软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2017,28(9):2334−2353 [doi: 10.13328/j.cnki.jos.005182]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

对象云存储中分类分级数据的访问控制方法*

杨腾飞[1,2], 申培松[1,2], 田 雪[1,2], 冯荣权[3]

[1](信息安全国家重点实验室(中国科学院 信息工程研究所),北京 100093)

[2](中国科学院大学 网络空间安全学院,北京 100093)

[3](北京大学 数学科学院,北京 100871)

通讯作者: 杨腾飞, E-mail: yangtengfei@iie.ac.cn

摘 要: 随着云计算技术的广泛应用,云存储中数据的安全性、易管理性面临着新的挑战.对象云存储系统是一种数据存储云计算体系结构,通常用来存储具有分类分级特点的非结构化数据.在云服务不可信的前提下,如何实现对云存储中大量具有分类分级特点资源的细粒度访问控制机制,保障云存储中数据不被非法访问,是云计算技术中亟需解决的问题.对近些年来国内外学者的成果进行研究发现,现有的方案并不能有效地应对这种问题.利用强制访问控制、属性基加密、对象存储各自的优势,并结合分类分级的属性特点,提出了基于安全标记对象存储访问控制模型.给出了 CGAC 算法及其安全证明,将分类分级特点的属性层级支配关系嵌入 ABE 机制中,生成固定长度的密文.该算法不仅访问控制策略灵活,具有层次化授权结构,还可以友好地与对象存储元数据管理机制结合.通过理论效率分析和实验系统实现,验证了所提出方案的计算、通信开销都相对较小,具有很高的实际意义.

关键词: 对象存储;云计算;数据安全;访问控制系统;分类分级数据;属性加密;安全标记

中图法分类号: TP316

中文引用格式: 杨腾飞,申培松,田雪,冯荣权.对象云存储中分类分级数据的访问控制方法.软件学报,2017,28(9):2334−2353. http://www.jos.org.cn/1000-9825/5182.htm

英文引用格式: Yang TF, Shen PS, Tian X, Feng RQ. Access control mechanism for classified and graded object storage in cloud computing. Ruan Jian Xue Bao/Journal of Software, 2017,28(9):2334−2353 (in Chinese). http://www.jos.org.cn/1000-9825/5182.htm

Access Control Mechanism for Classified and Graded Object Storage in Cloud Computing

YANG Teng-Fei[1,2], SHEN Pei-Song[1,2], TIAN Xue[1,2], FENG Rong-Quan[3]

[1](School of Cyber Security, State Key Laboratory of Information Security (Institute of Information Engineering, The Chinese Academy of Sciences), Beijing 100093, China)

[2](University of Chinese Academy of Sciences, Beijing 100093, China)

[3](School of Mathematical Sciences, Peking University, Beijing 100871, China)

Abstract: With the popularity of cloud computing, the security and manageability of cloud data faces new challenges. Object-based storage cluster is a cloud computing architecture, which is usually used to store classified and graded unstructured data. Under the premise of untrusted cloud service, how to achieve practicable fine-grained access control mechanism of massive classified and graded data while protecting data from unauthorized access, is an urgent issue to be handled. The proposed methods in recent years offer no effective ways to solve this new problem. By taking full advantages of mandatory access control method, attribute-based encryption and object storage

* 基金项目: 中国科学院战略性先导科技专项(XDA06040601); 国家电网公司科技项目(XXB17201400056); 新疆维吾尔自治区科技支撑计划(201230121); 国家自然科学基金(61370187)

Foundation item: Strategic Priority Research Program of Chinese Academy of Sciences (XDA06040601); Science and Technology Projects of State Grid Corporation of China (XXB17201400056); National Natural Science Foundation of China (61370187)

收稿时间: 2016-07-10; 修改时间: 2016-09-04; 采用时间: 2016-11-10; jos 在线出版时间: 2017-02-20

CNKI 网络优先出版: 2017-02-22 10:47:32, http://www.cnki.net/kcms/detail/11.2560.TP.20170222.1047.002.html

technology, and by combining with the characters of classified and graded data, this paper proposes a hierarchical secure label-based access control model in object cloud. Similarly, the core algorithm in this model, which is called CGAC and provably secure, provides a method to embed the hierarchical feature of classified and graded attributes into ABE mechanism, and get constant-size ciphertext. This algorithm not only has flexible access policy and hierarchical authorization structure, but also combines the benefits of metadata management of object storage. Finally, through the theoretical analysis and experimental system implementation, the paper verifies that the model's computation cost in encryption and decryption is acceptable, confirming the proposed method has high practical significance.

Key words: object storage; cloud computing; data security; access control model; classified and graded data; attribute-based encryption; secure label

1 引 言

近年来,云计算技术获得了长足的发展,云存储技术是云计算中的重要领域,云存储提供商集合集群和分布式文件系统等技术将数据存储在虚拟化的存储池中,对外提供在线存储模式的服务,具有海量级存储、动态扩展、持久性高等优点.目前,许多云计算厂商提供云存储的商业服务,如 AWS 的 S3、GoogleDrive、HDFS、百度云等.

对象存储是云存储数据的一种重要存储模式[1].与传统存储模式以块设备记录数据块位置的方式不同,对象存储基本单元是对象文件,由存储数据和元数据组成,提供了具有高性能、高扩展性、高可靠性、跨平台以及数据安全共享的存储体系结构.对象存储由于存在单独的元数据管理服务提供数据逻辑视图,从而将控制通路与数据通路分离,提升了文件读写能力,同时兼具 SAN 高速直接访问特点及 NAS 的分布式共享特点[1].

云计算中的分布式对象存储系统通常采用友好访问接口使对象存储拥有跨平台数据共享的特点,通过接口可以执行数据 CRUD 和对象属性等操作,适合加载 MB 级别的图片、文档、音频等到内存中进行快速处理,其计算效率和用户体验提升是显著的,因而对象存储更适合云计算模式下的互联网应用.对象存储的另一个特点是扁平的数据组织结构,抛弃了庞大的传统目录树管理,利用多维度的元数据属性进行存储数据的检索和管理[2].综上,对象存储并不是文件系统和实时数据存储系统,而是一个存储永久类型静态数据的无中心控制节点的存储系统,存储的静态数据可以进行检索、访问、必要时的更新.相较于 HDFS 等云存储模式,更适合存储图片、邮件、音频等数据,避免了单点故障和性能瓶颈,拥有更强的扩展性、冗余和持久性,适合高性能集群使用.美国国家标准化组织(ANSI)于 2005 年批准了对象存储的标准规范[3],目前,对象存储已得到了广泛应用,代表性的大规模实现如 AWS 的 S3、华为的 OBS、开源实现 OpenstackSwift[4]和 Ceph 等.

但是随着云存储技术的发展,其动态复杂性、开放性和资源高集中性等特点不可避免地带来了数据安全性问题,用户将数据托管给云服务提供商存储和管理,会失去对数据的控制.因此,需要提供一种灵活可靠的安全机制和体系结构保护用户的机密性、完整性和可用性[5].事实上,当前安全问题呈现上升趋势,已成为制约云存储发展的重要因素.2012 年,亚马逊数据中心中断,导致网络瘫痪;2014 年,iCloud 云平台被黑客攻击,导致用户信息泄露.因此,对数据资源的访问控制是云存储安全核心[6].

访问控制技术是保障数据不被非法访问的重要手段[6],实现对云存储中大量资源的访问控制机制,能够使用户将数据安全地托管至云平台.因此,许多学者提出了不同的方案,主要集中在 3 个方面,见表 1.

Table 1 Researchon access controlin cloud computing

表 **1** 云计算访问控制技术

名称	常用方法	作用
访问控制策略	访问控制列表 ACL,访问控制矩阵	准入控制
访问控制模型	DAC,MAC,RBAC 等	静态权限分配
属性加密机制	KPABE,CPABE	数据机密性

虽然云计算中的访问控制机制研究不断深入,但是在实际应用场景中,一方面,随着对象云存储的规模扩大,通常需要对图片、音视频等具有层级相关性特征的媒资数据进行分类管理.如存在分类关系:音乐→华语音乐→邓丽君,若用户 *A* 上传音乐 1.MP3 指定其分类属性为邓丽君,而一个拥有权限级别华语音乐的用户 *B*,应能

够访问邓丽君下的文件.另一方面,在海量对象数据的存储下,为实现数据的受控共享和存储隔离,应进行高效的细粒度的访问控制;为满足大规模用户的并发访问请求,还应适应云环境建立分布式授权中心.然而,由于与传统块存储在适用范围、底层架构、原理和模型方面的差异,基于对象数据格式的云存储结构面临着更多新的访问控制技术方面的挑战.

(1) 访问控制架构

对象存储的架构与传统块存储模式甚至 HDFS 不同,传统块存储由于缺少专有的对象存储元数据服务器,访问控制决策需要访问数据本身,导致授权和决策的效率慢,并发性能较低.同样的 HDFS 虽然存在简单的元数据服务器,但其元数据 Namenode 节点存在单点故障、检索效率过慢的问题,同样限制访问控制的性能.而对象存储采用分布式的平坦化存储,可以避免访问数据本身以及单点引发的效率问题,提高访问的并发性能.但是这种环境下,访问控制的架构也需满足并解决分布式授权、权限判定问题.同时,由于对象存储采用的一致性哈希算法,使得存储的不同等级数据随机分布在存储结点上,因此,对象存储中的访问控制还需要解决数据隔离的安全问题.

(2) 细粒度访问控制

在以块存储为核心的存储模式中,同样由于缺少丰富的属性信息,目前的方案只能依赖 ACL 列表实现粗粒度的访问控制,使用中存在提权风险;同时,由于访问控制策略不够灵活,这些方案在分布式的云计算环境中已不再适用于分类分级数据的应用间交互、网络资源共享.为解决这些瓶颈的限制,随着 Sahai 基于公钥密码在 IBE 的机制上提出 KPABE 和 CPABE[7]的访问控制模型,基于属性加密的方案受到了广泛关注.虽然这些方案以属性定义密钥,可以利用椭圆曲线的双线性对与访问控制结构结合,使得当用户属性符合加密时嵌入的访问控制规则即可解密,有效解决了云计算环境下细粒度访问控制和解密方不固定的问题.但是目前已提出的方案中尚无有效解决分类分级数据的访问控制方法,因此,面对对象云存储分类分级数据的对象属性描述的场景时,如何利用这些属性实现细粒度的访问控制策略,有效解决用户授权访问,保障云数据安全,是一个技术挑战.

(3) 元数据管理

传统的属性加密通常还有个缺陷——加密密文存储空间及加解密运算量随着属性数目增长而线性膨胀,对象云计算环境中面对海量属性数目,属性相关的密文元数据大小尤其将限制对象存储的元数据管理,不利于细粒度访问控制的应用.

为解决云存储中的上述安全和效率问题,本文提出了一种对象云存储中分类分级数据的访问控制方法,克服了上述的安全挑战,解决了已有方案中的缺陷,利用灵活访问策略适应了应用场景,并实现了如下目标.

(1) 提出一种访问控制模型以及对象云存储的访问控制体系结构,使得能够明确系统结构的实现方式,并从理论上形式化表达分类分级数据的访问控制策略;
(2) 提出的访问控制模型、算法能够解决分类分级特征的对象数据的细粒度访问控制;
(3) 提出的访问控制算法可以利用对象数据丰富的分类分级属性元数据参与访问控制策略的运算,生成只有满足分类分级层级支配策略的用户才可解密访问的密文数据;
(4) 提出的访问控制算法得到的密文可作为对象数据的访问控制属性储存在元数据服务器中,为提高管理及访问效率,得到的密文长度应固定;
(5) 提出的访问控制算法应满足分布式授权和分布式存储的云架构,同时,得到的对象数据应隔离;
(6) 通过理论分析及实验系统实现,验证其合理性.

综上,在分类分级特征下,本文结合对象存储的优点,解决了对象文件的细粒度访问控制问题,保障云存储安全,是具有很强的研究意义的.

1.1 本文贡献

基于上述原因,本文综合属性加密机制、强制访问控制、对象存储各自的优势,并结合分类分级的属性特点,提出了一个基于安全标记对象存储访问控制模型,同时,设计了云计算访问控制系统的体系结构.在该模型中,只有当用户拥有的安全标记满足一定的策略支配访问数据的安全标记时,通过具体的分类分级数据的属性

访问控制算法(fine-grained access control algorithm for classified and graded data,简称 CGAC 算法),用户才可以解密访问数据.该算法将分类属性树之间的层级支配关系反映在密钥和密文中,只有能支配的属性关系才能进行运算,因此较好地解决了分类分级特点文件的细粒度访问控制,实现了云平台存储池数据隔离,且本文的算法可以从数学上证明其具有抵抗选择明文攻击.另外,本文结合对象存储的特点实现了分布式层次授权管理机制,同时,利用定长密文的设计结构将其与对象存储元数据管理紧密结合,最终在一种对象存储中实现本文方案,并验证.

