# 分布式安全存储技术*)

韩德志[1,2]

(广东外语外贸大学信息学院 广州 510420)[1] (广东省信息安全重点实验室中山大学 广州 510275)[2]

**摘 要** 迅速增加的敏感数据迫使企业更加重视存储系统的安全性。本文详细地讨论了分布式安全存储系统所提供的安全服务,并对这些安全服务进行了比较,为企业安全存储系统的设计和构建提供有益的帮助。

**关键词** 网络安全,存储安全,网络文件系统,加密文件系统,存储入侵检测系统

## The Securing Distributed Storage Techniques

HAN De-Zhi[1,2]

(School of Information Science, Guangdong University of Foreign Studies, Guangzhou, Guangdong 510420)[1]

(Guangdong Key Laboratory of Information Security Technology, Sun Yat-sen University, Guangzhou 510275)[2]

**Abstract** The rapid increase of sensitive data has forced enterprises to pay serious attention to storage security. In the paper, we detailedly discuss the security services provided by the existing storage systems, and compare them, which can provide useful help for the design of the enterprise security storage system.

**Keywords** Network security, Storage security, Network file systems, Cryptographic file systems, Storage-based IDS

## 1 引言

当前信息技术已经广泛、深入地渗透到社会生活的方方面面。一方面,数据作为信息在计算机系统中存在的形式越来越成为企业、单位和个人最重要的财富。截至2002年底,财富500强企业平均拥有的存储容量达到48TB以上。国际技术集团(ITG)的一项研究表明,按照目前的趋势发展,这一数字到2007年底将超过230TB,且存储相关支出占企业IT预算的比重也将从2002年的17%增长为2007年的22%。另一方面,迅速增加的敏感数据,如健康记录、客户记录、金融数据等等迫使企业更加重视存储系统的安全性。如果没有采取存储安全性措施,数据存储设备及其上的数据将成为攻击者的既定目标。任何公司或机构,无论大小,都不敢置数据安全风险于不顾,由安全性问题带给企业的损失可能会远远超出在安全解决方案上的投资。据统计,仅美国的公司在2001年便为消除计算机病毒造成的损害而花费了大约123亿美元,而且病毒的攻击可能会造成更高的财产损失。任何人都无法保证系统免受攻击者恶意行为的伤害。存储系统设计人员和管理人员必须面对这样一个现实:很多种恶意攻击可以对网络中的设备以及其上的数据造成损害。

网络存储安全性包括网络安全性和存储安全性。CSI/FBI计算机犯罪与安全调查结果显示,大部分企业都遭受来自企业内部和外部的双重攻击。一般而言,网络安全系统位于网络存储系统的边界,负责检测、抵御外来的攻击,对内部攻击无能为力。存储系统的内部攻击需要存储安全系统来承担,从而形成安全存储系统。目前网络安全的研究已较为完备,出现了许多协议、标准和产品,如IPSec、防火墙等。存储安全性研究尚处于研究和应用的试验阶段。

目前,存储安全存在两种研究思路:①借鉴信息安全的S.I.S特性(安全性、完整性、保险性,简称S.I.S特性)[1,2],为某一特定应用提出专门的解决方案,如增强文件服务器的安全性、客户端加密文件系统、客户端直接访问磁盘的认证机制和高度可扩展文件系统[3,4]、基于存储的入侵检测系统[5]、远程备份等。安全性要保证数据不能被非法用户或未被授权用户复制和读取,完整性意味数据不能被非法修改(包括黑客、病毒和非法用户),保险性要求数据不能丢失(包括攻击、灾难性事件等);②从存储系统的体系结构入手,寻找一种安全的、高效的存储模式,比如对象存储模式等[6]。限于篇幅,本文从第一个方面着手对目前的存储系统提供的安全服务进行较全面的论述。

## 2 存储安全服务及分类

### 2.1 存储安全服务

存储系统提供的存储安全服务主要包括:认证和授权(Authentication and Authorization)、可用性(Availability)、机密性和完整性(Confidentiality and Integrity)、密钥共享和密钥管理(Key Sharing and Key Management)、审计和入侵检测(Auditing and Intrusion Detection),以及可使用性、可管理性和性能等方面。

(1)认证和授权。认证和授权是一个存储系统应该提供的最基本的安全服务。认证是确定一个实体或信息源的身份,前者叫实体的认证或鉴别,后者叫信息认证。存储服务器在允许数据的生产者、消费者和管理者访问(读或写)之前,应该认证他们的身份是否合法,如果合法,则给一定的访问权限,这个过程叫授权。认证是相互的,数据的生产者和消费者也要认证存储服务器的身份,认证通过后它们之间建立相互

*)本文受国家自然科学基金项目(60673191, 60603074)、国家973重大基础项目(2004CB318203)、广东省信息安全重点实验室项目、中国博士后科学基金项目(200060390749)、广东外语外贸大学科研创新团队项目(GW2006-AT-005)资助。韩德志 教授、博士后,主要研究方向:网络存储系统、网络存储安全。

信认的关系。信息认证是实体对另一个实体发送的原始信息的认证。认证可使用口令、数字签名和信息认证码(MAC)等技术。授权可通过访问控制列表(Unix 的 ACL 等)或使用容器证书(证书中列举了证书所有者的访问权限)。

(2)可用性(Availability)。大部企业需要保持数据的可用性。系统失效和拒绝服务攻击(DoS)是难以阻止的,一个嵌入强加密技术的系统也不能保证系统的可用性。目前常采用备份/恢复、容错等技术保证系统的可用性。

