支持多种特性的基于属性代理重加密方案

冯朝胜[1,2]，罗王平[1]，秦志光[2]，袁丁[1]，邹莉萍[1]
（1. 四川师范大学计算机科学学院，四川 成都 610101；2. 电子科技大学网络与数据安全四川省重点实验室，四川 成都 610054）

摘 要：一个理想的代理重加密方案通常具有单向性、非交互性、可重复性、可控性和可验证性，然而目前的方案普遍只满足其中的 2 个或 3 个，在一定程度上降低了实用性。为此，提出了一种支持 5 种特性的密文策略基于属性代理重加密（CP-ABPRE）方案。在所提方案中，云代理服务器只能利用重加密密钥重加密委托者指定的密文，抵御了满足重加密共享策略的用户与代理之间的共谋攻击；将多数加解密工作外包给云服务器，减轻了用户客户端的计算负担。安全分析表明，所提方案能抵御针对性选择明文攻击。

关键词：基于属性加密；代理重加密；外包加密；外包解密；选择明文安全
中图分类号：TP309
文献标识码：A
doi: 10.11959/j.issn.1000−436x.2019127

Attribute-based proxy re-encryption scheme with multiple features

FENG Chaosheng[1,2], LUO Wangping[1], QIN Zhiguang[2], YUAN Ding[1], ZOU Liping[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: A ideal proxy re-encryption scheme has five features, such as one-way encryption, non-interaction, repeatability, controllability and verifiability. The existing schemes, however, have only two or three of the five features, which reduces the utility of them to some extent. For this, a new ciphertext-policy attribute-based proxy re-encryption (CP-ABPRE) scheme with the above five features was proposed. In the proposed scheme, the cloud proxy server could only re-encrypt the ciphertext specified by the delegator by using the re-encryption key, and resist the collusion attack between the user and the proxy satisfying the re-encryption sharing policy. Most of encryption and decryption were outsourced to cloud servers so that it reduced the computing burden on the user's client. The security analysis show that the proposed scheme resists the selective chosen plaintext attack (SCPA).

Key words: attribute-based encryption, proxy re-encryption, outsourcing encryption, outsourcing decryption, chosen plaintext security

1 引言

如今，越来越多的企业将数据外包存储在云中，越来越多的人将个人信息存储在社交网络。然而，频发的信息泄露事件，使安全性和隐私性成为服务提供商必须面临的问题。解决外包数据存储安全的一种简单做法是在数据上传前进行加密，但加密又使数据共享变得非常困难。解决方法是采用密文策略基于属性加密（CP-ABE, ciphertext-policy attribute-based encryption）算法[1]，该算法因具有“一

收稿日期：2019−02−14；修回日期：2019−05−19
基金项目：国家科技支撑计划基金资助项目（No.2014BAH11F02）；国家自然科学基金资助项目（No.61373163）；网络与数据安全四川省重点实验室课题基金资助项目（No.NDS2019-1）；四川师范大学研究生优秀论文培育基金资助项目（川师研[2018]3 号-38）
Foundation Items: The National Key Technology R & D Program of the Ministry of Science and Technology of China (No.2014BAH11F02), The National Natural Science Foundation of China (No.61373163), Project of Network and Data Security Key Laboratory of Sichuan Province (No.NDS2019-1), The Postgraduate Excellent Paper Cultivation Foundation of Sichuan Normal University (Chuan Shi Yan [2018] No.3-38)

次加密，多人分享”和细粒度访问控制等优势受到人们的广泛关注[2-3]。

然而，现有的 CP-ABE 因加解密效率低和密文访问策略更改而导致的重加密效率低的问题，影响了其应用和推广。解决外包云数据重新加密的一般方案是用户先将密文数据从云服务器下载至本地，利用该用户私钥对密文数据执行一次解密算法从而得到明文数据，使用新的共享访问策略对明文数据再次加密后将新的密文数据上传至云服务器存储并共享。显然，这种方法不但加重了用户客户端的计算负担，而且增加了云服务器与用户之间的通信开销，使它难以应用于实际的环境中。为了更有效地进行数据共享，在 CP-ABE 中引入代理重加密技术（PRE, proxy re-encryption）。由于代理重加密技术允许一个半可信代理将一个用户能解密的密文转换成另一个用户能解密的具有相同明文的密文，而不会泄露数据的明文和授权者的私钥，整个过程不需要解密，不需要代理方之外的任何其他方参与，因此，用户仅需要计算一个重加密密钥，将大部分重加密工作外包给云服务器完成，使上述问题得到解决。然而，现有的代理重加密方案普遍存在以下 2 个明显问题。

1) 仅满足代理重加密方案要求的部分特性。一个理想的代理重加密方案应满足 5 个特性[4-5]：单向性、非交互性、可重复性、可控性和可验证性。然而现有的方案普遍只具有其中的 2 个或者 3 个，降低了在实际应用中的实用性。

2) 无法抵御替换攻击。在密文策略基于属性代理重加密的算法中，代理不仅能够利用用户提供的重加密密钥转换该用户希望重加密的密文，而且也能用此密钥转换其他密文数据，若一个满足新共享访问策略的其他用户与代理勾结，委托方的其他密文数据将被泄露。另一方面，满足新共享访问策略的其他用户采用一定手段截获到委托方的重加密密钥，该用户可以利用其私钥解密出嵌入重加密密钥中的随机因子，再结合委托方的重加密密钥也可以解密出委托方的其他密文数据。

针对以上问题，本文在 Bethencourt 等[1]提出的 CP- ABE 方案的基础上，结合线性整数秘密共享方案（LISS, linear integer secret-sharing scheme）[6]，提出了一种支持多种特性的基于属性代理重加密方案。该方案的特点如下。

1) 满足理想代理重加密方案所要求的所有特性。除了具有现有代理重加密方案普遍有的单向性、非交互性和可重复性外，还通过在加密时选择是否生成重加密密文解密时需要的一个密文子项实现了可控性，通过在数据密文中增加验证项来确保可验证性。

2) 能够防止替换攻击。为每个文件分配一个数据唯一标识符，并将其嵌入重加密密钥和数据密文中，只有当重加密密钥和文件密文的唯一标识符相匹配时，代理才能进行重加密操作。

3) 显著减少了客户端的计算负担。加密时，云服务器分担了近一半共享访问策略对应密文子项的计算工作；而解密时，客户端仅需要 3 次指数运算就能完成一个数据密文的解密工作。

2 相关研究

2007 年，Bethencourt 等[1]首先提出密文策略基于属性加密（CP-ABE）算法，在该算法中，用户私钥与属性集合相关联，数据密文与秘密共享访问结构相关联，只有用户的属性集合满足密文的秘密共享访问策略才能解密出该密文的明文数据，该方案在一般群模型和随机预言模型下可以对抗选择明文攻击（CPA, chosen plaintext attack）[7-8]。2011 年，Waters 等[9]提出一个采用线性秘密共享方案（LSSS, linear secret sharing scheme）实现秘密共享的 CP-ABE，相比 Bethencourt 等[1]的方案，该方案在效率上有所提升。2014 年，Balu 等[10]为了解决 Waters 等[9]提出的 CP-ABE 方案中属性出现次数有限制这一问题，提出用 LISS 代替 LSSS 实现 CP-ABE 方案，并给出构造矩阵的规则。LISS 和 LSSS 具有相同的表达力且同样基于 d-BDH 假设，不同的是，LSSS 是在有限群上实现秘密共享，而 LISS 是在整数区间上实现秘密共享，LISS 比 LSSS 具有更高的效率。2004 年，Canetti 等[11]提出了一种将 CPA 安全转换为选择密文攻击（CCA, chosen ciphertext attack）的方法，该方法的核心是签名。2007 年，Ling 等[12]采用 Canetti 等[11]提出的转换方法，提出了一种具有 CCA 安全的 CP-ABE 方案，但该方案在设计签名公钥的验证时，对公钥的每一个比特生成一个密文子项，这不但增加了密文空间的大小，而且加重了加解密过程中用户客户端的计算负担。2010 年，Zhao 等[13]也同样采用一次签名算法提出一种具有 CCA 安全的基于属性的条件代理重加密方案。2009 年，Liang 等[14]为了解决