1.2 本文组织结构

本文第 2 节给出访问控制方案的相关研究.第 3 节介绍相关预备知识.第 4 节构造一个灵活的适合分类分级特征的基于安全标记对象存储访问控制模型及系统结构.第 5 节给出详细的 CGAC 算法.第 6 节中给出算法证明及效率分析.最后,第 7 节中设计实现访问控制系统.

2 相关研究工作

BLP 模型[8]是传统的强制访问控制,主要用于保护机密等级高的系统,根据数据的敏感等级及分类特点,利用主客体标记,可以实现多级的安全策略,保障数据的单向流动性.因此,强制访问控制可以实现数据的安全隔离,具有高安全性的特点.目前,对 BLP 模型在云计算中的研究主要集中在修改传统的 BLP 模型,使其更适用于云计算环境中.Shen 等人[9]讨论了 BLP 模型在云计算中的体系结构和访问控制算法,Lin 等人[10]以 BLP 模型和 Biba 模型为基础,结合云计算环境特点,用行为等相关信息提出了 CCACSM 模型保障资源安全性.

在属性加密的机制中,有一部分关于层次化属性加密的方案.Horwitz[11]首次提出了具有等级结构的身份密码机制 HIBE,利用多 PKG 的 IBE 方案解决单个 PKG 压力大易攻陷的缺点.该方案中,每个 PKG 对应私钥由上层父节点生成,且每个 PKG 只负责部分用户密钥的生成.Wan 等人[12]在基于密文策略的属性集合加密(CP-ASBE)基础上提出了 HASBE 算法解决层次用户的访问控制.2014 年,Deng 等人[13]通过线性秘密共享和随机密钥分配技术提出了一种层次化 CP-HABE 算法,类似的还有文献[14].但这些方法都是针对用户的层次化解决方案,并不适用于分类属性的树形结构需求.Wang 等人[15]提出了一种将两个低层次文件合并形成高层次文件的分层访问控制算法.2015 年,Liu 等人[16]在 HABE 算法的基础上提出了一种解决树形结构的层次化属性基加密的方法.然而这几种方案的密文是不固定的,当属性扩张时,密文线性增长,不适用于云计算的场景,无法作为元数据进行存储管理.为提供固定长度的密文适用于云存储场景,Ge 等人[17]利用用户拥有定量默认属性的方式提供了阈值访问控制结构的定长密文算法.张欣晨等人[18]在 Ge 的方案基础上,在 Hadoop 平台实现了定长密文访问控制模型,实验验证了可行性.但 Ge 的方案中,层次化结构依然关注在用户的组织上.

而在对象存储的访问控制中,Biswas 等人[19]提出了面向内容级别的访问控制,利用标记作为 JSON 对象进行存储和策略判定.而在实际产品中,对象存储的厂商对访问控制关注度不够,如 Rackspace 公司开源的 Openstack Swift 产品只利用了其 Openstack 架构中 Keystone 组件的身份认证技术实现了简单的 RBAC 机制的人员管理,而对于对象数据,则依赖自主的 ACL 表[4].可见,如何将强制访问控制模型、基于属性访问控制与分类分级特点的资源相结合,利用各自优势在对象存储中实现安全的云存储系统方案,是值得进一步研究的.

3 预备知识

3.1 对象存储概述

在基于对象的存储中,对象数据是以固定接口提供非结构化文件访问操作的一类存储容器.同时维护一组描述文件数据特性属性值的元数据管理,通常可以利用元数据来实现阻止数据非法访问的安全策略.

如图 1 所示的对象存储与传统块存储的结构对比,块存储包括上层应用逻辑表达结构、命名服务、访问控制结构等用户组件和负责将数据逻辑结构映射到存储介质上的存储管理组件.对象存储与之相较,主要存在两处不同点,存储管理组建下移至于物理设备层,块接口变为文件访问对象接口,通过访问接口可以实现对象数据

的增删改查、设置属性等操作,具有跨平台的特点.另一方面,由于物理设备具有底层文件系统特征,分布式对象存储元数据管理只需通过负载均衡提供逻辑拓扑结构即可,从而避免 NAS 元数据服务器瓶颈.

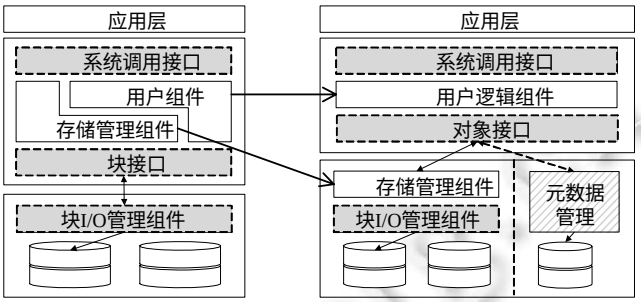


Fig.1 Difference between object storage and block storage

图 1 对象存储与传统块存储的结构对比

正是因为元数据管理的存在,通常利用这些属性信息进行对象存储访问控制的实现.如 Openstack Swift 对象存储组件将 ACL 列表存储在元数据中,以 JSON 的形式在 Header 信息中返回.此时,请求将由 Swift 的访问控制中间件处理,并结合身份认证中间件对用户操作进行访问控制决策.本文方案亦是通过元数据信息、对象存储 REST 接口及中间件处理流程将本文提出的分类分级属性访问控制方法与对象存储实现了很好的结合.

3.2 双线性映射及复杂性假设

Boneh[20]提出了椭圆曲线上的双线性映射后,双线性映射被广泛应用于加密、签名等领域.现有的 ABE 机制也大多基于双线性映射来实现,现给出双线性映射的定义.

定义 **1**. 设 G_1 和 G_2 是素数 p 的循环群,g 是 G_1 的生成元,则双线性映射 $e:G_1\times G_1\to G_2$ 具有如下性质.

- 双线性性:对于所有的 $u,v\in G_1$ 以及 $a,b\in Z_p$,都有 $e(u^a,v^b)=e(u,v)^{ab}$;
- 非退化性:对于生成元 g,始终有 $e(g,g)\neq 1$;
- 可计算性:对于任意的 $u,v\in G_1$,能够在一个多项式时间内计算 $e(u,v)$.

定义 **2**. 判定性 l-BDHE 假设(decisional l-bilinear Diffie-Hellman exponent assumption)[21]定义如下.

给定 $2l+1$ 个元素组成的向量 $\vec{y}=(g,h,g_1,g_2,...,g_l,g_{l+2},...,g_{2l})\in G_1$,其中,$g_i=g^{\alpha}$,$\alpha\in Z_p$ 未知.任取随机数 $T\in G_2$,判定 $e(g,h)^{\alpha^{l+1}}=T$ 是否成立.若对任意多项式时间,敌手优势均小于可忽略的值,则称判定性 l-BDHE 成立.

3.3 安全模型

本文方案的安全模型由如下一系列游戏来定义,由敌手 $\mathcal{A}$ 和模拟器 $\mathcal{B}$ 共同参与进行,如果在游戏中,敌手的优势是可以忽略的,则称 CGAC 算法在选择标记及适应性选择密文攻击下是不可区分的.具体定义如下.

(1) 初始化:在游戏开始之前,敌手 $\mathcal{A}$ 首先给出要攻击的客体安全标记 L^* 以及访问控制阈值 t^*;

(2) 系统建立:模拟器 $\mathcal{B}$ 收到敌手 $\mathcal{A}$ 选取的挑战客体标记 L^* 后,模拟器运行系统建立算法得到系统主密钥 MK 和系统公开参数 PK,模拟器 $\mathcal{B}$ 将 PK 发送给敌手 $\mathcal{A}$,并秘密保存 MK;

(3) 查询阶段 1:敌手 $\mathcal{A}$ 进行多项式次数适应性私钥提取和解密询问,模拟器 $\mathcal{B}$ 利用掌握的信息进行响应:

- 私钥提取询问:敌手 $\mathcal{A}$ 任意选择用户的安全标记 L,满足 $|L\cap L^*|<t^*$,模拟器运行私钥提取算法得到相应于用户安全标记 L 的私钥 sk,并将 sk 返回给敌手 $\mathcal{A}$;
- 解密询问:敌手 $\mathcal{A}$ 任意选择安全标记 L_i 和对应的消息 M 的密文 C_i,要求模拟器输出 C_i 对应的明文 M;

(4) 挑战阶段:敌手 $\mathcal{A}$ 向模拟器 $\mathcal{B}$ 提供两个等长的待挑战消息 $\{m_0,m_1\}$,模拟器随机选择 $\beta\in\{0,1\}$,利用初始化阶段敌手给定的 L^* 对 m_β 进行加密的密文 $CT^*=Enc(m_\beta,L^*,PK)$,并将密文 CT^* 发送给敌手 $\mathcal{A}$;

(5) 查找阶段 2:与查找阶段 1 类似,敌手$\mathcal{A}$仍可进行多项式次数的适应性私钥提取和解密询问,但是敌手$\mathcal{A}$不能提交被挑战密文(CT^*,L)的解密查询或满足$|L\cap L^*|\geq t^*$的 L 的私钥提取询问;

(6) 猜测:最终敌手$\mathcal{A}$给出β值的一个猜测β'.如果$\mathcal{A}$给出了正确的猜测$\beta=\beta'$,则称$\mathcal{A}$赢得了游戏.其中,$\mathcal{A}$在 IND-sLa-CCA 游戏中的优势被定义为$|\Pr[\beta'=\beta]-1/2|$.

定义 **3**. 若在多项式时间内,任何敌手在上述游戏中最多进行了 q_K 次私钥提取查询和至多 q_D 次解密查询,其优势仍是可以忽略的,则称本方案分类分级数据的属性加密算法(CGAC 算法)是 IND-sLa-CCA 安全的.

特别地,CGAC 算法的安全性是抗适应性选择密文攻击(CCA),比 CCA 安全性弱一些的模型是抗适应性选择明文攻击(CPA)的.在此安全模型下,敌手不允许进行解密查询.现在已经有许多成熟方案将具有抗适应性选择明文攻击的方案转化为抗适应性选择密文攻击方案,而仅仅需要增加一些少量运算,如文献[22,23]所提到的方案.因此,本文中的方案将重点讨论抗选择明文攻击下的算法安全性.

4 对象云存储中分类分级数据的访问控制模型

4.1 访问控制方案模型

本文旨在对象存储中分类分级数据的场景下,实现细粒度访问控制,保护云服务用户的数据安全.对象存储中拥有海量典型的分类属性特点的数据格式,如音频媒资等,本文提出的访问控制模型利用这种树形拓扑结构的特点,结合传统强制访问控制模型及基于属性的访问控制模型的优点,实现了对象数据的细粒度访问控制.

对象数据的分类特性表示数据的从属类别,一个大的类别可以由多个子类别的集合组成,子类别也可划分,层层分类构成树形拓扑的分类关系图,具有从属关系的类别构成分类树,分类树中的节点称做分类范畴.而分类分级特点在访问控制中的类似从属关系,表示为其上级对下级的支配关系,其定义如下.

定义 **4**. 定义安全级别为一个全序的集合$\mathbb{S}=\{level_1,level_2,\ldots,level_n|n\in\mathbb{N}^+\}$,其中,$level_1\geq level_2\geq\ldots\geq level_n$.则对于$\forall i,j|i\leq j$,有 $level_i\geq level_j$,称 $level_i$ 支配 $level_j$.

定义 **5**. 分类范畴集 $C=\{C_1,C_2,\ldots,C_n\}$是 n 个根节点分别为 $c_{1,0,0},c_{2,0,0},\ldots,c_{n,0,0}$ 的分类树组成的集合.对于根节点为 $c_{i,0,0}$ 的分类树,设其树的深度为 l_i,同时定义深度为 k 的第 j 个分类为 $c_{i,k,j}$,则分类 $c_{i,k',j'}$包含 $c_{i,k,j}$,表示 $0\leq k'\leq k$ 且存在分类 $c_{i,k',j'}$到分类 $c_{i,k,j}$ 的唯一路径,称为分类 $c_{i,k',j'}$支配 $c_{i,k,j}$,记做 $c_{i,k',j'}\succcurlyeq c_{i,k,j}$,路径为 $p\rightarrow(c_{i,k',j'},\ldots,c_{i,k,j})$.

如图 2 所示,实线箭头指示的路径表示分类$c_{1,k,j}$的从属关系,且其被$c_{1,0,0}$的支配路径表示为$p_{1,k,j}=(c_{1,0,0},c_{1,1,0},c_{1,2,1},\ldots,c_{1,k,j})$.简称为分类 $c_{1,k,j}$ 的分类路径.

