(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 113642007 B
(45)授权公告日 2023.12.26

(21)申请号 202111008330.1
(22)申请日 2021.08.30
(65)同一申请的已公布的文献号
申请公布号 CN 113642007 A
(43)申请公布日 2021.11.12
(73)专利权人 京东方科技集团股份有限公司
地址 100015 北京市朝阳区酒仙桥路10号
(72)发明人 赵凯 马希通
(74)专利代理机构 北京志霖恒远知识产权代理有限公司 11435
专利代理师 刘进
(51)Int.Cl.
G06F 21/57(2013.01)
G06F 21/60(2013.01)
G06F 21/64(2013.01)
(56)对比文件
CN 110020528 A,2019.07.16
US 2020342111 A1,2020.10.29
CN 107577937 A,2018.01.12
US 2018324547 A1,2018.11.08
US 2010106979 A1,2010.04.29
CN 109213684 A,2019.01.15
DE 102020201768 A1,2021.08.12
US 2010175104 A1,2010.07.08
US 2011013771 A1,2011.01.20
US 2015010143 A1,2015.01.08
US 2019154463 A1,2019.05.23
US 2019340369 A1,2019.11.07
US 2020211301 A1,2020.07.02
US 6973646 B1,2005.12.06
WO 0044127 A1,2000.07.27
WO 2011079583 A1,2011.07.07
黄申文;王飞;周良.基于手写签名的电子公文安全认证方案设计.中国制造业信息化.2009,(第15期),全文. (续)
审查员 宗小淇

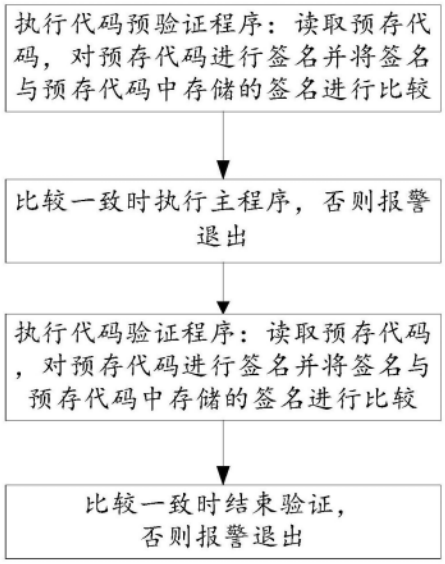
权利要求书1页 说明书6页 附图4页

(54)发明名称

代码验证方法、可联网的终端设备及可读存储介质

(57)摘要

本申请公开了一种代码验证方法、可联网的终端设备及可读存储介质，包括执行代码预验证程序：读取预存代码，对预存代码进行签名并将签名与预存代码中存储的签名进行比较；比较一致时执行主程序，否则报警退出；执行代码验证程序；比较一致时结束验证，否则报警退出。本申请实施例提供的技术方案，通过在主程序执行之前对代码进行预验证，首先确定代码的安全性，同时在预验证没有任何问题的情况下启动主程序，在主程序运行的过程中，同步启动对代码的验证程序，使得程序运行过程中进行验证守护，通过两段式的启动对代码进行验证和保护，同时代码验证过程中的签名随着时间的变化而变化，增加破解的复杂度。



CN 113642007 B

[转续页]

[接上页]

(56)对比文件

雷灵光;张中文;王跃武;王雷.Android系统代码签名验证机制的实现及安全性分析.信息网络安全.2012,(第08期),全文.

肖蕾;陈荣赏.椭圆曲线的数字签名技术在无线网络中的应用.电脑知识与技术.2008,(第25期),全文.

张娴.基于PKI技术代码签名实现原理以及应用.科技信息.2010,(第26期),全文.

1.一种代码验证方法,其特征在于,包括步骤:

执行代码预验证程序:读取预存代码,对预存代码进行签名并将签名与预存代码中存储的签名进行比较;所述预存代码前端为设备ID和时间戳,所述预存代码还设有签名区,预存有签名,所述时间戳根据代码运行的时间进行改变;具体包括:

读取预存代码的设备ID和时间戳,存放至第一测试区内,

判断预存代码的剩余代码长度是否小于第一长度;

若否,则顺次读取第一长度的剩余代码并进行签名,将签名存放至所述第一测试区的前端,

若是,则读取剩余代码并进行签名,并将签名存放至所述第一测试区的前端,将所述第一测试区内的签名与预存代码中存储的签名进行比较;

比较一致时执行主程序,否则报警退出;

执行代码验证程序:读取预存代码,对预存代码进行签名并将签名与预存代码中存储的签名进行比较;

比较一致时结束验证,

否则报警退出。

2.根据权利要求1所述的代码验证方法,其特征在于,所述执行代码验证程序包括:

读取预存代码的设备ID和时间戳,存放至第二测试区内,

判断预存代码的剩余代码长度是否小于第二长度;

