软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2016,27(6):1432−1450 [doi: 10.13328/j.cnki.jos.005003]

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

支持策略隐藏的加密云存储访问控制机制*

雷 蕾[1,2,3], 蔡权伟[1,2], 荆继武[1,2], 林璟锵[1,2], 王 展[1,2], 陈 波[4]

[1](中国科学院 信息工程研究所,北京 100093)

[2](中国科学院 数据与通信保护研究教育中心,北京 100093)

[3](中国科学院大学,北京 100049)

[4](College of Information Sciences and Technology, The Pennsylvania State University, University Park, USA)

通信作者: 蔡权伟, E-mail: qwcai@is.ac.cn

摘 要: 使用密码技术对云存储数据实施机密性保护和访问控制,是当前云计算安全研究的重要内容.选择加密(selective encryption)技术根据访问控制策略产生密钥推导图来分发密钥,在保证云存储数据机密性和细粒度访问控制的前提下,具有简化文件存储加密、系统密钥量少的优势.然而,已有选择加密方案需要完全或部分地公开访问控制策略,以用于密钥推导;该信息反映了用户/文件之间的授权访问关系,泄露了用户隐私.基于现有的研究工作,提出一种访问控制策略隐藏机制,在支持加密云存储数据的细粒度访问控制和高效密钥分发的前提下,能更好地隐藏访问控制策略信息,而且在密钥获取计算速度上有明显优势.

关键词: 云存储;访问控制;策略隐藏;选择加密;密钥分发

中图法分类号: TP309

中文引用格式: 雷蕾,蔡权伟,荆继武,林璟锵,王展,陈波.支持策略隐藏的加密云存储访问控制机制.软件学报,2016,27(6):1432−1450. http://www.jos.org.cn/1000-9825/5003.htm

英文引用格式: Lei L, Cai QW, Jing JW, Lin JQ, Wang Z, Chen B. Enforcing access controls on encrypted cloud storage with policy hiding. Ruan Jian Xue Bao/Journal of Software, 2016,27(6):1432−1450 (in Chinese). http://www.jos.org.cn/1000-9825/5003.htm

Enforcing Access Controls on Encrypted Cloud Storage with Policy Hiding

LEI Lei[1,2,3], CAI Quan-Wei[1,2], JING Ji-Wu[1,2], LIN Jing-Qiang[1,2], WANG Zhan[1,2], CHEN Bo[4]

[1](Institute of Information Engineering, The Chinese Academy of Sciences, Beijing 100093, China)

[2](Data Assurance and Communication Security Research Center, The Chinese Academy of Sciences, Beijing 100093, China)

[3](University of Chinese Academy of Sciences, Beijing 100049, China)

[4](College of Information Sciences and Technology, The Pennsylvania State University, University Park, USA)

Abstract: Enforcing access controls on cloud storage by cryptography is an important topic of cloud security. Based on access control policies, selective encryption builds key derivation graphs to distribute symmetric keys among users. Selective encryption can ensure the confidentiality and fine-grained access control of cloud storage data, while simplifying data encryption procedure and reducing the total number of keys. However, the existing selective encryption solutions have to fully or at least partially disclose the access control policies. This policy information unfortunately, is usually related to the authorization relation between users and files, leading to privacy leakage.

* 基金项目: 国家重点基础研究发展计划(973)(2014CB340603); 国家高技术研究发展计划(863)(2013AA01A214); 中国科学院战略性先导专项(XDA06010702)

Foundation item: National Program on Key Basic Research Project of China (973) (014CB340603); National High-Tech R&D Program of China (863) (2013AA01A214); Strategy Pilot Project of Chinese Academy of Sciences (XDA06010702)

收稿时间: 2015-08-15; 修改时间: 2015-10-09; 采用时间: 2015-12-05; jos 在线出版时间: 2016-01-21

CNKI 网络优先出版: 2016-01-25 11:09:30, http://www.cnki.net/kcms/detail/11.2560.TP.20160125.1109.001.html

This work significantly improves the existing policy-hiding schemes (of selective encryption) with much less privacy leakage and much faster key derivation, while supporting fine-grained access control on encrypted cloud storage.

Key words: cloud storage; access control; policy hiding; selective encryption; key distribution

云存储是当前最为常见、最受欢迎的云计算服务.目前,众多大型云服务提供商(cloud service provider)都提供云存储服务,例如亚马逊的 S3[1]、苹果的 iCloud[2]、百度的百度云[3]、微软的 Windows Azure[4]、金山的金山云[5]等.通过使用云存储网络服务,用户不需要维护自己的存储设备,即可以随时随地访问自己的私有数据.艾媒咨询 2014 年 12 月发布的《中国个人云存储行业及用户行为研究报告》显示,61.9%的中国网民使用过云存储产品[6].

云存储服务以其使用方便、费用低廉等特点得到了广泛使用,然而一系列相关安全事件的发生引起了人们对云存储数据安全的高度关注,特别是数据机密性和有效访问控制.例如,2014 年发生的苹果 iCloud 泄密门事件导致女星隐私照片泄露[7].

密码算法是实现云存储数据机密性的重要技术.选择加密(selective encryption)[8,9]是一种适用于云存储服务的访问控制机制,具有细粒度访问控制、密钥管理计算开销小、密钥分发效率高等特点.选择加密采用不同的对称密钥加密不同的文件,其中,具有相同访问用户集合的文件采用同一对称密钥加密.共享用户一般只需保存一个对称密钥作为用户密钥.选择加密根据访问控制策略生成的密钥推导图进行密钥分发.密钥推导图一般由若干顶点和若干有向边组成.每条有向边连接两个顶点,表示出发顶点的顶点密钥可推出终端顶点的顶点密钥.数据拥有者将每个共享用户和每个文件的访问用户集合视为密钥推导图中的一个顶点,利用访问用户集合的包含关系生成密钥推导图(详见第 1.1 节).为将密钥推导图中的密钥推导关系转变为用于密钥分发的公开信息,数据拥有者首先为每个密钥分配一个标签,之后为每条有向边生成一个对应的令牌.一般,令牌包括 3 个部分:密文、密文的解密密钥的标签(后简称解密标签)和解密密文后可获取的密钥的标签(后简称获取标签).为让共享用户快速找到获取目标密钥的令牌路径,数据拥有者还将生成一个用户密钥标签列表和文件解密密钥标签列表.最后数据拥有者将用户密钥标签列表、文件解密密钥标签列表和令牌列表作为公开信息存储在云存储服务器,使共享用户可根据其用户密钥和公开信息推导出其访问权限范围内的文件的解密密钥.

然而,由于选择加密机制的公开信息可被各方获取,攻击者可轻易利用公开信息恢复出密钥推导图,从而得到数据拥有者的访问控制策略.这也成了选择加密尚未解决的问题,即需公开访问控制策略才能实现对加密数据的访问控制.

访问控制策略反映了用户/文件之间的关系,公开该信息会导致敏感信息泄露、带来安全隐患.例如,数据拥有者将文件共享给某一用户,表示二者之间有业务往来,具有明显的隐私信息.更进一步地,如果数据拥有者是单位管理员,根据公开的访问控制策略,攻击者可大致推断出单位的员工总数、了解各员工的数据访问权限,进而可推断各员工在单位中的地位和单位组织架构等.此外,根据公开的访问控制策略信息,攻击者还可推断文件的重要性程度,从而选择攻击最重要的文件,使攻击效益最高.另外,采用相同密钥加密的文件将对应同一解密密钥标签,攻击者可据此推断出那些文件具有同一属性,泄露文件的权限信息.

在不影响对加密数据访问控制的前提下,文献[10]提出了一个旨在隐藏访问控制策略的选择加密方案——本文称其为 PCSP 方案.该方案工作原理是:在得到密钥推导图后,PCSP 方案运用标号设定算法[11]将顶点的可达性信息标记为权限区间,之后将顶点的权限区间通过加密的方式隐藏在公开信息中(详见第 1.2 节).其效果相当于隐藏了选择加密初始方案令牌中的获取标签.PCSP 方案存在如下问题:(1) 该方案并未有效实现策略隐藏.若数据拥有者是单位管理员,攻击者仍可根据公开信息获取部分访问控制策略,可大致推断出单位的员工总数、部分员工的数据访问权限.(2) 该方案并未解决采用相同密钥加密的文件对应同一解密密钥标签问题.(3) 该方案的密钥推导效率低.

本文致力于研究更安全更高效的支持访问控制策略隐藏的新方案.在新方案中,我们为每个令牌分配一个标签,不再为密钥分配标签.令牌标签将采用本文提出的一种特殊的计算方法计算得到(详见第 3.1 节).通过引入令牌标签既实现了访问控制策略的隐藏,又实现了目标密钥的快速获取.此外,本文还通过引入加密区间和文件

序列号,有效地隐藏了文件的权限信息.攻击者将不能从公开信息中获取访问控制策略的相关信息.本文主要贡献是:(1) 在文献[10]的基础上,本文提出了一种新的加密云存储访问控制方案.在实现针对加密数据的细粒度访问控制的前提下,新方案可有效支持访问控制策略的隐藏.(2) 与 PCSP 方案相比,本文提出的新方案能够实现高效的密钥分发.

本文第 1 节介绍背景工作并指出其存在的问题.第 2 节给出本文方案的服务模型和安全假设.第 3 节介绍本文方案的主要思想、具体算法并给出实例.第 4 节分析本文方案的安全性和性能.第 5 节介绍加密云存储访问控制的相关研究进展.第 6 节为总结.

1 背景和存在问题

1.1 选择加密

选择加密[8,9]的基本模型是:数据拥有者将文件 f 采用对称加密算法加密后存储在云存储服务器上,然后通过对称密钥的分发来实现用户群之间的共享和访问控制,即某用户 u 对 f 有访问权限,则数据拥有者通过密钥分发机制将 f 的解密密钥分发给用户 u.在本文中,每一个共享用户只拥有一个对称密钥作为其用户密钥.

访问控制策略即数据拥有者对每个共享用户的授权信息.访问控制策略的定义如下:

定义 1(访问控制策略). 令 U 和 F 分别为系统中的用户集合和文件集合,u 和 f 分别代表一个用户和一个文件.令 $p=\langle u,f\rangle$ 表示云数据拥有者允许用户 u 访问文件 f 的授权,P 为授权 p 的集合.则定义于 U 和 F 之上的访问控制策略为 $ACP=\langle U,F,P\rangle$.

