一个基于 Linux 的加密文件系统的设计与实现

邢常亮　卿斯汉　李丽萍
（中国科学院信息安全技术工程研究中心，北京 100080）
E-mail：clxing@126.com

摘　要　加密文件系统是保护用户敏感数据的一种有效手段。传统的加密文件系统（CFS、TCFS、AFS 等）因为数据保护不彻底、系统负载过大等缺陷，无法很好地满足用户需求。该文介绍了一种加密文件系统 SEFS 的设计与实现方案，该方案基于 Linux 操作系统，利用其绕回设备技术实现对用户数据和系统交换分区的加密保护，另外，该方案增强了自主访问控制策略，以满足高等级安全标准的需求。

关键词　信息安全　加密文件系统　Linux

文章编号 1002-8331-(2005)17-0101-04　**文献标识码** A　**中图分类号** TP309

The Design and Implementation of an Encrypted Filesystem for Linux

Xing Changliang　Qing Sihan　Li Liping
(Engineering Research Center for Information Security Technology,
Chinese Academy of Sciences, Beijing 100080)

Abstract: Using encrypted filesystem is an effective method to protect sensitive data.The traditional encrypted filesystems(such as CFS、TCFS、AFS) can't meet user requirements perfectly because of protecting data incompletely and high overload.This paper discribes the design and implementation of an encrypted filesystem named SEFS, which encrypts user data and system swap space using the loop device technology based on Linux OS.In addition, SEFS enforces DAC policy to meet the high grade security standard.

Keywords: information security, encrypted filesystem, Linux

1　引言

加密文件系统作为一种有效的数据加密存储技术而受到人们的青睐，它可以有效防止非法入侵者窃取用户的机密数据；另外，在多个用户共享一个系统的情况下，可以很好地保护用户的私有数据。

安全操作系统是计算机系统安全的基础，目前，高安全等级的操作系统（如 TCSEC 中定义的 B2、B3 级）对系统安全性的增强主要都是从访问控制和审计等方面入手，通过限制用户权限、监视用户行为来保证系统的安全性，然而这些措施在某些情况（如：通过另一个操作系统以管理员权限读取磁盘上的数据）下无法对用户数据实施有效的保护，而加密文件系统的使用可以很好地杜绝这种情况的发生。所以，在高安全等级的操作系统中引入加密文件系统将对系统整体安全性的增强起到至关重要的作用。

目前，已经有很多成熟的加密文件系统被广泛地应用，如基于 linux 系统的 CFS、TCFS、AFS，基于 window 系统的 EFS 等。这些加密文件系统对用户数据都实施了不同程度的保护，然而，它们普遍存在一些问题，如：对敏感数据的保护不够完善，性能比较低下等。所以，设计出一种性能高效、同时能满足高等级安全标准要求的加密文件系统是件非常有意义的事情。

文章在第 2 节对几个传统的加密文件系统进行了分析，指出了它们存在的一些不足，随后在第 3 节中提出了一种新的基于 linux 系统的加密文件系统（Secure Ecryption Filesystem）设计方案，对这些问题做了很好的解决，第 4 节对 SEFS 系统的一些关键技术做了进一步说明，第 5 节对 SEFS 性能做了分析，第 6 节总结全文。

2　对现有系统的分析及存在的问题

2.1　CFS（Cryptograhpic File System）

CFS（Cryptograhpic File System）是一个经典的加密文件系统，利用 NFS 文件系统的工作原理，实现为一个 NFS 服务器[1]。在把数据向磁盘写入时对数据进行加密，在从磁盘读出数据时对数据进行解密，从而保证了数据在磁盘上的密文存储。

CFS 最大的缺点在于其效率低下，这是因为它的加密操作在用户层完成，而且要频繁地进行用户层到核心层的数据交换所致[2]。另外，CFS 的加密是基于目录的，虽然它会把加密目录下的所有文件内容加密，但文件名、文件大小、访问时间、目录结构等信息都是以明文形式存储，这些极大地影响了系统整体

