Software Engineering and Applications 软件工程与应用, 2020, 9(2), 216-227
Published Online June 2020 in Hans. http://www.hanspub.org/journal/sea
https://doi.org/10.12677/sea.2020.93025

Hans汉斯

CP-ABE Access Control Scheme Based on Multi-Authorities in Cloud Storage

Yan Huang, Xiaoling Wu, Jie Ling

School of Computer, Guangdong University of Technology, Guangzhou Guangdong
Email: 1370438664@qq.com

Received: Jun. 7th, 2020; accepted: Jun. 17th, 2020; published: Jun. 24th, 2020

Abstract

Aiming at the security and overhead problems of the traditional Ciphertext Policy Attribute-Based Encryption (CP-ABE) scheme in the process of cloud storage access control, a CP-ABE access control scheme based on multi-authorities in cloud storage is proposed. The trust calculation is carried out on the basis of CP-ABE to judge the credibility of users, global key and attribute key are generated for the legal users by using multi-authorities, meanwhile, the proxy server is used to undertake most of the work of decryption calculation and the storage of user's attribute key. The security analysis shows that the scheme is chosen plaintext attack security and can resist the collusion attack, and the performance analysis shows that it is efficient and can reduce the overhead of the user.

Keywords

CP-ABE, Cloud Storage, Access Control, Multi-Authorities, Trust

云存储中基于多授权中心的CP-ABE访问控制方案

黄　艳，吴晓鸽，凌　捷

广东工业大学计算机学院，广东　广州
Email: 1370438664@qq.com

收稿日期：2020年6月7日；录用日期：2020年6月17日；发布日期：2020年6月24日

摘　要

针对传统的密文策略属性基加密(Ciphertext Policy Attribute-Based Encryption, CP-ABE)方案在云存储访问控制过程中存在的安全和开销问题，提出一种云存储中基于多授权中心的CP-ABE访问控制方案。

文章引用: 黄艳, 吴晓鸽, 凌捷. 云存储中基于多授权中心的 CP-ABE 访问控制方案[J]. 软件工程与应用, 2020, 9(2): 216-227. DOI: 10.12677/sea.2020.93025

在CP-ABE的基础上对用户进行信任计算判断用户的可信度，利用多授权中心为合法用户生成全局密钥与属性密钥，同时使用代理服务器承担大部分解密计算与存储用户属性密钥的工作。安全性分析表明本方案是选择明文攻击安全的并且能够抵抗合谋攻击，性能分析表明本方案具有高效性，能减少用户端的开销。

关键词

CP-ABE，云存储，访问控制，多授权中心，信任

1. 引言

云计算[1]是一种新的计算范式，为用户提供随需部署、动态优化与恢复、随需计费以及随时随地在互联网上可获得的海量存储和计算资源等服务[2]。云计算的快速发展和用户数据的爆炸式增长使云存储成为云环境中不可或缺的一部分[3]。在通过云存储实现数据存储与共享的过程中，数据拥有者一旦将数据上传至云端，将会失去对数据存储与访问的物理控制和直接控制[4]。因此，云存储中最重要的安全问题是云向未授权用户泄露敏感数据和未授权的用户非法访问数据。

由于安全和隐私方面存在的挑战，需要加密来保护敏感信息并且只允许授权用户访问数据。2005 年，Sahai 和 Waters 首次提出属性基加密(Attribute-Based Encryption, ABE)机制[5]。ABE 是一种将属性与访问控制策略、用户和数据相互关联的加密方案，用户如果想要访问云数据，必须拥有访问控制策略所定义的属性。

ABE 根据访问策略嵌入位置的不同，可以分为密钥策略属性基加密(Key-Policy Attribute-Based Encryption, KP-ABE) [6]和密文策略属性基加密(Ciphertext-Policy Attribute-Based Encryption, CP-ABE) [7]。在 KP-ABE 中，只有当与密文关联的属性集满足与用户解密密钥相关联的访问策略时，才能解密密文。而 CP-ABE 允许数据拥有者使用有效的访问结构或策略加密数据并将其存储到云中，用户只有在其属性满足密文的访问结构时才能访问数据。由于 CP-ABE 具有可扩展性、灵活性、支持细粒度访问等特点，成为云计算中一种有效的数据访问控制解决方案[8]。

在复杂多变的云计算环境中，信任管理可用于用户身份安全管理中的可信访问控制以及对非授权用户的身份认证，确保云存储数据服务的安全，在提高云计算的可靠性、可用性和安全性方面发挥着至关重要的作用[9] [10]。而访问控制技术又是解决云安全问题的核心，所以如何将信任与 CP-ABE 结合，实现对云存储数据进行安全高效、灵活和细粒度的访问控制成为云安全问题的重中之重。

2. 相关工作

传统情况下，直接将 CP-ABE 方案应用于云环境中，通常会在访问控制过程中产生较大的计算、存储通信开销问题以及安全性问题。大多数 CP-ABE 方案在解密阶段往往会涉及到双线性配对运算，该运算计算量大、耗时较长导致用户解密效率的低下。为了减少用户的计算开销，文献[11] [12]提出将密钥分发与解密阶段过程中涉及到的所有与访问策略和属性相关的操作分别外包给密钥生成服务器和代理服务器，来降低用户端的计算开销问题。

由于存在单个属性授权机构(Attribute Authority, AA)与用户通过合谋攻击造成密钥泄露的问题，且无法满足大规模分布式应用对不同机构协作的需求。文献[13] [14]依赖于一个中央授权机构(Central Authority, CA)来生成相关的系统参数和用户密钥，并使用 CA 管理和分发密钥。但 CA 在该方案中能够拥有足够的权限去解密密文，当 CA 不可信或遭受恶意攻击的情况下可能会造成密钥泄露，大大地降低了云存储数据的安全性和保密性。