为了实现对加密数据的访问控制,最简单的方法是将密钥直接分发给每一个有权限的用户,但是这将给数据拥有者带来繁重的密钥管理负担.选择加密采用对称密钥推导图的形式进行密钥分发,可有效减轻数据拥有者的密钥管理负担.选择加密方案生成用于实现访问控制的公开信息的主要步骤如下:

(1) 输入 ACP,生成密钥推导图.生成步骤如下:

- 创建顶点.将每个共享用户和每个文件的访问用户集合视为密钥推导图中的一个顶点.本文称密钥推导图中代表共享用户的顶点为用户顶点,代表文件访问用户集合的顶点为文件顶点.用户顶点密钥与用户密钥相同,用户顶点的访问用户集合只含有其所代表的用户.文件顶点密钥与文件解密密钥相同,文件顶点的访问用户集合等于文件的访问用户集合.
- 生成有向边.当顶点 v_j 的访问用户集合包含另一顶点 v_i 的访问用户集合时,则生成一条由 v_i 指向 v_j 的有向边 $e\langle v_i,v_j\rangle$.有向边 $e\langle v_i,v_j\rangle$ 表示出发顶点 v_i 的密钥可推出终端顶点 v_j 的密钥,本文称 v_i 为 v_j 的父顶点.
- 去除多余的有向边.由于每一条有向边对应一个令牌,删除多余的有向边将可有效减少令牌的数量.设顶点 v_i 拥有多个父顶点.先按照顶点的访问用户集合中的用户个数对父顶点进行排序,数量大者排前序.若用户个数相同,文件顶点排前序,用户顶点排后序.之后从排序第二的父顶点开始,检查其访问用户集合是否包含一个已检查的顶点(排序第一的父顶点默认已检查)的访问用户集合中没有的用户,如果不包含则删除当前检查的父顶点指向 v_i 的有向边,有则保留.
- 插入中间顶点,进一步减少有向边的数量.设 m 个子顶点共享 n 个父顶点,则连接这些顶点需 $m\times n$ 条有向边.通过插入一个访问用户集合为 n 个父顶点的访问用户集合并集的中间顶点,则只需生成 n 条由父顶点指向中间顶点和 m 条由中间顶点指向子顶点的有向边,即连接这些顶点只需生成 $m+n$ 条边.

(2) 生成密钥标签.在得到密钥推导图后,数据拥有者为每个密钥生成一个标签,并按照表 1 生成标签列表,其中,l_u 代表用户 u 的用户密钥标签,l_f 代表文件 f 的解密密钥标签.

(3) 生成令牌.对于密钥推导图中的每一条有向边 $e\langle v_i,v_j\rangle$,按照表 2 生成令牌,其中设 v_i 和 v_j 的密钥分别为 k_i 和 k_j,k_i 和 k_j 的标签分别为 l_i 和 l_j.

在得到用户密钥标签列表、文件解密密钥标签列表和令牌列表后,数据拥有者将这些列表作为公开信息存储在云存储服务器.

Table 1 Label format

表 1 标签格式

用户/文件	用户密钥标签/文件解密标签
u/f	l_u/l_f

Table 2 Token format

表 2 令牌格式

解密标签	获取标签	密文
l_i	l_j	$k_j \oplus hash(k_i, l_j)$

例 1(选择加密应用举例——根据访问控制策略生成用于实现访问控制的公开信息):设数据拥有者定义的访问控制策略见表 3.表 4 为由表 3 得到的文件访问用户集合列表.

Table 3 An ACP instance

表 3 访问控制策略实例

$ACP=\langle U,F,P\rangle$
$U=\{u_1,u_2,u_3,u_4,u_5,u_6\}$
$F=\{f_1,f_2,f_3,f_4,f_5,f_6,f_7\}$
$P=\{\langle u_1,f_1\rangle,\langle u_1,f_2\rangle,\langle u_2,f_1\rangle,\langle u_2,f_2\rangle,\langle u_2,f_3\rangle,\langle u_2,f_4\rangle,$ $\langle u_2,f_5\rangle,\langle u_2,f_6\rangle,\langle u_3,f_2\rangle,\langle u_3,f_3\rangle,\langle u_3,f_4\rangle,\langle u_3,f_5\rangle,$ $\langle u_3,f_6\rangle,\langle u_4,f_3\rangle,\langle u_4,f_4\rangle,\langle u_4,f_5\rangle,\langle u_4,f_6\rangle,\langle u_5,f_3\rangle,$ $\langle u_5,f_4\rangle,\langle u_5,f_7\rangle,\langle u_6,f_6\rangle,\langle u_6,f_7\rangle\}$

Table 4 Access user sets of files

表 4 文件的访问用户集合

文件	访问用户集合
f_1	$\{u_1,u_2\}$
f_2	$\{u_1,u_2,u_3\}$
f_3	$\{u_2,u_3,u_4,u_5\}$
f_4	$\{u_2,u_3,u_4,u_5\}$
f_5	$\{u_2,u_3,u_4\}$
f_6	$\{u_2,u_3,u_4,u_6\}$
f_7	$\{u_5,u_6\}$

设用户 u_1,u_2,u_3,u_4,u_5,u_6 的用户密钥分别为 k_1,k_2,k_3,k_4,k_5,k_6, 文件 $f_1,f_2,f_3,f_4,f_5,f_6,f_7$ 的解密密钥分别为 $k_7,k_8,k_9,k_9,k_{10},k_{11},,k_{12}$. f_3 和 f_4 具有相同的访问用户集合,因而二者均采用密钥 k_9 加密.根据表 3,生成的密钥推导图如图 1 所示.在图 1 中, v_1-v_6 为用户顶点, v_7-v_{12} 为文件顶点.设密钥 k_1-k_{12} 的标签分别为 l_1-l_{12}.按表 1 生成的标签列表见表 5.按表 2 生成的令牌列表见表 6.最后数据拥有者将表 5 和表 6 存储于云存储服务器.

例 2(共享用户获取目标密钥举例):设用户u_1需要解密文件f_2.用户u_1获取目标密钥步骤如下:

(1) 查询表 5,找到用户密钥的公开标签为l_1,文件f_2的解密密钥标签为l_8.

(2) 查询表 6,找到获取目标密钥的令牌路径,见表 7.

Table 5 Label list

表 5 标签列表

用户/文件	用户密钥标签/文件解密密钥标签
u_1	l_1
u_2	l_2
u_3	l_3
u_4	l_4
u_5	l_5
u_6	l_6
f_1	l_7
f_2	l_8
f_3	l_9
f_4	l_9
f_5	l_{10}
f_6	l_{11}
f_7	l_{12}

Table 6 Token list

表 6 令牌列表

解密标签	获取标签	密文
l_1	l_7	$k_7 \oplus hash(k_1, l_7)$
l_2	l_7	$k_7 \oplus hash(k_2, l_7)$
l_2	l_{10}	$k_{10} \oplus hash(k_2, l_{10})$
l_3	l_8	$k_8 \oplus hash(k_3, l_8)$
l_3	l_{10}	$k_{10} \oplus hash(k_3, l_{10})$
l_4	l_{10}	$k_{10} \oplus hash(k_4, l_{10})$
l_5	l_9	$k_9 \oplus hash(k_5, l_9)$
l_5	l_{12}	$k_{12} \oplus hash(k_5, l_{12})$
l_6	l_{11}	$k_{11} \oplus hash(k_6, l_{11})$
l_6	l_{12}	$k_{12} \oplus hash(k_6, l_{12})$
l_7	l_8	$k_8 \oplus hash(k_7, l_8)$
l_{10}	l_9	$k_9 \oplus hash(k_{10}, l_9)$
l_{10}	l_{11}	$k_{11} \oplus hash(k_{10}, l_{11})$

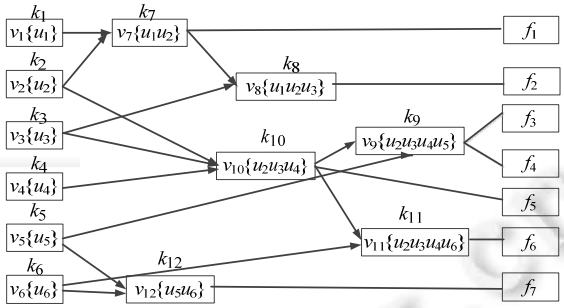

Fig.1 Key derivation graph

图 1 密钥推导图

Table 7 Derivation path for the target key

表 7 获取目标密钥的令牌路径

路径排序	解密标签	获取标签	密文
1	l_1	l_7	$k_7 \oplus hash(k_1,l_7)$
2	l_7	l_8	$k_8 \oplus hash(k_7,l_8)$

按照路径排序,首先利用用户密钥解密第 1 个令牌得到密钥 k_7,再利用 k_7解密第 2 个令牌得到目标密钥 k_8.

由于选择加密需要将用户密钥标签列表、文件解密密钥标签列表和令牌的解密标签和获取标签作为公开信息,攻击者将可轻易根据这些公开信息得到密钥推导图和访问控制策略.访问控制策略的公开将带来严重的安全威胁.若数据拥有者是单位管理员,攻击者可根据公开的访问控制策略大致推断出单位的员工总数、各员工的数据访问权限、各员工在单位中的地位、单位组织架构和文件的重要性程度.例如攻击者可根据表 5 和表 6 轻易得到访问控制策略表 3.如果数据拥有者是单位管理员,攻击者可根据表 3 推断出该单位有 6 名员工,各员工的数据访问权限见表 8.由表 8 可知,u_2的访问权限最大,在组织中具有较高的地位.

Table 8 User access privileges

表 8 员工的数据访问权限

员工	数据访问权限	员工	数据访问权限
u_1	$\{f_1, f_2\}$	u_4	$\{f_3, f_4, f_5, f_6\}$
u_2	$\{f_1, f_2, f_3, f_4, f_5, f_6\}$	u_5	$\{f_3, f_4, f_7\}$
u_3	$\{f_2, f_3, f_4, f_5, f_6\}$	u_6	$\{f_6, f_7\}$

1.2 访问控制策略的隐藏方案与不足

为应对上述挑战,Vimercati 等人在文献[10]中给出了一种用于实现访问控制策略隐藏的方案(本文称其为 PCSP 方案).在利用初始方案获取密钥推导图后,PCSP 方案首先利用标号设定算法[11]将密钥推导图中各顶点的可达性信息(即一个顶点通过图中的有向边或路径可达到的顶点集,例如图 4 中的 v_2,其可到达顶点为 v_7,v_8,v_9,v_{10},v_{11})标记为权限区间,之后将权限区间隐藏在公开信息中.PCSP 方案主要步骤如下:

(1) 运用选择加密初始方案[8,9]得到密钥推导图,为所有密钥生成标签.