(3) 机密性和完整性。当数据在一个或多个远程存储服务器上产生、传输和存储时,面对未被授权入侵者的破坏、修改和重发攻击显得很脆弱。并且,一个恶意服务器能用旧版本文件替代当前版本的文件。当数据在传输或存储在媒体上时,保证其安全性是关键。通过加密可以保证对非认证用户的机密性,通过数字签名或信息认证码可以保证数据的完整性。给每一个会话设置时间戳或随机数可以阻止敌手重发攻击。使用安全套接层协议 SSL 和 IPSec 协议可以保证数据传输安全,但存储服务器要先解密数据,然后再存放磁盘上。

数据的机密性和完整性也可通过端到端的安全来保证,即用户发送数据时在客户端加密和签名。读取数据时,先从存储服务器读到客户,然后检验完整性和解密。端到端的安全放置最小的信任到存储服务器边,并且只有持有相应密钥的用户才能访问这些数据,保证这些密钥的安全性极为重要。

(4)密钥共享和密钥管理。多用户网络文件共享是很普通的,共享一个文件的所有用户也必须共享这个文件的加密密钥。这些密钥的有效性和扩展管理是非常重要的,当从一个组中删除一个用户或合并两个组时,要求重新加密共享文件并且重新分布新的密钥。密钥管理的另一个重要方面是密钥恢复,即恢复丢失的密钥技术。一个密钥恢复系统是一个有备份解密容器的加密系统,这个解密容器允许在某些条件下被授权的用户在一个或多个信任实体提供信息的帮助下解密密文。这些信任实体有专用的数据恢复密钥,但这些密钥的存放、保存和删除是很重要的。

(5)审计和入侵检测。存储系统必须维持重要活动的审计日志。审计日志对系统恢复、入侵检测等非常重要。入侵检测领域已有广泛的研究,入侵检测系统(IDS)使用各种日志(如网络日志和数据访问日志)和网络流(如 RPCs、网络流等)来检测和报告攻击。在不同级配置 IDS(如文件服务器、存储服务器或存储设备等)是非常重要的。

### 2.2 存储安全分类

本文不考虑数据的可用性、系统失效和性能等,只根据存储系统提供的安全服务对安全存储系统进行分类。

(1)基于网络的文件系统。属于这类文件系统,存储服务器认证每个用户并且允许用户访问之前检查用户是否有合适的访问权限。这些系统大部分不提供端到端的数据加密安全,并且不保证存储在存储服务器中的数据完整性和机密性。并且,这些系统假定文件服务器和系统管理者是可信任的。

另外,通过口令认证、通过简单的 Unix 访问控制(用户或组标识符)实现系统的访问控制。例如,早期版本的网络文件系统(NFS)使用中心服务器认证客户并给用户和组信息给客户,然后传递这些信息到文件系统(在文件请求期间),再传到做访问控制决定的操作系统。因此,这种安全是极其原始,主要依靠文件系统认证和加密。最近的文件系统可实现分布式的认证和使用公钥基础(PKI)实现跨域的分布式文件共享。

(2)加密文件系统。这些文件系统的目标是实现端到端的安全。客户端的加密文件系统在客户端对数据进行加密,然后把加密数据存放在存储服务器中,这些数据对非认证用户和系统管理员都保密。这些系统嵌入加密操作(加密/解密或签名/校验)到文件系统内部。给文件系统最小的信任,从不能看到数据的明文包括加密/解密过程。加密文件系统又进一步分为共享和非共享加密文件系统系统两类。共享加密文件系统在一组用户之间共享加密文件,这些文件系统在实现加密服务时有更复杂的密钥管理,依靠这些复杂的密钥管理完成密钥共享和密钥撤消。非共享加密文件系统不存在密钥共享和密钥撤消问题,在这些文件系统中一个用户要与其它用户共享加密文件时,必须把他的文件加密密钥给其他用户。

(3)基于存储的入侵检测系统(IDS)。一个基于存储的入侵检测系统是一个嵌入存储设备或文件服务器的入侵检测系统。它分析数据的访问模式和数据修改特点,寻找攻击行为。运行在存储服务器上的基于存储的入侵检测系统的主要优点是:当主机操作系统的安全受到威胁时,而基于存储的 IDS 的安全不会受到威胁,即当主机的操作系统安全受到威胁时,基于存储的 IDS 仍能正常运行。因此,存储服务器通过分析客户请求在线检测攻击行为。

## 3 基于网络的文件系统

基于网络的文件系统主要包括:网络文件系统(Network File System,简称 NFS),Androw 文件系统(AFS)、自认证文件系统(Self-certifying File System,简称 SFS),附网存储设备(Network Attached Storage Device)。

### 3.1 网络文件系统(NFS)

NFS 是使用最广泛的附网文件系统。它使异构客户透明地共享存储在远程存储服务器上的文件,而不必考虑文件的位置,异构性和可移植性是设计 NFS 的驱动力。NFS 有两个基本部分组成:安装在客户端机器上的客户端程序和安装在服务器端的服务器端程序。NFS 客户端使用 RPC 和服务器端通信,即允许一个主机调用另一个主机上的函数。

一个系统管理员通过在文件/etc/exports 中例举输出目录,在输出目录中列举希望访问共享文件的文件系统的访问权。在输出操作期间,管理员能指出一个允许访问该输出目录的主机清单和客户访问输出文件的安全要求。客户通过联系服务器上的安装进程来安装输出文件系统,并依靠客户证书检查输出清单和相应的访问权限。在客户端安装了客户端文件系统以后,它是被集成到客户端的目录树中,并且客户能像访问本地的文件系统一样访问这个安装的文件系统。NFSv4 是一个最近的版本,它与早期的版本有很多不同之处。例如,早期版是一个无状态协议,它要求使用不同的协议来处理文件安装和文件的锁操作,而 NFSv4 则是一个有状态的协议,它集成全部不同的协议为一个标准协议,并且使用强安全机制。

