

基于虚拟文件系统的安全存储技术的研究

崔 奇

(西北工业大学 计算机学院, 陕西 西安 710072)

摘 要: 介绍了一种基于虚拟文件系统的加密存储方法, 通过文件系统开发技术设计并实现一个虚拟文件系统, 该文件系统能够将一个二进制文件映射成虚拟磁盘, 而且可以在虚拟磁盘上存取数据. 为了保证虚拟磁盘中的文件的安全性, 还在虚拟文件系统上集成了加密引擎、密钥管理和硬件绑定等模块. 这种安全存储技术操作简单, 并且能够提供高效实时的加密/解密措施, 具有一定的实用价值.

关键词: 安全存储; 虚拟文件系统; 虚拟磁盘; 硬件绑定

中图分类号: TP309

文献标识码: A

文章编号: 1000-7180(2013)06-0052-05

Secure Storage Technology Based on Virtual File System

CUI Qi

(School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China)

Abstract: This paper describes an encrypted storage method based on the virtual file system. We design and implement a virtual file system through the file system development technology. Virtual file system regards a binary file as virtual disk, and can access the data on the virtual disk. In order to ensure the security of the file in virtual disk, we integrate encryption engine, key management and hardware binding modules into virtual file system. This security storage technology is easy to use, and can provide efficient real-time encryption / decryption measures, so it has practical value.

Key words: encrypting storage; virtual file system; virtual disk; hardware binding

1 引言

随着电子信息技术的发展, 电子文档的应用越来越广泛, 电子文档的安全性也日益成为人们关注的问题, 其安全存储成为保障信息安全的首要要求.

针对现有的加密方式存在的缺陷, 本文提出了一种基于虚拟文件系统的安全存储技术.

2 系统框架

SVFS 系统主要由 5 部分构成, 如图 1 所示.

虚拟文件系统: 该部分用来对虚拟磁盘文件空间进行组织和分配, 具体地说它负责为用户创建并格式化虚拟磁盘文件, 建立文件, 存入、读出、修改以及删除文件^[1].

加密引擎: 该部分是被虚拟文件系统的设备层调用, 实现对数据的加密和解密, 并且支持多种加密算法.

硬件绑定: 该部分主要功能是将虚拟磁盘绑定到用户指定的硬件上, 以及验证虚拟磁盘的使用环境是否是其所绑定的硬件环境.

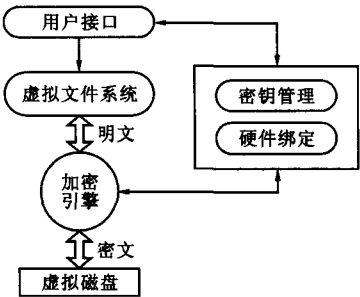


图 1 SVFS 的系统框架

收稿日期: 2012-08-29; 修回日期: 2012-10-27

基金项目: 西北工业大学研究生创业种子基金(z2012140)

密钥管理:在 SVFS 中有三种密钥:用户密钥,硬件密钥和磁盘密钥,用户密钥是掌握在用户手中的密钥。硬件密钥是 SVFS 收集计算机上的硬件信息生成的密钥,主要用来检验虚拟磁盘所处的硬件环境是否正确。使用硬件密钥对用户密钥进行加密,然后再对用户密钥的密文进行散列变换就生成磁盘密钥,一个磁盘密钥对应一个虚拟磁盘,可以用它对虚拟磁盘进行加密和解密。为了保障安全,这三种密钥都不在虚拟磁盘上存储,它们只在内存中存在^[2]。

用户接口:该部分主要向用户提供访问虚拟磁盘的接口,包括创建虚拟磁盘、格式化虚拟磁盘,加载虚拟磁盘,以及一些文件操作接口。

3 虚拟文件系统

3.1 虚拟文件系统概述

SVFS 系统中采用的虚拟文件系统格式类似于 Ext2 文件系统,在 Ext2 文件系统^[3]格式上做了一些调整。虚拟文件系统磁盘读写虚拟磁盘的基本单位是块(block),每块大小为 1024 个字节。虚拟磁盘分区的格式如图 2 所示。