(2) 运用标号设定算法[11]为密钥推导图中各顶点标记权限区间,可分为以下 5 步:

- 为密钥推导图添加一个根顶点 v_T,从根顶点v_T出发遍历图中所有顶点得到深度优先生成树 ST.
- 为每个顶点分配一个编号,从 ST 最右上角顶点出发,依次为ST上各顶点分配一个编号,同时标记顶点通过 ST 上的有向边或路径到达的顶点信息为权限区间.一个顶点可拥有多个权限区间.若一个顶点的权限区间为 $[n_1,n_2]_p$,表示该顶点可到达编号属于$[n_1,n_2]_p$的任意顶点(PCSP 方案中假定顶点可到达本身).例如,图 3 中的顶点 v_7,其权限区间为$[1,2]_p$,表示其可达到的顶点为其自身(自身编号为 2)和v_8(编号为 1).
- 去掉 T 并删除从v_T出发的有向边,标记顶点通过不在ST上的有向边或路径到达的顶点信息为权限区间.

例如图 4 中的有向边 $e\langle v_2,v_7\rangle$,表明v_2可通过$e\langle v_2,v_7\rangle$到达顶点v_7,因而可将顶点 v_7 的权限区间标记到顶点v_2的权限区间上,如图 4 所示.

• 标记有向边的权限区间.每条有向边的权限区间等于其终端顶点的权限区间.例如图 5 中的有向边$e\langle v_2,v_7\rangle$的权限区间为$[1,2]_p$,也即是顶点v_7的权限区间.

• 去除权限重叠的权限区间.如果一个顶点可通过多条路径到达另一顶点时,仅需保存其中最短的路径信息即可.当两条路径长度相同时,只需随意保存其中一条路径信息.

(3) 生成文件解密密钥标签列表、顶点密钥标签编号列表(顶点密钥标签编号等于顶点编号.例如图 3 中的顶点v_7编号为 2,则顶点密钥标签l_7的编号为 2)和令牌列表.对于密钥推导图中的每一条有向边$e\langle v_i,v_j\rangle$,按照表 9 格式生成令牌,其中设v_i和v_j的密钥分别为k_i和k_j,k_i和k_j的标签分别为l_i和l_j,$e\langle v_i,v_j\rangle$的权限区间为$[n_1,n_2]_p$.表 9 中的令牌号由数据拥有者随机生成,保证每个令牌的令牌号不同即可,其作用是区分具有相同解密标签的令牌.$E_{k_i}\{l_j,k_j\oplus h(k_i,k_j),[n_1,n_2]_p\}$表示用密钥$k_i$加密$\{l_j,k_j\oplus h(k_i,k_j),[n_1,n_2]_p\}$后得到的密文.

Table 9 Token format of PCSP

表 9 PCSP 方案令牌格式

令牌号	解密标签	密文
由数据拥有者随机生成	l_j	$E_{k_i}\{l_j,k_j\oplus h(k_i,l_j),[n_1,n_2]_p\}$

例 3(PCSP 方案应用举例——根据访问控制策略生成用于实现访问控制的公开信息):设访问控制策略与表 3 相同.经过步骤(1)后,可得到密钥推导图(如图 1 所示).步骤(2)的执行过程如图 2~图 5 所示,图 2 是为图 1 添加根顶点v_T后,从v_T出发,遍历图 1 上所有顶点得到的深度优先生成树 *ST*,图中有向实线表示在 *ST* 上的有向边,有向虚线表示不在 *ST* 上的有向边.有向边上的编号代表将该有向边添加至深度优先生成树上的序号.对图 2 中的各顶点编号并标记权限区间后,得到的密钥推导图如图 3 所示.之后,去掉v_T,并删除从v_T出发的有向边,标记顶点通过不在 *ST* 上的有向边或路径到达的顶点信息为权限区间,如图 4 所示.最后,标记各有向边的权限区间,并删除顶点权限区间后,可得到图 5.由于本例中没有权限重叠的权限区间,因而不需要执行去除权限重叠的权限区间操作.执行步骤(3),生成的文件解密密钥标签列表、密钥标签编号列表和令牌分别见表 10~表 12.最后,数据拥有者将表 10~表 12 存储于云服务器.

例 4(共享用户获取目标密钥举例):设用户u_2需要解密文件f_3.PCSP 方案假设用户密钥和用户密钥标签同时分发给共享用户(例如,可设用户密钥标签为密钥的 hash 值).故用户u_2已知其用户密钥标签为l_2.用户u_2获取目标密钥步骤如下:

(1) 查询表 10,找到文件 f_3 的解密密钥标签为 l_9.

(2) 查询表 11,找到标签 l_9 编号为 4.

(3) 解密以l_2为解密标签的令牌,直至找到权限区间包含编号 4 的令牌或目标令牌(即解密令牌后得到标签为l_9).

• 利用用户密钥k_2解密表 12 中令牌号为 2 的令牌密文,得到$\{l_7,k_7\oplus h(k_2,l_7),[1,2]_p\}$,其权限区间不包含 4,继续解密以$l_2$为解密标签的令牌.

• 利用用户密钥k_2解密表 12 中令牌号为 3 的令牌密文,得到$E_{k_2}\{l_{10},k_{10}\oplus h(k_2,l_{10}),[4,6]_p\}$,其权限区间包含编号 4,标签为 l_{10},说明以 l_{10} 为解密标签的令牌中,含有帮助获取目标密钥的令牌.利用k_2和 l_{10} 解密 $k_{10}\oplus h(k_2,l_{10})$ 得到密钥 k_{10}.

(4) 解密表 12 中以 l_{10} 为解密标签的令牌,直至找到权限区间包含编号 4 的令牌或目标令牌.

利用密钥 k_{10} 解密表 12 中令牌号为 12 的令牌密文,得到$\{l_9,k_9\oplus h(k_{10},l_9),[4,4]_p\}$.标签为$l_9$,说明该令牌为目标令牌.利用 k_{10} 和 l_9 解密 $k_9\oplus h(k_{10},l_9)$ 得到目标密钥k_9.

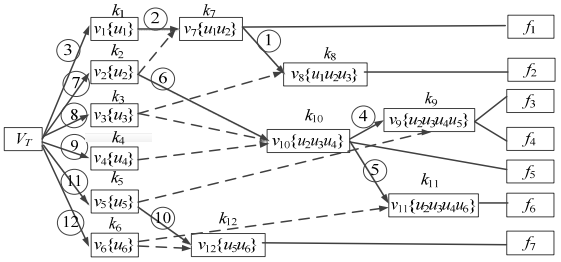

Fig.2 ST(depth-first spanning tree) of key derivation graph

图 2 密钥推导图的深度优先生成树

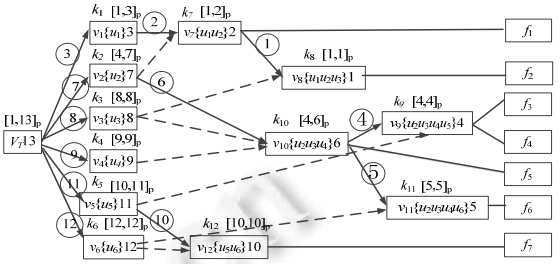

Fig.3 Marking reachability of ST as vertex privilege intervals

图 3 标记深度优先生成树上有向边的可达性信息

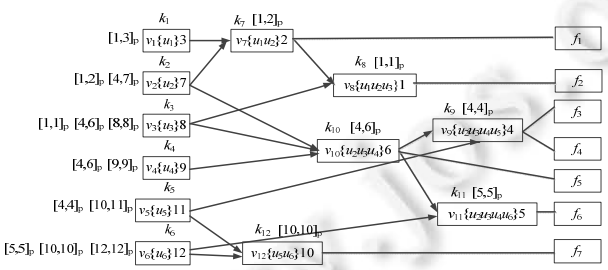

Fig.4 Marking reachability of key derivation graph as vertex privilege intervals

图 4 标记整个密钥推导图的可达性信息为顶点权限区间

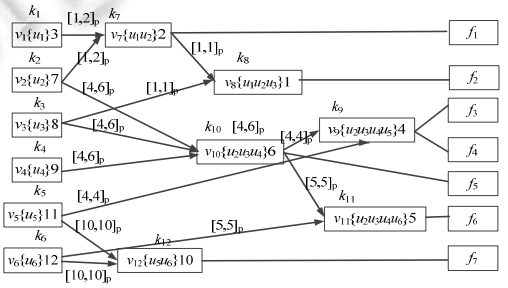

Fig.5 Marking directed edge privilege intervals

图 5 标记有向边的权限区间

Table 10 Decryption label list

表 10 文件解密标签列表

文件	解密标签
f_1	l_7
f_2	l_8
f_3	l_9
f_4	l_9
f_5	l_{10}
f_6	l_{11}
f_7	l_{12}

Table 11 Number list

表 11 标签编号列表

标签	编号
l_1	3
l_2	7
l_3	8
l_4	9
l_5	11
l_6	12
l_7	2
l_8	1
l_9	4
l_9	6
l_{10}	5
l_{11}	10

Table 12 Token list

表 12 令牌列表

令牌号	解密标签	密文
1	l_1	$E_{k_1}\{l_7, k_7 \oplus h(k_1, l_7), [1,2]_P\}$
2	l_2	$E_{k_2}\{l_7, k_7 \oplus h(k_2, l_7), [1,2]_P\}$
3	l_2	$E_{k_2}\{l_{10}, k_{10} \oplus h(k_2, l_{10}), [4,6]_P\}$
4	l_3	$E_{k_3}\{l_8, k_8 \oplus h(k_3, l_8), [1,1]_P\}$
5	l_3	$E_{k_3}\{l_{10}, k_{10} \oplus h(k_3, l_{10}), [4,6]_P\}$
6	l_4	$E_{k_4}\{l_{10}, k_{10} \oplus h(k_4, l_{10}), [4,6]_P\}$
7	l_5	$E_{k_5}\{l_9, k_9 \oplus h(k_5, l_9), [4,4]_P\}$
8	l_5	$E_{k_5}\{l_{12}, k_{12} \oplus h(k_5, l_{12}), [10,10]_P\}$
9	l_6	$E_{k_6}\{l_{11}, k_{11} \oplus h(k_6, l_{11}), [5,5]_P\}$
10	l_6	$E_{k_6}\{l_{12}, k_{12} \oplus h(k_6, l_{12}), [10,10]_P\}$
11	l_7	$E_{k_7}\{l_8, k_8 \oplus h(k_7, l_8), [1,1]_P\}$
12	l_{10}	$E_{k_{10}}\{l_9, k_9 \oplus h(k_{10}, l_9), [4,4]_P\}$
13	l_{10}	$E_{k_{10}}\{l_{11}, k_{11} \oplus h(k_{10}, l_{11}), [5,5]_P\}$

PCSP 方案在支持策略隐藏方面有所改进,然而其并未很好地解决该问题.存在以下问题:

(1) 访问控制策略的泄露.PCSP 方案并未有效地隐藏访问控制策略,攻击者仍可根据公开信息获取部分访问控制策略.若数据拥有者是单位管理员,攻击者仍可推断出单位的员工总数、员工的部分数据访问权限.例如,攻击者仍可根据顶点标号分配方法和表 10~表 12 推出密钥推导图的深度优先生成树上的有向边.另可推出员工总数为 6 人,u_1 的访问权限为 f_1 和 f_2,u_2 除能访问文件 f_3, f_4, f_5, f_6 外,至少还能访问 f_1, f_2 中的一个文件.u_3 至少能访问 $f_1, f_2, f_3, f_4, f_5, f_6$ 中的两个文件.u_4 至少能访问 $f_1, f_2, f_3, f_4, f_5, f_6$ 中的一个文件.u_5 除能访问文件 f_7 外,至少还

能访问 $f_1, f_2, f_3, f_4, f_5, f_6$ 中的一个文件,u_6 至少拥有两个文件的访问权限.

