软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2019,30(8):2517−2527 [doi: 10.13328/j.cnki.jos.005486]

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

一种面向公有云的密文共享方案*

罗王平[1], 冯朝胜[1], 秦志光[2], 袁 丁[1], 廖娟平[1], 刘 霞[1]

[1](四川师范大学 计算机科学学院,四川 成都 610101)

[2](网络与数据安全四川省重点实验室(电子科技大学),四川 成都 610054)

通讯作者: 冯朝胜, E-mail: csfenggy@126.com

摘 要: 针对已有的密文共享方案存在客户端计算量过大、用户管理密钥过多、不支持个人共享等问题,将公有云引入到密文共享方案的设计之中,提出一种面向公有云的安全文件共享框架.基于该框架设计了一种面向公有云的密文共享方案.该方案将绝大多数计算和存储都外包给公有云,用户只需保存两个空间占用很小的私钥子项且客户端只需进行少量计算即可完成共享文件的解密.安全分析结果表明,该方案不仅能够对抗恶意用户的合谋攻击,而且在一般群模型和随机预言模型下能够对抗选择明文攻击.

关键词: 密文共享;公有云;基于属性加密

中图法分类号: TP309

中文引用格式: 罗王平,冯朝胜,秦志光,袁丁,廖娟平,刘霞.一种面向公有云的密文共享方案.软件学报,2019,30(8):2517−2527. http://www.jos.org.cn/1000-9825/5486.htm

英文引用格式: Luo WP, Feng CS, Qin ZG, Yuan D, Liao JP, Liu X. Ciphertext sharing scheme for the public cloud. Ruan Jian Xue Bao/Journal of Software, 2019,30(8):2517−2527 (in Chinese). http://www.jos.org.cn/1000-9825/5486.htm

Ciphertext Sharing Scheme for the Public Cloud

LUO Wang-Ping[1], FENG Chao-Sheng[1], QIN Zhi-Guang[2], YUAN Ding[1], LIAO Juan-Ping[1], LIU Xia[1]

[1](School of Computer Science, Sichuan Normal University, Chengdu 610101, China)

[2](Network and Data Security Key Laboratory of Sichuan Province (University of Electronic Science and Technology of China), Chengdu 610054, China)

Abstract: In view of the existing ciphertext sharing scheme, there is too large calculation of user clients, and each user manages too many secret keys. Moreover, it does not support personal sharing and other issues. Thus the Public Cloud is introduced into the ciphertext sharing scheme, and a security file-sharing framework of the Public Cloud is proposed. And based on this framework, a new ciphertext sharing scheme is designed. In this scheme, the vast majority of computation and storage are outsourced to the Public Cloud. The user simply saves two private key components occupying small space. The client only needs a little computation to complete encryption and decryption of the shared file. The security analysis shows that the scheme can deal with not only the conspiracy attack from malicious users, but also the plaintext-chosen attack in the generic group model and the random oracle model.

Key words: ciphertext sharing; public cloud; attribute-based encryption

对于企业信息中心而言,如果要将数据存储外包给公有云并同时保证数据安全性和隐私性,一般做法为:将数据加密后再上传到云端.但如果企业信息中心还想利用这种外包模式构建文件共享框架,问题将变得非常复杂.

解决外包云数据共享的一般方案是:用每个共享用户的公钥加密数据加密密钥,然后将加了密的数据加密

* 基金项目: 国家科技支撑计划(2014BAH11F02); 国家自然科学基金(61373163); 网络与数据安全四川省重点实验室课题(NDS 2019-1)

Foundation item: National Key Technology R&D Program of China (2014BAH11F02); National Natural Science Foundation of China (61373163); Project of Network and Data Security Key Laboratory of Sichuan Province (NDS 2019-1)

收稿时间: 2017-06-15; 修改时间: 2017-10-09; 采用时间: 2017-11-09

密钥发送给共享者.显然,这种方法的代价过大:计算量和通信量都和共享用户的数量成正比,使得它难以实施.针对这个问题,学术界提出了一种称为密文策略的基于属性的加密算法 CP-ABE(ciphertext-policy attribute-based encryption).由于该算法包含的访问控制方法类似于企业信息系统常用的基于角色访问控制方法 RBAC(role-based access control),且具有“一次加密,多人分享”的特点,因此,在外包数据共享中具有可实施性.然而,该算法要求运算量较大的加解密运算都由用户客户端负责且由较多私钥子项组成的用户私钥也都由用户保管,这对其实施特别是在移动设备上的实施造成阻碍.针对这一问题,利用公有云中的云计算服务器来分担大量的计算工作和存储全部用户私钥,提出一种面向公有云环境的密文共享方案.本文的具体贡献包括:

(1) 提出一种面向公有云的文件共享框架.在该框架中,共享数据的数据加密密钥先要基于安全存储访问结构树在用户客户端进行加密.共享时,由云服务器分担近一半文件共享访问结构树对应密文子项的计算.访问结构树的叶子节点不仅可以对应属性,还可以直接对应用户标识符,使得该方案同时支持基于属性的共享和基于身份的共享.

(2) 设计一种面向公有云的文件共享方案.基于所提出的文件共享框架和经典的CP-ABE算法,设计一种具体的可实施的面向公有云的文件共享方案.在该方案中,构成用户私钥的绝大多数私钥子项由云服务器进行存储,用户只需要安全存储两个私钥子项.

(3) 证明方案的有效性.安全性分析表明,所提出的方案和经典的 CP-ABE 具有相同的安全性.性能分析则表明,该方案在计算量和存储量上对用户客户端要求很低,个人电脑和移动设备亦能胜任.

