云中可动态更新的属性基代理重加密方案

杨 耿[1,2], 郭 瑞[1,2], 庄朝源[1,2], 王旭涛[1,2]

[1]西安邮电大学网络空间安全学院 西安 中国 710121
[2]西安邮电大学无线网络安全技术国家工程实验室 西安 中国 710121

摘要 代理重加密是在保证重加密授权者私钥安全的前提下进行密文转换的操作, 实现了云中数据的动态共享。而在基于属性的代理重加密方案中, 其代理方可以在不泄露明文数据的前提下, 将访问策略下的密文经过重加密转换为不同的访问策略下的密文, 完成密态数据的安全外包计算。现有的属性代理重加密方案只是实现了密文策略的更新变换, 存在着实用性低, 计算量大等缺点。为了满足用户权限的动态更新, 以及传统属性加密体制中用户离线后不能向他人提供解密能力的问题, 本文提出了一种云中可动态更新的属性基代理重加密方案。通过在系统公开参数中加入用户集合信息并利用属性撤销技术, 分别实现了用户集合与属性集合的动态更新, 以保证用户权限的动态更新, 并且该方案满足单向性、非交互性、非传递性、非转移性和可验证性等特点。此外, 利用离线加密技术将加密操作分成两步实现, 大量的辅助计算在离线阶段进行, 降低了用户客户端在线加密的计算开销。同时, 受理者可以对代理重加密密文进行验证操作, 避免数据遭受第三方破坏。安全性方面, 在标准模型和判定性 q 阶双线性 Diffie-Hellman 假设下, 证明了本方案具有选择明文攻击下的密文不可区分性且可抵抗同谋攻击。最后, 通过效率分析发现, 本方案的在线加密阶段计算量较小且用户的密钥和密文存储开销低, 具有良好的实用性。

关键词 代理重加密; 属性加密; 访问策略; 动态更新; 离线加密
中图法分类号 TP309 **DOI 号** 10.19363/J.cnki.cn10-1380/tn.2022.05.04

Dynamically Updatable Attribute Based Proxy Re-encryption Scheme in Cloud

YANG Geng[1,2], GUO Rui[1,2], ZHUANG Chaoyuan[1,2], WANG Xutao[1,2]

[1] School of Cyberspace Security, Xi'an University of Posts and Telecommunications, Xi'an 710121, China;
[2] National Engineering Laboratory for Wireless Security, Xi'an University of Posts and Telecommunications, Xi'an 710121, China;

Abstract Proxy re-encryption is the operation of ciphertext conversion under the premise of ensuring the security of the re-encryption authorizer's private key, which realizes the dynamic sharing of data in cloud. In the attribute-based proxy re-encryption scheme, the agent can re-encrypt the ciphertext in the access policy into the ciphertext in different access policies without revealing the plaintext data to complete the encrypted data security outsourcing calculations. The existing attribute proxy re-encryption scheme only realizes the update and transformation of the ciphertext policy. That has the shortcomings of low practicability and large amount of calculation. In order to meet the dynamically updatable of user permissions and the problem that users cannot provide decryption capabilities to others after offline in traditional attribute encryption systems, this paper proposes a dynamically updateable attribute-based proxy re-encryption scheme in cloud. By adding user set information into system public parameters and using attribute revocation technology, the dynamic update of user set and attribute set is implemented to ensure the dynamic update of user permissions. The scheme meets the characteristics of one-way, non-interaction, non-transitivity, non-transferability and verifiability. In addition, the offline encryption technology is used to divide the encryption operation into two steps, and a large number of auxiliary calculations are performed in the offline phase, which reduces the computational overhead of online encryption at the user's client. At the same time, acceptor can verify the proxy re-encrypted ciphertext to prevent data from being damaged by a third party. In terms of security, under the standard model and the q-parallel bilinear Diffie-Hellman assumption, it is proved that this scheme has the indistinguishability of ciphertext under the selected plaintext attack and can resist the collusion attack. Finally, through the efficiency analysis, it is found that the online encryption stage of this scheme has a small amount of calculation and the user's key and ciphertext storage overhead is low, which has better practicability.

Key words proxy re-encryption; attribute encryption; access policy; dynamically updatable; offline encryption

---

通讯作者: 杨耿, 硕士, Email:yanggeng1996xupt@163.com。

国家自然科学基金(No. 62072369, No. 62072371, No. 61802303, No. 61772418), 陕西省重点研发计划(No.2021ZDLGY06-02, No. 2020ZDLGY08-04, No. 2019KW-053), 陕西省创新能力支持计划(No.2020KJXX-052, No. 2017KJXX-47), 陕西省自然科学基金(No.2019JQ-866, No. 2018JZ6001), 陕西省教育厅科研项目(No.19JK0803)
收稿日期: 2021-04-26; 修改日期: 2021-09-01; 定稿日期: 2022-03-15

1 引言

在计算机网络迅速发展的时代下, 信息技术对各行各业产生了深远影响。云计算作为互联网时代重要的一项技术, 已被人们广泛应用。其中, 云存储服务已经成为用户将数据外包的关键方式, 而云数据共享技术[1]是信息交互的重要方法, 极大方便了用户将数据进行授权共享。近年来, 数据安全问题频发, 用户隐私信息遭受频繁泄露, 云中用户隐私安全问题已经成为社会和大众关注的热点问题。因此, 当云服务器上存储敏感数据时, 如何保证托管数据的机密性, 是数据拥有者面临的主要挑战之一。

在云环境不可靠的背景下, 加密技术的发展是保证数据机密性的重要途径。用户先对数据进行加密操作, 再将加密后的结果上传至云服务器上, 当有用户需要时, 再进行下载并解密从而获取原始明文数据。针对于各种安全需求, 不同的密码算法和协议应运而生, 属性基加密[2]不仅可以实现“一对多”的加密数据共享, 而且拥有灵活的访问控制策略, 对用户可以达到细粒度的访问控制效果。但是, 在许多应用场景下, 云服务器中的大量密文都需要进行转换, 进而才能为其他用户获取解密。

代理重加密(Proxy Re-encryption, PRE[3])技术的出现既能保证用户数据安全性又能达到数据的灵活访问与共享[4]。在保证授权者私钥安全的前提下, 代理方对密文进行重加密操作, 实现了密文的安全转换。同时, 使用 PRE 技术能够大大减少云服务提供商的计算开销, 只需代理方直接对密文进行一些计算就可以完成转换。由于属性加密和代理重加密的种种优势和特性, 基于属性的代理重加密(Attribute-based Proxy Re-encryption, ABPRE[5])也相应地被提出, 代理方可以对密文进行重加密, 达到访问策略的更换, 在实际生活中, 由于用户属性集合具有动态变化的性质, 当属性发生撤销或者属性更新时, 用户的权限也跟着动态变化, 所以对用户属性私钥的更新和属性集合的撤销以及相关密文的转换是至关重要的。因此, 动态更新的属性代理重加密方案[6]的研究具有重要的实际意义。

1.1 相关工作

基于用户角色的访问控制策略[7-9]在云环境中的使用是多数的, 但是, 如果要对用户的权限进行细粒度划分, 就要为用户定义大量的角色, 这样就导致角色分配和管理困难的问题。模糊身份基加密方案是 Sahai 和 Waters[10]在 2005 年提出的, 他们将用户的属性集合来表示身份信息, 加密者可以使用属性集合来对数据进行加密, 当数据访问者属性私钥中的属性集合满足密文中的嵌套的属性集合时, 才可解密来恢复明文数据。在 2006 年, Goyal[11]提出了第一个完全的基于属性的加密方案, 分为密钥策略的属性基加密(Key Policy Attribute-based Encryption, KP-ABE[12]) 以及密文策略的属性基加密(Ciphertext Policy Attribute-based Encryption, CP-ABE[13]), 并对模糊身份基加密方案进行粒度细分, 扩充成为基于属性的加密, 而且给出了形式化的定义。在 2007 年, Ostrovsky 等人[14]在将单调访问结构中加入非逻辑改为非单调的形式, 使私钥可以表示任意属性上的访问公式。在以往的属性加密中, 由于密文长度与属性策略有关系, 这使得在加密过程中时间开销大且系统负担重。为提高加密效率, 马等人[15]在 2014 年提出了基于属性的在线/离线加密体制, 离线阶段在不知明文和策略的前提下, 首先对数据进行预处理, 生成一批临时密文, 当在线阶段获取到明文消息和属性策略后只需做简单的计算就可以生成完整密文。Su 等人[16]提出了在线/离线加密机制和关键字搜索结合的方案, 该方案能抵御关键字猜测攻击。

Liang 等人[17]于 2009 年在基于属性加密方案的基础上加入了代理重加密技术, 在访问控制的环境下, 允许用户将数据的部分解密能力授权给具有某些属性的其他人, 在系统中加入一个代理方, 他能够将原始访问结构下的密文转换为另一个不同访问结构下的密文数据, 实现“一对多”的密文转换。Luo 等人[18]提出了一个支持负值属性和通配符与门结构的 ABPRE 方案, Li 等人[19]于 2013 年提出了基于 LSSS 访问结构的 ABPRE 方案, 支持任意单调的访问结构, 之后 Sehehri 等人[20]提出了能将用户属性集合与访问策略均用向量表示的 ABPRE 方案, 且具有单向性、非交互性和重复性, 但用户私钥和数据密文占用存储资源过大。2019 年, Feng 等[21]人提出了一种支持多种特性的 ABPRE 方案, 该方案将大量解密工作外包给服务器, 大大减轻了用户计算负担。