为了解决此问题，Chase 等人最早在文献[15]中提出了多授权机构 CP-ABE 方案，防止用户之间进行合谋攻击，降低单个授权中心的负担和提高系统的效率。但该方案只支持门限算法，加密算法缺乏灵活性。文献[16] [17] [18] [19]通过使用多个属性授权机构对用户属性进行分组来共同生成密钥，从而避免授权中心被恶意攻破或合谋攻击问题的发生。文献[20]通过在访问结构和用户密钥中依赖一个基本属性来解决密钥分发的问题，利用用户全局标识符来抵抗属性授权机构间的合谋攻击，但该方案没有考虑计算量的转移，用户存在较大的计算开销问题。

针对上述问题，本文提出一种云存储中基于多授权中心的 CP-ABE 访问控制方案，利用多个属性授权机构分别管理不同的属性，AA 通过用户所拥有的属性计算用户的信任值并对用户进行身份认证，实现用户密钥的生成与分发。为防止用户、代理服务器与属性授权机构之间的合谋攻击，解密密钥由全局密钥和属性密钥两部分组成，以此增强用户密钥的安全性。同时将属性密钥存储在代理服务器中并将其用于执行外包解密操作，减少用户的计算和存储开销。

3. 预备知识

3.1. 双线性映射

定义 1. 双线性映射[21]：设 G_S 和 G_T 都是阶为素数 p 的循环群，g 为 G_S 的一个生成元，$e: G_S \times G_S \to G_T$ 是一个满足以下性质的双线性映射：

1) 双线性：对任意的 $a,b \in Z_p$，有 $e\left(g^a, g^b\right) = e(g,g)^{ab}$；

2) 非退化性：$e(g,g) \neq 1$，其中 1 为 G_T 中的单位元；

3) 可计算性：对任意的 $g_1, g_2 \in G_S$，存在有效的算法计算 $e(g,g)$。

3.2. 访问结构

定义 2. 访问结构[22]：设属性集 $P = \{P_1, P_2, \cdots, P_n\}$，某个用户的属性集 $A \subseteq \{P_1, P_2, \cdots, P_n\}$ 是集合 S 的一个非空子集，则 n 个属性最多可以定义 2^n 个不同属性集的用户。访问结构 T 是集合 $\{P_1, P_2, \cdots, P_n\}$ 的非空子集，当 $T \subseteq 2^{\{P_1, P_2, \cdots, P_n\}}$ 时，T 作为属性判断条件。对于任意集合 A，有 $A \subseteq T$，那么 A 称为授权集合，否则称为 A 非授权集合。令 T 表示一棵访问树，在 ABE 结构中，每个叶子节点表示一个属性，非叶子节点用门限表示，当且仅当用户的属性集完全符合访问策略 T 才能解密数据。

3.3. 线性秘密共享方案

定义 3. 线性秘密共享方案(LSSS) [23]：参与者集合 P 上的一个密钥共享方案 Π，如果满足以下 2 个条件，则被称为 Z_P 上的线性秘密共享方案。

1) 每个实体的秘密份额构成 Z_P 上的一个向量；

2) 对于每个秘密共享方案 Π，存在一个生成矩阵 $M(l \times n)$，对于矩阵 M 中的每一行 $i = 1,2,\cdots,l$，映射 $\rho: \{1,2,\cdots,l\} \to P$ 把 M 的每一行映射到参与者 $\rho(i)$，ρ 为单射函数。考虑向量 $v = (s, y_2, \cdots, y_n)$，$s \in Z_p$ 是共享密钥，$y_2, \cdots, y_n \in Z_P$ 随机选择用来隐藏 s，Mv 是 l 个秘密份额形成的向量，其中 $\lambda_i = (Mv)$ 表示参与者 $\rho(i)$ 所持有的秘密份额。

LSSS 方案具有线性重构性，假设 Π 是访问策略 T 的一个线性秘密共享，设 $S \in T$ 是一个访问授权集合，定义为 $I = \{i : \rho(i) \in S\}$，如果 $\{\lambda_i\}$ 是对秘密 s 的有效共享份额，那么可以在多项式时间内找到一组常数 $\{\varpi_i \in Z_P\}_{i \in I}$，使等式 $\sum_{i \in I} \varpi_i \lambda_i = s$ 成立。

3.4. 判定双线性问题

定义 4. 判定双线性问题(DBDH 问题) [24]：设 G_S 和 G_T 是两个阶为素数 p 的乘法循环群。设 g 是 G_S 的一个生成元，双线性对 $e : G_S \times G_S \to G_T$。挑战者随机选择 $a,b,c \in Z_p$ 和 $R \in G_T$，并计算 g^a, g^b, g^c。已知 $\left(g, g^a, g^b, g^c\right)$ 判断给定的一个元组是一个有效元组 $\left(g, g^a, g^b, g^c, e(g,g)^{abc}\right)$ 还是一个随机元组 $\left(g, g^a, g^b, g^c, R\right)$。

4. 系统模型

本文提出的系统模型由云服务器(Cloud Server, CS)、代理服务器(Proxy Server, PS)、属性授权机构(Attribute Authorities, AAs)、数据拥有者(Data Owner, DO)、用户(Data Users, DU)5 个实体组成。如图 1 所示。