(2) 密钥推导效率低.共享用户不能快速获取推导目标密钥的令牌路径,需要解密许多不相关的令牌,具体可见例 4.

(3) 文件权限信息的泄露.由于采用相同密钥加密的文件将对应同一解密密钥标签,攻击者可据此推断出那些文件具有同一属性.例如,在表 10 中,f_3 和 f_4 的解密密钥标签相同,说明二者能够被相同的用户群所访问.

2 服务模型和安全假设

2.1 服务模型

本文方案使用了与文献[12,13]相同的服务模型,包括数据拥有者、共享用户和云服务器 3 个主体,如图 6 所示.数据拥有者将加密文件和用于实现访问控制的公开信息存储于云服务器.数据拥有者不需要始终在线,在数据上传存储到云服务器之后即可处于离线状态.数据用户可随时将存储于云服务器的加密文件和公开信息下载至本地.云服务器始终在线,随时为数据拥有者和用户提供数据下载和上传服务.

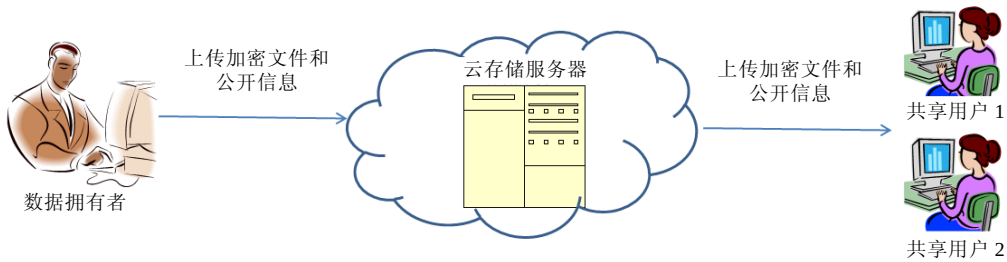

Fig.6 Service model

图 6 服务模型

2.2 安全假设

本文方案使用在云计算安全研究中广泛使用的可信且好奇(honest-but-curious)假设[8,9,12],即云服务器可信地执行数据拥有者的指令,但是会尝试获取存储数据的各种相关信息.本文假定云服务器对存储数据的明文和数据拥有者的访问控制策略、共享用户的数据访问权限和文件重要性程度(假定文件访问用户集合越小重要性程度越高)更感兴趣.此外,一些恶意用户希望能够访问不在其权限访问内的文件并尽可能多的获取其自身以外的访问控制策略.

另外,方案还基于如下假设:

(1) 所使用的密码算法是安全的;

(2) 共享用户能够保存好自己的密钥;

(3) 存在安全的带外信道用于数据拥有者与共享用户之间的对称密钥共享(例如,使用 SSL/TLS 等).

3 我们的方案

PCSP 方案面临的文件权限信息的泄露问题,其原因在于采用同一密钥加密的文件对应相同的公开标签.一种简单的解决办法是让每个文件对应不同的解密标签,然而这将导致在生成密钥推导图时将每个文件的访问用户集合视为一个不同的顶点,从而引入多余的有向边.为避免引入多余的有向边,同时又能提示每个文件的解密密钥信息,本文将引入文件序列号、加密区间和新的权限区间概念.文件序列号是每个文件在系统中的唯一标识,用于提示文件的解密密钥信息.本文将为每个文件分配一个独一无二的序列号.顶点的加密区间用于标记用该顶点的密钥加密的文件的序列号.若一个顶点的加密区间为$[s_1-s_2]_e$,表示共享用户可用该顶点的密钥解密文件序列号位于$[s_1-s_2]_e$ 区间内的文件.在本文方案中,顶点的权限区间标记该顶点密钥可推导出的密钥所加密的文件的序列号.若一个顶点的权限区间为$[s_1-s_2]_p$,表示共享用户可用该顶点的密钥推出文件序列号位于区间

$[s_1-s_2]_p$内的文件的解密密钥(本文的方案中假定顶点密钥不能推出自身).例如,在图 10 中,顶点 v_9 的密钥用于加密文件 f_3 和f_4,可为 f_3 和f_4分配连续文件序列号 3 和 4,同时将顶点 v_9 加密区间标记为[3–4],而顶点 v_9 没有子节点,即顶点 v_9 的密钥不能推出其他任何密钥,因而其权限区间为空.通过引入文件序列号、加密区间和新的权限区间概念,在没有引入多余的有向边的前提下,实现了文件权限信息的隐藏,使攻击者不能再从公开信息中推出哪些文件采用同一密钥加密,具有同一属性.

PCSP 方案存在的访问控制策略泄露问题,其原因在于:

(1) 需要将各令牌的解密密钥标签作为公开信息,攻击者可通过比对令牌列表和文件的解密密钥标签列表,推出哪些标签是用户密钥标签或中间顶点密钥标签.

(2) 需要公布标签的编号值,这使攻击者可据此推出密钥推导图的深度优先搜索生成树上的所有有向边.

此外,PCSP 方案存在的密钥推导效率低问题,其原因在于只有解密一个令牌,并获取其隐藏的权限区间后,共享用户才能判断该令牌是否属于获取目标密钥的令牌路径.本文提出了一种新的令牌格式和标签算法可同时解决上述两大问题.在本文方案中,将为每个令牌分配一个经过特殊计算的标签,不再为密钥分配标签.

3.1 令牌格式和令牌标签计算方法

设密钥推导图中一个顶点 v_0 包含有 n 个子顶点,分别为 $v_1,v_2,...,v_n$. $v_0,v_1,v_2,...,v_n$ 的顶点密钥分别为 $k_0,k_1,k_2,...,k_n$. 设 $v_1,v_2,...,v_n$ 的加密区间和权限区间见表 13.

Table 13 Encryption intervals and priviledge intervals of descendant vertices

表 13 子顶点加密区间和权限区间

子顶点	加密区间	权限区间
v_1	$[s_{11}-s_{12}]_e$	$[y_{11}-y_{12}]_{p_1},[y_{13}-y_{14}]_{p_2},...,[y_{1(2m_1-1)}-y_{1(2m_1)}]_{p_{m_1}}$
v_2	$[s_{21}-s_{22}]_e$	$[y_{21}-y_{22}]_{p_1},[y_{23}-y_{24}]_{p_2},...,[y_{2(2m_2-1)}-y_{2(2m_2)}]_{p_{m_2}}$
…	…	…
v_n	$[s_{n1}-s_{n2}]_e$	$[y_{n1}-y_{n2}]_{p_1},[y_{n3}-y_{n4}]_{p_2},...,[y_{n(2m_n-1)}-y_{n(2m_n)}]_{p_{m_n}}$

设顶点v_0的加密区间为$[s_{01}-s_{02}]_e$,按表 14 格式标记顶点v_0的加密区间和权限区间.在表 14 中,v_0的权限区间数量等于其子顶点个数,v_0的每个权限区间标记一个子顶点的加密区间和权限区间.当一个共享用户获取v_0的加密区间和权限区间后,可清楚知道该顶点的子顶点个数和各子顶点的权限信息.

Table 14 Encryption intervals and priviledge intervals of ancestor vertices

表 14 父顶点的加密区间和权限区间

顶点	加密区间	权限区间
v_0	$[s_{01}-s_{02}]_e$	$[s_{11}-s_{12},y_{11}-y_{12},y_{13}-y_{14},...,y_{1(2m_1-1)}-y_{1(2m_1)}]_{p_1},[s_{21}-s_{22},y_{21}-y_{22},y_{23}-y_{24},...,$ $y_{2(2m_2-1)}-y_{2(2m_2)}]_{p_2},...,[s_{n1}-s_{n2},y_{n1}-y_{n2},y_{n3}-y_{n4},...,y_{n(2m_n-1)}-y_{n(2m_n)}]_{p_n}$

父顶点v_0指向各子顶点的有向边 $e\langle v_0,v_1\rangle,e\langle v_0,v_2\rangle,\ldots,e\langle v_0,v_n\rangle$所对应令牌见表 15.