若否,则顺次读取第二长度的剩余代码并进行签名,将签名存放至所述第二测试区的前端,

若是,则读取剩余代码并进行签名,并将签名存放至所述第二测试区的前端,将所述第二测试区内的签名与预存代码中存储的签名进行比较。

3.根据权利要求2所述的代码验证方法,其特征在于,所述第一测试区和所述第二测试区的长度为固定长度。

4.根据权利要求3所述的代码验证方法,其特征在于,所述预存代码为初始代码或者上次程序结束代码。

5.根据权利要求3所述的代码验证方法,其特征在于,在最终报警退出前还包括步骤:

判断当前程序版本是否为预定版本,

若是,则报警退出;

否则,保存当前程序的签名并更新时间戳。

6.根据权利要求1或2所述的代码验证方法,其特征在于,所述签名方式为MD5签名。

7.一种可联网的终端设备,其特征在于,所述终端设备包括:

一个或多个处理器;

存储器,用于存储一个或多个程序,当所述一个或多个程序被所述一个或多个处理器执行时,使得所述一个或多个处理器执行如权利要求1-6中任一项所述的代码验证方法。

8.一种存储有计算机程序的计算机可读存储介质,其特征在于,该程序被处理器执行时实现如权利要求1-6中任一项所述的代码验证方法。

代码验证方法、可联网的终端设备及可读存储介质

技术领域

[0001] 本发明一般涉及计算机技术领域，尤其涉及代码验证方法、可联网的终端设备及可读存储介质。

背景技术

[0002] 物联网(The Internet of Things，简称IOT)是指通过各种信息传感器、射频识别技术、全球定位系统、红外感应器、激光扫描器等各种装置与技术，实时采集任何需要监控、连接、互动的物体或过程，采集其声、光、热、电、力学、化学、生物、位置等各种需要的信息，通过各类可能的网络接入，实现物与物、物与人的泛在连接，实现对物品和过程的智能化感知、识别和管理。物联网是一个基于互联网、传统电信网等的信息承载体，它让所有能够被独立寻址的普通物理对象形成互联互通的网络。

[0003] 复杂的物联网应用环境，高价值的信息数据生成都要求在信息的产生初期形成成熟的安全保护机制，数据的传输过程中可以使用加密等方法进行保护，但可联网设备的代码可以从根本上被攻击从而进行篡改。

发明内容

[0004] 鉴于现有技术中的上述缺陷或不足，期望提供一种代码验证方法、可联网的终端设备及可读存储介质。

[0005] 第一方面，提供一种代码验证方法，包括步骤：

[0006] 执行代码预验证程序：读取预存代码，对预存代码进行签名并将签名与预存代码中存储的签名进行比较；所述预存代码前端为设备ID和时间戳，所述预存代码还设有签名区，预存有签名；

[0007] 比较一致时执行主程序，否则报警退出；

[0008] 执行代码验证程序：读取预存代码，对预存代码进行签名并将签名与预存代码中存储的签名进行比较；

[0009] 比较一致时结束验证，

[0010] 否则报警退出。

[0011] 第二方面，提供一种可联网的终端设备，所述终端设备包括：

[0012] 一个或多个处理器；

[0013] 存储器，用于存储一个或多个程序，当所述一个活多个程序被所述一个或多个处理器执行时，使得所述一个或多个处理器执行上述代码验证方法。

[0014] 第三方面，提供一种存储有计算机程序的计算机可读存储介质，该程序被处理器执行时实现如上述的代码验证方法。

[0015] 根据本申请实施例提供的技术方案，通过在主程序执行之前对代码进行预验证，首先确定代码的安全性，同时在预验证没有任何问题的情况下启动主程序，在主程序运行的过程中，同步启动对代码的验证程序，通过一个进程负责对运行中的代码进行签名验证，

使得程序运行过程中进行验证守护，通过两段式的启动对代码进行验证和保护，同时代码验证过程中的签名随着时间的变化而变化，增加破解的复杂度。

附图说明

[0016] 通过阅读参照以下附图所作的对非限制性实施例所作的详细描述，本申请的其它特征、目的和优点将会变得更明显：

[0017] 图1为本实施例中代码验证方法流程图；

[0018] 图2为本实施例中执行代码预验证程序方法流程图；

[0019] 图3为本实施例中执行代码验证程序方法流程图；

[0020] 图4为本实施例中可联网的终端设备结构示意图。

具体实施方式

[0021] 下面结合附图和实施例对本申请作进一步的详细说明。可以理解的是，此处所描述的具体实施例仅仅用于解释相关发明，而非对该发明的限定。另外还需要说明的是，为了便于描述，附图中仅示出了与发明相关的部分。