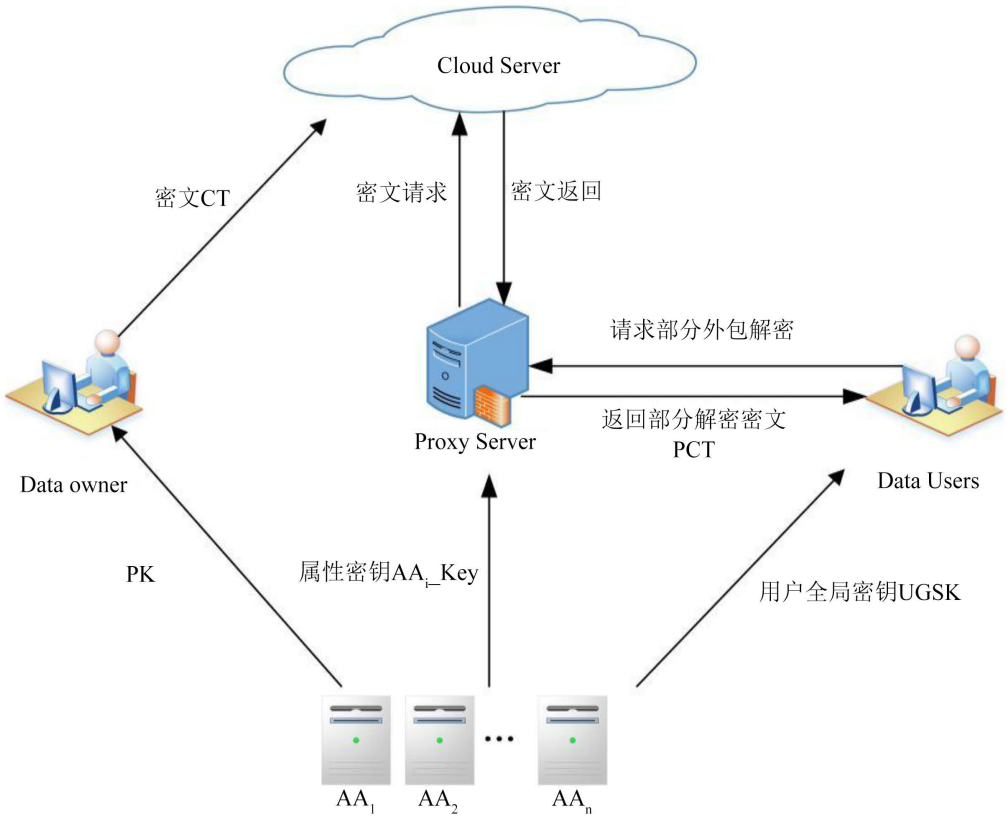

Figure 1. System model
图 1. 系统模型

云服务器是半可信的，能够为数据所有者和代理服务器分别提供数据存储服务与密文访问请求服务。

代理服务器也是半可信的，用来存储用户的属性密钥并利用该密钥执行外包解密操作后得到部分解密密文，并将其部分解密密文返回给用户。

属性授权机构负责管理用户的属性、用户信任值的计算、属性密钥与全局密钥的生成及分发、系统公钥和主密钥等公共参数的生成。

数据所有者可以通过构造访问树定义自己的访问策略，并对明文数据进行加密后上传至云服务器存储。

用户发出资源访问请求时可以向代理服务器发送部分外包解密请求，只有用户所拥有的属性满足数据所有者定义的访问策略，通过对代理服务器返回的部分解密密文使用全局密钥进行解密得到明文。

方案算法实现

方案主要分为初始化阶段、加密阶段、信任计算、密钥生成阶段和解密阶段五个阶段。

1) 初始化阶段

$Setup(\lambda)\to(PK,MSK)$：初始化算法由 AA 执行，以安全参数 λ 作为输入，输出系统公钥 PK 和主密钥 MSK。

选取两个阶为 p 的乘法群，设 g 为 G_S 群的一个随机生成元，让 $\rho:G_S\times G_S\to G_T$ 表示双线性映射，并定义公开映射 $H:\{0,1\}\to G_T$。AA 随机选择 $\alpha,\beta\in Z_p$，生成系统公钥 PK 和主密钥 MSK：

$$PK=\left(G_S,g,h=g^{\beta},f=g^{1/\beta},e(g,g)^{\alpha}\right) \tag{1}$$

$$MSK=\left(\beta,g^{\alpha}\right) \tag{2}$$

2) 加密阶段

$Encrpt(PK,M,T)\to CT$：加密算法由 DO 执行，以系统公钥 PK、明文 M 和访问树 T 作为输入，输出密文 CT。

设 R 为访问树 T 的根节点，首先从根节点 R 开始，自上而下为访问树 T 的每个节点 x (包括叶子节点)选择一个多项式 q_x，针对多项式 q_x 定义阶数 d_x 和门限值 k_x 的关系为 $d_x=k_x-1$。随机选择根节点 R 的多项式 q_R 中的常数项 $s\in Z_p$，则有 $q_R(0)=s$。计算 $\tilde{C}=Me(g,g)^{\alpha s},C=h^s$。若 $x\neq R$，对于每个子节点，其常数项的值由函数 $q_x(0)=s=q_{parents(x)}(index(x))$ 生成，其他 d_x 个点同样是随机生成的。对于叶子节点，令 Y 为访问树 T 中的所有叶子节点集合，计算每个叶子节点 $y\in Y$ 对应的 $C_y=g^{q_y(0)},C_y'=H\left(att(y)^{q_y(0)}\right)$。计算密文：

$$CT=\left(T,\tilde{C}=Me(g,g)^{\alpha s},C=h^s,\forall y\in Y:C_y=g^{q_y(0)},C_y'=H\left(att(y)^{q_y(0)}\right)\right) \tag{3}$$