在安全方面,NFS 早期的版本主要依靠下面的操作系统提供访问控制和弱认证机制。在 NFSv2 中定义了多种认证功能,像 Unix 认证(用户 ID 和组 ID)、基于 Diffie-Hellman 认证,基于 Kerberos v4 的认证。尽管 RPC 机制允许多种认证机制的使用,但像基于 UID 和 GID 认证的 Unix 认证机制使用最普遍。这种机制是不安全的,因为一个攻击者很容易哄骗(或重发)一个用户的证书。

NFSv4 使用 RPCSEC_GSS 机制[7],RPCSEC_GSS 提供

RPC 请求和响应的认证、机密性和完整性。在 NFSv4 中必须实现 Kerberos v5 和 LIPKEY(一个低级的基础公钥机制)[8],Kerberos 能在一个内部网内提供强的认证机制,使 NFSv4 协议提供强的认证机制和网络安全。LIPKEY 是被设计在 Internet 上使用,要求每一个文件服务器有一个公钥证书和一个相关私钥。一个用户通过使用服务器公钥加密登录信息并发送这些信息到服务器,然后与服务器之间建立一个安全通道。用户也通过检验服务器证书来认证服务器。在认证期间,一个用来保证用户和文件服务器之间安全通信的对称密钥被建立。因此,使用 LIPKEY,一个用户能跨 Internet 安全地访问他的存储服务器上的共享文件。

传统的 NFS 只支持 POSIX 模式,而 NFSv4 包括的 ACL 支持基于 Windows NT 模式,但不支持 POSIX 模式,因为 Windows NT 模式的操作语义比 POSIX 模式更丰富。在 Windows NT 模式中一个访问控制实体有四种意义:ALLOW,DENY,AUDIT 和 ALARM。因此,NFSv4 能提供强安全和灵活的文件共享。

### 3.2 Androw 文件系统(AFS)

AFS 最初开发是为校园网家目录(home directories)提供一个可扩展带宽的文件系统[9],它能有效运行在带宽受限的校园骨干网上。AFS 的主要服务包括可扩展性、缓存和简化寻址。

AFS 进一步发展为可扩展的分布式文件系统,它使协同操作主机能跨两个局域网或广域网共享文件系统资源。AFS 运行在自治单元内部,这种自治单元代表一个被独立管理的 AFS 站点,例如,cs. CMU. edu 是一个被卡基梅隆大学计算机系占有的自治单元。一个自治单元有自己的保护域、认证服务器、文件服务器、卷单元服务器和系统管理员。一个系统管理员能发布他自己单元信息,并且通过在本地单元服务器数据库上列举必要的远程单元清单使本地单元对清单中的单元可见。AFS 单元之间通过协同操作支持无缝文件共享。

在安全方面,AFS 使用 Kerberos 实现认证(对 Kerberos v4 作了轻微的改变),为了访问存储在一个单元的文件,这用户应该在那单元中有一个帐号。Kerberos 认证服务器维持一个用户口令数据库,这些口令是使用只有服务器(和系统管理员)知道的密钥加密的。用户的口令从不在网上发送,而是被使用来加密发送到认证服务器的用户请求,认证服务器使用用户的口令检验和认证用户的请求,认证后认证服务器发送一个票据(ticket)给用户,用户使用该票据与文件服务器相互认证。相互认证通过后,用户与文件服务器之间建立一个会话密钥,并用该会话密钥加密用户与文件服务器之间传送的所有流量。值得注意的是:Kerberos 不能有效地阻止口令猜测攻击。

访问控制由用户和组组成的保护域决定的。一个组包含一个所有者、一组用户以及其它组。如果一个用户是一个组的成员,而这组又是另一个组的成员,那么这个用户继承了两个组之间的成员关系。一个用户可以建立自己的组,组名以用户名为前缀。如用户 andrew 建立组 friends,这名为 andrew:friends。支持组成员关系和组继承关系的优点是管理和删除简单,并且管理保护域容易。通过使用与每个目录相关的访问列表实现访问控制,这种访问列表列举了用户、组以及它们的访问权限。有两种访问权列表:肯定权限列表和否定权限列表,否定权限列表总是比肯定权限列表优先。在 AFS 中的访问列表(读、写、锁、执行、删除、查寻、管理和插入)比标准 UNIX 的 ACL(读、写、执行)复杂的多,并且允许用户建立自己的组和组继承关系。

### 3.3 自认证文件系统(Self-certifying File System,SFS)

大部分文件系统(包括 NFS 和 AFS)都有管理范畴的概念,并且依靠中心实体来配置、管理和达到安全的目的。因此,中心实体的出现妨碍全局文件共享和阻止容易添加新的文件服务器。AFS 单元协同操作支持跨域的文件共享,一个客户要想访问存储在另一个单元的文件,他必须在远程文件服务器上有一个帐号,并且这个用户所在有自治单元是被列在这远程文件服务器的本地数据库中,他才能访问该文件服务器中的文件。SFS 通过分散控制提供安全全局文件共享。每个用户能设置 SFS 文件服务器,任何用户(有合适的权限)从任何客户端都可以访问任何服务器。