为了实现用户权限的动态更新, 在现有可撤销的属性加密方案中, 按照撤销执行者不同分为直接撤销、间接撤销和混合撤销, 按照撤销执行精度不同可分为系统属性撤销、用户属性撤销和用户身份撤销。Cui 等人[22]提出了用户间接撤销的 CP-ABE 方案, 未撤销的用户可以通过不受信任的服务器来完成密

文的转换。Qin 等人[23]提出了恒定属性密钥长度的CP-ABE 方案, 该方案解决了解密密钥暴露的问题。为了实现直接撤销, Attrapadung 和 Imai[24]利用广播加密技术实现了属性直接撤销, 不用对用户的密钥进行定期更新。Zhang 等人[25]设计了支持直接撤销和用户撤销的 CP-ABE 方案, 在方案加入辅助函数用来更新密文, 且该方案中的密文长度小且定长。宋等人[26]提出的属性撤销的无密钥托管的加密方案在支持解密外包的同时达到用户属性撤销。Chen 和 Wang[27]提出了具有双重撤销的 CP-ABE 方案, 并利用柯西矩阵满足 LSSS 的访问矩阵进行了优化, 实现了用户级撤销方案。

Huang[28]提出了基于属性的边缘计算加密, 该方案支持用户的权限动态变化, 适用于跨域应用中的细粒度访问控制, 但该方案用户端的解密计算开销较大。Wang[29]提出了基于代理重加密的 CP-ABE 访问控制方案, 利用云服务器对数据进行重加密操作, 在保证数据细粒度访问控制的同时还支持用户属性的直接撤销, Xu 等人[30]将时间戳引入到重加密中, 即使数据拥有者离线时, 也能达到细粒度访问。Shao 等人[31]在线上/线下加密基础上并利用转换密钥技术将大量解密计算外包给代理服务器, 用户端的开销成本大大降低, 在 Guo 等人[32]提出的方案中, 将线上线下加密和密文部分外包解密结合用于属性加密中, 既提高了算法加密的效率, 在解密时又降低了用户的时间开销, 并且利用变色龙哈希函数来实现用户的属性撤销。

1.2 本文贡献

为了在云环境下实现数据的安全共享和系统用户的权限更新, 本文提出的云中可动态更新的属性基代理重加密方案主要贡献如下:

(1) 利用在线/离线加密技术, 将属性加密算法分为两步, 在线阶段只需要通过离线阶段生成的临时密文和系统公钥来做少量加法和乘法就可生成最终密文。此外, 本方案可以让用户通过执行重加密验证算法来验证重加密密文的正确性和完整性, 以防代理方对数据造成破坏或更改。

(2) 本方案在属性代理重加密的基础上, 加入动态成员管理算法, 用户可以自由的加入和退出系统, 并且在系统中用户的属性集合可以实时变化, 对应的用户权限也随之更新。

(3) 本方案具有非转移性, 因此, 可以抵御代理者和受理者的同谋攻击, 在基于判定性并行双线性 Diffie-Hellman 假设证明了本方案是选择明文攻击下的密文不可区分性。

2 预备知识

2.1 双线性映射

假设 G, G_T 为乘法循环群, 其阶都是为大素数 p, g 为 G 的生成元, $\hat{e}$ 为双线性映射 $\hat{e}: G\times G\to G_T$, 且满足以下性质:

(1) 双线性: 对于任意的 $g_1, g_2\in G$ 以及 $a,b\in Z_p^*$, 都有 $\hat{e}({g_1}^a, {g_2}^b)=\hat{e}(g_1,g_2)^{ab}$ 成立。

(2) 非退化性: 存在 $g_0\in G$ 满足 $\hat{e}(g_0,g_0)\neq 1$。

(3) 可计算性: 存在有效的算法, 对于任意的 $g_1,g_2\in G$, 均可计算 $\hat{e}(g_1,g_2)$。

2.2 访问结构

设 $\psi=\{\psi_1,\psi_2,\cdots,\psi_n\}$ 是由 n 个属性组成的集合, 访问结构是 ψ 的某些非空子集构成的集族 Λ, 其中 2^{ψ} 表示 ψ 的所有子集构成的集合。则 $\zeta\subseteq 2^{\psi}\setminus\{\varnothing\}$, 对于任意集合 B、C 都有: 若 $B\in\Lambda$ 且 $B\subseteq C$, 则 $C\in\Lambda$, 访问结构 Λ 就是单调的, 包含在 Λ 中的集合为授权集合, 不包含在 Λ 中的集合为非授权集合。

2.3 线性秘密共享方案

假设在属性集合 $\psi=\{\psi_1,\psi_2,\cdots,\psi_n\}$ 上的秘密共享方案 $\prod$ 是线性的, 满足以下两点:

(1) 所有属性的秘密分享值组成 Z_p^* 上的一个向量。

(2) 秘密共享方案 $\prod$ 存在一个 $l\times n$ 的共享矩阵 M, 令 M 的每一行 $\vec{M}_i$ 映射到为属性 $\rho(i)$ 的属性集合中, $i=1,2,\cdots,l$。随机选取 $s,v_1,v_2,\cdots,v_q\in Z_p^*$, 构成一个列向量 $\vec{v}=(s,v_1,v_2,\cdots,v_q)^{\mathrm{T}}$, 其中 s 为需要共享的秘密值, $\lambda_i=\vec{M}_i\vec{v}$ 为分配给属性 $\rho(i)$ 的秘密份额, 为秘密 s 的 l 个共享子秘密。

线性重构性: 对于所有的授权集 $S\in\Lambda$, 定义 $I\subset\{1,2,\cdots,l\}$ 且 $I=\{i:\rho(i)\in S\}$, 存在常数集 $\{\omega_i\in Z_p^*\}_{i\in I}$ 使得 $\sum_{i\in I}\omega_i\vec{M}_i=(1,0,0,\cdots,0)$, 这些常数可以在于共享生成矩阵 $\boldsymbol{M}$ 的大小相关的多项式时间内计算出来, 对于任何的非授权集合, 这些常数是不存在。

2.4 判定性 q 阶双线性 Diffie-Hellman 假设

假设两个阶为大素数 p 的乘法循环群 G_1, G_2, 其中 g 是 G_2 的生成元。随机选择 $a,s,b_1,b_2,\cdots,b_q\in Z_p^*$, 公开以下参数:

$$\vec{y}=\{g,g^{s},g^{a},\cdots,g^{a^{q}},g^{a^{q+2}},\cdots,g^{a^{2q}},$$
$$\forall 1\leqslant j\leqslant q\colon g^{sb_j},g^{a/b_j},\cdots,g^{a^{q}/b_j},g^{a^{q+2}/b_j},\cdots,g^{a^{2q}/b_j}$$
$$\forall 1\leqslant j,k\leqslant q,k\neq j\colon g^{asb_k/b_j},g^{a^{2}sb_k/b_j},\cdots,g^{a^{q}sb_k/b_j}\}$$

判定性 q-parallel BDHE 假设是指对于敌手而言，区分 $\hat{e}(g,g)^{a^{q+1}s}\in G_2$ 和群 G_2 中的随机元素 R 是困难的。

3 算法设计

3.1 系统框架

如图 1 所示，本方案系统中总共包含 6 个实体部分，分别为：密钥生成中心(Key Generation Center, KGC)、云服务器(Cloud Server, CS)、半可信代理方(Semi Trust Agent, STA)、数据拥有者(Data Owner, DO)、重加密授权者(Re-encryption Authorizer, RA)和数据用户(Data User, DU)。