3) 信任计算

AA 根据用户所拥有的属性进行信任计算，用户信任值 T_t 可使用一个三元组 $T_t=[C,I,D]$ 来表示，且 $C,I,D\in[0,1]$。对每个 AA 所管理的属性进行分类，即分为正面属性集 P_A、负面属性集 N_A 和中性属性集 M_A 三类，通过用户所拥有的属性集 U_A 与每个 AA 所管理的各类属性的交集分别得到用户各类属性的数量。即正面属性数量 n'、负面属性数量 m'、中性属性数量 l'。具体计算如下：

$$n'=\left|U_A\cap P_A\right| \tag{4}$$

$$m'=\left|U_A\cap N_A\right| \tag{5}$$

$$l'=\left|U_A\cap M_A\right| \tag{6}$$

其中在三元组 $T_t=[C,I,D]$ 表达式中，C、I、D 分别表示可信、怀疑、不可信的信任程度。具体计算如下：

$$C=\frac{\sum_{i=1}^{n'}\omega_{p_{k,i}}+\frac{\sum_{x=1}^{l'}\omega_{m_{k,x}}}{2}}{\sum_{i=1}^{n'}\omega_{p_{k,i}}+\sum_{j=1}^{m'}\omega_{n_{k,j}}+\sum_{x=1}^{l'}\omega_{m_{k,x}}} \tag{7}$$

$$I=\frac{\sum_{j=1}^{m'}\omega_{n_{k,j}}+\frac{\sum_{x=1}^{l'}\omega_{m_{k,x}}}{2}}{\sum_{i=1}^{n'}\omega_{p_{k,i}}+\sum_{j=1}^{m'}\omega_{n_{k,j}}+\sum_{x=1}^{l'}\omega_{m_{k,x}}} \tag{8}$$

$$D=\frac{\sum_{x=1}^{l'}\omega_{m_{k,x}}}{\sum_{i=1}^{n'}\omega_{p_{k,i}}+\sum_{j=1}^{m'}\omega_{n_{k,j}}+\sum_{x=1}^{l'}\omega_{m_{k,x}}} \tag{9}$$

其中 $\omega_{p_{k,i}}$、$\omega_{n_{k,j}}$、$\omega_{m_{k,x}}$ 分别表示每个 AA 管理的每个正面属性、负面属性以及中性属性的初始权重，满足 $\omega_{p_{k,i}},\omega_{n_{k,j}},\omega_{m_{k,x}}\in[0,1]$。

由于信任具有时间相关性，不仅依赖于当前时刻计算的信任值，还要考虑之前某个特定时刻由 AA 计算的信任值。假设当前时刻的信任值为 $T_t=[C,I,D]_t$，t_x 时刻的信任值为 $T_{t_x}=[C,I,D]_{t_x}$，故用户信任值计算如下：

$$T_t=\varpi\times[C,I,D]_t+(1-\varpi)\times[C,I,D]_{t_x} \tag{10}$$

其中 ϖ 表示权重因子，且满足 $\varpi\in[0,1]$。

4) 密钥生成阶段

AA 根据步骤(3)中计算的信任值对用户进行身份认证，若信任值 $T_t=[C,I,D]$ 中的 C、I、D 满足 $C\geq\frac{1}{2}\&\,I<\frac{1}{2}\&\,D<\frac{1}{2}$，则为可信用户，执行以下算法。否则，不执行。

$KeyGen(U_{id},PK,S,\varphi)\rightarrow(UGSK,AA\_Key)$：密钥生成算法由 AA 执行，以全局身份标识 U_{id}、系统公钥 PK、用户属性集 S 和秘密值 φ 作为输入，输出用户的全局密钥 $UGSK$ 和属性密钥 AA_Key。

用户列表 UL 用来记录每一个可信用户，对于每一个需要注册的合法用户 U，AA 为其分配一个全局身份标识 U_{id}，其中 $U_{id}\in UL=\{1,2,\cdots,n\}$，对于每一个 U_{id}，随机选择 $\alpha,\beta,r,\gamma\in Z_p$，生成用户的全局密钥 $UGSK$ 以及秘密值 φ：

$$UGSK=\left(g^{(\alpha+r)/\beta},g^{\gamma}\right) \tag{11}$$

$$\varphi=g^{(r+\gamma)} \tag{12}$$

N 个属性授权机构 $AA_i(i\in N)$ 选择生成的秘密值 φ，并对所管理的属性 $j\in S$，随机选择一个数 $r_j\in Z_p$，生成用户的属性密钥 AA_Key：

$$AA\_key=\left(\forall i\in N,\{AA_i\_key\}\right)=\left(\forall j\in S:D_j=g^{(r+\gamma)}H(j)^{r_j},T_j=g^{r_j}\right) \tag{13}$$

并通过安全信道发送给 PS 存储，AA_Key 用于 PS 执行部分解密操作，以减少用户的存储开销与计算开销。而用户密钥 SK 由全局密钥 $UGSK$ 和属性密钥 AA_Key 组成。

$$\begin{aligned} SK &= \left(UGSK, AA\_key\right) \\ &= \left(\left(g^{(\alpha+r)/\beta}, g^{\gamma}\right), \left(\forall i \in N, \left\{AA_i\_key\right\}\right)\right) \\ &= \left(\left(g^{(\alpha+r)/\beta}, g^{\gamma}\right), \left(\forall j \in S : D_j = g^{(r+\gamma)} H\left(j\right)^{r_j}, T_j = g^{r_j}\right)\right) \end{aligned} \tag{14}$$