CP-ABE 共享访问策略更新问题，首次将代理重加密技术引入 CP-ABE 中，提出密文策略基于属性代理重加密方案，该方案允许一个代理将一种共享访问策略下的密文转换为另一种共享访问策略下具有相同明文的密文，而代理无法获取数据明文，但该方案仅具有单向性、非交互性和可重复性，无法对抗选择密文攻击。为了解决这一问题，2013 年，Liang 等[15]提出一个 CP-ABE 代理重加密方案，该方案能对抗选择明文攻击和选择密文攻击，然而重加密密钥的生成和重加密都需要很大的计算量，且密文空间与重加密次数呈线性关系。为了弥补一般服务器计算能力和存储能力不足的问题，2015 年，Liang 等[16]又提出了利用云服务器来进行代理重加密，采用强不可伪造一次签名（OTS, one-time signature）技术[11]实现 CCA 安全，但该方案过于复杂且不支持密文的多次重加密，与前面方案一样，仅具有单向性、非交互性。2010 年，Luo 等[17]提出的面向 CP-ABE 的代理重加密方案则较好地解决了这一问题，该方案允许用户在重加密密钥中嵌入一个随机的参数，并利用新的共享访问策略对该参数加密，代理重加密后，只有满足新共享访问策略的用户解密出该参数才能解密出数据明文，在密文中增加一个密文子项控制该密文是否能重加密，使只有在加密或重加密时生成了该密文子项的密文才能被重加密。该方案虽然具有单向性、非交互性、可控性和可重复性，但是其共享访问结构仅支持 AND 门，加解密计算量随重加密次数呈线性增长。2015 年，Li 等[18]受 Luan 等[19]的具有 CPA 安全的 CP-ABE 方案的启发，提出了一种具有 CCA 安全的 CP-ABE 代理重加密方案，该方案与 Luo 等[17]的方案一样具有单向性、非交互性、可重复性和可控性，但无法控制用户利用其私钥的多个与属性相关的密钥子项构造出一个并不具有的密钥子项，用户无法控制代理利用重加密密钥重加密该用户的其他密文。同年，Kawai[20]为了解决现有代理重加密方案生成重加密密钥时给用户带来繁重的计算负担，提出由授权中心来完成重加密密钥的生成工作，用户只需少量计算并向授权中心提出生成重加密密钥请求，但该方案不仅会造成授权中心的计算“瓶颈”，还失去了用户对重加密的控制，此外，该方案也仅具有单向性、非交互性。Fu[21]也提出一种具有单向性、非交互性、可重复性的基于属性的代理重加密方案。2016 年，一种具有隐藏访问策略的代理重加密方案被 Zhang 等[22]首次提出，该方案引入一种称为匹配再重加密的新技术，即在重加密之前，先利用代理重加密密钥和数据密文的特殊组件进行匹配计算，以检测该用户是否有权限进行重加密操作。但在 2017 年，Yin 等[23]发现该方案并不能实现访问策略的隐藏，因为存在敌手能利用部分密文组件及系统公钥测试随机选择的属性集合是否在密文属性集合中的问题。为了解决这一问题，Yin 等[23]提出一种改进的具有隐藏访问策略的代理重加密方案，该方案与 Zhang 等[22]提出的方案一样，具有单向性、非交互性、可重复性和可控性，但两者均采用 AND 门共享访问结构，致使其共享访问策略的表达力较低。2017 年，Sepehri 等[24]提出一种能实现数据安全共享的基于属性代理重加密方案，该方案将用户属性集合与密文共享访问策略均采用向量表示，当用户属性集合向量与密文共享访问策略向量内积为 0 时，该用户才能重加密该数据密文。该方案尽管具有单向性、非交互性和可重复性，但过于复杂，且用户私钥和数据密文占用存储空间较大。同年，Ma 等[25]提出一种可验证的外包加密和解密方案，加密和解密的外包运算分别由加密服务器（ESP, encryption service provider）和解密服务器（DSP, decryption service provider）完成。后来，Xiong 等[26]认为该方案并不具有可验证属性，证明该方案的 ESP 可以将伪造的中间密文返回给用户而不被检测到。Feng 等[27]提出一种完全安全的基于属性代理重加密方案，但仅支持 AND 门共享访问结构，缺乏表达力。2018 年，Ge 等[28]提出了一种具有 CCA 安全的密钥策略基于属性代理重加密方案，该方案的代理重加密过程与 Liang 等[16]的方案类似，同样仅具有单向性、非交互性。除上述文献外，文献[29-31]也对代理重加密问题进行了研究，但其方案与上面讨论的方案类似。

从上面的分析不难看出，现有的方案普遍只支持单向性、非交互性，已有较好的代理重加密方案（如文献[17-18,22-23]）虽然支持可重复性和可控性，但是在效率、安全性和访问结构的表达力方面还需要进一步的提高，几种方案的特性对比如表 1 所示。此外，现有的方案都存在代理能够利用用户提交的重加密密钥重加密该用户的其他密文的问题，并且在加密或重加密过程中，用户客户端承担了过多的计算量。

表 1 几种方案的特性对比

方案	访问结构	单向性	非交互性	可重复性	可控性	可验证性	安全性
文献[14]方案	AND 门	√	√	√	×	×	sCPA
文献[15]方案	LSSS	√	√	×	×	×	sCCA
文献[16]方案	LSSS	√	√	×	×	×	CCA
文献[17]方案	AND 门	√	√	√	√	×	sCPA
文献[18]方案	访问树	√	√	√	√	×	sCCA
文献[20]方案	LSSS	√	√	×	×	×	CPA
文献[21]方案	BLSS[32]	√	√	√	×	×	CPA
文献[22]方案	AND 门	√	√	√	√	×	sCPA
文献[23]方案	AND 门	√	√	√	√	×	sCPA
文献[24]方案	AND 门	√	√	√	×	×	sCPA
本文方案	LISS	√	√	√	√	√	sCPA

3 基本知识

3.1 双线性映射

设 G 和 G_T 都是阶为大素数 p 的乘法循环群，g 为 G 的生成元，e 为双线性映射，即 $e:G\times G\to G_T$。双线性映射 e 具有以下性质。

1) 双线性。对于任意的 $u,v\in G$ 和 $a,b\in Z_p^*$，有 $e(u^a,v^b)=e(u^b,v^a)=e(u,v)^{ab}$。

2) 非退化性。即有 $e(g,g)\neq 1$。

3) 可计算性。对于所有的 $u,v\in G$，$e(u,v)$ 都能被有效计算。

3.2 线性整数秘密共享方案

设含单个元素的矩阵为 $\boldsymbol{M}_u\in Z^{1\times 1}$，即 $\boldsymbol{M}_u=[1]$。对于矩阵 $\boldsymbol{M}_a\in Z^{d_a\times e_a}$，采用 $\boldsymbol{c}_a\in Z^d$ 表示 $\boldsymbol{M}_a$ 的第一列，而 $\boldsymbol{R}_a\in Z^{d_a\times(e_a-1)}$ 表示 $\boldsymbol{M}_a$ 除第一列外的其他所有列。利用共享访问策略构造 LISS 矩阵，具有如下 3 个规则[6,10]。

1) 对共享访问策略 P 的每个属性 a_i 表示为 $\boldsymbol{M}_u$。

2) 对于任意 OR 关系 $P=P_a\vee P_b$，令 $\boldsymbol{M}_a\in Z^{d_a\times e_a}$ 和 $\boldsymbol{M}_b\in Z^{d_b\times e_b}$ 分别表示 P_a 和 P_b，则 P 可以表示为 $\boldsymbol{M}_{\mathrm{OR}}\in Z^{(d_a+d_b)\times(e_a+e_b-1)}$，其中 $\boldsymbol{M}_{\mathrm{OR}}$ 的第一列为 $\boldsymbol{c}_a$ 和 $\boldsymbol{c}_b$ 向量级联后的向量，而接着的 (e_a-1) 列为所有 $\boldsymbol{R}_a$ 中的向量级联 d_b 个 0 后的向量，最后 (e_b-1) 列为 d_a 个 0 级联所有 $\boldsymbol{R}_b$ 中的向量后的向量。$\boldsymbol{M}_{\mathrm{OR}}$ 的矩阵形式为

$$\boldsymbol{M}_{\mathrm{OR}}=\begin{bmatrix}\boldsymbol{c}_a & \boldsymbol{R}_a & 0\\ \boldsymbol{c}_b & 0 & \boldsymbol{R}_b\end{bmatrix} \tag{1}$$

3) 对于任意 AND 关系 $P=P_a\wedge P_b$，令 $\boldsymbol{M}_a\in Z^{d_a\times e_a}$ 和 $\boldsymbol{M}_b\in Z^{d_b\times e_b}$ 分别表示 P_a 和 P_b，则 P 可以表示为 $\boldsymbol{M}_{\mathrm{AND}}\in Z^{(d_a+d_b)\times(e_a+e_b)}$，其中 $\boldsymbol{M}_{\mathrm{AND}}$ 的第一列为 $\boldsymbol{c}_a$ 级联 d_b 个 0 后的向量，第二列为 $\boldsymbol{c}_a$ 和 $\boldsymbol{c}_b$ 向量级联后的向量，而接着的 (e_a-1) 列为所有 $\boldsymbol{R}_a$ 中的向量级联 d_b 个 0 后的向量，最后 (e_b-1) 列为 d_a 个 0 级联所有 $\boldsymbol{R}_b$ 中的向量后的向量。$\boldsymbol{M}_{\mathrm{AND}}$ 的矩阵形式为

$$\boldsymbol{M}_{\mathrm{AND}}=\begin{bmatrix}\boldsymbol{c}_a & \boldsymbol{c}_a & \boldsymbol{R}_a & 0\\ 0 & \boldsymbol{c}_b & 0 & \boldsymbol{R}_b\end{bmatrix} \tag{2}$$

在 LISS 访问结构 $(\boldsymbol{M},\rho)$ 中，$\boldsymbol{M}$ 是访问矩阵，ρ 是矩阵每一行 $\boldsymbol{M}_i$ 到属性 $\rho(i)$ 的映射关系。