[0022] 需要说明的是，在不冲突的情况下，本申请中的实施例及实施例中的特征可以相互组合。下面将参考附图并结合实施例来详细说明本申请。

[0023] 请参考图1，本实施例提供一种代码验证方法，包括步骤：

[0024] 执行代码预验证程序：读取预存代码，对预存代码进行签名并将签名与预存代码中存储的签名进行比较；所述预存代码前端为设备ID和时间戳，所述预存代码还设有签名区，预存有签名；

[0025] 比较一致时执行主程序，否则报警退出；

[0026] 执行代码验证程序：读取预存代码，对预存代码进行签名并将签名与预存代码中存储的签名进行比较；

[0027] 比较一致时结束验证，

[0028] 否则报警退出。

[0029] 本实施例提供的代码验证方法通过在主程序执行之前对代码进行预验证，首先确定代码的安全性，同时在预验证没有任何问题的情况下启动主程序，在主程序运行的过程中，同步启动对代码的验证程序，通过一个进程负责对运行中的代码进行签名验证，使得程序运行过程中进行验证守护，通过两段式的启动对代码进行验证和保护，同时代码验证过程中的签名随着时间的变化而变化，增加破解的复杂度。

[0030] 其中的预存代码为初始代码，在进行代码的初次烧写时，将设备的ID和时间戳加载在代码的最前端，并自动生成整套代码的签名，并将签名存放在固定存放签名的签名区地址处，其中设备ID录入ID标识区地址处，时间戳录入时间区地址处；该代码也可以是上次程序结束的代码，其中程序中的时间戳会根据代码运行的时间进行改变。本实施例中在执行主程序前和执行主程序的同时均进行代码的验证，保证了代码的准确性，能够及时的并且实时的防止代码被篡改。

[0031] 可选的，所述执行代码预验证程序具体包括：

[0032] 读取预存代码的设备ID和时间戳，存放至第一测试区内，

[0033] 判断预存代码的剩余代码长度是否小于第一长度；

[0034] 若否，则顺次读取第一长度的剩余代码并进行签名，将签名存放至所述第一测试区的前端，

[0035] 若是，则读取剩余代码并进行签名，并将签名存放至所述第一测试区的前端，将所述第一测试区内的签名与预存代码中存储的签名进行比较。

[0036] 如图2所示，本实施例中执行代码预验证程序，首先将代码中的设备ID和时间戳读取出来并预存至第一测试区内，随后对剩余代码进行判断，判断剩余代码的长度是否小于第一长度，其中该步骤是为了判断预存代码是否到达代码末尾段，若剩余的代码还未到达末尾段，则按照预定的情况读取一定长度，此时为第一长度的代码进行签名，并且把签名储存在第一测试区的前端，与代码中的设备ID和时间戳形成一个用于验证的第一测试区数据；上述对剩余代码的判断是贯穿始终的，若未到达代码的末尾段，则顺次对代码进行读取并签名，当剩余的代码不足第一长度时，此时达到代码末尾段，直接将结尾剩余的代码读取出来进行签名即可；

[0037] 其中，上述第一测试区的长度为固定长度，首先将设备ID和时间戳存放至第一测试区，随后将签名后的代码放置在第一测试区的前端，后面代码的签名放置在第一测试区前端的时候覆盖上一次的签名，例如设定第一测试区为长度16字节，8个字节用来存放设备ID和时间戳，8个字节用来存放签名，则用来存放签名的8个字节内的内容是持续覆盖的，后面的结果覆盖前面的结果；

[0038] 其中，上面所说的第一长度是可以设定的长度，可以根据不同的情况进行设定，例如根据具体签名的方式不同选择不同的第一长度，其中签名的方式可以为ASE128加密方式、ASE192加密方式、MD5加密方式等等，其中ASE128加密方式需要设定上述第一长度为16字节，本实施例中优选的采用MD5加密方式，采用该方式进行签名，其中涉及到的第一长度可以根据实际需求进行选择，根据不同情况读取不同长度的代码，能够在一定程度下加快验证时间；

[0039] 上述步骤中，当所有代码读取完毕，并签名存储在第一测试区内后，将第一测试区内的签名与预存代码中存储的签名进行比较，只有在比较一致的情况下才进行主程序的启动，否则认定程序被篡改，需要报警并退出。

[0040] 可选的，所述执行代码验证程序包括：

[0041] 读取预存代码的设备ID和时间戳，存放至第二测试区内，

[0042] 判断预存代码的剩余代码长度是否小于第二长度；

[0043] 若否，则顺次读取第二长度的剩余代码并进行签名，将签名存放至所述第二测试区的前端，

[0044] 若是，则读取剩余代码并进行签名，并将签名存放至所述第二测试区的前端，将所述第二测试区内的签名与预存代码中存储的签名进行比较。

[0045] 如图3所示，本实施例中执行代码验证程序，该程序是在主程序启动后，主程序会启动若干个进程，其中通过一个进程负责对代码进行验证，重新进行签名，该同时进行的进程可以适用于程序更新升级或者防止程序在运行中被篡改。本实施例中的代码验证程序与前面的预验证程序相似，首先将代码中设备ID和时间戳读取出来并预存至第二测试区内，随后对剩余代码进行判断，判断剩余代码的长度是否小于第二长度，其中该步骤是为了判