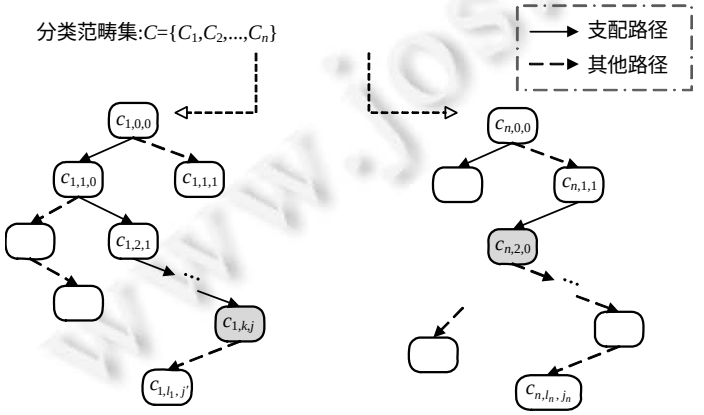


Fig.2 Classified tree and path

图 2 分类树及支配关系图

访问控制策略面向对象数据,也就是说,对象数据是策略的客体.实现细粒度访问控制,除把整个对象数据

作为客体外,其对象元数据的属性也作为访问控制客体考虑进策略中;同时,对于访问控制的主体使用者,每个用户也有一定的属性集合,结合 BLP 模型定义主客体标记.

定义 **6**. 对象数据的标记是三元组:$L_o=\langle A_1,\ldots,A_m,\{C_1,\ldots,C_n\},level\rangle$,同样地,用户的标记也是一个三元组:$L_s=\langle A_1,\ldots,A_{m'},\{C_1,\ldots,C_{n'}\},level\rangle$,其中,

- A_m 表示对象数据或用户的普通属性;
- C_n 表示对象数据或用户的分类范畴属性,即主客体所属分类属性;
- $level$ 是对象数据或用户的安全等级,$level\in\mathbb{S}$.

对于云存储服务的访问控制,其传统的操作读和写转化为了加密上传、写和下载解密、读的过程.数据的属主根据数据分类范畴集及安全级别定义访问控制结构,并加密数据上传至对象云存储服务器.用户访问对象数据,若用户拥有满足细粒度访问控制策略的普通属性集和分类范畴集,且用户的分类范畴支配数据分类属性,用户安全级别支配数据安全级别,则数据可被下载解密.在实际应用中,云存储的其他操作都可以由以上两类构成,故可将这些操作转换为上述操作的组合,再根据细粒度访问控制的策略进行控制.

定义 **7**. 访问控制策略中的其他参与元素定义如下.

(1) 主体集合 $Sub=\{sub_1,\ldots,sub_i\}$,表示访问操作的用户,其中,$Sub.L_s$ 表示用户的主体标记.而 $Sub.L_s.A$ 表示主体的普通属性集,$Sub.L_s.C$ 表示主体的分类范畴集,$Sub.L_s.level$ 表示主体的安全级别;

(2) 客体集合 $Obj=\{obj_1,\ldots,obj_j\}$,表示云存储中所有对象数据,其中,$Obj.L_o$ 表示数据的客体标记.而 $Obj.L_o.A$ 表示客体的普通属性集, $Obj.L_o.C$ 表示客体的分类范畴集,$Obj.L_o.level$ 表示客体的安全级别;

(3) 访问请求操作集合 $R=\{upload,download,read,write\}$,表示主体对客体的访问操作请求;

(4) 请求响应集合 $D=\{yes,no,error\}$,其中,yes 表示请求被允许,no 表示请求被拒绝,$error$ 表示请求出错;

(5) 系统执行状态集合 $V=\{S\times O\times R\}$,其中 $v\in\{S\times O\times R\}$ 表示哪些主体以哪些操作对哪些客体进行请求,$g(v)\in D$ 表示响应结果.

根据上述定义的元素、标记,制定访问控制系统的安全规则.

规则 **1**. 用户执行上传操作成功,同时,用户上传元数据成功,当且仅当用户从主体标记定义对象数据的客体标记及访问控制结构 Γ_{t,L_o} 成功,即:指定访问用户的主体标记包含且至少 t 个属性支配对象数据的属性才能访问,且用户主体标记的安全级别小于等于客体标记的安全级别.即:对于$\forall sub\in S,obj\in O$,有:

$$g(v(sub,obj,upload\vee write))==D.yes\Leftrightarrow$$
$$(sub.L_s\supseteq obj.L_o)\wedge\left(\prod_{i=k_1}^{k_m}sub.L_s.(A|C)_i\geq obj.L_o.(A|C)_i\right)\wedge(sub.L_s.level\leq obj.L_o.level) \tag{1}$$

其中,Def 表示定义操作,I表示访问控制结构,$m\geq t$ 表示分类范畴集的访问控制阈值.

规则 **2**. 用户执行下载和读操作成功,当且仅当用户的主体标记满足对象数据的访问控制结构,即用户主体标记包含且至少 t 个属性支配对象数据的标记,且用户主体标记的安全级别大于等于客体标记的安全级别.即对于$\forall sub\in S,obj\in O$,有:

$$g(v(sub,obj,downoad\vee read))==D.yes\Leftrightarrow$$
$$(\Gamma_{t,L_o}sub.L_s)\wedge(sub.L_s\supseteq obj.L_o)\wedge\left(\prod_{i=k_1}^{k_n}sub.L_s.(A|C)_i\geq obj.L_o.(A|C)_i\right)\wedge(sub.L_s.level\geq obj.L_o.level) \tag{2}$$

其中,n 表示主体标记支配客体的个数且满足$|A|+|C|\geq n\geq t$,I表示访问控制结构,t 表示访问控制阈值.

4.2 访问控制系统模型

本方案以 CGAC 算法为核心,结合云计算,特别是对象云存储系统为基础进行设计,如图 3 所示.本系统模型由用户 User、云存储服务 CSP、主从 KGC 这 3 类参与实体构成.其中:用户的操作为提交上传、下载等文件访问请求操作及对象数据加密、解密操作;云存储服务提供分布式对象云存储服务,包括对象文件存储管理及元数据管理;主从 KGC 为可信第三方主要功能为系统公开参数维护,授权及私钥产生等.3 种角色通过 CGAC 算法

串联,构成对象云存储访问控制模型,既可以解决分类分级存储的使用场景,又满足云存储提供商要求的存储格式,而且可以有效地基于用户的主体标记和对象数据的客体标记解决细粒度访问控制.

CGAC 算法基于 ABE 方案实现上述第 4.1 节描述的访问控制模型,类似文献[7],算法由系统初始化算法 *Setup*,KGC 节点授权算法 *Delegate*,用户授权及密钥产生算法 *KeyGen*,对象数据加密算法 *Encrypt*,密文数据解密算法 *Decrypt* 构成.各核心算法参与访问控制规则实现过程中,组成了如图 3 所示整个系统.具体描述如下.

(0) 主 KGC 执行系统初始化算法 *Setup*,输入安全参数λ,系统属性空间Ω包括普通属性和分类范畴属性,l为分类范畴集最大深度,输出系统公共参数 *PK* 和主密钥 *MK*.其中,*MK* 由主 KGC 安全存储,用以产生授权的私钥;而包含了属性空间的 *PK* 则被公开,参与到其他算法中;

(1) 主 KGC 执行 KGC 授权算法 *Delegate* 为新加入的从 KGC 授权,分直接授权和间接授权两种情况(其中:主 KGC 具有最高权限,即系统初始化时的所有分类属性集;同时为各个从 KGC 授权,通过分层的体系结构分散计算量,减轻主 KGC 负担):

a) 直接授权 *Delegate*,输入属性集 A'、主密钥 *MK*、公开参数 *PK*,由主 KGC 输出授权目标属性集 A'的私钥 SK_A;

b) 间接授权 *Delegate*,输入属性集 *A* 对应上级 KGC 私钥 SK_A、系统公开参数 *PK*、目标 KGC 的属性集 A',其中,A'是 *A* 的子集,输出授权目标属性集 A'的私钥 SK_A;

(2) 用户 *S* 访问云存储服务,提交上传操作请求;

(3) 云存储服务 CSP 验证 $Token_s$ 及 $H(L_s||T||r)$:若不存在或不合法,则重定向至 KGC 域,同时产生随机数 *r* 发送;否则,调转至步骤(8);

(4) 用户 *S* 提交自身主体标记 L_s 和欲生成的对象数据客体标记 L_o,标记包含分类范畴集和安全等级;

(5) 授权 KGC 验证用户标记是否满足规则 1,同时进行用户私钥生成.同样分直接和间接生成两种情况:

a) 直接生成 *KeyGen*,输入系统主密钥 *MK*、公开参数 *PK* 以及用户的主体标记 L_s,输出与用户主体标记关联的私钥 SK_s;

b) 间接生成 *KeyGen*,输入从 KGC 私钥 SK_A、*PK* 以及主体标记 L_s,输出与主体标记关联的私钥 SK_s;

(6) KGC 将生成的用户私钥 SK_s 及验证信息 $H(L_s||T||r)$返回;

(7) 同时,KGC 将验证结果及参数通过安全信道通知云存储服务 CSP;

(8) CSP 生成 $Token_s$ 并返回,同时缓存键值对 $Token_s$ 和用户信息 $H(L_s||T||r)$;

(9) 用户 *S* 通过加密算法 *Encrypt* 加密对象数据,并生成客体标记.该算法进行第 4.1 节描述的规则 1 的判定.算法 *Encrypt*,输入系统公开参数 *PK*、访问控制参数 *t*、文件加密密钥 *DEK*、主体标记 L_s、客体标记 L_o 和对象数据 *O* 进行访问控制:若符合规则 1,则输出密文 *CT* 和元数据 *Meta*;否则拒绝;

(10) 发送密文 *CT* 和元数据 *Meta*,$Token_s$ 到 CSP,CSP 验证 $Token_s$,若合法,则将密文文件作为对象文件存储,同时将元数据存储在云服务元数据管理器和对象数据扩展属性中,如 XFS 文件系统的 XATTRs 扩展属性中;

(11) 当用户 S'想访问对象数据时,同样通过授权获取私钥 $SK_{S'}$及 $Token_{s'}$,发送文件下载请求;

(12) 若用户 S'执行解密算法访问数据.算法 *Decrypt* 输入系统公开参数 *PK*、访问控制参数 *t*、用户私钥 $SK_{s'}$和主体标记 $L_{s'}$、密文 *CT* 和元数据 *Meta*,若 S'满足访问控制规则 2,则输出对象数据 *O*;否则,拒绝访问.

此模型中,云存储服务不仅可以是公有云,也可以为私有云,此时,从 KGC 可以退化融合到对象云存储的各个节点中,用户访问时直接生成 *SK* 和 *Token* 并缓存,不需要进行验证,从而节省部分开销.

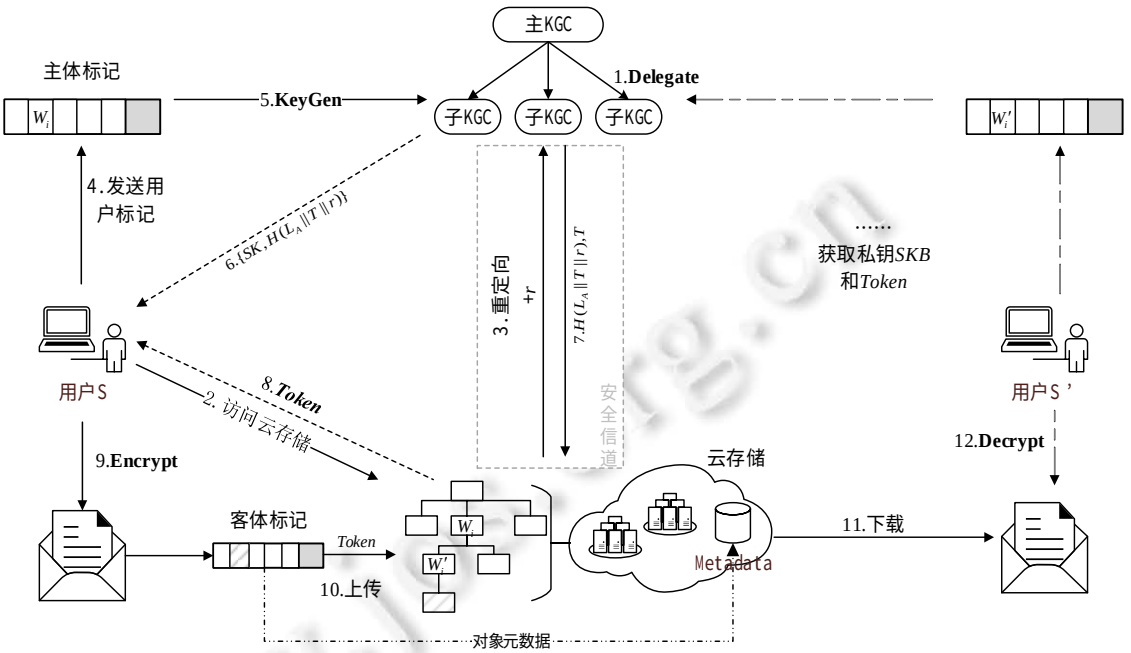


Fig.3 System structure

图 3 系统结构图

5 分类分级数据的属性访问控制算法详述

CGAC 算法结合基于属性加密机制,实现第 4.1 节提出的访问控制模型中的规则,保障云存储中用户数据安全.虽然目前已有大量的属性基加密方法,但是其中大多数 ABE 算法密文大小、加解密计算复杂度随属性集增长而线性增长,而这对于对象元数据管理及存储空间设计增加了负担,且当属性集过大时,过长的密文影响属性基细粒度访问控制效率,这些缺点都极大地限制了属性加密机制在云计算中的应用.分类分级的实际应用场景特点同样影响了基于属性加密的细粒度访问控制在云计算广泛应用.