5) 解密阶段

① 外包解密

$Decrypt\_outsource\left(PK, AA\_Key, CT\right) \to PCT$ 或⊥：外包解密算法由 PS 执行，以系统公钥 PK、属性密钥 AA_Key、密文 CT 作为输入，输出部分解密密文 PCT。

为降低用户的开销，当 DU 想要访问数据时，向 PS 请求部分解密，PS 从 CS 中接收密文 CT 和用户列表 UL 对用户进行认证。若用户 $U_{id} \in UL$，PS 使用存储的属性密钥 AA_Key 执行外包解密计算，承担大量的解密运算，并将部分解密密文 PCT 发送给 DU。否则，返回⊥。具体步骤如下：

$DecryptNode\left(CT, AA\_Key, x\right)$ 为定义的递归函数运算，令 $i = attr\left(x\right)$，若节点 x 是访问树 T 的叶子节点且 $i \in S$，计算如下：

$$\begin{aligned} &DecryptNode\left(CT, AA\_Key, x\right) = \frac{e\left(D_i, C_x\right)}{e\left(T_i, C_i'\right)} \\ &= \frac{e\left(g^{(r+\gamma)} \times H\left(i\right)^{r_i}, h^{q_x(0)}\right)}{e\left(g^{r_i}, H\left(i\right)^{q_x(0)}\right)} = e\left(g, g\right)^{(r+\gamma)q_x(0)} \end{aligned} \tag{15}$$

如果 $i \notin S$，则有：$DecryptNode\left(CT, AA\_Key, x\right) = \perp$。当节点 x 是访问树 T 的非叶子节点，则对 x 的所有子节点 z 调用递归函数 $DecryptNode\left(CT, AA\_Key, x\right)$，其函数输出值集合为 F_z，令子节点 z 中的所有 k_x 大小的集合为 S_x，当且仅当 $F_z \neq \perp$ 时，计算如下：

$$\begin{aligned} F_x &= \prod_{z \in S_x} F_z^{\Delta_{i, S_x'}(0)} \\ &= \prod_{z \in S_x} \left(e\left(g, g\right)^{(r+\gamma)q_z(0)}\right)^{\Delta_{i, S_x'}(0)} \\ &= \prod_{z \in S_x} \left(e\left(g, g\right)^{(r+\gamma)q_{parent(z)}^{(index(z))}}\right)^{\Delta_{i, S_x'}(0)} \\ &= \prod_{z \in S_x} \left(e\left(g, g\right)^{(r+\gamma)q_x(i)}\right)^{\Delta_{i, S_x'}(0)} \\ &= e\left(g, g\right)^{(r+\gamma)q_x(0)} \end{aligned} \tag{16}$$

其中 $i = index\left(z\right), S_x' = \left\{index\left(z\right) : z \in S_x\right\}$。

若属性集合 S 满足访问树 T，PS 执行部分解密操作并将部分解密密文 PCT 发送给 DU。计算部分解密密文：

$$\begin{aligned} A = PCT &= DecryptNode\left(CT, AA\_Key, R\right) \\ &= e\left(g, g\right)^{(r+\gamma)q_R(0)} = e\left(g, g\right)^{(r+\gamma)s} \end{aligned} \tag{17}$$

② 用户解密

$Decrypt\_user\left(PK, PCT, UGSK\right) \to M$：用户解密算法由 DU 执行，以系统公钥 PK 和部分解密密文 PCT，全局密钥 $UGSK$ 为输入，输出明文数据 M。计算明文：

$$B = g^{(\alpha+r)/\beta} \times g^{\gamma} = g^{(\alpha+r)/\beta+\gamma} \tag{18}$$

$$\begin{aligned}\tilde{C}/\left(e(C,B)/A\right)&=\tilde{C}/\left(e\left(h^{s},g^{(\alpha+r)/\beta+\gamma}\right)\right)/e(g,g)^{(r+\gamma)s}\\&=Me(g,g)^{\alpha s}/\left(e\left(g^{\beta s},g^{(\alpha+r)/\beta+\gamma}\right)\right)/e(g,g)^{(r+\gamma)s}\\&=M\end{aligned}\tag{19}$$

5. 安全性分析

5.1. 安全性证明

假设：如果敌手 A 在任意多项式时间内对求解 DBDH 问题具有不可忽略的优势，则说明本方案是选择明文攻击安全的。

证明：假设存在一个敌手 A 能够以不可忽略的优势 ε 赢得游戏，构造一个模拟器 χ 利用敌手 A 的能力并以不可忽略的优势 $\frac{\varepsilon}{2}$ 解决 DBDH 假设。

1) 初始化：敌手 A 控制一组受攻击的属性授权机构 $\{AA_k\}\subset AA_n$，且 AA_n 中至少有两个 AA 不受控制，其余的 AA_n/AA_k 由模拟器 χ 控制。敌手 A 选取挑战的访问策略 T，其中一些属性由模拟器 χ 控制的 AA_n/AA_k 负责管理，而一些属性由不受控制的 AA 管理。模拟器 χ 设置 $a=\sum d_k$，$b=\frac{\sum v_k}{\sum c_k}$，$c=s_0$，其中随机选取 $d_1,d_2,\cdots,d_n$，$v_1,v_2,\cdots,v_n$，$s_0\in Z_p$。同时模拟器 χ 设置公共参数 $Y=e(A,B)=e(g,g)^{ab}$，并且将公共参数发送给敌手 A。