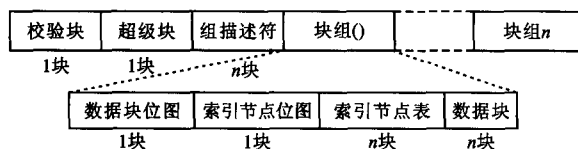


图2 虚拟磁盘的格式

从图2中可以看出,虚拟磁盘的第一块是校验块,第二块是超级块(super block),跟在后面是组描述符(group descriptors),虚拟磁盘剩下的空间被分为多个大小相同的块组(block group)进行管理,每个组块又包含了:块位图(block bitmap),索引节点位图(inode bitmap),索引节点表(inode table),和数据块(data blocks),这些部分根据需要占用1个或者多个块。校验块是为了在加载虚拟磁盘时验证虚拟磁盘所处硬件环境是否正确,从超级块开始才是真正的虚拟文件系统的格式。

虚拟的文件系统和实际的文件系统会有不同之处,这就要求在设计虚拟文件系统时对于 Ext2 中的一些参数进行取舍和调整,这点主要体现在对超级块和组描述符的设计上。

超级块记录的了虚拟文件系统的大小,以及虚拟文件系统当前的状态信息,其中包括:总块数、空闲块计数器、每组中的块数、总的索引节点数、空闲索引节点计数器以及每组中索引节点个数等等。与真正的 Ext2 文件系统相比,SVFS 中的虚拟文件系

统的超级块要简单一些,其中省去一些次要信息,比如:时间信息和一些兼容性的参数。

每个组块都有一个组描述符,记录了该组块块位图的位置、索引节点位图的位置、索引节点表的位置、以及空闲块和空闲索引节点的个数。

在 Ext2 文件系统中超级块和组描述符除了在第一个组块中存在,还在其他每个组块中都有副本,但是为了节约虚拟磁盘空间,在虚拟文件系统中就把这两部分从组块中提取出来,单独放在虚拟磁盘的最前面。

每个组块都有一个块位图,占一个数据块,用来描述这个组块中数据块的使用情况。块位图中的每一位(bit)表示这个组块中相应的数据块的使用情况,“1”表示该数据块已被占用,“0”表示空闲。

索引节点位图描述了组块中索引节点的使用情况,使用方法和块位图类似。索引节点位图也只占一个数据块。

索引节点表由组块中的文件的索引节点组成,占用多个连续的数据块。

在虚拟文件系统中每个文件(包括目录)都用一个索引节点来描述,这个数据结构记录了一个文件除了文件名和索引节点号之外的所有信息。在索引节点中存储了对应文件的属性和文件所占用的数据块号。

虚拟文件系统目录项记录了对应文件的文件名和索引节点号,其主要作用是提供文件名到索引节点的转换。

3.2 虚拟文件系统的结构

虚拟文件系统的设计采用了层次结构,从上至下依次是:API层,内核层,逻辑设备层和虚拟磁盘。如图3所示。

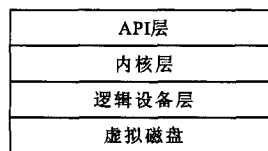


图3 虚拟文件系统的层次结构

API层为用户界面使用虚拟文件系统的功能提供接口,内核层提供了整个虚拟文件系统的服务,供上层调用。逻辑设备层负责处理上层的 I/O 请求。

3.3 关键问题

3.3.1 性能与缓存

访问索引节点是虚拟文件系统定位一个文件的基本途径,对块位图和索引节点位图的访问是虚拟文件系统新建文件以及删除文件必须的操作,这些

操作的执行频率都非常高,所以它们的执行效率对整个虚拟文件系统的性能有很大影响.为了解决这个问题,虚拟文件系统使用了高速缓冲区.当虚拟磁盘被加载时,文件系统会创建一个大小为 128 个数据块的缓冲区.当需要访问某一块数据时,就先在缓冲区中查找对应的块,如果查到该块,就直接执行操作,若是写操作还需要将该块缓冲区标记为“脏”.如果这个块不在缓冲区中就从虚拟磁盘中读取它,并且还要将该块加载到缓冲区中.如果缓冲区已满,就采用最近最少使用(LRU)策略从高速缓存中删除一个块,并把需要的数据块插入到缓冲区的第一个位置.在卸载虚拟磁盘的时候,虚拟文件系统会清空缓冲区,同时将被置为“脏”的缓冲区块写回到虚拟磁盘上,这样就保证了数据的一致性.缓冲区的工作流程如图 4 所示.

