应用程序组可以包括安装在计算系统100上的至少一个应用程序102。仅当与请求进程相关联的关联应用程序为与所述被请求数据文件关联的文件特定的应用程序组中一个应用程序时,请求进程才被允许访问被请求应用程序特定的受保护的数据文件。给定受保护的数据文件的文件特定的应用程序组可以被包含在该受保护的数据文件的应用程序特定的访问数据中。

[0320] 在某些情况下,可以将特定受保护的应用程序生成的每个文件与该应用程序的应用程序标识相关联。与受保护的数据文件相关联的应用程序标识可用来限制与其他应用程序相关联的进程对该受保护的数据文件的访问。例如,与特定应用程序标识相关联的数据文件只能由与该相同应用程序标识相关联的进程访问(根进程除外)。

[0321] 在某些情况下,与受保护的数据文件相关联的文件特定的应用程序组可以与额外的一些应用程序相关联。例如,当通过诸如getSharedPreferences(String,int),openFileOutput(String,int)之类的API创建新文件时,可以将应用程序配置为使用MODE_WORLD_READABLE和MODE_WORLD_RITEABLE标志,以允许额外的一些应用程序,同时和/或独立地,读取/写入受保护的数据文件。

[0322] 受保护的数据文件可以包括加密的受保护的数据文件。加密的受保护数据文件可以以加密/密文形式存储在非易失性存储器109中。这可以帮助防止未经授权的进程读取受保护数据文件的任何明文数据。每个加密数据文件的文件内容都可以通过加密过程得到保护。相应的加密密钥可以由可信环境150中的可信应用 115管理。

[0323] 可以使用文件特定的加密密钥来加密每个被加密保护的文件。计算机系统100可以被配置为将文件特定的文件访问数据与每个加密的受保护文件相关联。如上所述,文件访问数据可以包括密钥材料,该密钥材料指定用于该文件的相应加密/解密密钥的文件加密密钥信息。密钥材料可用于识别与该加密数据文件相对应的加密/解密密钥。在某些情况下,密钥材料可用于导出与该加密数据文件相对应的加密/解密密钥。

[0324] 文件访问数据可以由可信环境150中的可信应用程序115生成。文件访问数据可以与对应的加密数据文件相关联地存储在非易失性存储器109中。例如,文件访问数据可以通过文件系统108作为扩展的文件属性存储在非易失性存储器109中。

[0325] 在某些情况下,文件属性还可以提供与加密数据文件有关的其他数据。例如,文件属性可以定义标签,该标签提供一个说明相应文件是受保护的数据文件的指示。在某些情况下,受保护的文件标签可以作为文件访问数据的一部分包括在内。或者,可以与文件访问数据分开地提供受保护的文件标签。这可以有助于,例如在文件访问数据被加密的情况下,确定该文件是受保护的数据文件。

[0326] 如上所述,经处理的数据文件可包括一个或多个加密的数据文件。对于每个加密的数据文件,可以在可信环境150中生成相应的文件加密密钥。可以将密钥推导数据(即,文件访问数据)与该加密的数据文件相关联地存储,例如作为一个该文件的扩展属性。文件访问数据还可以包括与该受保护的数据文件相关联的应用程序的一个应用程序标识。

[0327] 在某些情况下,文件访问数据也可以被加密,例如使用用户码数据作为一个属性加密密钥来加密文件属性。这可以确保基于用户101定义的用户码通过可信应用程序115来保护密钥派生数据。这允许用户101,通过使用用户码,保护受保护数据文件的内容并管理对受保护数据文件的访问。

[0328] 启动应用程序的方法

[0329] 以下是一个启动应用程序的方法的描述,该方法可以在任何访问控制系统或方法中使用,或者与所公开的下列任何其他一个或多个特征以任何组合或子组合方式的使用,这包括:系统体系结构、用户码、应用程序标识、应用程序令牌、文件访问数据、将应用程序添加到受保护区域的方法、对进程进行认证的方法、授权过程访问数据文件的方法、受保护的数据文件、生成新的受保护数据文件的方法以及访问受保护数据文件的方法。

[0330] 计算机系统100可以提供用于实现密码学意义上安全的访问控制的一个应用程序启动(例如,应用程序启动)方法。当用户启动应用程序102时,可以启动相应的应用程序进程。例如,在使用Android操作系统进行操作的计算系统100中,一个新的进程是从合子过程派生出来的。合子进程是Android操作系统内的一个特殊系统进程,可以有效地为新启动的应用程序进程提供一个模板。

