Advances in Applied Mathematics 应用数学进展, 2021, 10(4), 1387-1402
Published Online April 2021 in Hans. http://www.hanspub.org/journal/aam
https://doi.org/10.12677/aam.2021.104148

Hans汉斯

离线/在线的可验证外包属性代理重加密方案

杨善慧[1,2]，侯金秋[2,3*]，彭长根[1,2,3]，张小玉[1,2]

[1]贵州大学数学与统计学院公共大数据国家重点实验室，贵州 贵阳
[2]贵州大学密码学与数据安全研究所，贵州 贵阳
[3]贵州大学计算机科学与技术学院，贵州 贵阳
Email: *Jinqiu_hou@126.com

收稿日期：2021年3月27日；录用日期：2021年4月15日；发布日期：2021年4月30日

摘 要

基于密文策略的属性代理重加密方案可以同时实现灵活的访问控制和云端密文共享功能。但现有的属性代理重加密方案多以双线性映射构造而成，面临着加解密运算效率低的问题。为解决上述问题，本文提出一种新的加密方案：离线/在线的可验证外包属性代理重加密方案(offline/online attribute-based proxy re-encryption with verifiable outsourced decryption, VF-OO-ABPRE)。基于已有的外包解密属性加密方案，利用离线/在线加密技术，对加密算法进行改进，提高加密效率，结合代理重加密的思想，实现密文共享。同时将解密工作外包给云服务商，并且能够快速地验证外包解密计算结果的正确性。理论分析表明本方案在随机预言机模型中满足选择明文攻击的不可区分安全性，并且提供了外包的可验证性证明，同时能抵抗共谋攻击。

关键词

属性代理重加密，离线/在线加密，外包可验证，可证明安全

Offline/Online Attribute-Based Proxy Re-Encryption with Verifiable Outsourced Decryption

Shanhui Yang[1,2], Jinqiu Hou[2,3*],Changgen Peng[1,2,3], Xiaoyu Zhang[1,2]

[1]State Key Laboratory of Public Big Data, College of Mathematics and Statistics, Guizhou University, Guiyang Guizhou
[2]Institute of Cryptography and Data Security, Guizhou University, Guiyang Guizhou
[3]College of Computer Science and Technology, Guizhou University, Guiyang Guizhou
Email: *Jinqiu_hou@126.com

*通讯作者。

文章引用: 杨善慧, 侯金秋, 彭长根, 张小玉. 离线/在线的可验证外包属性代理重加密方案[J]. 应用数学进展, 2021, 10(4): 1387-1402. DOI: 10.12677/aam.2021.104148

Received: Mar. 27th, 2021; accepted: Apr. 15th, 2021; published: Apr. 30th, 2021

Abstract

Ciphertext policy attribute-based proxy re-encryption (CP-ABPRE) can achieve both flexible access control and ciphertext sharing in the cloud. The existing CP-ABPRE schemes are mostly constructed by bilinear mapping, and the operations of encryption and decryption have low efficiency. To solve these problems, an offline/online attribute-based proxy re-encryption with verifiable outsourced decryption (VF-OO-ABPRE) is proposed in this paper. Based on the existing outsourcing of the decryption of ABE ciphertexts scheme, and by using offline/online encryption technology to improve the encryption algorithm, the proposed scheme can improve the encryption efficiency. Combined with the proxy re-encryption, the ciphertext sharing in the cloud is realized. At the same time, the scheme outsources the decryption work to the cloud service provider, and can verify the correctness of the computing results in an efficient way. The results of theoretical analysis show that the proposed scheme satisfies the chosen plaintext attack secure under the random oracle model and is provided with verifiable outsourced decryption's proof, it also can resist collusive attack.

Keywords

Attribute-Based Proxy Re-Encryption, Offline/Online Encryption, Verifiable Outsourced Decryption, Provable Security

1. 引言

基于属性的加密(attribute-based encryption, ABE) [1]可以实现加密数据的细粒度访问控制，在数据的隐私保护场景中应用广泛。主要有两种形式：基于密文策略的属性加密[2] (ciphertext-policy attribute-based encryption, CP-ABE)和基于密钥策略的属性加密[3] (key-policy attribute-based encryption, KP-ABE)。前者私钥关联授权属性集合，密文关联属性集合满足的访问策略，后者则相反。2007 年，Bethencourt 等[2]第一次提出基于密文策略的属性加密方案(CP-ABE)，采用访问树的结构进行秘密共享。2011 年，Waters 等[4]第一次提出线性秘密共享方案(linear secret sharing scheme, LSSS)的 CP-ABE，较 Bethencourt 等[2]的方案在共享效率上有所提升，但是面临着私钥生成，数据加密和解密阶段的大量运算，且计算量与属性集合或者访问策略的复杂度成线性增长，这将会给移动用户端带来严重的计算负担和电量消耗，为解决上述问题，2011 年，Green 等[5]提出一种解密外包的 ABE 方案，在解密阶段首先将数据密文发送给解密外包服务器，由解密外包服务器对数据密文进行一次密文转换，生成部分密文再发送给用户，用户利用自身私钥解密出数据明文，可以缓解数据用户端的解密计算压力。运用相同的思想，更多的外包 ABE 方案[6] [7] [8] [9]被相继提出。除了解密外包，一些学者也实现了加密外包。2012 年，Li 等[10]构造加密外包的 CP-ABE 方案中，用户只需要做一些简单的操作就可以获得部分密文，云服务器将通过使用外包加密密钥继续完成大部分的加密操作，以获取密文的其他部分。然而，该方案仍然需要用户做大量与属性相关的求幂运算。2016 年，Wang 等[11]提出了一种可验证外包的 ABE 方案，实现加密、解密和密钥生

成外包。但在方案中，密文的大小过大造成了密文存储空间的浪费。

2014 年，Hohenberger 等[12]提出离线/在线的 ABE 技术，将加密分成两个阶段进行，离线阶段在不知道相关的属性集合或者访问结构时，自动完成大量的加密计算预处理工作，在线阶段可以进行较少的计算快速完成属性加密工作。通过这种技术，可以将计算量大的工作尽可能多地在离线阶段完成，减少在线阶段的加密工作量。相较于加密外包，离线/在线同样减轻了用户加密阶段的计算负担，同时保证了数据的机密性。但是该方案不支持外包解密。2017 年，Liu 等[13]提出可验证外包解密的离线/在线 ABE 加密方案，采用离线/在线技术结合可验证外包计算技术，在标准模型下被证明是选择性明文攻击安全的，但是方案中离线/在线技术只用于私钥的生成，用户端的加密负担仍然存在。

单纯的离线/在线加密技术不足以解决解密授权转让和密文共享的问题，代理重加密[14] (proxy re-encryption, PRE)是一种很好的解决方式。2009 年，Liang 等[15]首次在 ABE 方案中引入代理重加密技术，提出第一个基于密文策略的属性代理重加密方案(ciphertext-policy attribute-based proxy encryption, CP-ABPRE)，解决了 ABE 方案中共享访问策略更新和数据密文共享的问题，访问策略采取“AND”门，策略的表达性不高。2013 年，Liang 等[16]结合 LSSS 提出一个 CP-ABPRE 方案，能抵抗选择明文攻击(chosen plaintext attack, CPA)和选择密文攻击(chosen ciphertext attack, CCA)，在合数阶双线性群中的 CP-ABPRE 方案被相继提出[17] [18]，都能实现标准模型中的自适应 CCA 安全。同样地，重加密的密钥生成和重加密都需要大量的计算。为弥补一般服务器在计算和存储能力上的不足问题，2014 年，Gritti 等[19]提出在线/离线的 CP-ABPRE 方案，虽然引入了代理重加密，实则利用代理者进行外包加密，不具备解密授权转让的功能。2015 年，Kawai 等[20]提出将重加密密钥外包的 CP-ABPRE 方案，重加密密钥不再由用户端生成，而是由授权中心来生成，这无疑增加了授权中心的计算负担。2017 年，Sepehri 等[21]提出将用户属性集合与共享访问策略用向量表示的 CP-ABPRE 方案，只有二者内积为 0 时，才能重加密密文，达到实现策略隐藏的目的。Yin 等[22]在 2017 年提出一种改进的具有访问策略隐藏功能的 CP-ABPRE 方案，采用“AND”门共享访问策略，表达性较低。Hong 等[23]在 2018 年提出一种具有密钥隔离功能的 CP-ABPRE 方案，实现了重加密密钥和用户密钥的前向安全。但是，以上的方案都存在加解密效率较低的问题。

基于上述研究，考虑资源受限用户利用移动设备进行加解密和数据共享的现实需求。本文在 Green 等[5]提出的外包 CP-ABE 方案基础上，结合离线/在线加密(offline/online encryption)技术，提出一种离线/在线的可验证外包属性代理重加密方案(offline/online attribute-based proxy re-encryption with verifiable outsourced decryption, VF-OO-ABPRE)。主要贡献如下：

1) 通过结合离线/在线 ABE 加密技术和外包解密技术来实现属性代理重加密方案，是率先以最小的在线计算代价同时实现数据的访问控制功能和密文共享功能。

2) 本文的方案中，大量的计算操作将在离线阶段执行或者外包给云服务商来执行，用户端在线加密和解密的计算控制在常数级别的指数运算，不会随属性集合或访问策略的复杂度而增加计算开销。

3) 给出了本文方案的安全性分析，分析表明方案基于判定性 q-parallel BDHE 假设在随机预言机模型下具有选择明文攻击的不可区分安全性，同时提供本方案的可验证性和抗共谋攻击证明，理论分析表明本方案在功能性和效率方面有优势。

2. 相关概念及方案

2.1. 双线性映射

令 $\mathbb{G}$ 和 $\mathbb{G}_T$ 是两个 p 阶乘法循环群，定义一个双线性映射 $e:\mathbb{G}\times\mathbb{G}\to\mathbb{G}_T$，满足以下性质：

1) 双线性：对于 $\forall u,v\in\mathbb{G}$， $a,b\in\mathbb{Z}_p$， $e\left(u^a,v^b\right)=e\left(u^b,v^a\right)=e(u,v)^{ab}$；

