·微机网络与通讯·

基于可信计算平台的加密文件系统*

章 勤, 刘树明

(华中科技大学计算机学院, 服务计算技术与系统教育部重点实验室,
集群与网格湖北省重点实验室, 武汉, 430074)

摘 要: 普通的加密文件系统能够对文件内容进行安全保护, 加密文件与密钥被绑定在一起。但是, 密钥仅仅通过弱口令来进行安全保护, 这对系统来说是一个安全隐患, 因此密钥保护是迫切需要解决的问题. 通过运用 TPM密钥树对整个文件系统中的密钥进行加密保护, 将加密密钥同 TPM所在平台进行绑定, 从而实现密钥的安全保护, 增强了整个系统的安全性。通过采用基于 HMAC的数据检验, 在保证安全性的同时, 又提高了完整性校验的性能。

关键词: 加密文件系统; 用户空间文件系统; 可信平台模块; 可信软件栈

中图分类号: TP3 文献标识码: A 文章编号: 1002-2279(2008)01-0039-04

An Encrypted File System Based on Trusted Computing Platform

ZHANG Qin, LIU Shu-ming

(HuaZhong University of Science & Technology, Services Computing Technology and System Lab,
Cluster and Grid Computing Lab, Wuhan 430074, China)

Abstract: Ordinary encrypted file system can protect the content of files, and the encrypted file is bound with the relevant key. But the key is just protected securely by the short password. This is a defect for the system and this question need to be solved exigently. This paper proposes a method based on TPM key-tree and data check-up with HMAC. The TPM key-tree encrypts all keys in file system and binds the keys with the TPM platform, thus keys are protected safely and security of the whole system is enhanced. The data check-up with HMAC not only ensures security but also improves the performance of integrality check-up.

Key words: Encrypted file system; FUSE; TPM; TSS

1 引 言

随着信息技术和互联网的日益发展, 各种信息资源的共享程度越来越高, 随之而来的就是越来越严重的安全问题。若不采用适当的安全措施, 那么由于截取、窜改而带来的某些机密信息的泄漏和丢失会造成巨大的损失, 因此对加密文件系统的需求非常迫切。

传统的加密文件系统 CFS在对加密文件共享时, 文件拥有者直接将文件加密密钥传递给其他用户, 因此其他用户对密钥的管理方式直接影响系统的安全性。 SIRIUS[1] 在传统的加密文件系统基础之上, 解决了多用户加密文件系统的文件共享和密钥管理问题。整个体系结构建立在基于身份密钥的加密方法上, 安全地实现了多用户共享, 所有加密操作都在用户机器上处理, 很大程度上提高了系统的安全性。但问题在于仍然没有解决好密钥的安全保护和绑定问题。

本在 SIRIUS的基础上提出了基于可信计算平台的加密文件系统。通过可信平台模块 TPM[2] (TrustedPlatform Module)来进行密钥的安全保护, 通过将加密文件、加密密钥同 TPM平台进行绑定, 来进一步增强系统的安全性。

2 基于可信计算平台的加密文件系统总体框架

在用户空间文件系统开发模型 FUSE[3] (Filesystem in Userspace)、TPM及其软件栈 TSS[4]

* 基金项目: 国家自然科学基金重大研究计划项目, 网络计算应用支撑中间件 /网络计算安全支撑环境 (90412010)
作者简介: 章勤 (1955-), 女, 湖北武汉人, 硕士研究生, 教授, 主研方向: 网格计算, 网格安全。
收稿日期: 2006-12-22

(TCG Software Stack)的基础之上,以 FTP服务器作为存储部件,开发出基于可信计算平台的加密文件系统,系统中所有文件经过客户机上的 TPM进行加解密处理,存储部件不参与任何加解密过程,从而保证系统的安全性。

