DOI:10.14188/j.1671-8836.2010.02.013
第56卷 第2期 武汉大学学报(理学版) Vol.56 No.2
2010年4月 J. Wuhan Univ.(Nat.Sci.Ed.) Apr. 2010, 223~226

文章编号:1671-8836(2010)02-0223-04

面向虚拟化平台的透明加密系统设计与实现

高汉军[1], 寇 鹏[1], 王丽娜[1,2†], 余荣威[1], 董永峰[1]
(1.武汉大学 计算机学院, 湖北 武汉 430072;
2.武汉大学 空天信息安全与可信计算教育部重点实验室, 湖北 武汉 430072)

摘 要:针对虚拟化平台下数据防泄漏系统的要求,本文结合可信平台模块(trusted platform module, TPM)的密钥管理优势,提出了一种基于eCrytpfs文件系统的透明加密系统设计与实现方法.该文件保护系统MeCryptfs(modified-eCryptfs)使用自定制TPM密钥管理模块,改善了eCryptfs用户空间的密钥管理部分,通过取消多用户模式进一步增强了整个系统的安全性.测试结果表明,该透明加密系统具有较高的性能,能够满足加解密透明性要求.

关 键 词:透明加密;密钥管理;虚拟机;MeCryptfs加密系统
中图分类号:TP 309 文献标识码:A

0 引 言

随着虚拟化技术的发展,其应用已从服务器虚拟化扩展到用户桌面系统虚拟化.国家信息安全测评认证中心调查结果表明[1]:在众多的攻击行为和事件中,最主要的安全事件是信息泄漏事件,且仍呈上升趋势.基于虚拟化平台的数据防泄漏研究已成为近年来信息安全研究领域的发展趋势.

虚拟化技术使用户可以在同一物理平台上同时支持多个隔离的虚拟域[1],分别用来运行不同安全级别的服务,进而在一个隔离良好的高机密级虚拟机中存储和处理敏感数据.加密文件系统是一种有效的数据加密存储技术,只有授权的用户才可以解读加密的数据.如基于Linux系统的加密文件系统(cryptographic file system, CFS)[2]、透明加密文件系统(transparent cryptographic file system, TCFS)[3]、安德鲁文件系统(Andrew file system, AFS)[4],基于Windows系统的加密文件系统(encrypting file system, EFS)[5]等.但这些加密文件系统的加解密效率较差,无法满足共享IT资源的多系统平台下透明加解密的效率要求.eCryptfs[6]加密文件系统的加解密效率较高,但它与数据防泄漏系统的自定制TPM密钥管理模块不兼容;同时与传统的加密文件系统类似,它运行于多用户模式下,存在一定的安全隐患.

近年来大力发展的可信计算技术理论采用从终端出发解决信息安全问题的研究思路,为防范信息泄漏提供了全新的思路,是对安全问题的本质回归[7].而虚拟化技术[8]提供的系统安全特性在一定程度上拓展了可信计算理论的应用.由此本文借助于可信计算技术,提出了一种MeCryptfs加密系统,该系统增加了与自定制TPM密钥管理模块兼容的守护进程模块,并修改了eCryptfs用户空间的密钥管理部分;取消了多用户模式,增强了整个系统的安全性.

1 对传统加密文件系统的分析

1.1 传统的加密文件系统

CFS实现了一个网络文件系统(network file system, NFS)服务器.由于其加解密操作在用户层完成,因此要频繁地在用户层与核心层之间进行数据交换,效率低下.TCFS实现了一个NFS客户端,其加解密操作在核心层完成.TCFS系统的安全性依赖于用户登录密码,而且加密密钥存放在磁盘上,这使得系统的安全性有所降低.AFS是分布式加密

收稿日期:2010-01-30 †通信联系人 E-mail:lnawang@163.com
基金项目:国家高技术研究发展计划(863)项目(2009AA01Z442, 2008AA01Z404);国家自然科学基金资助项目(90718006, 60970114)
作者简介:高汉军,男,博士生,现从事信息安全方面的研究, E-mail:ghjwhu@sina.com

文件系统,数据在该系统服务器上的明文存储方式存在很大的安全隐患.同时,这种方式使系统中的每一台服务器都要足够安全,否则,只要有一台服务器被入侵,整个系统的安全都将受到威胁.

Windows EFS 只能在 Windows 操作系统的 NTFS 卷上使用,具有一定的局限性.此外,如果需要重装操作系统,要同时备份加密文件和密钥证书,否则,重装系统后就无法再访问加密文件.