本文首先指出云密文共享存在的主要问题及解决思路并说明主要贡献.第 1 节对密文策略基于属性加密算法的研究现状进行总结,并分析其优缺点.第 2 节提出一种面向公有云的密文共享框架.第 3 节基于所提出的共享框架,构建一种面向公有云的密文共享方案.第 4 节从安全性和性能上对所提出的共享方案进行分析.第 5 节对本文方案进行实验分析.最后对本文进行总结.

1 CP-ABE 相关研究

Bethencourt 等人[1]率先提出 CP-ABE 的构造方法.该算法包括初始化、密钥生成、加密和解密 4 个基本部分.他们证明了所提出的方案在一般群模型和随机预言模型下是可以对抗选择明文攻击的,同时还证明了它可以抵御合谋攻击.2007 年,针对 Bethencourt 等人方案的访问策略不支持“NOT”逻辑运算的问题,Cheung 等人[2]将属性分成正、负和不相关 3 种状态,率先实现在 CP-ABE 方案中支持“NOT”逻辑运算.但他们的方案却不能支持“OR”逻辑运算.2008 年,Goyal 等人针对 Bethencourt 等人所提出算法的安全性仅仅建立在一般群模型上以及 Cheung 等人的方案虽然将安全性扩展到 d-BDH(decisional bilinear Diffie-Hellman)假设条件但访问策略仅能支持“AND”逻辑运算这一不足,提出一种有界 CP-ABE 算法[3].该算法与以往算法显著不同是:密文与访问树关联,密钥和访问树关联,这使得算法在d-BDH假设条件下支持“AND”和“OR”运算,但也使得该算法过于复杂而无法实施.Bobba等人[4]对基于属性的算法进行扩展,提出了密钥策略的基于属性集的加密算法CP-ASBE(ciphertext-policy attribute-set based encryption).与 CP-ABE 不同的是:分配给用户的数据集的元素本身可以不是属性而是属性子集;支持给用户分配多个一样的属性但这些属性分属不同的属性子集(同一属性子集不能出现相同的属性).与 CP-ABE 相比,CP-ASBE 能够处理更复杂的用户属性结构和访问控制逻辑,因而更加灵活,应用价值更大.但无论是用户属性的分配、访问结构树的构建,还是解密操作,该算法都过于复杂,特别是当用户属性结构深度超过 2 时,方案的复杂程度会大为增加.2011 年,Water 等人[5]发现,KP-ABE(key-policy attribute-based encryption)方案之所以较容易实现在 d-BDH 假设条件下的安全,主要原因在于密文和属性关联使得仅根据是否在密文属性集中就能实现属性公开参数的不同处理;而在 CP-ABE 中,密文关联的是复杂得多的访问策略(同一属性可能会多次出现),这使得属性公开参数的区别处理变得十分困难.鉴于此,Bethencourt 等人的方案[1]的安全性基于 GGM(generic group model),而 Goyal 等人[3]的方案需要将 CP-ABE 转变为 KP-ABE 才能实现 d-BDH 假设条件下的安全.考虑到任何访问策略都可以表示成树,而任何树都可以转变为 LSSS(linear secret sharing scheme)[6,7],他们提出基于 LSSS 的 CP-ABE 方案.在他们方案中,使用矩阵实现访问策略和秘密共享,矩阵的每一行都和一

个属性关联.只有满足访问策略的用户私钥才能恢复秘密共享数,进而解密密文.该方案的安全性基于d-BDH假设,在效率上也有较大的提升[8].2013年,Balu等人[9]针对Water等人的方案存在访问策略中属性重复需要专门方法处理且重复出现次数有限制这一问题,提出在CP-ABE方案中用线性整数秘密共享方案LISS(linear integer secret scheme)代替LSSS并给出了构建共享矩阵的3个原则.LISS和LSSS一样能够表示任意访问策略,但其不是在有限群上而是在整数区间上实现秘密共享且效率比LSSS更高.其方案的安全性同样基于d-BDH假设.2010年,Yu等人考虑到:在ABE的多数实际应用场景中,数据存储服务器一般是半可信的且始终在线,将代理加密技术引入到CP-ABE中,提出利用外包环境中的服务器来辅助用户属性撤销的CP-ABE方案[10].该方案在属性的状态上效仿Cheung等人的方案[2],要求用3倍于属性空间的数来描所有属性的正、负、不相关3种状态,要求用户存储的密钥子项数量等于属性空间大小,消耗了大量空间,增加了用户密钥管理负担.文献[11]针对已有的直接撤销模式的CP-ABE方案基本都是基于身份进行撤销即通过撤销用户所拥有的全部属性来撤销用户密文访问权限而无法实现对某一个或几个属性撤销的问题,在合数阶双线性群上,基于双系统加密的思想,提出一种支持完全细粒度属性撤销的CP-ABE方案并证明该方案在标准模型下是安全的.该方案建立在合数阶群上,增加了方案的复杂性;公钥参数数量随用户数量线性增加也是其不足.2011年,Hur等人针对授权机构可以解密任何密文的问题,他们将两方计算2PC(two party computation)协议引入到密钥的生成过程中,为每个用户生成标志参数;授权机构和数据存储中心基于该协议生成密钥生成参数,两个机构分别生成部分密钥子项[12].这一改变使得授权机构和数据存储中心都无法单独生成用户密钥,进而也就无法解密任一密文.文献[13]结合云计算的特点对CP-ASBE进行了改进,提出一种基于云计算环境的分层的CP-ASBE方案,即CP-HASBE(ciphertext-policy hierarchical attribute-set based encryption).在密钥管理和密钥分配上,该方案没有给出效率较高的方法,密钥分配非常麻烦,用户需要自己管理的密钥数据较多.在云计算的利用上,该方案只利用了云的存储能力,而没有利用云的计算能力.2014年,文献[14]提出一种面向云中大数据的支持访问控制策略动态更新的CP-ABE.该文献针对3种类型的访问策略即单调的访问控制树、基于线性秘密共享方案的访问结构和一般的具有门限功能的访问控制树,分别设计了策略动态更新算法.该方案之所以能够实现策略的动态更新,要点之一是加密方保存了数据加密时产生的随机数,而这些随机数正是新旧策略联系的纽带.该方案的另一大优势是充分利用了云计算服务器的存储能力和计算能力.但是,为实现动态策略更新要求保存大量随机数给数据所有者带来很大负担,特别是在文件较多的情况下.