4 代理重加密算法与安全模型

4.1 算法定义

定义 1 支持多种特性的代理重加密方案（CP-ABPRE）由以下 10 个算法构成，其关系如图 1 所示。

1) $\mathrm{Setup}(U,A,k)\to(\mathrm{PK},\mathrm{MK})$：初始化算法由授权中心执行，输入属性空间 U、系统用户共有的虚拟属性 A 和安全参数 k，输出系统公钥 PK 和系统主密钥 MK。

2) $\mathrm{KeyGen}(\mathrm{PK},\mathrm{MK},S)\to(\mathrm{SK})$：私钥生成算法由授权中心执行，输入系统公钥 PK、系统主密钥 MK 和用户属性集合 $S\subseteq U$，输出与属性集合 S 相关联的用户私钥 SK。

3) $\mathrm{Encrypt}(m,(\boldsymbol{M},\rho),\mathrm{PK})\to(\mathrm{CT}')$：加密算法由用户执行，输入待加密数据 m、线性整数秘密共享 LISS 访问结构 $(\boldsymbol{M},\rho)$ 和系统公钥 PK，输出数据 m 与 LISS 相关联的部分密文 CT′。

4) $\mathrm{OutEncrypt}(\mathrm{CT}',\mathrm{PK})\to(\mathrm{CT})$：外包加密算法由云服务器执行，输入用户计算的部分密文 CT′ 和系统公钥 PK，输出数据 m 的完整密文 CT。

5) $\mathrm{ReKeyGen}(\mathrm{PK},\mathrm{SK},(\boldsymbol{M}',\rho'))\to(\mathrm{RK})$：重加密密钥生成算法由用户执行，输入系统公钥 PK、用户私钥 SK 和新的线性整数秘密共享 LISS 访问结构 $(\boldsymbol{M}',\rho')$，输出一个重加密密钥 RK。若重加密密钥 RK 在用户私钥 SK 对应用户属性集合 S 满足密文的访问结构 $(\boldsymbol{M},\rho)$ 时，代理可以利用该重加密密钥 RK 将该密文的访问结构 $(\boldsymbol{M},\rho)$ 转换为新的访问结构 $(\boldsymbol{M}',\rho')$。

6) $\mathrm{ReEncrypt}(\mathrm{PK},\mathrm{CT},\mathrm{RK})\to(\mathrm{CT}^*)$：重加密算法由云服务器执行，输入系统公钥 PK、与访问结构 $(\boldsymbol{M},\rho)$ 相关联的密文 CT 和与访问结构 $(\boldsymbol{M}',\rho')$ 相

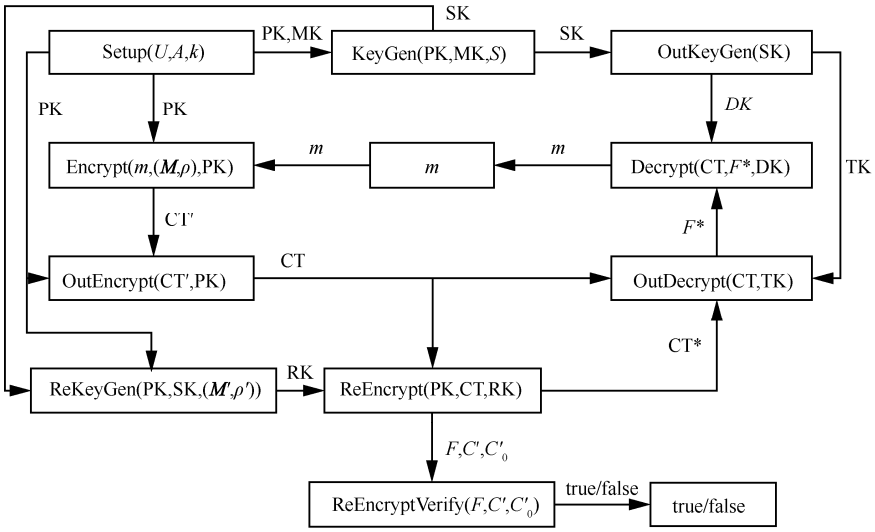


图 1　10 个算法的关系

关联的重加密密钥 RK，当密文 CT 被设置为不能重加密或 $S \not\models (\boldsymbol{M},\rho)$ 时，输出 $\perp$，否则输出与访问结构 $(\boldsymbol{M}',\rho')$ 相关联的密文 CT^*。

7) $\mathrm{ReEncryptVerify}(F,C',C_0') \to (\mathrm{true})$：重加密验证算法由用户执行，输入 CT 密文组件 C'、C_0' 和云服务器代理重加密计算结果 F，若验证通过，输出 true，否则直接输出 $\perp$。

8) $\mathrm{OutKeyGen(SK)} \to (\mathrm{TK,DK})$：转换密钥生成算法由用户执行，输入用户私钥 SK，输出外包云服务器部分解密的转换密钥 TK 和最后用户完全解密时使用的密钥 DK。

9) $\mathrm{OutDecrypt(CT,TK)} \to (F^*)$：外包解密算法由云服务器执行，输入密文 CT 和转换密钥 TK，当 $S \models (\boldsymbol{M},\rho)$ 时，输出密文 CT 的部分解密密文 F^*，否则直接输出 $\perp$。

10) $\mathrm{Decrypt(CT},F^*,\mathrm{DK}) \to (m)$：解密算法由用户执行，输入密文 CT、云服务器代理计算的部分解密密文 F^* 和转换密钥中用户保留的密钥 DK，输出明文数据 m。

4.2　代理重加密方案特性

代理重加密方案首先要具有正确性，正确性的定义如下。

对于任意安全参数 $k \in N$、任意属性集合 $S(S \subseteq U \cup A)$、由 U 中属性构建的任意访问结构 $(\boldsymbol{M},\rho)$ 和任意数据 $m \in \{0,1\}^k$，若 $\mathrm{Setup}(U,A,k) \to (\mathrm{PK,MK})$、$\mathrm{KeyGen(PK,MK,}S) \to (\mathrm{SK})$、$\mathrm{OutKeyGen(SK)} \to (\mathrm{TK,DK})$，对在系统中使用的属性集合 S，当 $S \models (\boldsymbol{M},\rho)$ 和 $S \models (\boldsymbol{M}',\rho')$ 时，有

$$\begin{aligned}
&\mathrm{OutEncrypt(Encrypt}(m,(\boldsymbol{M},\rho),\mathrm{PK}),\mathrm{PK}) \to \mathrm{CT},\\
&\mathrm{Decrypt(CT,OutDecrypt(CT,TK)},\mathrm{DK}) \to m,\\
&\mathrm{ReEncrypt(CT,ReKeyGen(SK,}(\boldsymbol{M}',\rho'))) \to \mathrm{CT}^*,\\
&\mathrm{Decrypt}(\mathrm{CT}^*,\mathrm{OutDecrypt}(\mathrm{CT}^*,\mathrm{TK}),\mathrm{DK}) \to m
\end{aligned} \tag{3}$$

除正确性外，代理重加密方案通常具有以下特性。

1) 单向性：代理可以利用用户提交的重加密密钥将一种共享策略下的密文 CT 转换为另一种共享策略下具有相同明文的密文 CT^*，但不允许利用该重加密密钥将密文 CT^* 转换为密文 CT。

2) 非交互性：加密方在构造重加密密钥过程中，不需要其他信任的第三方和授权中心参与。

3) 可重复性：代理可以对同一密文进行多次重加密。

4) 可控性：由用户在加密或重加密时决定密文是否可以重新加密。

5) 可验证性：加密方可以对代理进行重加密后的计算结果进行正确性验证。

4.3　安全模型

接下来，定义 CP-ABPRE 方案的针对性（selective）CPA 安全模型。

定义 2　如果没有一个概率多项式时间（PPT, probabilistic polynomial-time）敌手 Adv 能够以不可忽略的优势赢得下面的游戏，则代理重加密方案达到 sCPA 安全。在游戏中，C 是挑战者，k、U 和 A 分别是安全参数、属性空间和虚拟属性。

预备阶段　Adv 选择挑战的访问结构 $(\boldsymbol{M}'',\rho'')$。

初始化 C 运行 $(\mathrm{PK},\mathrm{MK})\leftarrow\mathrm{Setup}(U,A,k)$ 获得系统公钥 PK 和系统主密钥 MK，并将 PK 发送给 Adv。

阶段 1 敌手可以重复执行以下任何查询。

1) 私钥查询 $O_{\mathrm{sk}}(S)$：Adv 提交一个属性集合 S，该属性集合 S 不满足挑战访问结构 $(\boldsymbol{M}'',\rho'')$，C 返回用户私钥 $\mathrm{SK}\leftarrow\mathrm{KeyGen}(\mathrm{PK},\mathrm{MK},S)$ 给 Adv。

