Computer Science and Application 计算机科学与应用, 2013, 3, 1-5
http://dx.doi.org/10.12677/csa.2013.38A003 Published Online November 2013 (http://www.hanspub.org/journal/csa.html)

Design and Implementation of an Attribute Based Access Control System for Cloud Storage*

Pengjian Sun[1,2], Siyue Zhang[2,3], Chuanyi Liu[2,3], Cong Wang[2,3]

[1]School of Computer Science, Beijing University of Posts and Telecommunication, Beijing
[2]Key Laboratory of Trustworthy Distributed Computing and Service (BUPT), Beijing University of Posts and Telecommunication, Beijing
[3]School of Software, Beijing University of Posts and Telecommunications, Beijing
Email: pengjiansun@163.com, wangc@bupt.edu.cn

Received: Sep. 29th, 2013; revised: Oct. 25th, 2013; accepted: Nov. 8th, 2013

Abstract: To satisfy the security demand of cloud storage application, this paper designed and implemented an attribute based access control mechanism named EncFS, which is suitable for cloud storage system. EncFS is a file system of user space, which is based on fuse and access control strategy for Linux operating system. It uses identity information as attributes for access control. Therefore, this system can simplify password management and storage, realize the fine-grained access control and solve the problem of dynamic expansion of large-scale users.

Keywords: Access Control; Attribute Based; Cloud Storage

基于属性的云存储访问控制系统的设计与实现*

孙鹏建[1,2]，张思悦[2,3]，刘川意[2,3]，王　枞[2,3]

[1]北京邮电大学计算机学院，北京
[2]北京邮电大学可信分布式计算与服务教育部重点实验室，北京
[3]北京邮电大学软件学院，北京
Email: pengjiansun@163.com, wangc@bupt.edu.cn

收稿日期：2013 年 9 月 29 日；修回日期：2013 年 10 月 25 日；录用日期：2013 年 11 月 8 日

摘　要：本文针对云存储应用的安全访问需求，以及目前适用于云存储环境的基于属性的方案，设计和实现了一种基于属性的云存储系统访问控制机制：EncFS。EncFS 是基于 fuse 的用户态文件系统，权限鉴别建立在 Linux 系统对用户的划分基础上，将用户的身份信息作为属性进行访问控制，不需要输入口令，从而简化了对身份和口令的管理和存储，实现了云存储下的细粒度访问控制，解决了大规模用户动态扩展问题。

关键词：访问控制；基于属性；云存储

1. 引言

在云存储服务模式[1]中，数据存储由不可信的服务器控制，处于用户控制之外，访问控制的目标是通过对数据进行控制，确保只有被授权的用户才能访问数据，从而保证云存储中数据的保密性。

主流的访问控制模型[2]有三种：自主访问控控制(DAC)、强制访问控制(MAC)、基于角色的访问控制(RBAC)。但这三种模型都无法满足云存储的应用需求。主要表现为：1) DAC 存在大规模用户动态扩展问题，即随着用户和资源数量的增长，DAC 中的 ACL 规模急剧增加，难以管理和维护；2) MAC 模式灵活性不够；3) RBAC[3]难以实现细粒度访问控制，因为若要进行细粒度的访问控制，必须对用户进行精确区分，从而使 RBAC 需要定义大量的用户角色，这给角

*资助信息：本研究得到国家高技术研究发展计划 863 项目(2012AA012600, 2012AA012606)的资助。

色的分配和管理带来困难。

针对上述问题，本文提出了一种适用于云存储应用需求的基于属性的访问控制模型(attribute based access control, ABAC)，该模型将相关实体的属性作为授权的基础进行访问决策，通过用户属性和数据属性的匹配关系确定解密能力，不仅能够简化云存储环境下的权限管理，还能解决传统访问控制模型中存在的大规模用户动态扩展问题、灵活性问题和细粒度访问控制问题。

2. 相关工作

访问控制技术的目标是通过对用户访问资源的活动进行有效监控，使合法的用户在合法的时间内获得有效的访问控制权限，防止非授权用户访问系统资源，从而保证信息资源不被非法使用和访问。