本文提出的 CGAC 算法在文献[17]的基础上,结合第 4.2 节的访问控制系统模型,主要解决了分类分级的属性加密和分类分级下定长密文构造两个难题,实现了分类分级对象云存储的细粒度访问控制方案.

5.1 前提约束

CGAC 算法依赖于秘密分享方案[24],因此我们定义拉格朗日系数$\Delta_{i,S}$,其中,$i\in Z_p$;S 是一个元素在 Z_p 上的集合: $\Delta_{i,S}(x)=\prod_{j\in S,j\neq i}\frac{x-j}{i-j}$.

为简化算法设计,在算法描述前进行一些结构的定义,可以将普通属性集和安全级别映射为分类树,故下面不再区分讨论两种属性,从而只考虑分类情况下的属性加密机制,使运算和算法描述更简洁.

定理 **1**. 设普通属性集 $A=\{A_1,\ldots,A_m\}$,对每个属性 A_i,可看作深度为 0 的分类树的根,参与访问控制运算.

定理 **2**. 设安全级别空间为$\mathbb{S}=\{level_1,level_2,\ldots,level_n|level_1\geq level_2\geq\ldots\geq level_n,n\in\mathbb{N}^+\}$,随机选取 $level_0$,定义分类树 $\tilde{\mathbb{S}}$ 为根是 $level_0$,叶子属性是 $level_n$ 的分类树 $level_0\rightarrow level_1\rightarrow\ldots\rightarrow level_n$. $\tilde{\mathbb{S}}$ 有以下属性:深度 k 的节点属性为 $level_k$,且根节点到该节点路径为$(level_0,level_1,\ldots,level_k)$,则由分类树的定义,任意深度 $1\leq i\leq j\leq n$,满足支配关系 $level_i\geq level_j$,故 $\tilde{\mathbb{S}}\Leftrightarrow\mathbb{S}$.

于是,根据定理 1、定理 2 可知,任意普通属性及安全级别可转化为分类属性,则可将访问控制规则 2 中的访问控制结构定义转化为:

主体标记包含且至少有 t 个分类属性树中的分类范畴支配客体标记对应的分类范畴,则可解密访问数据.

5.2 具体步骤

通过上节方法可简化系统复杂度,故在 CGAC 算法中只讨论主体标记包含且至少有 t 个分类范畴支配客体标记的情况.下面将按照核心算法的步骤进行阐述.

(1) $Setup(1^{\lambda},\Omega,l)\rightarrow(PK,MK)$

设 G_1 是一个阶为大素数 p 的双线性群,安全参数 λ 决定了群的大小,g 是 G_1 的一个生成元.选取双线性映射:$e:G_1\times G_1\rightarrow G_2$.设标记属性空间 Ω 是普通属性集,分类范畴集和安全级别的空间集合,记 $\Omega=\{A_1,\ldots,A_m\}\cup\{C_1,\ldots,C_n\}\cup\{level_i|level_1,\ldots,level_k\}=\Omega.A\cup\Omega.C\cup\Omega.level$,其中,$|\Omega|$ 表示空间集的属性个数,$|\Omega|=m+n+1$.设通过定理 1、定理 2,将 Ω 中的属性转化为分类范畴集的 $m+n+1$ 个分类树表示云存储中的分类关系,令 $C_0=\{C_{1,0,0},\ldots,C_{m+n+1,0,0}\}$ 为 $m+n+1$ 棵分类树的根,且每棵树的最大深度为 $d_i(0\le i\le n)$,则最大树深记为 $d=\{d_1,\ldots,d_{n+m},k\}$,其中,前 m 个深度为 0:$d_1=\ldots=d_m=0$.同时,记所有的 $m+n$ 个伴随属性 $V=\{v_0,\ldots,v_{m+n-1}\}$,且每个用户默认拥有所有伴随属性.定义一个哈希函数 $H:\{0,1\}^*\rightarrow Z_p^*$,随机选取 $\alpha\in Z_p^*$,计算 $g_1=g^{\alpha}$.同时,随机选取 $g_2,u_0',u_1',\ldots,u_{2n+2m+1}'\in G_1$,然后,根据分类树的深度选取 $u_{1,1},\ldots,u_{1,d_1},u_{2,1},\ldots,u_{2,d_2},\ldots,u_{n+m+1,1},u_{n+m+1,k}\in G_1$,并且定义 $u_{1,0}=u_{2,0}=\ldots=u_{m+n+1,0}=1$.

事实上,由安全等级转化的分类树与其他分类树无区别,因此可以将其看做是第 $m+n+1$ 棵分类树,其深度 $d_{m+n+1}=k$.于是,系统生成的公开参数 $PK=\langle g,g_1,g_2,e(g_1,g_2),\{u_i'\}_{0\le i\le 2n+2m+1},\{u_{i,j}\}_{0\le i\le m+n+1,0\le j\le d_i}\rangle$,主密钥 $MK=\alpha$.

(2) *KeyGen*

KeyGen 是 KGC 为用户产生私钥的过程,根据方案模型中描述,用户密钥产生包括直接生成、间接生成.

a) 直接生成 $KeyGen(MK,PK,L_s)\rightarrow SK_s$:

- 随机选择 $m+n$ 阶多项式 q,满足 $q(0)=\alpha$;
- 若主体标记 L_s 的属性空间属于 KGC 的属性空间,即 $\Omega_s\subseteq\Omega_{KGC}$,则用户的计算属性集为 $\Omega_s\cup V$.对于每一个元素 $c\in\Omega_s\cup V$,由定理 1 和定理 2 可知,c 为 Ω 中某个分类树中的元素,设 $c_i=c_{h_i,k_i,j_i}$ 为第 h_i 个分类树中深度为 k_i 的第 j_i 个属性元素,则其路径为 $(c_{h_i,0,0},c_{h_i,1,j_1},\ldots,c_{h_i,k_i-1,j_{k_i-1}},c_{h_i,k_i,j_i})$;
- 对于每个属性 c_i,随机选择 $r\in Z_p^*$,计算 $sk_{c_i}=\langle a_i,b_i,\{d_{i,j}\}_{j\ne i,0\le j\le 2m+2n+1},\{e_{h,j}\}_{h\ne i,0\le j\le d_h;h=i,k_i<j\le d_h}\rangle$,其中,$a_i=g_2^{q(H(c_i))}(u_0'u_i'\prod_{\delta=0}^{k_i}u_{i,\delta}^{c_{h_i,\delta,\xi}})^{r_i}$,$b_i=g^{r_i}$,$d_{i,j}=u_j'^{r_i}$,$e_{h,j}=u_{h,j}^{r_i}$.特别的,如果属性 c 在分类中的位置为根节点,则 k=0,则 sk_c 中的退化为 $a_i=g_2^{q(H(c))}(u_0'u_i')^{r_i}$;
- 输出生成的 L_s 的私钥:$SK_s=\{sk_c\}_{c\in\Omega_s\cup V}$;

b) 间接生成 $KeyGen(SK_A,PK,L_s)\rightarrow SK_s$:

- 若主体标记 L_s 的属性空间 $\Omega_s\subseteq\Omega_{KGC}$,判断其满足 KGC 的属性集:$\Omega_s\subseteq A$,否则返回 $\perp$;
- 由 *SK* 的生成算法可知属性集为 A 的 KGC 私钥 SK_A 组成形式为 $SK_A=\{sk_c\}_{c\in A\cup V}$.对于每个属性 $c_i'\in\Omega_s\cup V$,若 c_i 覆盖 c_i',随机选择 $r\in Z_p^*$,并计算:

$$\begin{aligned}sk_{c_i'}&=\langle a_i',b_i',\{d_{i,j}'\}_{j\ne i,1\le j\le 2m+2n+1},\{e_{h,j}'\}_{h\ne i,0\le j\le d_h;h=i,k_i<j\le d_h}\rangle\\&=\langle a_i\prod_{j=k_i}^{k_i'}e_{i,j}(u_0'u_i'\prod_{\delta=0}^{k_i'}u_{i,\delta}^{c_{h_i,\delta,\xi}'})^{r_i},b_ig^{r_i},\{d_{i,j}u_j'^{r_i}\}_{j\ne i,1\le j\le 2m+2n+1},\{e_{h,j}u_{h,j}^{r_i}\}_{h\ne i,0\le j\le d_h;h=i,k_i'<j\le d_h}\rangle\end{aligned};$$

- 输出生成的 L_s 的私钥:$SK_s=\{sk_{c'}\}_{c'\in\Omega_s\cup V}$.

(3) *Delegate*

Delegate 为主 KGC 为新加入的从 KGC 授权生成私钥的算法,类似于 *KeyGen*,KGC 授权产生分两种情况:直接授权、间接授权.

a) 直接授权 $Delegate(MK,PK,A')\rightarrow SK_A$:

- 随机选择 $m+n$ 阶多项式 q,满足 $q(0)=\alpha$;
- 若从 KGC 属性空间 $A'\subseteq\Omega$,则用户的计算属性集为 $A'\cup V$.对于每一个元素 $c\in A'\cup V$,由定理 1 和定理 2 可知,c 为 Ω 中某个分类树中的元素,设 $c_i=c_{h_i,k_i,j_i}$ 为第 h_i 个分类树中深度为 k_i 的第 j_i 个元

素,则其路径为 $(c_{h_i,0,0},c_{h_i,1,j_1},...,c_{h_i,k_i-1,j_{k_i-1}},c_{h_i,k_i,j_i})$;

- 对于每个属性 c_i,随机选择 $r\in Z_p^*$,计算 $sk_{c_i}=\langle a_i,b_i,\{d_{i,j}\}_{j\neq i,0\le j\le 2m+2n+1},\{e_{h,j}\}_{h\neq i,0\le j\le d_h;h=i,k_i<j\le d_h}\rangle$,其中, $a_i=g_2^{q(H(c_i))}(u_0'u_i'\prod_{\delta=0}^{k_i}u_{i,\delta}^{c_{h_i,\delta,\xi}})^{r_i}$, $b_i=g^{r_i}$, $d_{i,j}=u_j'^{r_i}$, $e_{h,j}=u_{h,j}^{r_i}$;
- 输出生成的 L_s 的私钥:$SK_{A'}=\{sk_c\}_{c\in A'\cup V}$;

b) 间接授权 $Delegate(SK_A,PK,A'\subseteq A)\to SK_{A'}$:

- 若从 KGC 属性空间 $A'\subseteq\Omega$,判断其满足上级 KGC 的属性集:$A'\subseteq A$,否则返回⊥;
- 由 *SK* 的生成算法可知,属性集为 A 的 KGC 私钥 SK_A 组成形式为 $SK_A=\{sk_c\}_{c\in A\cup V}$.对于每个属性 $c_i'\in A'\cup V$,随机选择 $r\in Z_p^*$,并计算:

$$sk_{c_i'}=\langle a_i',b_i',\{d_{i,j}'\}_{j\neq i,1\le j\le 2m+2n+1},\{e_{h,j}'\}_{h\neq i,0\le j\le d_h;h=i,k_i<j\le d_h}\rangle$$
$$=\langle a_i\prod_{j=k_i}^{k_i'}e_{i,j}(u_0'u_i'\prod_{\delta=0}^{k_i'}u_{i,\delta}^{c_{h_i,\delta,\xi}'})^{r_i},b_ig^{r_i},\{d_{i,j}u_j'^{r_i}\}_{j\neq i,1\le j\le 2m+2n+1},\{e_{h,j}u_{h,j}^{r_i}\}_{h\neq i,0\le j\le d_h;h=i,k_i'<j\le d_h}\rangle;$$

- 输出生成的 L_s 的私钥:$SK_{A'}=\{sk_{c'}\}_{c'\in A'\cup V}$.

(4) $Encrypt(PK,t,DEK,L_s,L_o,O)\to(CT,Meta)$ or (⊥)

为保证对象数据机密性和完整性,用户将数据加密后存储至云服务 CSP.由于属性加密算法不适用于直接加密文件,故通常的做法是:使用密钥 *DEK* 及对称加密算法加密数据生成密文文件 $E_{DEK}(File)$,再使用 CGAC 算法加密该 *DEK*,得到对称密钥密文 $E_{CGAC}(DEK)$.访问用户通过依次解密密钥密文和密文文件从而访问数据.

