(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 112395587 B
(45)授权公告日 2024.05.24

(21)申请号 202010052436.0

(22)申请日 2020.01.17

(65)同一申请的已公布的文献号
申请公布号 CN 112395587 A

(43)申请公布日 2021.02.23

(30)优先权数据
16/541,218 2019.08.15 US

(73)专利权人 新唐科技股份有限公司
地址 中国台湾新竹科学工业园区

(72)发明人 尤佛·科斯纳尔

(74)专利代理机构 北京三友知识产权代理有限公司 11127
专利代理师 赵平 周永君

(51)Int.Cl.
G06F 21/44(2013.01)
G06F 21/45(2013.01)

(56)对比文件
CN 103105783 A,2013.05.15
EP 1659472 A1,2006.05.24
JP 2003162511 A,2003.06.06
JP 2008112443 A,2008.05.15
US 2017372056 A1,2017.12.28

审查员 庄鑫

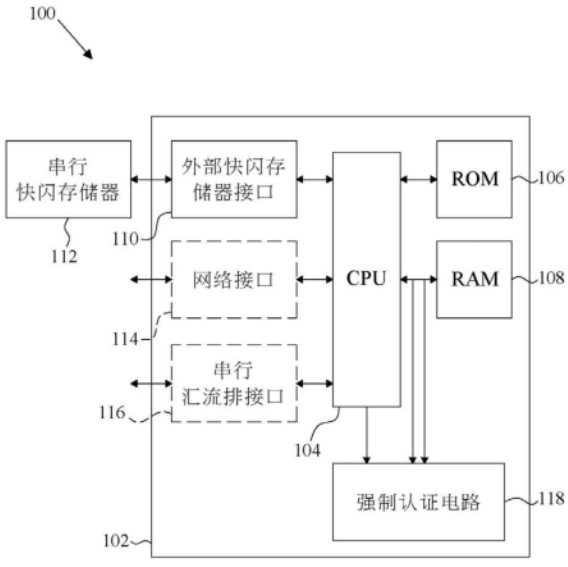
权利要求书2页 说明书7页 附图4页

(54)发明名称

计算机系统及强制自行认证方法

(57)摘要

本发明揭露一种计算机系统及强制自行认证方法,该计算机系统包含存储器、处理器和认证执行硬件。处理器是执行软件,该软件包含对存储在存储器中的数据进行认证的认证程序。认证执行硬件耦接处理器,并且用以验证(i)处理器以至少一指定频率周期性地执行认证程序,以及(ii)认证程序成功认证数据。



CN 112395587 B

1.一种计算机系统,其特征在于,包括:

一存储器;

一处理器,用以执行软件,所述软件包含对存储在所述存储器中的数据进行认证的一认证程序;以及

一认证执行硬件,其耦接所述处理器并用以验证是否所述处理器以至少一指定的频率定期执行所述认证程序,以及验证是否所述认证程序成功认证所述数据;

所述认证执行硬件包括:一计时器、比较器、逻辑门和使能认证指示正反器,所述比较器用以比较计时器输出的时间与一预设门槛值,当此时间等于门槛值时,比较器产生一重置信号;所述逻辑门用以在所述使能认证指示正反器被设定时,将认证OK指示从中央处理单元传递到计时器;

所述处理器被配置为从只读存储器执行所述认证程序,以及所述认证执行硬件检测是否从所述只读存储器提取了认证程序的指令以及所述处理器是否是从所述只读存储器中以至少一指定的频率定期执行所述认证程序;

所述计时器输出的时间包括:收到两次认证OK指示之间的时间;

所述认证执行硬件包括ROM指令提取检测器和RAM指令提取检测器,用于监测所述处理器访问所述存储器的所述ROM指令提取检测器检测到所述处理器从所述只读存储器提取指令时,所述使能认证指示正反器才被设置,并且其中,仅当所述RAM指令提取检测器检测到所述处理器从随机存取存储器提取指令时,所述使能认证指示正反器才被重置,并且其中,仅通过在从所述只读存储器提取指令之后和从所述随机存取存储器提取指令之前的所述认证OK指示重置所述计时器。

2.根据权利要求1所述的计算机系统,其特征在于,所述认证执行硬件用以:当所述处理器未能以所述至少一指定频率执行所述认证程序时,启动一回应措施。

3.根据权利要求1所述的计算机系统,其特征在于,所述认证执行硬件用以:当所述认证程序无法认证所述数据时,启动一回应措施。

4.根据权利要求1所述的计算机系统,其特征在于,所述认证程序在成功认证所述数据时指示所述处理器设定一信号有效,并且所述认证执行硬件包含一计时器用以验证所述信号是否以至少所述指定频率被设定有效。

5.根据权利要求1所述的计算机系统,其特征在于,所述处理器用以从一只读存储器执行所述认证程序,并且只有当验证从所述只读存储器执行一既定执行时,所述认证执行硬件才决定所述认证程序的所述既定执行成功完成。

6.根据权利要求5所述的计算机系统,其特征在于,所述认证执行硬件藉由检测所述认证程序的命令是否从所述只读存储器取得,以验证所述既定执行是否是从只读存储器中执行。

7.一种强制自行认证方法,其特征在于,应用权利要求1至6任一项所述的计算机系统实现,该方法包括:

使用一处理器执行软件,其中所述软件包含一认证程序用以对存储在一存储器中的数据进行认证;以及

使用耦接所述处理器的一认证执行硬件,验证是否所述处理器以至少一指定频率定期执行所述认证程序,以及验证是否认证程序成功认证所述数据;