Table 15 Label format

表 15 令牌格式

标签	密文
$l_{0,1}$	$E_{k_0}\{k_1,[s_{11}-s_{12}]_e,[y_{11}-y_{12}]_{p_1},[y_{13}-y_{14}]_{p_2},...,[y_{1(2m_1-1)}-y_{1(2m_1)}]_{p_{m_1}}\}$
$l_{0,2}$	$E_{k_0}\{k_2,[s_{21}-s_{22}]_e,[y_{21}-y_{22}]_{p_1},[y_{23}-y_{24}]_{p_2},...,[y_{2(2m_2-1)}-y_{2(2m_2)}]_{p_{m_2}}\}$
…	…
$l_{0,n}$	$E_{k_0}\{k_n,[s_{n1}-s_{n2}]_e,[y_{n1}-y_{n2}]_{p_1},[y_{n3}-y_{n4}]_{p_2},...,[y_{n(2m_n-1)}-y_{n(2m_n)}]_{p_{m_n}}\}$

表 15 中的标签按照以下方式计算:

$$l_{0,1}=hash(k_0,[s_{11}-s_{12},y_{11}-y_{12},y_{13}-y_{14},...,y_{1(2m_1-1)}-y_{1(2m_1)}]_{p_1})$$

$$l_{0,2}=hash(k_0,[s_{21}-s_{22},y_{21}-y_{22},y_{23}-y_{24},...,y_{2(2m_2-1)}-y_{2(2m_2)}]_{p_2})$$

…

$$l_{0,n}=hash(k_0,[s_{n1}-s_{n2},y_{n1}-y_{n2},y_{n3}-y_{n4},...,y_{n(2m_n-1)}-y_{n(2m_n)}]_{p_n})$$

对于攻击者而言,由于其不能获取顶点v_0的密钥和权限区间,因而其从表 15 中将得不到任何有意义的信息.对于具有访问权限的共享用户而言,由于其能够获取v_0的顶点密钥和权限区间,因而可根据令牌的标签快速获取得到目标密钥令牌路径.例如,一个共享用户u_1(其已获取顶点v_0的密钥和权限区间)需解密序列号为s_{22}的文件.由于s_{22}包含在v_0的第 2 个权限区间$[s_{21}-s_{22},y_{21}-y_{22},y_{23}-y_{24},...,y_{2(2m_2-1)}-y_{2(2m_2)}]_{p_2}$内,$u_1$将利用$k_0$和第 2 个权限区间计算 hash 值: $hash(k_0,[s_{21}-s_{22},y_{21}-y_{22},y_{23}-y_{24},...,y_{2(2m_2-1)}-y_{2(2m_2)}]_{p_2})$.之后从表 15 中找到标签值为 $hash(k_0,[s_{21}-s_{22},y_{21}-y_{22},y_{23}-y_{24},...,y_{2(2m_2-1)}-y_{2(2m_2)}]_{p_2})$ 的令牌.之后用k_0解密这个令牌,得到 $k_2,[s_{21}-s_{22}]_e,[y_{21}-y_{22}]_{p_1},[y_{23}-y_{24}]_{p_2},...,[y_{2(2m_2-1)}-y_{2(2m_2)}]_{p_{m_2}}$, 说明密钥$k_2$即为目标密钥.

由于用户顶点处于密钥推导图的最上层,因而需要考虑如何将用户顶点的权限区间信息(本文假定用户密钥不用于加密任何文件,因而用户顶点的加密区间为空)分发给共享用户.一种简单的方法是在分发用户密钥时一同将权限区间信息分发给用户,然而这将使共享用户需要保存用户密钥以外的数据,这将增加共享用户的管理负担.本文将为用户生成用于传递用户顶点权限区间信息的令牌,使系统中的用户仅需保存一个密钥即可根据公开信息推出其权限访问内所有文件的密钥.

设共享用户u_1的用户密钥为k_1,密钥推导图中代表其的用户顶点为v_1,v_1的权限区间为 $[y_{11}-y_{12}]_{p_1},[y_{13}-y_{14}]_{p_2},...,[y_{1(2m-1)}-y_{1(2m)}]_{p_m}$. 为用户传递权限区间信息的令牌按照表 16 格式生成.

Table 16 Token containing the priviledge intervals of user vertex

表 16 传递用户顶点权限区间的令牌格式

标签	密文
$hash(k_1)$	$E_{k_1}\{[y_{11}-y_{12}]_{p_1},[y_{13}-y_{14}]_{p_2},...,[y_{1(2m-1)}-y_{1(2m)}]_{p_m}\}$

3.2 具体算法

算法中用到的各项参数见表 17.

Table 17 System parameters

表 17 参数说明

参数	描述	参数	描述
aug	代表一个访问用户集合	*e*	代表一条有向边
AUG	*aug*的集合	*e.svertex*	有向边的出发顶点
v	代表密钥推导图中的一个顶点	*v.fvertex*	有向边的终端顶点
v.attribute	顶点属性(用户/文件/中间顶点)	*e.order*	将有向边添加至 *ST* 上的序号
v.aug	顶点 *v* 的访问用户集合	*e.attribute*	值为 0 表示有向边属于 *ST*
v.einterval	顶点*v*的加密区间	*Edge*	代表有向边的集合
v.pinterval	顶点*v*的权限区间	*V*	密钥推导图上所有顶点的集合
v.k	顶点*v*的密钥	*G*(*V*,*Edge*)	密钥推导图
v.filegroup	顶点密钥 *v.k* 加密的文件集合	v_T	添加的根顶点
v.filenumber	顶点密钥*v.k*加密的文件数量	*ST*	密钥推导图的深度优先生成树
v.order	将顶点添加至 *ST* 上的序号	*STEdge*	*ST* 上的有向边集合
v.reachability	值为 0 表示顶点不包含子顶点	*t*	表示一个令牌
f	代表一个文件	*t.value*	令牌的密文
f.sn	文件序列号	*t.lable*	令牌的标签
F	所有文件的集合	*E*	对称加密算法

具体算法如图 7 所示.算法分为 4 个步骤:

(1) 初始化;

(2) 生成密钥推导图;

(3) 标记密钥推导图上顶点的加密区间和权限区间;

(4) 分配文件序列号,生成文件序列号列表和令牌列表.下面详细介绍算法的每个步骤.

(1) 初始化.

• 为每个访问用户集合创建一个顶点.顶点 v 是一个结构,包含 9 个成员:顶点属性$v.attribute$、顶点访问用户集合$v.aug$、顶点加密区间$v.einterval$、顶点权限区间$v.pinterval$、顶点密钥 $v.k$、顶点密钥加密的文件集合$v.filegroup$、顶点密钥加密的文件数量$v.filenumber$、顶点序号$v.order$和$v.reachability$(值为 0 时表示顶点不包含子顶点).

在本文方案中,将每个共享用户视为一个只包含自身的访问用户集合(称为共享用户的访问用户集合).在输入访问控制策略后,算法首先生成访问用户集合的集合AUG(包含所有共享用户和文件的访问用户集合,当文件的访问用户集合只有一个共享用户时,将其视为与共享用户访问用户集合不同的集合).之后为每个访问用户集合 aug 创建一个顶点 v.若aug为共享用户的访问用户集合,则令$v.attribute$=0,若为文件的访问用户集合,则令$v.attribute$=1.令$v.aug=aug$,$v.k$等于访问用户集合为aug的用户的密钥/文件的加密密钥.$v.filegroup$是利用$v.k$加密的文件的集合.$v.filenumber$等于$v.filegroup$中的文件数量.

• 为每个文件创建一个序列号变量 $f.sn$.

(2) 生成密钥推导图.可利用选择加密初始方案,得到密钥推导图.设得到的密钥推导图为$G(V,Edge)$,其中,V为初始化过程中创建的顶点和插入的中间顶点的集合,$Edge$为密钥推导图中有向边的集合.其中,插入的中间顶点的顶点属性$v.attribute$设定为 2.e 代表一条有向边.e 是一个结构,包含 4 个成员:出发顶点$e.svertex$、终端顶点$v.fvertex$、有向边序号$e.order$和有向边属性$e.attribute$(默认$e.attribute$初值为 0.当有向边位于深度优先生成树上时,将该值设为 1).

```
算法输入:访问控制策略、用户密钥集合和文件解密密钥集合;
算法输出:文件序列号列表和令牌列表.
MAIN
Step 1. 初始化
根据访问控制策略生成AUG
  for each aug∈AUG do
    create vertex v
    v.attribute=0(aug 为共享用户的访问用户集合)/1(aug为文件的访问用户集合).
    v.aug=aug
    v.einterval=null
    v.pinterval=null
    v.k=访问用户集合为 aug 的用户的密钥/文件的加密密钥
    v.filegroup=利用v.k加密的文件集合
    v.filenumber=v.filegroup 中的文件数目
    v.order=null
    v.reachability=1
  for each file f∈F do
    create f.sn
    f.sn=null
Step 2. 生成密钥推导图
Step 3. 标记密钥推导图上顶点的加密区间和权限区间
Step 4. 分配文件序列号,生成文件序列号列表和令牌列表.
```

Fig.7 Our algorithm

图 7 本文算法

(3) 标记密钥推导图上顶点的加密区间和权限区间(算法细节如图 8 所示).

• 生成深度优先生成树,并设定顶点的顶点序号值$v.order$、$v.reachability$和有向边序号值$e.order$.添加根顶点 v_T 和v_T指向用户顶点的有向边(二者不分配顶点序号和有向边序号),从根顶点v_T出发遍历图中所有顶点得到深度优先生成树 $ST(V\cup v_T, STEdge)$, $STEdge$ 表示ST上有向边的集合.在此过程中,算法已设定顶点的顶点序号$v.order$、$v.reachability$和有向边的有向边序号$e.order$.其中顶点序号表示将顶点添加至 $ST(V\cup v_T, STEdge)$ 上的序号.有向边序号表示将有向边添加至 $ST(V\cup v_T, STEdge)$ 上的序号.若顶点没有子顶点,将其$v.reachability$值设为 0.例如,在图 10 中,设根顶点v_T从顶点v_1开始搜索,沿着有向边一直搜索至顶点v_8.v_8没有子顶点,将其添加至深度优先生成树上,并设定 $v_8.order=1$, $v.reachability=1$.之后,算法返回至顶点v_7,v_7除顶点v_8外没有其他子顶点,则将其添加至深度优先生成树上,并设定 $v_7.order=2$. 同时设定有向边 $e\langle v_7, v_8\rangle.order=1$.

- 删除根顶点v_T和v_T指向用户顶点的有向边,标记各顶点的加密区间.后续工作已不再需要用到根顶点v_T及其指向用户顶点的有向边,因而可删除v_T和这些有向边.按照顶点序号值从小到大依次标记顶点的加密区间.
- 将属于*STEdge*的有向边的可达性信息标记为顶点的权限区间信息.
- 将属于集合 *Edge*/*STEdge* 的有向边的可达性信息标记为顶点的权限区间信息.

(4) 为文件分配序列号,生成文件序列号列表和令牌列表(算法细节如图 9 所示).生成各有向边对应令牌的方法为:利用有向边出发顶点的顶点密钥加密终端顶点的顶点密钥、加密区间和权限区间得到令牌密文,利用第 3.1 节中的标签计算方法得到令牌标签.之后,生成向用户传递其权限区间信息的公开令牌.利用用户密钥加密用户顶点的权限区间得到令牌密文.计算用户密钥的 hash 值作为令牌标签.

例 5(本文方案应用举例——根据访问控制策略生成用于实现访问控制的公开信息):设访问控制策略与表 3 相同.经过步骤(1)和步骤(2)后,可得到密钥推导图(如图 1 所示).步骤(3)首先利用深度优先搜索算法得到图的深度优先生成树 *ST*,如图 2 所示.