随机选取 $DEK\in G_2$.令 Ω_s 和 Ω_o 分别表示主客体的属性空间.

根据用户的主体标记 L_s 和访问控制阈值 t 判断对象数据的客体标记 L_o,如果不满足下列关系之一,则返回⊥:普通属性集 $L_o.A\subseteq L_s.A$;分类范畴集 $L_o.C\subseteq L_s.C$;安全等级 $L_o.level\ge L_s.level$;阈值 $1\le t\le|L_o.A|+|L_o.C|$,其中,t 表示用户主体标记有 t 个支配客体标记方可访问数据.

接着,随机选择 $s\in Z_p^*$ 和伴随属性的前 $m+n+1-t$ 个 $V_t=\{v_0,\ldots,v_{m+n-t}\}$.对于每一个分类属性 $c\in L_o.C$,设其是第 j 棵树种深度为 k 的第 ξ 个分类,则分类属性 c 其路径为 $P=(c_{j,0,0},\ldots,c_{j,k,\xi})$.按如下方式计算:

$$E_{CGAC}(DEK)=(E_1,E_2,E_3),\text{其中},E_1=DEK\cdot e(g_1,g_2)^s,E_2=g^s,E_3=\left(u_0'\prod_{j\in\Omega_o\cup V_t}\left(u_j'\prod_{\delta=0}^{k_j}u_{j,\delta}^{c_{j,\delta,\xi}}\right)\right)^s \tag{3}$$

输出密文 $E_{CGAC}(DEK)$和 $E_{DEK}(File)$,定义密文文件的元数据信息 *Meta* 包含客体标记 L_o 和 $E_{CGAC}(DEK)$,密文文件存储格式如下:$(Meta,Data)=((OID,L_o,E_{CGAC},\ldots),E(PT))$,其中,*OID* 表示元数据中的对象数据唯一 *ID*.

(5) $Decrypt(PK,SK_{s'},L_{s'},CT,Meta,t)\to O$ or ⊥

用户 *S'* 申请访问对象数据,云服务 CSP 验证用户提交的主体标记 $L_{s'}$,如果满足规则 2,则返回密文数据 *CT*.用户获取密文后,同样首先调用 *Decrypt* 算法,如果用户主体标记 $L_{s'}$ 满足访问控制规则 2,说明用户 *S'* 拥有足够多的属性支配客体的相应属性,因此存在一个子集 $\Omega_{S'}=\{v|v\le w,w\in\Omega',v\in\Omega_o\}$,满足 $|\Omega_{S'}|=t$,且当满足 $L_{s'}.level\ge L_o.level$ 时,主体标记的属性集中有 t 个分类范畴(普通属性和分类属性、安全等级)支配客体标记,使得用户 *S'* 可以使用自己的私钥 $SK_{s'}=\{sk_i\}_{i\in\Omega_{s'}\cup V}$ 解密密文 $E_{CGAC}(DEK)$获取密钥 *DEK*,然后,使用对称密钥解密密文数据获得原始明文.

同时,对于每一个 Ω_o 中的分类属性 $c_i=c_{h_i,k_i,j_i}$,其根路径为 $(c_{h_i,0,0},c_{h_i,1,j_1},...,c_{h_i,k_i-1,j_{k_i-1}},c_{h_i,k_i,j_i})$.设 $c_i'=c_{h_i,k_i',j_i'}$ 是主体属性中支配 c_i 的属性,则其路径同样从根 $c_{h_i,0,0}$ 开始为 $(c_{h_i,0,0},c_{h_i,1,j_1},...,c_{h_i,k_i'-1,j_{k_i-1}'},c_i')$.满足以下公式:

$$c_{h_i,\delta,j_\delta}=c'_{h_i,\delta,j_\delta},1\le\delta\le k_i'.$$

已知密文 $E_{CGAC}(DEK)=(E_1,E_2,E_3)$,根据用户私钥计算下列等式:

$$d_{i,0}''=a_i'\cdot\prod_{\delta=k_i'+1}^{k_i}e_{i,\delta}'^{c_{h_i,\delta,j_\delta}} \tag{4}$$

令 D_1,D_2 为解密算子,同时计算下列等式:

$$D_1=\prod_{i\in\Omega_{s'}\cup V_t}\left(d''_{i,0}\cdot\prod_{j\in\Omega_o\cup V_t,j\neq i}\left(d'_{i,j}\cdot\prod_{\delta=1}^{k_j}e'^{c_{h_j,\delta,j_\delta}}_{i,\delta}\right)\right)^{\Delta_{H(c),\Omega_{s'}\cup V_t}(0)} \tag{5}$$

$$D_2=\prod_{i\in\Omega_{s'}\cup V_t}(b'_i)^{\Delta_{H(c),\Omega_{s'}\cup V_t}(0)} \tag{6}$$

则 *DEK* 可以计算解密出:

$$DEK=E_1\cdot e(E_3,D_2)/e(E_2,D_1) \tag{7}$$

然后,利用解压的 *DEK* 解密密文文件得到数据:

$$File=D_{DEK}(E_{DEK}(File)) \tag{8}$$

6 算法证明及效率分析

6.1 正确性证明

假设用户和数据的标记满足访问控制规则 2,即:主体标记 $L_{s'}$的属性空间任选 $t-1$ 个普通属性和分类属性支配客体中属性且 $L_{s'}.level\geq L_o.level$,组成 t 个元素$\Omega_{s'}$,则有:

$$\left.\begin{aligned}D_1&=\prod_{i\in\Omega_{s'}\cup V_t}\left(d''_{i,0}\cdot\prod_{j\in\Omega_o\cup V_t,j\neq i}\left(d'_{i,j}\cdot\prod_{\delta=1}^{k_j}e'^{c_{h_j,\delta,j_\delta}}_{i,\delta}\right)\right)^{\Delta_{H(c),\Omega_{s'}\cup V_t}(0)}\\&=\prod_{i\in\Omega_{s'}\cup V_t}\left(g_2^{q(H(c))}(u'_0)^{r_i}\cdot\prod_{j\in\Omega_o\cup V_t}\left(d'_{i,j}\cdot\prod_{\delta=0}^{k_j}e'^{c_{h_j,\delta,j_\delta}}_{i,\delta}\right)\right)^{\Delta_{H(c),\Omega_{s'}\cup V_t}(0)}\\&=g_2^{\sum\limits_{i\in\Omega_{s'}\cup V_t}q(H(c))\Delta_{H(c),\Omega_{s'}\cup V_t}(0)}\cdot\left(u'_0\cdot\prod_{j\in\Omega_o\cup V_t}\left(u'_j\cdot\prod_{\delta=0}^{k_j}u_{j,\delta}^{c_{h_j,\delta,j_\delta}}\right)\right)^{\sum\limits_{i\in\Omega_{s'}\cup V_t}r_i\cdot\Delta_{H(c),\Omega_{s'}\cup V_t}(0)}\end{aligned}\right\} \tag{9}$$

令 $E_k=u'_0\cdot\prod_{j\in\Omega_o\cup V_t}\left(u'_j\cdot\prod_{\delta=0}^{k_j}u_{j,\delta}^{c_{h_j,\delta,j_\delta}}\right)$,则 D_1 可简化为

$$D_1=g_2^{\sum\limits_{i\in\Omega_{s'}\cup V_t}q(H(c))\Delta_{H(c),\Omega_{s'}\cup V_t}(0)}\cdot(E_k)^{\sum\limits_{i\in\Omega_{s'}\cup V_t}r_i\cdot\Delta_{H(c),\Omega_{s'}\cup V_t}(0)}=g_2^{\alpha}\cdot(E_k)^{\sum\limits_{i\in\Omega_{s'}\cup V_t}r_i\cdot\Delta_{H(c),\Omega_{s'}\cup V_t}(0)} \tag{10}$$

对于已经获得的密文数据,有:

$$\begin{aligned}E_1\cdot\frac{e(E_3,D_2)}{e(E_2,D_1)}&=DEK\cdot\frac{e(g_1,g_2)^s\cdot e\left((E_k)^s,\prod\limits_{i\in\Omega_{s'}\cup V_t}(b'_i)^{\Delta_{H(c),\Omega_{s'}\cup V_t}(0)}\right)}{e\left(g^s,g_2^{\alpha}\cdot(E_k)^{\sum\limits_{i\in\Omega_{s'}\cup V_t}r_i\cdot\Delta_{H(c),\Omega_{s'}\cup V_t}(0)}\right)}\\&=DEK\cdot\frac{e(g_1,g_2)^s\cdot e\left((E_k)^s,(g)^{\sum\limits_{i\in\Omega_{s'}\cup V_t}r_i\cdot\Delta_{H(c),\Omega_{s'}\cup V_t}(0)}\right)}{e(g,g_2^{\alpha})^s\cdot e\left(g^s,(E_k)^{\sum\limits_{i\in\Omega_{s'}\cup V_t}r_i\cdot\Delta_{H(c),\Omega_{s'}\cup V_t}(0)}\right)}\\&=DEK.\end{aligned}$$

6.2 安全性证明

本节通过将层次分类分级属性访问控制算法规约至 *l*-BDHE 假设,将证明第 5 节给出的 CGAC 算法的选择明文安全性.同样,为简化安全证明,假设访问控制策略中,主体和客体的安全标记都通过定理 1、定理 2,转化为至少有 t 个分类属性支配客体即可访问数据的情况下,则本节给出该模型的安全性证明.

定理 **3**. 设双线性群(G_1,G_2)内阶为大素数 p 的判定性 *l*-BDHE 假设是成立的,则本方案 CGAC 算法是 IND-sLa-CPA 安全的,即:在挑战攻击游戏中,敌手的优势是可忽略的.

证明:设存在一个多项式时间 t 内的敌手$\mathcal{A}$可以通过最多 q 次私钥提取查询,以不可忽略优势ε攻破本方案的 IND-sLa-CPA 安全,则可构造概率多项式时间的算法,以概率不小于ε'、时间不多于 t'的优势攻破判定性 l-BDHE.我们可以构造一个模拟器$\mathcal{B}$在敌手$\mathcal{A}$的帮助下,以同样的优势攻破群 G 中判定性 l-BDHE 问题.

首先,模拟器$\mathcal{B}$选定 l-BDHE 问题的相关参数,生成挑战元组$(g,e(g,g),G_1,G_2,h,y_1,\ldots,y_l,y_{l+2},\ldots,y_{2l},T)$,其中,$g$ 是 G_1 的生成元, $x\in\mathbb{Z}_p^*$.定义 $y_i=g^{x^i}$,为兼容一般性,其中, $y_0=g^{x^0}=g$.并且 T 为 l-BDHE 解 $e(g^{x^{l+1}},h)=e(y_{l+1},h)$ 或 G_2 中的随机元素 $e(g,h)^{\gamma},\gamma\in\mathbb{Z}_p^*$.当 T 是 l-BDHE 解时,模拟器$\mathcal{B}$输出 1;否则,T 是随机数,输出 0.因此,模拟器$\mathcal{B}$扮演游戏中的挑战者,并与敌手$\mathcal{A}$进行如下交互.

- 初始化

在游戏开始之前,敌手$\mathcal{A}$首先给出要攻击的主体安全标记 $L^*=\langle A_1^*,...,A_m^*,\{C_1^*,...,C_n^*\},level^*\rangle$ 以及访问控制阈值 t^*.

- 系统建立

模拟器$\mathcal{B}$收到敌手$\mathcal{A}$选取的挑战主体标记 L^*,根据定理 1、定理 2,将$\bigcup_{i=1}^{m}A^*$,$\bigcup_{i=1}^{n}C^*$ 和 $level^*$转化合并为分类属性 $\bigcup_{i=1}^{m+n}C_{k_i}^*$.设所有的深度为$\{k_1,\ldots,k_{m+n}|0\le k_i\le l\}$分类属性树记做空间$\Omega^*$,则$|\Omega^*|=m+n$,且 $\Omega^*=\bigcup_{i=1}^{m+n}C_{k_i}^*$. 相应地,分类属性 c_{h_i,k_i,j_i}^* 是第 h_i 个分类属性树中深度为 k_i 的第 j_i 个属性,其根为 $c_{h_i,0,0}^*$,则该属性的路径定义为

$$p_{h_i,k_i,j_i}^*=(c_{h_i,0,0}^*,...,c_{h_i,k_i-1,j_{i'}}^*,c_{h_i,k_i,j_i}^*).$$

模拟器$\mathcal{B}$首先生成判定性 l-BDHE 问题相关的系统参数,为简化表示,令 $2n+2m+1=l$ 生成伴随属性集 $V=\bigcup_{i=1}^{m+n-1}v_i$, 其中, $v_i\in\mathbb{Z}_p^*$ 且 $v_1,\ldots,v_{m+n-1}$ 互不相同,全属性空间为$\Omega\cup V$.在一次访问请求中,$m+n-t^*$伴随属性被选中参与运算,表示为 $V_t^*=\{v_1,v_2,...,v_{m+n-t^*}\}$. 其次,模拟器$\mathcal{B}$选择随机数 $\alpha'\in\mathbb{Z}_p^*$,隐含地令 $\alpha=\alpha'+x^l$,设置 $g_1=g^{\alpha}=g^{\alpha'}\cdot y_l$,$g_2=g^x$.进一步,$\mathcal{B}$随机选择 $\alpha_i\in\mathbb{Z}_p^*(0\le i\le l)$,计算 $u_0'=g^{\alpha_0}\prod_{i\in\Omega^*\cup V_t^*}u_i'^{-1}$ 和 $u_i'=g^{\alpha_i}\cdot y_{l-i+1},1\le i\le l$.最终,$\mathcal{B}$随机选择 $\theta_{i,j}\in\mathbb{Z}_p^*$,令 $u_{i,j}=g^{\theta_{i,j}}$,其中,$u_{i,0}=1$.