所述认证执行硬件包括:一计时器、比较器、逻辑门和使能认证指示正反器,所述比较器比较计时器输出的时间与一预设门槛值,当此时间等于门槛值时,比较器产生一重置信号;所述逻辑门在所述使能认证指示正反器被设定时,将认证OK指示从中央处理单元传递到计时器。

8.根据权利要求7所述的强制自行认证方法,其特征在于,还包括:

当所述处理器未能以所述至少一指定频率执行所述认证程序时,由所述认证执行硬件启动一回应措施。

9.根据权利要求7所述的强制自行认证方法,其特征在于,还包括:

当所述认证程序未能认证所述数据时,由所述认证执行硬件启动一回应措施。

10.根据权利要求7所述的强制自行认证方法,其特征在于,还包括:所述认证程序指示所述处理器在成功认证所述数据时设定一信号有效;

相对应的,验证所述处理器以所述至少一指定频率执行所述认证程序的步骤还包括:

使用所述认证执行硬件的一计时器,其中所述信号是以所述至少一指定频率被设定为有效。

11.根据权利要求7所述的强制自行认证方法,其特征在于,执行所述软件的步骤包括:

从一只读存储器执行所述认证程序;

相对应的,其中验证所述认证程序成功认证所述数据的步骤包括:

验证所述认证程序的一既定执行是否从所述只读存储器执行,若是,才决定所述认证程序的所述既定执行成功完成。

12.根据权利要求11所述的强制自行认证方法,其特征在于,验证所述既定执行是否从所述只读存储器执行的步骤包括:

检测所述认证程序的命令是否是从所述只读存储器取得。

计算机系统及强制自行认证方法

技术领域

[0001] 本发明是有关于一种安全计算环境,特别是有关于强制一计算机系统进行自行验证的方法以及系统。

背景技术

[0002] 计算机系统通常包括一个或多个中央处理器(CPU)和存储器,其中CPU执行存储在存储器中的软件程序。在某些计算机系统中,计算机使用加密技术对计算机执行的软件进行验证。

[0003] 例如在2003年3月28日发行的“安全要求的加密模块实施指南(SECURITY REQUIREMENTS FOR CRYPTOGRAPHIC MODULES,Implementation Guidelines)”,NIST-FIPS 140-2;2008年7月发行的FIPS PUB 198-1中的“密钥散列消息认证码(The Keyed-Hash Message Authentication Code)”;2015年8月发行的NIST-FIPS 180-4“安全哈希标准(SHS)”;以及,2017年8月发行的“统一可扩展固件接口(Unified Extensible Firmware Interface,UEFI)规范”,版本2.7(勘误A),皆描述验证固件(和其他软件或数据)真伪的方法。

发明内容

[0004] 根据一实施例,本发明提出一种计算机系统,包含:一存储器;一处理器,用以执行软件,该软件包含对存储在该存储器中的数据进行认证的一认证程序;以及一认证执行硬件,其耦接该处理器并用以验证是否该处理器以至少一指定的频率定期执行该认证程序,以及验证是否该认证程序成功认证该数据。

[0005] 根据一实施例,认证执行硬件用以:当该处理器未能以该至少一指定频率执行该认证程序时,启动一回应措施。根据一实施例,认证执行硬件用以:当该认证程序无法认证该数据时,启动一回应措施。

[0006] 根据一实施例,认证程序在成功认证该数据时指示该处理器设定一信号有效,并且该认证执行硬件包含一计时器用以验证该信号是否以至少该指定频率被设定有效。

[0007] 根据一实施例,处理器用以从一只读存储器执行该认证程序,并且只有当验证从该只读存储器执行一既定执行时,该认证执行硬件才决定该认证程序的该既定执行成功完成。根据一实施例,认证执行硬件藉由检测该认证程序的命令是否从该只读存储器取得,以验证该既定执行是否是从只读存储器中执行。

[0008] 根据一实施例,本发明提出一种强制自行认证方法,包含:使用一处理器执行软件,其中该软件包含一认证程序用以对存储在一存储器中的数据进行认证;使用耦接该处理器的一认证执行硬件,验证是否该处理器以至少一指定频率定期执行该认证程序,以及验证是否认证程序成功认证该数据。

附图说明

[0009] 图1为示意性地示出根据本发明实施例的具有强制认证的计算机系统的方块图。

[0010] 图2为根据本发明一实施例示意性地示出当自行认证失败时对图1的计算机系统进行保护的时序波形图。

[0011] 图3为根据本发明一实施例示意性地示出当软件无法执行自行认证软件时图1的计算机系统进行安全保护的时序波形图。

[0012] 图4为根据本发明一实施例示意性绘示在计算机系统中强制自行认证的电路的结构。

[0013] 上述图式为示意性且并未按比例缩放。图式中相对尺寸与比例因精确与/或方便的目的而放大或缩小,且尺寸为任意的且不限于此。于图式中相似的参考符号代表相似的元件。