基金项目：中国科学院知识创新工程方向性项目（编号：KGCX2-SW-104）
作者简介：邢常亮（1979-），硕士研究生，研究方向为安全操作系统。卿斯汉（1939-），研究员，博士生导师，主要研究领域为信息系统安全理论和技术。
李丽萍（1976-），博士研究生，研究方向为安全操作系统。

的安全性。

2.2 TCFS(Transparent Cryptograhpic File System)[3]

TCFS(Transparent Cryptograhpic File System)也是利用了NFS文件系统的工作原理,保留了CFS的优点,与CFS的不同在于它实现为一个NFS客户端。TCFS对数据进行加密时,对每个文件使用不同的"文件密钥"进行加密,对一个文件的不同部分使用的不同的"块密钥"进行加密,这就保证了用户无法通过比较两个文件来判断它们的明文是否相同,也无法判断同一文件的不同部分的明文是否相同。用户的"主密钥"由用户的登陆密码加密后存放在文件中。与CFS不同,TCFS的数据加密、解密操作在核心层完成,所以性能有所改善。

TCFS的问题在于系统的安全性过多地依赖于用户的登陆密码,而且加密密钥存放在磁盘上的方式也在一定程度上降低了系统的安全性,因为其基于NFS的工作机制,每次读写操作都会涉及到多次核心层与用户层之间的数据交换,所以其效率的低下还是难以避免。另外,TCFS对文件名、文件大小、访问时间、目录结构等一些敏感信息也没有做很好的保护。

2.3 AFS(Andrew FileSysterm)[4]

AFS是一个分布式加密文件系统,它通过一个统一的访问接口把多个服务器连接起来,形成一个庞大的数据存储空间。客户端在访问服务器上的数据时,只要把共享目录挂载到本地目录上就可以了,无需去关心数据到底存放在哪个服务器上。AFS实施了严格的访问控制,每个目录的访问控制列表可以有20个条目,这样,目录结构、目录下文件的文件名、文件大小等敏感信息得到了很好的保护;AFS提供了一个安全的数据传输路径,客户端与服务器端进行数据交换时用会话密钥进行加密。

AFS存在一个严重的缺陷:数据在服务器端是明文存储。这就要求数据服务器必须绝对安全、可信,而这一点很难做到。另外,分布式的数据存储方式使得只要有一台服务器被非法侵入,则整个系统的安全性都将被破坏。

2.4 存在的问题

通过上面的分析,可以发现传统的加密文件系统都存在着不同程度的问题,主要集中在两方面:

(1)数据保护不完全。CFS和TCFS虽然实现了数据的密文存储,但对目录结构、文件名、文件大小等敏感信息没有做妥善的保护;而AFS对数据采取明文存储的方式存在很大的安全隐患;另外,无论是CFS、TCFS还是AFS,他们对用户数据的保护主要都是从密文存储和访问控制两方面入手的,而对系统内存和系统交换分区均未采取任何保护措施,这是一个很大的疏漏。加密文件系统的使用过程中,用户数据在内存和交换分区中都是以明文的形式存在,不妥善处理就很容易造成用户数据的泄漏,内存中的用户数据存在的时间相对较短,而且直接读取内存中的用户数据相对来说也比较困难,所以危害较小,可以暂时不予考虑。而交换分区中的用户数据可能会因为计算机的突然关闭而残留在磁盘上,通过直接读取磁盘的方法将很容易得到,所以系统交换分区是一个必须保护的对象。

(2)性能低下。CFS/TCFS系统由于采用了NFS文件系统的工作原理,当对本地数据进行加密保护时,系统性能非常低下,与ext2文件系统相比,读、写时间增加了约20~30倍[2]。在大数据量访问的情况下,这种大幅度的性能下降是用户无法忍受的。