从上面的分析不难发现,已有的密文共享还存在诸多问题,用户客户端计算量过大、密钥保存量过多和只支持基于属性的共享而不支持基于身份的共享是其中3个比较突出的问题.

2 面向公有云的安全共享框架

2.1 安全共享机制

针对用户客户端计算量过大、密钥存储量过多的问题,将授权中心标识符作为特殊属性引入到用户属性中.为授权中心标识符生成密钥子项,该子项利用安全信道发送给用户秘密保存.改造访问结构树:根节点操作符为“AND”,左子树为反映访问逻辑表达式的属性树或用户身份标识符对应的叶子节点,而根节点的右子树为叶子节点,与授权中心标识符对应.

改造后,将主要的计算和存储迁移到云端,除授权中心标识外所有用户属性对应的密钥子项都存储在云端.加密时,客户端只负责近一半密文子项的生成,另一半则由云服务器完成.解密时,除授权中心标识符对应的密文子项需要由用户客户端解密完成外,其他解密工作都外包给云服务器.所有密文子项也全部存储在云服务器上.由于缺少授权中心标识符对应的密钥子项这一关键数据,云服务器无法解密存放在其上的任何密文数据.

针对已有CP-ABE方案的密文共享只能基于属性而无法针对个人用户,将身份标识符作为特殊属性加入到用户个人属性中.在用户私钥分配时,给用户身份标识符分配相应的密钥子项.在创建访问结构时,利用用户属性或身份标识符构建访问结构树.构建访问结构树时,如果只使用用户属性,则和以前基于属性的密文共享方式一样,实现的是基于角色的访问控制;如果只使用用户标识符属性,就属于基于个人身份的共享,实现的是基

于身份的访问控制;如果既有用户属性又有个人属性,则属于既包含属性授权又包含个人授权的混合访问授权.

2.2 面向公有云的密文共享框架

在面向公有云的密文共享框架中,主要包括授权中心、云服务器、数据所有者和用户,如图 1 所示.授权中心位于企业可信域中,负责用户密钥的生成、分配、更新和撤销.云服务器位于云域,负责密文和用户私钥的存储.用户,可能在可信域中也可能在不可信域中,或作为数据所有者,利用客户端加密数据后上传到云端保存;或作为数据共享者,从云端读取并解密授权共享密文.在本文中,假设云服务器是半可信的(会忠实执行用户要求的操作,但也可能利用自己所拥有的数据甚至和恶意用户合谋去解密非授权密文).

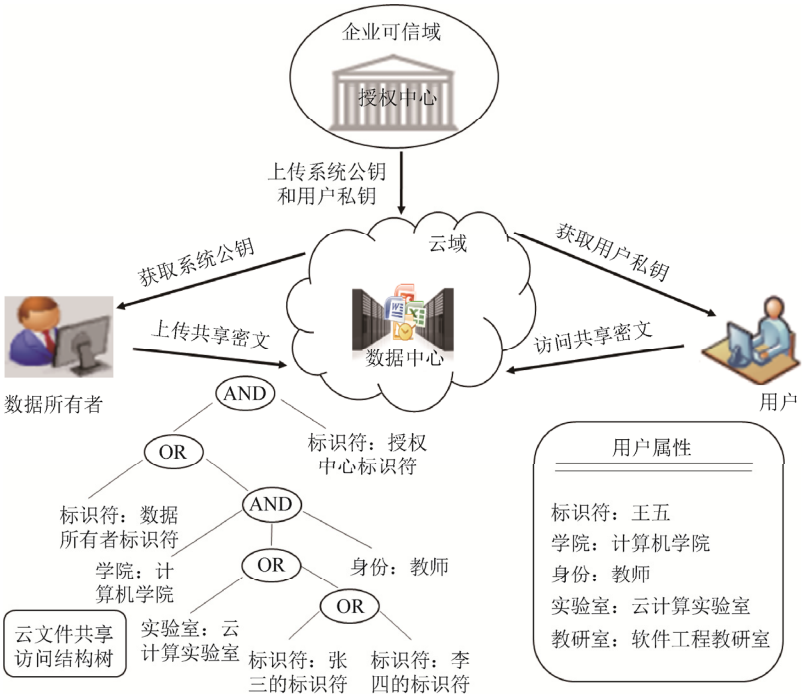

Fig.1 Ciphertext-sharing framework for public cloud

图 1 面向公有云的密文共享框架

先用对称加密算法对共享文件的数据进行加密,再采用 ABE 算法对数据加密密钥进行加密.

在已有的密文共享方案中,几乎都采取的是文件加密和密文共享同时在用户客户端进行的方式,而在实际的文件共享中,无论文件是否需要加密,都是先要将文件上传到存储服务器,再进行文件共享.基于该事实,将文件共享分成前后相随的两个阶段:文件安全存储和密文共享,下面分别进行说明.