[0014] 符号说明

[0015] 100:方块图

[0016] 102:计算机系统

[0017] 104:中央处理单元

[0018] 106:只读存储器

[0019] 108:随机存取存储器

[0020] 110:外部快闪存储器接口

[0021] 112:串行快闪存储器

[0022] 114:网络接口

[0023] 116:串行汇流排接口

[0024] 118:强制认证电路

[0025] 200、300:时序波形图

[0026] 202、204、206、208、302、304、306、308:波形

[0027] 210、212、214、216、310:时间指示

[0028] 400:方块图

[0029] 402:计时器

[0030] 404:比较器

[0031] 406:逻辑门

[0032] 408:使能认证指示正反器

[0033] 410:ROM指令提取检测器

[0034] 412:RAM指令提取检测器

具体实施方式

[0035] 以下将配合图式及实施例来详细说明本发明的实施方式,藉此对本发明如何应用技术手段来解决技术问题并达成技术功效的实现过程能充分理解并据以实施。

[0036] 当在此使用时,除非文中另行明确地表示,否则“一”、“该”、“此”等单数型式亦旨在包含复数型式。

[0037] 计算机系统(具体而言,微控制器)通常包含随机存取存储器(RAM),其存储固件代

码(FW)和数据。例如,存储在一只读存储器(ROM)中的嵌入式开机引导程序(bootloader)可以从串行快闪存储器下载固件代码,然后从随机存取存储元件中执行。在其它实施例,固件代码可以通过并行或是串行汇流排,以有线或无线的方式,从网络或从其它外部来源下载。

[0038] 嵌入式开机引导程序可对其下载的固件代码进行认证,例如,使用加密技术,像是基于金钥的签章;当ROM代码被认为是安全的,则下载的固件代码是可信赖的。然而,一旦下载后,有时候只要供应给计算机系统的电源不中断,固件代码可能会执行很长的时间,例如几个月或几年。一旦黑客破坏计算机系统安全并载入一修改过的代码,例如,黑客可对芯片的供应电源产生一突波、或是使能一除错端口、或是直接注入数据至随机存取存储元件,则上述修改过的代码会被永远执行,变成严重的安全隐患。以下段落中,上述修改过的固件代码会称为"恶意固件代码"或是"恶意代码"。

[0039] 本发明的实施例会揭露一种方法以及系统,用以强制一计算机系统周期地自行验证,藉此降低恶意固件代码取代真实固件代码而长时间执行的风险。在一实施例中,固件代码包含认证功能,其用于周期地(例如,藉由一可编程计时器来触发)认证存储器内容或是其一部分。如果认证失败,计算机系统发出一重置信号;或是,在一实施例中,发出不可屏蔽的中断(NMI);或是,在另一实施例中,计算机系统停止执行。接着,计算机系统将重开机,废弃储存在随机存取存储元件中的固件代码。

[0040] 在一些实施例中,计算机系统包含强制认证电路(forced-authentication circuit,FAC),其亦称为认证执行硬件,用以验证(i)固件代码有周期性且足够频率地执行认证功能,以及(ii)认证成功完成。在一范例实施例,如果认证程序成功完成(也就是说,固件代码或是其一部分通过认证),则固件代码设定一认证OK信号(其通常是暂存器中的一位)有效。在此例中,强制认证电路包含一计时器,当认证OK信号设定有效时,计时器重置;实际上,计时器会从最后一次认证成功一直计数时间。如果计时器达到一预设门槛值,则强制认证电路会强制计算机系统重开机,例如,强制认证电路可发出重置信号或是一NMI。

[0041] 在实施例中,强制认证电路不可由软件存取,软件只能设定认证OK信号。在一些实施例中,认证功能储存在只读存储器(ROM),固件代码周期地调用认证功能。在一实施例中,为了防止不是成功认证的伪造认证OK信号,如果最后一个命令不是从只读存储器取得的,强制认证电路会忽视认证OK信号;因此,如果储存在必定安全的只读存储器中的认证功能表示成功,才可接受认证OK信号。

[0042] 虽然以上说明是有关于固件代码的认证,但本发明的实施例不限于认证全部固件代码,而可仅认证部分固件代码,或是可认证计算机系统的存储器中的数据。

[0043] 综上所述,本发明的实施例可强制计算机系统周期地认证储存在随机存取存储元件的数据(例如全部固件代码)。计算机系统中的电路可确保当在一预设时间门槛值中随机存取存储元件中的数据认证失败会造成系统重置、NMI、或者停止固件代码执行、或是其他回应措施。除了基于ROM的固件代码可通知认证成功,因为固件代码不可存取此电路,所以固件代码无法伪造一成功认证。除了周期地调用认证功能,上述新增机制不会影响使用者软件性能。

[0044] SYSTEM DESCRIPTION

[0045] 根据本发明的实施例,本发明提出几个计算机系统及其元件的范例。应注意的是,本发明的范围不受此些范例的限制。

[0046] 图1为根据本发明一实施例示意性绘示具有强制认证功能的计算机系统的方块图100。

[0047] 计算机系统102包含一中央处理单元(CPU)104(其亦称为处理器),用以执行储存在存储器的程序;一只读存储器(ROM)106,其用以储存初始启动代码、以及其他功能与数据,例如信任固件代码功能;以及一随机存取存储器(RAM)108,其用以储存固件代码以及数据。只读存储器106以及随机存取存储器108可共同称为计算机系统存储器。

[0048] 为了下载固件代码,计算机系统102还可包含一外部快闪存储器接口110,其用以与串行快闪存储器112进行通讯,串行快闪存储器112可在计算机系统外部储存固件代码。可选地,计算机系统102可包含其他用于下载固件代码的接口,例如网络接口114,其用于计算机系统以及网络(例如,以太网)之间的通讯,并从网络下载固件代码;一串行汇流排接口116,用以进行计算机系统以及外部装置在串行汇流排(例如,集成电路汇流排(I^2C))上的通讯,且在串行汇流排上下载固件代码。可选地,计算机系统102可包含其他接口用以从外部来源下载固件代码。将在以下段落提供外部来源的范例。