(1) 密钥生成中心：KGC 是系统中的可信机构，负责进行系统的初始化并生成公共参数和主密钥，为注册用户生成公私钥对。

(2) 数据拥有者：DO 在系统中向其他用户进行数据共享的实体。DO 将要共享的数据进行密文策略属性基加密，然后将密文 CT 上传至云服务器 CS。

(3) 云服务器: CS 是负责外包存储 DO 上传的原始密文数据和 STA 上传的重加密密文数据。

(4) 重加密授权者: RA 是系统中的一组实体，且属性集合能够满足原始密文策略。RA 输入新的访问策略来生成重加密密钥 RK，并发送至 STA。

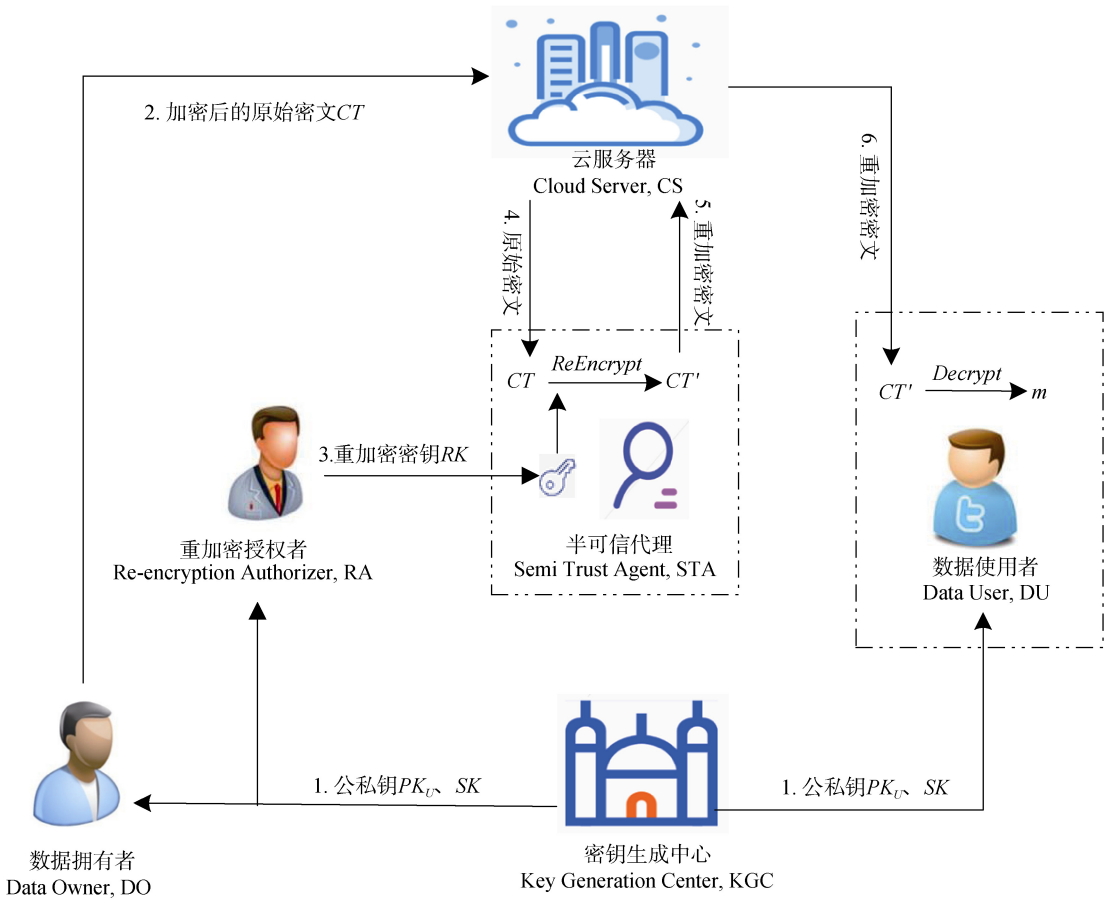

图 1 数据共享模型

Figure 1 Data sharing model

(5)半可信代理方：STA 是系统中半可信第三方，负责对原始密文进行重加密操作，并将重加密密文发送给 CS。

(6)数据使用者：DU 为访问重加密密文 CT' 的实体用户，先对重加密密文进行验证，若通过，则利用自己的公私钥进行重加密密文的解密操作，进而获取明文数据。

3.2 属性撤销框架

如图 2 所示，当系统中有用户发生属性撤销时，KGC 会更新系统公共参数和与该属性相关的其他用户的属性私钥 SK，发送给对应用户，并生成转换密钥 UK 发送至 CS，CS 收到后 UK 后，更新与被撤销

属性相关联的密文组件。

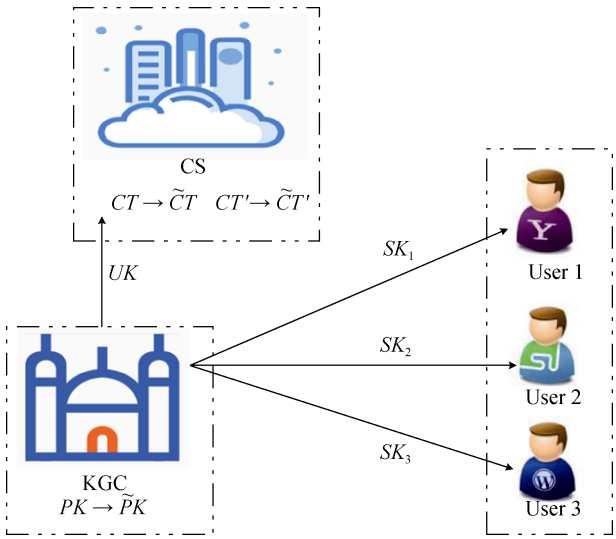

图 2 属性撤销模型

Figure 2 Attribute revocation model

3.3 符号说明

本文用到的符号定义在表 1 中。

表 1 系统符号

Table 1 System symbol

符号	含义
k	系统安全参数
U	系统属性空间
Ω	系统用户集合
S	用户属性集合
W	访问结构
W'	不同于 W 的访问结构
M	策略矩阵
M'	不同于 M 的策略矩阵

3.4 算法的形式化定义

定义 3.1 本文提出的云中可动态更新的属性基代理重加密算法分为初始化算法(*Setup*)、密钥生成算法(*KeyGen*)、加密算法(*Encrypt*)、重加密密钥生成算法(*RekeyGen*)、重加密算法(*ReEncrypt*)、重加密验证算法(*ReEncryptVerify*)、解密算法(*Decrypt*)和动态成员管理算法(*DMML*), 其中加密算法分为离线加密算法($Encrypt_{off}$)和在线加密算法($Encrypt_{on}$)。

(1) 初始化算法 $Setup(1^k) \to (PK, MK)$: 该算法由 KGC 执行, 输入系统安全参数 1^k, 输出系统的公共参数 PK 和系统主密钥 MK, PK 公开, MK 则由 KGC 自行保留。

(2) 密钥生成算法 $KeyGen(PK, MK, S) \to (PK_U, SK)$: 该算法由 KGC 执行, 输入系统公共参数 PK、系统主密钥 MK 和用户属性集合 $S \subseteq U$, 输出用户的公钥 PK_U 和与属性 S 相关的私钥 SK。

(3)离线加密算法 $Encrypt_{off}(PK) \to (IC)$: 该算法由 DO 客户端执行, 输入系统的公共参数 PK, 输出临时密文 IC。

(4) 在线加密算法 $Encrypt_{on}(m, W, IC) \to (CT)$: 该算法由 DO 客户端执行, 输入待加密数据 m、线性整数秘密共享 LSSS 访问结构 W, 临时密文 IC, 输出原始密文 CT。

(5)重加密密钥生成算法 $RekeyGen(PK, PK_U, SK, W') \to (RK)$: 该算法 RA 执行, 输入系统公共参数 PK、用户公私钥 PK_U、SK 和新的线性整数秘密共享 LSSS 访问结构 W', 输出重加密密钥 RK。

(6) 重加密算法 $ReEncrypt(PK, RK, CT) \to (CT' 或 \perp)$: 该算法由 STA 执行, 输入系统公共参数 PK、密文 CT 和重加密密钥 RK, 首先检查 RK 是否合法, 若合法, 输出与访问结构 W' 相关的重加密密文 CT', 否则当密文被设置为不能重加密或者 $S \models (M, \rho)$ 时, 输出 $\perp$, 终止操作。

(7) 重加密验证算法 $ReEncryptVerify(C_q, C', C_0, C^*) \to (true 或 \perp)$: 该算法由解密者 B 执行, 输入 STA 代重加密计算结果 C_q 和 CT 密文组件 C, C_0, 以及 C^*。若验证通过, 输出 $true$, 否则直接输出 $\perp$。

(8) 解密算法 $Decrypt(PK, PK_{UB}, SK_B, S_B, CT') \to (m 或 \perp)$: 该算法由解密者 B 执行, 输入系统公共参数 PK、用户公私钥 PK_{UB}、SK_B、以及用户属性集合和重加密密文 CT'。输入完成后, 首先检查解密者 DU_B 的属性集合 S_B 是否满足密文 CT' 中的访问结构 W', 若满足, 则用公私钥进行解密操作得到明文消息 m, 否则, 输出 $\perp$, 终止操作。

(9) 动态成员管理算法 *DMML*: 该算法由 KGC 和 CS 执行, 可以动态的管理用户, 当用户登记注册或者退出离开时, 更新系统公共参数 PK, 当用户发生属性撤销时, KGC 更新公共参数 PK 和相关用户的属性私钥 SK, 并且生成密文转换密钥 UK, CS 使用转换密钥对相关密文进行更新。

3.5 安全模型

接下来, 定义 CP-ABPRE 方案的安全模型。

定义 3.2 假设没有一个概率多项式时间(Probabilistic Polynomial-Time, PPT)的敌手 $\mathcal{A}$ 能够以一个不可忽略的优势赢得下面这个游戏, 那么 CP-ABPRE 方案满足选择访问结构和选择明文攻击下的不可区分性(Indistinguishability Selective Access Structure and Chosen Plaintext Attack, IND-sAS-CPA)。在游戏中, $\mathcal{A}$ 是敌手, $\mathcal{C}$ 是挑战者, k 和 U 分别是安全参数和系统属性集合。