2) 非退化性：存在 g 是 $\mathbb{G}$ 的生成元，使得 $e(g,g)\neq 1$；

3) 可计算性： $\forall u,v\in\mathbb{G}$，存在一个有效算法可计算出 $e(u,v)$。

2.2. 困难性假设

判定性 q-parallel BDHE 假设：给定一个安全参数 $\kappa\in\mathbb{N}$，整数 q，设 $\mathbb{G}$， $\mathbb{G}_T$ 是两个素数阶 p 的乘法循环群。群生成器 $\mathcal{G}(\kappa)$ 生成一个双线性群组 $(p,\mathbb{G},\mathbb{G}_T,e)$，$g$ 是 $\mathbb{G}$ 的生成元，取随机值 $a,s,b_1,\cdots,b_q\in\mathbb{Z}_p$。公开：$\boldsymbol{Y}=g,g^s,g^a,\cdots,g^{a^q},g^{a^{q+2}},\cdots,g^{a^{2q}}$， $\forall 1\le j\le q,g^{s\cdot b_j},g^{a/b_j},\cdots,g^{a^q/b_j},g^{a^{q+2}/b_j},\cdots,g^{a^{2q/b_j}}$，$\forall 1\le j,k\le q,k\neq j,g^{a\cdot s\cdot b_k/b_j},\cdots,g^{a^q\cdot s\cdot b_k/b_j}$， $R\in\mathbb{G}_T$。在概率多项式时间(probabilistic polynomial time, PPT)内不存在算法 $\mathcal{B}$ 以不可忽视的优势区分出随机元素 $R\in\mathbb{G}_T$ 和 $T=e(g,g)^{a^{q+1}s}\in\mathbb{G}_T$。则称判定性 q-parallel BDHE 假设是成立的。

CDH 假设(computational Diffie-Hellman, CDH)：给定一个安全参数 $\kappa\in\mathbb{N}$，设 $\mathbb{G}$， $\mathbb{G}_T$ 是两个素数阶 p 的乘法循环群。群生成器 $\mathcal{G}(\kappa)$ 生成一个双线性群组 $(p,\mathbb{G},\mathbb{G}_T,e)$， g 是 $\mathbb{G}$ 的生成元，取随机值 $\alpha,\beta\in\mathbb{Z}_p$，计算 $g^\alpha,g^\beta,g^{\alpha\beta}$。在多项式时间(PPT)内不存在算法 $\mathcal{B}$ 以不可忽视的优势在已知 g^α 和 g^β 情况下计算出 $g^{\alpha\beta}$，则称 CDH 假设是成立的。

2.3. 基础解密外包的 ABE 方案

Green 等[5]CPA 安全的解密外包的 ABE 方案一共包含五个算法：系统初始化算法 *Setup*、密钥生成算法 *KeyGen*、外包密钥生成算法 $KeyGen_{out}$、加密算法 *Encrypt*、转换密文生成算法 $Transform_{out}$、转换密文解密算法 $Decrypt_{out}$。算法运行过程如下：

Setup(λ,U)：输入系统参数 $\lambda\in\mathbb{N}$ 和属性全集 $U=\{0,1\}^*$，选择两个 p 阶的循环群 $\mathbb{G}$ 和 $\mathbb{G}_T$， $\mathbb{G}$ 的生成元为 g， $e(g,g)\subset\mathbb{G}_T$，定义杂凑函数 $F:\{0,1\}^*\to\mathbb{G}$。选择两个随机数 $\alpha,a\in\mathbb{Z}_p^*$，输出系统主密钥 $MSK=\left(g^\alpha,PK\right)$ 和系统公钥 $PK=\left(g,e(g,g)^\alpha,g^a,F\right)$，公开 PK。

KeyGen(PK,MSK,S)：输入系统主密钥 MSK、系统公钥 PK 属性集合 S，随机选择 $t\in\mathbb{Z}_p$，对于所有的 $x\in S$，计算 $K=g^\alpha g^{at}$， $L=g^t$， $\left\{K_x=F\left(\rho(x)^t\right)\right\}_{x\in S}$。输出私钥 $SK_S=\left(S,K,L,\{K_x\}_{x\in S}\right)$。

KeyGen$_{out}(MSK,S,SK_S)$：输入系统主密钥 MSK、属性集合 S，私钥 SK_S，随机选择 $z\in\mathbb{Z}_p$，计算 $K_{tk}=K^{(1/z)}$， $L_{tk}=L^{(1/z)}$， $\left\{K_{x,tk}=K_x^{(1/z)}\right\}_{x\in S}$，输出外包密钥 $DTK_S=\left(S,K_{tk},L_{tk},\{K_{x,tk}\}_{x\in S}\right)$，用户解密密钥 $Key=\left(z,DTK_S\right)$。

Encrypt$\left(PK,m,(M,\rho)\right)$：输入系统公钥 PK 和明文 m，属性 S 满足的 LSSS 访问结构 (M,ρ)(其中 M 是一个 $l\times n$ 的矩阵， $\rho(\cdot)$ 是一个单射函数可以把矩阵 M 的每一行映射成一个属性)， $s\in\mathbb{Z}_p$ 是共享的秘密值，取随机数 $y_2,\cdots,y_n\in\mathbb{Z}_p$，向量 $v=(s,y_2,\cdots,y_n)\in\mathbb{Z}_p^n$，$M_i$ 表示矩阵 M 的第 i 行，计算 $\lambda_i=\left\{M_iv^{\mathrm{T}}\right\}_{i\in\{1,\cdots,l)}$。取随机数 $r_1,\cdots,r_n\in\mathbb{Z}_p$， 对于任意 $i\in\{1,\cdots,l\}$， 计算得到密文组件： $C=m\cdot e(g,g)^{\alpha s}$， $C_1=g^s$，$\left\{C_{2,i}=\mathrm{g}^{a\lambda_i}F\left(\rho(i)\right)^{-r_i},C_{3,i}=g^{r_i}\right\}_{i\in\{1,\cdots,l\}}$。输出密文 $CT=\left(C,C_1,\{C_{2,i},C_{3,i}\}_{i\in\{1,\cdots l\}}\right)$。

Transform$_{out}(DTK_S,CT)$：输入密文 CT 和外包密钥 DTK_S，若用户属性集 S 不满足访问结构 (M,ρ)，则输出 $\perp$。若用户属性集 S 满足访问结构 (M,ρ)，则定义 $I\subset\{1,\cdots,l\}$ 为 $I=\{i:\rho(i)\in S\}$， $\{\lambda_i\}$ 是根据矩阵 M 对秘密 s 的有效共享，存在一个常数集 $\left\{\omega_i\in\mathbb{Z}_p\right\}_{i\in I}$，使得 $\sum_{i\in I}\omega_i\cdot\lambda_i=s$。计算：$CT_{part}=e(C_1,K_{tk})/\prod_{i\in I}\left(e\left(C_{2,i},L_{tk}\right)\cdot e\left(C_{3,i},K_{\rho(i),tk}\right)\right)^{\omega_i}=e(g,g)^{\alpha s/z}$。输出外包解密转换密文 $TC=\left(C,CT_{part}\right)$。

Decrypt$_{out}(Key,CT)$：输入密文 CT 和解密密钥 Key，若密文 CT 未进行外包解密，则首先运行转换密文生成算法 $Transform_{out}\left(DTK_S,CT\right)$，若算法输出 $\perp$，最终也输出 $\perp$。否则，计算 $C/CT_{part}^z=m$。

3. VF-OO-ABPRE 方案定义及安全模型

本方案的系统中，主要包含 5 个实体：属性授权中心(attribute authority, AA)、数据拥有者(data owner, DO)、数据用户(data user, DU)、存储云服务商(storage-cloud service provider, S-CSP)、解密云服务商(decryption cloud service provider, D-CSP)、云服务代理商 Proxy。

属性授权中心 AA 提供密钥生成服务，是一个完全可信的机构；数据拥有者 DO 可以使用移动设备加密明文信息并存储在云端；数据用户 DU 可以使用移动设备从云端下载密文并解密，包含两类：授权用户 DU 和被授权用户 DU，授权用户 DU 可以直接解密由数据拥有者 DO 加密的原始密文，当数据需要重加密时，生成重加密密钥发送给云服务代理商 Proxy，被授权用户 DU 用户只能解密被重加密的密文；云服务代理商 Proxy 提供数据密文重加密服务；存储云服务商 S-DCP 提供数据存储服务；解密云服务商 D-CSP 提供数据解密服务，所有的云服务商都是诚实且好奇的。

3.1. 方案形式化定义

离线/在线的可验证外包属性代理重加密方案(VF-OO-ABPRE)包含八个算法：系统初始化算法 *Setup* 、密钥生成算法 *KeyGen* 、外包密钥生成算法 $KeyGen_{out}$ 、加密算法 *Encrypt* (包含离线加密算法 *Offline.Enc* 和在线加密算法 *Online.Enc* 两种)、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* (包含转换密文生成算法 $Transform_{out}$ 、转换密文解密算法 $Decrypt_{out}$ 两种)。其中，解密算法包含了初始密文解密算法 Dec_1 和重加密密文解密算法 Dec_2。算法运行过程如下：

1) 系统初始化算法 $Setup\left(1^{\lambda},U\right)$：由可信方 AA 执行，输入系统参数 $\lambda\in\mathbb{N}$ 和属性全集 U，系统主密钥 MSK 和系统公钥 PK，公开 PK。

2) 密钥生成算法 $KeyGen\left(PK,MSK,S\right)$：可信方 AA 根据系统公钥 PK，系统主密钥 MSK 和与用户相关的属性集 S，为用户生成私钥 SK_S 并通过安全通道分发给用户。

3) 外包密钥生成算法 $KeyGen_{out}\left(S,SK_S\right)$：用户 DU 输入自己的私钥 SK_S，属性集合 S，生成解密密钥 *Key* 用户秘密保存，外包密钥 DTK_S 通过安全通道发给解密云服务商 D-CSP。

4) 加密算法 *Encrypt*

a) 离线加密算法 $Offline.Enc\left(PK,P\right)$：数据拥有 DO 在闲时执行，输入系统公钥 PK 和假设用户最大可能的属性集合 $P\subseteq U$ 。生成临时密钥 TK 保存在本地，输出中间密文 ICH 并将其上传至存储云服务商 S-CSP。

b) 在线加密算法 $Online.Enc\left(PK,m,ICH,TK,\left(M,\rho\right)\right)$：数据拥有者 DO 在真正加密时执行，输入系统公钥 PK，明文信息 m，中间密文 ICH，临时密钥 TK，以及属性 S 满足的访问结构 $\mathbb{A}=\left(M,\rho\right)$，生成初始密文 $C_{(M,\rho)}$ 和验证标识 VK_m 发送给存储云服务商 S-CSP 进行存储。

5) 重加密密钥生成算法 $ReKeyGen\left(PK,SK_S,\left(M',\rho'\right)\right)$：授权用户 DU 根据系统公钥 PK，用户自己私钥 SK_S 和另一个访问结构 $\left(M',\rho'\right)$，生成重加密密钥 $RK_{S\to(M',\rho')}$ 并发送给云服务器代理商 Proxy。注意：其中访问结构 $\left(M,\rho\right)$ 和访问结构 $\left(M',\rho'\right)$ 是不相交的。

6) 重加密算法 $ReEnc\left(PK,RK_{S\to(M',\rho')},CT_{(M,\rho)}\right)$：云服务代理商 Proxy 根据系统公钥 PK，重加密密文组件 $RK_{S\to(M',\rho')}$，加密初始密文 $CT_{(M,\rho)}$ 得到重加密密文 $C_{S\to(M',\rho')}$ 和验证标识 VK_m 并发送给解密云服务商 D-CSP。