[0049] 根据本发明的实施例,当计算机系统102载入固件代码,计算机系统会使用例如加密签章,以认证固件代码(或是其一部分)。认证程序(至少一部分)通常是储存在只读存储器106中,如果认证失败,则计算机系统将不载入固件代码,例如计算机系统可停止执行、重置系统、或是产生不可屏蔽的中断(NMI)。如果认证成功,中央处理单元104会载入固件代码至随机存取存储器108,接着,中央处理单元104从随机存取存储器执行此固件代码。此时可断开串行快闪存储器112。

[0050] 在计算机系统的一些应用中,中央处理单元可长时间执行此固件代码。例如,在生产车间的计算机系统可下载稳定且成熟的程序控制固件代码,接着执行此固件代码,如果电力不中断,可能会执行几个月或是几年。在这段长时间内黑客会有机会攻击计算机系统,并改变固件代码;例如,黑客可能会在供电源输入突波、使能一除错端口、或是直接注入数据至随机存取存储器。因此,在固件代码载入之前由计算机系统执行固件代码的认证是不够的。

[0051] 为了减轻此风险,在本发明的实施例中,计算机系统执行的固件代码必须包含对随机存取存储器数据的周期性认证。例如,每10秒进行一次认证,而根据性能与花费在认证上的电力,以及安全隐患以及恢复时间之间的取舍,实际的次数可以从几秒到几小时。在一些实施例中,认证率不是固定的,但是必须界定计算机系统执行两次认证之间的最大时间。

[0052] 然而,黑客可禁能此周期性认证而让恶意的固件代码能长时间执行。为了减轻此风险,计算机系统102还包含一强制认证电路(FAC)118。每当一认证成功执行完成,强制认证电路118从中央处理单元104接收表示认证成功的信号。强制认证电路可包含一计时器,并在不会比预设门槛值更长的区间中验证新的认证执行为中央处理单元发出信号。

[0053] 根据搭配图1描述的实施例,认证程序(至少一部分)储存在只读存储器106,并藉由调用基于只读存储器的功能在随机存取存储器中执行固件代码。强制认证电路更用以监控中央处理单元对存储器的存取,且除非一成功认证的指示是由基于只读存储器的指令执行结果,否则强制认证电路会阻挡此认证成功的指示。因此,黑客无法伪造认证成功执行的指示,且强制认证电路118将及时检测到认证失败。在一预先定义时间周期中,当认证失败或无法执行认证时,将结束恶意固件代码的执行。

[0054] 可以理解的是,图1所示的运算系统102的实施例为一实施例范例。根据本发明所揭露的技术的运算系统不限于上述举例说明。在其他实施例,例如,可使用任何外部快闪存储器接口110、网络接口114以及串行汇流排接口116,以从串行快闪存储器及/或网络及/或串行汇流排下载固件代码。在一些实施例中,固件代码可经由适当接口以无线方式下载;在其他的实施例,固件代码可通过快速系统汇流排下载,例如通过快捷外设互联标准(PCIe);以及每一个接口110、114、116可与多个装置连接。

[0055] 在实施例中,中央处理单元104可为一个以上相同或是不同型式的中央处理单元的组合;只读存储器106及/或随机存取存储器108可包含多个只读存储器/随机存取存储器实体。

[0056] 图2为根据本发明一实施例的当自行认证失败时运算系统进行保护的时序波形图200。时序波形包含固件执行波形202,其绘示各种固件代码执行来源;一认证结果波形204,其表示认证执行失败或是成功。计时器波形206,其表示用于验证重复认证执行的计时器的操作;以及一重置波形208,其表示计算机系统的重置操作。

[0057] 时序波形图200还包含时间指示210以及212,用以分别表示认证执行开始以及停止;时间指示214,用以表示中央处理单元开始执行恶意代码的时间点;以及一时间指示216,其用以表示回应认证失败而计算机系统重置。

[0058] 一开始,从随机存取存储器执行固件代码。接着,在时间指示210,固件代码调用储存在只读存储器中的认证程序。在时间指示212,认证完成,且固件代码产生一认证OK信号,其传送至图1的强制认证电路118。

[0059] 当强制认证电路接收此认证OK信号时,计时器重复地增加计数且重置以测量经过时间(elapsed time)。当新信号总是能及时接收到时,计时器不会达到门槛值。

[0060] 上述顺序包含从随机存取存储器执行固件代码,接着从只读存储器执行认证,认证通过指示重复三次,直到在时间指示214,固件代码开始执行被篡改的(恶意)代码。下一次认证软件执行时将会产生认证失败信号(在时间指示216),因此计算机系统重置。

[0061] 图3示意性绘示根据本发明一实施例的当软件无法执行自行认证软件时对运算系统进行保护的时序波形图300。

[0062] 时序波形图300一开始类似时序波形图200,在时间指示210之间两者是相同的。然而,载入至随机存取存储器的恶意固件代码在时间指示214不调用储存在只读存储器中的认证功能。因此,没有认证OK指示产生,计时器不会重置,而计时器在时间指示310达到门槛值。强制认证电路接着产生重置信号,使得计算机系统重启动。

[0063] 综上所述,根据搭配图2与图3描述的范例实施例,如果在时间指示214下载的固件代码不是正确的,则此固件代码会造成随机存取存储器数据的周期性认证失败或是无法在预设门槛值到达之前认证成功,而在认证失败或是计时器达到门槛值时,计算机系统将直接重置或是由强制认证电路重置。

[0064] 可以理解的是,图2与图3所绘示的运算系统的波形仅为范例实施例。本发明所揭露的技术的计算机系统的波形并不限于上述举例。在其他实施例,例如,认证程序可分开成多区段,而在此些区段之间,从随机存取存储器执行固件代码(例如,在必须有快速回应时间(fast response time)的应用中,其无法让固件代码为了一完整认证程序中而停止执行。在其他的实施例中,强制认证电路118不产生重置信号,而是强制认证电路可停止所有