系统的总体架构如图 1所示,整个加密文件系统主要由 6部分组成:FUSE内核模块、FUSE用户模块、TPM、TPM驱动程序、TSS和文件服务器。其中 FUSE内核模块和 FUSE用户模块是整个系统的核心,FUSE内核模块将虚拟文件系统 VFS提交的读写请求传递给应用层的 FUSE用户模块,由 FUSE用户模块来执行具体的文件读写。FUSE用户模块除了执行具体的文件读写外,还需要根据加密文件系统的逻辑结构提供对所有加密文件的加解密处理。

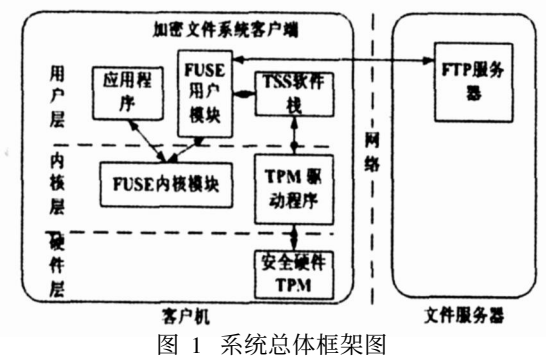

图 1 系统总体框架图

2.1 TPM和 TSS

TPM是一个含有密码运算器件和存储部件的小型嵌入式系统。在用户初始化 TPM时,产生 SRK密钥,保存在 TPM内部,并通过该密钥来加密保护所有将来产生的密钥以及需要保护的外部数据。敏感的私钥从来不会在硬件之外以明文的形式呈现。通过使用 TPM进行加密操作,可以将所加密的数据绑定到该 TPM所在的平台上。通过对在 BIOS里对 TPM的状态进行配置,可以使得 TPM拒绝加密任何信息。

TSS是可信平台模块的支撑软件,它在 TPM之上又扩充了 TPM的功能.由 TSS Device Driver Library、TSS Core Service和 TSS Service Provider三层组成。它主要提供了 TPM管理、上下文管理、密钥管理、数据加解密、数据 HASH、策略和属性管理等功能。

2.2 FUSE

FUSE是一个用户空间文件系统开发模型。通过提供一个抽象的接口,使用户可以复用已开发的应用层代码来开发自己的文件系统。FUSE主要由三个模块组成:FUSE内核模块、FUSE开发库和 FUSE用户模块,其工作原理如图 2所示。

a)应用程序通过系统调用中的读写函数进行 I/O调用。

b)虚拟文件系统 VFS根据文件系统的类型将应用程序的读写请求转交给 FUSE内核模块去执行。

c)FUSE内核模块同用户层开发的 FUSE用户模块通过 /dev/fuse虚拟设备和信号机制进行通信,具体的文件读写和文件系统逻辑处理任务由 FUSE用户模块来完成。

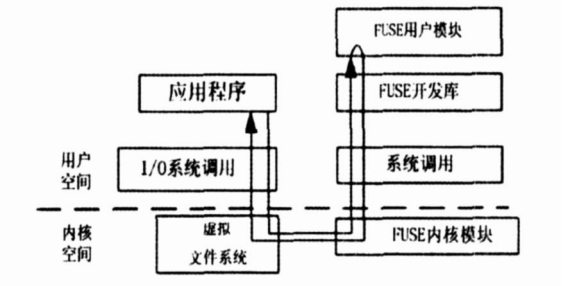

图 2 FUSE原理图

由于在 TPM硬件上开发出来的 TSS软件栈都处在用户层,而本系统又要使用 TPM的功能进行数据的加密保护,从而选择借助用户空间文件系统模型进行开发。

3 基于 HMAC的数据检验

基于密码学的数据完整性检验通常有两种方式:一种是通过 HMAC实现,另一种是通过公钥签名实现。两种方式都是公认的具有较高安全性的完整性检验方式。与前一种方式相比,采用后一种方式算法较为简单,但公钥签名同时在更大程度上增加了系统读写的时间消耗。综合考虑系统的性能和用户量,系统中采用基于 HMAC的方式,其算法描述如下:

(1)在文件创建时,由文件 owner随机产生一个 HMAC密钥,记为 K^{w}_{hmac}。hmac表示密钥用于文件 HMAC值的计算和验证,w表示密钥由具有写权限的用户共享。

(2)当需要授权给用户 A时,如果是写权限,则创建者将 K^{w}_{hmac}共享给 A;如果是读权限,则创建者根据 K^{w}_{hmac}计算出 $K^{r_i}_{hmac}$并共享给 A。r_i用来表示第 i个具有读权限的用户。

(3)$K^{r_i}_{hmac}$由 K^{w}_{hmac}通过 one-way HASH函数计算得出:$K^{r_i}_{hmac}=h(K^{w}_{hmac}, i)$。

(4)修改文件时,修改者使用 K^{w}_{hmac}为所有具有写权限的用户计算文件 HMAC检验码;使用 $K^{r_i}_{hmac}$为所有具有读权限的用户分别计算文件 HMAC检验

码。

(5)读取文件时, 读取者需找到自己的 HMAC检验码。如果用户具有写权限, 那么使用 K^{w}_{hmac}验证文件完整性; 如果用户仅具有读权限, 那么使用 $K^{r_i}_{hmac}$验证文件完整性。

综上所述, 修改文件时, 用户必须计算系统中每个用户的文件 HMAC检验码; 读取文件时, 用户获取自己的文件检验码, 并验证文件的完整性。所有具有写权限的用户的 HMAC密钥相同; 任何一个仅具有读权限的用户的 HMAC密钥都不同。具有写权限的用户都有 K^{w}_{hmac}, 所以能计算每个具有读权限的用户的 HMAC密钥。由于 one-way HASH具有单向性, 因此所有仅具有读权限的用户很难由 $K^{r_i}_{hmac}$计算出 K^{w}_{hmac}, 从而保证了系统的安全性。将每个文件所对应密钥的集合称作该文件的密码元数据。具有写权限的用户利用 K^{w}_{hmac}冒充创建者对文件的密码元数据进行修改是算法的一个主要缺陷, 在下一节中将运用多层密钥保护结构, 将密码元数据作为密钥树中的一个节点进行保护, 从而来保证密码元数据的机密性和完整性。

4　多层密钥保护结构与加密文件操作

在可信计算中, TPM通过硬件来保护根密钥的安全性, 然后采用密钥树的方式从根开始层层加密, 从而提供对整个树中节点的安全保护。因此将整个文件系统树型结构中的各节点引入密钥树, 运用 TPM密钥树保护思想能够对整个文件系统提供安全保护。其结构如图 3所示。