计算上述参数完毕后,模拟器$\mathcal{B}$将模拟生成的系统参数 $(g,e,G_1,G_2,g_1,g_2,\{u_i'\}_{0\le i\le 2n+2m+1},\{u_{i,j}\}_{0\le i\le m+n+1,0\le j\le d_i})$ 作为公钥发送给敌手$\mathcal{A}$.注意:本步中所有参数均为 G 中独立均匀分布.

- 查询阶段 1

在此阶段,敌手$\mathcal{A}$可以进行多项式次数的适应性私钥提取询问,模拟器$\mathcal{B}$回答相应的询问.

设敌手$\mathcal{A}$对属性集Ω提交不超过 q 次私钥查询,且提交的属性集Ω^*不能通过访问控制结构,即$|\Omega\cap\Omega^*|<t^*$.模拟器$\mathcal{B}$构造一个私钥 SK 通过 $KeyGen$ 过程后传输给$\mathcal{A}$.当模拟器接受到一次私钥查询时,$\mathcal{B}$构造Ω的一个属性子集Γ,使得Γ中的属性支配Ω^*中的属性,即 $\Gamma=(\Omega\cap\Omega^*)\cup V_t^*$.同样,定义Γ',使得 $\Gamma\subseteq\Gamma'\subseteq\Omega^*\cup V_t^*$,且$|\Gamma'|=m+n$,令 $S=\Gamma'\cup\{0\}$.对于每个属性 $c\in\Gamma'$,模拟器随机选择 w,令 $q(H(c))=w$.当 $q(0)=\alpha=\alpha'+x^l$ 时,利用插值公式可唯一确定 $m+n$ 次多项式函数 $q(z)$,于是,模拟器可以对每一个分类属性 $c\in\Omega\cup V$ 按照如下公式计算其对应的私钥 sk_c:

(1) 对于每个属性 $c_i=c_{h_i,k_i,j_i}\in\Gamma'$,$\mathcal{B}$随机选取 $r_i'\in\mathbb{Z}_p^*$,令 $r_i=x^i+r_i'$,结合 $q(H(c_i))=w_i$,计算私钥如下:

$$sk_{c_i}=\langle a_i,b_i,\{d_{i,j}\}_{j\ne i,0\le j\le 2m+2n+1},\{e_{h,j}\}_{h\ne i,0\le j\le d_h;h=i,k_i<j\le d_h}\rangle \tag{11}$$

其中,第 1 部分:

$$\left.\begin{aligned}a_i&=g_2^{q(H(c_i))}(u_0'u_i'\prod_{\delta=1}^{k_i}u_{i,\delta}^{c_{h_j,\delta,\xi}})^{r_i}=(g_2^{w_i})(g^{\alpha_0}\prod_{i\in\Omega^*\cup V_t^*}u_i'^{-1}\cdot u_i'\cdot\prod_{\delta=1}^{k_i}(u_{i,\delta})^{c_{h_i,\delta,\xi}})^{x^i+r_i'}\\&=(g_2^{w_i})(u_0'u_i')^{r_i'}\left(g^{\alpha_0+\sum_{\delta=1}^{k_j}\theta_{j,\delta}c_{j,\delta,\xi}}\prod_{j\in\Omega^*\cup V_t^*,j\ne i}u_i'^{-1}\right)^{x^i}\end{aligned}\right\} \tag{12}$$

第 2 部分: $b_i=g^{r_i}=g^{r_i'+x^i}=y_i\cdot g^{r_i'}$;第 3 和第 4 部分易得: $d_{i,j}=u_j''^{r_i}=(u_j')^{r_i'+x^i}$,$e_{h,j}=u_{h,j}^{r_i}=(g^{r_i'}\cdot y_i)^{\theta_{h,j}}$.

注意,模拟构造 sk_{c_i} 的困难之处在于其包含模拟器未知的 $g^{x^{l+1}}$,由于划分了 Γ,Γ' 和 S 这 3 个集合,对于 $c_i\in\Gamma$, a_i 中的因子 $(u_0'u_i')^{x^i}$ 可以消掉未知的 $g^{x^{l+1}}$.

(2) 对于每个属性 $c_i=c_{h_i,k_i,j_i}\notin\Gamma'$,也就是说 $c\notin\Omega^*\cup V_t^*$,可以通过拉格朗日插值公式计算:

$$q(H(c_i))=\sum_{c'\in\Gamma'}\Delta_{c',S}(H(c_i))\cdot q(H(c'))+\Delta_{0,S}(H(c_i))\cdot q(0) \tag{13}$$

$\mathcal{B}$随机选取 $r_i'\in\mathbb{Z}_p^*$,令 $r_i=r_i'-\Delta_{0,S}(H(c_i))\cdot x^i$,计算私钥如下.

其中,第 1 部分(注意,此时 $u_j'=g^{\theta_j}$):

$$\left.\begin{aligned}a_i&=g_2^{\sum_{c_j\in\Gamma'}\Delta_{c',S}(H(c_i))\cdot w_j+\Delta_{0,S}(H(c_i))\cdot q(0)}\cdot(u_0'u_i'\prod_{\delta=1}^{k_i}u_{i,\delta}^{c_{h_i,\delta,\xi}})^{r_i'-\Delta_{0,S}(H(c_i))\cdot x^i}\\&=g_2^{\sum_{c_j\in\Gamma'}\Delta_{c',S}(H(c_i))\cdot w_j+\Delta_{0,S}(H(c_i))\cdot\alpha'}(u_0'u_i')^{r'}(u_0')^{-\Delta_{0,S}(H(c_i))\cdot x^i}(y_i)^{-\Delta_{0,S}(H(c_i))\alpha_i}\end{aligned}\right\} \tag{14}$$

注意:此时对于 $c_i\notin\Gamma$,a_i 中的因子 $(g_2)^{q(0)}(u_i')^{x^i}$,可以消掉未知的 $g^{x^{l+1}}$.

第 2 部分: $b_i=g^{r_i}=g^{r_i'+x^i}=y_i\cdot g^{r_i'}$;第 3 和第 4 部分易得: $d_{i,j}=u_j'^{r_i}=(u_j')^{r_i'+x^i}$,$e_{h,j}=u_{h,j}^{r_i}=(u_j)^{r_i'+x^i}$.

因此,模拟器$\mathcal{B}$可以计算构造$|\Omega\cap\Omega^*|<t^*$的身份私钥,其分发过程与原有系统模式相同.

- 挑战阶段

敌手$\mathcal{A}$提交两个相同长度的挑战明文 m_0 和 m_1,以及属性空间Ω.模拟器$\mathcal{B}$随机抛一枚硬币,即,随机选择 $\beta\in\{0,1\}$,并构造返回 m_β的密文给敌手$\mathcal{A}$:

$$CT\to\left(m_\beta\cdot T\cdot e(y_1,h^{\alpha'}),h,h^{\alpha_0+\sum_{j\in\Omega^*\cup V_t^*}\sum_{\delta=1}^{k_j}\theta_{j,\delta}c_{j,\delta,\xi}}\right) \tag{15}$$

我们将讨论:当 T 是 l-BDHE 的挑战时,CT 是 m_β的有效加密;当 T 是随机时,CT 是一个随机信息的加密.

首先,注意:由于 h 是 l-BDHE 问题中的均匀分布,故第 2 部分的随机性是均匀分布的.敌手$\mathcal{A}$只得到与 h 相关的 CT.然后计算当 $h=g^c$ 的第 3 部分的正确形式:

$$h^{\alpha_0+\sum_{j\in\Omega^*\cup V_t^*}\sum_{\delta=1}^{k_j}\theta_{j,\delta}c_{j,\delta,\xi}}=\left(g^{\alpha_0}\cdot\prod_{j\in\Omega^*\cup V_t^*}(u_j'^{-1})\prod_{j\in\Omega^*\cup V_t^*}(u_j')g^{\sum_{j\in\Omega^*\cup V_t^*}\sum_{\delta=1}^{k_j}\theta_{j,\delta}c_{j,\delta,\xi}}\right)^c=\left(u_0'\cdot\prod_{j\in\Omega^*\cup V_t^*}\left(u_j'\cdot\prod_{\delta=0}^{k_j}u_{j,\delta}^{c_{j,\delta,\xi}}\right)\right)^c \tag{16}$$

最后,针对同样的 c,可以得到:

$$e(g,h)^{x^{l+1}}\cdot e(y_1,h^{\alpha'})=(e(y_1,y_l)\cdot e(y_1,g^{\alpha'}))^c=e(y_1,y_l\cdot g^{\alpha'})^c=e(g_1,g_2)^c \tag{17}$$

因此,对比上述 CT 中的未知元素 T,对于选定的属性集Ω^*,当 $T=e(g,h)^{x^{l+1}}$ 时,$\mathcal{B}$可以构造 m_β的有效密文:

$$CT=\left(m_\beta\cdot e(g_1,g_2)^c,g^c,\left(u_0'\cdot\prod_{j\in\Omega^*\cup V_t^*}\left(u_j'\cdot\prod_{\delta=0}^{k_j}u_{j,\delta}^{c_{j,\delta,\xi}}\right)\right)^c\right) \tag{18}$$

当 T 是随机数时,则 CT 是一个随机信息的加密.

- 查找阶段 2

与查找阶段 1 类同,模拟器$\mathcal{B}$相应的响应敌手$\mathcal{A}$的查询.

- 猜测

最终,敌手$\mathcal{A}$输出猜测的β',若$\beta=\beta'$模拟器$\mathcal{B}$输出 1,即,猜测 $T=e(g,h)^{x^{l+1}}$;否则,模拟器$\mathcal{B}$输出 0,表明它认为 T 是 G_2 中的随机元素.

- 概率分析

上述挑战-应答游戏成功,即,挑战者解决判定性 l-BDHE 问题的成功概率分析如下.

(1) 如果模拟器$\mathcal{B}$的输出为 1,即 $T=e(g,h)^{x^{l+1}}$,挑战密文 CT 是对 m_β的有效密文,敌手$\mathcal{A}$的环境被完美模拟,

因此,敌手有ε概率成功解密,$\left|\Pr[\beta=\beta']-\frac{1}{2}\right|\geq\varepsilon$;

(2) 如果模拟器$\mathcal{B}$的输出为 0,即 T 是 G_2 中的随即元素,敌手.$\mathcal{A}$不能得到有关β的任何有效信息,因此,敌手有ε概率成功解密,$\left|\Pr[\beta\neq\beta']-\frac{1}{2}\right|=0$.

综上,模拟器$\mathcal{B}$可以以不可忽略的优势解决 *l-BDHE* 问题:

$$|\Pr[\mathcal{B}=1|T=e(g,h)^{x^{l+1}}]-\Pr[\mathcal{B}=0|T\in Random]|\geq\left|\left(\frac{1}{2}+\varepsilon\right)-\frac{1}{2}\right|=\varepsilon \tag{19}$$

证毕. □

6.3 效率分析

本节中,我们将分别将 CGAC 算法与已有的使用层次化结构的属性基加密算法、已有的定长密文属性加密算法进行效率和安全性对比.其中,计算开销主要由加密运算和解密运算组成,而通信开销以及存储空间长度需要分析密文长度、私钥空间.除此之外,本文还给出了方案之间的访问控制结构和安全性对比,具体见表 2.