7) 重加密验证算法 $ReEncVerify\left(PK,\delta,C_6,C_7\right)$：授权用户 DU 输入系统公钥 PK，群元素 δ，重加密密钥组件 $\left(C_6,C_7\right)$，验证重加密密文是否被云服务代理商 Proxy 正确加密。

8) 解密算法 *Decrypt*

a) 初始密文解密算法 Dec_1

- 转换密文生成算法 $Transform_{out}\left(PK, DTK_S, C_{(M,\rho)}\right)$：解密云服务商 D-CSP 输入系统公钥 PK，外包密钥 DTK_S 和初始密文 $C_{(M,\rho)}$，系统首先检验外包密钥 DTK_S 中的属性集合 S 是否满足 $C_{(M,\rho)}$ 中的访问结构 (M,ρ)，若满足，则预解密初始密文 $C_{(M,\rho)}$ 得到外包解密转换密文 TC 并发送给授权用户 DU。否则，输出 $\perp$，终止操作。
- 转换密文解密 $Decrypt_{out}\left(PK, Key, TC, VK_m\right)$：授权用户 DU 执行，输入解密密钥 Key、来自解密云服务商的外包解密转换密文 TC、验证标识 VK_m，解密并对结果进行验证，验证正确得到明文 m。否则，输出 $\perp$。

b) 重加密密文解密算法 Dec_2

- 转换密文生成算法 $Transform_{out}\left(PK, DTK_{S'}, C_{S\to(M',\rho')}\right)$：解密云服务商 D-CSP 输入系统公钥 PK，外包密钥 $DTK_{S'}$ 和重加密密文 $C_{S\to(M',\rho')}$，系统首先检验外包密钥 $DTK_{S'}$ 中的属性集合 S' 是否满足 $C_{S\to(M',\rho')}$ 中的访问结构 (M',ρ')，若满足，则预解密重加密密文 $C_{S\to(M',\rho')}$ 得到外包解密转换密文 TC' 并发送给被授权用户 DU。否则，输出 $\perp$，终止操作。
- 转换密文解密 $Decrypt_{out}\left(PK, Key', TC', VK_m\right)$：被授权用户 DU 执行，输入解密密钥 Key'、来自解密云服务商的外包解密转换密文 TC'、验证标识 VK_m，解密并对结果进行验证，验证正确得到明文 m。否则，输出 $\perp$。

3.2. 方案安全模型

选择性选择明文攻击(selective chosen plaintext attack, s-CPA)安全博弈游戏：一个 VF-OO-ABPRE 方案是 s-CPA 安全的，则没有一个敌手 $\mathcal{A}$ 能在概率多项式时间(PPT)内以不可忽略的优势赢得下面的博弈。定义 $\mathcal{C}$ 为挑战者，λ 为系统安全参数，U 为属性全集。博弈游戏如下：

初始化阶段 *Init* $\mathcal{A}$ 选择一个挑战的访问结构 (M^*,ρ^*)。

系统建立阶段 *Setup* $\mathcal{C}$ 运行系统初始化算法 $Setup\left(1^\lambda, U\right)$ 得到系统主密钥 MSK 和系统公钥 PK，输出系统公钥 PK 给 $\mathcal{A}$，自己保留主密钥 MSK。

查询阶段 1 *Phase*1 $\mathcal{A}$ 可以重复向预言机进行如下查询：

1) 私钥查询 $\mathcal{O}_{sk}(S)$：$\mathcal{A}$ 提交一个属性集合 S，该属性集合 S 不满足挑战访问结构 (M^*,ρ^*)，$\mathcal{C}$ 运行私钥生成算法 $KeyGen(PK, MSK, S)$ 和外包密钥生成算法 $KeyGen_{out}(S, SK_S)$，并返回私钥 SK_S 和外包密钥 DTK_S 给 $\mathcal{A}$。

2) 重加密密钥查询 $\mathcal{O}_{rk}\left(S,(M',\rho')\right)$：$\mathcal{A}$ 提交一个属性集合 S，该属性集合 S 不满足挑战访问结构 (M^*,ρ^*)，满足访问结构 (M',ρ')，$\mathcal{C}$ 运行重加密密钥生成算法 $ReKeyGen\left(PK, SK_S, (M',\rho')\right)$ 返回 $RK_{S\to(M',\rho')}$ 给 $\mathcal{A}$。其中 SK_S 由算法 $KeyGen(PK, MSK, S)$ 生成。

挑战阶段 *Challenge* $\mathcal{A}$ 输出两个相同长度的消息 m_0 和 m_1 给 $\mathcal{C}$。$\mathcal{C}$ 随机选择 $b\in\{0,1\}$ 并运行加密算法 $Online.Enc\left(PK, m_b, ICH^*, TK^*, (M^*,\rho^*)\right)$ 得到挑战密文 C_b 给 $\mathcal{A}$。其中 $Offline.Enc(PK,P)\to\left(TK^*, ICH^*\right)$。

查询阶段 2 *Phase* 2 $\mathcal{A}$ 继续查询阶段 1 中的查询。

猜测阶段 *Guess* $\mathcal{A}$ 输出一个猜测 $b'\in\{0,1\}$。如果 $b'=b$，则 $\mathcal{A}$ 获胜。$\mathcal{A}$ 获胜的优势被定义为：

$$\varepsilon_1 = Adv_{\mathcal{A}}^{s-CPA}(\lambda) = \left|Pr[b'=b] - \frac{1}{2}\right|。$$

定义 3.1 一个 VF-OO-ABPRE 方案是 s-CPA 安全的，则没有一个敌手 $\mathcal{A}$ 能在概率多项式时间(PPT)内以不可忽略的优势赢得上面的博弈，即 $\varepsilon_1 \le negl(\lambda)$。

可验证博弈游戏：可验证主要是确保外包解密阶段的转换密文是否被正确执行。一个 VF-OO-ABPRE 方案具有可验证性，则没有一个敌手 $\mathcal{A}$ 能在概率多项式时间(PPT)内以不可忽略的优势赢得下面的博弈。定义 $\mathcal{C}$ 为挑战者，λ 为系统安全参数，U 为属性全集。博弈游戏如下：

系统建立阶段 *Setup* $\mathcal{C}$ 运行系统初始化算法 $Setup\left(1^{\lambda},U\right)$ 得到系统主密钥 MSK 和系统公钥 PK，输出系统公钥 PK 给 $\mathcal{A}$，自己保留主密钥 MSK。

查询阶段 1 *Phase 1* $\mathcal{A}$ 可以重复向预言机进行如下查询：

1) 私钥查询 $\mathcal{O}_{sk}\left(S\right)$：$\mathcal{A}$ 提交一个属性集合 S，该属性集合 S 不满足挑战访问结构 $\left(M^{*},\rho^{*}\right)$，$\mathcal{C}$ 运行私钥生成算法 $KeyGen\left(PK,MSK,S\right)$ 和外包密钥生成算法 $KeyGen_{out}\left(S,SK_{S}\right)$，并返回私钥 SK_S 和外包密钥 DTK_S 给 $\mathcal{A}$。

2) 重加密密钥查询 $\mathcal{O}_{rk}\left(S,\left(M',\rho'\right)\right)$：$\mathcal{A}$ 提交一个属性集合 S，该属性集合 S 不满足挑战访问结构 $\left(M^{*},\rho^{*}\right)$，满足访问结构 $\left(M',\rho'\right)$，$\mathcal{C}$ 运行重加密密钥生成算法 $ReKeyGen\left(PK,S,SK_{S},\left(M',\rho'\right)\right)$ 返回 $RK_{S\to\left(M',\rho'\right)}$ 给 $\mathcal{A}$。其中 SK_S 由算法 $KeyGen\left(PK,MSK,S\right)$ 生成。

挑战阶段 *Challenge* $\mathcal{A}$ 输出两个相同长度的消息 m_0 和 m_1 以及一个挑战访问结构 $\left(M^{*},\rho^{*}\right)$ 给 $\mathcal{C}$。$\mathcal{C}$ 随机选择 $b\in\left\{0,1\right\}$ 并运行加密算法 $Online.Enc\left(PK,m_b,ICH^{*},TK^{*},\left(M^{*},\rho^{*}\right)\right)$ 得到挑战密文 C_b 给 $\mathcal{A}$。其中 $Offline.Enc\left(PK,P\right)\to\left(TK^{*},ICH^{*}\right)$。

查询阶段 2 *Phase 2* $\mathcal{C}$ 按照查询阶段 1 的方式响应 $\mathcal{A}$ 的询问。但是 $\mathcal{A}$ 不能询问满足访问结构 $\left(M^{*},\rho^{*}\right)$ 的属性集合 S。

猜测阶段 *Guess* $\mathcal{A}$ 输出满足挑战访问结构 $\left(M^{*},\rho^{*}\right)$ 的属性集合 S^{*} 和满足属性集合 S^{*} 的外包解密转换密文 TC^{*}。若能成功恢复出明文 m_b，则 $\mathcal{A}$ 获胜。$\mathcal{A}$ 获胜的优势为 $\varepsilon_2=Adv_{\mathcal{A}}^{Ver.}\left(\lambda\right)=\left|Pr\left[\mathcal{A}\text{ Wins}\right]\right|$。

定义 3.2 一个 VF-OO-ABPRE 方案具有可验证性，则没有一个敌手 $\mathcal{A}$ 能在概率多项式时间 $\left(PPT\right)$ 内以不可忽略的优势赢得上面的博弈，即 $\varepsilon_2\le negl\left(\lambda\right)$。

抗共谋攻击博弈游戏：一个 VF-OO-ABPRE 方案是抗共谋攻击的。则没有一个敌手 $\mathcal{A}$ 能在概率多项式时间(PPT)内以不可忽略的优势赢得下面的博弈。定义 $\mathcal{C}$ 为挑战者，λ 为系统安全参数，U 为属性全集。博弈游戏如下：

系统建立阶段 *Setup* $\mathcal{C}$ 运行系统初始化算法 $Setup\left(1^{\lambda},U\right)$ 得到系统主密钥 MSK 和系统公钥 PK，输出系统公钥 PK 给 $\mathcal{A}$，自己保留主密钥 MSK。

查询阶段 *Phase* 重加密密钥查询 $\mathcal{O}_{rk}\left(S^{*},\left(M',\rho'\right)\right)$：$\mathcal{A}$ 提交一个属性集合 $S^{*}\subseteq U$，该属性集合 S^{*} 不满足访问结构 $\left(M',\rho'\right)$，如果 S^{*} 从未被询问过，$\mathcal{C}$ 首先运行 $KeyGen\left(PK,MK,S^{*}\right)$ 生成 $SK_{S^{*}}$，运行重加密密钥生成算法 $ReKeyGen\left(PK,S^{*},SK_{S^{*}},\left(M',\rho'\right)\right)$ 并返回 $RK_{S^{*}\to\left(M',\rho'\right)}$ 给 $\mathcal{A}$。同时运行 $KeyGen\left(PK,MK,S'\right)$ 得到 $SK_{S'}$ 发送给 $\mathcal{A}$，其中 $SK_{S'}$ 满足访问结构 $\left(M',\rho'\right)$。