(1) 文件安全存储

敏感文件在共享前,需要先存储到云端.为确保文件数据的机密性,需要对数据进行加密.这里,也采用 CP-ABE 进行加密,访问结构如图 2 所示,uid_i 和 uid_0 分别为数据所有者标识符和授权中心标识符.在共享前,由于能够匹配密文访问结构树的只有数据所有者,故只有他能够解密数据.

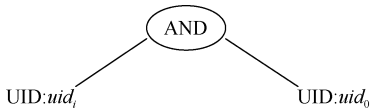

Fig.2 Access tree of secure storage of cloud file

图 2 云文件安全存储访问结构树

(2) 密文共享

为实现共享,数据所有者首先根据访问策略或访问逻辑表达式构建文件共享访问结构树;然后,根据共享访问结构树计算各叶子节点的秘密共享数;最后计算每个叶子节点对应的两个密文子项中的一个.将文件共享结构树和叶子节点对应的一半密文子项以及用于计算另一半密文子项的相关数据上传到云端.

云服务器根据安全存储访问结构树和文件共享结构树构造云文件共享访问结构树(如图 3 所示).基于安全存储访问结构树和文件共享访问结构树,构造云文件访问结构树的具体过程为:生成“OR”节点,用其替代安全存储访问结构树中代表数据所有者的用户标识符属性对应的节点,而用户标识符属性对应的叶子节点成为这个新建“OR”节点的左孩子,文件共享访问结构树则成为这个新建“OR”节点的右子树.图 3 中的文件共享访问结构树表达的是能够访问密文的用户:要么是计算机学院云计算实验室的教师(基于属性的访问控制),要么是身份标识符 UID 的值为 uid_j 或 uid_k 的计算机学院教师(基于身份的访问控制).云文件共享访问结构树除包含文件共享访问结构树的逻辑外,还允许数据所有者访问自己以密文形式存储在云端的数据.

云服务器还要利用数据所有者上传的相关数据,完成另一半密文子项的计算.

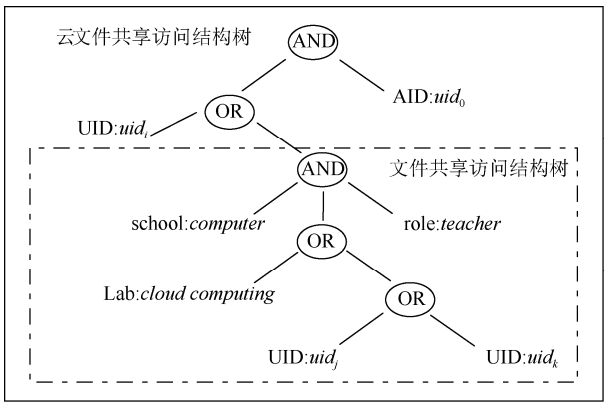

Fig.3 Access tree of cloud file sharing

图 3 云文件共享访问结构树

3 面向公有云的安全文件共享方案

和一般的 ABE 方案一样,面向公有云的安全文件共享方案也包括初始化、私钥生成、加密和解密 4 个模块.

(1) 初始化:*Setup*(A,A_{ID},d)

d 为安全参数.授权中心选择一个阶为大素数 p 的双线性群 G_0 和 G_1,记 G_0 的生成元为 g,对应的双线性映射为 $e: G_0 \times G_0 \to G_1$.定义系统所需的属性空间 $A = \{a_1, a_2, \ldots, a_n\}$,和用户身份空间 $A_{ID} = \{uid_0, uid_1, uid_2, \ldots, uid_m\}$ (uid_0 为授权中心标识符),定义一个哈希函数 $H: \{0,1\}^* \to G_0$.最后随机选择 $\alpha, \beta \in Z_p^*$,将如下系统公钥信息发往云服务器并公开:

$$PK = \{G_0, g, h = g^{\beta}, e(g,g)^{\alpha}\}.$$

授权中心秘密保存主私钥: $MK = \{g^{\alpha}, \beta\}$.

(2) 生成用户私钥:*KeyGen*(w,MK)

授权中心为用户 U_i 分配一对公私钥 (PUK_{uid_i},SEK_{uid_i}) ,将 PUK_{uid_i} 发往云服务器并公开, SEK_{uid_i} 由用户秘密保存.

设用户 U_i 对应的属性集合为 $w' \subseteq A$,令用户属性身份集 $w = w' \cup \{uid_0, uid_i\}$ (分别为授权中心和用户的标识符).随机选择 $r \in Z_p^*$;对于每个元素 $a_j \in w$,随机选择 $r_j \in Z_p^*$.生成的私钥如下:

$$SK_{uid_i}=(SK^{(1)}=g^{\frac{\alpha+r}{\beta}},\forall a_j\in w: SK_{a_j}^{(2)}=g^r\cdot H(a_j)^{r_j}, SK_{a_j}'^{(3)}=g^{r_j}).$$

将 uid_0 对应的私钥子项 $(SK_{uid_0}^{(2)}, SK_{uid_0}'^{(3)})$ 通过安全信道发送给用户秘密保存,将其他私钥子项保存在云端的用户私钥表中.

(3) 加密:*Encrypt*(*m*,*T*,*PK*)

数据所有者 U_i 构造安全存储访问结构树 *T*(如图 2 所示).在需要共享时,将 *T* 改造为云文件共享访问结构树(如图 3 所示).