中央处理单元的执行;在一实施例中,强制认证电路产生一NMI,而在另一实施例中,如果认证失败,强制认证电路可产生重置信号,而如果计时器达到门槛值,则强制认证电路产生NMI。

[0065] 再者,如果(i)认证程序没有在至少指定频率下被调用,或是(ii)如果认证程序的某次调用没有成功完成认证,则强制认证电路118可启动其他任何适当的回应措施。

[0066] 图4示意性绘示根据本发明一实施例的计算机系统中强制自行认证的电路结构的方块图400。中央处理单元104与只读存储器106以及随机存取存储器108进行通讯。强制认证电路118用以监控中央处理单元、只读存储器以及随机存取存储器之间的数据处置。强制认证电路亦经由认证OK线(AUTHENTICATION-OK wire,图中标示为AUTH.-OK)耦接于中央处理单元104,中央处理单元可使用认证OK线以表示此认证软件成功完成。

[0067] 强制认证电路118包含一计时器402,其用以计数收到两次认证OK信号之间的时间(例如,计数固定频率时脉信号的周期)。以及一比较器(CMP)404,其用以比较计时器402输出的时间与一预设门槛值(图中标示为THRESHOLD),当此时间等于门槛值时,比较器404产生一重置信号(图中标示为RESET)。在图4的范例实施例中,认证OK信号为中央处理单元104的其中一IO接脚,而藉由中央处理单元的一输出指令来表示认证OK信号。

[0068] 恶意固件代码可能会尝试愚弄上述的强制认证机制,例如,恶意固件代码可周期地设定认证OK指示。如果只有当储存在只读存储器的指令被执行而使得中央处理单元表示认证OK,才会让计时器402重置,则上述风险可以得到解决。

[0069] 强制认证电路118还包含一逻辑门406(图中显示为与(AND)门);一使能认证指示正反器408;ROM指令提取检测器410以及RAM指令提取检测器412。只有当使能认证指示正反器408被设定时,逻辑门406才能将认证OK指示从中央处理单元104传递到计时器402。当用以监控中央处理单元对存储器存取的ROM指令提取检测器410检测到中央处理单元从只读存储器提取一指令时,正反器才会被设定(图中标示为SET),当RAM指令提取检测器412检测到中央处理单元从随机存取存储器提取指令时,正反器被重置(图中标示为CLEAR)。因此,只有从只读存储器提取指令之后以及从随机存取存储器提取指令之前的认证OK指示能让计时器402重置(图中标示为CLEAR)。

[0070] 在计算机系统102的一些实施例中,执行管路可能会造成写入动作延迟,相对于对应的指令提取。因此,对认证OK指示设定有效的只读存储器指令可藉由至少一指令延迟对应的指令提取,如果下一指令是从随机存取存储器执行,则可阻止计时器重置。在那些实施例中,认证软件必须继续从只读存储器执行几个周期,例如执行预设数量的NOP指令,直到执行管路清空。

[0071] 在再一实施例中,ROM指令提取检测器410用以设定一"使能认证辨识旗标",以回应对只读存储器中认证常用程序的第一位址的提取。因此,恶意固件代码不能跳至常用程序(其用以设定认证OK信号有效)的尾部,因此只读存储器常用程序可完整执行。

[0072] 可以理解的是,图4的强制认证电路118的实施例为一举例式实施例。本发明所揭露的技术的强制认证电路不限于上述举例。在其他实施例,例如,中央处理单元104可写入一存储器地址(指向既有存储器或不存在的存储器)以表示认证OK信号。在此实施例,强制认证电路118包含一认证OK检测器,其监控中央处理单元对存储器的存取以及检测认证OK信号。此认证OK检测器的输出信号是输入逻辑门406,以来自中央处理单元104的认证OK线。

在一实施例中,CMP404产生NMI,而不产生重置信号;在另一实施例中,CMP404产生停止信号,以停止中央处理单元。

[0073] 因此,根据本发明的上述实施例,长时间从随机存取存储器执行固件代码的计算机系统可保护防范固件代码被未经授权的修改。此保护包含,第一,固件代码(包含全部固件代码或是其一部分)必须周期地认证储存在随机存取存储器中的数据;第二,认证软件(至少一部分)储存在只读存储器中,相对而言不易受黑客攻击;第三,计算机中的电路包含一计时器,如果认证延迟一预设门槛值以上,则此电路重置、停止或是中断中央处理单元;第四,电路藉由验证认证OK指示是由基于只读存储器的指令所启动,以保护认证OK指示不受黑客攻击。

[0074] 可以理解的是,图1至图4中计算机系统以及强制认证电路的实施例为举例式实施例。本发明所揭露的技术的计算机系统以及强制认证电路不受上述举例说明的限制。在其他实施例,例如,固件代码是从外部快闪存储器(而非随机存取存储器)执行,因此其需要周期地自行验证。随机存取存储器108可为静态或是动态存储器、嵌入式或是外部存储器。中央处理单元104可为任何一种微控制器(例如,RISC或CISC)、或是多个处理器。

[0075] 在本发明的一些实施例中,中央处理单元可包含一快取存储器,用以频繁存取数据;在此些实施例中,认证软件的至少一部分通常在非快取模式下执行。

[0076] 计算机系统102或是其元件可用任何适当的硬件来实现,例如特殊应用集成电路(ASIC)、或是受保护的现场可编程逻辑门阵列(FPGA)。在一些实施例中,控制器的一些或是全部元件可用软件、硬件或是硬件与软件的组合来实现。