传统主流的访问控制模型有三种：自主访问控制、强制访问控制、基于角色访问控制。文献[2]是早期经常被使用的安全策略模型，自主访问控制模型，它通过访问控制列表 ACL 实现权限管理，简单方便，可以实现细粒度访问控制，其不足之处是随着用户和资源数量的增长，DAC 中的 ACL 规模急剧增加，难以管理和维护，而云存储是多租户平台环境，所以 DAC 不能满足云存储的应用需求。文献[2]提到的强制访问控制模型，虽然不存在大规模用户动态授权扩展问题，但是其灵活性不够，云存储服务需要提供给形形色色的用户使用，所以这种模型也不适合云存储应用需求。基于角色的访问控制模型[3]可以解决云存储环境下的大规模用户动态授权问题，但是却难以实现细粒度访问控制，云存储环境下需要对用户进行精确区分，以便准确地满足用户的服务需求，若要用 RBAC 进行细粒度的访问控制，需要定义大量的用户角色，这会给角色的分配和管理带来困难。

随着云计算等新型计算模式的出现和普及，已有学者开始针对云计算平台访问控制进行研究。文献[4]基于 RBAC 模型采用分层的方法针对云平台提出了一种访问控制方法，解决了云平台访问角色命名冲突等问题，但缺少对实现方法的探讨。文献[5]提出了一个引入部门的多租户访问控制模型，但无法实现租户层次的继承，并且将管理职责全部交由租户管理员导致其权限过大，无法对其控制。云存储系统自身的结构特点，决定了其需要实现大规模动态用户扩展和细粒度访问控制，文献[6]提出的基于 CP-ABE 算法的密文属性访问控制机制，其主要思想是基于密文属性进行加密，虽然可以满足云存储环境对于访问控制系统的需求，但是其数据所有者直接进行分布式分发密钥，密钥发布与管理相当复杂。借鉴基于密文属性的访问控制机制，本文将 Linux 系统的用户的组属性作为属性信息对数据进行加密，既解决了大规模用户动态扩展问题、细粒度访问控制问题，同时避免了复杂的密钥发布与管理，大大简化了云存储环境下的权限管理。

3. 基于属性的访问控制技术

基于属性的访问控制把与访问控制相关的时间、实体空间位置、实体行为、访问历史等信息当作主体、客体、权限和环境的属性来统一建模，通过定义属性之间的关系描述复杂的授权和访问控制约束，能够灵活地表达细粒度[5]、复杂的访问授权和访问控制策略，从而增强访问控制系统的灵活性和可扩展性。

3.1. 基本框架

ABAC 中的基本元素包括请求者，被访问资源，访问方法和条件，这些元素统一使用属性来描述，而且各个元素所关联的属性可以根据系统需要定义。策略中的访问者是通过访问者属性来描述；同样，被访问资源，方法也是通过资源和方法的属性来描述；而条件用环境属性来描述。环境属性[7]，通常是一类不属于主体，资源和方法的动态属性，如访问时间，历史信息等。条件有时也会用来描述不同类型属主具有的属性之间的关系。ABAC 框架[8]示意图如图 1 所示。

策略执行点(PEP)接收原始访问请求(NAR)，然后根据原始访问请求，利用不同的属性权威(AA)中存储的属性信息构建一个基于属性的访问请求(AAR)，基于属性的访问请求描述了请求者、资源、方法和环境属性，策略执行点将基于属性的访问请求传递给策略判定点(PDP)，策略判定点根据从策略管理点(PAP)处获取的策略对基于属性的访问请求进行判定，并将判定结果传给策略执行点，策略执行点执行此访问判定结果。