预备阶段 $\mathcal{A}$ 选择挑战的访问结构 (M^*,ρ^*)。

初始化 $\mathcal{C}$ 运行 *Setup* (1^k), 输出系统公共参数 PK 和系统主密钥 MK, 将公钥 PK 发送给 $\mathcal{A}$, 主密钥 MK 自行保存。

查询阶段 1 在本阶段, 敌手 $\mathcal{A}$ 可以重复执行以下任何询问。

(1) 密钥提取查询 $O_{KG}(S)$: $\mathcal{A}$ 输入一个属性集合 S, 该属性集合 S 不满足挑战访问结构 (M^*,ρ^*)。$\mathcal{C}$ 运行 *KeyGen* (PK, MK S) $\rightarrow$ (PK_U, SK), 然后将用户公私钥 PK_{UA} 和 SK_A 交给 $\mathcal{A}$。

(2) 重加密密钥提取查询 $O_{RK}(S,(M'\rho'))$: $\mathcal{A}$ 输入一个属性集合 S, 该属性集合 S 不满足挑战访问结构 (M^*,ρ^*) 和另一个访问结构 $(M'\rho')$。$\mathcal{C}$ 运行 *RekeyGen* (PK, PK_U, SK, W') $\rightarrow$ (RK), 然后将重加密密钥 RK 发送给 $\mathcal{A}$。

挑战阶段 $\mathcal{A}$ 输出两个等长的消息 m_0 和 m_1 提交给 $\mathcal{C}$, $\mathcal{C}$ 随机选择 $b\in\{0,1\}$, 运行 *Encrypt*$_{on}$ (m, IC) $\rightarrow$ (CT)给 $\mathcal{A}$。

查询阶段 2 $\mathcal{A}$ 继续类似阶段 1 中的询问。

猜测 $\mathcal{A}$ 输出一个猜测值 $b'\in\{0,1\}$, 如果 $b'=b$, $\mathcal{A}$ 赢得游戏。$\mathcal{A}$ 赢得游戏的优势被定义为 $\varepsilon=A\mathrm{dv}(1^k,U)=|\Pr[b'=b]-\frac{1}{2}|$。

4 算法方案

4.1 方案流程

本方案流程分为: 数据共享流程和属性撤销流程两个部分, 其中数据共享流程如图 3 所示, 动态成员流程如图 4 所示。

4.2 方案构造

在本方案中, 包含初始化、密钥生成、离线加密、在线加密、重加密密钥生成、重加密、重加密验证、解密和动态成员管理共 9 个算法。

(1) 初始化 *Setup* (1^k): k 为系统安全参数, KGC 选择选取阶为大素数 p 的双线性群循环群 G 和 G_T, 使其满足双线性映射 $\hat{e}:G\times G\rightarrow G_T$, 记 $g\in G$ 为 G 的生成元, 初始的用户集合 Ω 为 $\varnothing$。从群 G 中选取一个随机数 g_0, 对于系统属性集合 U 中的每一个属性 i, 选取 $h_i\in Z_p^*$ 计算 $T_i=g^{h_i}$, 选择 $\alpha,\beta\in Z_p^*$, 定义编码 $E:G\rightarrow G_T$, KGC 构造系统公共参数 PK 并向云服务器和所有用户公开, 构造系统主密钥 MK 并进行保存。其中系统公共参数 $PK=<G,G_T,g,g_0,\hat{e}(g,g)^{\alpha},E,\Omega,T_i>$, 系统主密钥为 $MK=<g^{\alpha},\beta>$。

(2) 密钥生成 *KeyGen* (PK, MK, S): 对于重加密授权者 A 来说, KGC 随机选取 $t_u\in Z_p^*$ 作为 A 的身份标识, 对于 A 属性集合 S_A 中的每一个属性 x, 计算 $K_A=g^{\beta t_u}$, $K_{Ax}=g^{\alpha}T_x^{\beta t_u}$, 用户 A 的公钥为 PK_{UA}=K_A, 私钥为 $SK_A=\{K_{Ax}\}_{\forall x\in S_A}$。同理, 数据解密者 B 的公钥为 PK_{UB}=K_B, 私钥为 $SK_B=\{K_{Bx}\}_{\forall x\in S_B}$, 并且更新用户集合 $\Omega=\Omega\bigcup\{A\}\bigcup\{B\}$。

(3) 离线加密 *Encrypt*$_{off}$ (PK): DO 随机选取 $s,\lambda'\in Z_p^*$, 计算 $key=e(g,g)^{\alpha s}$, $C_0=g^{key}$, $C'=g_0^s$, $C''=g^{\lambda'}$, $C_i=T_i^{\lambda'}$, 其中 $i\in[1,P]$, P 是系统属性集合 U 中的属性个数, 输出临时密文 $IC=<key,C_0,C',C'',\forall i\in[1,P](C_i)>$。

(4) 在线加密 *Encrypt*$_{on}$ (m, W, IC): DO 选取访问结构 $W=(M,\rho)$ 对输入数据 m 进行加密。其中 $\boldsymbol{M}$ 是一个 $l\times q$ 的矩阵, 且满足 $l\leqslant|P|$, ρ 是矩阵每一行 M_i 到属性 $\rho(i)$ 的映射关系, DO 随机选取 $v_2,v_3,\ldots,v_q,\in Z_p^*$, 构成一个列向量 $(s,v_2,v_3,\cdots,v_q)^{\mathrm{T}}$, 计算 $\lambda_i=M_i\vec{v}$, 并计算原始密文分量 $C=m\cdot key=m\hat{e}(g,g)^{\alpha s}$, $C_{ti}=\lambda_i-\lambda'$ 并生成原始密文 $CT=<W,C,C_0,C',C'',\{C_i,C_{ti}\}_{1\leqslant i\leqslant l}>$。

(5) 重加密密钥生成 *RekeyGen* (PK, PK_U, SK, W'): RA 选取新的访问结构 $W'=(M',\rho')$, 其中 M' 是一个 $l'\times q'$ 的矩阵, ρ' 是矩阵每一行 M_i' 到属性 $\rho'(i)$ 的映射关系。RA 选取随机数 $d\in Z_p^*$, 分别计算 g^d 和 g_0^d, 对 g^d 进行编码得 $E(g^d)$, 再根据密文策略属性基的方法使用 W' 访问结构对 $E(g^d)$ 进

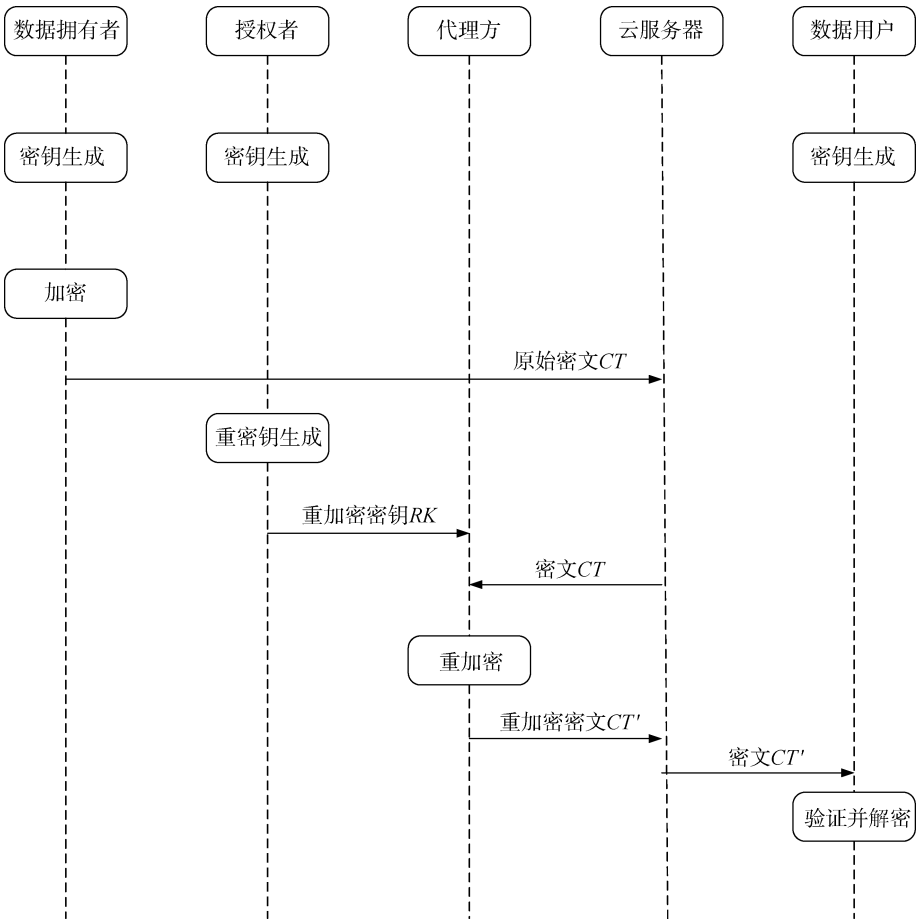

图 3　数据共享流程

Figure 3　Data sharing process

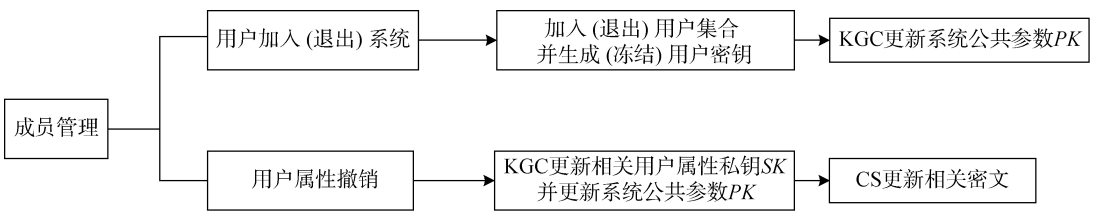

图 4　动态成员流程

Figure 4　Dynamic membership process

行加密得到密文 C^*，同时计算 $K_{Ax}^* = g_0^d K_{Ax} = g_0^d g^{\alpha} T_x^{\beta t_u}$，生成重加密密钥为：$RK = < S_A, K_A, \{K_{Ax}^*\}_{\forall x \in S_A}, C^* >$。