断预存代码是否到达代码末尾段，若剩余的代码还未到达末尾段，则按照预定的情况读取一定长度，此时为第二长度的代码进行签名，并且把签名储存在第二测试区的前端，与代码中的设备ID和时间戳形成一个用于验证的第二测试区数据；上述对剩余代码的判断是贯穿始终的，若未到达代码的末尾段，则顺次对代码进行读取并签名，当剩余的代码不足第二长度时，此时达到代码末尾段，直接将结尾剩余的代码读取出来进行签名即可；

[0046] 其中，上述第二测试区的长度也为固定长度，其与第一测试区情况相同，首先将设备ID和时间戳存放至第二测试区，随后将签名后的代码放置在第二测试区的前端，后面代码的签名放置在第二测试区前端的时候覆盖上一次的签名，例如设定第二测试区为长度16字节，8个字节用来存放设备ID和时间戳，8个字节用来存放签名，则用来存放签名的8个字节内的内容是持续覆盖的，后面的结果覆盖前面的结果；

[0047] 其中，上面所说的第二长度可以是设定的长度，可以根据不同的情况进行设定，例如根据具体签名的方式不同选择不同的第二长度，与前文中第一长度的情况相同，本实施例中优选的采用MD5加密方式，采用该方式进行签名，其中涉及到的第二长度可以根据实际需求进行选择，根据不同情况读取不同长度的代码，能够在一定程度下加快验证时间，其中第一长度和第二长度可以为相同长度；

[0048] 上述步骤中，当所有代码读取完毕，并签名存储在第二测试区内后，将第二测试区内的签名与预存代码中存储的签名进行比较，只有在比较一致的情况下才结束验证，否则认定程序被篡改，需要报警并退出。

[0049] 可选的，在最终报警退出前还包括步骤：

[0050] 判断当前程序版本是否为预定版本，

[0051] 若是，则报警退出；

[0052] 否则，保存当前程序的签名并更新时间戳。

[0053] 如图3所示，本实施例中的代码验证程序过程中，当第二测试区内的签名与预存的签名不一致时，还可能存在程序版本升级的情况，因此，在最终报警退出之前，对程序的版本进行验证，当确定当前程序版本是预定的版本，即程序版本没有改变时，签名不一致，此时程序有可能被篡改，需要进行报警；若验证程序版本并非为预存的版本，则说明程序可能进行了升级，需要将第二测试区内的签名进行保存并更新其中的时间戳，将最新程序的最近时间更新至签名中，以便于下一次程序启动时进行验证。

[0054] 本实施例还提供一种可联网的终端设备，所述终端设备包括：

[0055] 一个或多个处理器；

[0056] 存储器，用于存储一个或多个程序，当所述一个活多个程序被所述一个或多个处理器执行时，使得所述一个或多个处理器执行上述的盘点方法。

[0057] 本实施例提供装置如图4所示，其中示出了适用于来实现本申请实施例的装置的计算机系统300的结构示意图。

[0058] 如图4所示，计算机系统包括中央处理单元(CPU)301，其可以根据存储在只读存储器(ROM)302中的程序或者从存储部分加载到随机访问存储器(RAM)303中的程序而执行各种适当的动作和处理。在RAM303中，还存储有系统操作所需的各种程序和数据。CPU 301、ROM 302以及RAM 303通过总线304彼此相连。输入/输出(I/O)接口303也连接至总线304。

[0059] 以下部件连接至I/O接口303：包括键盘、鼠标等的输入部分306；包括诸如阴极射

线管(CRT)、液晶显示器(LCD)等以及扬声器等的输出部分;包括硬盘等的存储部分308;以及包括诸如LAN卡、调制解调器等的网络接口卡的通信部分309。通信部分309经由诸如因特网的网络执行通信处理。驱动器也根据需要连接至I/O接口303。可拆卸介质311,诸如磁盘、光盘、磁光盘、半导体存储器等等,根据需要安装在驱动器310上,以便于从其上读出的计算机程序根据需要被安装入存储部分308。

[0060] 特别地,根据本发明的实施例,上文参考流程图1描述的过程可以被实现为计算机软件程序。例如,本申请公开的代码验证的实施例,其包括承载在计算机可读介质上的计算机程序,该计算机程序包含用于执行流程图所示的方法的程序代码。在这样的实施例中,该计算机程序可以通过通信部分从网络上被下载和安装,和/或从可拆卸介质被安装。在该计算机程序被中央处理单元(CPU)301执行时,执行本申请的系统中限定的上述功能。