[0331] 在启动与特定应用程序相关联的第一进程时,计算机系统100可以验证与该相关联的应用程序相对应的应用程序证书。

[0332] 在验证证书之后,计算机系统100可以确定用户码在可信执行环境150中是否可用。计算机系统100可以在加载和执行应用程序类之前确定用户码是否可用。

[0333] 在一些情况下,计算机系统100可以响应于验证应用程序证书而生成一个验证的应用程序启动提示。该经过验证的应用程序启动提示可定义为提示用户101提供一个用户码输入。

[0334] 然后,计算机系统100可以响应于经验证的应用程序启动提示来接收经验证的应用程序启动输入(例如,定义了所声称的用户码的一个用户码输入)。然后,计算机系统100可以使用本文所述的各种过程来确定经验证的应用启动输入是否对应于有效用户码。例如,控制门104可以确定用户码输入的哈希版本。该用户码输入的哈希版本可以通过框架130中的控制门104提供给内核模块106,并在可信环境150中进行验证。

[0335] 如果用户码的验证失败,则可以终止上述进程启动过程。如果用户码的验证成功,则可以允许应用程序(和相应的进程)执行。仅在已经验证了应用程序证书并且计算机系统100确定已验证的应用程序启动输入对应于有效的用户码之后,才可以允许应用程序(和相应的进程)执行。

[0336] 当用户码的验证成功时,计算机系统100可以进行进一步的操作,从而在用户101和可信执行环境150之间建立信任链。例如,如本文所述,应用程序令牌可以是为相应的应用程序确定。可以在控制门104处确定应用程序令牌并将其发送到内核模块106。然后,内核模块106可以如上所述将应用程序令牌发送到可信执行环境150。在标题为“可信执行环境中的个性化、密码学意义安全的访问控制”的美国专利申请No.16 / 521,945中进一步详细描述了用于验证应用程序令牌和使用应用程序令牌管理应用操作的示例过程,其全部内容以引用方式合并于此。

[0337] 计算机系统100还可以将当前进程的进程标识包括在由内核模块106维护在内核

内存中的授权进程数据结构中,作为所有授权进程的白名单。然后可以授权当前进程去访问存储在非易失性存储器109中的一些或全部受保护文件。

[0338] 在某些情况下,如果应用程序是受保护的应用程序之一,则计算机系统100可能仅在应用程序执行之前需要用户码。一旦应用程序证书已经被验证,则计算机系统100可以确定相关联的应用程序是否是受保护的应用程序之一。如果相关联的应用程序是受保护的应用程序,则计算机系统100可以确定用户码是否可用。

[0339] 计算机系统100可以通过确定相关联的应用程序的唯一应用程序标识来确定相关联的应用程序是否是受保护的应用程序之一。然后,计算机系统100可以确定相关应用程序的唯一应用程序标识是否包括在内核模块106存储的访问策略数据结构中。控制门104可以计算相应应用程序的应用程序标识并检查该应用程序标识。该应用程序标识存储在访问策略数据结构中,该访问策略数据结构是由内核模块106在内核内存中的进行维护的。

[0340] 如果关联应用程序的应用程序标识不包括在访问策略数据结构中,则计算机系统100可以确定关联应用程序是不受保护的应用程序。在确定相关联的应用程序是不受保护的应用程序时,计算机系统100不需要用户码即可以允许相关联的应用程序执行。使用此处描述的过程,无论用户码是否可用,仍可以阻止不受保护的应用程序访问受保护的数据文件。

[0341] 当关联的应用程序的唯一应用程序标识包括在访问策略数据结构中时,计算机系统100可以确定关联的应用程序是受保护的应用程序之一。然后控制门104可以确定用户码在计算系统100内是否可用。

[0342] 在某些情况下,控制门104可以提示最终用户101直接或间接输入用户码,如本文所述。在确定关联的应用是受保护的应用之一之后,控制门104可以生成受保护的应用程序启动提示。该已验证的应用程序启动提示可以被定义为提示用户101提供用户码输入。