SFS 通过引进自认证路径名(self-certifying pathnames)将密钥管理完全从文件服务器中分离出来。自认证路径名是一个包含远程服务器公钥的文件名。在文件访问期间,SFS 客户使用嵌入在路径名中的公钥验证 SFS 服务器的真实性。SFS 结构包含三个实体:SFS 文件服务器、SFS 客户、SFS 认证服务器。认证服务器中存放用户组成员关系和公钥信息。

每个 SFS 用户和服务器有一对公-私钥,SFS 文件系统的可访问目录/sfs/Location/Hostid。这可以是一个 DNS 中的一个主机名或一个 IP 地址。当用户请求存储在远程服务器上的一个文件时,客户端的认证代理初始化一个认证协议。认证协议在客户端机器为该用户产生一对公-私钥,客户端机器通过自认证路径名获得 SFS 服务器公钥,然后检查密钥 hash 是否与路径名中的 Hostid 相匹配,如果是则建立一个共享的会话密钥,并且使用公钥加密协议完成与服务器的相互认证。会话密钥被使用来在客户端和远程服务器端建立安全通道(加密和认证)。这完成后,认证代理通过安全通道向文件服务器发送一个使用用户私钥签名的请求,收到签名请求后,文件服务器传递请求到认证服务器,认证服务器检验签名并传递用户证书到文件服务器,文件服务器缓存该证书并使用它来认证用户将来的请求和执行访问控制。

### 3.4 附网存储设备

大部分分布式文件系统(包括前面讨论的文件系统)都是通过文件服务器从存储设备中取回数据,然后提交数据到用户,请求和数据都通过文件服务器转发,这样容易形成文件服务器瓶颈,影响系统的扩展性和性能。这样,附在存储网络中的存储设备不关注存储安全,而且假定安全都是由文件服务器保证。其访问流程如图 1 所示:①客户向文件服务器提出请求;②文件服务器认证用户的身份和访问权限;③文件服务器从存储服务器中取数据;④文件服务器将用户请求数据传给用户。

NASD 把存储设备直接挂到网上,而从数据路径上移去文件服务器。用户直接与存储设备交互,存储不再受文件服务器保护,存储设备负责认证和访问控制。其访问流程如图 2 所示。其过程如下:①当用户访问 NASD 中一个文件时,首先向文件服务器发一个特定对象请求,接到请求后,文件服务器认证用户并为该用户产生一个容器密钥。容器密钥由文件管理器与 NASD 共享的密钥对用户公钥证书产生的信息 MAC 得到。公钥证书中包含有对象 ID、用户对指定对象的访问权限、和容器密钥的过期时间等。②文件管理器传递容器密钥(通过一个安全通道)和相关的公钥证书到用户。③用户发送对象请求和使用容器密钥产生的请求 MAC 到

NASD。④收到用户请求后，NASD产生容器密钥检验用户请求的MAC。如果MAC检验通过，NASD准许用户按公钥证书规定的权限访问指定的对象。NASD存在问题：文件管理器成为一个单一中心失效点。

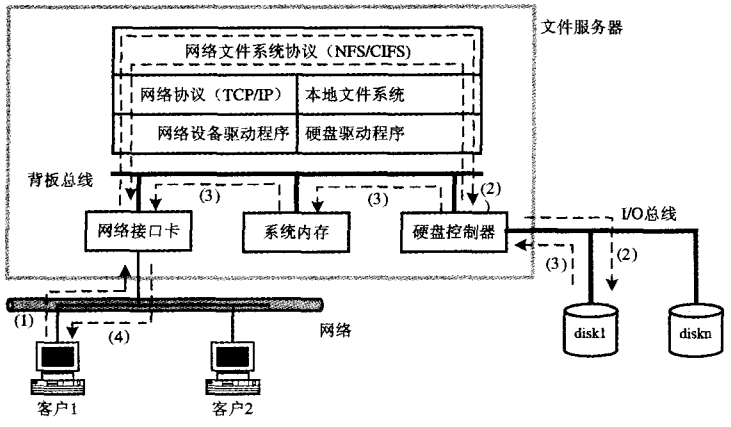


图1　分布式文件系统访问流程

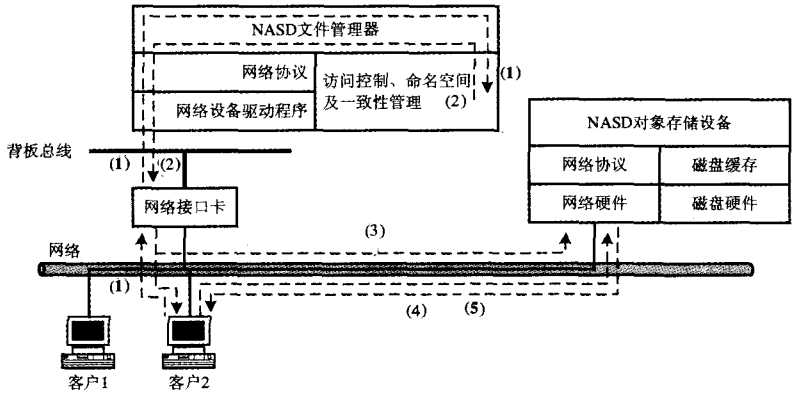


图2　NASD访问流程

## 4　加密文件夹系统

加密文件系统指在客户端加密文件，然后将加密文件以密文形式送到存储服务器（文件服务器）存放，有权限用户从服务器端读取加密文件到客户端解密。存储服务器和系统管理员都看不到文件的明文。加密文件系统包括非共享加密文件系统和共享加密文件系统两大类。