[0061] 需要说明的是,本发明所示的计算机可读介质可以是计算机可读信号介质或者计算机可读存储介质或者是上述两者的任意组合。计算机可读存储介质例如可以是——但不限于——电、磁、光、电磁、红外线、或半导体的系统、装置或器件,或者任意以上的组合。计算机可读存储介质的更具体的例子可以包括但不限于:具有一个或多个导线的电连接、便携式计算机磁盘、硬盘、随机访问存储器(RAM)、只读存储器(ROM)、可擦式可编程只读存储器(EPROM或闪存)、光纤、便携式紧凑磁盘只读存储器(CD-ROM)、光存储器件、磁存储器件、或者上述的任意合适的组合。在本发明中,计算机可读存储介质可以是任何包含或存储程序的有形介质,该程序可以被指令执行系统、装置或者器件使用或者与其结合使用。

[0062] 而在本发明中,计算机可读的信号介质可以包括在基带中或者作为载波一部分传播的数据信号,其中承载了计算机可读的程序代码。这种传播的数据信号可以采用多种形式,包括但不限于电磁信号、光信号或上述的任意合适的组合。计算机可读的信号介质还可以是计算机可读存储介质以外的任何计算机可读介质,该计算机可读介质可以发送、传播或者传输用于由指令执行系统、装置或者器件使用或者与其结合使用的程序。计算机可读介质上包含的程序代码可以用任何适当的介质传输,包括但不限于:无线、电线、光缆、RF等等,或者上述的任意合适的组合。

[0063] 附图中的流程图和框图,图示了按照本申请显示设备屏幕亮度实时调控装置、方法和计算机程序产品的可能实现的体系架构、功能和操作。在这点上,流程图或框图中的每个方框可以代表一个模块、程序段、或代码的一部分,上述模块、程序段、或代码的一部分包含一个或多个用于实现规定的逻辑功能的可执行指令。也应当注意,在有些作为替换的实现中,方框中所标注的功能也可以以不同于附图中所标注的顺序发生。例如,两个接连地表示的方框实际上可以基本并行地执行,它们有时也可以按相反的顺序执行,这依所涉及的功能而定。也要注意的是,框图或流程图中的每个方框、以及框图或流程图中的方框的组合,可以用执行规定的功能或操作的专用的基于硬件的系统来实现,或者可以用专用硬件与计算机指令的组合来实现。

[0064] 描述于本发明实施例中所涉及到的单元可以通过软件的方式实现,也可以通过硬件的方式来实现,所描述的单元也可以设置在处理器中。其中,这些单元的名称在某种情况下并不构成对该单元本身的限定。所描述的单元或模块也可以设置在处理器中,例如,可以描述为:一种处理器包括第一获取模块、第二获取模块及计算模块。

[0065] 作为另一方面,本申请还提供了一种计算机可读存储介质,该计算机可读存储介

质可以是上述实施例中描述的电子设备中所包含的;也可以是单独存在,而未装配入该电子设备中。上述计算机可读介质承载有一个或者多个程序,当上述一个或者多个程序被一个该电子设备执行时,使得该电子设备实现如上述实施例中所述的代码验证方法,包括步骤:

[0066] 执行代码预验证程序:读取预存代码,对预存代码进行签名并将签名与预存代码中存储的签名进行比较;所述预存代码前端为设备ID和时间戳,所述预存代码还设有签名区,预存有签名;

[0067] 比较一致时执行主程序,否则报警退出;

[0068] 执行代码验证程序:读取预存代码,对预存代码进行签名并将签名与预存代码中存储的签名进行比较;

[0069] 比较一致时结束验证,

[0070] 否则报警退出。

[0071] 应当注意,尽管在上文详细描述中提及了用于动作执行的设备的若干模块或者单元,但是这种划分并非强制性的。实际上,根据本公开的实施方式,上文描述的两个或更多模块或者单元的特征和功能可以在一个模块或者单元中具体化。反之,上文描述的一个模块或者单元的特征和功能可以进一步划分为由多个模块或者单元来具体化。

[0072] 此外,尽管在附图中以特定顺序描述了本公开中方法的各个步骤,但是,这并非要求或者暗示必须按照该特定顺序来执行这些步骤,或是必须执行全部所示的步骤才能实现期望的结果。附加的或备选地,可以省略某些步骤,将多个步骤合并为一个步骤执行,以及/或者将一个步骤分解为多个步骤执行等。

[0073] 通过以上的实施方式的描述,本领域的技术人员易于理解,这里描述的示例实施方式可以通过软件实现,也可以通过软件结合必要的硬件的方式来实现。

[0074] 作为另一方面,本申请还提供了一种计算机可读介质,该计算机可读介质可以是上述实施例中描述的电子设备中所包含的;也可以是单独存在,而未装配入该电子设备中。上述计算机可读介质承载有一个或者多个程序,当上述一个或者多个程序被一个该电子设备执行时,使得该实现如上述实施例中所述的代码验证方法。