挑战阶段 *Challenge* 最后，$\mathcal{A}$ 提交一个密钥 $SK_{S''}$，如果 $SK_{S''}=SK_{S^{*}}$ 则 $\mathcal{A}$ 赢得游戏。$\mathcal{A}$ 赢得游戏的优势为 $\varepsilon_3=Adv_{\mathcal{A}}^{CR}\left(\lambda\right)=\left|Pr\left[SK_{S''}=SK_{S^{*}}\right]-\frac{1}{2}\right|$。

定义 3.3 一个 VF-OO-ABPRE 方案是抗共谋攻击的。则没有一个敌手 $\mathcal{A}$ 能在概率多项式时间(PPT)内以不可忽略的优势赢得上面的博弈，即 $\varepsilon_3\le negl\left(\lambda\right)$。

4. VF-OO-ABPRE 方案构造

4.1. 具体方案

1) 系统初始化算法 $Setup\left(1^{\lambda},U\right)$

该算法由 AA 执行。AA 输入安全参数 $\lambda\in\mathbb{N}$ 和属性全集 $U=\left\{1,2,\cdots,n\right\}\subset\mathbb{Z}_p^{*}$，选择一个 p 阶的乘法

循环群$\mathbb{G}$，生成元为g，$g_1,h_1,\cdots,h_n\in\mathbb{G}$为随机群元素。另一个$p$阶的乘法循环群为$\mathbb{G}_T$，$e(g,g)\subset\mathbb{G}_T$，选择两个随机数$\alpha,a\in\mathbb{Z}_p^*$，定义抗碰撞杂凑函数$H_0:\{0,1\}^*\to\mathbb{Z}_p$，$H_1:\{0,1\}^*\to\{0,1\}^\lambda$，$H_2:\{0,1\}^{2\lambda}\to\{0,1\}^*$。最后生成系统公钥$PK=\left(p,g,e(g,g)^\alpha,g^a,g_1,h_1,\cdots,h_n,H_0,H_1,H_2\right)$，主密钥$MSK=\left(g^\alpha\right)$，可信机构将$MSK=\left(g^\alpha\right)$作为主密钥保存起来。

2) 密钥生成算法 $KeyGen(PK,MSK,S)$

该算法由AA执行。输入系统公钥PK，主密钥MSK和属性集$S\subseteq U$，AA随机选择$t\in\mathbb{Z}_p$，对于所有的$i\in S$，计算$K=g^\alpha g^{at}$，$L=g^t$，$\left\{K_i=h_i^t\right\}_{i\in S}$，最后输出用户私钥$SK_S=\left(S,K,L,\left\{K_i\right\}_{i\in S}\right)$通过安全通道发给用户。

3) 外包密钥生成算法 $KeyGen_{out}\left(S,SK_S\right)$

该算法由解密用户DU执行，输入用户私钥SK_S，随机选择$z\in\mathbb{Z}_p$，满足$gcd(z,p)=1$，计算$K_{tk}=K^{(1/z)}$，$L_{tk}=L^{(1/z)}$，$\left\{K_{i,tk}=K_i^{(1/z)}\right\}_{i\in S}$。输出解密密钥$Key=z$用户秘密保存，外包密钥$DTK_S=\left(S,K_{tk},L_{tk},\left\{K_{i,tk}\right\}_{i\in S}\right)$通过安全通道发给解密云服务商D-CSP。

4) 加密算法 *Encrypt*

加密算法分为两个阶段，由数据拥有者DO在闲时执行的离线加密算法*Offline.Enc*和真正加密阶段执行的在线加密算法*Online.Enc*。

a) 离线加密算法$Offline.Enc(PK,P)$。该阶段算法由数据拥有者DO执行，利用文献[12]的“pooling”思想，输入系统公钥PK，用户根据自己需求假设最大可能的属性集合$P\subseteq U$。对于每一个$i\in P$，随机选取$\lambda_{i,1},s_i\in\mathbb{Z}_p$，计算$\left\{C_{2,i}=g^{a\lambda_{i,1}}h_i^{-s_i},C_{3,i}=g^{s_i}\right\}_{i\in P}$作为密文组件，输出中间密文$ICH=\left(\left\{C_{2,i},C_{3,i}\right\}_{i\in P}\right)$，可将其上传至存储云服务商S-CSP以节省本地存储资源。将临时密钥$TK=\left\{\lambda_{i,1}\right\}_{i\in P}$保存在本地。

b) 在线加密算法$Online.Enc(PK,m,ICH,TK,(M,\rho))$：该阶段算法由数据拥有者DO执行。输入系统公钥$PK$，明文信息$m\in\{0,1\}^\lambda$，中间密文$ICH$，临时密钥$TK$，以及属性$S$满足的访问结构$\mathbb{A}=(M,\rho)$（其中$M$是一个$l\times n$的矩阵，$\rho(\cdot)$是一个单射函数可以把矩阵$M$的每一行映射成一个属性值)。从“pooling”中随机选取l个ICH，随机选择$W\in\mathbb{G}$，并计算$s=H_0(W,m)$，取随机数$y_2,\cdots,y_n\in\mathbb{Z}_p$，向量$v=(s,y_2,\cdots,y_n)\in\mathbb{Z}_p^n$，$M_j$表示矩阵$M$的第$j$行，计算$\lambda_j=\left\{M_jv^{\mathrm{T}}\right\}_{j\in\{1,\cdots,l\}}$。对于任意$j\in\{1,\cdots,l\}$，计算密文组件得：

$$C=W\cdot e(g,g)^{\alpha s},\ C_1=g^s,\ C_{2,j}=C_{2,\rho(j)},\ C_{3,j}=C_{3,\rho(j)},\ C_{4,j}=\lambda_j-\lambda_{j,1},\ k=H_1(W),$$
$$C_5=k\oplus m,\ C_6=g^{H_1(e(g,g)^{\alpha s})},\ VK_m=H_2\left(k\|C_5\right)$$

最后输出密文初始密文$C_{(M,\rho)}=\left(C,C_1,\left\{C_{2,j},C_{3,j},C_{4,j}\right\}_{j\in\{1,\cdots,l\}},C_5,C_6\right)$和验证标识$VK_m=H_2\left(k\|C_5\right)$发送给存储云服务商S-CSP进行存储。

5) 重加密密钥生成算法 $ReKeyGen(PK,SK_S,(M',\rho'))$

该算法由授权用户DU执行。假设授权用户A满足访问结构$\mathbb{A}=(M,\rho)$，用户A运行重加密密钥生成算法*ReKeyGen*，输入系统公钥PK、和用户A的私钥SK_S，新的访问结构$\mathbb{A}'=(M',\rho')$（其中M'是一个$l'\times n'$的矩阵，$\rho'(\cdot)$是一个单射函数可以把矩阵M'的每一行映射成一个属性值，且属性集合S'满足$\mathbb{A}'=(\boldsymbol{M}',\rho'))$。随机选择$\delta\in\mathbb{G}$，$s'\in\mathbb{Z}_p^*$，其中$s'$是共享的秘密值。取随机数$y_2',\cdots,y_n'\in\mathbb{Z}_p$，向量$v'=(s',y_2',\cdots,y_n')$，$M_j'$表示矩阵$M'$的第$j$行，$\lambda_j'=\left\{M_j'v'^{\mathrm{T}}\right\}_{j\in\{1,\cdots,l'\}}$。首先，随机选取$\lambda_{j,1}',s_j'\in\mathbb{Z}_p$，对于$j\in\{1,\cdots,l'\}$，计算密文组件：

$$C'=\delta\cdot e(g,g)^{\alpha s'},\ C_1'=g^{s'},\ C_{2,j}'=g^{a\lambda_{j,1}'}h_{\rho'(j)}^{-s_j'},\ C_{3,j}'=g^{s_j'},\ C_{4,j}'=\lambda_j'-\lambda_{j,1}'$$

记$C_{(M',\rho')}=\left(C',C_1',\left\{C_{2,j}',C_{3,j}',C_{4,j}'\right\}_{j\in\{1,\cdots,l'\}}\right)$。随机选择$\theta\in\mathbb{Z}_p$，并计算$rk_1=K^{H_1(\delta)}\cdot g_1^\theta$，$rk_2=g_1^\theta$，$rk_3=L^{H_1(\delta)}$，

对于所有的 $j\in\{1,\cdots,l'\}$， $rk_4=C_{(M',\rho')}$， $rk_j=K_j^{H_1(\delta)}$，最后输出重加密密钥 $RK_{S\to(M',\rho')}=\left(S,rk_1,rk_2,rk_3,rk_4,rk_j\right)$ 发送给 Proxy。

6) 重加密算法 $ReEnc\left(PK,RK_{S\to(M',\rho')},CT_{(M,\rho)}\right)$

该算法由 Proxy 执行。Proxy 收到重加密密钥 $RK_{S\to(M',\rho')}$ 后，运行重加密算法 *ReEncrypt*，输入系统公钥 PK，重加密密钥 $RK_{S\to(M',\rho')}$ 和初始密文 $C_{(M,\rho)}$。若 $C_{(M,\rho)}$ 与访问结构 $\mathbb{A}=(M,\rho)$ 有关，且属性集 S 满足访问结构 $\mathbb{A}$，则定义 $I\subset\{1,\cdots,l\}$ 为 $I=\{j:\rho(j)\in S\}$，$\{\lambda_j\}$ 是根据矩阵 M 对秘密 s 的有效共享，存在一个常数集 $\{\omega_j\in\mathbb{Z}_p\}_{j\in I}$，使得 $\sum_{j\in I}\omega_j\cdot\lambda_j=s$。计算：

$$C_7=\frac{e(C_1,rk_1)/e(C_1,rk_2)}{\prod_{j\in I}\left(e\left(C_{2,j}\cdot g^{a\cdot C_{4,j}},rk_3\right)\cdot e\left(C_{3,j},rk_{\rho(j)}\right)\right)^{\omega_j}}=e(g,g)^{\alpha s H_1(\delta)},$$

最后输出重加密密文 $C_{S\to(M',\rho')}=\left((M',\rho'),C,C_5,C_6,C_7,rk_4\right)$ 和验证标识 VK_m 发送给解密云服务商 D-CSP。

7) 重加密验证算法 $ReEncVerify\left(PK,\delta,C_6,C_7\right)$

该阶段由授权用户 DU 执行。验证重加密密文是否被云服务代理商 Proxy 正确加密时，δ 对授权用户 DU 是已知的，输入来自云服务器 Proxy 的 C_7，计算 $V=(C_7)^{H_1(\delta)^{-1}}$，若 $C_6=g^{H_1(V)}$，则说明代理商 Proxy 重加密计算结果正确，输出 *True*，否则输出终止符 $\perp$。