[0343] 计算机系统100可以响应于受保护的应用程序启动提示而接收受保护的应用程序启动输入(例如,一个用户码输入)。计算机系统100可以确定受保护的应用启动输入是否对应于有效的用户码,例如,通过将输入的用户码的哈希版本与存储的用户码的哈希版本进行比较。仅在处理器已确定受保护的应用程序启动输入对应于有效用户码之后,计算机系统100才允许关联的应用程序执行。

[0344] 创建一个受保护的文件

[0345] 以下是对一个生成新的受保护数据文件的方法的描述,该新保护的数据文件本身可以在任何访问控制系统或方法中使用,或者与包括所公示的下列任何其他一个或多个特征以任何组合或子组合方式使用,这包括:系统体系结构、用户码、应用程序标识、应用程序令牌、文件访问数据、将应用程序添加到受保护区域的方法、对进程进行认证的方法、授权进程访问数据文件的方法、受保护的数据文件、启动应用程序的方法以及访问受保护的数据文件的方法。

[0346] 计算机系统100可以被配置为生成新的受保护的数据文件。每个受保护的数据文件都可以与一组文件特定的应用程序相关联,这些应用程序被允许访问该受保护的数据文件。在某些情况下,所有受保护的应用程序都可能被允许访问特定的受保护数据文件。在某些情况下,对一个或多个受保护数据文件的访问可能仅限于部分受保护的应用程序(在某些情况下仅单个应用程序)。

[0347] 受保护的数据文件可以包括将该文件标识为受保护的数据文件的一个文件属性。该文件属性还可以包括文件访问数据。当受保护的数据文件是加密数据时,文件访问数据可以包括密钥材料。可以在可信环境150中生成文件访问数据的一部分,例如密钥材料,并将其发送回文件系统108,并存储在非易失性存储器109中的扩展属性域中。

[0348] 在某些情况下,当首次生成数据文件时,可以将数据文件定义为受保护文件。例如,计算机系统100可以被配置为针对由受保护的应用之一生成的每个新文件,自动地将该文件存储为受保护的数据文件之一。即,由与受保护的应用相对应的授权进程创建的新文件可以被自动保护(即,生成后即为受保护的数据文件)。

[0349] 在某些情况下,数据文件可能最初被生成和/或存储为未受保护的数据文件。例如,应用程序最初可能是不受保护的应用程序。该应用程序在未受保护的情况下生成的数据文件可以作为未受保护的数据文件存储在非易失性存储器109中。该应用随后可以被标识为受保护的应用,例如,用户将该应用程序添加到受保护的区域中。在这种情况下,可以将与该应用程序相对应的现有文件从不受保护的数据文件转换为受保护的数据文件。

[0350] 当一个应用被识别为受保护的应用时,计算机系统100可以识别存储在非易失性存储器109中的与该特定应用相对应的一组现有数据文件。然后,计算机系统100可以修改每个不受保护的数据文件使之成为受保护的数据文件。例如,注册服务105可以完成将数据文件从不受保护的数据文件修改为受保护的数据文件的操作。

[0351] 如上所述,与受保护的应用程序相关联的进程可以将其对应的进程标识存储在由内核模块106管理的内核内存中的授权进程数据结构中。在某些情况下,内核模块106还可以在内核内存中定义一个额外的转换进程数据结构(也称为conversion_PID数据结构)。转换进程数据结构可用于标识可将现有(未保护的)数据文件转换为其受保护版本的进程。

[0352] 在将进程标识添加到转换进程数据结构之前,注册服务105可以确定可信环境150中是否存在用户码。如上所述,这可能涉及提示用户提供用户码输入。这可以允许最终用户101使用注册服务105定义允许哪个或哪些进程执行转换操作。

[0353] 在某些情况下,内核模块106可以从在富操作系统中运行的特定进程接收写操作请求。写入操作可能与受保护的数据文件有关。内核模块106可以被配置为仅在授权的进程数据结构中包括特定进程的进程标识的情况下才允许该写操作请求发生。

[0354] 当在富操作系统中操作的进程创建一个新文件时,计算机系统100可以确定进程标识是否包括在转换进程数据结构中。

[0355] 当进程标识PID存在于转换进程数据结构中时,计算机系统100可以配置为将新文件存储为受保护的数据文件。如果计算机系统确定进程标识PID不在转换进程数据结构中,则内核模块106可以进一步检查PID是否存在于内核内存中的授权进程数据结构中。如果进程标识包括在授权进程数据结构中,则计算机系统100可以配置为将新文件存储为受保护的数据文件。