### 4.1　非共享加密文件系统

非共享加密文件系统指加密文件只能供一个用户访问，其它用户不能共享。代表性的是基于UNIX的非共享加密文件系统。

#### 4.1.1　基于UNIX的非共享加密文件系统(CFS)

CFS集成文件加密服务到文件系统内部，是运行在客户端机器用户层的一个虚拟文件系统。它主要执行加密和密钥管理功能，其它功能留给下面的文件系统。CFS典型的安装目录是/crypt。用户使用cmdir命令建立一个加密目录，这命令要求用户输入一个ASCLL短语，CFS使用该短语建立一个密钥，然后用户用cattach命令映射加密目录成为一个虚拟目录，这个虚拟目录在/crypt下可以看到。

CFS使用OFB(output feedback mode)和EBC(electronic code book)组合加密。CFS要求用户输入超过16个字符长度的短语，用这种短语产生两个密钥K1和K2。K1用来产生OFB模式使用的长伪随机位掩码(通常0.5MB)。文件是一块一块加密的，每块在加密前都要与长伪随机掩码作异或运算。然后用K2加密。在CFS中，对加密目录/crypt的访问是受UNIX保护机制的控制。在CFS中，也考虑到密钥恢复和密钥安全存储问题。

CFS加密不是完全透明，加密粒度是一个目录，用户要记住对这个目录加密的密钥。CFS不保证数据和元数据完整性。并且，组共享很麻烦，因为它不包括任何密钥分发机制。

#### 4.1.2　其它非共享加密文件系统

很多解决方案允许用户在本地磁盘上建立一个安全分区，然后透明地加密存放在这个分区（逻辑磁盘）上的数据[10,11]。密钥是使用用户口令或用户选择的短语生成的。

这样的系统有Gutmann's Secure FileSystem(GSFS，为DOS和Windows)以及Swank's SecureDrive。SecureDrive是另一个为DOS和Windows提供安全加密服务的设备驱动程序。CryptoGraphic disk driver(CGDD)在NetBSD提供加密服务，它由内核级的虚拟磁盘驱动程序组成，它访问原始盘执行块级的加密/解密。这些系统的不足：不支持多用户，是用口令产生密钥。

### 4.2　共享加密文件系统

共享加密文件系统指被加密文件可供用户共享，主要包括六种：基于UNIX的透明加密文件系统、NCryptfs、基于Windows的加密文件系统、基于智能卡的加密文件系统、SiRiUS和Plutus。

#### 4.2.1　基于UNIX的透明加密文件系统

基于UNIX的透明加密文件系统(Transparent Crypto-

graphic File System for UNIX,简称为 TCFS) 是一个内核级为用户提供加密服务的文件系统。类似 CFS,TCFS 提供端到端的加密,即在客户加密和解密文件。与 CFS 相比,TCFS 为用户提供透明的加密服务,提供数据的完整性,加密数据可在组用户之间共享。用户必须维护一个加密全部文件密钥的口令(不像 CFS 中一个目录一个口令)。

在 TCFS 中每个用户建立一个主控密钥(master-key),并用用户的登录口令加密存放在中心数据库中。这个中心数据库可以位于客户端机器上,也可位于远程的密钥服务器上。与每个文件相关的是一个文件密钥(file-key),并用主控密钥加密存放在文件头的文件密钥域中。通过 hash 每个文件的文件密钥和对应的块号为该文件中的每个块产生一个块密钥。块密钥按 CBC(cipher-block chaining)模式加密对应的块。TCFS 通过组密钥使组成员共享组加密文件。TCFS 的结构如图 3 所示,一个加密文件由加密数据和文件密钥域组成。

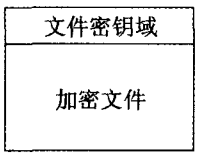


图 3 TCFS 的文件结构

4.2.2 NCryptfs

NCryptfs 提供内核级加密服务的堆栈文件系统[12],NCryptfs 的主要目标是提供透明的加密服务,并且不影响系统性能,容易移植。系统管理员安装 NCryptfs 在/mnt/ncryptfs 目录下,并且可在系统安装点吸附一个或多个授权实体,每个实体指定一个口令。在访问时,一个用户如果输入的口令与授权实体中原指定的口令相匹配时,则有相关的访问特权。

在安全方面:类似 CFS,在 NCryptfs 中,一个用户要对一个目录加密,必须在 NCryptfs 安装点建立该目录实体,并且为该实体输入一个短语,用该短语产生加密密钥。这个密钥常驻内存,并为存放在对应的吸附目录中的文件和文件名加密。

NCryptfs 的特点:1. NCryptfs 提供分布文件共享非常不方便,因为它是靠用户口令加密的;2. NCryptfs 是有效和可移植,因为它是堆栈式在内核中实现的;3. 它不保证数据和元数据的完整性,并且它的密钥管理是很原始的,它的密钥是短语,密钥恢复是留给用户。密钥仅有所有者知道,它存储在内核内存中,从不暴露其它用户。

4.2.3 基于 Windows 加密文件系统(EFS)

EFS 的目标提供透明的端到端加密服务[13],只有合法客户才能加密/解密这些文件,用户可选择不同加密算法。EFS 提供加密文件可在成员较少的用户组之间共享,EFS 有恢复丢失密钥的机制,并且能够提供数据和元数据的完整性。在安全方面:EFS 自动为每个用户建立一对公-私和从管理员配置的认证机构 CA 中获得一个公钥证书。如果没有 CA,EFS 自已对公钥签名。在 EFS 中,文件或目录用对称密钥加密。即一个用户想对文件或目录进行加密,EFS 为每个文件产生一个加密密钥(FEK),并使用该密钥对文件进行加密,加密密钥又用用户的公钥加密并同加密文件一起存放在叫做数据解密域(DDF)中。EFS 适合人数较少的小组共享文件。每个用户的私钥可存放在智能卡或集成基于软件保护存储。