8) 解密算法 *Decrypt*

解密算法包含了对初始密文的解密算法 Dec_1 和对重加密密文的解密算法 Dec_2。二者均包含了由外包云服务器 D-CSP 提供的外包解密转换密文生成算法 $Transform_{out}$ 和用户端提供的转换密文解密 $Decrypt_{out}$。

a) 初始密文解密算法 Dec_1

- 转换密文生成算法 $Transform_{out}\left(PK,DTK_S,C_{(M,\rho)}\right)$：该阶段由解密云服务商 D-CSP 执行。若用户属性集 S 不满足访问结构 $\mathbb{A}=(M,\rho)$，则输出 $\perp$。若 $C_{(M,\rho)}$ 与访问结构 $\mathbb{A}$ 有关，授权人外包密钥 $DTK_S=\left(S,K_{tk},L_{tk},\{K_{i,tk}\}_{i\in S}\right)$ 与用户属性集 S 相关，且用户属性集 S 满足访问结构 $\mathbb{A}$，则定义 $I\subset\{1,\cdots,l\}$ 为 $I=\{j:\rho(j)\in S\}$，$\{\lambda_j\}$ 是根据矩阵 M 对秘密 s 的有效共享，存在一个常数集 $\{\omega_j\in\mathbb{Z}_p\}_{j\in I}$，使得 $\sum_{j\in I}\omega_j\cdot\lambda_j=s$。计算：

$$CT_{part}=\frac{e(C_1,K_{tk})}{\prod_{j\in I}\left(e\left(C_{2,j}\cdot g^{a\cdot C_{4,j}},L_{tk}\right)\cdot e\left(C_{3,j},K_{\rho(j),tk}\right)\right)^{\omega_j}}=e(g,g)^{\alpha s/z}$$

将外包解密转换密文 $TC=\left(C,CT_{part},C_5\right)$ 发送给用户。

- 转换密文解密 $Decrypt_{out}\left(PK,Key,TC,VK_m\right)$：该阶段由授权用户 DU 执行，输入解密密钥 $Key=z$、来自解密云服务商的外包解密转换密文 TC、验证标识 VK_m，计算 $W=C/CT_{part}^{Key}$，$k=H_1(W)$。若 $VK_m\neq H_2\left(k\|C_5\right)$，则输出终止符 $\perp$；否则计算 $m=C_5\oplus k$，$s=H_0(W,m)$。若 $C=W\cdot e(g,g)^{\alpha s}$，$CT_{part}=e(g,g)^{\alpha s/z}$，输出 m；否则输出终止符 $\perp$。

b) 重加密密文解密算法 Dec_2

- 转换密文生成算法 $Transform_{out}\left(PK,DTK_{S'},C_{S\to(M',\rho')}\right)$：该阶段由解密云服务商 S-CSP 执行。若用户属性集 S' 不满足访问结构 $\mathbb{A}'=(M',\rho')$，则输出 $\perp$。若 $C_{S\to(M',\rho')}$ 与访问结构 $\mathbb{A}'$ 有关，被授权人外包密钥 $DTK_{S'}=\left(S,K'_{tk},L'_{tk},\{K'_{i,tk}\}_{i\in S}\right)$ 与用户属性集 S' 相关，且用户属性集 S' 满足访问结构 $\mathbb{A}'$，则定义 $I'\subset\{1,\cdots,l'\}$ 为 $I'=\{j:\rho'(j)\in S'\}$，$\{\lambda'_j\}$ 是根据矩阵 M' 对秘密 s' 的有效共享，存在一个常数集 $\{\omega'_j\in\mathbb{Z}_p\}_{j\in I'}$，使得 $\sum_{j\in I'}\omega_j\cdot\lambda_j=s'$。计算：

$$CT'_{part}=\frac{e\left(C'_1,K'_{tk}\right)}{\prod_{j\in I'}\left(e\left(C'_{2,j}\cdot g^{a\cdot C'_{4,j}},L'_{tk}\right)\cdot e\left(C'_{3,j},K'_{\rho'(j),tk}\right)\right)^{\omega'_j}}=e\left(g,g\right)^{\alpha s'/z'}$$

将 $TC'=\left(C,C',CT'_{part},C_5,C_7\right)$ 发送给用户。

- 转换密文解密 $Decrypt_{out}\left(PK,Key',TC',VK_m\right)$：该阶段由被授权用户 DU 执行，输入解密密钥 $Key'=z'$、来自解密云服务商的外包解密转换密文 TC'、验证标识 VK_m，计算 $\delta=C'/CT'^{\,Key'}_{part}$，$W=C/\left(C_7\right)^{H_1(\delta)}$，$k=H_1\left(W\right)$。若 $VK_m\neq H_2\left(k\|C_5\right)$，则输出终止符 $\perp$；否则计算 $m=C_5\oplus k$，$s=H_0\left(W,m\right)$。若 $C=W\cdot e\left(g,g\right)^{\alpha s}$，$CT'_{part}=e\left(g,g\right)^{\alpha s'/z'}$，输出 m；否则输出终止符 $\perp$。

4.2. 正确性

1) 原始密文的解密正确性。如果用户属性集 S 满足访问结构 $\mathcal{A}$，存在 $\sum_{j\in I}\omega_j\cdot\lambda_j=s$。则计算：

$$CT_{part}=\frac{e\left(C_1,K_{tk}\right)}{\prod_{j\in I}\left(e\left(C_{2,j}\cdot g^{a\cdot C_{4,j}},L_{tk}\right)\cdot e\left(C_{3,j},K_{\rho(j),tk}\right)\right)^{\omega_j}}=\frac{e\left(g^s,\left(g^{\alpha}g^{at}\right)^{1/z}\right)}{\prod_{j\in I}\left(e\left(g^{a\lambda_{j,1}}h^{-s_j}_{\rho(j)}\cdot g^{a\cdot(\lambda_j-\lambda_{j,1})},g^{t/z}\right)\cdot e\left(g^{s_j},h^{t/z}_{\rho(j)}\right)\right)^{\omega_j}}$$

$$=\frac{e\left(g^s,\left(g^{\alpha}g^{at}\right)^{1/z}\right)}{\prod_{j\in I}\left(e\left(g^{a\lambda_j}h^{-s_j}_{\rho(j)},g^{t/z}\right)\cdot e\left(g^{s_j},h^{t/z}_{\rho(j)}\right)\right)^{\omega_j}}=\frac{e\left(g,g\right)^{\alpha s/z}\cdot e\left(g,g\right)^{ats/z}}{e\left(g,g\right)^{at/z\sum_{j\in I}\lambda_j\omega_j}}=e\left(g,g\right)^{\alpha s/z}$$

接着计算 $\frac{C}{CT^{\,Key}_{part}}=\frac{W\cdot e\left(g,g\right)^{\alpha s}}{\left(e\left(g,g\right)^{\alpha s/z}\right)^z}=W$，$k=H_1\left(W\right)$，$k\oplus C_5=k\oplus k\oplus m=m$ 得到明文。

2) 重加密密文的解密正确性。如果用户属性集 S' 满足访问结构 $\mathcal{A}'=\left(M',\rho'\right)$，存在 $\sum_{j\in I'}\omega'_j\cdot\lambda'_j=s'$。先计算重加密密文组件 C_7，计算得：

$$C_7=\frac{e\left(C_1,rk_1\right)/e\left(C_1,rk_2\right)}{\prod_{j\in I}\left(e\left(C_{2,j}\cdot g^{a\cdot C_{4,j}},rk_3\right)\cdot e\left(C_{3,j},rk_j\right)\right)^{\omega_j}}=\frac{e\left(g^s,\left(g^{\alpha}g^{at}\right)^{H_1(\delta)}\cdot g_1^{\theta}\right)/e\left(g^s,g_1^{\theta}\right)}{\prod_{j\in I}\left(e\left(g^{a\lambda_{j,1}}h^{-s_j}_{\rho(j)}\cdot g^{a\cdot(\lambda_j-\lambda_{j,1})},\left(g^t\right)^{H_1(\delta)}\right)\cdot e\left(g^{s_j},\left(h^t_{\rho(j)}\right)^{H_1(\delta)}\right)\right)^{\omega_j}}$$

$$=\frac{e\left(g,g\right)^{s\alpha H_1(\delta)}e\left(g,g\right)^{sat H_1(\delta)}}{\prod_{j\in I}\left(e\left(g^{a\lambda_j}h^{-s_j}_{\rho(j)},\left(g^t\right)^{H_1(\delta)}\right)\cdot e\left(g^{s_j},\left(h^t_{\rho(j)}\right)^{H_1(\delta)}\right)\right)^{\omega_j}}=\frac{e\left(g,g\right)^{s\alpha H_1(\delta)}e\left(g,g\right)^{sat H_1(\delta)}}{e\left(g,g\right)^{at/H_1(\delta)\sum_{j\in I}\lambda_j\omega_j}}=e\left(g,g\right)^{s\alpha H_1(\delta)}$$

同样的，按照 CT_{part} 的计算方法，可以计算出 $CT'_{part}=e\left(g,g\right)^{\alpha s'/z'}$，综合以上结果，接着计算 $C'/CT'^{\,Key'}_{part}=\delta\cdot e\left(g,g\right)^{\alpha s'}/\left(e\left(g,g\right)^{\alpha s'/z'}\right)^{z'}=\delta$，$C/\left(C_7\right)^{H_1(\delta)^{-1}}=W\cdot e\left(g,g\right)^{\alpha s}/\left(e\left(g,g\right)^{s\alpha\cdot H_1(\delta)}\right)^{H_1(\delta)^{-1}}=W$，$k=H_1\left(W\right)$，$k\oplus C_5=k\oplus k\oplus m=m$ 得到明文。

5. VF-OO-ABPRE 方案分析

5.1. 安全性分析

定理 1. 如果判定性 q-parallel BDHE 假设成立，那么 VF-OO-ABPRE 方案是随机预言机模型下 s-CPA 安全的。

证明 假设存在某个敌手 $\mathcal{A}$ 能在概率多项式时间(PPT)内以不可忽略的优势攻破本文提出的方案，那么可以构造仿真者 $\mathcal{B}$ 在概率多项式时间(PPT)内以不可忽略的优势解决判定性 q-parallel BDHE 假设困难问题。

Init $\mathcal{A}$ 选择一个挑战的访问结构$\left(M^*,\rho^*\right)$发送给 $\mathcal{B}$。$\mathcal{B}$ 输入判定性 q-parallel BDHE 假设挑战元组$\left(p,\mathbb{G},\mathbb{G}_T,Z,\mathrm{Y}\right)$，其中，$Z$ 为群 $\mathbb{G}_T$ 中的随机元素 $R\in\mathbb{G}_T$ 或者 $e\left(g,g\right)^{a^{q+1}s}\in\mathbb{G}_T$，
$\boldsymbol{Y}=g,g^s,g^a,\cdots,g^{a^q},g^{a^{q+2}},\cdots,g^{a^{2q}}$，$\forall 1\le j\le q,g^{s\cdot b_j},g^{a/b_j},\cdots,g^{a^q/b_j},g^{a^{q+2}/b_j},\cdots,g^{a^{2q/b_j}}$，
$\forall 1\le j,k\le q,k\ne j,g^{a\cdot s\cdot b_k/b_j},\cdots,g^{a^q\cdot s\cdot b_k/b_j}$。