[0356] 为了将新文件存储为受保护的数据文件,内核模块106可以将请求发送到可信环境150以生成加密密钥和用于新文件的文件访问数据。 可信环境150可以为新文件生成随机文件加密密钥。加密密钥可以用于在将文件存储在非易失性存储器109中之前对文件进行加密。文件访问数据可以被提供给内核模块106以与新创建的文件相关联(例如,作为扩展属性)。

[0357] 如果在转换进程数据结构中不包括该进程标识并且在授权进程数据结构中不包括该进程标识,则计算机系统100可以被配置为将相应进程产生的新文件存储为不受保护的数据文件。例如,可以在没有相应扩展属性的情况下创建该文件。

[0358] 访问现有的受保护数据文件

[0359] 以下是对一个访问受保护的数据文件的方法的描述,该方法可以由其自身在任何访问控制系统或方法中使用,或者与所公开的下列任何一个或多个其他特征进行任何组合或子组合使用,这包括:系统体系结构、用户码、应用程序标识、应用程序令牌、文件访问数据、将应用程序添加到受保护区域的方法、进行进程认证的方法、授权过程访问数据文件的方法、受保护的数据文件、启动应用程序的方法以及生成新的受保护数据文件的方法。

[0360] 计算机系统100可以管理是否允许请求处理访问被请求数据文件。例如,在所被请求数据文件是受保护的数据文件的情况下,计算机系统100可以防止请求进程访问受保护的数据文件,除非该请求进程对应于某受保护的应用。

[0361] 计算机系统100可以定义一组受保护的应用程序。该组受保护的应用程序可以包括安装在计算设备100上的应用程序102中的至少一个。然后,仅当关联的应用程序是受保护的应用程序之一时,计算机系统100才可以允许请求进程访问所被请求数据文件。

[0362] 计算机系统100可以将每个受保护的数据文件与一个文件特定的应用程序组相关联。所述文件特定的应用程序组可以包含至少一个安装在所述计算设备上的应用程序。例如,每个受保护的数据文件可以与所述文件特定的应用程序组中的每个应用程序的应用程序标识关联。仅当与请求进程相关联的应用程序是与被请求数据文件相关联的文件特定的应用程序组中的一个应用程序时,计算机系统100才可以允许请求进程访问所述被请求数据文件。

[0363] 访问被请求数据文件的过程可以包括对被请求数据文件的文件属性进行初始评估。例如,计算机系统100可以确定所述被请求数据文件是否包括指示所被请求数据文件是受保护的数据文件的扩展属性,譬如一个受保护文件的标签。如果所被请求数据文件不具有与其相关联的受保护文件属性(例如,如果该数据文件不包括任何扩展属性),则计算机系统100可以确定所被请求数据文件不受保护。然后可以向请求进程提供对所被请求数据文件的访问。

[0364] 如果计算机系统确定受保护的文件属性与所被请求数据文件相关联,则计算机系统100可以确定该请求进程是否是授权过程。例如,计算机系统100可以通过确定请求进程的进程标识是否包括在存储在内核内存中的数据结构中来确定请求进程是否是授权过程,该数据结构可以是上述授权进程数据结构和/或上述转换进程数据结构。如果计算机系统确定该请求进程是授权过程,则可以允许该请求进程访问所被请求数据文件。

[0365] 在某些情况下,在计算机系统100确定请求进程的进程标识包括在转换进程数据结构中的情况下,可以为请求进程提供直接访问所被请求数据文件的权限。可以向请求过程提供对所被请求数据文件的访问,而无需评估与所被请求数据文件相关联的应用程序。这种情况下,进程认证过程的其余部分可以被绕过,并且内核模块106可以将请求发送到可信环境150以获得文件加密密钥。该请求可以包括与所被请求数据文件相关联的文件访问数据(或其至少一部分,例如密钥材料)。在此过程中,当将加密文件提供给内核模块106时,可信应用程序115可能不会验证数据文件是否与特定应用程序相关联。