Table 2 Comparison with the proposed scheme and existing schemes

表 2 本文算法与已有方案之间的对比

方案	密文长度	最长私钥长度	加密	解密	安全	特点	控制结构
AL[25]	$2G_1+G_2$	$(2n+5)G_1$	$4E$	$3P+(n-1)E$	CPA	定长	固定
HLR[26]	$2G_1+G_2$	$(2n-1)G_1$	$(n+t+1)E$	$3P+O(t^2)E$	CPA	定长	$(t-n)$阈值
HASBE[12]	$(2n+3)G_1+G_2$	$n(l+n+3)G_1$	$(2n+4)E$	$(2P+E)n$	CPA	层次	与或门
HABE[27]	$(n+1)G_1+G_2$	$(l+1)nG_1$	$(n+2)E$	$2n(2P+E)$	CPA	层次	$(t-n)$阈值
本文 CGAC	$2G_1+G_2$	$((2+l)(n+1)+3)G_1$	$3E$	$2P+2nE$	CPA	层次+定长	$(t-n)$阈值

在上述性能比较中,加密、解密分别表示加密解密的计算复杂度,密文长度和最长私钥长度表示存储的空间复杂度,安全表示论文中所给出的方案安全性证明,特点及控制结构分别表示论文方案的特点和其访问控制的结构.设 P 表示最耗时的一个双线性对运算时间,E 表示相对次耗时的一个幂指运算时间,G_1和 G_2分别表示所在群中元素的长度.同时,在表格中我们定义空间的所有属性个数为 n,层次属性的最大深度为 l,用户可以解密数据的属性个数阈值 t,则$(t-n)$阈值表示访问控制结构为 n 个属性中存在 t 个属性满足即可访问.

通过对比本文所提出的方案,相较于文献[26,27],其密文长度、加解密运算所需的双线性对运算与系统属性个数相关,本文方案所带来的计算开销及存储空间都将大大缩小.同时,与其他定长方案[25,26]相比,本文的访问控制结构比较灵活,同时,加解密的计算开销较小.此外,本文方案同样满足 CPA 安全性.但需要指出的是:与其他方案相比,由于本方案将属性空间相关参数嵌入私钥,使计算后的密文空间达到固定长度,因而产生的私钥长度较大,带来相应的存储空间开销.尽管如此,本方案通过牺牲用户的私钥存储空间,达到了降低访问通信带宽开销、提高计算效率的目的.因为在实际运用的网络环境中,系统间通信带宽通常成为制约的瓶颈,提高维护通信带宽的成本代价又非常昂贵,而相对的存储的增加和维护十分容易,以 SS512 的群为例,100 个分类属性和最大深度为 30 的私钥存储开销最大仅为 20M,因此,本文方案是可行的.

7 系统实现与结果分析

7.1 系统设计

Openstack Swift 是一种典型并且开源的对象存储,其作为云基础服务 Openstack 的核心子项目之一,为其他子项目提供存储服务.Swift 利用便宜的基础硬件存储,通过软件层面的算法,引入一致性散列技术实现数据冗余性和均衡分布,同时支持多租户模式、容器和对象读写操作,适用于存储互联网应用场景下的非结构化数据.

7.1.1 访问控制中间件

Swift 利用 Proxy Server 模块对外提供标准的基于 HTTP 的 REST 接口,对账户、容器和对象进行 CRUD

操作.而在 Swift 内部,它利用 Python 的 WSGI 模型(Web services gateway interface)和 Python Paste 框架构建,根据 Pipeline 配置中的调用顺序,依次通过中间件处理 Swift 的请求链.中间件类似于洋葱结构包裹在 Swift 核心模块之上,请求会依次通过各个加载中间件,我们可以定制自己的中间件组件,处理进出中间件的响应请求,在到达核心 Swift 之前修改其中的请求数据,或者直接交给下层中间件处理,也可以在本层直接响应结果.

如图 4 所示,本文的方案是将访问控制策略通过 Swift 中间件实现,用户通过 REST 接口向 proxyserver 提交访问请求.其后,请求被交给访问控制中间件处理.访问控制中间件从请求中获取 Token,从而根据第 4.2 节中访问控制模型中的流程验证用户身份并获取主体标记;同时,从请求中获取客体标记,利用第 4.1 节访问控制规则 1 和规则 2 进行判定:如果不满足规则要求,则直接返回响应状态码“403 Forbidden”;否则,交给下层中间件进一步进行下个逻辑的处理.

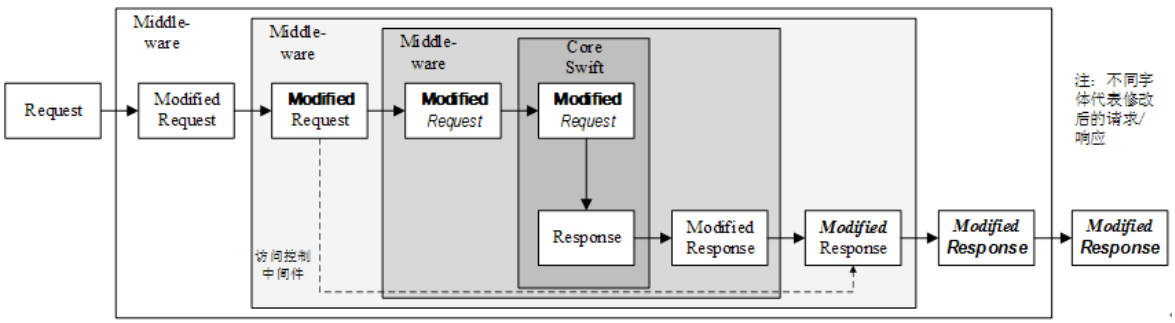


Fig.4 Process flows of Swift middle wares

图 4 Swift 中间件处理流程

7.1.2 系统实现

本节主要基于 Openstack Swift 介绍第 4.2 节中访问控制模型中的对象云存储的具体实现,分别对模型中的 3 个参与者进行详细描述.

(1) 对于访问控制模型中的云存储服务 CSP,主要实现对象存储如图 5 所示,其中:访问决策模块进行访问控制策略的判定由 Swift 中间件实现,其编写规则参照 WSGI 标准;标记解析模块负责主客体标记的解析,通过 Token 从 Keystone 获取用户的主体标记,从元数据管理模块获取客体的标记,交给访问控制决策模块分析;当用户满足访问控制规则,由访问决策模块交给后端控制模块 Controller,通过一致性散列技术完成相应的对象数据或元数据的 CURD 操作.

需要特别指出的是:由于 Swift 的元数据最大长度默认为 256B,元数据越短,服务器可以缓存更多的元数据,保持较高的响应速度;而当元数据长度增长时,会消耗大量硬件计算资源和存储资源,使云存储服务的性能急剧降低,因此,固定长度的密文可以作为元数据存储.当用户进行上传操作时,Proxy Server 通过 REST 接口获取到用户 POST 的数据客体标记,及通过 CGAC 算法加密的固定长度的密文后,将上述信息作为对象数据密文的元数据存储.

另外,由于采用无状态 REST 协议,代理服务 proxy server 和存储结点都可以横向扩展来实现负载均衡,避免单点故障,此时,访问控制中间件都需要配置并加载在 proxy server 的 pipeline 中.同时还需要修改 Swift 的 Cache 集群结构,利用一致性散列分配地址空间,缓存 Token 的验证和主体标记.

(2) 访问控制模型中的主从 KGC 则由 Keystone 身份认证模块负责实现,进行用户的身份认证管理及 CGAC 算法中的 *Setup*,*KeyGen* 和 *Delegate* 算法实现,完成 CGAC 系统的初始化和用户、子 KGC 的私钥产生、授权等,具体实现通过修改 Openstack 的认证组件 Keystone 来完成;认证模块与访问控制决策模块的交互,包括 Token 同步等,通过 Fernet 机制进行.

(3) 访问控制模型中的用户 User 端,需要实现的模块主要包括提供用户身份信息进行认证并获取用户私

钥进行存储,通过调用对象存储接口实现对象数据的上传、下载,以及实现 CGAC 算法中的 *Encrypt* 和 *Decrypt* 进行用户数据的加密或云数据的解密等操作.

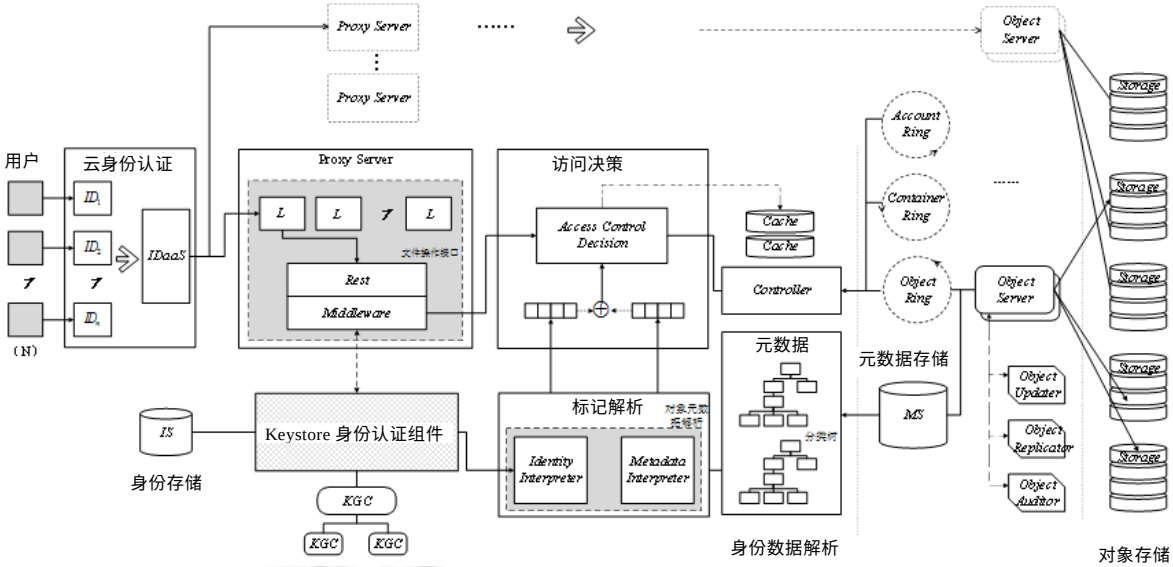


Fig.5 Implementations of access control system for objectstorage

图 5 对象云存储访问控制系统实现

7.2 实验环境

根据上述的系统结构,在 2.7GHZ inteli5 CPU,4GB DDR3RAM 的电脑上通过 VMWARE 建立一个 4 核 CPU 和 4GB 内存的虚拟机,运行 Debian Linux8.2jessie 系统.并在该系统中建立 Swift 对象存储,其中,其配置上只有 1 个代理节点和利用本地回环建立的 4 个存储结点.而在用户客户端的加解密模块和用户私钥生成模块的实现中,我们使用了 Charm-Crypto[28]作为双线性密码计算的库,并选择群中元素 g 的大小$|g|$为 512 比特,利用 python 语言实现了对象云存储中分类分级数据的访问控制系统.限于篇幅,本文系统的部分核心实现可在 GITHUB[29] 上获取.

7.3 结果分析

通过在上述环境中实现了图 5 架构下的整个系统,并获取了一些系统运行截图:

如图 6 左所示,展示用户通过 REST 接口获取 Token 的运行图;而图 6 右则展示了当用户请求下载数据时,由于主体标记中的安全级别小于客体的安全级别,下载失败.

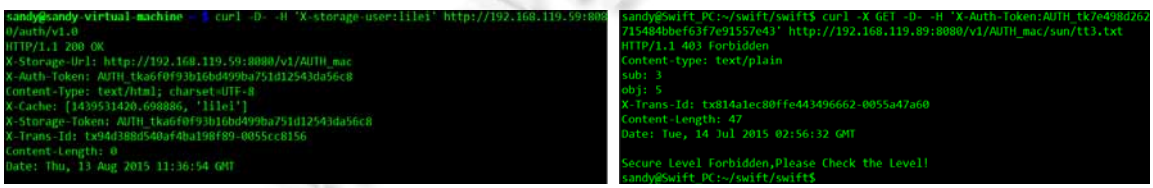


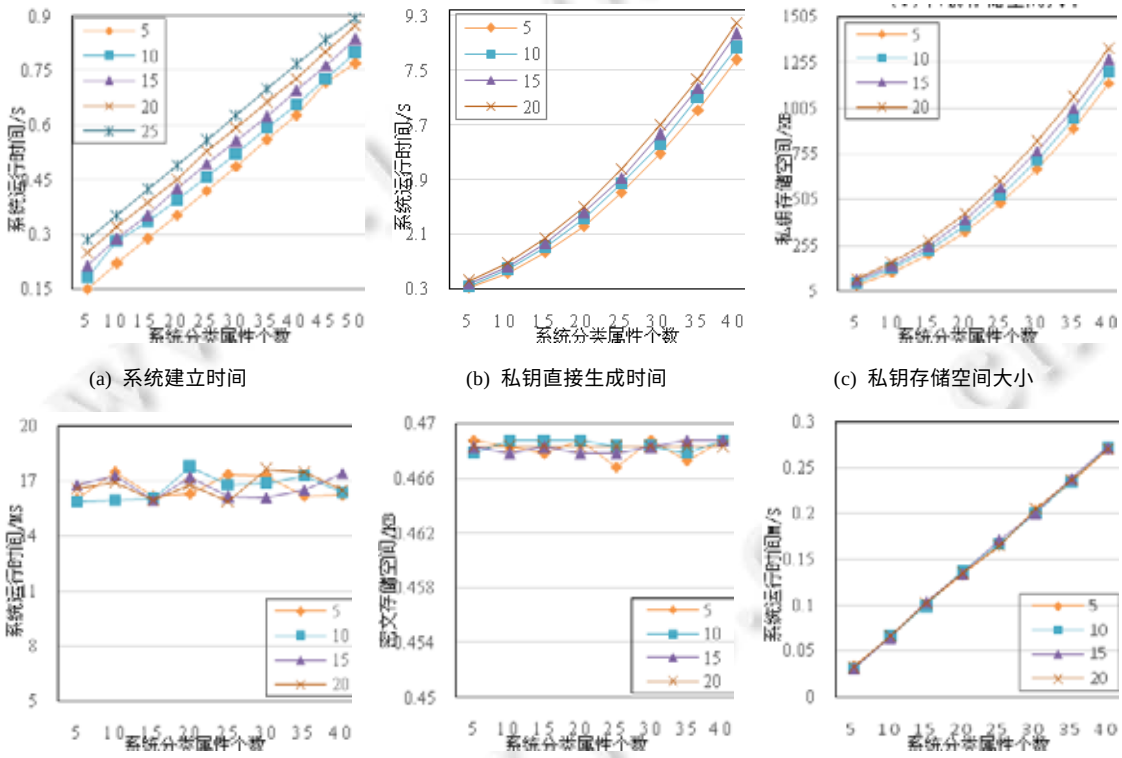
Fig.6 Screen shots of system

图 6 系统运行截图

通过对整个对象存储访问控制系统进行测试,得出下面的整个系统运行结果图,系统运行时间包含了 CGAC 算法执行时间、加密上传或下载解密、访问控制决策时间、对象存储检索时间、元数据管理时间等系