1.2 eCryptfs 文件系统

Linux 采用虚拟文件系统(virtual file system, VFS)[9]来管理底层文件系统. VFS 为 Linux 处理不同的文件系统提供了统一的接口.

eCryptfs[6]是 Linux 内核 2.6.19 版本中引入的一个企业级加密文件系统,为应用程序提供透明、动态、高效和安全的加密功能,它位于 VFS 与底层文件系统之间.对于 VFS 来说,eCryptfs 扮演了标准文件系统的角色,但它是将 VFS 传递下来的系统调用加上加解密操作,再传递到下层文件系统.

eCryptfs 为每一文件随机产生一个对称文件加密密钥(file encryption key, FEK).eCryptfs 使用用户口令或 TPM 的公钥对 FEK 进行加密保护;加密 FEK 的口令或公钥称为文件加密密钥(file encryption key encryption key, FEKEK),加密后的 FEK 则称为加密的文件加密密钥(encrypted file encryption key, EFEK).

虽然 eCryptfs 的加解密效率较高,满足了透明加解密的要求,但它与数据防泄漏系统中的自定制 TPM 密钥管理模块不兼容;而且 eCryptfs 运行在多用户模式下,低权限的用户就可以使用一些现有的技术提升自己的权限从而入侵系统关键区域.针对这些问题,本文提出了 MeCryptfs 加密系统.

2 MeCryptfs 加密系统设计方案

2.1 系统模块结构

MeCryptfs 系统的模块结构如图 1 所示,文件系统模块由用户空间模块和内核模块构成;守护进程模块由用户操作命令子模块、密钥处理子模块和脚本程序子模块构成.

用户操作命令子模块主要提供给用户密钥请求、更新、销毁命令.密钥处理子模块与下层密钥管理模块相连接;它根据用户操作命令调用相应的函数,获取密钥或相关信息,然后调用脚本程序子模块完成相应操作.脚本程序子模块调用了相关系统命令,使文件系统模块完成相应操作.用户空间模块修改了原

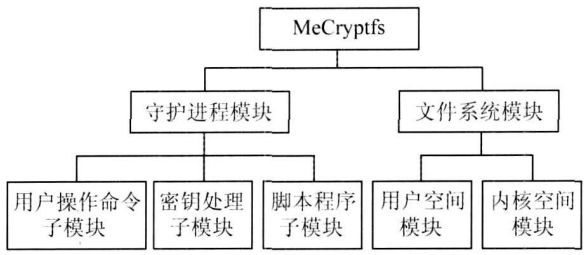

图 1 MeCryptfs 系统模块结构

eCryptfs 系统的密钥管理部分,更加精炼、安全、易维护.内核空间模块产生文件加密密钥,并调用底层文件系统完成文件的读写等操作.

MeCryptfs 使用的密钥是由数据防泄漏系统中的密钥管理模块进行管理的.下一节将对其进行简单介绍.

2.2 密钥管理模块

自定制的 TPM 密钥管理模块使用 TPM 对加密系统使用的会话密钥进行加密保护,大大增加了密钥的安全性.

密钥管理模块主要负责接收密钥处理命令,调用相应的 TPM 命令,完成密钥的处理.它与 MeCryptfs 的通信的数据包格式如图 2,其中"密钥信息"、"返回信息"和"错误类型"是一个联合体.

图 2 数据包格式

2.3 密钥处理方案

密钥处理是 MeCryptfs 中最重要的部分,如图 3 所示.它主要分为 3 个操作:密钥请求操作、密钥更新操作和密钥销毁操作.

MeCryptfs 在工作时,首先用一个会话密钥将用于存储文件密文的隐私目录挂载到存储对应明文的工作目录上,这两个目录合称为 P/W 目录对.

1)密钥请求操作过程:

(a)守护进程与密钥管理模块交互,获得密钥并将其写入临时密钥文件.(b)守护进程发出挂载文件系统的命令.(c)文件系统模块读取密钥,删除

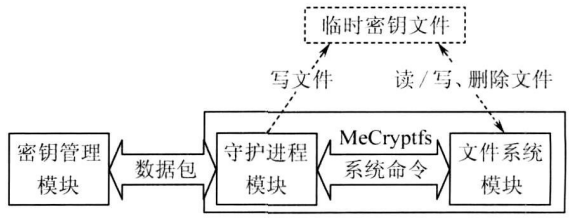

图 3 密钥处理操作图

临时密钥文件并利用此密钥完成挂载.