一个用户无论什么时候想访问加密文件,EFS 客户端自动获得 FEK(通过使用用户的私钥解密加密的 FEK),并使用 FEK 解密/加密文件,这个过程对用户是透明的。如果远程文件存放在一个 Web 文件夹内,加密/解密是在客户端进行。如果远程文件是以文件共享方式存放,则加密/解密是在存放该文件机器上进行,然后以明文方式传到用户所在客户端。系统管理员可定义合法用户的文件恢复策略,这种策略可在本地、域和组织级进行配置。管理员通过建立多个恢复代理,每个代理有一对公-私钥,每个文件加密密钥都用该公钥加密存放在文件的数据恢复域(DRF)中。当用户丢失了私钥时,可把加密文件发送到相应的恢复代理,恢复代理解密该文件,然后发给用户。EFS 不保证数据的完整性及元数据的完整性和机密性。EFS 结构如图 4 所示。

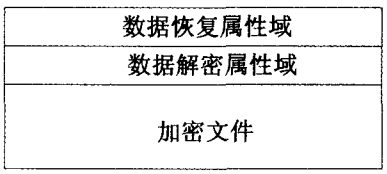


图 4 EFS 文件结构

4.2.4 基于智能卡的安全文件(Smart-Card-based, Secure File System)

通过基于智能卡的安全文件系统(SSFS)用户能使用普通的网络协议把数据安全存放在本地或远程的站点中。SSFS 允许属于一个组织或不同的组织的两个或多个组实现安全文件共享[14]。除了加密和分布式的访问控制外,SSFS 也提供密钥恢复和安全密钥存储。全部私钥(用户和组)都存放在一个智能卡上。SSFS 有三个实体:用户(数据的生产和消费者)、组服务器、文件服务器。组服务器维护每个组并被完成信任。组服务器为每个组制定组成员关系和访问控制。

在 SSFS 中,每个用户或组都有一对公-私钥。私钥是存放在个人的智能卡上。SSFS 支持基于 XML 的访问控制列表。一个加密文件的所有者可以在访问控制列表中指定允许解密该文件的用户或组。如果只同个别用户共享文件,文件的所有者使用自己的公钥加密该文件,如果同组用户共享此文件,文件的所有者用组服务器公钥加密该文件。SSFS 存在的问题:它只提供端到端的加密,但不提供元数据的机密性和数据、元数据的完整性。另外,用户要想访问一个加密文件,组服务器必须在线,所以组服务器成为一个中心失效点。并且,SSFS 要求机器支持智能卡,这是很不方便的。

SSFS 的组共享访问流程。Alice 想让她的文件与奔腾组成员共享,她用奔腾组服务器公钥加密这文件密钥,假定奔腾组成员 bob 想访问 Alice 的文件,其流程:①bob 首先向文件服务器要求这个文件;② bob 所在客户端机器从文件服务器收到该加密文件后,首先检查 XML ACL,看 bob 有没有权限,如果有则向 bob 的组服务器请求文件密钥,即把用 bob 的智能卡签名的请求和被加密的文件密钥一块发给奔腾组服务器;③组服务器检验 bob 的请求和权限,如果行则用奔腾组服务器私钥解密文件密钥,然后用 bob 的公钥重新加密文件密钥后发给 bob 所在的客户端;④收到被加密的文件密钥后,bob 可用自己的智能卡解密。

4.2.5 SiRiUS

SiRiUS 是一个用户级的文件系统[15],它被设计位于不

安全的网络层或像 NFS、CIFS、CceanStore 和 Yahoo! 公文包等点对点文件系统之上。它在人数比较少的小组之内为文件级的共享提供读-写加密访问控制。它的主要目标是在不对文件服务器作任何改变的情况下，改进文件系统的安全性能。

在安全方面，SiRiUS 除保证数据的机密性和完整性外，也保证元数据松散的完整性。SiRiUS 将存储文件服务器中的所有文件分成两个部分：包含加密数据的数据文件部分(d-file)和包含访问控制信息的元数据文件部分(md-file)。将数据文件与元数据文件分开存放是便于 SiRiUS 运行在任何存储服务器上，只要 SiRiUS 客户端按存储服务器的语义与存储服务器交互，它就能为用户数据提供安全保证。所有 SiRiUS 用户有一对公-私钥，与文件相关的有两个密钥：加密密钥 FEK 和文件签名密钥 FSK，只有 FEK 的用户只有读文件权，同时有 FEK 和 FSK 的用户才有文件的读/写权。FEK 和 FSK 用用户的公钥用加密。

如果一个文件准许在多个用户之间共享，文件的所有者将在 md-file 中为每个用户建立一个 entry，如果某用户只有读权限，该 entry 中只用该用户公钥加密的 FEK，如果某一用户有读/写权限，该 entry 中包含有用该用户公钥加密的 FEK 和 FSK。

4.2.6 Plutus

Plutus 能够保证端到端的数据和元数据机密性，以及完整性[16]。密钥管理和密钥分发在客户端进行，存储服务器不涉及任何安全操作。它提供注销机制，读/写权限与 SiRiUS 相似。