3 SEFS的提出

3.1 设计目标

通过上面对传统加密文件系统的分析,可以发现这些系统确实都还存在一些需要改进的地方,于是作者提出一种新的加密文件系统的设计与实现方案,即SEFS(Secure Ecryption FileSystem)。在SEFS的设计过程中,着重强调系统的安全性,力图给用户数据实施最全面的保护,以满足高安全级别的需求,在保证安全的前提下,兼顾系统性能,尽可能地降低系统负载。

3.2 工作原理

Linux中有一种loop设备(绕回设备)技术,可以将一个连续的块文件映射为一个虚拟磁盘,然后就可以像使用普通物理磁盘一样在该虚拟磁盘上创建文件系统、存取数据。SEFS就是利用了loop设备的工作原理,实现了一个虚拟磁盘设备,将要保护的用户数据放入该虚拟磁盘中,并在虚拟磁盘的设备驱动中增加了加密引擎,以实现对数据的加密保护。

虚拟磁盘以一个块文件为存储介质,当它不工作时,对外部看来就只是一个文件而已。这种数据加密方式不仅实现了数据的加密存储,而且对目录结构、文件大小在内的一些敏感信息也起到了很好的保护作用。

3.3 系统框架

SEFS系统主要有这几部分构成:

设备驱动:像前面提到的一样,SEFS是通过将一个块文件映射为虚拟磁盘,并对该虚拟磁盘进行加密从而实现对用户数据进行保护的。而对虚拟磁盘的访问是通过一个设备驱动来完成的,该设备驱动向上层提供与磁盘相关的各种系统调用,并对存入虚拟磁盘的数据进行加密操作。为增加系统灵活性,该设备驱动实现为一个可动态加载模块,可以在需要的时候,载入到系统内核中。

加密引擎:该部分主要是被设备驱动模块调用,实现对数据的加密、脱密。SEFS目前支持三种加密算法:twofish、gost、blowfish,密钥长度均为256位。考虑到系统的可扩展性,将每种加密算法实现为一个具有同一接口的可动态加载模块,使得扩充新的加密算法变得非常方便。

密钥管理:SEFS的密钥管理部分涉及到三种密钥:文件密钥、主密钥、用户口令。文件密钥用来对虚拟磁盘进行加密,每个文件密钥对应一个虚拟磁盘,由主密钥加密后存储。为了系统的安全性,用户的主密钥不在磁盘上存储,而是根据用户口令由一个单向函数变换生成。

用户接口:该部分用来向用户提供使用SEFS的访问接

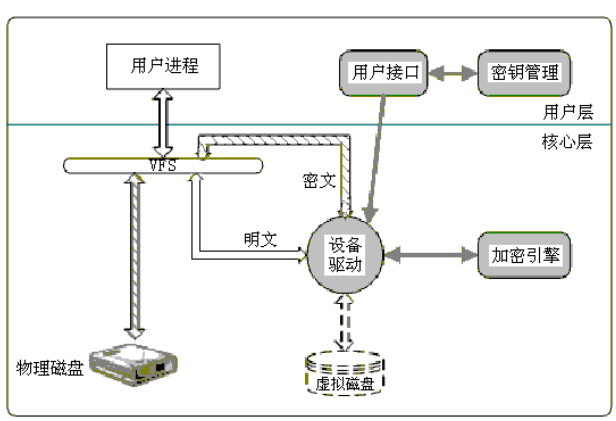

图1 SEFS体系结构图

口，提供了对虚拟设备进行创建、格式化、挂载等操作的一系列命令。

SEFS 各部分之间结构如图 1 所示。

4 关键技术的实现

4.1 虚拟磁盘的加密

在 SEFS 中，虚拟磁盘的存储介质是一个块文件，如果简单起见，直接对该块文件进行加密便可以实现虚拟磁盘的整体加密，但这样做有很多弊端：首先，每次对虚拟磁盘进行访问，都必须把整个块文件进行解密，影响系统性能；另外，解密后的块文件如存放不当，容易泄漏其中存储的数据。