/*Step 3*/
/*生成深度优先生成树,并设定*v.order*、*v.reachability*和*e.order*的值.*/
添加根顶点v_T和v_T指向用户顶点的有向边
利用深度优先搜索法生成深度优先生成树 $ST(V \cup v_T, STEdge)$
for each $v \in V$
assign *v.order*
assign *v.reachability*
for each $E \in STEdge / \{e \mid e.svertex$ 等于$v_T\}$
assign *e.order*
/*标记文件顶点的加密区间*/
删除根顶点v_T和v_T指向用户顶点的有向边
$STEdge = STEdge / \{e \mid e.svertex$ 等于$v_T\}$
SN=0
for $v.order = 1:|V|$ do
If $v.attribute = 1$ do
$v.einterval = [(SN+1)-(SN+v.filenumber)]$
If $v.reachability = 0$ do
delete *v.pinterval*
$SN = SN + v.filenumber$
/*将属于 *STEdge* 的有向边的可达性信息标记为顶点的权限区间信息.*/
for $e.order = 1:|STEdge|$ do
find $e.svertex = v_x$
find $e.fvertex = v_y$
find $v_y.einterval = [s_1 - s_2]_e$
find $v_y.pinterval = [y_1 - y_2]_p$
$v_x.pinterval = [s_1 - s_2]_p[y_1 - y_2]_p$ /*若 v_y 的权限区间与加密区间(或v_y的几个权限区间)区间值连续,在将其写入v_x的权限区间时,可将这些值写出一个连续的区间.例如,图 11 中的顶点v_{10}的加密区间和权限区间为$[5\text{-}5]_e$,$[3\text{-}4]_p$,$[6\text{-}6]_p$.标记为顶点v_2的权限区间时,可将三者合并,写为$[3\text{-}6]_p$.下同*/
/*将属于集合 *Edge*/*STEdge* 的有向边的可达性信息标记为顶点的权限区间信息*/
for each $e \in (Edge / STEdge)$ /*从位于密钥推导图最底层的有向边(不属于*STEdge*)逐层向上进行*/
find $e.svertex = v_x$
find $e.fvertex = v_y$
$v_x.pinterval = v_x.pinterval \cup [v_y.einterval, v_y.pinterval]_p$

Fig.8 Step 3

图 8 步骤 3

/*Step 4*/
/*为文件分配序列号*/
for $v.order = 1:|V|$ do
If $v.attribute = 1$ do
find $v.einterval = [s_1 - s_2]_e$
for $i = 1: f.filenumber$ do
$f_i.sn = s_1 + i$ /* $f_i \in v.filegroup$ */
/*生成有向边对应的令牌*/
for each $e \in E$
find $e.svertex = v_x$
find $e.fvertex = v_y$
create t
$t.value = E_{v_x.k}\{v_y.k, v_y.einterval, v_y.pinterval\}$
$t.label = hash(v_x.k, [v_y.einterval, v_y.pinterval]_p)$
/*生成向用户传递其权限区间信息的公开令牌.*/
for each $v \in V$ do
If $v.attribute = 0$ do
create t
$t.value = E_{v.k}\{v.pinterval\}$
$t.label = hash(v.k)$

Fig.9 Step 4
图 9 步骤 4

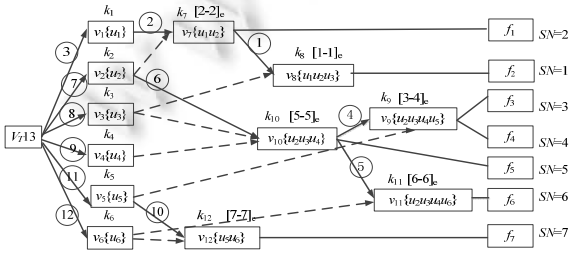

Fig.10 Marking vertex encryption intervals
图 10 标记顶点的加密区间

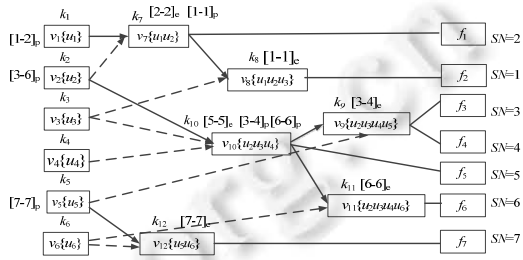

Fig.11 Marking reachability of ST as vertex privilege intervals
图 11 标记深度优先生成树的可达性信息为顶点权限区间

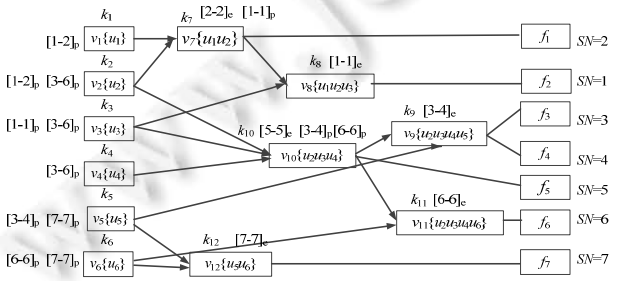

Fig.12 Marking reachability of key derivation graph as vertex privilege intervals
图 12 标记整个密钥推导图的可达性信息为顶点权限区间

之后删除根顶点 v_T 和v_T指向用户顶点的有向边,标记顶点的加密区间,如图 10 所示.接着将 *STEdge* 内的有向边的可达性信息标记为顶点的权限区间信息,如图 11 所示.然后,将集合 *Edge*/*STEdge* 内的有向边的可达性信

息标记为顶点的权限区间信息,如图 12 所示.最后,为文件分配序列号,生成文件序列号列表和令牌列表.算法生成的文件序列号列表和令牌列表见表 18 和表 19.

Table 18 SN list

表 18 文件序列号列表

文件	序列号
f_2	1
f_1	2
f_3	3
f_4	4
f_5	5
f_6	6
f_7	7

Table 19 Token list

表 19 令牌列表

标签	密文	标签	密文
$hash(k_1)$	$E_{k_1}\{[1-2]_p\}$	$hash(k_3,[3-6]_p)$	$E_{k_3}\{k_{10},[5-5]_e,[3-4]_p,[6-6]_p\}$
$hash(k_2)$	$E_{k_2}\{[1-2]_p,[3-6]_p\}$	$hash(k_4,[3-6]_p)$	$E_{k_4}\{k_{10},[5-5]_e,[3-4]_p,[6-6]_p\}$
$hash(k_3)$	$E_{k_3}\{[1-1]_p,[3-6]_p\}$	$hash(k_5,[3-4]_p)$	$E_{k_5}\{k_9,[3-4]_e\}$
$hash(k_4)$	$E_{k_4}\{[3-6]_p\}$	$hash(k_5,[7-7]_p)$	$E_{k_5}\{k_{12},[7-7]_e\}$
$hash(k_5)$	$E_{k_5}\{[3-4]_p,[7-7]_p\}$	$hash(k_6,[6-6]_p)$	$E_{k_6}\{k_{11},[6-6]_e\}$
$hash(k_6)$	$E_{k_6}\{[6-6]_p,[7-7]_p\}$	$hash(k_6,[7-7]_p)$	$E_{k_6}\{k_{12},[7-7]_e\}$
$hash(k_1,[1-2]_p)$	$E_{k_1}\{k_7,[2-2]_e,[1-1]_p\}$	$hash(k_7,[1-1]_p)$	$E_{k_7}\{k_8,[1-1]_e\}$
$hash(k_2,[1-2]_p)$	$E_{k_2}\{k_7,[2-2]_e,[1-1]_p\}$	$hash(k_{10},[3-4]_p)$	$E_{k_{10}}\{k_9,[3-4]_e\}$
$hash(k_2,[3-6]_p)$	$E_{k_2}\{k_{10},[5-5]_e,[3-4]_p,[6-6]_p\}$	$hash(k_{10},[6-6]_p)$	$E_{k_{10}}\{k_{11},[6-6]_e\}$
$hash(k_3,[1-1]_p)$	$E_{k_3}\{k_8,[1-1]_e\}$	–	–

例 6(共享用户获取目标密钥举例):设用户 u_2 需要解密文件f_3.u_2获取目标密钥的步骤如下:

(1) 计算个人密钥的 hash 值 $hash(k_2)$,从表 19 中找到标签值为$hash(k_2)$的令牌,解密 $E_{k_2}\{[1-2]_p,[3-6]_p\}$得到自己的权限区间为 $[1-2]_p,[3-6]_p$.

(2) 查询表 18,找到文件f_3的文件序列号为 3.序列号 3 包含于第 2 个权限区间,故计算 hash 值$hash(k_2,[3–6]_p)$.

(3) 找到标签值为 $hash(k_2,[3–6]_p)$的令牌,解密 $E_{k_2}\{k_{10},[5-5]_e,[3-4]_p,[6-6]_p\}$得到 $k_{10},[5-5]_e,[3-4]_p,[6-6]_p$.序列号 3 包含于第 1 个权限区间,故计算 hash 值 $hash(k_{10},[3-4]_p)$.

找到标签值为$hash(k_{10},[3–4]_p)$的令牌,解密 $E_{k_{10}}\{k_9,[3-4]_e\}$得到 $k_9,[3-4]_e$.序列号 3 包含于该点的加密区间,说明密钥k_9即为目标密钥.

4 安全性和性能分析

4.1 安全性分析

定理 1. 文件、文件顶点和中间顶点密钥、加密区间和权限区间的机密性等价于所采用的对称加密算法和 hash 算法的安全性.

证明:设系统中文件集合为 F,文件序列号集合为 SN,令牌集合为 T,采用的对称加密算法为 E,采用的 hash 算法为 hash.对于敌手而言,文件序列号相当于文件的一个编号.敌手不能从公开的SN中获取文件、文件顶点和中间顶点密钥、加密区间和权限区间的任何机密性信息.当一个用户获取其权限范围内一个文件的解密密钥,设其所需解密的令牌路径为 $p=t_u,t_{u,1},t_{u,2},...,t_{n-1,n}$.令牌集合$T$等于系统中所有用户的令牌路径的并集.$\forall f\in F$,设$f$的加密密钥为$k_f$,对应密文为 $E_{k_f}\{f\}$,其文件序列号为 n.具有f访问权限的共享用户 u 获取文件加密密钥k_f的令牌路径可写为表 20.