Setup $\mathcal{B}$ 运行系统初始化算法 $Setup\left(1^\lambda,U\right)\to\left(PK,MSK\right)$，随机选择 $\alpha'\in\mathbb{Z}_p$，使得 $\alpha=\alpha'+a^{q+1}$。在此阶段 $\mathcal{B}$ 不知道 α 的值，但是可计算 $e\left(g,g\right)^\alpha=e\left(g^a,g^{a^q}\right)e(g,g)^{\alpha'}$。对于任意 $x\in U$，随机选取 $z_x\in\mathbb{Z}_p$，当 $\rho^*\left(i\right)=x$，$\mathcal{B}$ 计算 $h_x=g^{z_x}\cdot g^{a\boldsymbol{M}_{i,1}^*/b_i}\cdot g^{a^2\boldsymbol{M}_{i,2}^*/b_i}\cdots g^{a^{n^*}\boldsymbol{M}_{i,n^*}^*/b_i}$，否则计算 $h_x=g^{z_x}$。输出 $PK=\left(p,g,e\left(g,g\right)^\alpha,g^a,g_1,h_1,\cdots,h_n,H_0,H_1,H_2\right)$ 给 $\mathcal{A}$。

Phase 1 $\mathcal{B}$ 初始化空表 $\mathcal{T}_0$，$\mathcal{T}_1$，$\mathcal{T}_2$，$\mathcal{A}$ 可重复进行以下查询：

1) $H_0\left(W,m\right)$：若表 $\mathcal{T}_0$ 中已经存在$\left(W,m,s\right)$，则返回 s；否则选取一个随机值 $s\in\mathbb{Z}_p$，并将$\left(W,m,s\right)$记录在表 $\mathcal{T}_0$ 中，返回 s。

2) $H_1\left(W\right)$：若表 $\mathcal{T}_1$ 中已经存在$\left(W,k\right)$，则返回 k；否则选取一个随机值 $k\in\left\{0,1\right\}^\lambda$，并将$\left(W,k\right)$记录在表 $\mathcal{T}_1$ 中，返回 k。

3) 私钥查询 $\mathcal{O}_{sk}\left(S\right)$：$\mathcal{A}$ 可重复发出查询请求。输入属性集 S，$\mathcal{B}$ 做出如下回应：

a) 属性集合 $S\models\left(M^*,\rho^*\right)$ 时，$\mathcal{B}$ 选择向量 $w=\left(w_1,w_2,\cdots,w_{n^*}\right)\in\mathbb{Z}_p^{n^*}$，其中 $w_1=-1$，对于所有的 i 满足 $\rho^*\left(i\right)\in S$ 时，$M_i^*\cdot w=0$。随机选取 $r\in\mathbb{Z}_p$，定义 $t=r+w_1a^q+w_2a^{q-1}+\cdots+w_{n^*}a^{q-n^*+1}$。计算 $L=g^r\prod_{i=1,\cdots,n^*}\left(g^{a^{q+1-i}}\right)^{w_i}=g^t$，$K=g^{\alpha'}g^{ar}\prod_{i=2,\cdots,n^*}\left(g^{a^{q+2-i}}\right)^{w_i}=g^\alpha g^{at}$。对于 $\rho^*\left(i\right)=x$，计算

$K_x=L^{z_x}\prod_{j=1,\cdots,n^*}\left(g^{\left(a^j/b_i\right)\cdot r}\prod_{\substack{k=1,\cdots,n^*\\k\ne j}}\left(g^{a^{q+1+j-k/b_i}}\right)^{w_k}\right)^{\boldsymbol{M}_{i,j}^*}$，其余情况则计算 $K_x=L^{z_x}=h_x^t$。用户私钥 $SK_S=\left(S,K,L,\left\{K_x\right\}_{x\in S}\right)$。随机选取 $z\in\mathbb{Z}_p$ 作为解密密钥。计算外包密钥得 $DTK_S=\left(K^{(1/z)},L^{(1/z)},K_x^{(1/z)}\right)$。将元组$\left(S,SK_S,DTK_S\right)$记录在表 $\mathcal{T}_2$ 中，发送 DTK_S 给 $\mathcal{A}$。

b) 属性集合 $S\models\left(M^*,\rho^*\right)$ 时，无法查询 S 对应的私钥。按如下方法选择一个“伪”外包密钥：$\mathcal{B}$ 随机选取 $d\in\mathbb{Z}_p^*$，运行 $KeyGen\left(PK,d,S\right)$ 生成密钥 SK_S'，令 $DTK_S=SK_S'$，$SK_S=\left(d,DTK_S\right)$，将元组$\left(S,SK_S,DTK_S\right)$记录在表 $\mathcal{T}_2$ 中，发送 DTK_S 给 $\mathcal{A}$。

4) 重加密密钥查询 $\mathcal{O}_{rk}\left(S,\left(M',\rho'\right)\right)$：$\mathcal{A}$ 输入属性集 S 和访问结构$\left(M',\rho'\right)$，若 $S\models\left(M^*,\rho^*\right)$ 时，$\mathcal{B}$ 运行重加密密钥生成算法 $ReKeyGen\left(PK,S,SK_S,\left(M',\rho'\right)\right)\to RK_{S\to\left(M',\rho'\right)}$ 返回 $RK_{S\to\left(M',\rho'\right)}$ 给 $\mathcal{A}$。其中 SK_S 由算法 $KeyGen\left(PK,MSK,S\right)\to SK_S$ 生成。

Challenge $\mathcal{A}$ 向 $\mathcal{B}$ 输出两个相同长度的消息 m_0 和 m_1，$\mathcal{B}$ 执行如下计算：

1) 随机选择 $W\in\mathbb{G}_T$，$b\in\left\{0,1\right\}$，$k^*\in\mathbb{Z}_p$，$C_6^*\in\mathbb{G}$ 计算得到部分密文 $C_b^{part}=\left(C^*,C_1^*,C_5^*,C_6^*\right)$ 得：
$C^*=W\cdot Z\cdot e\left(g^{\alpha'},g^s\right)$，$C_1^*=g^s$，$C_5^*=k^*\oplus m_b$，$C_6^*=C_6^*$；

2) 随机选择 $c_2,c_3,\cdots,c_{n^*}\in\mathbb{Z}_p$，计算 $v=\left(s,sa+c_2,sa^2+c_3,\cdots,sa^{n^*-1}+c_{n^*}\right)$，对于 $i\in\left\{1,\cdots,l^*\right\}$，$M_i^*$ 表示矩阵 M^* 的第 i 行，$\lambda_i^*=\left\{M_i^*v^{\mathrm{T}}\right\}_{i\in I}$，随机选择 $s_i^*\in\mathbb{Z}_p$，对于 $i=\left\{1,\cdots,l^*\right\}$，定义 R_i 表示 $k\ne i$ 且 $\rho^*\left(i\right)=\rho^*\left(k\right)$ 时的集合。计算：

$$C_{2,i}^*=h_{\rho^*(i)}^{s_i^*}\left(\prod_{j=2,\cdots,n^*}\left(g^a\right)^{\boldsymbol{M}_{i,j}^*c_j}\right)\left(g^{b_i\cdot s}\right)^{-z_{\rho^*(i)}}\cdot\left(\prod_{k\in R_i}\prod_{j=1,\cdots,n^*}\left(g^{a^j\cdot s\cdot\left(b_i/b_k\right)}\right)^{\boldsymbol{M}_{k,j}^*}\right),\quad C_{3,i}^*=g^{-s_i^*}g^{-sb_i}$$

3) $\mathcal{B}$ 将挑战密文 $C_b=\left(C^*,C_1^*,\left\{C_{2,i}^*,C_{3,i}^*,C_{4,i}^*\right\}_{i\in\{1,\cdots,l^*\}},C_5^*,C_6^*\right)$ 发送给 $\mathcal{A}$。

Phase 2 $\mathcal{A}$ 继续查询阶段 1 中的查询，但是不能查询任何属性集 S 满足 $\left(M^*,\rho^*\right)$。

Guess $\mathcal{A}$ 输出一个猜测 $b'\in\{0,1\}$。若仿真者 $\mathcal{B}$ 认为 $b'=b$，则猜测 $Z=e(g,g)^{a^{q+1}s}$，否则，猜测 Z 是群 $\mathbb{G}_T$ 的随机元素。如果 $Z=e(g,g)^{a^{q+1}s}$，仿真者 $\mathcal{B}$ 赢得该游戏的概率是：$Pr\left[\mathcal{B}\left(\boldsymbol{Y},Z=e(g,g)^{a^{q+1}s}\right)=1\right]=\frac{1}{2}+\varepsilon$。如果 Z 是群 $\mathbb{G}_T$ 的随机元素，消息 m_b 对敌手来说是隐藏的，仿真者 $\mathcal{B}$ 赢得该游戏的概率是：$Pr\left[\mathcal{B}\left(\boldsymbol{Y},Z=R\right)=1\right)=\frac{1}{2}$。因此，挑战者在解决判定性 q-parallel BDHE 假设问题上具有不容忽视的优势：$Adv_{\mathcal{B}}^{q-BDHE}=Pr\left[\mathcal{B}\left(\boldsymbol{Y},Z=e(g,g)^{a^{q+1}s}\right)=1\right]-Pr\left[\mathcal{B}\left(\boldsymbol{Y},Z=R\right)=1\right]=\varepsilon$。

这与判定性 q-parallel BDHE 假设成立矛盾，所以定理 1 得证。**证毕**

定理 2. 假设 H_1,H_2 是抗碰撞的杂凑函数，那么 VF-OO-ABPRE 方案具有可验证性。

证明 假设存在敌手 $\mathcal{A}$ 能在概率多项式时间(PPT)内以不可忽略的优势攻破可验证性，那么可以构建一个仿真者 $\mathcal{B}$ 打破底层杂凑函数 H_1,H_2 的抗碰撞能力。$\mathcal{A}$ 提交 2 个挑战杂凑函数 H_1^*,H_2^*，$\mathcal{B}$ 进行如下的仿真实验过程：

Setup $\mathcal{B}$ 执行 *Setup* 算法获取系统公钥 PK 和主密钥 MSK，并用 H_1^*,H_2^* 替换 PK 中的 H_1,H_2。注：$\mathcal{B}$ 知道主密钥 MSK。

Phase 1 $\mathcal{A}$ 可以重复向预言机进行如下查询：

1) 私钥查询 $\mathcal{O}_{sk}(S)$：$\mathcal{A}$ 提交一个属性集合 S，该属性集合 S 不满足挑战访问结构 $\left(M^*,\rho^*\right)$，$\mathcal{C}$ 运行私钥生成算法 $KeyGen\left(PK,MSK,S\right)$ 和外包密钥生成算法 $KeyGen_{out}\left(S,SK_S\right)$，并返回私钥 SK_S 和外包密钥 DTK_S 给 $\mathcal{A}$。

