(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 111159726 A
(43)申请公布日 2020.05.15

(21)申请号 201911258670.2

(22)申请日 2019.12.10

(71)申请人 中国电子科技网络信息安全有限公司
地址 610207 四川省成都市双流县西南航空港经济开发区工业集中区内

(72)发明人 黄沾 幸享宏 杨景贺

(74)专利代理机构 成都九鼎天元知识产权代理有限公司 51214
代理人 夏琴

(51)Int.Cl.
G06F 21/60(2013.01)
G06F 21/78(2013.01)

权利要求书1页 说明书3页 附图1页

(54)发明名称

一种基于UEFI环境变量的全盘加解密方法及系统

(57)摘要

本发明涉及数据加解密技术领域，公开了一种基于UEFI环境变量的全盘加解密方法。将存储分为bootloader、内核和文件系统三个部分；将密钥存储在UEFI的环境变量中，bootloader启动过程中自动从UEFI环境变量中获取密钥并将内核与文件系统进行解密，然后再启动解密后的内核。该方案的关键将密钥存放在设备的硬件之中，达到与设备绑定的目的，保持对现有设备的兼容性，提高了存储区域的安全性；另外该方案不需要对现有应用做其它更改，易用性较强。另外，本发明还公开了一种基于UEFI环境变量的全盘加解密系统。

CN 111159726 A

1.一种基于UEFI环境变量的全盘加解密方法,其特征在于,包括:

将存储分为bootloader(引导加载程序)、内核和文件系统三个部分;

将密钥存储在UEFI的环境变量中,bootloader启动过程中自动从UEFI环境变量中获取密钥并将内核与文件系统进行解密,然后再启动解密后的内核。

2.如权利要求1所述的基于UEFI环境变量的全盘加解密方法,其特征在于,所述bootloader采用grub2。

3.如权利要求2所述的基于UEFI环境变量的全盘加解密方法,其特征在于,所述grub2包括MBR、grub2镜像,所述MBR包含分区表和grub2的stage1加载代码。

4.如权利要求3所述的基于UEFI环境变量的全盘加解密方法,其特征在于,所述MBR位于磁盘第一个扇区,所述grub2镜像位于UEFI的ESP分区。

5.如权利要求1所述的基于UEFI环境变量的全盘加解密方法,其特征在于,所述密钥保留16bytes以上。

6.如权利要求1所述的基于UEFI环境变量的全盘加解密方法,其特征在于,所述UEFI的环境变量中存储的密钥采用pbkdf2算法进行处理。

7.如权利要求2－6任一项所述的基于UEFI环境变量的全盘加解密方法,其特征在于,还包括系统初始化及加密的过程:随机生成密钥并将密钥写入UEFI的环境变量中;调用cryptsetup工具将存储中需要加密的分区进行加密,所述需要加密的分区包括内核和文件系统。

8.如权利要求7所述的基于UEFI环境变量的全盘加解密方法,其特征在于,所述grub2通过UEFI的环境变量访问,获取密钥进行解密。

9.如权利要求8所述的基于UEFI环境变量的全盘加解密方法,其特征在于,当存储所在的设备运行到内核并切换到文件系统后,从UEFI的环境变量中获取密钥对加密的分区再次进行解密。

10.一种基于UEFI环境变量的全盘加解密系统,包括:存储单元和UEFI的环境变量单元;

所述存储单元分为Bootloader单元、内核单元、文件系统单元三部分,所述UEFI的环境变量单元用于存储密钥,所述Bootloader启动过程中自动从UEFI的环境变量中获取密钥并将内核与文件进行解密,然后再启动解密后的内核。

一种基于UEFI环境变量的全盘加解密方法及系统

技术领域

[0001] 本发明涉及数据加解密技术领域,特别是一种基于UEFI环境变量的全盘加解密方法及系统。

背景技术

[0002] 随着计算机和网络的飞速发展,海量的数据存储在各种设备中,其中块设备是最主要的存储设备,携带着大量的涉密文档。倘若系统存在漏洞被未授权使用或者块设备被盗、丢失都会引起政府、企业或个人的重大经济和精神上的损失。如何有效地保护涉密文档的安全性,尤其在计算机丢失或失窃后,防止机密信息非法泄露,这种应用需求对目前普遍存在的存储安全提出了新的挑战。而在通用消费者领域,Android智能手机操作系统在其3.0版本中即提供了存储加密功能,而微软的Windows也推出了BitLocker功能,可以对整个磁盘进行加密。但这些解决方案都需要大量的用户响应(如输入密码)等,无疑这些方案都不适应于在工业环境中运行的设备,这些设备要求7×24h运行,无人工干预。

发明内容

[0003] 本发明所要解决的技术问题是:针对上述存在的问题,提供了一种基于UEFI环境变量的全盘加解密方法及系统。

[0004] 本发明采用的技术方案如下:一种基于UEFI环境变量的全盘加解密方法,包括:

[0005] 将存储分为bootloader(引导加载程序)、内核和文件系统三个部分;

[0006] 将密钥存储在UEFI的环境变量中,bootloader启动过程中自动从UEFI环境变量中获取密钥并将内核与文件系统进行解密,然后再启动解密后的内核。

[0007] 进一步的,所述bootloader采用grub2。

[0008] 进一步的,所述grub2包括MBR、grub2镜像,所述MBR包含分区表和grub2的stage1加载代码。

[0009] 进一步的,所述MBR位于磁盘第一个扇区,所述grub2镜像位于UEFI的ESP分区。

[0010] 进一步的,所述密钥保留16bytes以上。

[0011] 进一步的,所述UEFI的环境变量中存储的密钥采用pbkdf2算法进行处理。

[0012] 进一步的,所述基于UEFI的环境变量中的全盘加密方法还包括系统初始化及加密的过程:随机生成密钥并将密钥写入UEFI的环境变量中;调用cryptsetup工具将存储中需要加密的分区进行加密,所述需要加密的分区包括内核和文件系统。