① 文件安全存储

设需要安全存储的文件为 *f*.用户客户端使用对称加密算法(如 AES)和数据加密密钥 k_f 加密文件得数据密文 $E_{k_f}(f)$.为安全存储访问结构树 *T* 根节点随机选择一个一元一次多项式 $Q_r(x)$和 $s\in Z_p^*$,使得 $s=Q_r(0)$且 $Q_r(1)$和 $Q_r(2)$分别为其左右子树根节点的取值.然后计算数据加密密钥的密文如下:

$$CT=(T,\tilde{C}=k_f\cdot e(g,g)^{\alpha s}, C=h^s, C_{uid_i}=g^{Q_r(1)}, C'_{uid_i}=H(uid_i)^{Q_r(1)},$$
$$C_{uid_0}=g^{Q_r(2)}, C'_{uid_0}=H(uid_0)^{Q_r(2)}).$$

用户客户端将 $E_{k_f}(f)$、*CT* 和 $Q_r(1)$的密文 $E_{PUK_{uid_i}}(Q_r(1))$ 一起上传保存到云端.

② 密文共享

在文件共享阶段,数据所有者根据访问逻辑表达式构建文件共享访问结构树 *T*′.为在访问结构树 *T*′中每个操作符为“AND”的非叶子节点 *x* 随机选择一个一元多项式函数 $Q_x(x)$.令 *T*′的根节点对应的秘密共享数为 $Q_r(1)$(用 SEK_{uid_i} 解密求得).对于任意节点 *x* 且其父节点操作符为“AND”, $Q_x(0)=Q_{parent(x)}(index(x))$, *index*(*x*)为 *x* 在兄弟中的序号(从左往右进行编号);对于任意节点 *x* 且父节点操作符为“OR”, $Q_x(0)=Q_{parent(x)}(0)$.按照以上方式从上至下最终可以为每个叶子节点赋值 $\{s_{l_i}\}_{l_i\in L_{T'}}$, L_T 表示 *T*′叶子节点的集合.随机选择 $s'\in Z_p^*$,计算用户共享密文子项:

$$CT'=(T',g^{s'},\forall l\in L_{T'}: C_l'=H(att(l))^{s_l}).$$

其中,*att*( )用来求叶子节点对应的属性.将 *CT*′和 $\{s_{l_i}/s'\}_{l_i\in L_{T'}}$ 上传到云端.

云服务器先利用 *T*′将 *T* 改造为云文件共享访问结构树.计算 *T*′对应的云共享密文子项:

$$CT''=(T',\forall l\in L_{T'}: C_l=(g^{s'})^{s_l/s'}=g^{s_l}).$$

k_f 完整的共享密文如下:

$$CT=(T,\tilde{C}=k_f\cdot e(g,g)^{\alpha s}, C=h^s, C_{uid_i}=g^{Q_r(1)}, C'_{uid_i}=H(uid_i)^{Q_r(1)}, C_{uid_0}=g^{Q_r(2)},$$
$$C'_{uid_0}=H(uid_0)^{Q_r(2)}, \forall l\in L_{T'}: C_l=g^{s_l}, C_l'=H(att(l))^{s_l}).$$

(4) 解密:*Decrypt*(*CT*,*SK*)

解密时,先在云服务器上完成部分解密,再在用户客户端上完成最后的解密工作.

① 云服务器解密

共享情况下解密基于 *T*′,叶子节点和非叶子节点有着不同的解密算法.

对于叶子节点 $x(a_j=att(x))$,其解密算法为

$$\begin{aligned}DecNode(x)&=\frac{e(SK_{a_j},C_x)}{e(SK'_{a_j},C'_x)}\\&=\frac{e(g^r\cdot H(a_j)^{r_j},g^{s_x})}{e(g^{r_j},H(att(x))^{s_x})}\\&=e(g,g)^{rs_x}\\&=e(g,g)^{rQ_x(0)}.\end{aligned}$$

对于非叶子节点 *x*,解密算法为

$$\begin{aligned}DecNode(x)&=\prod_{z\in S_x}DecNode(z)^{\Delta_{i,S'_x}(0)},\ \text{where}^{i=index(z)}_{S'_x=\{index(z):z\in S_x\}}\\&=\prod_{z\in S_x}(e(g,g)^{r\cdot Q_z(0)})^{\Delta_{i,S'_x}(0)}\\&=\prod_{z\in S_x}(e(g,g)^{r\cdot Q_{parent(z)}(index(z))})^{\Delta_{i,S'_x}(0)}\\&=\prod_{z\in S_x}e(g,g)^{r\cdot Q_x(i)\cdot\Delta_{i,S'_x}(0)}\\&=e(g,g)^{r\cdot Q_x(0)}.\end{aligned}$$

其中,$\Delta_{i,S}(x)=\prod_{j\in S,j\neq i}\frac{x-j}{i-j}$.

T'的根节点的解密为

$$DecNode(root_{T'})=e(g,g)^{r\cdot Q_{rootT'}(0)}=e(g,g)^{r\cdot Q_r(1)}.$$

安全存储情况下只对用户标识符对应的叶子节点解密,其解密算法同于上面叶子节点解密算法,解密结果为: $e(g,g)^{r\cdot Q_r(1)}$.