[0075] 以上描述仅为本申请的较佳实施例以及对所运用技术原理的说明。本领域技术人员应当理解,本申请中所涉及的发明范围,并不限于上述技术特征的特定组合而成的技术方案,同时也应涵盖在不脱离所述发明构思的情况下,由上述技术特征或其等同特征进行任意组合而形成的其它技术方案。例如上述特征与本申请中公开的(但不限于)具有类似功能的技术特征进行互相替换而形成的技术方案。

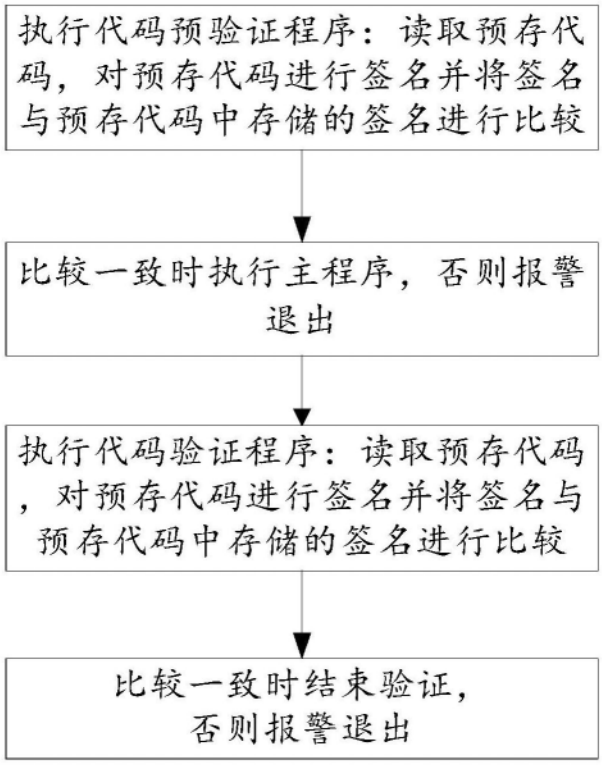

执行代码预验证程序：读取预存代码，对预存代码进行签名并将签名与预存代码中存储的签名进行比较
比较一致时执行主程序，否则报警退出
执行代码验证程序：读取预存代码，对预存代码进行签名并将签名与预存代码中存储的签名进行比较
比较一致时结束验证，否则报警退出