3.2. 访问流程

ABAC 中的访问流程如下：

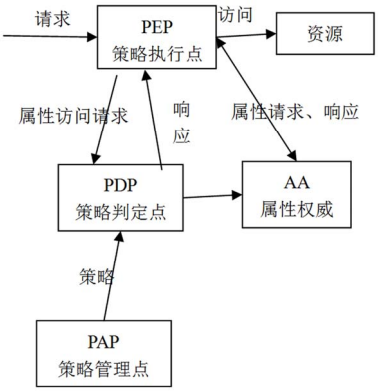

Figure 1. Schematic: ABAC framework
图 1. 基于属性访问控制框架示意图

首先，用户申请访问，在身份验证阶段通过向系统提交公钥证书证明自己的身份。被申请者对该证书进行验证，包括签名的验证，以及是否过期或被撤销；

如果证书合法，用户通过身份验证，则从中提取属性信息；否则，拒绝该用户的非法访问，该次申请随即失效并终止；

系统将得到的用户属性信息进行授权策略验证，匹配该属性与数据属性，对权限对象进行约束性检查，得出用户所有的合法访问权限；

如需进行授权委托，则由被申请者进行相关的委托策略；否则对比用户的合法权限集与资源要求权限集，在前者包含后者的情况下，创建会话对象，允许其访问资源，否则拒绝用户的访问，这次申请随机失效并终止；

访问完成后，被申请者关闭会话，释放系统资源。

4. 基于属性的访问控制技术的实现

4.1. 软件功能

EncFS 是一个运行在 Linux 上的文件管理软件。它把 Linux 系统对用户的分组情况作为划分不同属性的依据，因此用户可能拥有的属性分为三类：数据属主，同组用户，其它用户。

软件由 Client 端和 Server 端两部分组成，启动 Server 端后可在 Client 端将指定文件夹挂载，挂载完成后，系统可以对不同用户对挂载文件夹里的文件操作进行管理。这里实现的是：数据属主和同组用户可以读取明文，其他用户只可以读取密文。

软件的功能如下：

1) 当指定文件夹被挂载时，Server 端和 Client 端会将挂载文件夹中的文件遍历一次。

2) 数据属主和同组用户可以对挂载文件夹中的文件进行任何操作，如遍历子文件夹显示所有内容、读取文件、修改文件等。数据属主和同组用户在读取文件时可以读取明文。

3) 其他用户对挂载文件夹中的文件进行访问时，只能读取到密文。

4.2. 实现原理

4.2.1. 系统结构

EncFS 是一个在 FUSE 系统的基础上编写的文件系统它对文件操作命令的处理都基于 FUSE 文件系统的操作。

FUSE 全称为 Filesystem in Userspace 用户空间的文件系统。FUSE[9]是为开发用户空间的文件系统提供的一个框架，包括一些接口和一个内核模块。通过调用 FUSE 提供的接口可以创建一个守护进程，这个进程负责管理文件系统。FUSE 的作用就是将文件访问的指令传递给守护进程，然后再将结果返回给文件访问程序。

FUSE 的作用可以这样说明：在未使用 FUSE 的文件系统，如 Ext4 文件系统中，如图 2 所示，当用户输入 ls-l/home/命令后，系统会调用 Ext4 文件系统内的相关函数对文件进行处理并返回结构；而在基于 FUSE 系统所编写的用户态文件系统 EncFS 中，系统用户在该文件系统(/tmp/fuse 为 EncFS 的挂载点)内所执行的 ls-l/tmp/fuse 命令通过 FUSE 最终会调用到 EncFS 里所写的回调函数，如图 3 所示。

EncFS 是在 FUSE 系统上实现的，也就是说 EncFS 相当于系统中的守护进程。在 EncFS 系统中定义了很多回调函数，系统通过执行这些回调函数来实现对文件的操作。

4.2.2. 实现过程

在 FUSE 的基础上，EncFS 系统分为 Client 端和 Server 端两部分。系统运行的流程图 4 所示。

1) 当用户对文件进行操作时，会发出一系列文件操作指令，一般情况下这些操作指令会被发送给操作系统并执行。当被操作的文件在 EncFS 系统的挂载目录下时，这些指令会被 Client 端拦截下来并发送给 Server 端，这些指令在发送时会按照一定的指令格式