2) 查询阶段 1：敌手 A 想要查询尽可能多的属性密钥 AA_Key，而对应的属性集 $A_1,A_2,\cdots,A_q$ 被多个 AA_k 负责管理，其中没有一个满足访问策略 T。模拟器 χ 接收到密钥查询请求后，计算密钥组件来响应敌手 A 的请求。对于所有属性 $i\in AA_u$，模拟器 χ 随机选取 $r_i\in Z_p$，计算 $D_i=AA\times H\left(att(i)\right)^{r_i},T_i=g^{r_i}$。然后模拟器 χ 将生成的属性密钥发送给敌手 A。

3) 挑战：敌手 A 提交两条挑战消息 m_0 和 m_1 给模拟器 χ，模拟器 χ 随机抛出一枚硬币 γ，生成密文 $CT^{*}=\left(T_0,E_0=m_\gamma\times Z,\left\{C_i=g^{q_i(0)},C_i'=H\left(att(i)\right)^{q_i(0)}\right\}_{i\in AA^{T_0}}\right)$ 并将密文发送给敌手 A。

如果 $\mu=0$，$Z=e(g,g)^{abc}$，则有 $Z=e(g,g)^{abc}=\left(e(g,g)^{ab}\right)^{c}=Y^{S_0}$ 和 $D_i=g^{\sum d_k H(att(i))^{r_i}}$，因此 CT^{*} 是消息 m_γ 的有效密文，D_i 是密钥的有效组件。否则，如果 $\mu=1$，$Z=e(g,g)^{z}$，则 $E_0=m_\gamma\times e(g,g)^{z}$。由于 $z\in Z_p$ 是一个随机元素，所以从敌手 A 的角度来看，E_0 是 G_T 中的一个随机元素，因此 CT^{*} 是无效密文，不包含关于 m_γ 的信息。

4) 查询阶段 2：自适应地重复查询阶段 1 的操作。

5) 猜测：敌手 A 提交一个 γ 的猜测 γ'。如果 $\gamma=\gamma'$，模拟器 χ 输出 $\mu=0$，表明是一个有效的 DBDH 元组 $e\left(g,a,B,C,Y^{S_0}\right)$。如果 $\gamma\neq\gamma'$，模拟器 χ 输出 $\mu=1$，表明得到一个随机元组 (g,a,B,C,Z)。当 $\mu=0$，敌手 A 得到一个有效密文 m_γ。根据定义，敌手 A 在这种情况下成功的概率为：$P_r\left[\gamma=\gamma'\mid\mu=0\right]=\frac{1}{2}+\varepsilon$。当 $\mu=1$ 时，敌手 A 不知道 γ 的任何消息，则成功的概率为：$P_r\left[\gamma\neq\gamma'\mid\mu=1\right]=P_r\left[\gamma=\gamma'\mid\mu=1\right]=\frac{1}{2}$。因此，模拟器 χ 在与挑战者之间的游戏中成功的概率为：

$$P_r\left[\mu=\mu'\mid\mu=0\right]\times P_r\left[\mu=0\right]+P_r\left[\mu=\mu'\mid\mu=1\right]\times P_r\left[\mu=1\right]-\frac{1}{2}=\frac{1}{2}\left(\frac{1}{2}+\varepsilon\right)+\frac{1}{2}\cdot\frac{1}{2}-\frac{1}{2}=\frac{\varepsilon}{2}$$

由上述证明可知，在 DBDH 假设中，如果敌手 A 的优势在多项式时间内为不可忽略的 ε，那么模拟器 χ 的优势 $\frac{\varepsilon}{2}$ 也是不可忽略的。由于本方案在任意多项式时间内可以解决 DBDH 问题，因此本方案对于选择明文攻击是安全的。

5.2. 抵抗合谋攻击

在本方案中，用户密钥 SK 由属性密钥 AA_Key 和全局密钥 $UGSK$ 组成。AA 将生成的全局密钥 $UGSK=\left(g^{(\alpha+r)/\beta},g^{\gamma}\right)$ 通过安全信道发送给用户，当非法用户获取该密钥时，并不能直接对密文进行解密操作，而全局密钥 $UGSK$ 通过随机选择 α，β，r，γ 来防止用户之间的合谋攻击，非法用户不能通过合谋得到解密密钥。代理服务器利用属性密钥 AA_Key 执行部分解密工作，若代理服务器试图通过解密操作获取明文，则需要获得存储在用户端的全局密钥 $UGSK$。AA_Key 通过随机选择 r_j，γ 来防止 AA 之间的合谋攻击，同时秘密值 $\varphi=g^{(r+\gamma)}$ 被用来防止用户与 AA 之间的合谋攻击。因此，本方案能有效地抵抗非法用户、代理服务器以及属性授权机构之间的合谋攻击。

6. 性能分析

6.1. 理论分析

为了说明本方案的特点，将所提出的方案与现有方案进行性能对比分析，包括综合性能分析和计算开销对比分析。具体从多授权中心、外包解密、抗合谋攻击、信任机制、加密阶段与解密阶段的开销等几个方面进行分析。如表 1、表 2 所示。

相关符号表示如下：C_m 表示在群 G_S 下的一次乘法操作，C_e 表示在群 G_S 下的指数运算，C_p 表示在群 G_T 下的双线性配对运算，l 表示访问策略中属性的个数，m 表示用户满足解密需求的属性个数。

从表 1 可以得知，与其它方案相比，由于支持多授权中心、抗合谋攻击和信任机制，并通过外包解密减少用户的开销，本方案更适用于云存储环境下实现安全高效和细粒度的访问控制。

Table 1. Comprehensive comparative analysis of each scheme
表 1. 各方案综合比较分析

方案	多授权中心	外包解密	抗合谋攻击	信任机制
文献[13]	×	√	√	×
文献[17]	√	×	√	×
文献[18]	√	√	√	×
文献[20]	√	×	√	×
本方案	√	√	√	√