[0077] 通常,中央处理单元104包含一通用处理器,其可用软件编程以执行本发明上述的功能。此软件可通过网络以电子信号形式下载至处理器,例如,此软件可提供及/或储存在非暂时性有形媒体上,例如磁性存储器、光学存储器或是电子存储器。

[0078] 只读存储器106可用其他类型存储器来仿真,例如使用快闪存储器、随机存取存储器或是一次性可编程存储器,其具有写入/抹除禁能逻辑电路,因此可仿真不能修改的只读存储器。

[0079] 虽然本发明以前述的实施例揭露如上,然其并非用以限定本发明,任何本领域相关技术人员,在不脱离本发明的精神和范围内,当可作些许的更动与润饰,因此本发明的专利保护范围须视权利要求书所界定者为准。

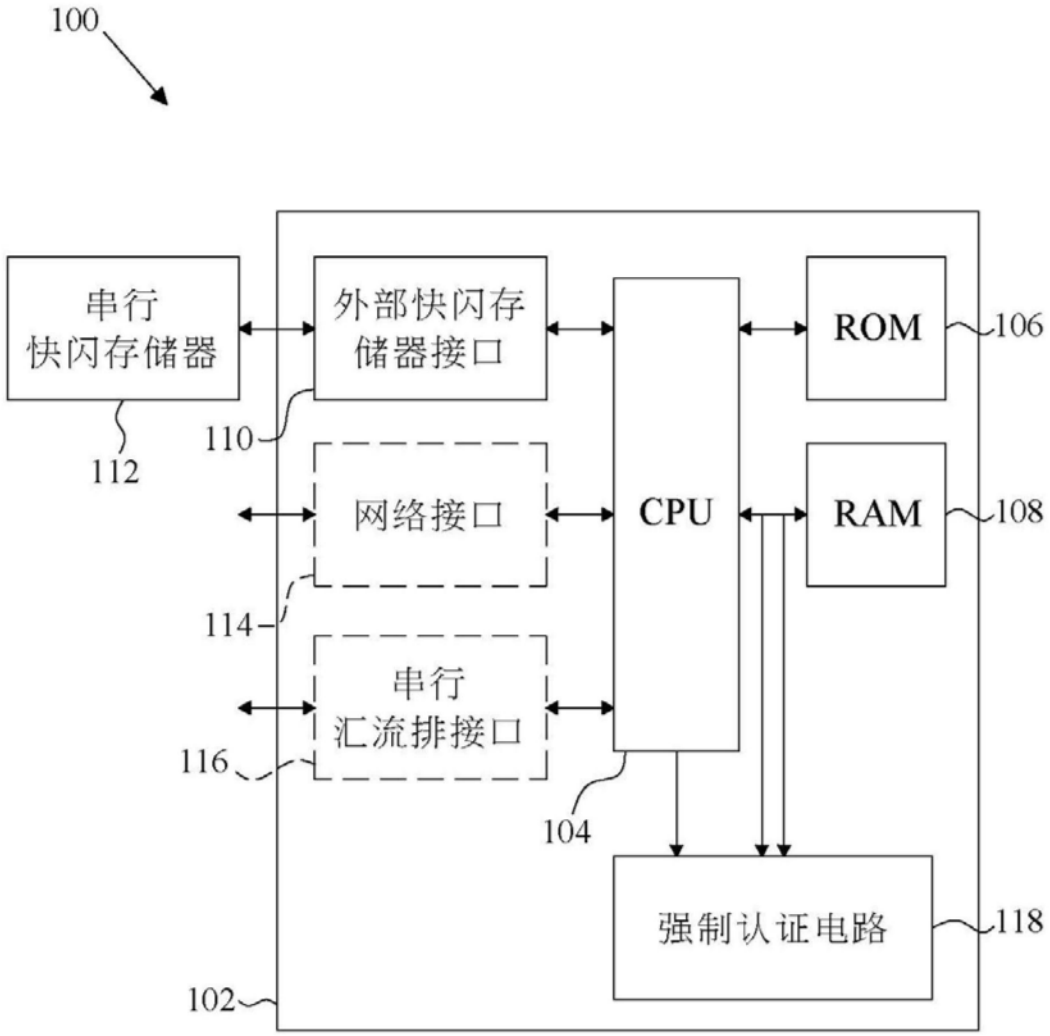

100
串行
快闪存储器
外部快闪存
储器接口
ROM
106
110
112
网络接口
CPU
RAM
108
114
串行
汇流排接口
116
104
强制认证电路
118
102