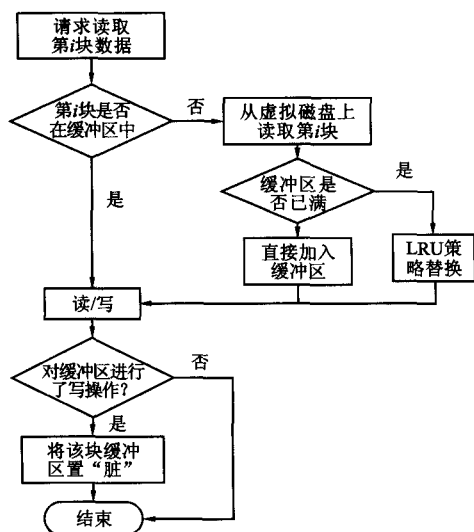


图 4 缓冲区的工作流程

3.3.2 虚拟磁盘与宿主机的数据交换

我们将虚拟文件系统运行时所在的主机称为宿主机.由于虚拟文件系统并没有集成到宿主机操作系统中,因此,虚拟磁盘中的数据对操作系统是不可见的.这样避免的虚拟磁盘中的数据因操作系统中的不安全因素而被窃取或破坏,但是宿主机上的合法应用程序也无法直接访问到虚拟磁盘中的数据.为了解决这个问题,我们采用了临时文件的方法.

当用户试图打开某个文件时,虚拟文件系统就将该文件复制到宿主机的临时目录中,然后在调用宿主机上相应的应用程序将该临时文件打开.同时虚拟文件系统还要监控被打开的临时文件,一旦发现临时文件被修改,虚拟文件系统就将被修改过的临时文件同步到虚拟磁盘中.当用户关闭临时文件时,虚拟文件系统自动将临时文件彻底清除掉,而且

确保将该临时文件被清除掉,以免留下安全隐患.临时文件和虚拟磁盘中对应的文件保持数据同步以及清除临时文件等操作均在后台进行,对用户是透明的,不会影响用户对临时文件的使用.

使用这种方法只有正被用户使用的文件才会在宿主机上产生临时文件,而且临时文件在关闭之后会被立即清除,这样暴露给宿主机环境的文件的数量是非常少的,而且时间也比较短.而文献[1]中的虚拟磁盘一旦被加载,就像宿主机的一个硬盘分区一样,其中的所有文件都会被暴露给宿主机环境,而且在整个使用虚拟磁盘的过程中,都有可能出现数据泄露和数据损坏的风险^[4].

4 安全机制

4.1 加密与解密

在虚拟文件系统中加密的对象有很多,选用不同的对象进行加密,实现的难易程度不同,所获得的性能和安全性也是不同的.第一种方法是对整个虚拟磁盘文件整体进行加密,这种方法实现简单,缺点是每次加载虚拟磁盘都先要将虚拟磁盘文件进行解密,不仅效率低下,而且会产生临时的虚拟磁盘文件.第二种方法是以块为单位对虚拟磁盘进行加密,虚拟磁盘上的每个数据块都被加密存放,在设备层对虚拟磁盘的进行读操作时只对需要的块进行解密,而不用解密整个虚拟磁盘,加密和解密伴随着每次对虚拟磁盘的读写,这种方式不仅透明而且效率高.目前使用这种加密方式的系统有很多.第三种方式是文件系统加密,这种方法和第二种方法一样提供了透明的加密功能,但是又比第二种方法提供的加密粒度更细.文件系统加密可以支持文件粒度的加密,用户可以选择只对某些文件或目录进行加密,由于此方法加密的数据块比第二种方法少,它的性能会更好一些.由于 SVFS 的设计是针对虚拟磁盘上所有的文件进行保护,因此它采用第二种加密方法.