Table 20 Derivation path for the target key

表 20 获取目标密钥的令牌路径

计算标签	解密令牌
$l_u = hash(k_u)$	$t_u = E_{k_u}\{[y_{01}-y_{02}]_{p_1},[y_{03}-y_{04}]_{p_2},...,[y_{0(2m_0-1)}-y_{0(2m_0)}]_{p_{m_0}}\}$
$l_{u,1} = hash(k_u,[y_{01}-y_{02}]_{p_1})$	$E_{k_u}\{k_1,[s_{11}-y_{12}]_e,[y_{11}-y_{12}]_{p_1},[y_{13}-y_{14}]_{p_2},...,[y_{1(2m_1-1)}-y_{1(2m_1)}]_{p_{m_1}}\}$
$l_{u,2} = hash(k_1,[y_{11}-y_{12}]_{p_1})$	$E_{k_1}\{k_2,[s_{21}-y_{22}]_e,[y_{21}-y_{22}]_{p_1},[y_{23}-y_{24}]_{p_2},...,[y_{1(2m_2-1)}-y_{1(2m_2)}]_{p_{m_2}}\}$
…	…
$l_{n-2,n-1} = hash(k_{n-2},[y_{(n-2)1}-y_{(n-2)2}]_{p_1})$	$E_{k_{n-2}}\{k_{n-1},[s_{(n-1)1}-y_{(n-1)2}]_e,[y_{(n-1)1}-y_{(n-1)2}]_{p_1},[y_{(n-1)3}-y_{(n-1)4}]_{p_2},...,$ $[y_{(n-1)(2m_{(n-1)}-1)}-y_{(n-1)(2m_{(n-1)})}]_{p_{m_{(n-1)}}}\}$
$l_{n-1,n} = hash(k_{n-1},[y_{(n-1)1}-y_{(n-1)2}]_{p_1})$	$E_{k_{n-1}}\{k_n,[s_{n1}-y_{n2}]_e,[y_{n1}-y_{n2}]_{p_1},[y_{n3}-y_{n4}]_{p_2},...,[y_{n(2m_n-1)}-y_{n(2m_n)}]_{p_{m_n}}\}$

其中,设:

$$y_{01} \leqslant y_{11} \leqslant y_{21} \leqslant ... y_{(n-2)1} \leqslant y_{(n-1)1} \leqslant s_{n1} \leqslant n \leqslant s_{n2} \leqslant y_{(n-1)2} \leqslant y_{(n-2)2} ... \leqslant y_{22} \leqslant y_{12} \leqslant y_{02},$$

$$k_n = k_f.$$

由表 20 可知,只要 hash 算法和对称加密算法 E 是安全的,敌手将不能从令牌路径中获取文件、文件顶点和中间顶点密钥、加密区间和权限区间信息.由于 f 从 F 任取,令牌集合 T 于系统中所有用户的令牌路径的并集,因而,敌手将不能从令牌集合 T 中获取任何文件、文件顶点和中间顶点密钥、加密区间和权限区间信息.文件、文件顶点和中间顶点密钥、加密区间和权限区间的机密性等价于所采用的对称加密算法和 hash 算法的安全性.

本文假定采用的对称加密算法和 hash 算法是安全的.本文方案具有以下安全特性:

(1) 文件的机密性保护.在本方案中,数据拥有者需要分发给共享用户的文件均采用对称加密算法加密后再上传云端.由定理 1 可知,云服务器和攻击者将不能获取存储文件的明文信息,实现了有效的文件机密性保护.

(2) 细粒度的访问控制.本方案将具有相同访问用户集合的文件采用同一密钥加密,将具有不同访问用户集合的文件采用不同的密钥加密,具有合适的加密粒度.本文采用文献[9]中的方法生成密钥推导图,密钥推导图等价于访问控制策略.确保了只有属于文件访问用户集合的用户才能从公开信息中推出相应文件的解密密钥,实现了有效地细粒度访问控制.

(3) 文件权限信息的隐藏.在本方案中,由于每个文件均拥有一个不同的文件序列号,因而云服务器和攻击者将不能根据文件序列号列表,判断哪些文件具有相同的访问用户集合,实现了文件权限信息的隐藏.

(4) 访问控制策略的隐藏.在本文方案中,即使攻击者与云服务器合谋,二者所获取信息也不会多于云服务器所获取的信息.因而本方案中仅需考虑云服务器意在获取数据拥有者访问控制策略的单独攻击和云服务器与少数恶意用户的合谋攻击.

- 抗云服务器单独攻击.在本文方案中,公开信息仅包括文件序列号列表和令牌列表.由定理 1 可知,文件、文件顶点和中间顶点密钥、加密区间和权限区间的机密性等价于所采用的对称加密算法和 hash 算法的安全性,而本文假定所采用的对称加密算法和 hash 算法是安全的,因而云服务器除了从公开信息中获取数据拥有者的文件总数外,并不能从公开信息中获取其他任何访问控制策略的相关信息.
- 抗云服务器与少数恶意共享用户的合谋攻击.当云服务器与少数恶意用户合谋攻击时,他们仅能解密恶意用户权限范围内的文件以及相应的令牌路径集合 T_E.根据定理 1 和本文假设,云服务器与少数恶意共享用户并不能解密其他令牌集合 $T–T_E$,因而也就不能获取他们权限范围外的加密区间、权限区间和密钥信息.对于云服务器和少数恶意共享用户而言,他们仅能获取涉及恶意用户自身的访问控制策略信息.当云服务器与多数恶意共享用户合谋时,他们可以获取他们权限访问外的文件的敏感程度信息.例如,若数据拥有者拥有 n 个文件,多数恶意共享用户可以访问其中的 $n–1$ 个文件,据此,他们可推断剩余的一个文件敏感度较高.除此之外,云服务器和多数恶意用户合谋不能获取其他访问控制策略信息,例如共享用户的数量,其他共享用户的具体访问权限.在现实应用场景中,一般只会出现较少的恶意用户,因而,本文方案假定云服务器仅与少数恶意共享用户合谋是合理的.

综上所述,本方案有效地实现了访问控制策略的隐藏.值得说明的是,本方案也面临与其他存储方案同样的问题——云服务器可以通过记录每个共享用户的存取记录,获取数据拥有者的访问控制策略,这可通过随机存取方法加以解决.

4.2 性能分析

本节首先给出本文方案的计算复杂度,并与 PCSP 方案进行比较;之后对两个方案的公开信息空间复杂度作分析和比较;最后,对两个方案中共享用户获取目标密钥需遍历的顶点数量和速度进行分析和比较.

(1) 计算复杂度

设系统中有 N 个共享用户,M 个文件,密钥推导图由 L 个顶点和 S 条有向边构成.为密钥推导图添加根顶点后,得到深度优先生成树上共有 X 条有向边.在输入数据拥有者的访问控制策略后,本文方案和 PCSP 方案均采用文献[8,9]的方法生成密钥推导图,其计算复杂度与文献[8,9]方案相同,计算复杂度为 $O(NL^2)$.此外,假定本文方案与 PCSP 方案均采用文献[27]中的方法获取密钥推导图的深度优先生成树,其计算复杂度为 $O(L+S)$.本文主要对得到深度优先生成树后各操作的计算复杂度进行分析.

本文方案首先为 L 个顶点分配加密区间,该步骤的计算复杂度为$O(L)$.为顶点分配加密区间后,删除根顶点及根顶点指向用户顶点的有向边,并标记通过深度优先生成树上的有向边到达的顶点信息为权限区间,该步骤的计算复杂度为 $O(X–N)$.随后,标记顶点通过不在 ST 上的有向边或路径到达的顶点信息为权限区间,该步骤的计算复杂度为 $O(S–X+N)$.之后,为文件分配序列号,该步骤的计算复杂度为$O(M)$.最后生成序列号列表和令牌列表的计算复杂度分别为$O(M)$和$O(S+N)$.本文方案所有操作的计算复杂度为 $O(NL^2+S+M+X)$.

PCSP 方案首先为 $L+1$ 个顶点添加顶点编号,该步骤的计算复杂度为$O(X)$.对顶点进行编号后,标记顶点通过深度优先生成树上 ST 的有向边到达的顶点信息为权限区间,该步骤的计算复杂度为$O(X)$.删除根顶点及根顶点指向用户顶点的有向边后,标记顶点通过不在 ST 上的有向边或路径到达的顶点信息为权限区间,该步骤的计算复杂度为$O(S–X+N)$.随后,标记有向边的权限区间,该步骤的计算复杂度为$O(S)$.

Table 21 Computational complexity comparison

表 21 公开信息具体操作计算复杂度比较

	操作	复杂度		操作	复杂度
PCSP 方案	标记顶点编号	$O(X)$	本文方案	标记顶点加密区间	$O(L)$
	标记 ST 上的可达性信息为顶点的权限区间	$O(X)$		标记 ST 上的可达性信息为顶点的权限区间	$O(X–N)$
	标记 E/ST 上的可达性信息为顶点的权限区间	$O(S–X+N)$		标记E/ST上的可达性信息为顶点的权限区间	$O(S–X+N)$
	标记有向边的权限区间	$O(S)$		为文件分配序列号	$O(M)$

(1) 公开信息的空间复杂度比较

为了实现密钥分发,PCSP 方案需要公布的信息是:文件解密标签列表、用户/文件标签对应编号列表和令牌列表.本文方案需要公布以下信息:文件序列号列表和令牌列表.设两个方案中的对称加密算法均采用 AES 算法,密钥长度为 256bit.hash 算法采用的是 SHA-1,消息摘要长度为 160bit,两个方案中的标签的位长均等于消息摘要长度.PCSP 方案中的文件名、标签编号和令牌号和本文方案中的文件序列号的位长等于密钥长度.

在 PCSP 方案中,文件解密标签列表包含 M 行数据,其空间复杂度为$O(M)$.标签编号列表包含 L 行数据,其空间复杂度为$O(L)$.令牌列表包含 S 行数据,在最坏的情况下,一个令牌中包含$\lfloor L/2 \rfloor$个权限区间,每个权限区间的位长等于两个标签编号的长度,因此,令牌列表的空间复杂度为$O(LS)$.

在本文方案中,文件序列号列表包含 M 行数据,其空间复杂度为$O(M)$.令牌列表包含 $N+S$ 条数据,在最坏的情况下,一个令牌中包含一个加密区间和$\lfloor M/2 \rfloor$个权限区间,每个加密区间和权限区间的位长等于两个文件编号的长度,因此,令牌列表的空间复杂度为$O(MS)$.

(2) 获取目标密钥遍历的顶点数量和密钥获取速度比较

设共享用户获取目标密钥速度的令牌路径长度为 x,并假定令牌路径上各顶点均包含 y 个子顶点.本文假定

查表运算、hash 运算和解密令牌运算可在常数时间内完成.