② 用户客户端解密

数据所有者或共享用户将密文子项 $\tilde{C}$ 、C、 $C_{uid_0}=H(uid_0)^{Q_r(2)}$ 、 $C'_{uid_0}=g^{Q_r(2)}$ 和解密得到的 $e(g,g)^{r\cdot Q_r(1)}$ 从云端下载到本地.用私钥子项 $SK^{(2)}_{uid_0}$ 和 $SK'^{(3)}_{uid_0}$ 解密 uid_0 对应叶子节点得到 $e(g,g)^{r\cdot Q_r(2)}$,再利用非叶子节点解密算法计算: $DecNode(root)=e(g,g)^{r\cdot Q_r(0)}=e(g,g)^{rs}$.解密数据加密密钥 k_f 的算法如下:

$$Decrypt(CT,SK)=\frac{\tilde{C}\cdot DecNode(root)}{e(C,SK^{(1)}_i)}=\frac{k_f\cdot e(g,g)^{\alpha s}\cdot e(g,g)^{rs}}{e(g^{\beta s},g^{\frac{\alpha+r}{\beta}})}=k_f.$$

用户再利用 k_f 将文件密文进行解密以恢复文件 f.

4 安全性与性能分析

4.1 安全性分析

采用和 Bethencourt 等人提出原始方案(下面称作 BSW 方案)进行对比来分析所提出方案的安全性.和 BSW 方案相比,所提出方案向云服务器多暴露了所有用户除授权中心标识符外其他所有属性对应的私钥子项.下面通过两个挑战游戏证明方案能够确保数据的机密性.

游戏 0:BSW 方案采用的安全游戏.

游戏 1:基于本文所提出方案开展的游戏.与 BSW 方案相比,攻击者还拥有所有用户的除授权中心标识符外其他所有属性对应的私钥子项;攻击者还可以从授权中心获得指定属性身份集(不满足密文访问策略)的私钥.

引理.攻击者赢得游戏 1 和赢得游戏 0 的优势是一样.

证明:在安全游戏中,攻击者(含云服务器在内的恶意用户)向挑战者(数据所有者)提交两个等长的消息 m_0 和 m_1.挑战者随机从{0,1}中选择一个数(记为 b),加密 m_b 并将其密文 E 发回给攻击者.攻击者根据自身拥有的信息猜测 E 是 m_0 还是 m_1 的密文.设攻击者的猜测值为 b',如果猜中即 b=b',攻击者赢得游戏.如果攻击者每次猜中的概率与猜不中的概率的差值可忽略不计,则称为以可忽略优势赢得游戏即可抵御选择明文攻击;否则,称作以不可忽视优势赢得游戏即不可抵御选择明文攻击.

用多项式时间算法 A 和 B 分别表示游戏 1 和游戏 0 的攻击者.

将游戏 1 表示为 $b\leftarrow A(E,PK,\{\{sk_{i,j}\}_{j\in w_i/uid_0}\}_{1\leqslant i\leqslant m},\{SK_i\}_{1\leqslant i\leqslant q})$ (其中,m 为用户数,w_i 为用户 i 的属性身份集,q 为用户密钥查询次数).基于游戏 0 和算法 A,构建算法 B 如下:

$$B(E,PK,\{SK'_i\}_{1\leqslant i\leqslant m+q}).$$

Begin

假设前 m 次密钥查询使用的属性身份集刚好对应于 m 个用户的属性身份集(不包括数据所有者身份标识符);

令 $\{sk_{i,j}\}_{j\in w_i/uid_0}=SK'_i, i\in[1,m]$,有 $\{\{sk_{i,j}\}_{j\in w_i/uid_0}\}_{1\leqslant i\leqslant m}=\{SK'_i\}_{1\leqslant i\leqslant m}$;

令 $SK_i=SK'_{m+i}, i\in[1,q]$,有 $\{SK_i\}_{1\leqslant i\leqslant q}=\{SK'_k\}_{m+1\leqslant k\leqslant m+q}$;

$$b\leftarrow A(E,PK,\{\{sk_{i,j}\}_{j\in w_i/uid_0}\}_{1\leqslant i\leqslant m},\{SK_i\}_{1\leqslant i\leqslant q});$$

End

由算法 B 可知,算法 A 赢得游戏 1 的优势并不比算法 B 高,当然也不会比 B 低(因为知道更多信息),故算法 A 和算法 B 有着一样的优势.因此,攻击者赢得游戏 1 和赢得游戏 0 的优势是一样. □

定理.本文所提出的方案,在一般群模型和随机预言模型下可抵御选择明文攻击.

证明:如引理所证,攻击者赢得游戏 1 和赢得游戏 0 的优势是一样的,而 Bethencourt 等人已经证明在一般群模型和随机预言模型下,攻击者赢得游戏 0 的优势可忽略不计,故在同样条件下,攻击者赢得游戏 1 的优势同样可忽略不计.所以本文所提出的方案,即使多个用户合谋或云服务器与用户合谋,在一般群模型和随机预言模型下可抵御选择明文攻击. □

4.2 性能分析

下面从计算和存储两个方面讨论所提出方案的性能.

(1) 计算性能

由于数据处理时,最耗费时间的运算依次是双线性运算 B 和指数运算 E,所以用这 2 个指标来衡量性能.

加密安全存储时,用户客户端只需要计算 $\tilde{C}, C, C_{uid_i}, C'_{uid_i}, C_{uid_0}$ 和 C'_{uid_0} 这 6 个密文子项,用户客户端的计算代价为:$6E$.密文共享时,用户客户端需要为 T'每个叶子节点执行 2 次指数运算($H(\ )$需要 1 次),计算 $g^{s'}$需要执行 1 次指数运算,计算代价为 $(2|T'_L|+1)E=(2|T_L|-4+1)E=(2|T_L|-3)E$ ($|T_L|$为 T 的叶子节点个数),比 BSW 方案少 $|T_L|+2E$.云服务器为 T'每个叶子节点执行一次指数运算计算代价为 $|T'_L|E=(|T_L|-2)E$.