Table 2. The comparison of computation cost
表 2. 计算开销对比

方案	加密阶段	解密阶段
文献[13]	$C_p+(l+1)C_e+3C_m$	$3C_p+3C_m$
文献[20]	$lC_p+(3l+3)C_e+(l+2)C_m$	$(2m+2)C_p+mC_e+(2m+1)C_m$
本方案	$C_p+(2l+1)C_e+C_m$	$3C_p+3C_m$

为了分析本方案的计算开销，将所提出的方案与现有方案进行对比，如表 2 所示。算法执行时间主要花费在指数运算、点乘运算以及双线性配对操作上面，其中双线性配对运算代价最高。而系统整体运算过程中的效率主要由数据拥有者加密和用户解密过程中的运算量决定。

由上述表格表明，本方案与文献[13]中用户在解密阶段中的运算量为 $3C_p+3C_m$，但本方案使用多授权中心提高密钥分发的效率和抗合谋攻击的能力。文献[20]中加密阶段与解密阶段的运算量分别为 $lC_p+(3l+3)C_e+(l+2)C_m$ 和 $(2m+2)C_p+mC_e+(2m+1)C_m$。而本方案在加密和解密过程中的运算量要比文献[20]小，效率更高。

6.2. 实验分析

为了进一步对本方案的效率进行评估，本方案的实验环境为 Inter(R) Core(TM) i5-4210U CPU @1.70GHz，主频 2.4 GHz，内存 8.00 GB，基于 Windows 7 操作系统安装的 IntelliJ IDEA 环境下进行仿真实验。实验结果如图 2 和图 3 所示，分别从加密时间和解密时间随属性个数的增加而变化的情况，将本方案与文献[20]进行对比分析。

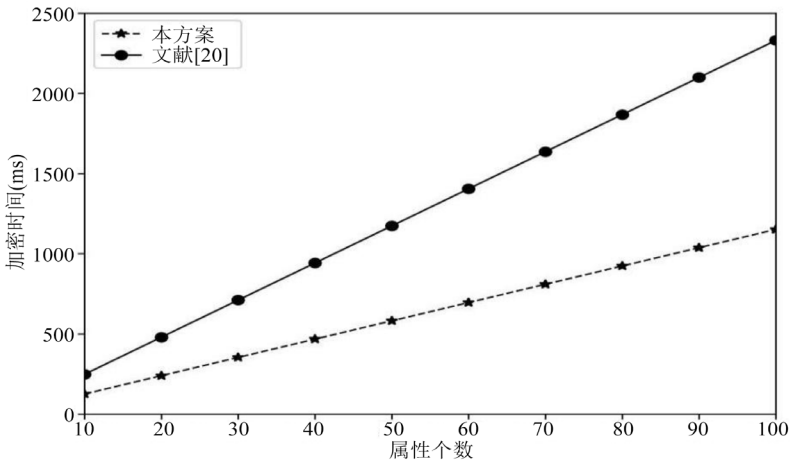

Figure 2. The Comparison of encryption time cost of different number of attributes
图 2. 不同属性个数的加密时间开销比较

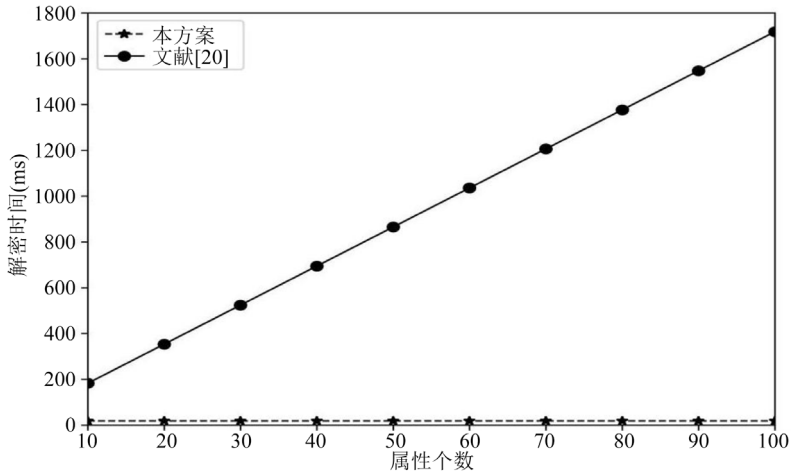

Figure 3. The Comparison of decryption time cost of different number of attributes
图 3. 不同属性个数的解密时间开销比较

7. 结论

本文提出一种云存储中基于多授权中心的 CP-ABE 访问控制方案。该方案中的密钥由属性密钥和全局密钥组成，增强用户密钥的安全性，有利于抵抗合谋攻击。同时结合信任机制，多授权中心根据用户的信任值进行密钥的生成与分发，使用代理服务器执行部分解密操作并存储用户的属性密钥，减少用户的计算开销和存储开销。通过安全性和性能分析证明了方案的安全性和高效性，能够在云计算环境下实现安全、高效和细粒度的访问控制。

基金项目

本文得到广东省重点领域研发计划项目(2019B010139002)，广州市科技计划项目(201902020006、201902020007、201902010034)的资助。

参考文献

[1] Mell, P. (2010) The NIST Definition of Cloud Computing. *Communications of the ACM*, **53**, 50. https://doi.org/10.6028/NIST.SP.800-145

[2] Wang, S., Zhou, J., Liu, J.K., *et al*. (2016) An Efficient File Hierarchy Attribute-Based Encryption Scheme in Cloud Computing. *IEEE Transaction on Information Forensics and Security*, **11**, 1265-1277. https://doi.org/10.1109/TIFS.2016.2523941

[3] Sukhodolskiy, I.A. and Zapechnikov, S.V. (2017) An Access Control Model for Cloud Storage Using Attribute-Based Encryption. *IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering*, St. Petersburg, 578-581. https://doi.org/10.1109/EIConRus.2017.7910620

[4] 洪澄, 张敏, 冯登国. 面向云存储的高效动态密文访问控制方法[J]. 通信学报, 2011, 32(7): 125-132.