(6) 重加密 *ReEncrypt* (PK，RK，CT): 收到重加密密钥 RK 的 STA 先进行判断, 当密文被设置为不能重加密或者 $S_A \models (M, \rho)$ 时, 输出 $\perp$, 反之, 重加密密钥 RK 是合法的。选择常数 ω_i 使得 $\sum_{\rho(i) \in S_A} \omega_i M = (1, 0, 0, \cdots, 0)$, 半可信代理方 STA 计算

$$C_q = \prod_{i \in S_A} \left(\frac{\hat{e}(K_{Ai}^*, g^{C_{ti}} C'')}{\hat{e}(K_A, T_i^{C_{ti}} C_i)} \right)^{\omega_i}$$

生成重加密密文 $CT' = < W', C, C_0, C', C_q, C^* >$。

(7) 重加密验证 *ReEncryptVerify* (C_q, C', C_0, C^*): 解密者 B 使用重加密后的 C_q, C', C_0, C^* 对半可信代理 Proxy 重加密结果进行正确性验证, 对于未被重加密的密文 C^*, 解密者 B 可以恢复出解密重加密密文的秘密值 m^*, 并得到 g^d, 接着计算

$$T = \frac{C_q}{\hat{e}(C', g^d)} = \hat{e}(g^{\alpha}, g^s)$$

若 $C_0 = g^T$，则说明 STA 计算结果正确，输出 *true*，否则直接输出⊥。

(8) 解密 *Decrypt* (PK， PK_{UB}， SK_B， S_B CT'): 该算法先检查密文解密者 B 的属性集合 S_B 是否满足 LSSS 访问结构 W'，若不满足，直接输出⊥，否则解密者 B 可以利用公私钥对重加密密文 $CT' =< W', C, C_0, C', C_q, C^* >$ 进行解密操作，恢复出明文消息 m。

对于未被重加密的密文 C^*，解密者 B 可以恢复出解密重加密密文的秘密值 m^*，计算如下:

$$F = \prod_{i \in S_B} (\frac{\hat{e}(K_{Bi}, g^{C_{ti}} C'')}{\hat{e}(K_B, T_i^{C_{ti}} C_i)})^{\omega_i}$$

$$m' = \frac{C}{F} = E(g^d)$$

解码得 $m^* = g^d$，接着对重加密密文进行解密计算 $m = \frac{C\hat{e}(g^d, C')}{C_q}$，进而恢复出明文消息 m。

(9) 动态成员管理算法 *DMML*：对于 $\varphi \in \Omega$ 的用户来说，当用户 φ 退出系统时，KGC 将其唯一身份值 t_u 从列表中删除，并将该用户 φ 从用户集合 Ω 中撤出，更新 $\Omega = \Omega \setminus \{\varphi\}$，并冻结用户密钥。当用户 φ 撤销属性 y 时，其 KGC 就会进行以下操作 $x \in U$，$if(x = y)$：选取 $h_{y'} \in Z_p^*$，并计算 $UK = \frac{h_{y'}}{h_x}$，$T_{y'} = T_x^{UK}$，$K_{y'} = g^{\alpha} T_{y'}^{\beta t_u}$，更新系统公共参数中的 T_i 和系统中与属性 y 相关用户的属性私钥 SK 分别为 $T_i = (T_{y'}, \forall i \in U \setminus \{y\} : T_i)$，$SK = (K_{y'}, \forall x \in S \setminus \{y\} : K_x)$，并将转换密钥 UK 发送至云服务器，CS 将收到的转换密钥用于更新与撤销属性相关的密文，结果如下:

原始密文 $CT =< W, C, C_0 C', C'', \forall 1 \le i \le l, C_{ti},$

$$if(x \ne y) : C_i, if(x = y) : C_i^{UK} >$$

同理，由于重加密密文 CT' 中的密文分量 C^* 也为原始密文，其策略为 W'，所以也进行同样的更新。

4.3 正确性分析

首先对重加密密文中密文分量 C^* 解密的正确性验证如下:

$$F = \prod_{i \in S_B} (\frac{\hat{e}(K_{Bi}, g^{C_{ti}} C'')}{\hat{e}(K_B, T_i^{C_{ti}} C_i)})^{\omega_i}$$

$$= \prod_{i \in S_B} (\frac{\hat{e}(g^{\alpha} T_i^{\beta t_u}, g^{\lambda_i - \lambda'} g^{\lambda'})}{\hat{e}(g^{\beta t_u}, T_i^{\lambda_i - \lambda'} T_i^{\lambda'})})^{\omega_i}$$

$$= \prod_{i \in S_B} (\frac{\hat{e}(g^{\alpha} T_i^{\beta t_u}, g^{\lambda_i})}{\hat{e}(T_i^{\beta t_u}, g^{\lambda_i})})^{\omega_i}$$

$$= \prod_{i \in S_B} \hat{e}(g^{\alpha}, g^{\lambda_i})^{\omega_i}$$

$$= \hat{e}(g, g)^{\alpha \sum_{i \in S_B} \lambda_i \omega_i}$$

$$= \hat{e}(g, g)^{\alpha s}$$

$$m' = \frac{C}{F}$$

$$= E(g^d)$$

解码得 $m^* = g^d$;

接着对重加密密文 CT' 解密的正确性验证如下:

$$m = \frac{C\hat{e}(g^d, C')}{C_q}$$

$$= \frac{m\hat{e}(g^{\alpha}, g^s)\hat{e}(g^d, g_0^s)}{\hat{e}(g^{\alpha} g_0^d, g)^s}$$

$$= m$$

5 安全性分析

定理 1. 若判定性 q -parallel BDHE 假设在(G， G_T)上成立，那么没有一个概率多项式时间(PPT)的敌手 $\mathcal{A}$ 能够选择选择 $l^* \times n^*$ ($l^*, n^* \leqslant q$)大小的挑战访问矩阵 $(\boldsymbol{M}^*, \rho^*)$ 来攻破 CP-ABPRE 方案，即该方案在标准模型下可证 IND-sAS-CPA 安全。

证明: 假设存在一个概率多项式时间(PPT)的敌手 $\mathcal{A}$ 能在 IND-sAS-CPA 游戏中以 $\varepsilon = Adv_{\mathcal{A}}$ 的优势攻破该方案，那么就可以构建一个挑战者 $\mathcal{C}$ 能够以不可忽略的优势攻破判定性 q -parallel BDHE 假设。$\mathcal{C}$ 和 $\mathcal{A}$ 进行如下 IND-sAS-CPA 游戏:

$\mathcal{C}$ 输入(p， g， G， G_T， $\hat{e}$)， q -parallel BDHE 假设中的 $\vec{y}$ 和 T，判定 $T = \hat{e}(g, g)^{a^{q+1} s}$ 还是 $T = G_T$。

初始化. $\mathcal{A}$ 将要挑战的访问结构 $(\boldsymbol{M}^*, \rho^*)$ 给 $\mathcal{C}$，$\boldsymbol{M}^*$ 是一个 $l^* \times n^*$ 大小的矩阵，l^* 是行数，n^* 是列数，$l^*, n^* \leqslant q$。

阶段 1. $\mathcal{C}$ 初始化空表 SK^{List}, 在该阶段挑战者 $\mathcal{C}$ 回答敌手 $\mathcal{A}$ 提出的一系列询问。

(1) 密钥提取询问 $O_{KG}(S)$: $\mathcal{A}$ 对不满足访问矩阵 $\boldsymbol{M}^*$ 的属性集合 S 做私钥提取询问。$\mathcal{A}$ 输入一个属性集合 S, $\mathcal{C}$ 选择一个随机数 $r\in Z_{\mathrm{p}}^*$, 求向量 $\vec{\boldsymbol{\omega}}=(\omega_1,\omega_2,\ldots,\omega_{n^*})\in Z_p^{n^*}$ 满足 $\omega_1=-1$ 并且对于满足使得 $\rho^*(i)\in S$ 的所有 i, 都有 $\vec{\boldsymbol{\omega}}\cdot\boldsymbol{M}_i^*=0$, 接着 $\mathcal{C}$ 生成公钥 PK_U, 计算属性集合 S 对应的私钥 SK。最后, $\mathcal{C}$ 将元祖 (S,SK) 添加到表 SK^{List} 中, 并返回 PK_U、SK 给 $\mathcal{A}$。