2) 重加密密钥查询 $\mathcal{O}_{rk}\left(S,\left(M',\rho'\right)\right)$：$\mathcal{A}$ 提交一个属性集合 S，该属性集合 S 不满足挑战访问结构 $\left(M^*,\rho^*\right)$，满足访问结构 $\left(M',\rho'\right)$，$\mathcal{C}$ 运行重加密密钥生成算法 $ReKeyGen\left(PK,S,SK_S,\left(M',\rho'\right)\right)$ 返回 $RK_{S\to(M',\rho')}$ 给 $\mathcal{A}$。其中 SK_S 由算法 $KeyGen\left(PK,MSK,S\right)$ 生成。

Challenge $\mathcal{A}$ 提交两个相同长度的消息 m_0 和 m_1 以及一个挑战的访问结构 $\left(M^*,\rho^*\right)$ 给 $\mathcal{B}$。$\mathcal{B}$ 随机选择 $b\in\{0,1\}$，$W^*\in\mathbb{G}_T$，并计算 $CT^{W^*}=\left(C,C_1,\left\{C_{2,i},C_{3,i},C_{4,i}\right\}_{1\le i\le l}\right)$，$k^*=H_1^*\left(W^*\right)$，$C_5^*=k^*\oplus m_b$，$VK_{m_b}^*=H_2^*\left(k^*\middle\|C_5^*\right)$。$\mathcal{B}$ 将 $C_{\left(M^*,\rho^*\right)}^*=\left(CT^{W^*},C_5^*\right)$ 和 $VK_{m_b}^*$ 发送给 $\mathcal{A}$。同时自己保留 $VK_{m_b}^*$ 和 $\left(W^*,C_5^*\right)$。

Phase 2 $\mathcal{B}$ 按照查询阶段 1 的方式响应 $\mathcal{A}$ 的询问。但是 $\mathcal{A}$ 不能询问满足访问结构 $\left(M^*,\rho^*\right)$ 的属性集合 S。

Guess $\mathcal{A}$ 输出属性集合 S^* 和外包解密转换密文 $TC^*=\left(C,CT_{part},C_5\right)$。

若敌手 $\mathcal{A}$ 可攻破可验证性，那么仿真者 $\mathcal{B}$ 将通过转换密文解密 $Decrypt_{out}\left(PK,Key^*,TC^*,VK_{m_b}^*\right)$ 恢复出明文 m_b。若 $VK_{m_b}^*\ne H_2^*\left(k\middle\|C_5^*\right)$，则解密算法输出终止符 $\perp$，其中 $k=H_1^*\left(W\right)$，$W=C/CT_{part}^{\ Key'}$。考虑如下情况：

情况 1：$\left(k,C_5\right)\ne\left(k^*,C_5^*\right)$。因仿真者 $\mathcal{B}$ 知道 $\left(k^*,C_5^*\right)$，若出现此种情况，则 $\mathcal{B}$ 可得到杂凑函数 H_2^* 的碰撞。

情况 2：$\left(k,C_5\right)=\left(k^*,C_5^*\right)$。但是 $W\ne W^*$，因为 $H_1^*\left(W\right)=k=k^*=H_1^*\left(W^*\right)$，若出现此种情况，则将打破杂凑函数 H_1^* 的抗碰撞能力。

通过上述分析完成定理 2 的安全证明。**证毕**

定理 3. 假定 CDH 假设成立，那么 VF-OO-ABPRE 方案是抗共谋攻击的。

证明 假设存在某个敌手 $\mathcal{A}$ 能在概率多项式时间(PPT)内以不可忽略的优势攻破本文提出的方案，那么可以构造仿真者 $\mathcal{B}$ 在概率多项式时间(PPT)内以不可忽略的优势解决 CDH 困难问题。

Setup $\mathcal{B}$ 运行 $Setup\left(1^{\lambda},U\right)$ 算法，随机选择 $h_1,\cdots,h_{|U|}\in\mathbb{G}$， $\alpha,a,b\in\mathbb{Z}_p$，计算 $e(g,g)^{\alpha}$ 和 $g_1=g^b$，得到系统公钥 $PK=\left(p,g,e(g,g)^{\alpha},g^a,g_1,h_1,\cdots,h_{|U|},H_0,H_1,H_2\right)$ 和主密钥 $MSK=g^{\alpha}$。

Phase 1 对于不满足访问结构 $\left(M^*,\rho^*\right)$ 但是满足访问结构 (M',ρ') 的属性集合 $S'\subseteq U$， $\mathcal{B}$ 运行 $KeyGen\left(PK,MSK,S'\right)\to SK_{S'}$ 生成 $SK_{S'}=\left(S',K',L',\left\{K_i'\right\}_{i\in S'}\right)$ 将其发送给 $\mathcal{A}$。对于满足访问结构 $\left(M^*,\rho^*\right)$ 的属性集合 $S^*\subseteq U$，$\mathcal{B}$ 运行 $KeyGen\left(PK,MSK,S^*\right)\to SK_{S^*}$ 生成 $SK_{S^*}=\left(S^*,K^*,L^*,\left\{K_i^*\right\}_{i\in S^*}\right)$ 由 $\mathcal{B}$ 自己秘密保存。对于关于 $\left(S^*,(M',\rho')\right)$ 的重加密密钥询问，如果 S^* 从未被询问过，通过访问结构 (M',ρ')， $\mathcal{B}$ 随机选择 $\theta^*\in\mathbb{Z}_p$，计算 $g^{\theta^*}=\delta^*\in\mathbb{G}$，对于 $j\in\{1,\cdots,l'\}$，计算 $C'=\delta^*\cdot e(g,g)^{\alpha s^*}$， $C_1'=g^{s'}$， $C_{2,j}'=g^{a\lambda_{j,1}'}h_{\rho(j)}^{-s_j'}$，$C_{3,j}'=g^{s_j'}$， $C_{4,j}'=\lambda_j'-\lambda_{j,1}'$，得到 $C_{(M',\rho')}=\left(C',C_1',\left\{C_{2,j}',C_{3,j}',C_{4,j}'\right\}_{j\in\{1,\cdots,l'\}}\right)$， $rk_1=\left(K^*\right)^{H_1\left(\delta^*\right)}\cdot g_1^{\theta^*}$， $rk_2=g_1^{\theta^*}$，$rk_3=\left(L^*\right)^{H_1\left(\delta^*\right)}$，对于所有的 $j\in\{1,\cdots,l'\}$， $rk_4=C_{(M',\rho')}$， $rk_j=\left(K_j^*\right)^{H_1\left(\delta^*\right)}$，最后输出重加密密钥 $RK_{S^*\to C_{(M',\rho')}}=\left(S,rk_1,rk_2,rk_3,rk_4,rk_j\right)$ 和私钥 $SK_{S'}$ 发送给 $\mathcal{A}$。

Challenge 最后， $\mathcal{A}$ 提交一个密钥 $SK_{S'}$，如果 $SK_{S''}=SK_{S^*}$ 则 $\mathcal{A}$ 赢得游戏。

注：$\left(M^*,\rho^*\right)$ 与 (M',ρ') 是不相交的， $C_{(M',\rho')}$ 实际上是 δ^* 在访问结构 $\mathbb{A}'=(M',\rho')$ 下 Waters 等[4]方案的 ABE 加密， $\mathcal{A}$ 可以通过对应的私钥 $SK_{S'}$ 恢复得到 $\delta^*=g^{\theta^*}$，为了从重加密密钥 $RK_{S^*\to(M',\rho')}$ 中恢复 SK_{S^*}， $\mathcal{A}$ 还需计算出 $g_1^{\theta^*}$，已知 g^{θ^*}， $g_1=g^b$，计算 $g_1^{\theta^*}=g^{b\theta^*}$ 相当于解决 CDH 困难问题。**证毕**

5.2. 效率分析

本方案将与文献[5]中外包解密 CP-ABE 方案和文献[13]可验证外包 CP-ABE 方案，文献[15]和文献[21] [22] [23] CP-ABPRE 方案进行具体的性能分析。B 表示双线性运算，E 表示 $\mathbb{G}$ 群的指数运算，E_T 表示 $\mathbb{G}_T$ 群的指数运算，l 表示访问结构 (M,ρ) 中矩阵 M 的行数，s 表示用户私钥属性数量，N 表示系统属性空间属性数量。$|\mathbb{G}|$ 表示 $\mathbb{G}$ 中的元素长度， $|\mathbb{G}_T|$ 表示 $\mathbb{G}_T$ 中的元素长度。忽略实际运算中涉及的杂凑函数及异或运算的计算开销和存储开销。

表 1 展示了几种方案用户端的计算开销对比，表 2 展示了几种方案的私钥与密文存储空间占用对比。本文基于文献[5]提出 VF-OO-ABPRE 方案，实现了离线/在线加密且外包解密可验证的功能，减少了用户端实际加密过程中的计算开销和带宽。在原始文献[5]的基础上结合代理重加密，实现了密文共享功能，原始密文加密计算只消耗了 $4E_T$，虽然原始密文解密计算多了 $2E_T$，但是实现了可验证的功能；文献[13]可验证的外包解密的离线/在线 ABE 加密方案，虽然用户端在线加密计算只消耗了 $2E$，但是密文解密阶段的计算量较大的双线性对的运算 B 消耗较多，且计算开销随属性集合的数量呈线性增长的关系。而本方案将解密计算外包，所以用户端的解密计算消耗 $4E_T$；文献[15]仅支持“AND”门访问结构，表达能力有限。本方案采取离线/在线加密操作以及解密外包，将复杂的双线性运算外包给 D-CSP，使得解密只需进行 4 个 $\mathbb{G}_T$ 群的指数运算，重加密密钥的生成也相对应地节省了在 $\mathbb{G}$ 群上的指数计算开销。大大提高了计算效率，节省了带宽。对比文献[21] [22] [23]不同功能的属性代理重加密方案，计算开销与所占带宽的优势明显。

综合以上分析，本方案实现了离线/在线加密操作的同时，将解密外包且支持外包可验证性。离线/在线的加密操作适用于使用资源受限设备加密的用户，可以在设备插入电源时离线完成大部分加密过程，实际加密时，只需以较小的能耗就能迅速完成在线加密。理论分析表明，本方案非常适用于资源受限的移动设备。