操作系统对磁盘的访问是一种随机访问，一般以磁盘扇区为单位进行，每个扇区的大小一般为 512 Bytes。鉴于这个原因，SEFS 对虚拟磁盘的加密采取以扇区为单位的方式，每个扇区被加密后作为块文件数据区内的一个数据块存放在特定位置，这样，当读某个文件时，只需将该文件所占用的若干磁盘扇区解密即可，不用解密整个磁盘，大大提高了系统性能和安全性。为了进一步增强系统的安全性，采用了 CBC(Cipher Block Chaining)加密模式，以达到隐藏扇区的数据结构的目的。

4.2 加密引擎的模块化

SEFS 目前提供三种加密算法，基本上可以满足用户的一般需求，但考虑到系统以后的扩展性，作者把加密引擎设计成一种可动态扩展的实现模式。加密引擎由算法管理模块、加密算法模块两部分构成：算法管理模块用来对已注册的加密算法进行调度、管理；加密算法模块用来实现对数据的加密操作，每个算法被封装成一个具有统一接口的动态加载模块，在需要时由算法管理模块动态载入到内核中。

加密引擎的这种实现方式对于添加新的加密算法非常方便，首先，将新加密算法封装成一个动态加载模块，按照规定的格式向外部提供调用接口；然后，向算法管理模块注册该加密算法，并提供相应的注册信息(算法标识符、密钥长度、动态加载模块名称等)。这样，用户便可以使用新加密算法进行加密了。

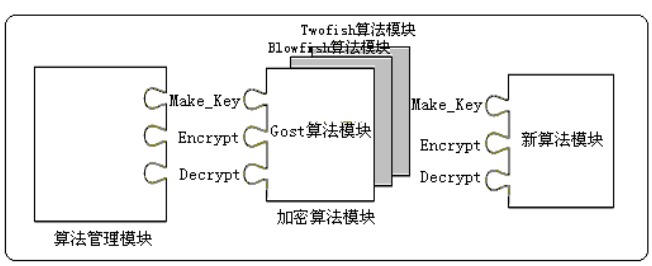

图 2 加密引擎结构

4.3 交换分区的加密

SEFS 这种利用虚拟磁盘实现对用户数据进行加密存储的工作模式，使得对交换分区的加密变得非常容易。交换分区是操作系统中用来暂时存放内存中数据的一块磁盘空间，通常是一个磁盘分区。既然交换分区是一个磁盘分区，而 SEFS 又可以实现对磁盘(虚拟磁盘)的整体加密，那么将交换分区存放到一个被加密的虚拟磁盘上，便可实现对交换分区的加密保护。具体实现如下：首先，创建一个指定大小(可以与原交换分区大小保持一致)的块文件；然后，将该块文件映射成一个加密的虚拟磁盘，并在虚拟磁盘上创建交换分区；最后，用新创建的交换分区替换掉系统原来的交换分区即可。

交换分区是操作系统在运行过程中需要经常访问的一块空间，对交换分区进行加密保护虽然可以有效地防治敏感数据的泄漏，但对操作系统整体性能的影响可能较大，所以在 SEFS 中，交换分区的加密功能一般只是在对安全性有较高要求的情况下才建议使用。

4.4 访问控制的实施

对于一个加密的虚拟磁盘，经常会出现需要多个用户共同访问的情况，这就需要对不同使用者的访问权限加以限制，以防止某些使用者对磁盘数据的恶意窜改。Linux 操作系统采用 9-bit 位的访问控制策略，对文件的使用者按照属主、同组用户、其他用户划分为三类，分别赋予不同的访问权限(读、写、执行)。这种实现方式对用户权限的控制粒度过于粗大，无法满足作者的要求，于是在 SEFS 中引入了 ACL(访问控制列表)技术。