(2) 重加密密钥提取询问 $O_{RK}(S,(M'\rho'))$: $\mathcal{A}$ 用一个属性集合 S 和一个新的访问结构 $(M'\rho')$ 来查询 O_{RK}。$\mathcal{A}$ 输入一个属性集合 S, 若属性集合 S 满足挑战访问结构 (M^*,ρ^*), 则 $\mathcal{C}$ 在 $\{0,1\}$ 中任意输出一个值并终止本次游戏; 否则, $\mathcal{C}$ 运行 $RekeyGen(PK, SK, W'=(M'\rho'))\rightarrow RK$, 然后将重加密密钥 RK 发送给 $\mathcal{A}$。

挑战阶段.敌手 $\mathcal{A}$ 输出两个等长的消息 m_0 和 m_1 提交给 $\mathcal{C}$, $\mathcal{C}$ 随机选择 $b\in\{0,1\}$ 并进行加密操作, 然后将挑战密文 CT^* 返回给 $\mathcal{A}$。

阶段 2. 敌手 $\mathcal{A}$ 继续向挑战者 $\mathcal{C}$ 进行类似阶段 1 的询问, 其限制与阶段 1 相同。

猜测.敌手 $\mathcal{A}$ 输出一个对 b 的猜测值 b', 其中 $b'\in\{0,1\}$, 如果 $b'=b$, 那么游戏中挑战者 $\mathcal{C}$ 输出 1 来猜测 $T=\hat{e}(g,g)^{a^{q+1}s}$, 否则输出 0 表示 $T=G_T$。

当输出为 1 时, 也就是当 $T=\hat{e}(g,g)^{a^{q+1}s}$, 即为敌手 $\mathcal{A}$ 得到了有个关于 m_b 的有效密文。通过安全模型中的敌手 $\mathcal{A}$ 的优势定义 $Adv_{\mathcal{A}}=|\Pr[b'=b]-\frac{1}{2}|$, 可以知道:

$$Adv_{\mathcal{A}}=|\Pr[b'=b]-\frac{1}{2}|=|\Pr[\mathrm{B}(\vec{y},T=\hat{e}(g,g)^{a^{q+1}s})=0]-\frac{1}{2}|$$

当输出为 0 时, 也就是当 $T=G_T$, 即为敌手 $\mathcal{A}$ 得不到关于 m_b 的任何消息, 因此, $|\Pr[\mathrm{B}(\vec{y},T=R)=0]-\frac{1}{2}|$, 此时挑战者 $\mathcal{C}$ 的优势为:

$$Adv_B=\frac{1}{2}\Pr[\mathrm{B}(\vec{y},T=\hat{e}(g,g)^{a^{q+1}s})=0]+\frac{1}{2}\Pr[\mathrm{B}(\vec{y},T=R)=0]-\frac{1}{2}=\frac{1}{2}(\frac{1}{2}+Adv_A)+\frac{1}{4}-\frac{1}{2}=\frac{\varepsilon}{2}$$

由于敌手 $\mathcal{A}$ 的优势 ε 是不可忽略的, 因此挑战者 $\mathcal{C}$ 也有不可忽略的优势 $\frac{\varepsilon}{2}$ 攻破 q -parallel BDHE 困难问题。

证毕

定理 2. 假设一个单向非转移的 CP-ABPRE 方案是 IND-sAS-CPA 安全的, 那么该方案是选择可抵抗同谋攻击的。

证明: 在 IND-sAS-CPA 游戏中, 当 $\mathcal{A}$ 的属性集合分别满足 $S|=(M^*,\rho^*)$ 和 $S'|=(M',\rho')$ 时, 就可从 O_{RK} 中获取到 $RK_{S\rightarrow(M',\rho')}$ 和 $RK_{S'\rightarrow(M'',\rho'')}$。因为在游戏中定义的限制, $\mathcal{A}$ 不能询问任意 $S'|=(M',\rho')$ 的私钥 $SK_{S'}$, 也不能询问 $S|=(M^*,\rho^*)$ 的私钥 SK_S, 但可以询问任意的 $S''|=(M'',\rho'')$ 的私钥 $SK_{S''}$。

假设一个 sAS-CPA-CP-ABPRE 是不能抵抗同谋攻击的。那么 $\mathcal{A}$ 能从 $RK_{S'\rightarrow(M'',\rho'')}$ 通过同谋获取 $SK_{S'}$ 并通过询问获取 $SK_{S''}$, 那么 $\mathcal{A}$ 就可以使用 $RK_{S\rightarrow(M',\rho')}$ 重加密挑战密文 CT^*, 生成 CT'。最后 $\mathcal{A}$ 用 $SK_{S'}$ 解密重加密密文 CT', 从而输出 b 的值。显然, 这与 IND-sAS-CPA 安全完全矛盾。

在本方案中, 授权者 A 的属性集合为 S_A, 对于每个属性 x, 其对应的私钥为 $K_{Ax}=g^{\alpha}T_x^{\beta t_u}$, 在代理方的重加密密钥中, 有关授权者属性私钥的部分为 $K_{Ax}^*=g_0^dK_{Ax}=g_0^dg^{\alpha}T_x^{\beta t_u}$, 要想获取授权者的私钥必须要求出秘密值 g_0^d, 若代理方和受理者同谋, 受理者可通过自己的私钥获得 g^d, 而无法得到 g_0^d, 即不能获得授权者的私钥。

证毕

6 性能分析

6.1 功能对比

本文提出的动态更新的属性基代理重加密方案与文献[17-19]、文献[21]在算法功能上进行分析比较, 比较内容分为加密方式、访问控制结构、困难问题、重加密验证和可控性, 如表 2 所示, 从中可以得出结论: 方案[17]和[18]只支持与门的访问结构, 而方案[19]、[21]和本方案都是利用基于矩阵来进行秘密共享的, 所以访问控制结构支持任意单调的访问策略; 由于本方案在加密阶段利用了离线机密技术, 所以, 本方案在用户在线加密阶段效率更高。此外, 本方案还加入重加密验证算法, 用户只需要进行一次双线性运算和一次指数运算就能够验证重加密密

表 2 不同方案的功能对比

Table 2 Function comparison of different schemes

方案	离线加密	访问控制结构	困难问题	重加密验证	可控性
[17]	否	正负属性的与门	CTDH, ADBDH	否	是
[18]	否	多属性的与门	CBDH, DBDH	否	是
[19]	否	任意单调的访问策略	q-parallel BDHE	否	否
[21]	否	任意单调的访问策略	DBDH	是	是
本方案	是	任意单调的访问策略	q-parallel BDHE	是	是

文的正确性, 避免半可信的第三方代理对数据进行非法操作, 通过在重加密密文中加入一个原始密文来增加验证项实现了数据的可控性。

6.2 性能对比

此外, 我们将本方案与其他方案从计算开销和通信开销两个方面进行对比分析, 并讨论本文所提方案的性能。

在计算开销中, 对比具体包括: 用户密钥生成、加密、重加密密钥生成以及重加密, 为了方便表述, 令 N 表述系统中属性 i 的可能取值个数, 令 n 表示用户属性集合中的属个数, A_n 表示策略密文中的属性个数。本次对比只包含最耗时间的运算, 包括: 双线性运算 T_p , 指数运算 T_E, 其中 $T_{E(\mathrm{G})}$ 和 $T_{E(T)}$ 分别表示群 G 、群 G_T 中的指数运算。观察表 3 可知, 在密钥生成阶段, 其他方案计算开销中的指数运算个数基本都与系统属性个数成多倍关系, 文献[21]中和本方案其计算开销只与用户属性个数有关, 并且本方案成单倍关系; 加密算法中, 由于本方案使用离线加密技术, 所以在线加密只需执行一次指数运算即可; 重加密算法中, 本方案的计算开销是 $O(1)$ 级别; 本文在重加密算法里的计算开销也是只与用户属性个数有关, 与系统属性个数无关, 比例系数小。所以, 本方案在以上算法中的计算开销均低于其他方案, 有较好的实用性。

表 3 不同方案的计算开销对比

Table 3 Comparison of computing cost of different schemes

方案	密钥生成	加密	重密钥生成	重加密
[17]	$(2N+1)T_{E(G)}$	$(N+2)T_{E(G)}+T_{E(T)}$	$(3N+3)T_{E(G)}+T_{E(T)}$	$(N+1)T_p$
[18]	$3NT_{E(G)}$	$(N+2)T_{E(G)}+T_{E(T)}$	$(3N+3)T_{E(G)}+T_{E(T)}$	$(2N+1)T_p$
[20]	$9NT_{E(G)}$	$(12N+2)T_{E(G)}+T_{E(T)}$	$(25N+2)T_{E(G)}+T_{E(T)}$	$(4N+2)T_p$
[21]	$(2N+5)T_{E(G)}$	$(A_n+7)T_{E(G)}+T_{E(T)}$	$(A_n+10)T_{E(G)}+T_{E(T)}$	$(2n+1)T_p+nT_{E(T)}$
本方案	$(n+1)T_{E(G)}$	$T_{E(T)}$	$2T_{E(G)}+T_{E(T)}$	$2nT_p+nT_{E(T)}$