[0013] 进一步的,所述grub2通过UEFI的环境变量访问,获取密钥进行解密。

[0014] 进一步的,当存储所在的设备运行到内核并切换到文件系统后,从UEFI的环境变量中获取密钥对加密的分区再次进行解密。

[0015] 本发明还公开了一种基于UEFI环境变量的全盘加解密系统,包括:存储单元和UEFI的环境变量单元;

[0016] 所述存储单元分为Bootloader单元、内核单元、文件系统单元三部分,所述UEFI的

环境变量单元用于存储密钥,所述Bootloader启动过程中自动从UEFI的环境变量中获取密钥并将内核与文件进行解密,然后再启动解密后的内核。

[0017] 与现有技术相比,采用上述技术方案的有益效果为:

[0018] (1)本发明的技术方案将密钥存放在设备的硬件之中,达到与设备绑定的目的,保持对现有设备的兼容性,可以兼容所有支持UEFI SecureBoot的设备。

[0019] (2)将密钥存储在UEFI环境变量中,每台设备的密钥都可以随机设置,避免一台主机被破解后所有主机都告破,形成事实上的"后门",提高存储区域的安全性。

[0020] (3)本方案的grub2支持UEFI环境变量访问,每个存储的访问都与该存储所在设备绑定,这样将存储通过物理方法取出直接访问,或者放置到其他任何设备都无法直接访问存储,提高加密内容的安全性。

[0021] (4)本方案可以对现有应用不做任何更改,尽可能的减少对现有应用的干扰,易用性强。

[0022] (5)本方案通过Bootloader自动加载程序,获得密钥进行解密操作,实现非交互式模式;这样有利于设备7×24h无人值守的运行要求。

附图说明

[0023] 图1是本发明实施例中的存储分区示意图。

具体实施方式

[0024] 下面结合附图对本发明做进一步描述。

[0025] 一种基于UEFI环境变量的全盘加解密方法,包括:

[0026] 实施例1:如图1所示,将存储分为bootloader(引导加载程序)、内核和文件系统三个部分(根据需要,磁盘上可能还存在其他区域,其他区域根据需要设置是否加密处理,本实施例的其他区域为加密的分区);

[0027] 基于设计目标,需要将密钥存放于设备的硬件之中,以达到与设备绑定的目的;同时存储的密钥信息必须保留;另外,为了安全性,密钥至少需要保留16Bytes以上,以防止暴力破解;基于以上考虑,将密钥(本实施例的密钥采用的是对称密钥)存储在UEFI的环境变量中,bootloader启动过程中自动从UEFI的环境变量中获取密钥并将内核与文件进行解密,然后再启动解密后的内核。需要说明的是:UEFI的环境变量相当于原来bios的配置,可以保存比如启动顺序这类的信息,UEFI的环境变量是一个UEFI定义的标准,这里是我们自定义一个保存密钥的环境变量。

[0028] 本方案是一个在支持UEFI SecureBoot设备上,使用软件方法来提高设备存储私密性,同时对现有应用做到无缝衔接。

[0029] 实施例2:在实施例1的基础上,所述bootloader采用grub2(是一个来自GNU项目的多操作系统启动程序)。

[0030] grub2包括:MBR(主引导记录)、grub2镜像,所述MBR包含分区表和grub2的stage1加载代码。由于设备本身没有加解密设施,所以grub2是未被加密。而内核与文件系统以及后面的其他分区都已被加密处理。

[0031] 其中,所述MBR位于磁盘第一个扇区(512bytes),所述grub2镜像位于第一个分区,

即UEFI的ESP分区。在开启SecureBoot后，需要将grub2镜像进行签名并存储于ESP分区；这样可以确保只有经过自己签名的grub2镜像能够得到执行，所有未经授权的访问UEFI环境变量的程序都不会得到执行。

[0032] 另一个实施例，实施例1中的关键是将密钥存放在UEFI的环境变量中，在实施例1的基础上，可以对UEFI的环境变量中的信息做多重变换，本实施例对UEFI的环境变量中存储的密钥采用pbkdf2算法进行处理，增加破解难度；也可以采用其他它算法，例如哈希算法等进行多重变换。

[0033] 另一个实施例，在实施例1的基础上，在设备未加密前，需要进行系统初始化的工作。在进入系统后：随机生成密钥并将密钥写入UEFI的环境变量中；调用cryptsetup工具将存储中需要加密的分区进行加密，所述需要加密的分区包括内核和文件系统，图1的实施例中还需要对其他区域进行加密。

[0034] 另一个实施例，在实施例2的基础上，在grub2中，需要应对分区加密与未加密两种情况。最重要的是获取密钥并解密的工作，因为grub2支持UEFI环境变量访问，可以直接获取密钥，进行grub2解密。这一层解密是为了实现存储设备的运行。

[0035] 另一个实施例，存储设备运行之后，当存储所在的设备运行到内核并切换到文件系统后，从UEFI的环境变量中获取密钥将存储空间中加密的分区进行解密，经过这一层解密之后，才能使其它应用程序访问到加密的分区。而这次解密过程也比较简单、方便，获取密钥以及解密都有现有的工具可用。

[0036] 本发明并不局限于前述的具体实施方式。本发明扩展到任何在本说明书中披露的新特征或任何新的组合，以及披露的任一新的方法或过程的步骤或任何新的组合。如果本领域技术人员，在不脱离本发明的精神所做的非实质性改变或改进，都应该属于本发明权利要求保护的范围。

图1