在 SEFS 中，ACL 列表和其他关键信息(如：密钥长度、加密密钥、算法类型等)放在块文件的文件头中。对每个虚拟磁盘最多可以设置 20 个 ACL 条目，每个条目对应一个用户对该磁盘的访问权限，ACL 列表的第一个条目对应的是属主用户，只有属主用户可以修改其他用户的访问权限。存储结构如图 3。

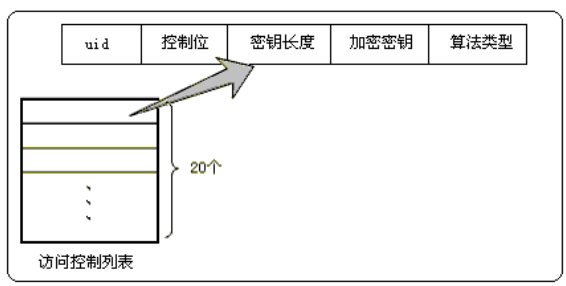

图 3 访问控制列表结构图

uid：即该用户的系统 uid，用来标识用户身份。
控制位：表示该用户对磁盘的读、写权限。
密钥长度：加密密钥的实际长度。
加密密钥：由该用户主密钥加密后的文件密钥。
算法类型：采用的加密算法的标识符。

5 性能分析

5.1 几个加密文件系统的比较

论文从加密位置、访问控制、可扩充性等几个方面，把 SEFS 与几个经典加密文件系统做了比较，见表 1。

表 1

	CFS	TCFS	AFS	SEFS
工作机制	nfs 服务端	nfs 客户端	--	绕回设备
加密位置	应用层	核心层	核心层	核心层
可扩充性	差	差	差	好
访问控制	一般	一般	好	好
数据存储	密文	密文	明文	密文
敏感信息的保护	不完整	不完整	--	完整
在磁盘存放密钥	否	是	否	否
对内核的依赖性	小	大	小	小

工作机制：CFS/TCFS 基于 nfs 文件系统的工作机制使得在对本地数据进行加密时性能非常低下，而 SEFS 则要高效得多。

加密位置：一般来说，在核心层进行加密操作将会得到更

好的效率和安全性。

可扩充性：SEFS 的加密引擎模块化机制使得用户可以很方便地添加新的加密算法。

访问控制：SEFS 增加了访问控制列表机制，可以对用户操作进行更加严格的控制。

数据存储：密文存储是保证数据安全性的一个重要条件。敏感信息的保护：SEFS 对文件名、文件大小、目录结构等敏感信息的保护更加彻底。

在磁盘存放密钥：TCFS 把用户主密钥存放在文件中的做法在一定程度上损害了系统的安全性。

对内核的依赖性：SEFS 以动态模块的形式加载，对内核的依赖性较小，可以支持各种版本内核，而 TCFS 目前只能支持 2.2.17 及以前的版本。

5.2 性能测试

论文对 SEFS 工作性能进行测试，目的是比较 SEFS 加密文件系统与普通 loop 文件系统和 ext3 文件系统相比，性能下降的程度。测试环境为：奔腾Ⅲ450MHz 处理器/128M 内存/8G(5400 转)硬盘，操作系统为 RedHat8.0，内核版本 2.4.18-14。测试内容是关于三种文件系统的数据读、写带宽的测量，数据量为 16M，读写单位为 1M。loop 设备和 SEFS 虚拟磁盘设备的容量均为 50M，创建的文件系统为 ext3，SEFS 加密采用的算法为 blowfish 算法。

表 2

	数据量(MB)	消耗时间(sec)		带宽(MB/s)	
		读	写	读	写
ext3FS	16×1MB	0.125218	0.274703	127.18	58.24
loopFS	16×1MB	0.129687	0.401582	123.37	39.84
SEFS	16×1MB	0.135759	0.404668	117.86	39.54