[0366] 那些其进程标识包含在转换进程数据结构中进程,其对文件加密密钥执行的操作会受到一些限制。例如,该进程可能仅限于使用加密密钥来加密文件和/或写入新的受保护数据文件(而不是解密数据文件)。这可以支持转换进程的操作,因为通常转换过程可以简单地读取先前未受保护的明文文件并将该文件转换为密文格式作为受保护的数据文件写入非易失性存储器109。因此,可信应用 115不检查应用程序特定的访问数据(例如,用于指示文件与一个或多个受保护的应用程序之间的关联的文件所有权数据)这一事实不会引入安全漏洞。另外,如上所述,可以通过验证用户码来保护将进程标识添加到内核存储器中的转换进程数据结构中。

[0367] 在某些情况下,在计算机系统100确定转换进程数据结构中不包括请求进程的进程标识的情况下,计算机系统100可以确定在授权进程数据结构中是否包括由请求进程的进程标识。如果请求进程的进程标识未包括在授权进程数据结构中,则可以禁止请求进程执行与非易失性存储器109中的受保护文件有关的任何读取操作和/或写入操作。

[0368] 如果请求进程的进程标识包括在授权进程数据结构中,则内核模块106可以进一步评估是否允许请求进程访问特定的请求文件。例如,计算机系统100可以评估请求进程是否与也与所被请求数据文件相关联的应用程序相关联。内核模块106可以查询内核内存中的进程映射数据结构,以确定与请求进程相关联的应用程序标识。内核模块106可以将与请求进程相关联的应用程序的应用程序标识和所被请求数据文件的文件访问数据(或其一部分)提供给可信执行环境150中的可信应用程序115。

[0369] 可信应用程序115可以将与请求进程相关联的应用程序的应用程序标识与与所被请求数据文件相关联的应用程序的应用程序标识进行比较(例如,根据文件访问数据来确定),以便确定是否允许请求进程访问被请求数据文件。当进程的应用程序标识与文件访问数据中包含的应用程序特定的访问数据(例如文件所有者信息)匹配时,可信应用程序115可以仅向内核模块106提供与所被请求数据文件相对应的加密密钥。因此,可以允许每个授权进程访问与请求进程相同的应用程序相关的文件(以及将关联的应用程序标识为允许的应用程序的任何其他文件)。类似地,可以防止授权的进程访问与其他应用程序关联的受保护文件,即那些受保护的文件,其文件特定的应用程序组不包括与请求进程关联的应用程序。

[0370] 计算机系统100可以被配置为基于在富操作系统中运行的进程的生命周期来更新内核内存中的授权进程数据结构和/或转换进程数据结构。内核模块106可以被配置为通过监视在富操作系统中运行的进程来更新授权的进程数据结构和/或转换进程数据结构。

[0371] 计算机系统100可以检测在富操作系统中运行的行将终止进程的终止动作。当授权的进程终止时,计算机系统100可以更新进程白名单。内核模块106可以确定授权的进程数据结构包括终止进程的进程标识。然后,内核模块106可以更新授权的进程数据结构以去除该终止进程的进程标识。内核模块106还可以更新进程映射数据结构。这可以允许计算机系统100在启动和终止进程时重新使用进程标识,同时防止错误的授权继承。即,可以防止新进程继承被授予已过期的进程的许可,这里已过期的进程使用了同样的进程标识并且该进程标识被重新使用并分配给了新的进程。

[0372] 尽管以上描述呈现了实施例示例的特征,但是需要理解的是,在不脱离所描述实施例的精神和操作原理的情况下,所描述实施例的一些特征和/或功能可能会被修改。例

如,借助于所代表的实施例或示例描述的各种特性可以选择性地彼此组合。在其他情况下,这里没有详细描述公知的方法、过程和组件,以免混淆实施例的描述。因此,以上所描述的内容旨在例示所要求保护的概念,并且是非限制性的。本领域技术人员将理解,在不脱离如所附权利要求所限定的本发明的范围的情况下,可以做出其他变型和修改。权利要求的范围不应由优选实施例和示例所限制,而应给出与整个说明书一致的最宽泛的解释。