在通信开销的, 对比具体包括: 系统主公钥长度、主密钥长度、用户密钥长度、密文长度、重加密密文长度。其中 $|G|$ 表示群 G 中的元素长度, $|G_T|$ 表示群 G_T 中的元素长度, $|Z|$ 表示域 Z_p^* 中的元素长度。由表 4 可以看出, 本方案中的系统公钥长度和系统主密钥长度比其他方案都小, 且系统密钥长度为定长与属性个数无关, 在用户密钥生成阶段, 其他方案的用户私钥长度与用户属性个数的比值都在 2 以及以上, 而本方案的用户密钥长度与属性个数 n 只是单倍关系, 每个用户只拥有一个公钥和 n 个属性私钥, 并且本文密文长度和重加密密文长度都较小。所以, 在本方案中, 用户客户端存储空间占用较小, 且在用户与服务器之间或代理方与服务器之间的数据传输过程中, 通信开销也较小。

6.3 效率对比

本方案的仿真实验: 采用操作系统版本为 Windows 10 家庭中文版, 处理器是 AMD Ryzen 7 5800H with Radeon Graphics 3.20 GHz, 运行内存为 16.00GB 的电脑, 采用 JPBC(Java pairing-based cryptography)库进行实验代码的编写。

图 5 中描述了本方案的各个算法的计算开销情况。在实现过程中, 将用户的属性个数选取为 5 的整数倍, 直至 100 个, 而用户的最少属性个数设置为 1。对每个算法多次独立测试, 最后取得平均值。

表 4 不同方案的通信开销对比

Table 4 Comparison of communication overhead of different schemes

方案	公钥长度	主密钥长度	用户密钥长度	密文长度	重加密密文长度
[17]	$(2N+2)\|G\|+\|G_T\|$	$(N+1)\|Z\|$	$(2n+1)\|G\|$	$(A_n+2)\|G\|+\|G_T\|$	$(A_n+3)\|G\|+3\|G_T\|$
[18]	$(3N+3)\|G\|+\|G_T\|$	$(3N+1)\|Z\|$	$(3n+1)\|G\|$	$(A_n+2)\|G\|+\|G_T\|$	$(A_n+3)\|G\|+3\|G_T\|$
[20]	$(8N+6)\|G\|+\|G_T\|$	$(8N+4)\|Z\|$	$(4n+2)\|G\|$	$(12A_n+2)\|G\|+\|G_T\|$	$(12A_n+4)\|G\|+3\|G_T\|$
[21]	$(N+6)\|G\|+\|G_T\|$	$2\|Z\|+\|G\|$	$(2n+4)\|G\|$	$(2A_n+5)\|G\|+\|G_T\|$	$(2A_n+7)\|G\|+3\|G_T\|$
本方案	$(N+2)\|G\|+\|G_T\|$	$\|G\|+\|Z\|$	$(n+1)\|G\|$	$(2A_n+3)\|G\|+\|G_T\|$	$(2A_n+5)\|G\|+3\|G_T\|$

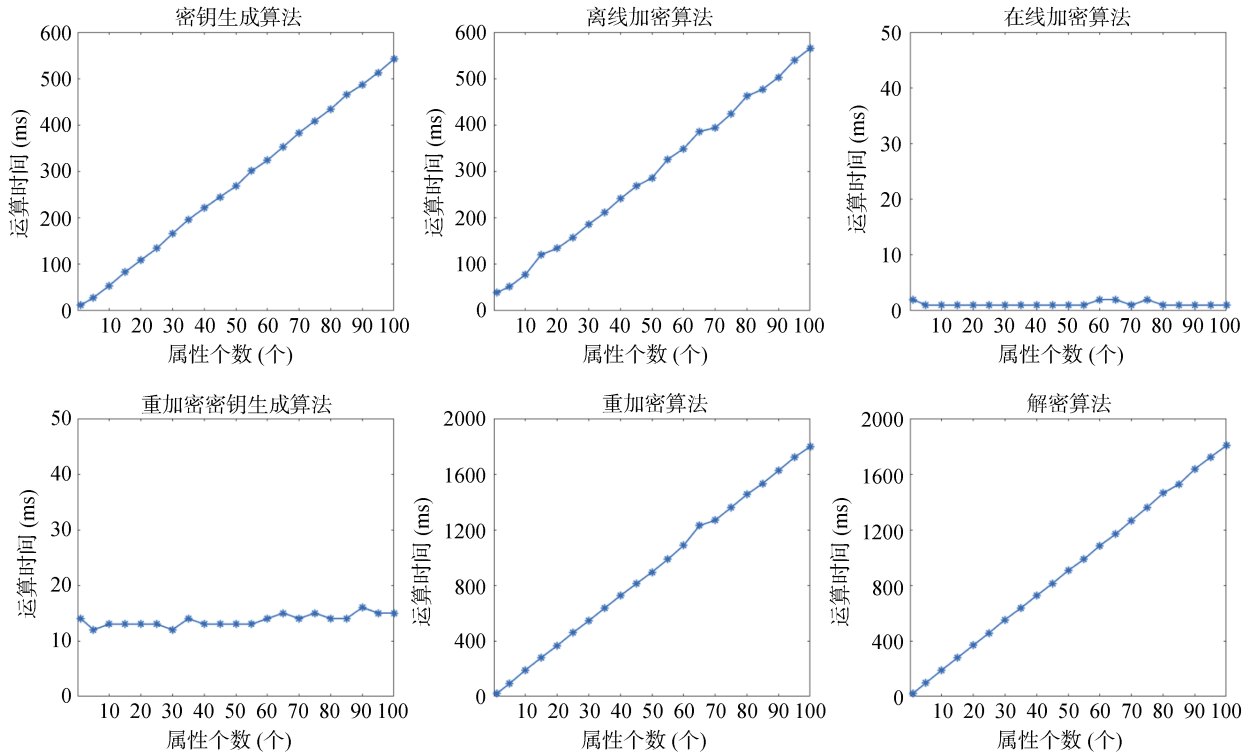

图 5 各算法的计算开销

Figure 5 The computational cost of each algorithm

由此图可知, 在密钥生成阶段, 即使当用户属性个数达到 100 时, KGC 不到 0.6s 可为用户产生公私钥对; 在加密阶段, 通过采用离线加密技术, 将大部分的加密计算都交给客户端离线阶段进行操作, 在线加密的计算效率是非常快的, 而且与用户属性个数无关, DO 几乎不消耗时间就可以生成完整密文, 提高了加密速度; 在重加密密钥生成算法中, RA 需要先进行加密操作, 再将自己的属性私钥和选取的秘密值进行乘法运算, 因为加密效率高且乘法的时间开销低, 所以重加密密钥生成的时间开销与属性个数几乎无关; 在重加密算法中, 因为要进行密文策略的转换操作, 需要先对原有的策略进行解密消除, 因此, 时间开销与密文策略中的属性个数成正比, 当属性个数达到 100 个时, 运行时间约为 1.8s, 在本方案中, 重加密算法是代理方执行的, 所以达到了计算外包; 在解密阶段, DU 先对重加密密文的分量进行按原始密文的进行解密操作, 再用秘密值进行对重加密密文的解密操作, 当用户的属性个数达到 100 时, 时间消耗为 1.8s 左右; 由于重加密代理方是半可信的, 所以本方案加入重加密验证算法, 受理者可以验证重加密密文的正确性和完整性, 避免代理方对数据造成破坏, 解密者 B 只需要进行一次双线性运算和一次指数运算就能够验证重加密密文的正确性; 在动态成员管理算法中, 本方案将所有的计算开销都外包给密钥生成中心 KGC 和云服务器 CS, 当系统中的用户集合和用户的属性集合发生更新时, KGC 生成转换密钥, 为相关用户进行属性私钥的更新, 并更新系统公共参数, CS 利用转换密

钥对相关密文进行更新操作。

7 结束语

本文针对在云系统中大量密文都需要进行转换和实际应用中用户属性集合具有动态变化的问题, 为了实现数据灵活的访问与共享和用户权限的动态更新, 提出了云中可动态更新的属性基代理重加密方案。此外, 本文给出了算法的具体实现过程和方案的流程图, 方案在系统公共参数中加入用户集合, 实现用户身份的动态加入和退出, 当用户的属性集合发生更新时, 密钥生成中心会及时更新相关用户的密钥并生成转换密钥, 且云服务利用转换密钥更新相关的密文组件, 实现了计算外包。为了减少属性加密的计算开销, 利用离线加密技术大大提高了用户端的加密效率, 加入重加密验证算法来判断重加密密文的正确性以防止重加密过程中第三方对数据的恶意破坏。在标准模型下、判定性 q -parallel BDHE 假设下进行安全性分析, 证明本方案在选择明文攻击下满足密文不可区分性, 且抵抗同谋攻击。通过仿真分析表明, 本方案在功能对比中具有很好的实用价值, 且通信开销和计算开销较低。

参考文献

[1] Yao S M, Dayot R V J, Kim H J, et al. A Novel Revocable and Identity-Based Conditional Proxy re-Encryption Scheme with Ciphertext Evolution for Secure Cloud Data Sharing[J]. *IEEE Access*, 2021, 9: 42801-42816.