图1

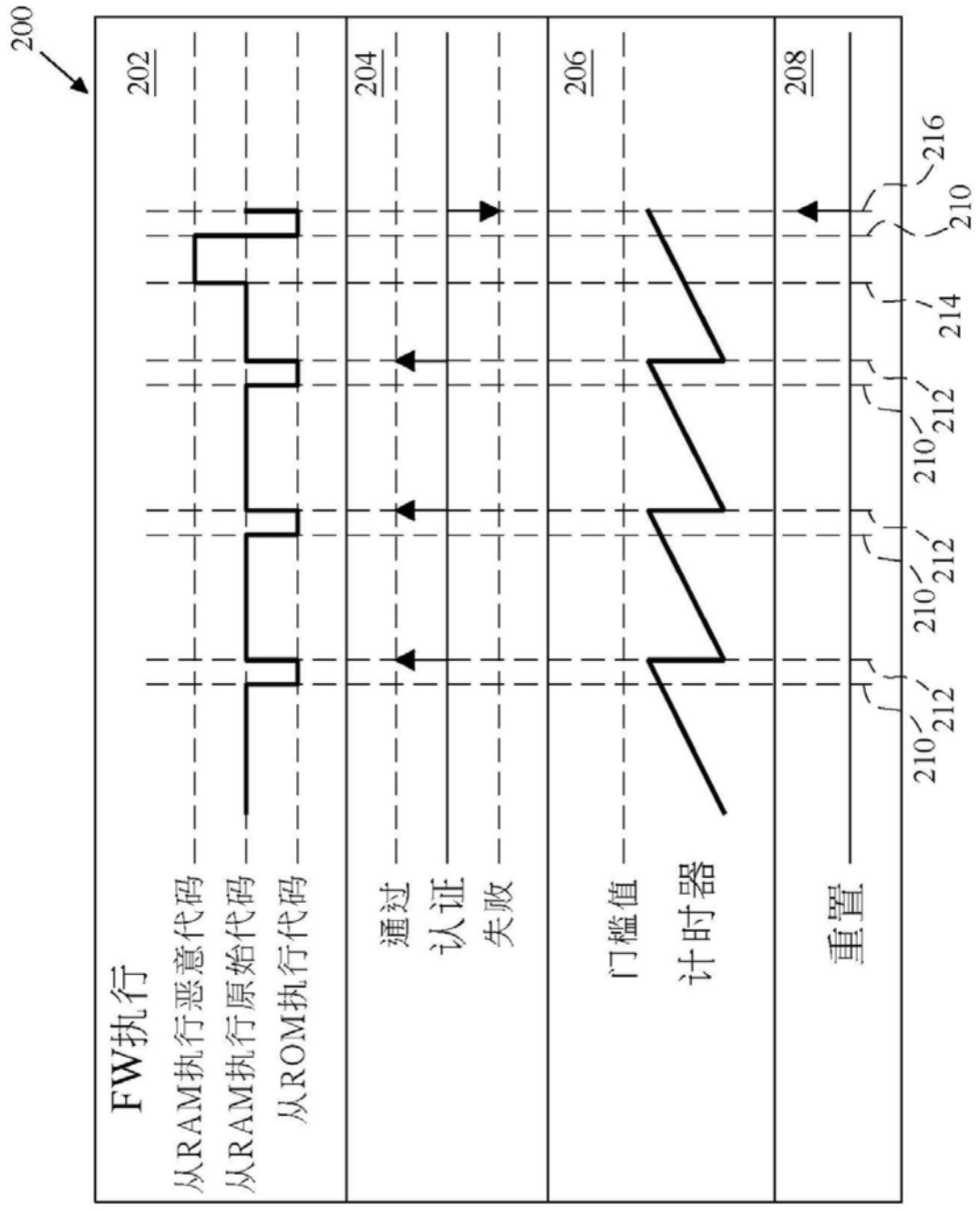

200
202
FW执行
从RAM执行恶意代码
从RAM执行原始代码
从ROM执行代码
204
通过
认证
失败
206
门槛值
计时器
208
重置
216
210
214
212
210
212
210
212
210