解密时,客户端解密一次叶子节点(与授权中心标识符对应)的计算代价为 $2B$.解密云共享文件访问结构树根节点的代价为 $2E$,完成最后的解密代价为 $1B$,故用户客户端解密总的计算代价为 $3B+2E$.数据所有者解密时,云服务器只需对一个叶子节点即 uid_i 对应的密文子项进行解密,计算代价为 $2B$.共享者解密时,在非叶子节点对应逻辑操作都为“AND”的情况下,云服务器解密文件共享访问结构树 T'的每个节点(根除外)都要参与一次指数运算,故云服务器的计算代价为 $2(|T_L|-2)B+(|T|-3)E$. BSW 方案与本文所提出方案在用户客户端的计算代价比较见表 1.

Table 1 Comparison of user client computing overhead

表 1 用户客户端计算开销对比

方案	加密时间	解密时间
BSW 方案	$5E+3\|T_L\|E$	$(1+2\|T_L\|)B+(\|T\|+1)E$
本文方案	$6E+(2\|T_L\|-3)E$	$3B+2E$

(2) 存储性能

在所提出的方案中,无论是用户私钥还是用户数据都以外包形式存放在公有云上,用户唯一需要安全保存的是授权中心分配给他的与授权中心标识对应的私钥子项,大大减轻了密钥管理的负担.云服务器负责管理其他用户私钥子项,所需存储空间大小与用户所拥有的属性数量成线性关系.对某一个文件,采用本方案需要的密文存储空间和采用基本 CP-ABE 方案需要的存储空间相当,只是多了两个密文子项(需要 4 个群元素的存储空间),分别和用户标识符与授权中心标识符对应.

5 实验分析

利用双线性对加密库(http://crypto.stanford.edu/pbc/)和 CP-ABE 开发工具包(http://acsc.csl.sri.com/cpabe/),基于本文所提出方案在 Hadoop 环境下实现了一个面向公有云的密文共享系统并进行性能实验.实验使用的虚拟机配置为:1 个 Intel(R) Xeon(R) CPU(E5-2620 2.0GHZ);内存 1GB;系统 CentOS6.5 64 位.所有算法采用 Java 语言编写,双线性映射和幂运算等有关椭圆曲线加密的操作均来自双线性对加密库 JPBC.所实现的系统采用 160 位椭圆曲线群,椭圆曲线为 512 位有限域上的超奇异椭圆曲线 $y^2=x^3+x$.

加密和解密实验分别针对 BSW 方案和本文方案各进行 20 轮,每轮使用相同大小的数据和相同的文件访问结构树(为方便比较,访问结构树的非叶子节点对应的逻辑运算均取为“AND”);每轮实验进行 20 次,取 20 次实验的平均值为最终实验结果.访问结构树叶子节点数量依轮次递增,前 10 轮实验加密时间对比见表 2.

Table 2 Comparison of experimental data of encryption time

表 2 加密时间实验数据对比

访问策略树的叶子节点数	BSW 方案(ms)	本文方案(ms)
2	218.45	380.45
4	422.05	390.15
6	617.35	543.05
8	724.50	644.30
10	924.85	779.55
12	1088.25	945.20
14	1246.40	1068.10
16	1518.80	1180.10
18	1582.65	1315.70
20	1959.85	1536.10

从实验数据可以看出,BSW 方案客户端和本文方案加密时间都随着叶子节点数量不断增加,加密时间与叶子节点数量呈线性关系(如图 4 所示),但本文方案的加密时间更短.随着叶子节点数量增加,二者加密时间的差距增大即本文方案节约的时间增多,这和上一节所作的性能分析结果是一致的.

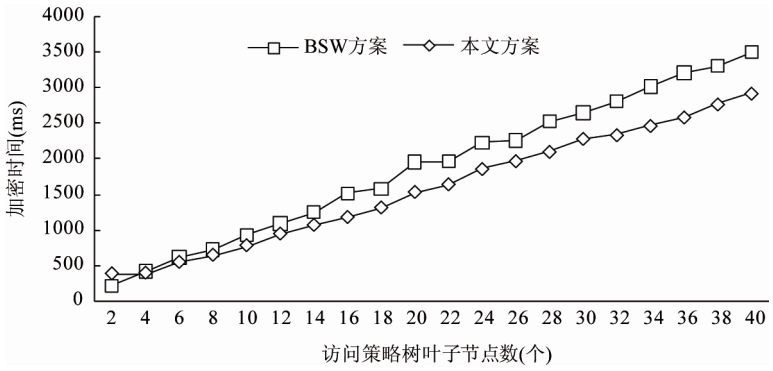

Fig.4 Comparison chart of encryption time

图 4 加密时间的对比图

解密前 10 轮实验解密时间(客户端)对比见表 3.

Table 3 Comparison of experimental data of decryption time

表 3 解密时间实验数据对比

用户与访问结构树属性匹配数量	BSW 方案(ms)	本文方案(ms)
2	75.70	60.95
4	103.35	44.45
6	159.20	46.30
8	203.90	42.35
10	265.20	41.85
12	326.80	41.65
14	373.05	41.30
16	466.75	43.75
18	480.30	41.35
20	590.70	45.20

从实验结果(如图 5 所示)可以看出,BSW 方案解密时间随用户与访问结构树匹配属性数增加而线性增加,而本文方案客户端解密时间比较稳定,在 42ms 左右,这使得一般个人电脑、平板电脑和手机(和实验虚拟机配置相当)也能较快地解密共享密文.