2) 密钥更新操作过程:

(a) 守护进程与密钥管理模块交互获取新的密钥, 并将其写入临时密钥文件. (b) 守护进程发出创建新 P/W 目录对的命令, 文件系统模块完成创建操作. (c) 守护进程发出在新 P/W 目录对上挂载文件系统的命令, 文件系统模块读取密钥, 删除临时密钥文件并完成挂载. (d) 守护进程发出复制旧 P/W 目录对中文件到新 P/W 目录对中命令, 文件系统模块完成复制操作. (e) 守护进程发出卸载和删除旧 P/W 目录对的命令, 文件系统完成卸载和删除操作. (f) 守护进程发出将新 P/W 目录对名称改为旧 P/W 目录对名称命令, 文件系统模块完成改名操作.

3) 密钥销毁操作过程:

(a) 守护进程与密钥管理模块交互, 传送密钥销毁的命令和结果. (b) 如果销毁成功, 守护进程发出卸载文件系统和删除隐私目录的命令, 文件系统模块完成卸载和删除. (c) 守护进程发出创建同名隐私目录的命令, 文件系统模块完成创建操作.

2.4 系统实现

本系统需要编码实现的部分主要有守护进程模块以及对 eCryptfs 用户空间模块的修改.

1) 守护进程模块的实现

用户命令操作子模块向用户提供了操作 MeCryptfs 的接口, 主要包括 3 个命令: 请求、更新、销毁密钥. 它们调用密钥处理子模块中的相关函数, 其基本流程为发送命令包、接收应答包、解析应答包、调用脚本程序子模块. 守护进程模块和密钥管理模块通过管道进行数据包的传送. 脚本程序子模块通过一系列系统命令对 MeCryptfs 的文件系统模块进行操作, 以完成密钥的处理.

2) eCryptfs 用户空间模块的修改

密钥由密钥管理模块产生. MeCryptfs 文件系统模块中的用户空间模块从临时密钥文件中获取密钥, 代替原来的用户输入.

3 结果分析与讨论

本文对 MeCryptfs 工作性能进行了测试, 目的是分析本系统对性能下降的影响程度. 测试环境为: Inter Pentium M 处理器, 1.70 GHz/768 Mb 内存/Ubuntu 9.04 操作系统, Linux 2.6.30 内核/ext3 文件系统. 测试在是否挂载 MeCryptfs 的情况下, 复制数据的时间消耗. 测试选用的加密算法是高级加密标准(advanced encryption standard, AES), 密钥长度 128 位. 表 1 给出了性能测试的结果.

表 1　未使用与使用 **MeCryptfs** 性能测试结果

数据量/MB	消耗时间/s		带宽/Mb·s^{-1}	
	未使用	使用	未使用	使用
699.3	55.157	69.247	12.68	10.10
30	0.189	1.111	158.73	27.00

在复制 699.3 Mb 的大数据时, 使用 MeCryptfs 的情况下, 时间消耗增加了 25.5%, 带宽下降了 20.35%, 性能下降较少. 在复制 30 Mb 这样的小数据时, 带宽下降了 82.99%, 性能下降较多, 但因其本身消耗时间较少, 用户仍可承受. TCFS 与 ext3 相比其读写带宽分别下降了 95.65%和97.63%[10]. 与 TCFS 相比, MeCryptfs 的系统负载要小很多.

4 结　论

本文基于 eCryptfs 文件系统, 提出了面向虚拟化平台的透明加密系统(MeCryptfs). 该系统改善了 eCryptfs 用户空间的密钥管理部分, 并取消多用户模式, 增强了安全性. 与传统文件加密系统相比, 本系统主要有以下几个方面的优点:

1) 效率高: 根据测试结果, 使用本系统对文件进行加解密过程的效率较高.

2) 实现简单: 由系统结构图(图 1)来看, 系统各模块之间关系清晰明了, 模块间的耦合程度小, 便于代码的编写和调试.

3) 数据安全性高: 本系统不在磁盘介质上存储会话密钥, 每次加载和使用都必须从密钥管理模块获取, 增加了密钥的安全性. 系统采用了管道技术, 一旦数据被接收进程读取, 管道中的数据就被清空, 即密钥数据包被清空, 提高了安全性.