图1

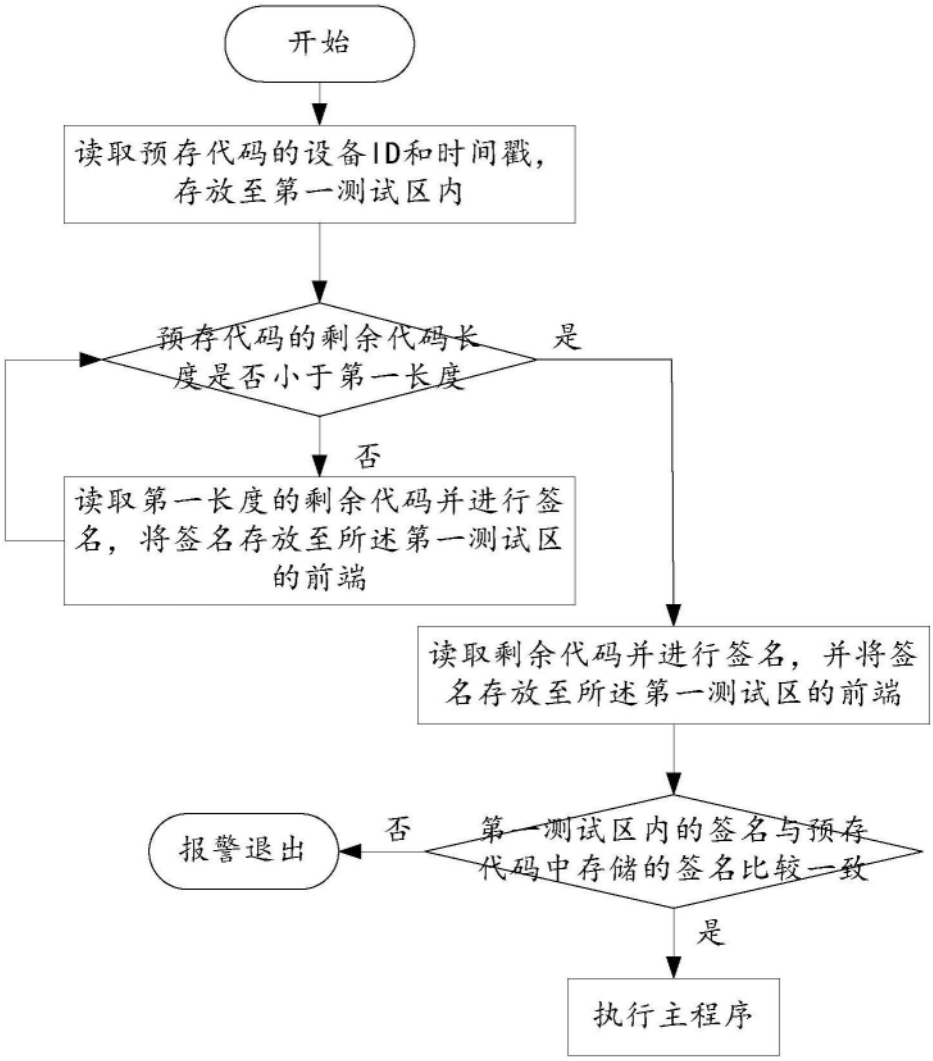

开始
读取预存代码的设备ID和时间戳，存放至第一测试区内
预存代码的剩余代码长度是否小于第一长度
是
否
读取第一长度的剩余代码并进行签名，将签名存放至所述第一测试区的前端
读取剩余代码并进行签名，并将签名存放至所述第一测试区的前端
第一测试区内的签名与预存代码中存储的签名比较一致
否
报警退出
是
执行主程序