[5] Sahai, A. and Waters, B. (2005) Fuzzy Identity-Based Encryption. In: *International Conference on Theory and Application of Cryptographic Techniques*, Springer Verlag, Berlin, 457-473. https://doi.org/10.1007/11426639_27

[6] Goyal, V., Pandey, O., *et al*. (2006) Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data. *Proceedings of the* 13*th ACM Conference on Computer and Communications Security*, Alexandria, VA, October 30-November 3 2006, 89-98. https://doi.org/10.1145/1180405.1180418

[7] Bethencourt, J., Sahai, A. and Waters, B. (2007) Ciphertext-Policy Attribute-Based Encryption. *IEEE Symposium on Security and Privacy*, Berkeley, CA, 20-23 May 2007, 321-334. https://doi.org/10.1109/SP.2007.11

[8] 房梁, 殷丽华, 郭云川, 方滨兴. 基于属性的访问控制关键技术研究综述[J]. 计算机学报, 2017, 40(7): 1680-1698.

[9] 何颖, 徐军, 侯雅婷. 云计算中的信任机制研究[J]. 计算机技术与发展, 2017, 27(10): 101-105.

[10] Riad, K. (2016) Multi-Authority Trust Access Control for Cloud Storage. 4*th International Conference on Cloud Computing and Intelligence Systems*, Beijing, 429-433. https://doi.org/10.1109/CCIS.2016.7790297

[11] Li, J., Huang, X., Li, J., *et al*. (2014) Securely Outsourcing Attribute-Based Encryption with Check Ability. *IEEE Transactions on Parallel and Distributed Systems*, **25**, 2201-2210. https://doi.org/10.1109/TPDS.2013.271

[12] Shao, J., Zhu, Y. and Ji, Q. (2017) Efficient Decentralized Attribute Based Encryption with Outsourced Computation for Mobile Cloud Computing. *IEEE International Symposium on Parallel and Distributed Processing with Applications and* 2017 *IEEE International Conference on Ubiquitous Computing and Communications* (*ISPA/IUCC*), Guangzhou, 417-422. https://doi.org/10.1109/ISPA/IUCC.2017.00067

[13] Zhang, P., Chen, Z., Liu, J.K., Liang, K. and Liu, H. (2016) An Efficient Access Control Scheme with Outsourcing Capability and Attribute Update for Fog Computing. *Future Generation Computing Systems*, **12**, 753-762. https://doi.org/10.1016/j.future.2016.12.015

[14] Michalas, A. and Weingarten, N. (2017) HealthShare: Using Attribute-Based Encryption for Secure Data Sharing between Multiple Clouds. *IEEE* 30*th International Symposium on Computer-Based Medical Systems*, Thessaloniki, 22-24 Jun 2017, 811-815. https://doi.org/10.1109/CBMS.2017.30

[15] Chase, M. and Chow, S.S.M. (2009) Improving Privacy and Security in Multi-Authority Attribute-Based Encryption. *Proceedings of the* 16*th ACM Conference on Computer Communications Security*, Chicago, IL, 9-13 November 2009, 121-130. https://doi.org/10.1145/1653662.1653678

[16] 雷丽楠, 李勇. 基于密文策略属性基加密的多授权中心访问控制方案[J]. 计算机应用研究, 2018, 35(1): 248-252+276.

[17] 谭跃生, 章世杨, 王静宇. 基于多授权中心的 CP-ABE 属性撤销方案[J]. 计算机工程与应用, 2019, 55(13): 78-84.

[18] Sandor, V.K.A., *et al*. (2019) Efficient Decentralized Multi-Authority Attribute Based Encryption for Mobile Cloud Data Storage. *Journal of Network and Computer Applications*, **129**, 25-36. https://doi.org/10.1016/j.jnca.2019.01.003

[19] Wei, J., Liu, W. and Hu, X. (2018) Secure and Efficient Attribute-Based Access Control for Multiauthority Cloud Storage. *IEEE Systems Journal*, **12**, 1731-1742. https://doi.org/10.1109/JSYST.2016.2633559

[20] Vaanchig, N., Xiong, H., Chen, W. and Qin, Z. (2018) Achieving Collaborative Cloud Data Storage by Key Escrow Free Multi-Authority CP-ABE Scheme with Dual Revocation. *International Journal of Network Security*, **20**, 95-109.

[21] Boneh, D. and Franklin, M.K. (2001) Identity-Based Encryption from the Weil Pairing. *Siam Journal on Computing*, **32**, 213-229. https://doi.org/10.1007/3-540-44647-8_13

[22] Josh, B. and Jerry, L. (1990) Generalized Secret Sharing and Monotone Functions. In: *Advances in Cryptology CRYPTO*99, Spnnger-Verlag, Berlin Heidelberg, 27-35. https://doi.org/10.1007/0-387-34799-2_3

[23] Blakley, G.R. (1979) Safeguarding Cryptographic Keys. *National Computer Conference*, New York, 4-7 June 1979, 313-317. https://doi.org/10.1109/MARK.1979.8817296

[24] Yacobi, Y.A. (2002) Note on the Bilinear Diffie-Hellman Assumption. Iacr Cryptology Eprint Archive, 45-57.