2) 重加密密钥查询 $O_{\mathrm{rk}}(S,(\boldsymbol{M}',\rho'))$：Adv 提交一个属性集合 S（该属性集合 S 不满足挑战访问结构 $(\boldsymbol{M}'',\rho'')$）和一个访问结构 $(\boldsymbol{M}',\rho')$，C 返回重加密密钥 $\mathrm{RK}\leftarrow\mathrm{ReKeyGen}(\mathrm{PK},\mathrm{SK},(\boldsymbol{M}',\rho'))$ 给 Adv，其中用户私钥 $\mathrm{SK}\leftarrow\mathrm{KeyGen}(\mathrm{PK},\mathrm{MK},S)$。

挑战阶段 Adv 向 C 提交 2 个等长的明文 m_0 和 m_1，C 随机选择 $b\in\{0,1\}$，返回 $\mathrm{CT}''\leftarrow\mathrm{OutEncrypt}(\mathrm{CT}',\mathrm{PK})$ 给 Adv，其中 $\mathrm{CT}'\leftarrow\mathrm{Encrypt}(m_b,(\boldsymbol{M}'',\rho''),\mathrm{PK})$。

阶段 2 Adv 继续阶段 1 的查询。

猜测 Adv 输出一个猜测值 $b'\in\{0,1\}$，如果 $b=b'$，Adv 赢得游戏。Adv 赢得游戏的优势被定义为 $\Pr[b'=b]-\frac{1}{2}$。

5 方案构造

方案包括初始化、私钥生成、加密、外包加密、重加密密钥生成、重加密、重加密验证、转换密钥生成、外包解密和解密共 10 个算法。

1) 初始化：$\mathrm{Setup}(U,A,k)$

k 为系统安全参数。授权中心选择阶为大素数 p 的双线性群 G 和 G_T，记 $g\in G$ 为 G 的生成元，双线性映射 $e:G\times G\rightarrow G_T$。设属性空间为 U，其中的虚拟属性 A 为所有用户共有。选择 $g_2\in G$，$a,b,\alpha,\beta\in Z_p^*$，对虚拟属性 A 和任意 $i\in U$，选择 T_A、$T_i\in G$，计算 $Y=e(g,g)^{\alpha}$，$h=g^{\beta}$，$h_0=g^{a}$，$B=g^{b}$。定义散列函数 $H_1:\{0,1\}^*\rightarrow Z_p^*$，$H_2:\{0,1\}^*\rightarrow\{0,1\}^k$，定义编码变换 $E:G\rightarrow\{0,1\}^k$。授权中心将系统公钥 PK 向云服务器和所有用户公开，系统主密钥 MK 由授权中心秘密保存。系统公钥 PK 为

$$\begin{aligned}\mathrm{PK}=<\ & p,g,g_2,G,G_T,e,Y,T_A,\\ & \forall i\in U:T_i,h,h_0,B,H_1,H_2,E>\end{aligned}\tag{4}$$

系统主密钥 $\mathrm{MK}=<a,\beta,g^{\alpha}>$。

2) 生成用户私钥：$\mathrm{KeyGen}(\mathrm{PK},\mathrm{MK},S)$

生成私钥算法随机选择 $r\in Z_p^*$，为每个属性 $i\in S$ 随机选择 $r_i\in Z_p^*$，计算用户私钥 SK 为

$$\begin{aligned}SK=<\ & S,D=g^{\frac{\alpha+r}{\beta}},D_B=B^{\frac{r}{a}},D_A=T_A^{\frac{r}{a}},\\ & D_A'=g^{\frac{r}{a}},\forall i\in S:D_i=g^{r}T_i^{r_i},D_i'=g^{r_i}>\end{aligned}\tag{5}$$

其中，D_A和D_A'为虚拟属性密钥子项。

3) 加密：$\mathrm{Encrypt}(m,(\boldsymbol{M},\rho),\mathrm{PK})$

加密算法输入数据 $m\in\{0,1\}^k$，LISS 访问结构 $(\boldsymbol{M},\rho)$，其中 $\boldsymbol{M}$ 是一个 $l\times q$ 的矩阵，ρ 是矩阵每一行 $\boldsymbol{M}_i$ 到属性 $\rho(i)$ 的映射关系。

随机选择 $z\in Z_p^*$，计算 $Z=g^{z}$，$\eta=H_1(\mathrm{FILE})$，其中，FILE 为数据唯一标识符。

选择一个随机向量 $\boldsymbol{v}=(s',v_2,\cdots,v_q)^{\mathrm{T}}$，其中 s'，$v_2,\cdots,v_q\in Z_p^*$，计算 $\lambda_i=\boldsymbol{M}_i\boldsymbol{v}$。随机选择 $R\in G_T$，计算 $s=H_1(R,m)$，$r=H_2(R)$。令 $s''=s-s'$，计算 $C=RY^{s}=Re(g,g)^{\alpha s}$，$C_1=m\oplus r$，$C_0=h^{s}$，$C_0'=g_2^{s'}$，$\forall i\in[1,\ l](C_i'=T_{\rho(i)}^{\lambda_i}$，$C_i''=\frac{\lambda_i}{z})$，$C'=g^{H_1(e(g,g)^{\alpha s})}$。为虚拟属性 A 选择一个随机数 $r_A\in Z_p^*$，计算 $C_A'=h_0^{s''}B^{r_A}T_A^{\eta r_A}$，$C_A''=\frac{r_A}{z}$。将部分密文 CT′ 以及 Z、$\{C_i''\}_{i\in[1,l]}$ 和 C_A'' 上传至云服务器，部分密文 CT′ 为

$$CT'=<(\boldsymbol{M},\rho),C,C_1,C_0,C_0',C',\forall i\in[1,l](C_i'),C_A'>\tag{6}$$

若将该密文设置为不能被重加密，只需去掉密文子项 C_0' 即可。

4) 外包加密：$\mathrm{OutEncrypt}(\mathrm{CT}',\mathrm{PK})$

云服务器接收到部分密文 CT′ 以及 Z、$\{C_i''\}_{i\in[1,l]}$ 和 C_A'' 后，计算 $\forall i\in[1,l](C_i=Z^{C_i''}=g^{\lambda_i}),C_A=Z^{C_A''}=g^{r_A}$。数据 m 的完整密文 CT 为

$$\begin{aligned}\mathrm{CT}=<\ & (\boldsymbol{M},\rho),C,C_1,C_0,C_0',C',\\ & \forall i\in[1,l](C_i,C_i'),C_A,C_A'>\end{aligned}\tag{7}$$

5) 生成重加密密钥：$\mathrm{ReKeyGen}(\mathrm{PK},\mathrm{SK},(\boldsymbol{M}',\rho'))$

生成重加密密钥算法使用用户私钥 SK 和新的 LISS 访问结构 $(\boldsymbol{M}',\rho')$ 生成重加密密钥 RK，其中 $\boldsymbol{M}'$ 是一个 $l'\times q'$ 的矩阵，ρ' 是矩阵每一行 $\boldsymbol{M}_i'$ 到属性 $\rho'(i)$ 的映射关系。

选择随机因子 $d\in Z_p^*$，计算 g^{d}，编码得 $E(g^{d})$，$\eta=H_1(\mathrm{FILE})$。计算重加密密钥 RK 为

$$\begin{aligned}&\mathrm{RK}=<S,(\boldsymbol{M}',\rho'),\mathrm{rk}_D=D=g^{\frac{\alpha+r}{\beta}},\\ &\mathrm{rk}_{D_A}=D_BD_A^{\eta}=B^{\frac{r}{a}}T_A^{\frac{\eta r}{a}},\end{aligned}$$

$$\mathrm{rk}_{D'_A}=D'_A=g^{\frac{r}{a}},$$

$$\forall i\in S:\mathrm{rk}_{D_i}=D_i g_2^d=g^r T_i^{r_i} g_2^d, rk_{D'_i}=D'_i=g^{r_i},$$

$$C'_{\mathrm{rk}}=\mathrm{Encrypt}(E(g^d),(\boldsymbol{M}',\rho'),\mathrm{PK})> \quad (8)$$

6) 重加密： ReEncrypt(PK,CT,RK)

若密文被设置为不能重加密或 $S\neq(\boldsymbol{M},\rho)$，则直接输出 $\perp$；否则计算 $C_{\mathrm{rk}}=\mathrm{OutEncrypt}(C'_{\mathrm{rk}},\mathrm{PK})$，选择常数 w_i 使 $\sum_{\rho(i)\in S} w_i M_i=(1,0,\cdots,0)$，对虚拟属性计算

$$F'=\frac{e(C'_A,\mathrm{rk}_{D'_A})}{e(C_A,\mathrm{rk}_{D_A})}=\frac{e(h_0^{s''}B^{r_A}T_A^{\eta r_A},g^{\frac{r}{a}})}{e(g^{r_A},B^{\frac{r}{a}}T_A^{\eta\frac{r}{a}})}=e(g^{s''},g^r) \quad (9)$$

对密文共享访问策略计算

$$F''=\prod_{i\in S}\left(\frac{e(\mathrm{rk}_{D_i},C_i)}{e(\mathrm{rk}_{D'_i},C'_i)}\right)^{w_i}=\prod_{i\in S}\left(\frac{e(g^r T_i^{r_i}g_2^d,g^{\lambda_i})}{e(g^{r_i},T_i^{\lambda_i})}\right)w_i=$$
$$e(g^r,g^{s'})e(g_2^d,g^{s'}) \quad (10)$$