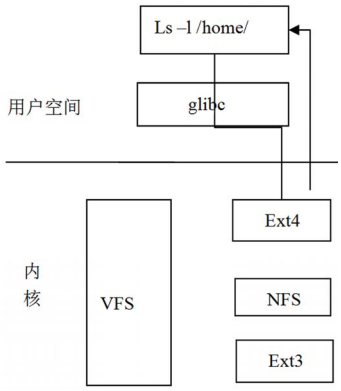

Figure 2. File operations process of EXT4
图 2. EXT4 系统的文件操作流程

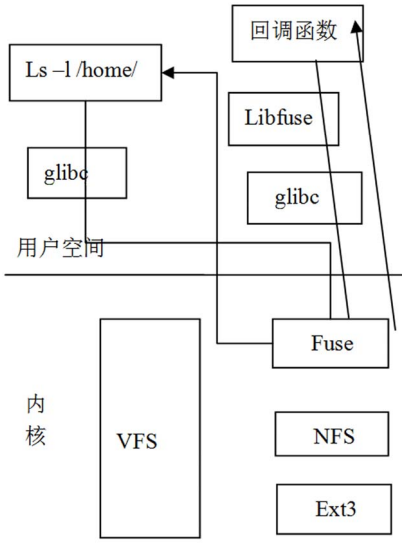

Figure 3. File operations process of ENCFS
图 3. EncFS 系统的文件操作流程

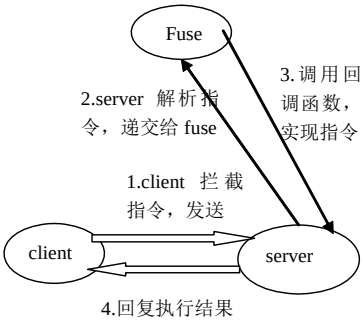

Figure 4. Implementation of ENCFS
图 4. EncFS 的实现过程

构造成指令数据包传输。

2) Server 端接收数据包，并按照约定好的数据包格式进行解析，从数据包中提取出鉴权过程和指令实现过程中所需的信息，如用户信息(包括用户名、所在组信息等)、指令信息(指令类型、指令内容等)，然后将它们传递给 FUSE 系统进行下一步操作。

3) FUSE 系统在收到指令数据包以后，会根据指令内容调用 EncFs 系统中的回调函数，对用户的操作权限进行判定，如果判定通过，则调用相应回调函数实现指令内容。

4) Server 端将权限鉴别结果发送给 Client 端，如果权限鉴别成功，则将操作结果一并返回。

4.2.3. 权限鉴别过程

EncFS 系统将用户的身份信息作为看做用户被分配的属性，以此作为权限鉴别的依据，这是以 Linux 系统对用户的划分为基础实现的。

由于 EncFS 系统对用户权限的鉴别[10]不依据用户名和口令，因此系统不用为每个用户维护身份－口令信息，需要存储的只有加密文件时所使用密钥。当用户鉴权成功后，系统会将解密密文所需要使用的密钥分发给用户。

EncFS 的鉴权过程如图 5 所示。

在系统和部署成功以后，用户将需要共享的文件目录挂载到 EncFs 系统中。在挂载过程中，系统会对目录结构下的所有文件加密，加密文件所使用的密钥存放在一个由系统管理的锁结构(lock)中。在处理用户对文件的操作指令过程中，Fuse 系统调用 EncFS 中定义的相关回调函数对用户的身份进行鉴别。这一过程依据 Linux 系统中存放用户信息的结构。这些用户信息包括用户名、创建时间、所属用户组、登录口令、创建者等。EncFS 中定义的一个回调函数 do_getattr()会读取用户信息并解析，从中获得用户的属性信息。EncFS 中定义的另一个回调函数 do_access()将得到的用户属性信息与被操作文件所对应的属性－权限规则进行比较，如果比较结果相符，则鉴权成功，用户获得进行所要求的操作的权限，FUSE 系统继续调用相关回调函数实现操作；如果比较结果不相符，则鉴权失败，用户所要求的操作被拒绝。例如，对于读操作，相匹配的属性信息为数据属主或数据属主同组用户。当发出读操作指令的用户属性与这一规则匹配时，系统从锁结构(lock)中提取出加密文件时说使用的密钥，解密密文后将明文返回 Client 端，用户可以阅读文件明文；当与用户属性与这一规则不匹配时，无法获取密钥，用户只能读取密文。