[2] Rathod S, Ubale S A, Apte S S. Attribute-Based Encryption along with Data Performance and Security on Cloud Storage[C]. *2018 International Conference on Information , Communication, Engineering and Technology*, 2018: 1-3.

[3] Su M, Wu B, Fu A M, et al. Assured Update Scheme of Authorization for Cloud Data Access Based on Proxy re-Encryption[J]. *Journal of Software*, 2020, 31(5): 1563-1572.
(苏铓, 吴槟, 付安民, 等. 基于代理重加密的云数据访问授权确定性更新方案[J]. *软件学报*, 2020, 31(5): 1563-1572.)

[4] Ramteke A, Talmale G. Access Control Mechanism for Multi-User Data Sharing in Social Networks[C]. *2014 Fourth International Conference on Communication Systems and Network Technologies*, 2014: 578-582.

[5] Gao J T, Yu H Y, Zhu X Q, et al. Blockchain-Based Digital Rights Management Scheme via Multiauthority Ciphertext-Policy Attribute-Based Encryption and Proxy re-Encryption[J]. *IEEE Systems Journal*, 2021, 15(4): 5233-5244.

[6] Yasumura Y, Imabayashi H, Yamana H. Attribute-Based Proxy re-Encryption Method for Revocation in Cloud Storage: Reduction of Communication Cost at re-Encryption[C]. *2018 IEEE 3rd International Conference on Big Data Analysis*, 2018: 312-318.

[7] Nyame G, Qin Z G. Precursors of Role-Based Access Control Design in KMS: A Conceptual Framework[J]. *Information*, 2020, 11(6): 334.

[8] Dou L J. Research on Multi-Domain Cloud Access Control of Optical Network Information Based on Role Hierarchy Tree[J]. *Laser Journal*, 2021, 42(1): 139-143.
(窦立君. 基于角色等级树的光网络信息多域云访问控制研究[J]. *激光杂志*, 2021, 42(1): 139-143.)

[9] Chen Y R. A Scheme Based on CPK Role Access Control[J]. *Journal of Information Security Research*, 2021, 7(2): 184-189.
(陈亚茹. 一种基于 CPK 角色访问控制的方案[J]. *信息安全研究*, 2021, 7(2): 184-189.)

[10] Sahai A, Waters B. Fuzzy Identity-Based Encryption[M]. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005: 457-473.

[11] Goyal V, Pandey O, Sahai A, et al. Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data[C]. *The 13th ACM conference on Computer and communications security*, 2006: 89-98.

[12] Li J G, Yu Q H, Zhang Y C, et al. Key-Policy Attribute-Based Encryption Against Continual Auxiliary Input Leakage[J]. *Information Sciences*, 2019, 470: 175-188.

[13] Ma X X, Huang Y. Publicly Traceable Accountable Ciphertext Policy Attribute Based Encryption Scheme Supporting Large Universe[J]. *Computer Science*, 2020, 47(S1): 420-423.
(马潇潇, 黄艳. 大属性可公开追踪的密文策略属性基加密方案[J]. *计算机科学*, 2020, 47(S1): 420-423.)

[14] Ostrovsky R, Sahai A, Waters B. Attribute-Based Encryption with Non-Monotonic Access Structures[C]. *The 14th ACM conference on Computer and communications security*, 2007: 195-203.

[15] Ma H Y, Zeng G S, Wang Z J, et al. Efficient and Provably Secure Attribute-Based Online/ Offline Encryption Schemes[J]. *Journal on Communications*, 2014, 35(7): 104-112.
(马海英, 曾国荪, 王占君, 等. 高效可证明安全的基于属性的在线/离线加密机制[J]. *通信学报*, 2014, 35(7): 104-112.)

[16] Su H, Zhu Z Q, Sun L. Online/Offline Attribute-Based Encryption with Keyword Search Against Keyword Guessing Attack[C]. *2017 3rd IEEE International Conference on Computer and Communications*, 2017: 1487-1492.

[17] Liang X H, Cao Z F, Lin H, et al. Attribute Based Proxy re-Encryption with Delegating Capabilities[C]. *The 4th International Symposium on Information, Computer, and Communications Security*, 2009: 276-286.

[18] Luo S, Hu J B, Chen Z. Ciphertext Policy Attribute-Based Proxy Re-encryption[M]. Information and Communications Security. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010: 401-415.

[19] Li K Y. Matrix Access structure Policy used in Attribute-Based Proxy Re-encryption[J]. *International Journal of Computer Science Issues*. 2013. 9.

[20] Sepehri M, Trombetta A. Secure and Efficient Data Sharing with Atribute-Based Proxy re-Encryption Scheme[C]. *The 12th International Conference on Availability, Reliability and Security*, 2017: 1-6.

[21] Feng C S, Luo W P, Qin Z G, et al. Attribute-Based Proxy

re-Encryption Scheme with Multiple Features[J]. *Journal on Communications*, 2019, 40(6): 177-189.
(冯朝胜, 罗王平, 秦志光, 等. 支持多种特性的基于属性代理重加密方案[J]. *通信学报*, 2019, 40(6): 177-189.)

[22] Hui Cui, Robert H. Deng, Yingjiu Li, et al. Server-aided revocable attribute-based encryption[C]. *The European Symposium on Research in Computer Security, Springer*: 570–587.

[23] Qin B D, Zhao Q L, Zheng D, et al. (Dual) Server-Aided Revocable Attribute-Based Encryption with Decryption Key Exposure Resistance[J]. *Information Sciences*, 2019, 490: 74-92.

[24] Attrapadung N, Imai H. Conjunctive Broadcast and Attribute-Based Encryption[M]. Pairing-Based Cryptography-Pairing 2009. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009: 248-265.

[25] Zhang Y H, Chen X F, Li J, et al. FDR-ABE: Attribute-Based Encryption with Flexible and Direct Revocation[C]. *2013 5th International Conference on Intelligent Networking and Collaborative Systems*, 2013: 38-45.

[26] Song S, Zhang X L. Attribute-Based Encryption Scheme without Key Escrow Supporting Attribute Revocation in Cloud Environment[J]. *Netinfo Security*, 2020, 20(8): 62-70.
(宋硕, 张兴兰. 云环境下支持属性撤销的无密钥托管属性基加密方案[J]. *信息网络安全*, 2020, 20(8): 62-70.)

[27] Chen Y Q, Wang Y. Efficient Conversion Scheme of Access Matrix in CP-ABE with Double Revocation Capability[C]. *2020 IEEE International Conference on Progress in Informatics and Computing*, 2020: 352-357.

[28] Huang K Q. Online/Offline Revocable Multi-Authority Attribute-Based Encryption for Edge Computing[C]. *2020 12th International Conference on Measuring Technology and Mechatronics Automation*, 2020: 563-568.

[29] Wang H Y, Peng Y, Guo K X. CP-ABE Access Control Scheme Based on Proxy re-Encryption in Cloud Storage[J]. *Journal of Computer Applications*, 2019, 39(9): 2611-2616.
(王海勇, 彭垚, 郭凯璇. 云存储中基于代理重加密的 CP-ABE 访问控制方案[J]. *计算机应用*, 2019, 39(9): 2611-2616.)

[30] Xu Q, Tan C X, Fan Z J, et al. An Efficient Searchable Encryption Scheme with Designed Tester and Revocable Proxy re-Encryption[J]. *Journal of Computer Research and Development*, 2018, 55(5): 994-1013.
(徐潜, 谭成翔, 樊志杰, 等. 指定验证者与可撤销重加密的可搜索加密方案[J]. *计算机研究与发展*, 2018, 55(5): 994-1013.)

[31] Shao J Y, Zhu Y Q, Ji Q J. Privacy-Preserving Online/Offline and Outsourced Multi-Authority Attribute-Based Encryption[C]. *2017 IEEE/ACIS 16th International Conference on Computer and Information Science*, 2017: 285-291.

[32] Guo R, Yang G, Shi H X, et al. O_3R-CP-ABE: An Efficient and Revocable Attribute-Based Encryption Scheme in the Cloud-Assisted IoMT System[J]. *IEEE Internet of Things Journal*, 2021, 8(11): 8949-8963.

杨耿 于 2019 年在西安邮电大学信息对抗技术专业获得学士学位。现在西安邮电大学电子与通信工程专业攻读硕士研究生学位。研究领域为密码学。研究兴趣包括: 云计算安全、属性加密等。Email: yanggeng1996xupt@163.com

郭瑞 于 2014 年在北京邮电大学获得信息安全专业博士学位。现在西安邮电大学网络空间安全学院副教授。研究领域为云计算安全, 区块链技术。研究兴趣包括密码学、区块链等。Email: guorui@xupt.edu.cn

庄朝源 于 2019 年在西安邮电大学信息安全专业获得学士学位。现在西安邮电大学电子与通信工程专业攻读硕士研究生学位。研究领域为密码学。研究兴趣包括: 云存储安全、区块链中的隐私保护等。Email: zhuangchaoy@163.com

王旭涛 于 2019 年在西安邮电大学信息安全专业获得学士学位。现在西安邮电大学网络空间安全专业攻读硕士研究生学位。研究领域为密码学。研究兴趣包括区块链技术、代理重加密等。Email: wangxutao2019@163.com