最后计算 F 为

$$F=\frac{e(C_0,\mathrm{rk}_D)}{F'F''}=\frac{e(h^s,g^{\frac{\alpha+r}{\beta}})}{e(g^{s''},g^r)e(g^r,g^{s'})e(g_2^d,g^{s'})}=\frac{e(g^s,g^\alpha)}{e(g_2^d,g^{s'})} \quad (11)$$

完整的重加密密文 CT^* 为

$$\mathrm{CT}^*=<(\boldsymbol{M}',\rho'),C,C_1,C'_0,C',F,C_{\mathrm{rk}}> \quad (12)$$

7) 重加密验证： ReEncryptVerify(F,C',C'_0)

用户使用重加密后的 F、C'、C'_0 对云服务器重加密计算结果进行正确性验证，计算

$$V=Fe(C'_0,g^d)=e(g^s,g^\alpha) \quad (13)$$

若 $C'==g^{H_1(V)}$，则说明云服务器的代理重加密计算结果正确，输出 true，否则直接输出 $\perp$。

8) 生成转换密钥： OutKeyGen(SK)

选择一个随机数 $\delta\in Z_p^*$，计算 $\eta=H_1(\mathrm{FILE})$，计算用于授权云服务器部分解密的转换密钥 TK 为

$$\mathrm{TK}=<S,\mathrm{tk}_D=D^\delta=g^{\frac{\delta(\alpha+r)}{\beta}},$$

$$\mathrm{tk}_{D_A}=(D_B D_A^\eta)^\delta=B^{\frac{\delta r}{a}}T_A^{\frac{\delta\eta r}{a}},\mathrm{tk}_{D'_A}=D'^{\delta}_A=g^{\frac{\delta r}{a}},$$

$$\forall i\in S:\mathrm{tk}_{D_i}=D_i^\delta=g^{\delta r}T_i^{\delta r_i},\mathrm{tk}_{D'_i}=(D'_i)^\delta=g^{\delta r_i}> \quad (14)$$

由用户保存的密钥为 $\mathrm{DK}=<\delta>$。

9) 外包解密： OutDecrypt(CT,TK)

若用户属性集合 $S\neq(\boldsymbol{M},\rho)$，则直接输出 $\perp$；否则选择常数 w_i 使 $\sum_{\rho(i)\in S} w_i M_i=(1,0,\cdots,0)$，对虚拟属性计算

$$F'=\frac{e(C'_A,\mathrm{tk}_{D'_A})}{e(C_A,\mathrm{tk}_{D_A})}=\frac{e(h_0^{s''}B^{r_A}T_A^{\eta r_A},g^{\frac{\delta r}{a}})}{e(g^{r_A},B^{\frac{\delta r}{a}}T_A^{\frac{\delta\eta r}{a}})}=e(g^{s''},g^{\delta r}) \quad (15)$$

对密文共享访问策略计算

$$F''=\prod_{i\in S}\left(\frac{e(\mathrm{tk}_{D_i},C_i)}{e(\mathrm{tk}_{D'_i},C'_i)}\right)^{w_i}=$$
$$\prod_{i\in S}\left(\frac{e(g^{\delta r}T_i^{\delta r_i},g^{\lambda_i})}{e(g^{\delta r_i},T_i^{\lambda_i})}\right)^{w_i}=e(g^{\delta r},g^{s'}) \quad (16)$$

最后计算 F^* 为

$$F^*=\frac{e(C_0,\mathrm{tk}_D)}{F'F''}=\frac{e(h^s,g^{\frac{\delta(\alpha+r)}{\beta}})}{e(g^{s'},g^{\delta r})e(g^{\delta r},g^{s'})}=e(g^s,g^\alpha)^\delta \quad (17)$$

10) 解密： Decrypt(CT,F^*,DK)

对于未被重加密的密文，数据所有者或共享用户将密文子项 C、C_1 及云服务器部分解密 F^* 从云服务器下载到本地。解密 R 的算法为

$$\mathrm{Decrypt(CT,DK)}=\frac{C}{(F^*)^{\delta^{-1}}}=\frac{Re(g,g)^{\alpha s}}{e(g^s,g^\alpha)^{\delta\delta^{-1}}}=R \quad (18)$$

计算 $m=C_1\oplus H_2(R)$，$s=H_1(R,m)$，若 $C==R\cdot e(g,g)^{\alpha s}$ 且 $F^*==e(g,g)^{\alpha s\delta}$，输出 m，否则输出 $\perp$。

对于被重加密的密文，按照上述算法对随机因子的密文 C_{rk} 解密得到 $E(g^d)$，解码得 g^d。将重加密前的密文子项 C、C_1、C'_0 及重加密时计算的 F 从云服务器下载到本地，解密 R 的算法为

$$\mathrm{Decrypt(CT,DK)}=\frac{C}{Fe(C'_0,g^d)}=$$
$$\frac{Re(g,g)^{\alpha s}}{\frac{e(g^s,g^\alpha)}{(g_2^d,g^{s'})}e(g_2^{s'},g^d)}=R \quad (19)$$

计算 $m=C_1\oplus H_2(R)$，$s=H_1(R,m)$，若 $C==R\cdot e(g,g)^{\alpha s}$ 且 $F==e(g,g)^{\alpha s}e(C'_0,g^d)^{-1}$，输出 m，否则输出 $\perp$。

对于被多次重加密的密文，采用第一种解密算法解密出最后一次重加密的数据，再重复使用第二

种解密算法，最后得出明文数据 m。

6 特性与安全性分析

6.1 特性分析

1) 单向性

重加密密钥由用户私钥嵌入随机因子后的数据及随机因子与新访问结构（$\boldsymbol{M}', \rho'$）相关联的密文数据构成，在重加密密钥满足访问结构（$\boldsymbol{M}, \rho$）的情况下，代理利用该重加密密钥将密文的访问结构由（$\boldsymbol{M}, \rho$）转换为（$\boldsymbol{M}', \rho'$）时能计算出一个包含随机因子的半解密密文（此时若知道随机因子，就可以解密出明文数据），但缺少随机因子与访问结构（$\boldsymbol{M}, \rho$）相关联的密文而无法将密文的访问结构由（$\boldsymbol{M}', \rho'$）转换为（$\boldsymbol{M}, \rho$）。

2) 非交互性

重加密密钥计算过程由两部分构成：在用户私钥中嵌入随机因子和生成随机因子与新访问结构相关联的密文数据。然而这两部分都可以由用户客户端独立完成，不需要其他信任的第三方和授权中心参与。

3) 可重复性

重加密密钥由用户私钥嵌入了随机因子后的数据及随机因子与新访问结构相关联的密文数据构成，代理利用该重加密密钥重加密时，将原始密文转换为含有随机因子的半解密密文（此时若知道随机因子，就可以解密出明文数据），重加密后的密文由原始密文（除了 C_0）、半解密密文和随机因子的密文构成。再次重加密时，对随机因子的密文数据进行相同的操作即可达到多次重加密的目的。

4) 可控性

重加密密文在解密时，需要原始密文的一个密文子项 C_0' 的参与才能解密，但原始密文解密时并不需要该密文子项，故该密文子项控制了加密或重加密后的密文是否可以再次重加密。

5) 可验证性

①如果数据所有者想要验证云服务器代理重加密结果是否正确，只需将密文子项 C'、C_0' 及云服务器代理重加密结果 F 传回用户客户端，此时用户客户端已知 g^d，故用户客户端可以计算

$$V = Fe(C_0', g^d) = e(g^s, g^\alpha) \tag{20}$$

再计算 $g^{H_1(V)}$ 并与 C' 进行对比，相等即通过验证。

② 如果数据所有者想要验证云服务器是否按要求进行外包加密，在将待计算密文项上传的同时计算挑战的密文项。在收到返回的密文子项后，将挑战密文项与返回的对应密文项进行对比，相等即通过验证。

③ 如果数据所有者想要验证云服务器是否按要求进行外包解密[33]，待用户客户端解密出数据 m 后，计算 $s = H_1(R, m)$，再计算 $Re(g,g)^{\alpha s}$ 和 $e(g,g)^{\alpha s\delta}$ 并分别与密文子项 C 和云服务器部分解密 F^* 进行对比，相等即通过验证。

6.2 抗替换攻击

为防止云服务器利用数据所有者提交的重加密密钥重加密数据所有者其他密文数据，从而导致满足新共享策略的用户可以非法解密数据所有者其他密文数据，在外包数据时为每个数据分配一个数据唯一标识符 η，并将其嵌入重加密密钥和数据密文中。云服务器代理重加密时，只有嵌入重加密密钥和数据密文的数据唯一标识符匹配时，才能进行重加密操作。

为避免云服务器能够将重加密密钥子项和密文子项中的数据唯一标识符 η 直接替换为其他的数据唯一标识符，该方案采用了随机化方法，即在重加密密钥子项 rk_{D_A} 中，用 $\frac{r}{a}$ 对 η 进行随机化，再使用 $B^{\frac{r}{a}}$ 对 $T_A^{\frac{\eta r}{a}}$ 随机化；在密文子项 C_A' 中，用 r_A 对 η 进行随机化，再使用 $h_0^{s''}B^{r_A}$ 对 $T_A^{\eta r_A}$ 随机化。