Table 1. Comparison of user client computing overhead
表 1. 用户客户端计算开销对比

方案	原始密文加密	生成重加密密钥	原始密文解密	重加密密文解密	可验证性	访问结构
文献[5]	$(1+2l)E+E_T$	——	E_T	——	NO	LSSS
文献[13]	$2E$	——	$(2+3s)E+(2+3s)B$	——	YES	LSSS
文献[15]	$(2+3N)E+E_T$	$(2+3N)E+E_T$	$(3+N)B$	$E_T+(3+N)B$	NO	AND 门
文献[21]	$(6+4N)E+E_T$	$(8+4N)E+2E_T$	$(6+3N)B$	$E_T+(7+3N)B$	NO	AND 门
文献[22]	$(2+12N)E+E_T$	$(2+25N)E+E_T$	$(2+4N)B$	$(3+4N)B$	NO	AND 门
文献[23]	$(2+2N)E+E_T$	$(3+4N)E$	——	$E_T+(1+2N)B$	NO	AND 门
本方案	E_T	$(2+3l)E+E_T$	$3E_T$	$3E_T$	YES	LSSS

Table 2. Comparison of private key and ciphertext storage overhead
表 2. 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[5]	$(2+s)\lvert\mathbb{G}\rvert$	$\lvert\mathbb{G}_T\rvert+(1+2l)\lvert\mathbb{G}\rvert$	——
文献[13]	$(2+2s)\lvert\mathbb{G}\rvert$	$\lvert\mathbb{G}_T\rvert+(2+3l)\lvert\mathbb{G}\rvert$	——
文献[15]	$2N\lvert\mathbb{G}\rvert$	$\lvert\mathbb{G}_T\rvert+2N\lvert\mathbb{G}\rvert$	$5\lvert\mathbb{G}_T\rvert+(4+N)\lvert\mathbb{G}\rvert$
文献[21]	$(2+4N)\lvert\mathbb{G}\rvert$	$2\lvert\mathbb{G}_T\rvert+(2+12N)\lvert\mathbb{G}\rvert$	$3\lvert\mathbb{G}_T\rvert+(4+12N)\lvert\mathbb{G}\rvert$
文献[22]	$(7+3N)\lvert\mathbb{G}\rvert$	$2\lvert\mathbb{G}_T\rvert+(5+3N)\lvert\mathbb{G}\rvert$	$4\lvert\mathbb{G}_T\rvert+(6+3N)\lvert\mathbb{G}\rvert$
文献[23]	$N\lvert\mathbb{G}_T\rvert+\lvert\mathbb{G}\rvert$	$(1+N)\lvert\mathbb{G}_T\rvert+(2+N)\lvert\mathbb{G}\rvert$	$(3+N)\lvert\mathbb{G}_T\rvert+(1+N)\lvert\mathbb{G}\rvert$
本方案	$(2+s)\lvert\mathbb{G}\rvert$	$\lvert\mathbb{G}_T\rvert+(2+2l)\lvert\mathbb{G}\rvert$	$3\lvert\mathbb{G}_T\rvert+(2+2l)\lvert\mathbb{G}\rvert$

6. 总结

本文针对 CP-ABPRE 加密的性能问题，提出新的离线/在线加密且可验证外包解密的 VF-OO-ABPRE 方案，主要将加密算法分为两个阶段，离线加密和在线加密。用户可以在插入电源时首先离线完成大部分的加密预处理计算，在线加密则可以利用离线加密的计算结果，快速高效地完成最终加密，有效地提高了 CP-ABPRE 案的加密性能。将解密外包，并且能验证外包计算的正确性，同时抗合谋攻击，提高了方案的安全性。有效缓解了资源受限用户的加解密负担，同时，本方案在随机预言机模型下被证明是选择明文攻击的不可区分安全的。通过对比已有方案的计算开销和存储开销，本方案在功能性和效率方面均具有明显优势，具有现实的应用价值。

基金项目

国家自然科学基金(No. U1836205, 61662009, 61772008)，贵州省科技计划项目(No. 黔科合重大专项字[2018] 3001，黔科合重大专项字[2018] 3007，黔科合重大专项字[2017] 3002，黔科合支撑[2019] 2004，黔科合支撑[2018] 2162，黔科合支撑[2018] 2159，黔科合基础[2019] 1049，黔科合基础[2017] 1045)，“十三五”国家密码发展基金(No. MMJJ20170129)。

参考文献

[1] Sahai, A. and Waters, B. (2005) Fuzzy Identity-Based Encryption. In: *LNCS* 3494: *EUROCRYPT*'05, Springer, Berlin, 457-473. https://doi.org/10.1007/11426639_27

[2] Bethencourt, J., Sahai, A. and Waters, B. (2007) Ciphertext-Policy Attribute-Based Encryption. *IEEE Symposium on Security and Privacy* (*SP*'07), Berkeley, 20-23 May 2007, 321-334. https://doi.org/10.1109/SP.2007.11

[3] Goyal, V., Pandey, O., Sahai, A. and Waters, B. (2006) Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data. In: *Proceedings of the* 13*th ACM Conference on Computer and Communications Security* (*CCS*'06), Association for Computing Machinery, New York, 89-98. https://doi.org/10.1145/1180405.1180418

[4] Waters, B. (2011) Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization. In: *Public Key Cryptography PKC* 2011, Springer, Berlin, Vol. 6571, 53-70. https://doi.org/10.1007/978-3-642-19379-8_4

[5] Green, M., Hohenberger, S. and Waters, B. (2011) Outsourcing the Decryption of ABE Ciphertexts. *Proceedings of the* 20*th USENIX Conference on Security* (*SEC*'11), San Francisco, 8-12 August 2011, 34.

[6] Lai, J., Deng, R.H., Guan, C., *et al*. (2013) Attribute-Based Encryption with Verifiable Outsourced Decryption. *IEEE Transactions on Information Forensics & Security*, **8**, 1343-1354. https://doi.org/10.1109/TIFS.2013.2271848

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

[8] Zhang, J., Wang, B., *et al*. (2018) Energy-Efficient Secure Outsourcing Decryption of Attribute Based Encryption for Mobile Device in Cloud Computation. *Journal of Ambient Intelligence and Humanized Computing*, **10**, 429-438. https://doi.org/10.1007/s12652-017-0658-2

[9] Liao, Y., He, Y., Li, F., *et al*. (2018) Analysis of an ABE Scheme with Verifiable Outsourced Decryption. *Sensors*, **18**, 176. https://doi.org/10.3390/s18010176

[10] Li, J., Jia, C., Li, J. and Chen, X. (2012) Outsourcing Encryption of Attribute-Based Encryption with MapReduce. In: Chim, T.W. and Yuen, T.H., Eds., *Information and Communications Security*. *ICICS* 2012, Lecture Notes in Computer Science, Vol. 7618, Springer, Berlin, 191-201. https://doi.org/10.1007/978-3-642-34129-8_17

[11] Wang, H., He, D., Shen, J., *et al*. (2017) Verifiable Outsourced Ciphertext-Policy Attribute-Based Encryption in Cloud Computing. *Soft Computing*, **21**, 7325-7335. https://doi.org/10.1007/s00500-016-2271-2

[12] Hohenberger, S. and Waters, B. (2014) Online/Offline Attribute-Based Encryption. In: Krawczyk, H., Eds., *Public-Key Cryptography—PKC* 2014, Lecture Notes in Computer Science, Springer, Berlin, Vol. 8383, 293-310. https://doi.org/10.1007/978-3-642-54631-0_17

[13] Liu, Z., Jiang, Z.L., Wang, X., Huang, X., Yiu, S.M. and Sadakane, K. (2017) Offline/Online Attribute-Based Encryption with Verifiable Outsourced Decryption. *Concurrency and Computation*: *Practice and Experience*, **29**, e3915. https://doi.org/10.1002/cpe.3915

[14] Blaze, M., Bleumer, G. and Strauss, M. (1998) Divertible Protocols and Atomic Proxy Cryptography. In: Nyberg, K., Ed., *Advances in Cryptology—EUROCRYPT*'98, Lecture Notes in Computer Science, Springer, Berlin, Vol. 1403, 127-144. https://doi.org/10.1007/BFb0054122

[15] Liang, X.H., Cao, Z.F., Lin, H. and Shao, J. (2009) Attribute Based Proxy Re-Encryption with Delegating Capabilities. In: *Proceedings of the* 4*th International Symposium on Information, Computer, and Communications Security* (*ASIACCS*'09), Association for Computing Machinery, New York, 276-286. https://doi.org/10.1145/1533057.1533094

[16] Liang, K.T., Fang, L.M., Susilo, W., *et al*. (2013) A Ciphertext-Policy Attribute-Based Proxy Re-Encryption with Chosen-Ciphertext Security. *The IEEE* 5*th International Conference on Intelligent Networking and Collaborative Systems*, Xi'an, 9-11 September 2013, 552-559. https://doi.org/10.1109/INCoS.2013.103

[17] Liang, K.T., Au, M.H., Susilo, W., *et al*. (2014) An Adaptively CCA-Secure Ciphertext-Policy Attribute-Based Proxy Re-Encryption for Cloud Data Sharing. In: *International Conference on Information Security Practice and Experience*, Springer, Cham, 448-461. https://doi.org/10.1007/978-3-319-06320-1_33

[18] Liang, K.T., Man, H.A., Liu, J., *et al*. (2015) A Secure and Efficient Ciphertext-Policy Attribute-Based Proxy Re-Encryption for Cloud Data Sharing. *Future Generation Computer Systems*, **52**, 95-108. https://doi.org/10.1016/j.future.2014.11.016

[19] Gritti, C., Susilo, W., Plantard, T., Liang, K. and Wong, D.S. (2014) Empowering Personal Health Records with Cloud Computing: How to Encrypt with Forthcoming Fine-Grained Policies Efficiently. *Journal of Wireless Mobile Networks Ubiquitous Computing & Dependable Applications*, **4**, 3-28.

[20] Kawai, Y. (2015) Outsourcing the Re-Encryption Key Generation: Flexible Ciphertext-Policy Attribute-Based Proxy Re-Encryption. In: Lopez, J. and Wu, Y., Eds., *Information Security Practice and Experience*, *ISPEC* 2015, Lecture

Notes in Computer Science, Springer, Cham, Vol. 9065, 301-315. https://doi.org/10.1007/978-3-319-17533-1_21

[21] Sepehri, M. and Trombetta, A. (2017) Secure and Efficient Data Sharing with Atribute-Based Proxy Re-Encryption Scheme. In: *Proceedings of the* 12*th International Conference on Availability*, *Reliability and Security* (*ARES*'17), Association for Computing Machinery, New York, Article 63, 1-6. https://doi.org/10.1145/3098954.3104049

[22] Yin, H. and Zhang, L. (2017) Security Analysis and Improvement of an Anonymous Attribute-Based Proxy Re-Encryption. In: *Security*, *Privacy*, *and Anonymity in Computation*, *Communication*, *and Storage*, *Spa CCS* 2017, Lecture Notes in Computer Science, Springer, Cham, Vol. 10656, 344-352. https://doi.org/10.1007/978-3-319-72389-1_28

[23] Hong, H. and Sun, Z. (2018) Sharing Your Privileges Securely: A Key-Insulated Attribute Based Proxy Re-Encryption Scheme for IoT. *World Wide Web*, **21**, 595-607. https://doi.org/10.1007/s11280-017-0475-8