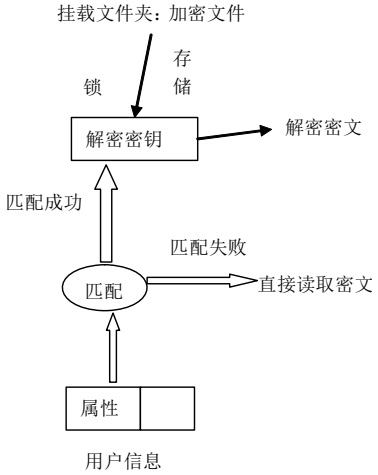

Figure 5. Permission authentication process of ENCFS
图 5. EncFS 的权限鉴别过程

4.3. EncFS 在云存储系统上的应用

EncFS 在一个基于 eucalyptus 的 YUN 基础设施存储平台上通过了测试，可以正确地实现基于属性的密文访问控制。这个云存储系统由一个控制节点和若干个存储节点构成。存储节点为云存储系统提供存储空间，在控制节点上运行着若干虚拟机镜像，整个系统的存储空间通过这些镜像来划分，这些镜像运行所需要的计算空间和存储空间都是由存储节点提供的。用户通过访问控制节点上运行的虚拟机来使用云存储系统提供的存储空间。

将 EncFS 系统中的 Server 端部署到控制节点的虚拟机中，用来对云存储系统提供的存储空间进行管理。当用户 A 要将数据上传到云存储系统中时，它运行本地机器上 EncFS 系统中的 Client 端，通过设定系统连接用的 IP 地址和端口号与控制节点上的虚拟机里运行的 Server 端建立连接，并将要上传的数据所在的文件夹挂载到 EncFS 系统中，其他用户就可以通过访问控制节点内的虚拟机来访问这些数据。当用户 B 想访问这些数据时，它首先访问控制节点内的虚拟机镜像，通过虚拟机对这些文件进行操作，而用户 B 对挂在文件夹中的受保护文件进行访问时，它的所用操作都在 EncFS 的管理下。云存储系统可以通过管理各用户访问虚拟机时所使用的账户信息对它们的权限进行管理。

5. 结论

传统的访问控制模型无法满足云存储系统的应用需求，ABAC 是非常理想的访问控制模型，为此本文在介绍改进的访问控制技术——基于属性的访问控制模型的基础上，研究了一种 ABAC 技术的实现 EncFS，分析了它的实现原理、优势，并对它在云存储系统上的应用做了详细的说明。

近年来，云存储的安全技术不断发展，本文实现了一种较为可行的云存储访问控制方案，但是还有很多不足之处，尚有待改进。相信在不久的将来云存储会给我们的生产生活带来巨大的变化。

参考文献 (References)

[1] 俞能海, 郝卓, 徐甲甲, 张卫明, 张驰 (2013) 云安全研究进展综述. *电子学报*, **2**, 371-381.

[2] 李凤华, 苏锦, 史国振, 马建峰 (2012) 访问控制模型研究进展及发展趋势. *电子学报*, **4**, 805-813.

[3] 常彦德 (2011) 基于角色的访问控制技术研究进展. *计算机与现代化*, **12**, 5-8.

[4] 张斌, 张宇 (2012) 基于属性和角色的访问控制模型. *计算机工程与设计*, **10**, 3807-3811.

[5] 盖新貌, 沈昌祥, 刘毅, 周明 (2011) 基于属性访问控制的 CSP 模型. *小型微型计算机系统*, **11**, 2217-2222.

[6] 熊智, 王平, 徐江燕, 蔡伟鸿 (2013) 一种基于属性的企业云存储访问控制方案. *计算机应用研究*, **2**, 513-517.

[7] Wang, Q., Wang, C. and Ren, K. (2011) Enabling public auditability and data dynamics for storage security in cloud computing. *IEEE Transactions on Parallel and Distributed Systems*, **22**, 847-859.

[8] Echeverria, V., Liebrock, L.M. and Shin, D. (2010) Permission management system: Permission as a service in cloud computing. 2010 *IEEE* 34*th Annual Computer Software and Applications Conference Workshops* (*COMPSACW*), 19-23 July 2010, 371-375.

[9] Moses, T. (2012) eXtensible access control markup language (XACML) version 2.0.

[10] Vipul, G., Omkant P. and Amit, S. (2006) Attribute-based encryption for fine grained access control of encrypted data. *Proceedings of the* 13*th ACM Conference on Computer and Communications Security*, 89-98.