6.3 机密性

定理 1 本文所提出的方案，可抵御针对性选择明文攻击。

证明 假设敌手 Adv_1 在一般群模型和随机预言模型能以不可忽略优势攻破本文所提出的方案，那么可以基于 Adv_1 构建敌手 Adv_2，使其可以在同样模型下攻破 BSW 方案，这与在一般群模型和随机预言模型下 BSW 方案可以抵御选择明文攻击矛盾，故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。接下来，说明敌手 Adv_2 的构建过程。

预备阶段 模拟器 Adv_2 启动敌手 Adv_1，Adv_1 选择一个挑战访问结构 $(\boldsymbol{M}'', \rho'')$，$\mathrm{Adv}_2$ 将其对应的访问结构树 T 传递给 BSW 方案的挑战者。

初始化 敌手 Adv_2 获取 BSW 方案的公钥 $\mathrm{PK} = (G_0, g, h = g^\beta, e(g,g)^\alpha)$ 并将其发送给 Adv_1。

阶段 1　Adv_2 初始化空表 T、T_1、T_2，Adv_1 能完成如下查询。

1) $H_1(R,m)$：若 (R,m,s) 已经在 T_1 中，返回 s；否则选择一个随机值 $s\in Z_p^*$，将 (R,m,s) 记录在 T_1 中并返回 s。

2) 随机预言机散列函数 $H_2(R)$：若 (R,r) 已经在 T_2 中，返回 r；否则选择一个随机值 $r\in\{0,1\}^k$，将 (R,r) 记录在 T_2 中并返回 r。

3) 私钥查询 $O_{\mathrm{sk}}(S)$：敌手 Adv_1 可重复发出查询请求。Adv_1 发出一次查询后，Adv_2 对查询进行如下处理。

① 属性集合 $S\models(\boldsymbol{M}'',\rho'')$。将 S 发送给 BSW 方案挑战者，由 BSW 方案挑战者利用密钥生成算法生成与 S 对应的私钥 SK' 并返回给 Adv_2。Adv_2 选择一个随机数 $n\in Z_p^*$，由 SK' 计算出转换密钥 TK，计算 $\mathrm{SK}=(n,\mathrm{TK})$。将 SK 返回给 Adv_1，将 $(S,\mathrm{SK},\mathrm{TK})$ 存入到表 T 中。

② 属性集合 $S\models(\boldsymbol{M}'',\rho'')$。无法查询 S 对应私钥，故只能按如下方法生成伪转换私钥。Adv_2 随机选择 $d,z\in Z_p^*,t\in G_0$，运行 $\mathrm{KeyGen}(\mathrm{PK},(d,z,t),S)$ 生成密钥 SK'，令 $\mathrm{TK}=\mathrm{SK}',\mathrm{SK}=(d,\mathrm{TK})$。将 TK 返回给 Adv_1，将 $(S,\mathrm{SK},\mathrm{TK})$ 存入表 T 中。

4) 重加密密钥查询 $O_{rk}(S,(\boldsymbol{M}',\rho'))$：$\mathrm{Adv}_1$ 提交一个属性集合 S 和一个访问结构 $(\boldsymbol{M}',\rho')$。若 $S\models(\boldsymbol{M}'',\rho'')$，$\mathrm{Adv}_2$ 返回 $\mathrm{RK}\leftarrow\mathrm{ReKeyGen}(\mathrm{PK},\mathrm{SK},(\boldsymbol{M}',\rho'))$ 给 Adv_1，其中 $\mathrm{SK}\leftarrow\mathrm{KeyGen}(\mathrm{PK},\mathrm{MK},S)$。

挑战阶段　Adv_1 向 Adv_2 提交 2 个等长的数据明文 m_0 和 m_1，Adv_2 执行如下操作。

1) 随机选择 2 个消息 $(R_0,R_1)\in G_T^2$ 并将其发送给 BSW 方案挑战者。BSW 方案挑战者随机选择一个消息加密并返回密文 $(C,C_0,(C_i,C_0')_{i\in[1,l]})$。

2) 随机选择 $C_1\in\{0,1\}^k,C_0',C'\in G$，计算 $\{C_A,C_A'\}$。

3)将挑战密文 $(C,C_0,C_1,C_0',C',\{C_i,C_i'\}_{i\in[1,l]},C_A,C_A')$ 发送给 Adv_1。

阶段 2　Adv_1 继续阶段 1 的查询。

猜测　Adv_1 输出猜测值 $b'\in\{0,1\}$，Adv_2 忽视 Adv_1 的猜测值。Adv_2 检索 T_1 和 T_2 以确定 R_0 或 R_1 是否出现在表中，换言之，确定 Adv_1 是否发起过 $H_1(R,\cdot)$ 或 $H_2(R)$ 的查询。如果这 2 个数都出现或都没有出现在这 2 个表中，Adv_2 随机输出猜测值 b'；如果只出现 $R_{b'}$，Adv_2 输出 b'。

7　性能分析

7.1　理论分析

下面从计算和存储 2 个方面讨论本文所提方案的性能。

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（E_G、E_T 分别表示 G 群、G_T 群的指数运算），因此用这 2 个指标来衡量性能。为了方便比较，假设每次加密或重加密后的密文能再次被重加密。

用户客户端加密数据时，密文子项 Z、C、C_0、C_0'、C'、C_A' 共需要执行 8 次指数运算，ρ 中每个属性 $\rho(i)$ 需要执行一次指数运算，计算代价为 $(7+N_P)E_G+E_T$（N_P 为共享访问策略中的属性个数）。云服务器为共享访问策略中每个属性 $\rho(i)$、虚拟属性 C_A 执行一次指数运算的计算代价为 $(1+N_P)E_G$。

生成重加密密钥时，用户客户端计算 g^d、g_2^d、rk_{D_A} 的计算代价为 $3E_G$，此外还需执行一次加密过程，用户客户端的计算代价为 $(10+N_P)E_G+E_T$。

云服务器重加密时，计算虚拟属性的代价为 $2B$。在共享访问策略对应逻辑关系都为“AND”的情况下，计算共享访问策略的每个属性 $\rho(i)$ 都要参与一次指数运算和 2 次双线性运算，计算共享访问策略的代价为 $N_PE_T+2N_PB$。计算 F 需要执行一次双线性运算，云服务器为新的共享访问策略中每个属性 $\rho(i)$、虚拟属性的 C_A 执行一次指数运算的计算代价为 $(1+N_P)E_G$，云服务器重加密时的计算代价为 $(1+N_P)E_G+N_PE_T+(3+2N_P)B$。用户客户端验证云服务器是否正确执行重加密时的计算代价为 E_G+B，故在整个重加密过程中，用户客户端的计算代价为 $(11+N_P)E_G+E_T+B$。

原始密文解密时，用户客户端需要对 F^* 执行一次指数运算，验证需要两次指数运算，计算代价为 $3E_T$。重加密密文解密时，用户客户端还需要执行一次双线性计算，用户客户端解密重加密密文的计算代价为 $4E_T+B$。云服务器代理解密过程与云服务器重加密过程类似（除重加密过程中计算 $E(g^d)$ 部分密文子项），计算代价为 $N_PE_T+(3+2N_P)B$。

几种方案的计算开销对比如表 2 所示。其中，N_P 表示共享访问策略中属性数量，N_S 表示用户私钥属性数量，N 表示系统属性空间属性数量，$|\mathrm{SYM}|$

表 2 **几种方案的用户客户端计算开销对比**

方案	加密	生成重加密密钥	原始密文解密	重加密密文解密
文献[1]方案	$(1+4N_P)E_G+E_T$	—	$(2N_P-2)E_T+(1+2N_P)B$	—
文献[14]方案	$(2+N)E_G+E_T$	$(3+3N)E_G+E_T$	$(1+N)B$	$E_G+(2+N)B$
文献[15]方案	$(3+4N_P)E_G+2E_T$	$(6+4N_P+N_S)E_G+2E_T$	$(2+3N_P)E_G+(1+N_P)E_T+(7+3N_P)B$	$4E_G+(3+N_P)E_T+(3+2N_P)B$
文献[16]方案	$(5+3N_P)E_G+E_T+\|\text{OTS}\|$	$(14+3N_P+2N_S)E_G+E_T+\|\text{OTS}\|$	$(1+2N_P)E_G+N_PE_T+(10+3N_P)B+\|\text{OTS}\|$	$(2+6N_P)E_G+1E_T+(16+6N_P)B+2\|\text{OTS}\|+\|\text{SYM}\|$
文献[17]方案	$(2+N)E_G+E_T$	$(4+N)E_G+E_T$	$(1+2N)B$	$E_T+(2+2N)B$
文献[18]方案	$(4+N_P)E_G+E_T$	$(8+2N_P+N_S)E_G+3E_T$	$E_G+(5+N_P)B$	$2E_G+(8+N_P)B$
文献[21]方案	$(1+N_P)E_G+E_T$	$(4+N_P)E_G+E_T$	$(1+N_P)B$	$E_G+(2+N_P)B$
文献[22]方案	$(3+3N)E_G+2E_T$	$(6+3N)E_G+2E_T$	$(3+3N)B$	$E_T+(4+3N)B$
文献[23]方案	$(6+4N)E_G+2E_T$	$(8+4N)E_G+2E_T$	$(6+3N)B$	$E_T+(7+3N)B$
文献[24]方案	$(2+12N)E_G+E_T$	$(2+25N)E_G+E_T$	$(2+4N)B$	$(3+4N)B$
本文方案	$(7+N_P)E_G+E_T$	$(10+N_P)E_G+E_T$	$3E_T$	$4E_T+B$