加密引擎是安全机制的核心^[5],它包括两部分:算法管理模块和算法模块.算法管理模块主要负责调度现有的算法以及添加新的算法.算法模块提供了两类算法:散列算法和加密/解密算法.散列算法负责对原始密钥进行处理,生成与加密/解密算法相应的密钥长度.加密/解密算法负责对虚拟磁盘上的数据进行加密和解密,为了提高加密引擎的可扩展性,所有的加密算法都采用相同的接口,只要按照指定的接口编写的加密算法模块就会很容易被集成到

加密引擎中。

虽然加密引擎提供了很多算法模块,但是一个虚拟磁盘只采用一种散列算法和一种加密算法,加密算法的选择是随机的,散列算法会根据所选加密算法所要求的密钥长度随机选择,而且在虚拟磁盘不添加任何关于加密算法标识和版本的信息,加载虚拟磁盘时,虚拟文件系统将会依次将依次尝试在散列算法中用不同的算法计算出磁盘密钥,然后用该磁盘密钥依次使用解密算法对虚拟磁盘的超级块进行解密直到超级块能够被虚拟文件系统识别为止,如果所有的加密算法都尝试过之后,超级块仍然无法被识别,则认为用户提供的密钥是错误的。

在虚拟磁盘中经常会重复出现相同的数据片段,比如:一些没有被占用的数据块全部被“0”填充,一些文件中也会出现重复的数据,而且这种情况非常频繁。使用加密算法对这些重复出现的明文片段进行加密时,加密后产生的密文也往往是相同的,这些大量密文片段重复出现的情况会给虚拟磁盘的安全性带来很大的隐患。为了解决这个问题,加密引擎在加密时除了用密钥对数据块进行加密之外,还会将每个字节的密文信息与其位置信息(块号,块内偏移量)进行一系列复杂的运算,从而消除了大量密文重复出现的情况。

4.2 硬件绑定与密钥管理

要实现安全存储,仅仅只依靠安全的加密算法是不够的,一旦用户密钥泄露,SVFS的安全机制就失去作用,所以对虚拟磁盘的加密不仅要依靠密钥,还需要将其绑定到实体的硬件上,才能进一步提高系统的安全性。计算机上的很多硬件资源以及一些移动存储设备都拥有序列号或者唯一标识符,将这些众多与硬件相关的信息结合在一起,可以产生一个能够唯一标识这台计算机的标识符。在SVFS中使用的硬件信息包括:硬盘序列号,主板序列号,CPU序列号,网卡MAC地址,如果用户指定了使用USB移动存储设备,还可以选择U盘或者移动硬盘。只选用一到两种上述硬件信息也可以唯一的确定一个硬件环境,但为了保证硬件绑定模块的可靠性,硬件绑定模块就综合了尽可能全面的硬件信息。

为了说明硬件绑定模块的工作原理,先介绍一下校验块的格式。校验块只使用了前512个字节,后512个字节作为系统保留区。校验块的前16个字节是以明文储存的,第16~511字节是以密文存储的。第0~8字节是软件版本号,第9~11字节分别表示

该虚拟磁盘是否使用了计算机上的硬件、U盘和移动硬盘进行硬件绑定,“0”表示“否”,“1”表示“是”。第12~15字节是第16~511字节明文的CRC-32校验码。

在SVFS初始化一块虚拟磁盘时,如果用户设置了硬件绑定选项,系统会读取计算机上的硬盘序列号,主板序列号,CPU序列号,网卡MAC地址等信息,如果用户还指定了要绑定的移动存储设备,还要获取这些设备的信息,对于无法获取信息的硬件,使用默认值代替。硬件信息获取完毕之后,密钥管理模块就会使用散列算法处理这些信息生成硬件密钥。接着,密钥管理模块会生成496个字节的随机数,并且计算这些随机数的CRC-32校验码,填充在校验块的第12~15字节,然后系统使用硬件密钥对这些随机数进行加密,密文填充在校验块的第16~511字节。密钥管理模块进一步使用硬件密钥对用户密钥进行加密生成最终对虚拟磁盘进行加密的磁盘密钥。