在 PCSP 方案中,在最坏的情况下,共享用户需遍历 xy 个顶点.要解密以令牌路径上各顶点(目标文件顶点除外)为起始顶点的有向边所对应的令牌.为了获取目标密钥,共享用户需要进行(xy+2)次查表运算,xy次解密令牌运算、x次 hash 运算和x次异或运算,获取目标密钥的计算复杂度为 $O(xy)$.

在本文方案中,为了获取目标密钥,共享用户只需遍历x个顶点.共享用户只需进行(x+2)次查表运算,(x+1)次解密令牌运算、(x+1)次 hash 运算,获取目标密钥的计算复杂度为$O(x)$.

5 相关工作

对于云存储用户而言,主要通过利用密钥分发实现对存储于云端数据的访问控制.本文按照密钥分发所采用的加密算法,将现有的云存储访问控制方案分为两大类:基于对称加密的方案和基于非对称加密的方案.

基于对称加密方案主要是选择加密.文献[8]首次将选择加密用于外存储环境的访问控制.米兰大学的研究人员在此方面做了一系列的研究工作[8−10,14,15].文献[14]实现了对用户读写权限的同时赋予.文献[16]提出了一种双头层结构,可实现访问控制策略的高效更新.

基于非对称加密的方案可分为单一加密策略和混合加密策略两种.单一加密策略主要包括基于属性的加密和基于代理重加密的两种.基于属性的加密(ABE)机制由 Sahai 和 Waters 在 2005 年的欧密会上提出[17],之后 ABE 发展为 KP-ABE 和 CP-ABE 两种.KP-ABE 在 2006 年由 Goyal 等人提出[18],即用户私钥与访问控制结构相关联,密文与属性相关联.在该机制下,当密文属性集合满足用户的访问控制树时,用户便可以解密密文.CP-ABE 在 2007 年由 Bethencourt 等人提出[19],在该机制下,用户私钥与属性相关联,密文与访问控制结构相关联,当用户私钥属性集合满足密文的访问控制树时,用户才能解密密文.匿名 ABE 首次由 Kapadia 等人[20]提出,主要是为了防止密文加密规则的泄露,用户每次解密都必须使用全部的属性私钥,不能仅选取部分私钥进行解密.文献[21]在减少匿名 ABE 的密文大小方面进行了研究.代理重加密的概念由 Blaze 等人在 1998 年的欧密会上提出[22].代理重加密即一个代理可以利用由 Alice 生成的代理重加密密钥,将由 Alice 公钥加密的密文直接转换为用 Bob 私钥可以解密的密文,且代理不能获得关于密文所对应明文的任何信息.ATENIESE 等人第 1 次将单向代理重加密用于分布式存储系统的密钥管理[23].在该方案中,将加密文件的加密密钥用一个主公钥加密.数据拥有者根据所有拥有权限的用户的公钥,生成代理重加密密钥,并将代理重加密密钥发送给代理服务器.当用户需要相应数据时,只需向服务器发送请求,服务器将存储在其上的加密数据重加密为拥有权限用户可解密的密文.混合加密策略方案将多种加密策略结合起来用于实现访问控制.Yu 等人在 2010 年的 IEEE INFOCOM 会议上提出了第 1 个能够同时实现细粒度、可升级和保证数据机密性的云计算数据访问控制方案[12].在该方案中,云服务器用于存储和代理重加密需要升级的加密数据,不获取任何关于明文的信息.该方案结合 KP-ABE、代理重加密和延迟重加密等多种加密技术.此外,在 2010 年的 ASIACCS 会议上,Yu 等人还提出了一种将 CP-ABE 与代理重加密结合的访问控制方案[24],该方案实现单个属性的细粒度撤销,但是效率偏低.文献[25]提出了一种将 CP-ABE、盲解密和秘密共享结合的访问控制方案.文献[26]提出了一种细粒度、基于时间及时更新密文的访问控制方案.

6 总 结

云存储服务以其低成本、按需付费等特点得到了广泛的使用.在享受云存储带来的益处的同时,如何保证数据的机密性和用户隐私并实现有效地访问控制是进入云时代以来一个重要的研究课题.基于现有的研究工作,本文提出了一个新的访问控制策略隐藏机制,在保证云存储数据机密性和细粒度访问控制的前提下, 有效地实现了访问控制策略的隐藏和密钥的高效分发.

References:

[1] http://aws.amazon.com/cn/s3/

[2] https://www.icloud.com/

[3] http://yun.baidu.com/?ref=ppzq

[4] http://www.windowsazure.cn/?fb=002

[5] http://www.ksyun.com/

[6] http://www.iimedia.cn/38351.html

[7] http://popcrush.com/apple-releases-statement-icloud-celeb-photo-hacks

[8] De Capitani di Vimercati S, Foresti S, Jajodia S, Paraboschi S, Samarati P. Over-Encryption: Management of access control evolution on outsourced data. In: Wolfgang K, ed. Proc. of the 33rd Int'l Conf. on Very Large Data Bases. Vienna: VLDB Endowment, 2007. 123−134.

[9] De Capitani di Vimercati S, Foresti S, Jajodia S, Paraboschi S, Samarati P. Encryption policies for regulating access to outsourced data. ACM Trans. on Database Systems, 2010,35(2):12. [doi: 10.1145/1735886.1735891]

[10] De Capitani di Vimercati S,Foresti S, Jajodia S, Paraboschi S, Pelosi G, Samarati P. Preserving confidentiality of security policies in data outsourcing. In: AtluriV, ed. Proc. of the 7th ACM Workshop on Privacy in the Electronic Society. New York: ACM, 2008. 75−84. [doi: 10.1145/1456403.1456417]

[11] Agrawal R, Borgida A, Jagadish HV. Efficient management of transitive relationships in large data and knowledge bases. In: Clifford J, ed. Proc. of the 1989 ACM SIGMOD Int'l Conf. on Management of data. New York: ACM, 1989. 253−262. [doi: 10.1145/66926.66950]

[12] Yu SC, Wang C, Ren K, Lou WJ. Achieving secure, scalable, and fine-grained data access control in cloud computing. In: Mandyam G, ed. Piscataway: IEEE, 2010. 1−9. [doi: 10.1109/INFCOM.2010.5462174]

[13] Wang Q, Wang C, Li J, Ren K, Lou WJ. Enabling public verifiability and data dynamics for storages security in cloud computing. In: Proc. of the 14th European Symp. on Research in Computer Security. Berlin, Heidelberg: Springer-Verlag, 2009. 355−370. [doi: 10.1007/978-3-642-04444-1_22]

[14] De Capitani di Vimercati S, Foresti S, Jajodia S, Livraga G, Paraboschi S, Samarati P. Enforcing dynamic write privileges in data outsourcing. Computers & Security, 2013,47−63. [doi: 10.1016/j.cose.2013.01.008]

[15] Blundo C, Cimato S, De Capitani di Vimercati S, Santis AD, Foresti S, Paraboschi S, Samarati P. Managing key hierarchies for access control enforcement: Heuristic approaches. Computer & Security, 2010,29:533−547. [doi: 10.1016/j.cose.2009.12.006]

[16] Jiang WY, Wang Z, Liu LM, Gao N. Towards efficient update of access control policy for cryptographic cloud storage. In: Proc. of the SeucreComm Workshop on Data Protection in Mobile and Pervasive Computing. 2014.

[17] Sahai A,Waters B. Fuzzy identity-based encryption. In: Cramer R, ed. Advances in Cryptology–EUROCRYPT 2005. Berlin, Heidelberg: Springer-Verlag, 2005. 457−473. [doi: 10.1007/11426639_27]

[18] Goyal V, Pandey O, Sahai A, Waters B. Attribute-Based encryption for fine-grained access control of encrypted data. In: Juels A, ed. Proc. of the 13th ACM Conf. on Computer and Communications Security. New York: ACM, 2006. 89−98. [doi: 10.1145/1180405.1180418]

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

[20] Kapadia A, Tsang PP, Smith SW. Attribute-Based publishing with hidden credentials and hidden policies. In: Proc. of the 14th Annual Network and Distributed System Security Symp. 2007. 179−192.

[21] Li XH, Gu D, Ren YL, Ding N, Yuan K. Efficient ciphertext-policy attribute based encryption with hidden policy. In: Yang X, ed. Proc. of the 5th Int'l Conf. Berlin, Heidelberg: Springer-Verlag, 2012. 146−159. [doi: 10.1007/978-3-642-34883-9_12]

[22] Blaze M, Bleumer G, Strauss M. Divertible protocols and atomic proxy cryptography. In: Nyberg K, ed. Advances in Cryptology —EUROCRYPT'98. Berlin, Heidelberg: Springer-Verlag, 1998. 127−144. [doi: 10.1007/BFb0054122]

[23] Ateniese G, Fu K, Gren M, Hohenberger S. Improved proxy re-encryption schemes with applications to secure distributed storage. ACM Trans. on Information and System Security, 2006,9(1):1−30. [doi: 10.1145/1127345.1127346]

[24] Yu SC, Wang C, Ren K, Lou WJ. Attribute based data sharing with attribute revocation. In: Feng DG, ed. Proc. of the 5th ACM Symp. on Information, Computer and Communications Security. New York: ACM, 2010. 261−270. [doi: 10.1145/1755688.1755720]

[25] Tang Y, Lee PPC, Lui JCS, Perlman R. Secure overlay cloud storage with access control and assured deletion. IEEE Trans. on Dependable and Secure Computing, 2012,9(6):903−916. [doi: 10.1109/TDSC.2012.49]

[26] Liu Q, Wang GJ, Wu J. Time-Based proxy re-encryption scheme for secure data sharing in a cloud environment. Information Sciences, 2012,258(2014):355−370. [doi: 10.1016/j.ins.2012.09.034]

[27] Weiss MA. Data Structures and Algorithm Analysis in C. Addison-Wesley, 1996.

雷蕾(1987－),男,湖南永州人,博士生,CCF 学生会员,主要研究领域为云存储安全.

蔡权伟(1987－),男,博士,助理研究员,主要研究领域为网络与系统安全.

荆继武(1964－),男,博士,研究员,博士生导师,CCF 高级会员,主要研究领域为网络与系统安全.

林璟锵(1978－),男,博士,研究员,CCF 高级会员,主要研究领域为网络与系统安全.

王展(1986－),女,博士,助理研究员,CCF 会员,主要研究领域为云安全,移动安全.

陈波(1982－),男,博士,博士后研究员,主要研究领域为云计算安全,存储安全.