表示执行一次对称加密算法的计算开销，|OTS|表示执行一次签名算法的计算开销。本文方案的加密和生成重加密密钥的计算开销与文献[21]方案接近，而 2 种解密方式的计算开销在对比的几种方案中是最低的且为固定值。

2) 存储性能

在本文方案中，用户私钥由用户秘密保存，而用户数据则外包给云服务器存储。用户私钥密钥子项 D 和 D_B，共 2 个 G 群元素，另外每个属性包括 2 个 G 群元素（包括虚拟属性），故用户私钥存储代价为$(4+2N_S)|G|$。用户数据密文子项 C、C_0、C_0' 和 C'，共一个 G_T 群元素和 3 个 G 群元素存储空间，虚拟属性共 2 个 G 群元素，共享策略中每个属性包括 2 个 G 群元素，数据密文存储代价为$(5+2N_P)|G|+|G_T|$。对于重加密密文，对随机因子生成一个与原始密文相同大小的密文，而原始密文仅保留 C、C_1、C_0'、C' 以及新生成的密文子项 F，故重加密一次后的重加密密文存储代价为 $3|G_T|+(7+2N_P)|G|$。其中，$|G|$表示 G 中一个元素所需的存储空间，$|G_T|$表示 G_T 中一个元素所需的存储空间，$|\sigma|$表示签名数据所需的存储空间。

几种方案存储空间占用对比如表 3 所示。本文方案与文献[1]方案相比，为了抵抗满足重加密共享策略的用户与代理之间的共谋攻击，用户私钥增加了 3 个 G 群元素，而用户数据密文增加了 2 个 G 群元素；为了实现重加密可控和可验证性，各自增

表 3 **几种方案的私钥与密文存储空间占用对比**

方案	私钥	原始密文	重加密密文
文献[1]方案	$(1+2N_S)\|G\|$	$\|G_T\|+(1+2N_P)\|G\|$	—
文献[14]方案	$(1+2N)\|G\|$	$\|G_T\|+(2+N)\|G\|$	$3\|G_T\|+(3+N)\|G\|$
文献[15]方案	$(2+N_S)\|G\|$	$2k+(3+2N_P)\|G\|$	$4k+(6+4N_P)\|G\|$
文献[16]方案	$(3+N_S)\|G\|$	$\|G_T\|+(4+2N_P)\|G\|+\|\sigma\|$	$\|G_T\|+(3+2N_P)\|G\|+\|\sigma\|+\|\text{SYM}\|$
文献[17]方案	$(1+4N)\|G\|$	$\|G_T\|+(2+N)\|G\|$	$3\|G_T\|+(3+N)\|G\|$
文献[18]方案	$(1+N_S)\|G\|$	$\|G_T\|+(3+N_P)\|G\|$	$4\|G_T\|+(7+2N_P)\|G\|$
文献[21]方案	$(1+N_S)\|G\|$	$\|G_T\|+(1+N_P)\|G\|$	$2\|G_T\|+(2+N_P)\|G\|$
文献[22]方案	$(4+4N)\|G\|$	$2\|G_T\|+(3+3N)\|G\|$	$4\|G_T\|+(4+3N)\|G\|$
文献[23]方案	$(7+3N)\|G\|$	$2\|G_T\|+(5+3N)\|G\|$	$4\|G_T\|+(6+3N)\|G\|$
文献[24]方案	$(2+4N)\|G\|$	$\|G_T\|+(2+12N)\|G\|$	$3\|G_T\|+(4+12N)\|G\|$
本文方案	$(4+2N_S)\|G\|$	$\|G_T\|+(5+2N_P)\|G\|$	$3\|G_T\|+(7+2N_P)\|G\|$

加了一个 G 群元素。故用户数据密文共增加了 7|G|。

7.2　实验分析

为了评估本文所提出的代理重加密方案的性能，在 Hadoop 环境下采用 Java 语言实现了本文所提出的算法并对其进行了性能实验。在实现的算法中，采用双线性对加密库和 CP-ABE 开发工具包作为基础开发包，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。从素数阶 $y^2 = x^3 + x$ 中选取群 G_1、G_2、G_T，采用对称双线性映射 $e(g,g)$，即 G_1=G_2=G，群 G 和 G_T 中的元素长度为 1 024 位。实验的用户客户端使用的虚拟机配置为一个 Intel(R) Xeon(R) CPU (E5-2620 2.0 GHZ)，内存为 1 GB，系统为 CentOS6.5 64 位。

加密、重加密密钥生成、原始密文解密和重加密密文解密实验分别针对文献[1]方案（仅有加密、原始密文解密）、文献[14,18,22-23]方案和本文方案各进行 20 轮，每轮使用相同大小的数据和属性数量相同的共享访问结构（为了方便比较，共享访问结构的逻辑关系均取为“AND”），分配给用户的属性数量 8 个；每轮实验进行 50 次，取 50 次实验结果的平均值为最终实验结果，共享访问结构的属性数量依次递增。

文献[1,14,18,22-23]方案及本文方案的用户客户端加密时间对比如图 2 所示，文献[14,18,22-23]方案与本文方案的重加密密钥生成时间对比如图 3 所示。在 2 种对比方式中，每种方案的用户客户端计算时间都随共享访问结构属性数量的不断增加呈线性增长，但与文献[1,22-23]方案的加密时间相比，本文方案的加密时间更短且增长较为缓慢；与文献[14,18]方案的加密时间相比，本文方案具有相同的增长趋势，加密时间分别仅多了 90 ms 和 37 ms，但与其重加密密钥生成时间相比，本文方案的重加密密钥生成时间更短，而且随着共享访问结构属性数量的增加，其他方案与本文方案的差距逐渐增大。

文献[1,14,18,22-23]方案以及本文方案的用户客户端原始密文解密时间对比如图 4 所示，文献[14,18,22-23]方案和本文方案的用户客户端重加密密文解密时间对比如图 5 所示。在 2 种对比方式中，其他方案的解密时间都随着访问结构的属性数量不断增加，解密时间与属性数量呈线性关系，而本文方案的解密时间比较稳定，原始密文解密时间约 4.5 ms，重加密密文解密时间约 30.9 ms。