当SVFS加载虚拟磁盘时,会根据校验块中所记录的其所绑定的硬件类别去读取当前计算机上相应的硬件信息,进而生成硬件密钥。SVFS再用硬件密钥对校验块中的第16~511字节进行解密,然后对生成的明文进行CRC-32校验。如果校验通过就进行生成磁盘密钥,对超级块进行解密,加载虚拟磁盘等这些操作,否则就提示用户硬件环境不正确,退出系统。

4.3 认证流程

硬件绑定和密钥管理三个模块,这几部分是虚拟磁盘加载过程中安全认证的核心。虚拟磁盘在加载过程中首先进行硬件环境验证,然后进行用户密钥的验证,只有这两项验证都通过了,虚拟磁盘才会被加载。综合前两节的内容,SVFS认证虚拟磁盘的流程如图5所示。

5 结束语

本文提出了一种安全虚拟文件系统,SVFS,该系统的开发利用了文件系统开发技术,它的主要构成部分是一个虚拟文件系统,它能够将一个二进制文件映射成为一个虚拟磁盘,并且提供了丰富的接口,使用户可以方便的使用虚拟磁盘。SVFS中的加密引擎,硬件绑定模块以及密钥管理模块共同为虚拟文件系统提供了高效可靠,透明的实时加密/解密措施。使用SVFS对文件进行加密比传统的文件加密方法的操作更简洁^[6],安全性更高,这种方案适合

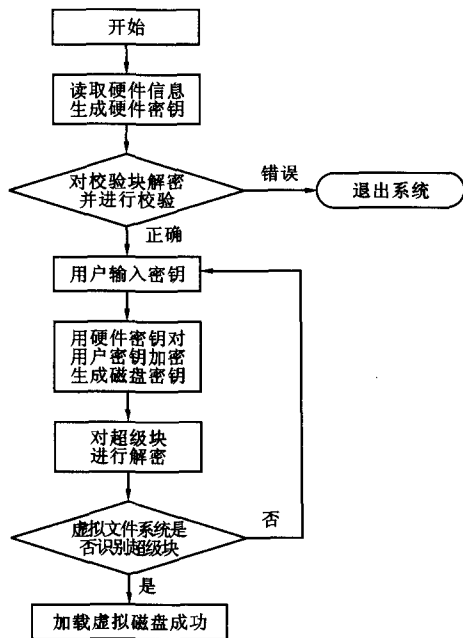


图5 虚拟磁盘认证的流程

于对移动存储设备中机密文件进行保护,因此 SVFS 具有更高的实用性。

参考文献:

- [1] 李清俊,甘萌. 基于虚拟磁盘的文件加密方法[J]. 计算机工程与设计,2006,27(15):2835—2838.
- [2] 朱征宇,周发贵,于春雷. 一种基于计算机指纹的 U 盘文档监控与保护方法[J]. 重庆理工大学学报:自然科学,2011,25(3):62—68.
- [3] 包怀忠. EXT2 文件系统的分析与研究[J]. 计算机工程与设计,2005,26(4):1022—1024.
- [4] 陈莉君,张琼声,张宏伟,等译. 深入理解 Linux 内核[M]. 3 版. 北京:中国电力出版社,2007.
- [5] 魏丕会,卿斯汉,刘海峰. 基于安全操作系统的透明加密文件系统的设计[J]. 计算机科学,2003,30(7):132—135.
- [6] 王俊华,译. 操作系统设计与实现[M]. 北京:清华大学出版社,2008.

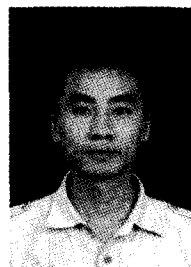
作者简介:

崔奇 (1987—),硕士. 研究方向为计算机应用技术.

(上接第 51 页)

- [6] Li W, Yang Y, Yuan D. A novel cost-effective dynamic data replication strategy for reliability in cloud datacentres[C] // International Conference on Cloud and Green Computing. Sydney,2011:496—502.
- [7] Calheiros R N, Ranjan R. CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms[J]. Software: Practice and Experience, 2011, 41(1):23—50.

作者简介:



吴永发 (1987—),硕士研究生. 研究方向为分布式网络存储、云存储、嵌入式软件技术.

曾碧女,(1963—),教授. 研究方向为移动计算与物联网技术、嵌入式系统.