与 Ext3fs 相比，SEFS 的读、写带宽分别下降了 7.33%和 32.11%；与 loopFS 相比，SEFS 的读、写带宽分别下降了 4.7%和 0.75%。由于 TCFS 只支持 2.4.17 以前的内核版本，所以这里没有把 SEFS 与 TCFS 直接进行比较，而是将 TCFS 与 ext3 文件系统做了比较，测试结果显示，用 TCFS 对本地数据加密时，其读、写带宽与 ext3 相比分别下降了 95.65%和 97.63%。与 TCFS 相比，SEFS 的系统负载要小得多。

6 总结

论文提出一种基于 linux 的加密文件系统的设计与实现方案，对其工作原理和关键技术做了详细的论述，并对其工作性能做了进一步分析。与传统的加密文件系统相比，SEFS 最大的特点在于其良好的安全性，不仅对目录结构、文件大小等敏感信息做了完全的保护，而且实现了对系统交换分区的加密。SEFS 采用了 ACL 技术，能更好地满足多用户共享的情况。另外，SEFS 的工作性能较某些传统加密文件系统有了较大改善。

(收稿日期：2004 年 12 月)

参考文献

1.M Blaze.A Cryptographic File System for Unix[C].In：First ACM Conference on Communication and Computing Security，Fairfax VA，1993：158~165

2.E Zadok，I Badulescu，A Shender.Cryptfs：A Stackable Vnode Level Encryption File System.1998

3.Giuseppe Cattaneo，Luigi Catuogno，Aniello Del Sorbo.The Design and Implementation of a Transparent Cryptographic Filesystem for UNIX.FREENIX 2001，2001-06

4.Howard J H.An Overview of the Andrew File System[C].In：Proceedings of the USENIX Winter Technical Conference，Dallas，TX，1988-02

(上接 89 页)

所用时间略长于 L-M 方法，但比 SGAFANN 方法所需时间短很多，而其学习效果比前两种方法提高很多。

表 1 各种方法性能比较

训练方法	运行时间(秒)	拟合误差	测试误差
L-M	12	0.153801	0.203912
SGAFANN	11679	0.135746	0.202458
3GANN	53	0.018694	0.027644

4 结论

基于互补遗传算子的前馈神经网络三阶段学习方法(3GANN)，把复杂的前馈神经网络学习问题分解成结构辨识阶段、参数辨识阶段和剪枝阶段等前后联系的三个阶段，利用遗传算法的宏观搜索能力进行网络结构和初始参数的选择，采用局部搜索能力更强的常规优化方法进行进一步的参数辨识，同时，利用剪枝方法提高神经网络的泛化能力。该方法的采用，降低了整个神经网络学习过程的难度，提高了每一阶段的学习效率，在学习过程的可控性以及神经网络的逼近精度、复杂度和泛化能力之间达到了满意平衡。同时，根据在遗传算法优化中发现的组合遗传算子的互补效应而提出的把遗传算子分为交叉算子、变异算子和局部搜索算子三个功能互补又相对独立的互补遗传算子的概念，能够大大提高遗传算法的搜索效率，并对设计新的更高效率的组合遗传算子具有指导意义。

(收稿日期：2005 年 3 月)

参考文献

1.王小平，曹立明.遗传算法—理论、应用与软件实现[M].西安：西安交通大学出版社，2002

2.M T Hagan，M B Menhaj.Training feedforward networks with the Marquardt algorithm[J].IEEE Trans.Neural Networks，1994，5：989~993

3.Liang Wang，John Yen.Extracting fuzzy rules for system modeling using a hybrid of genetic algorithm and Kalman filter[J].Fuzzy Sets and systems，1999，101：353~362

4.陈小平，石玉，于盛林.快速寻优的遗传交叉策略[J].控制理论与应用，2002，19(6)：981~984

5.邓志军，邓守淇.进化神经网络中的变异算子研究[J].软件学报，2002，13(4)：726~731

6.徐秉铮，张百灵，韦岗.神经网络理论与应用[M].广州：华南理工大学出版社，1994