本系统存在一个安全性弱点: 临时密钥文件的存在. 按照正常的流程, 该文件由守护进程模块建立并写入密钥. 用户空间模块读取密钥后便删除该文件. 但如果程序中途意外终止, 临时密钥文件就会留在磁盘介质上, 增加了密钥的泄露风险.

MeCryptfs 只能从自定制 TPM 密钥管理模块获取密钥. 获取密钥需要提供密钥名称和口令, 密钥名称和口令是不保存的. 所以, 即使用户知道了密钥, 也无法使用. 通过加强对密钥名称与口令的保护和验证就可以有效地降低此类安全威胁.

参考文献:

[1] 苏航, 吴庆波, 李永. 基于虚拟机技术的安全系统研究

[J]. 计算机安全, 2008, 3: 49-52.
Su Hang, Wu Qingbo, Li Yong. Research on safe-system based VM technology[J]. *Computer Security*, 2008, 3: 49-52(Ch).

[2] Zadok E, Badulesce I, Shender A. Cryptfs: A stackable vnode level encryption file system[R]. New York: Department Computer Science, Columbia University, 1998.

[3] Giuseppe C, Luigi C, Aniello D S. The design and implementation of a transparent cryptographic filesystem for UNIX[C]//*FREENIX Track*, 2001, *USENIX Technical Conference*, Boston: USENIX Association, 2001.

[4] Tobbike R. Distributed file systems: focus on Andrew file system/distributed file service (AFS/DFS)[C]// 13*th IEEE Symposium on Mass Storage Systems*, Annecy: Computer Society Press, 1994.

[5] 黄敢为. 基于Windows XP的EFS加密文件系统研究[J]. 科学技术与工程, 2008, **8**(15): 4158-4160.
Huang Ganwei. Encrypting files system application for Window XP[J]. *Science Technology and Engineering*, 2008, **8**(15): 4158-4160(Ch).

[6] Halcrow M. eCryptfs: A stacked cryptographic filesystem[J]. *Linux Journal*, 2007, **156**: 54-58.

[7] 张焕国, 罗捷, 金刚, 等. 可信计算研究进展[J]. 武汉大学学报(理学版), 2006, **52**(5): 513-518.
Zhang Huanguo, Luo Jie, Jin Gang. Development of trusted computing research[J]. *Journal of Wuhan University (Natural Science Edition)*, 2006, **52**(5): 513-518(Ch).

[8] 张献涛. 基于VMM的系统虚拟机关键技术及安全问题研究[D]. 武汉: 武汉大学计算机学院, 2008.
Zhang Xiantao. The research on key technologies of VMM-based system virtual machine and security considerations[D]. Wuhan: School of Computer, Wuhan University, 2008(Ch).

[9] 廖光忠. Linux虚拟文件系统机制[J]. 计算机技术与发展, 2006, **16**(11): 114-116.
Liao Guangzhong. Research on mechanism of virtual file system of Linux[J]. *Computer Technology and Development*, 2006, **16**(11): 114-116(Ch).

[10] 邢常亮, 卿斯汉, 李丽萍. 一个基于Linux的加密文件系统的设计与实现[J]. 计算机工程与应用, 2005, **17**: 101-104.
Xing Changliang, Qing Sihan, Li Liping. The design and implementation of an encrypted filesystem for Linux[J]. *Computer Engineering and Allpications*, 2005, **17**: 101-104(Ch).

Design and Implementation of a Virtualization Platform Oriented Transparent Cryptographic System

GAO Hanjun[1], KOU Peng[1], WANG Lina[1,2], YU Rongwei[1], DONG Yongfeng[1]
(1. School of Computer, Wuhan University, Wuhan 430072, Hubei, China;
2. Key Laboratory of Aerospace Information Security and Trusted Computing of Ministry of Education, Wuhan University, Wuhan 430072, Hubei, China)

Abstract: For the requirements of the virtualization platform of data leakage prevention system, considering the trusted platform module's advantages of key management, a transparent cryptographic filesystem based on the eCrytpfs is proposed and the corresponding implementation is introduced in this paper. The MeCryptfs, modified-eCryptfs, improves the key management in eCryptfs's userspace by customizing a TPM key management module and eliminates multi-user mode to enhance the security of the whole system. Finally, test results show that the new transparent cryptographic system has a high performance to meet the transparency requirements of encryption and decryption.

Key words: transparent cryptographic; key management; virtual machine; MeCryptfs cryptographic system