在安全方面，Plutus 把有同样共享属性的文件分在一个组。与组相关的有一个锁盒密钥(lockbox key)，与文件相关的有一个文件块密钥(file-block key)，文件块密钥是用来加密该文件中的每一个块，锁盒密钥用来加密一个组中的每一个文件块密钥。如果一个用户想与其他用户共享他的文件，他必先建立一个文件组(成员清单，及第 $i$ 个成员一个允许位)和一个锁盒密钥，然后分发到组的所有成员。类似 SiRiUS，有签名密钥的用户才有写权限。存储服务器通过写标记来认证一个有写权限用户，写标记是由数据的所有者分发给该文件组的所有写权限用户，这可阻止了非法用户对存储服务器中的写入。在安全共享文件系统中，注销代价是昂贵的，因为一个用户被注销，要重新分发密钥，重新加密该用户能访问的所有文件。Plutus 采用赖惰注销法，即文件推迟加密直到文件被更新。

## 5 基于存储的入侵检测

主要包括两种：自安全存储系统(Self-Securing Storage System)[17]和卡基梅隆大学的基于存储的入侵检测系统(CMU Storage-based IDS)[18]两类。

### 5.1 自安全存储系统

卡基梅隆大学开发的自安全文件系统在指定时间期限内保持数据的旧版本。这样，系统管理员能根据旧版本数据的历史执行入侵诊疗和恢复数据。自安全存储系统运行在存储服务器边，独立于任何客户端的软件、操作系统或文件系统，它也可运行在磁盘的固件中。当存储服务器接到一个写操作时，它保持两个版本的数据：旧版本和新版本，旧版本直到执行清除命令或者旧版本的时间窗口已过期才被清除，否则一直保存着。这可阻止意外事件或恶意删除造成的破坏。

### 5.2 基于存储的入侵检测系统(SIDS)

SIDS 是采用自安全存储系统类似的原理开发，SIDS 嵌入存储设备内部并分析数据的访问模式和数据的修改特点，从而寻找攻击行为。SIDS 与基于主机的 IDS 相比，直接运行在存储服务器或磁盘固件中的 SIDS 的主要优点是：当一个入侵者攻入主机并使主机的 IDS 失效，但对 SIDS 没有影响。另一方面，SIDS 与主机 IDS 相比有一些受限的地方：它对文件系统有一定要求。

CMU SIDS 是一个基于规则的入侵检测系统，使用一套规则检测可疑的数据修改。SIDS 主要支持四类可疑活动检测：①不希望改变重要系统文件和二进制文件；②没有追加修改的模式改变和 inode 时间反向。这些模式表明攻击者设法修改系统日志文件，又想企图抹去恶意踪迹；③被特别禁止的内容变化到一个关键文件中(例如，不合法内容插入到文件/etc/passwd. 中)；④特定文件名的出现(例如，被隐藏的 dot 文件，已知的病毒或攻击工具等)。

### 5.3 其它基于存储的入侵检测系统

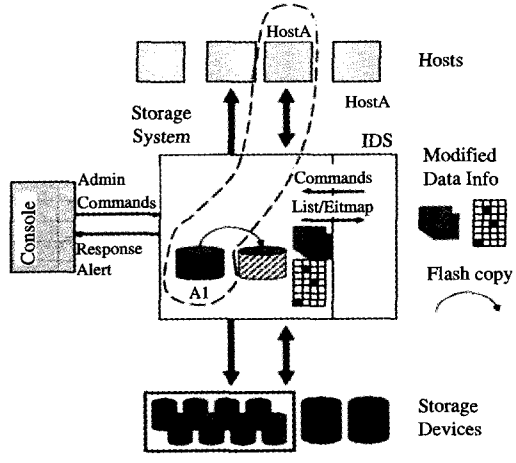


图 5 SFSIC 的工作原理图

目前，已有几个其它基于存储的入侵检测系统帮助管理员监视他们的文件系统，寻找攻击行为。其代表性的是基于存储的文件系统完整检查器(Storage-based File System Integrity Checker，简称 SFSIC)[19]。该系统使用时间和空间上的有效即时点复制以及执行文件系统的完整性检查来检测入侵，即当有用户对存储系统中的某一盘进行写操作时，SIDS 即时建立该盘的逻辑快照盘。并且，SIDS 为每个盘建立一个位图，位图的每一位对应该盘中的每一块，修改块与没有被修改块的用不同的位表示，SIDS 通过跟踪修改的文件块，使系统的扫描更有效，当入侵出现时用未有破坏的副本恢复损坏了的数据。其工作原理如图 5 所示。

## 6 比较和讨论

本节将从实体认证、访问控制、信息安全等方面对上述各种安全存储系统进行比较。

### 6.1 实体认证

在 NFS、AFS、SFS、TCFS 中，客户认证由可信任的中心服务器来完成。在 NASD 中，文件管理器认证用户并给用户一容器标记，NASD 存储设备通过用户所持的标记认证用户。SSFS 依靠中心组服务器认证用户。SFS 提供两级认证：第一级认证，SFS 客户与 SFS 文件服务器相互认证；第二级认证，SFS 认证服务器 AS 使用第一级建立的安全通道认证用户。

EFS和CFS依靠下面的系统认证机制解决认证问题。在Plutus中,存储服务器通过写标记认证用户的写权限,而SiRiUS不提供任何实体认证机制。

### 6.2 访问控制

SSFS通过中心组服务器(AS)执行访问控制。NASD中,中心文件管理器给一个包含对特定对象访问权限的证书给用户,存储设备简单地按这些规定权限执行访问控制。SiRiUS访问控制完全按密钥的占有决定:一个用户只有FEK时,只有读权限,如果一个用户同时有FEK和FSK时,则同时具有读和写的权限,在SiRiUS中文件服务器不执行访问控制。Plutus除了具有SiRius访问控制外,文件的所有者给每个有写权限的用户一个写标记,文件服务器验证这些标记,验证通过后才能执行写。在AFS、NFSv4和SFS中,文件服务器都是通过访问控制列表对用户进行访问控制。在CFS、TCFS和EFS中,文件服务器不提供访问控制功能。