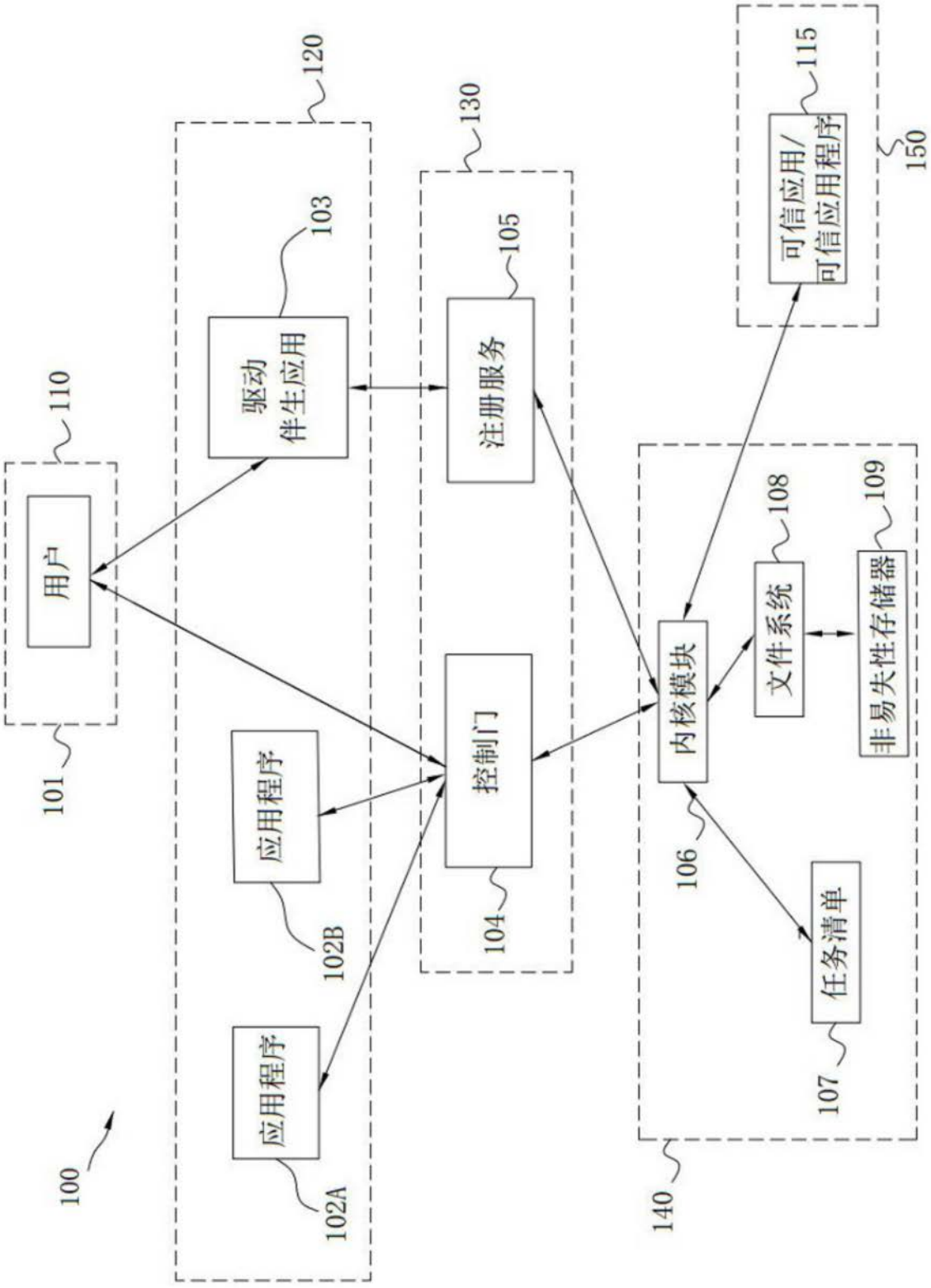

100
101
用户
110
120
驱动伴生应用
103
应用程序
102B
应用程序
102A
130
注册服务
105
控制门
104
140
内核模块
106
文件系统
108
非易失性存储器
109
任务清单
107
可信应用/可信应用程序
115
150