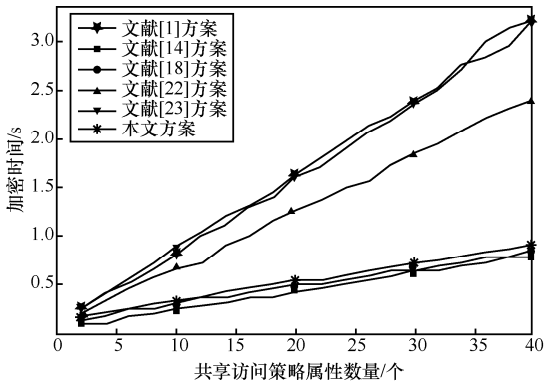


图 2　用户客户端加密时间对比

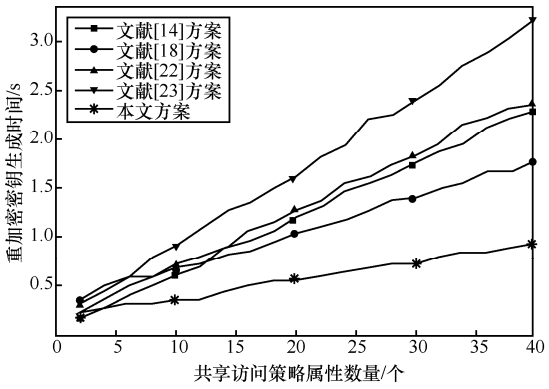


图 3　重加密密钥生成时间对比

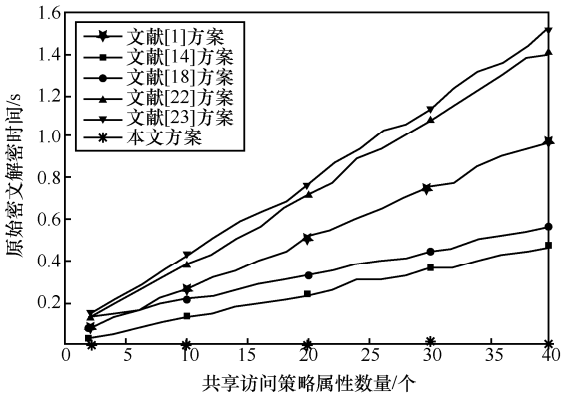


图 4　用户客户端原始密文解密时间对比

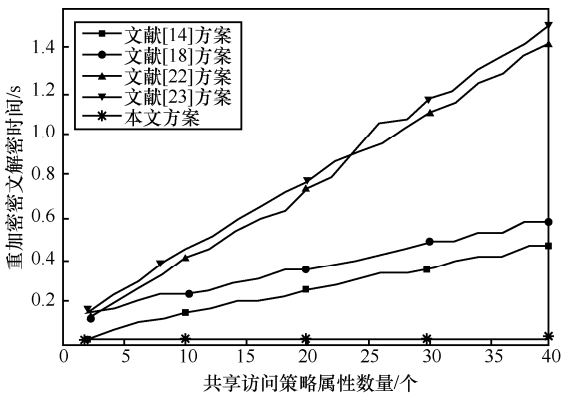


图 5　用户客户端重加密密文解密时间对比

8 结束语

代理重加密技术在代理无法获取明文的情况下，利用用户提供的重加密密钥将一种共享策略下的密文转换为另一种共享策略下的密文，不但降低了用户客户端的性能需求，而且减小了密文数据往返传输所带来的带宽和时间开销。然而，已有的基于 CP-ABE 的代理重加密方案存在仅支持 2 个或 3 个特性、客户端计算量过大、代理能够利用用户提供的重加密密钥重加密该用户所有的密文数据等问题。针对这些问题，本文提出了一种支持多种特性的密文策略基于属性代理重加密方案。该方案支持多种特性的同时将绝大多数计算工作外包给云服务器，用户客户端仅需少量的计算。所提方案构建于经典 CP-ABE 方案的基础之上，不但能对抗恶意用户的合谋攻击，而且还能抵御满足重加密共享策略的用户与云代理服务器之间的合谋攻击，防止满足重加密共享策略的用户非法解密共享者其他密文数据。安全分析表明，所提方案能抵御针对性选择明文攻击。

参考文献：

[1] BETHENCOURT J, SAHAI A, WATERS B. Ciphertext-policy attribute-based encryption[C]/The 2007 IEEE Symposium on Security and Privacy. IEEE, 2007: 321-334.

[2] 冯朝胜, 秦志光, 袁丁. 云数据安全存储技术[J]. 计算机学报, 2015, 38(1): 150-163.
FENG C S, QIN Z G, YUAN D. Techniques of secure storage for cloud data[J]. Chinese Journal of Computers, 2015, 38(1): 150-163.

[3] 冯朝胜, 秦志光, 袁丁, 等. 云计算环境下访问控制关键技术[J]. 电子学报, 2015, 43(2): 312-319.
FENG C S, QIN Z G, YUAN D, et al. Key techniques of access control for cloud computing[J]. Acta Electronica Sinica, 2015, 43(2): 312-319.

[4] ATENIESE G, FU K, GREEN M, et al. Improved proxy re-encryption schemes with applications to secure distributed storage[J]. ACM Transactions on Information and System Security, 2006, 9(1): 1-30.

[5] GREEN M, ATENIESE G. Identity-based proxy re-encryption[C]//The 5th International Conference on Applied Cryptography and Network Security. Springer, 2007: 288-306.

[6] THORBEK R. Linear integer secret sharing and distributed exponentiation[C]//The 9th International Conference on Theory and Practice of Public-Key Cryptography. Springer, 2006: 75-90.

[7] BEIMEL A. Secure schemes for secret sharing and key distribution[D]. Haifa: Israel Institute of Technology, 1996.

[8] KARCHMER M, WIGDERSON A. On span programs[C]//The Eighth Annual Structure in Complexity Theory. IEEE, 1993: 102-111.

[9] WATERS B. Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization[C]//The 14th International Conference on Practice and Theory in Public Key Cryptography Conference on Public Key Cryptography. Springer, 2011: 53-70.

[10] BALU A, KUPPUSAMY K. An expressive and provably secure ciphertext-policy attribute-based encryption[J]. Information Sciences, 2014, 276(4): 354-362.

[11] CANETTI R, HALEVI S, KATZ J. Chosen-ciphertext security from identity- based encryption[C]//The Advances in Cryptology Eurocrypt. Springer, 2004: 207-222.

[12] LING C, NEWPORT C. Provably secure ciphertext policy abe[C]//The 14th ACM Conference on Computer and Communications Security. ACM, 2007: 456-465.

[13] ZHAO J, FENG D G, ZHANG Z F. Attribute-based conditional proxy re-encryption with chosen-ciphertext security[C]//The Global Telecommunications Conference. IEEE, 2010: 1-6.

[14] 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. ACM, 2009: 276-286.

[15] LIANG K T, FANG L M, WONG D S, et al. A ciphertext-policy attribute-based proxy re-encryption with chosen-ciphertext security[C]//The 5th International Conference on Intelligent Networking and Collaborative Systems. IEEE, 2013: 552-559.

[16] LIANG K T, AU M H, LIU J K, et al. A secure and efficient ciphertext-policy attribute-based proxy re-encryption for cloud data sharing[J]. Future Generation Computers Systems, 2015, 52(C): 95-108.

[17] LUO S, HU J B, CHEN Z. Ciphertext policy attribute-based proxy re-encryption[J]. Information & Communications Security, 2010, 6476(4): 401-415.

[18] LI J J, LIU Z H, ZU L H. Chosen-ciphertext secure multi-use unidirectional attribute-based proxy re-encryptions[C]//The Ninth Asia Joint Conference on Information Security. IEEE, 2015: 96-103.

[19] LUAN I, TANG Q, HARTEL P, et al. Efficient and provable secure ciphertext-policy attribute-based encryption schemes[C]//The 5th International Conference on Information Security Practice and Experience. Springer, 2009: 1-12.

[20] KAWAI Y. Outsourcing the re-encryption key generation :flexible ciphertext-policy attribute-based proxy re-encryption[C]//The Information Security Practice and Experience. Springer, 2015: 301-315.

[21] FU X B. Unidirectional proxy re-encryption for access structure transformation in attribute-based encryption schemes[J]. International Journal of Network Security, 2015, 17(2): 142-149.

[22] ZHANG Y H, LI J, CHEN X F, et al. Anonymous attribute-based proxy re-encryption for access control in cloud computing[J]. Security & Communication Networks, 2016, 9(14): 2397-2411.

[23] YIN H J, ZHANG L Y. Security analysis and improvement of an anonymous attribute-based proxy re-encryption[C]//The International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage. Springer, 2017: 344-352.

[24] SEPEHRI M, TROMBETTA A. Secure and efficient data sharing with attribute-based proxy re-encryption scheme[C]//The 12th International Conference on Availability, Reliability and Security. ACM, 2017: 1-63.

[25] MA H, ZHANG R, WAN Z G, et al. Verifiable and exculpable outsourced attribute-based encryption for access control in cloud computing[J]. IEEE Transactions on Dependable & Secure Computing, 2017, 14(4): 679-692.

[26] XIONG H, SUN J F. Comments on “verifiable and exculpable outsourced attribute-based encryption for access control in cloud computing”[J]. IEEE Transactions on Dependable & Secure Computing, 2017, 14(4): 461-462.

[27] FENG X Y, LI C, LI D, et al. Fully secure hidden ciphertext policy

attribute-based proxy re-encryption[C]//The International Conference on Information and Communications Security. Springer, 2017: 192-204.

[28] GE C, SUSILO W, FANG L, et al. A cca-secure key-policy attribute-based proxy re-encryption in the adaptive corruption model for dropbox data sharing system[J]. Designs Codes & Cryptography, 2018(1): 1-17.

[29] YANG Y J, ZHU H Y, LU H B, et al. Cloud based data sharing with fine-grained proxy re-encryption[J]. Pervasive and Mobile Computing, 2016, 28(C): 122-134.

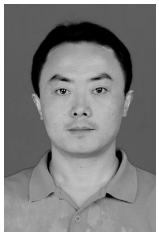
[30] XU P, CHEN H W, ZOU D Q, et al. Fine-grained and heterogeneous proxy re-encryption for secure cloud storage[J]. Chinese Science Bulletin, 2014, 59(32): 4201-4209.

[31] HUANG Q L, MA Z F, YANG Y X, et al. Improving security and efficiency for encrypted data sharing in online social networks[J]. China Communications, 2014, 11(3): 104-117.

[32] BENALOH J, LEICHTER J. Generalized secret sharing and monotone functions[C]//Advances in Cryptology. Springer, 1990: 27-35.

[33] GREEN M, HOHENBERGER S, WATERS B. Outsourcing the decryption of ABE ciphertexts[C]//The 20th Usenix Conference on Security. USENIX Association, 2011: 34.

[作者简介]



冯朝胜（1971− ），男，四川广元人，博士，四川师范大学教授、硕士生导师，主要研究方向为云计算、隐私保护、数据安全。

罗王平（1993− ），男，四川广安人，四川师范大学硕士生，主要研究方向为云计算与大数据安全。

秦志光（1956− ），男，四川荣昌人，博士，电子科技大学教授、博士生导师，主要研究方向为信息安全、分布式计算。

袁丁（1967− ），男，四川宜宾人，四川师范大学教授、硕士生导师，主要研究方向为密码学、信息安全。

邹莉萍（1994− ），女，四川乐山人，四川师范大学硕士生，主要研究方向为云计算与大数据安全。