### 6.3 信息安全性

AFS、NFSv4、Plutus、NASD和SFS数据是通过安全通道在客户端和文件服务器(存储设备)之间传送,并且,用户在客户端要进行完整性检查,所以信息是安全的。CFS、TCFS、EFS、SiRiUS和SSFS除对文件进行加密外,而不对信息存放的安全性提供保证。

### 6.4 端到端的数据和元数据的机密性和完整性

除AFS、NFSv4、NASD和SFS外,CFS、TCFS、EFS、SiRiUS、Plutus和SSFS都支持端到端的加密,AFS、NFSv4、SFS和NASD在客户机器和服务器之间加密网络流,但不提供数据、元数据的机密性和完整性。CFS支持数据和元数据的机密性,但不支持数据和元数据的完整性。TCFS支持数据的机密性和完整性,但不支持元数据的机密性和完整性。EFS和SSFS只支持数据的机密性,不支持元数据机密性,以及数据和元数据的完整性。SiRiUS和Plutus同时支持数据和元数据的机密性和完整性。

### 6.5 端到端的密钥管理

通常用分组减少密钥管理工作的负担,因为多个用户共享加密文件时,他们之间也必须共享加密密钥。AFS、NASv4、NASD和SFS不适合分组机制,也不存在密钥分发问题。TCFS提供阈值组共享,使用加密的密钥在一个组成员之间分发。EFS和SiRiUS不分组,加密文件的所有者用自己的公钥加密文件密钥。Plutus把具有同样共享属性(同样的所有者、分组访问权限)的所有文件分在同一个文件组中,一个有权限访问的用户要想访问加密文件,他必须与密钥的所有者联系,才能获得文件密钥。SSFS将同一个单位或企业的用户分在一个组中,文件的所有者使用用户公钥或组公钥加密文件密钥。

密钥恢复是支持加密和签名存储系统应该提供的另一个基本特性,密钥恢复机制对恢复丢失密钥是非常有用的。AFS、NFSv4、NASD和SFS不应用密钥恢复机制,TCFS、Plutus和SiRiUS没有密钥恢复机制,CFS、EFS和SSFS有相应的密钥恢复机制。

### 6.6 用户删除(撤回)

当把一个用户从一个组中删除掉,阻止这个用户对该组共享文件访问也是很重要的。AFS、NFSv4使用访问控制列表(ACLs)控制删除用户对共享文件的访问,CFS没有相应的控制措施,TCFS、EFS、SiRiUS和SSFS使用新密钥立即重新加密共享文件,Plutus使用懒惰重新加密的方法,NASD使用过期和访问控制版本处理用户删除。SFS通过文件服务器来控制用户删除。

**结论** 本文通过对已存在的安全文件系统的安全机制作了全面的分析,并且列举了安全文件系统提供的基本存储安全服务。这对存储系统安全的研究和实际安全系统的设计提供有用的参考。

## 参考文献


1 O'Hare M, Orsini R, Giuliano F, et al. Winick, SecureParser White Paper. Security First Corp. May 2005

2 Johnson R. Secure Storage Using SecureParser. Unisys, June 2005

3 Shepler S, Callaghan B, et al. Noveck. NFS version 4 protocol. RFC 3530, April 2003

4 Kher V, Kim Y. Securing Distributed Storage: Challenges, Techniques, and Systems. StorageSS'05 November 11, 2005

5 Banikazemi M. Storage-based File System Integrity Checker. StorageSS'05 November 11, 2005

6 Factor M, Nagle D, Naor D. The OSD Security Protocol. SISW05, 2005

7 Eisler M, Chiu A, Ling L. RPCSEC GSS protocol specification. RFC 2203, September 1997

8 Eisler M. LIPKEY —a low infrastructure public key mechanism using SPKM. RFC 2847, June 2000

9 Howard J. An overview of the andrew file system. In: Proceedings of the USENIX Winter Technical Conference, Dallas, TX, February 1998

10 Dowdeswell R C, Ioannidis J. The CryptoGraphic Disk Driver. In: Proceedings of the FREENIX Track: 2003 USENIX Annual Technical Conference, June 2003

11 Swank E. SecureDrive. http://www.stack.nl/ ~galactus/remailers/securedrive.html

12 Charles E Z, Wright P, Martino M C. Ncryptfs: A secure and convenient cryptographic file system. In: USENIX Annual Technical Conference, June 2003

13 Corporation M. Encrypting File System for Windows 2000. White Paper, July 1999

14 Hughes C F J. Architecture of the secure file system. In: Proceedings of the Eighteenth IEEE Symposium on Mass Storage Systems, April 2001

15 Goh E, Shacham H, Modadugu N, et al. SiRiUS: Securing Remote Untrusted Storage. In: NDSS, 2003. 131~145

16 Kallahalla M, Riedel E, Swaminathan R, et al. Plutus - scalable secure file sharing on untrusted storage. In: USENIX File and Storage Technologies (FAST), 2003

17 Strunk J D, Goodson G R, et al. Self-Securing storage: Protecting data in compromised systems. In: OSDI, 2000

18 Pennington A, Strunk J, Griffin J, et al. Storage-based intrusion detection: Watching storage activity for suspicious behavior. In: 12th USENIX Security Symposium, Washington, D C, August 2003

19 Banikazemi M. Storagebased File System Integrity Checker. StorageSS'05, Fairfax, Virginia, USA, 2005