图1

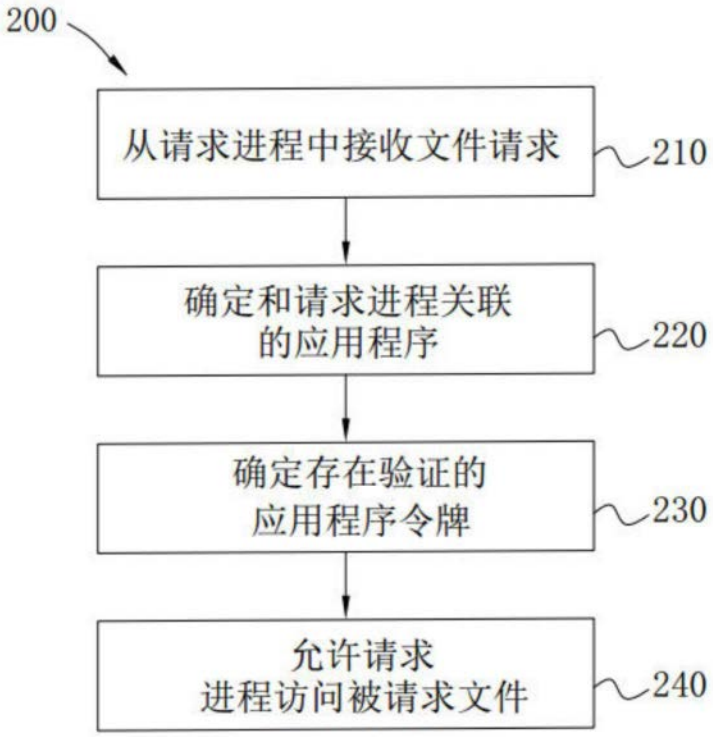

200
从请求进程中接收文件请求
210
确定和请求进程关联的应用程序
220
确定存在验证的应用程序令牌
230
允许请求进程访问被请求文件
240

图2

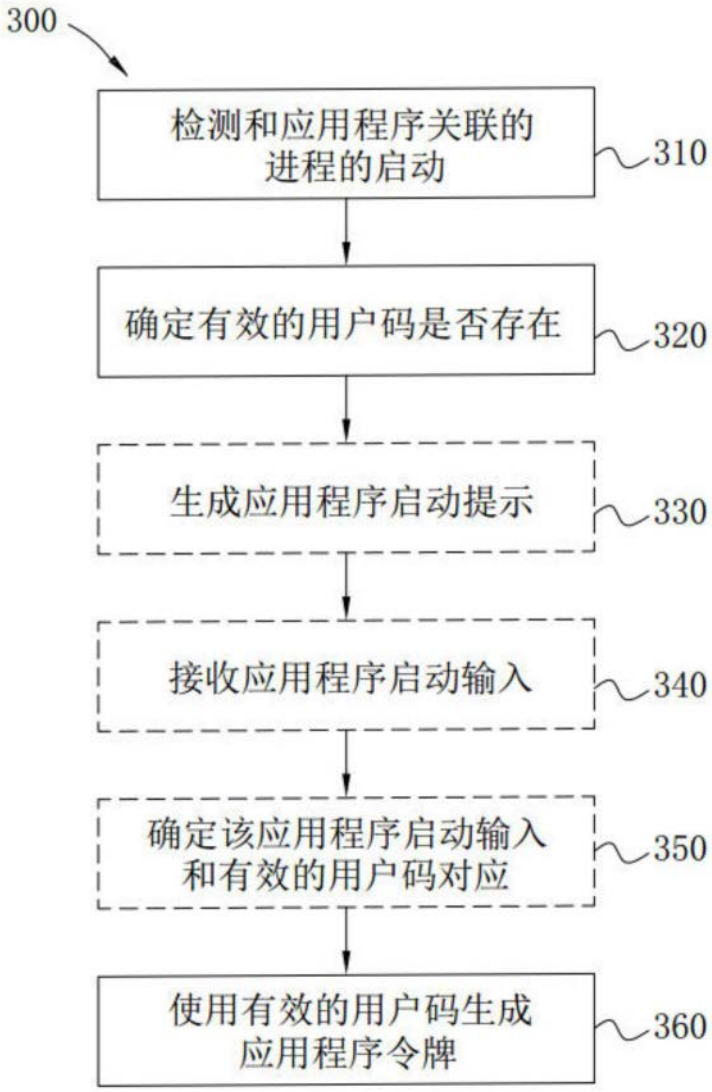

300
检测和应用程序关联的进程的启动
310
确定有效的用户码是否存在
320
生成应用程序启动提示
330
接收应用程序启动输入
340
确定该应用程序启动输入和有效的用户码对应
350
使用有效的用户码生成应用程序令牌
360

图3