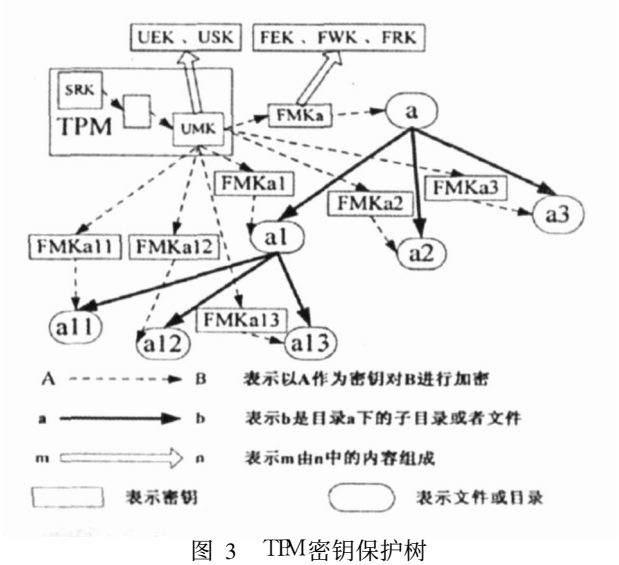

图 3　TPM密钥保护树

其基本思想是: ①每个 TPM中有且仅有一个存储根密钥 SRK, SRK作为存储保护的根密钥, 与硬件绑定在一起。用户通过 TPM产生一个用户主密钥 UMK, UMK通过 SRK直接进行加密保护或者通过 TPM产生的其他密钥进行加密保护; ②每个用户的 UMK都由两对非对称密钥组成: 一个用来加密, 将其称作用户加密密钥 UEK, 另一个用来签名和验证, 将其称作用户签名密钥 USK。通过 UEK的公钥来加密保护文件主密钥 FMK, 从而使得只有合法的用户才可以得到 FMK的明文。USK用来保护加密后的 FMK的完整性; ③每个 FMK都由三类对称密钥组成: 第一类密钥用来对文件进行加密保护, 将其称作文件加密密钥 FEK, 第二类密钥用来计算文件 HMAC校验值, 由具有写权限的用户拥有, 将其称作文件写密钥 FWK, 第三类密钥用来对文件的 HMAC进行校验。由具有读权限的用户拥有, 将其称作文件读密钥 FRK。上一章中的 K^{w}_{hmac}和 $K^{r_i}_{hmac}$分别对应 FWK和 FRK。

4.1　创建文件

在系统中创建一个文件需要经过以下步骤:

(1)在创建文件前, 随机产生三个密钥 FEK, FWK和 FRK(对于具有写权限的用户 FRK=FWK)。

(2)同用户交互, 并从 TPM中加载正确的 UEK和 USK。UEK用来对(1)中随机产生的三个密钥进行加密保护, 将加密结果称为该用户的 EKB(加密密钥块)。

(3)对(2)中产生的 EKB进行 HASH, 并通过 USK私钥对该 HASH进行签名, 用来保证数据的完整性。

(4)EKB及其 HASH签名组成创建者访问该文件时需要用到的密码元数据, 并保存在 md-file中。

4.2　共享文件

假设 Alice是文件的创建者, 她要将文件的读或写权限授予 Bob, 需要经过以下步骤:

(1)Alice通过某种方式得到 Bob的 UEK公钥。

(2)Alice读取该文件的 md-file, 通过自己的 USK公钥验证密码元数据的完整性, 然后从中获取自己的密码元数据。

(3)Alice用自己的 UEK解密密码元数据后, 按照基于 HMAC的数据检验算法计算出 Bob的 HMAC密钥 FRK和 FWK, 然后 Alice用 Bob的 UEK公钥加密 FEK, FWK和 FRK, 对于读权限用户 FWK无效。其加密结果是 Bob的 EKB。

(4)对(3)中产生的 EKB进行 HASH, 并通过 Alice的 USK私钥对该 HASH进行签名, 用来保证数据的完整性。

(5)将 EKB及其 HASH签名作为 Bob访问该文件时的密码元数据,并增加到 md— file中。

5 性能测试

系统运行环境为 Celeron CPU 2.00GHz,256MB内存,对称密码学采用 128位密钥的 AES算法,公钥密码学采用 1024位密钥的 RSA加密和签名算法,用户数目 10个。实验数据如表 1所示(TCBFS表示本系统)。因为需要产生加解密密钥和 HMAC密钥,同时还要对文件数据和密码元数据进行加密、签名等操作。因此文件操作要比 NFS慢的多,对于 1MB大小的文件,读操作所花费的时间是 NFS的 3倍左右,而写操作所花费的时间是 NFS的 8倍左右。与 SIRIUS对比,TCBFS的读写速度慢一些。从表 1可以看出,TCBFS创建和共享文件的速度大约为 SIRIUS的 1.4倍,对 1MB文件进行写操作的速度为 SIRIUS的 1.6倍,但由于采用基于 HMAC的完整性校验方式,明显提高了读操作的速度,两者基本相同。由于所有 RSA密钥及加解密操作都采用 TPM硬件来完成,而目前的 TPM硬件加解密速度比 openssl等软件要慢,同时用户空间文件系统开发模型的使用也增加了系统运行的时间消耗。考虑到上述因素的客观性,在系统实现中,对目录树加密元数据和加密文件采用较好的缓存和索引算法来尽量提高系统的性能。