图2

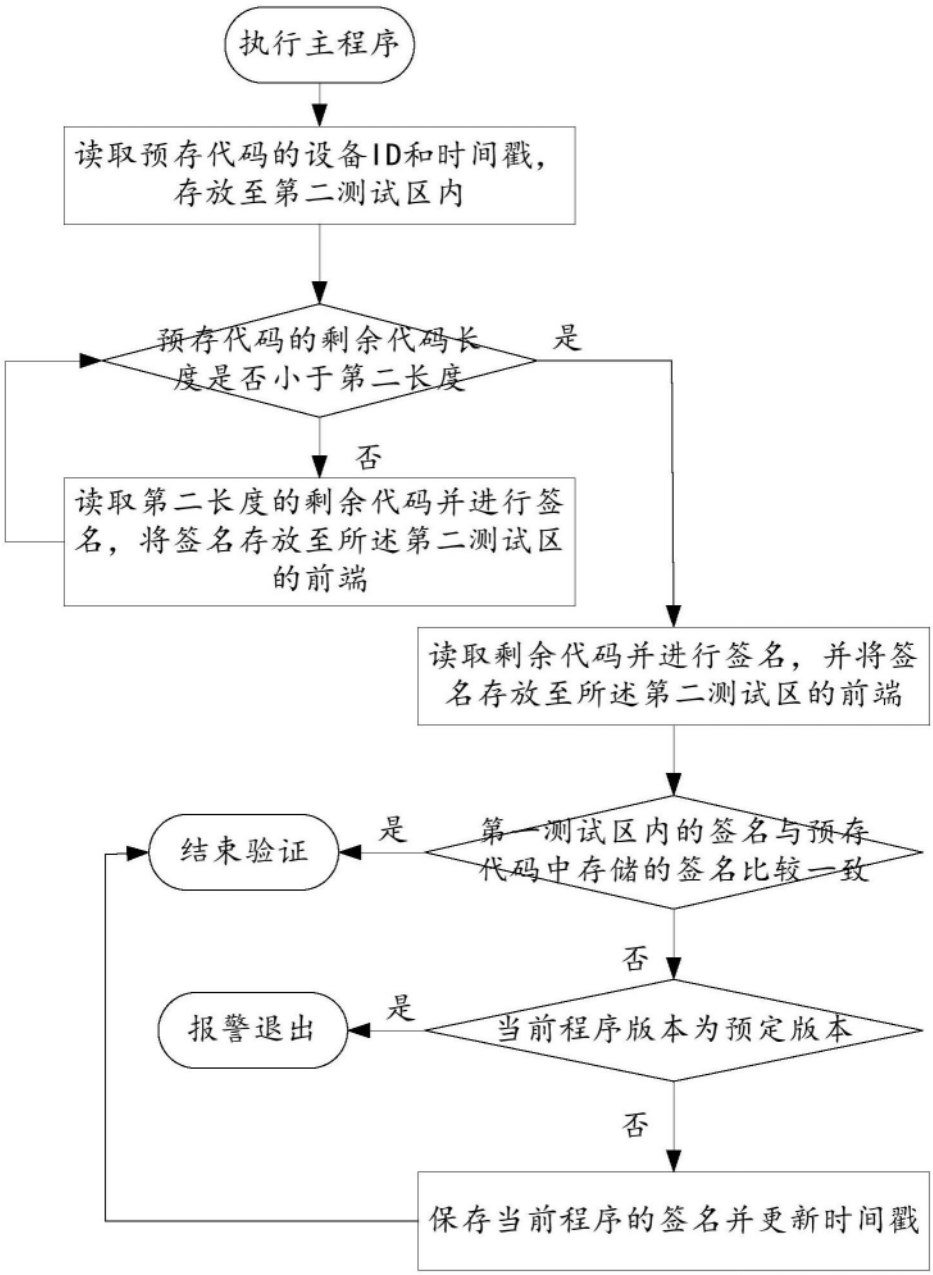

执行主程序
读取预存代码的设备ID和时间戳，存放至第二测试区内
预存代码的剩余代码长度是否小于第二长度
是
否
读取第二长度的剩余代码并进行签名，将签名存放至所述第二测试区的前端
读取剩余代码并进行签名，并将签名存放至所述第二测试区的前端
第一测试区内的签名与预存代码中存储的签名比较一致
是
结束验证
否
当前程序版本为预定版本
是
报警退出
否
保存当前程序的签名并更新时间戳

图3

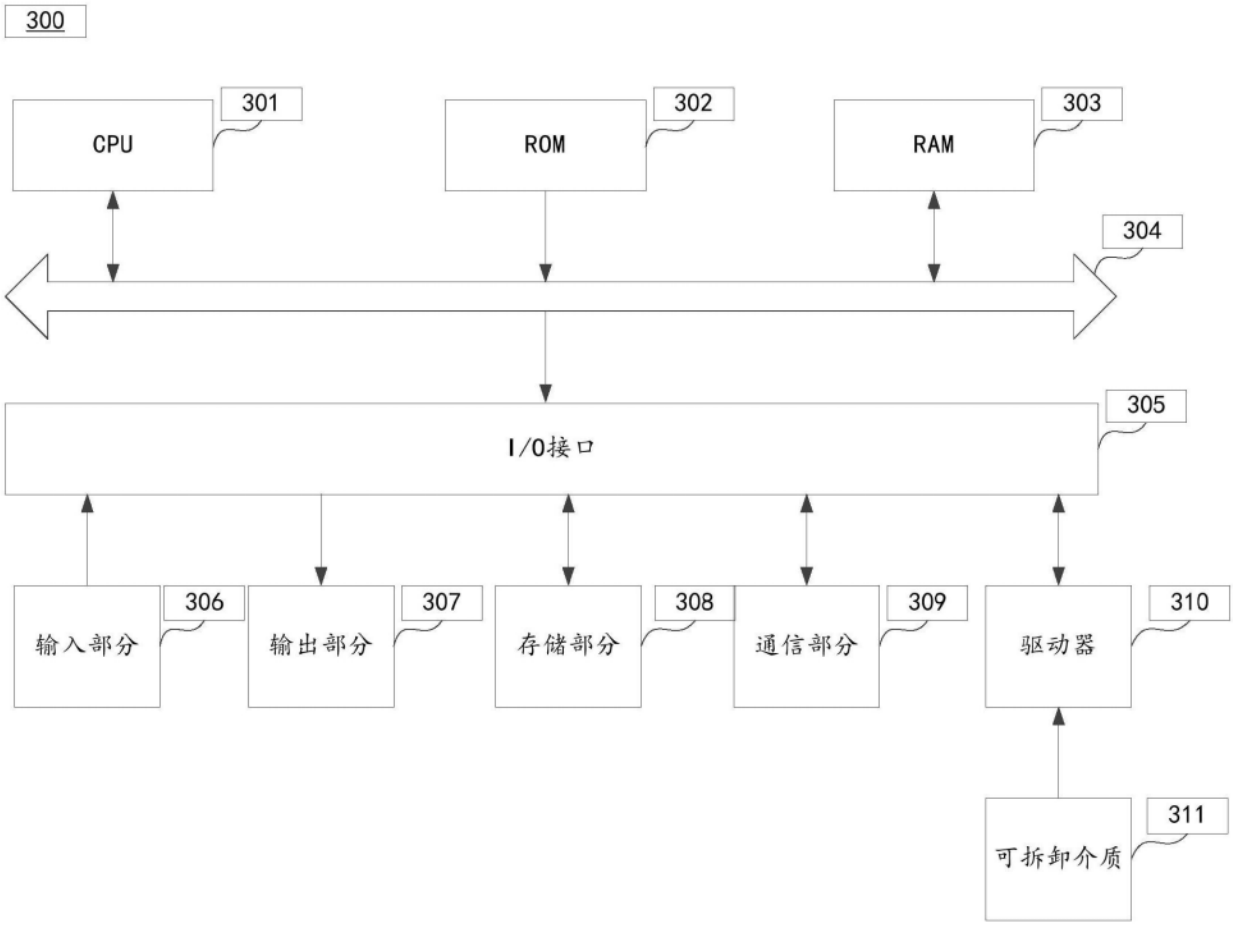

300
CPU
301
ROM
302
RAM
303
304
I/O接口
305
输入部分
306
输出部分
307
存储部分
308
通信部分
309
驱动器
310
可拆卸介质
311

图4