统操作时间.由于在本地测试,统计时间不包含网络延时.时间结果也反映了用户客户端数据加密解密效率,即 CGAC 算法的 *Encrypt* 和 *Decrypt* 效率.同时,私钥生成时间也反映了分布式 keystone 的 *Keygen* 执行效率.

随机生成 t=2 的一个分类拓扑进行模拟访问,通过实际实验的结果,不同的曲线表示不同的分类树深度在系统分类属性个数下的运行效果.图 7(a)表明,系统建立时间与系统属性个数及分类树深度成正比.图 7(b)表明:私钥的生成时间与系统属性个数及分类树深度同样成正比关系,且增长速度变快.图 7(c)表明:私钥的存储空间与系统属性个数及分类树深度同样成正比关系,且与私钥生成时间的曲线相符.图 7(d)表明:加密算法的执行时间与系统属性个数及分类树深度相关性不大,且波动较小.还可以发现,加密时间很短平均只有 17ms,因此加密算法的效率是很高的.图 7(e)表明:对称密钥密文存储空间与系统属性个数及最大分类树深度相关性不大,且波动较小,平均对称密钥密文长度为 0.47KB,因此只要设置对象存储中单个元数据的大小大于 0.5KB 即可.图 7(f)表明:解密算法的执行时间,由于参与运算的分类树深度是固定的,解密时间所以与分类树深度相关性不大,而与系统属性个数程正相关性.



(a) 系统建立时间 (b) 私钥直接生成时间 (c) 私钥存储空间大小

(d) 加密算法执行时间 (e) 密文存储空间大小 (f) 解密算法执行时间

Fig.7 Run time and communications of system

图 7 系统运行时间及通信量

通过实际运行结果的效率对比图,发现其结果与算法分析中相符,随着属性个数的增多,私钥的存储空间大将会增长很快.但是这种开销在存储廉价的现代是可以忽略的,因而,本文提出的 CGAC 可以将对称密钥加解密计算量和空间限定,使其更好的与对象云存储相结合,实现了对分类分级特征对象数据的细粒度访问控制.在真实云计算环境中,用户私钥开销可以接受,且当用户数目和系统属性个数增加时,更能体现本方案优势.此外,现有的 ABE 密文访问控制系统中都存在访问权限的更改,包括策略和属性变化时,尤其是用户属性的撤销设计难度大的难题,通常需要进行重加密,导致效率不高.CGAC 算法虽然同样具有以上问题,但是由于对象存储的场景下,存储的多为图片音频等静态数据,更新的频率很小,因此此处的性能损耗同样是在可以接受的范围内.

8 结束语

在云计算越来越普及的环境下,云存储利用网络对存储资源整合利用所面临的数据安全问题越来越多.本文针对分类分级特点的对象存储服务,提出了一套事实可行的访问控制方案和模型;同时,借助 ABE 机制,设计出一种可靠的基于分类分级属性的属性加密算法.该算法将强制访问控制、定长密文的属性加密、对象存储与分类分级特性的优势相结合,不仅提高了数据的安全性,解决了细粒度访问控制问题,同时使得计算开销和通信开销大大减少,提高了系统效率.本文同时给出了基于 OpenstackSwift 对象存储的具体实现,验证了本方案的可行性.在下一步的工作中,将研究更高效的算法降低系统复杂度,同时对阈值访问控制结构进行扩展;另外,研究基于代理重加密的撤销机制,降低撤销时的开销.

References:

[1] Factor M, Meth K, Naor D, Rodeh O, Satran J. Object storage: The future building block for storage systems. In: Proc. of the Local to Global Data Interoperability-Challenges and Technologies. IEEE, 2005. 119−123. [doi: 10.1109/LGDI.2005.1612479]

[2] Mesnier M, Ganger GR, Riedel E. Object-Based storage. IEEE Communications Magazine, 2003,41(8):84−90. [doi: 10.1109/MCOM.2003.1222722]

[3] Committee AIT. Project t10/1355-d working draft: Information technology—SCSI objectbased storage device commands. 2004.

[4] Arnold J. OpenStack Swift: Using, Administering, and Developing for Swift Object Storage. O'Reilly Media, Inc., 2014.

[5] Hamlen K, Kantarcioglu M, Khan L, Thuraisingham B. Security issues for cloud computing. International Journal of Information Security and Privacy 2010,4(2):39−51.

[6] Wang YD, Yang JH, Xu C, Ling X, Yang Y. Survey on access control technologies for cloud computing. Ruan Jian Xue Bao/Journal of Software, 2015,26(5):1129−1150 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4820.htm [doi: 10.13328/j.cnki.jos.004820]

[7] Bethencourt J, Sahai A, Waters B. Ciphertext-Policy attribute-based encryption. In: Proc. of the IEEE Symp. on Security and Privacy (SP 2007). IEEE, 2007. 321−334. [doi: 10.1109/SP.2007.11]

[8] Bell DE, Padula LJL. Secure computer system: Unified exposition and multics interpretation. In: Proc. of the Secure Computer System Unified Exposition & Multics Interpretation. 1976. 161.

[9] Shen C. Application analysis of BLP model in cloud storage. Computer & Digital Engineering, 2012,40:65−66 (in Chinese with English abstract). [doi: 10.3969/j.issn.1672-9722.2012.06.021]

[10] Lin GY, He S, Huang H, Wu JY, Wei C. Access control security model based on behavior in cloud computing environment. Journal on Communications, 2012,33(3):59−66 (in Chinese with English abstract).

[11] Horwitz J, Lynn B. Toward hierarchical identity-based encryption. In: Proc. of the Advances in Cryptology—EUROCRYPT. Springer-Verlag, 2002. 466−481. [doi: 10.1007/3-540-46035-7_31]

[12] Wan Z, Liu JE, Deng RH. HASBE: A hierarchical attribute-based solution for flexible and scalable access control in cloud computing. IEEE Trans. on Information Forensics and Security, 2012,7:743−754. [doi: 10.1109/TIFS.2011.2172209]

[13] Deng H, Wu Q, Qin B, Domingo-Ferrer J, Zhang L, Liu J, Shi WC. Ciphertext-Policy hierarchical attribute-based encryption with short ciphertexts. Information Sciences, 2014,275:370−384. [doi: 10.1016/j.ins.2014.01.035]

[14] You L, Wang L. Hierarchical authority key-policy attribute-based encryption. In: Proc. of the 2015 IEEE 16th Int'l Conf. on Communication Technology (ICCT). IEEE, 2015. 868−872. [doi: 10.1109/ICCT.2015.7399963]

[15] Wang S, Zhou J, Liu JK, Yu J, Chen J, Xie W. An efficient file hierarchy attribute-based encryption scheme in cloud computing. IEEE Trans. on Information Forensics and Security, 2016,11:1265−1277. [doi: 10.1109/TIFS.2016.2523941]

[16] Liu Z, Yan H, Lin Z, Xu L. An improved cloud data sharing scheme with hierarchical attribute structure. Journal of Universal Computerence, 2015,21(3): 454−472. [doi: 10.3217/jucs-021-03-0454]

[17] Ge A, Zhang R, Chen C, Ma C, Zhang Z. Threshold ciphertext policy attribute-based encryption with constant size ciphertexts. In: Proc. of the Information Security and Privacy. Springer-Verlag, 2012. 336−349. [doi: 10.1007/978-3-642-31448-3_25]

[18] Zhang XC, Yang G. Attribute-Based access control model with constant-size ciphertext in Hadoop cloud environment. Computer Engineering and Applications, 2015,51(23):87−93 (in Chinese with English abstract). [doi: 10.3778/j.issn.1002-8331.1311-0372]

[19] Biswas P, Patwa F, Sandhu R. Content level access control for openstack swift storage. In: Proc. of the 5th ACM Conf. on Data and Application Security and Privacy. ACM Press, 2015. 123−126. [doi: 10.1145/2699026.2699124]

[20] Boneh D. Identity-Based encryption from the Weil pairing. In: Proc. of the Advances in Cryptology—CRYPTO 2001. Springer-Verlag, 2001. 213−229. [doi: 10.1007/3-540-44647-8_13]

[21] Boneh D, Boyen X, Goh EJ. Hierarchical identity based encryption with constant size ciphertext. In: Proc. of the Annual Int'l Conf. on the Theory and Applications of Cryptographic Techniques. Springer-Verlag, 2005. 440−456. [doi: 10.1007/11426639_26]

[22] Ran C, Halevi S, Katz J. Chosen-Ciphertext security from identity-based encryption. Siam Journal on Computing, 2007,36: 1301−1328.

[23] Fujisaki E, Okamoto T. Secure integration of asymmetric and symmetric encryption schemes. Journal of Cryptology, 2013,26: 80−101. [doi: 10.1007/s00145-011-9114-1]

[24] Shamir A. How to share a secret. Communications of the ACM, 1979,22:612−613. [doi: 10.1145/359168.359176]

[25] Agrawal S, Freeman DM, Vaikuntanathan V. Functional encryption for inner product predicates from learning with errors. In: Proc. of the Advances in Cryptology—ASIACRYPT 2011, Int'l Conf. on the Theory and Application of Cryptology and Information Security. Seoul, 2011. 21−40. [doi: 10.1007/978-3-642-25385-0_2]

[26] Herranz J, Laguillaumie F, Ràfols C. Constant size ciphertexts in threshold attribute-based encryption. In: Proc. of the Int'l Conf. on Practice and Theory in Public Key Cryptography. 2010. 19−34. [doi: 10.1007/978-3-642-13013-7_2]

[27] Li J, Wang Q, Wang C, Ren K. Enhancing attribute-based encryption with attribute hierarchy. Mobile Networks and Applications, 2011,16:553−561. [doi: 10.1007/s11036-010-0233-y]

[28] Akinyele JA, Green M, Rubin A. Charm: A framework for rapidly prototyping cryptosystems. Cryptology ePrint Archive, Report. 2011/617. 2011.

[29] Yang T. Sample code of "an access control mechanism for classified and graded object storage in cloud computing". 2016. https://github.com/hbhdytf

附中文参考文献:

[6] 王于丁,杨家海,徐聪,凌晓,杨洋.云计算访问控制技术研究综述.软件学报,2015,26(5):1129−1150. http://www.jos.org.cn/1000-9825/4820.htm [doi: 10.13328/j.cnki.jos.004820]

[9] 沈承东,严明向.BLP 模型在云存储中应用分析.计算机与数字工程,2012,40:65−66. [doi: 10.3969/j.issn.1672-9722.2012.06.021]

[10] 林果园,贺珊,黄皓,等.基于行为的云计算访问控制安全模型.通信学报,2012,33(3):59−66.

[18] 张欣晨,杨庚.Hadoop 环境中基于属性和定长密文的访问控制方法.计算机工程与应用,2015,51(23):87−93. [doi: 10.3778/j.issn.1002-8331.1311-0372]

杨腾飞(1990－),男,河北邯郸人,博士生,主要研究领域为云计算安全,网络与系统安全.

田雪(1986－),女,助理研究员,主要研究领域为云计算安全.

申培松(1993－),男,博士生,主要研究领域为云计算安全,系统安全.

冯荣权(1966－),男,博士,教授,博士生导师,主要研究领域为密码学,信息安全.