图2

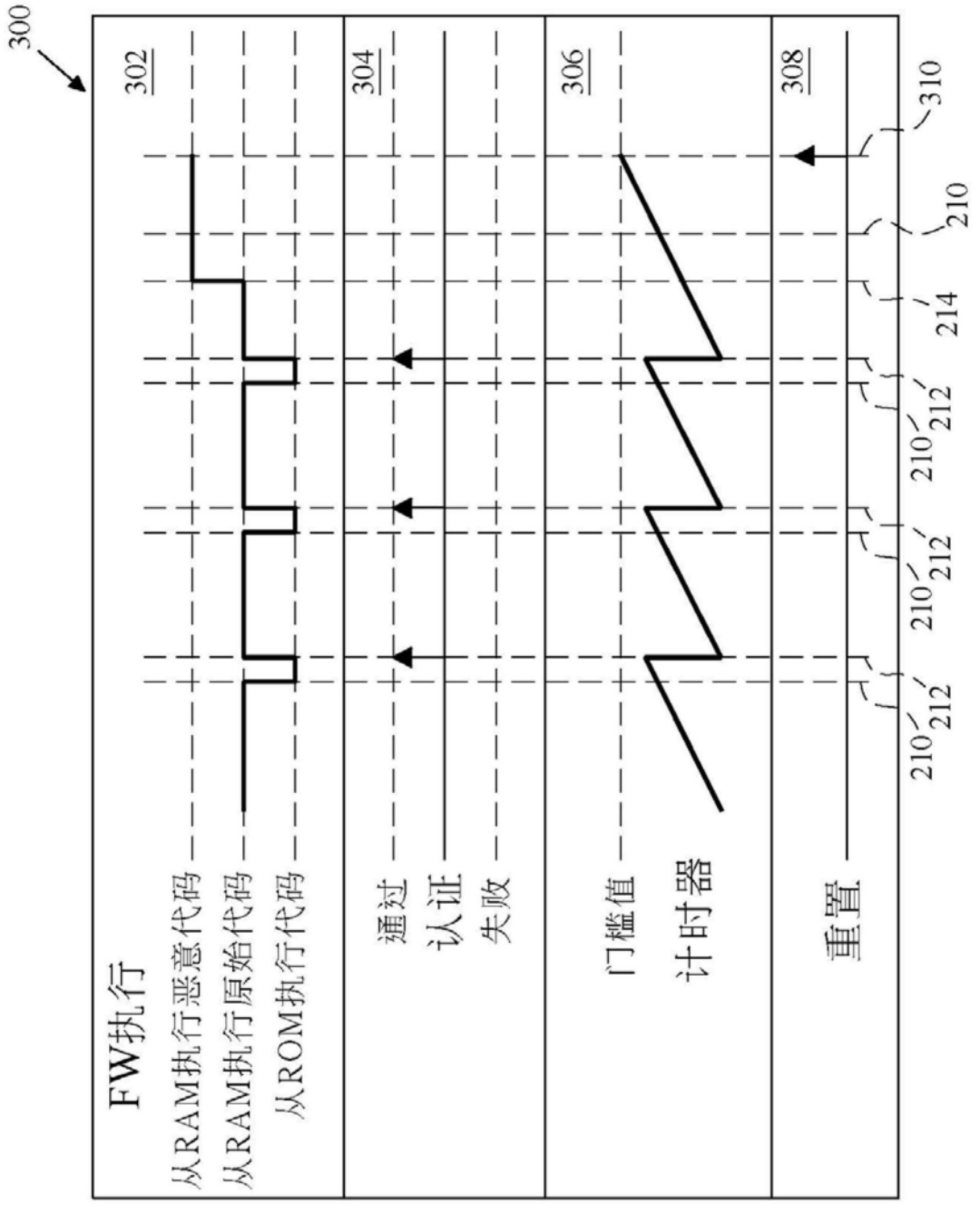

300
302
FW执行
从RAM执行恶意代码
从RAM执行原始代码
从ROM执行代码
304
通过
认证
失败
306
门槛值
计时器
308
重置
310
210
214
212
210
212
210
212
210

图3

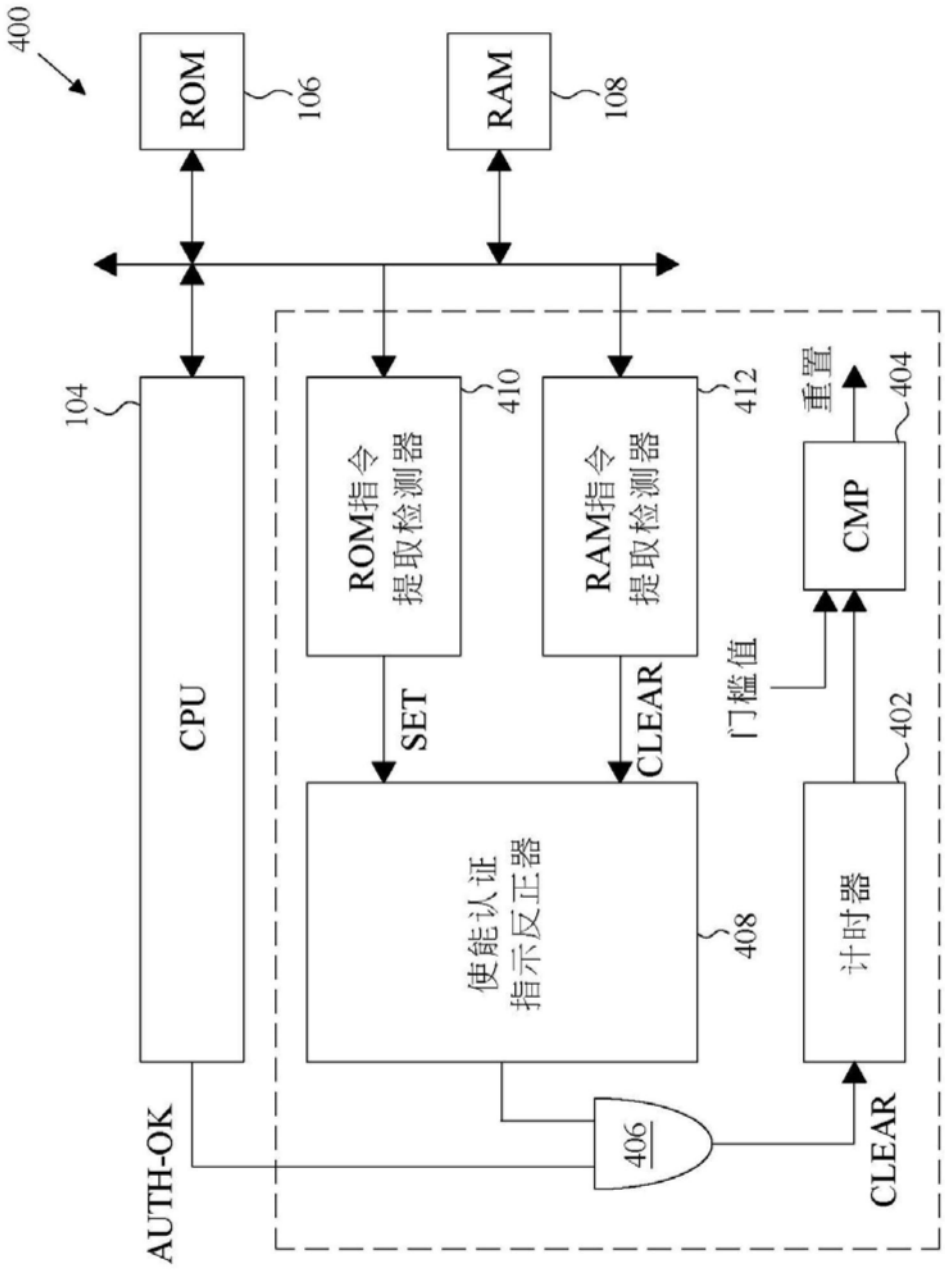

400
ROM
106
RAM
108
104
CPU
ROM指令
提取检测器
410
RAM指令
提取检测器
412
重置
404
CMP
门槛值
SET
CLEAR
402
使能认证
指示反正器
408
计时器
406
AUTH-OK
CLEAR

图4