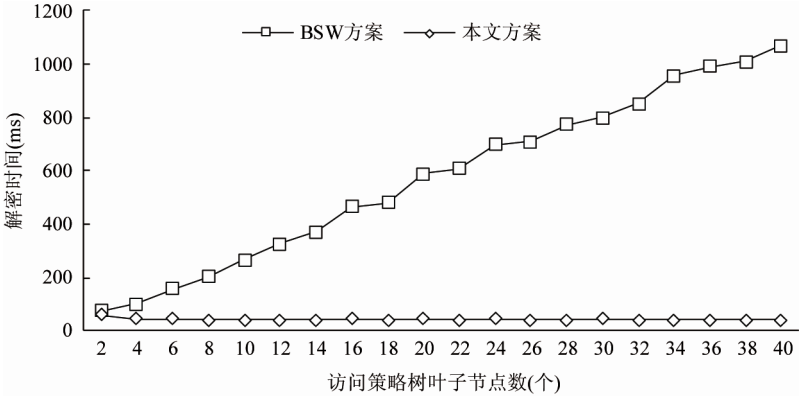

Fig.5 Comparison chart of decryption time

图 5 解密时间的对比图

6 结束语

由于实现了细粒度的访问控制和在密文共享上具有“一对多”的显著优势,加之和企业信息系统广泛采用的基于角色访问控制方法相似,CP-ABE 可能成为企业外包云密文数据共享的重要访问控制方法.然而,已有的基于 CP-ABE 的密文共享方案存在客户端计算量过大、用户管理密钥过多、不支持个人共享等问题.考虑到公有云强大的计算能力和存储能力,将公有云引入到密文共享方案的设计之中,提出一种面向公有云的安全文件共享框架,基于该框架设计一种面向公有云的密文共享方案.该方案将绝大多数计算和存储都外包给公有云,用户只需保存两个空间占用很小的私钥子项且客户端只需进行少量计算就能完成共享文件的解密.安全性分析表明,该方案在不仅能够对抗恶意用户的合谋攻击,还能在一般群模型和随机预言模型下对抗选择明文攻击.

References:

[1] Bethencourt J, Sahai A, Waters B. Ciphertext-policy attribute-based encryption. In: Proc. of the 2007 IEEE Symp. on Security and Privacy. Washington: IEEE Computer Society, 2007. 321−334. [doi:10.1109/SP.2007.11]

[2] Cheung L, Newport C. Provably secure ciphertext policy ABE. In: Proc. of the 14th ACM conference on Computer and Communications Security. New York: ACM Press, 2007. 456−465. [doi:10.1145/1315245.1315302]

[3] Goyal V, Jain, Pandey O, Sahai A. Bounded ciphertext policy attribute based encryption. In: Proc. of the 35th Int’l Colloquium on Automata, Languages and Programming. Berlin: Springer-Verlag, 2008. 579–591.

[4] Bobba R, Khurana H, Prabhakaran M. Attribute-sets: A practically motivated enhancement to attribute-based encryption. In: Proc. of the ESORICS. Berlin: Springer-Verlag, 2009. 587–604.

[5] Waters B. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In: Proc. of the Public Key Cryptography (PKC 2011). Berlin: Springer-Verlag, 2011. 53−70. [doi:10.1007/978-3-642-19379-8_4]

[6] Beimel A. Secure schemes for secret sharing and key distribution [Ph.D. Thesis]. Haifa: Israel Institute of Technology, 1996.

[7] Lewko A, Waters B. Decentralizing attribute-based encryption. In: Paterson K, ed. Advances in Cryptology—EUROCRYPT 2011. Berlin: Springer-Verlag, 2011. 568–588. [doi:10.1007/978-3-642-20465-4_31]

[8] Waters B. Efficient identity-based encryption without random oracles. In: Proc. of the EUROCRYPT 2005. Berlin, Heidelberg: Springer-Verlag, 2005. 114–127. [doi:10.1007/11426639_7]

[9] Balu A, Kuppusamy K. An expressive and provably secure ciphertext-policy attribute-based encryption. Information Sciences, 2014,276(4):354–362.

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

[11] Wang PP, Feng DG, Zhang LW. CP-ABE scheme supporting fully fine-grained attribute revocation. Ruan Jian Xue Bao/Journal of Software, 2012,23(10):2805–2816 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4184.htm [doi: 10.3724/SP.J.1001.2012.04184]

[12] Hur J. Improving security and efficiency in attribute-based data sharing. IEEE Trans. on Knowledge & Data Engineering, 2013, 25(10):2271–2282.

[13] 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(2):743–754.

[14] Yang K, Jia XH, Ren K, Xie RT, and Huang LS. Enabling efficient access control with dynamic policy updating for big data in the cloud. In: Proc. of the INFOCOM 2014. Toronto: IEEE Press, 2014. 2013–2021. [10.1109/INFOCOM.2014.6848142]

附中文参考文献:

[11] 王鹏翩,冯登国,张立武.一种支持完全细粒度属性撤销的 CP-ABE 方案.软件学报,2012,23(10):2271–2282. http://www.jos.org.cn/1000-9825/4184.htm [doi: 10.3724/SP.J.1001.2012.04184]

罗王平(1993－),男,四川广安人,硕士生,主要研究领域为信息安全,云计算,大数据安全.

冯朝胜(1971－),男,博士,教授,CCF 高级会员,主要研究领域为网络与信息安全,云计算,大数据安全.

秦志光(1956－),男,博士,教授,博士生导师,CCF 杰出会员,主要研究领域为信息安全,分布式计算.

袁丁(1967－),男,博士,教授,主要研究领域为密码学,信息安全.

廖娟平(1990－),女,硕士,主要研究领域为数据安全.

刘霞(1978－),女,讲师,主要研究领域为网络安全.