表 1 性能对比表(单位:ms)

测试	文件大小	NFS	SIRIUS	TCBFS
创建	0	0.28	10.2	14.3
共享	0	0.21	6.2	9.1
读	1MB	67.7	154.7	159.2
写	1MB	71.2	434.4	649.4

6 结 束 语

提出了一种基于可信计算平台的加密文件系统的设计方法。在整个系统中,每个文件采用单独的加密密钥,同时所有文件目录的加密密钥都由用户的 TPM加密主密钥保存,从而既有效地实现了对密钥的安全管理,又将加密文件的内容绑定到该 TPM机器上,大大增加了系统的安全性。同时,采用用户空间文件系统模型的设计,隐藏了几乎全部的密码学操作,实现了对用户的透明性。下一步,为了进一步提高系统的安全性,重点将研究用户动态拥有文件读写权限时的密钥更新问题。

参考文献:

[1] Goh E J, Shacham H, Modadugu N, SiRiUS: Securing Remote Untrusted Storage[J]. Proceedings of the Distributed Systems Security Symposium 2003(11): 131—145.

[2] Dalit Naor, Amir Shenhav, Toward securing untrusted storage without public— key operations[J]. Proceedings of the 2005 ACM workshop on Storage security and survivability, 2005(5): 51—56.

[3] TCG: TCG Infrastructure Architecture[EB/OL]. http://www.trustedcomputinggroup.org/specs/IWG/.

[4] Miklos Szeredi, FUSE: Filesystem in Userspace [EB/OL]. http://fuse.sourceforge.net

(上接第 38页)

则允许进一步的 Read/Write操作。Mifare1射频卡上有 16个扇区,每个扇区都可分别设置各自的密码,互不干涉。因此每个扇区可独立地应用于一个应用场合。卡中的其他扇区由于有其各自的密码,因此不能对其进行进一步的操作。

Control& Arithmetic Unit——控制及算术运算单元:这一单元是整个卡的控制中心,对整个卡的各个单元进行微操作控制,协调卡的各个步骤;同时它还对各种收/发的数据进行算术运算处理,递增/递减处理,CRC运算处理等等,是射频卡中内建的中央微处理机(MCU)单元。

RAM/ROM——存储器单元:RAM主要配合控制及算术运算单元,将运算的结果进行暂时存储,ROM中固化了射频卡运行所需要的必要的程序指令,由控制及算术运算单元取出去对每个单元进行微指令控制。

Crypto Unit——数据加密单元:该单元完成对数据的加密处理及密码保护。加密的算法可以为 DES标准算法或其他。

EEPROM INTERFACE/EEPROM MEMORY——存储器及其接口电路:主要用于存储数据。EEPROM中的数据在卡失掉电源后(卡离开读写器天线的有效工作范围内)仍将被保持。用户所要存储的数据被存放在该单元中。

6 结 束 语

射频识别被许多专家称为未来的明星产业,究竟它的魅力何在,可以从其具备的特性来窥探一二。RFID自动识别科技可以突破条形码或接触式 IC卡须人工扫瞄、一次读一个的限制;也可以在恶劣的环境下作业、长距离的读取、同时读取多个卷标等。另外还具有实时追踪、重复读写内容及高速读取的优势,应用范围之广让人对它充满了无限的期待。

参考文献:

[1] 廖民. RF IC技术的若干发展方向[J]. 电子产品世界,2005,8(1):42—46.

[2] 大卫,李际. 全球掀起 RF ID热潮[J]. 电子产品世界,2004,11(1):121—122