

云计算多授权中心 CP-ABE 代理重加密方案

刘尚, 郭银章

(太原科技大学计算机科学与技术学院, 山西 太原 030024)

摘要: 代理重加密技术可使代理在不知道明文的情况下实现密文访问策略转换, 这使代理重加密成为用户之间进行数据分享的重要技术。然而, 代理重加密方案大多数是在单授权中心下构建的, 存在授权机构权限大、易出现性能瓶颈和用户的计算开销大等问题。同时, 大多数方案不满足代理重加密应具备的 5 个基本特性: 单向性、可控性、非交互性、可重复性与可验证性。为解决以上问题, 提出支持重复可控特性的云计算多授权中心 CP-ABE (ciphertext-policy attribute-based encryption) 代理重加密方案。在密文策略属性加密方案的基础上, 引入代理加密和代理解密服务器从而减小用户客户端的计算开销, 设置多个属性授权中心来分散中央机构权限。对代理重加密技术进行改进: 在重加密密钥中设置随机因子和密文子项来实现单向性和可控性; 设置的重加密密钥由客户端独立生成, 不需要其他服务器参与, 可实现非交互性, 即可在数据拥有者为不在线状态时也可以进行数据分享; 在初始密文中设置密文子项, 对其多次加密即可实现重复性; 在初始密文中设置验证子项, 用户可验证外包以及重加密结果正确与否。通过与其他方案对比发现, 所提方案的用户客户端计算开销较小, 用户只需进行常数次的指数运算即可对原始密文解密, 且安全性分析表明, 所提方案基于 q -parallel BDHE 假设, 在标准模型下可抵抗选择密文攻击。

关键词: 重复性; 可控性; 云计算; 多授权中心; 代理重加密

中图分类号: TP309

文献标志码: A

DOI: 10.11959/j.issn.2096-109x.2022033

Multi-authority based CP-ABE proxy re-encryption scheme for cloud computing

LIU Shang, GUO Yinzhong

School of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan 030024, China

Abstract: Proxy re-encryption allows a proxy to convert a ciphertext related to a certain access policy into another one with a different access policy without uncovering the underlying plaintext, which makes the proxy re-encryption technology to be an important technology for data sharing between users. However, the proxy re-encryption schemes are mostly based on single authority, which have some problems such as single power and performance bottleneck

收稿日期: 2021-09-30; 修回日期: 2021-11-01

通信作者: 郭银章, guoyinzhong@263.net

基金项目: 山西省应用基础研究项目 (201901D111266)

Foundation Item: Applied Basic Research Project of Shanxi Province (201901D111266)

引用格式: 刘尚, 郭银章. 云计算多授权中心 CP-ABE 代理重加密方案[J]. 网络与信息安全学报, 2022, 8(3): 176-188.

Citation Format: LIU S, GUO Y Z. Multi-authority based CP-ABE proxy re-encryption scheme for cloud computing[J]. Chinese Journal of Network and Information Security, 2022, 8(3): 176-188.

of the authority organization and high computing burden on user's client. At the same time, most schemes do not have the five basic characteristics of proxy re-encryption: one-way encryption, non-interaction, repeatability, controllability and verifiability. In order to solve these problems, a CP-ABE proxy re-encryption scheme for cloud computing based on multi-authority with repeatability and controllability was proposed. Based on the ciphertext policy attribute encryption scheme, proxy servers of encryption and decryption were introduced to reduce the computing burden on the user's client, and multi-attribute authorization were set to disperse the authority of the central authority. The proxy re-encryption was improved from multiple aspects. Random factors and ciphertext sub-item set in the re-encryption key can realize one-way encryption and controllability. The re-encryption key was independently generated by the client without the participation of servers, which can support users to share data when the data owner is not online. Encrypting the ciphertext sub-item set in the initial ciphertext many times can achieve repeatability. The verification sub-item set in the initial ciphertext allowed the user to verify whether the outsourcing and re-encryption results were correct. Compared with other schemes, the user's client computing overhead of the proposed scheme was small, and the user can decrypt the original ciphertext only by performing constant times of exponential operation. Based on q -parallel BDHE assumption, the security analysis showed that the proposed scheme is secure against chosen-ciphertext attack (CCA) under the standard model.

Keywords: repeatability, controllability, cloud computing, multi-authority, proxy re-encryption

0 引言

随着云计算服务领域的不断扩大,越来越多的企业和个人把数据外包存储在云端。但云服务提供商(CSP, cloud service provider)是半可信的,人们对存储的数据尤其是隐私数据的安全性和完整性未知^[1],失去了对数据的掌控^[2]。而密文访问控制技术可以解决数据的安全访问问题,采用密文策略基于属性加密(CP-ABE, ciphertext-policy attribute-based encryption)算法,实现了细粒度的访问控制。

但是,现有CP-ABE仍然存在加解密开销大、访问策略的变化导致重加密效率不高以及分享数据文件不便等问题。因此引入代理重加密技术实现数据安全高效的分享。1998年,Blaze等^[3]提出代理重加密的概念,其允许一个半可信代理服务器(PS, proxy sever)在不解密出原文的情况下,将授权人(delegatee) Alice 公钥加密的密文转换为被授权人(delegatee) Bob 公钥加密的密文,在解密过程中PS无法获取明文的任何信息。

CP-ABE方案中的数据属于主用访问结构加密数据,无须了解用户的身份,嵌入在用户私钥中的属性与数据属主设置的访问结构匹配时,用户可对数据进行访问,这是实现细粒度访问控制的有效手段之一^[4]。将代理重加密技术引入

CP-ABE中可实现策略转换,即代理服务器将一种访问策略加密的密文转换为另一种访问策略加密的密文,可以使满足后一种访问策略的用户进行数据的访问,实现数据分享,避免授权人想要分享给第三方时还需客户端下载、解密再重加密的麻烦,整个过程在代理服务器完成,利用云计算强大计算能力的同时又不会带来数据泄露的风险。

代理重加密方案大多是单属性授权机构为所有用户颁发密钥,存在权限过大,易成为系统的性能瓶颈等问题,因此本文提出多属性授权机构管理的方式。同时,现有的多属性授权机构下的代理重加密方案只满足应具有的部分特性,如单向性、非交互性,而不满足可重复性、可控制性与可验证性^[5],因此基于文献^[6]的方案,本文提出了支持重复可控特性的云计算多授权中心CP-ABE代理重加密方案。该方案具有如下特点。

1) 在多属性授权中心下支持代理重加密方案的5种特性。在重加密密钥中设置随机因子,代理无法得知随机因子也就无法进行解密,从而无法逆向重加密,实现了单向性;重加密密钥在授权者客户端独立执行,不需第三方交互,从而实现了非交互性;通过对随机因子密文的重加密实现了重复性;是否拥有初始密文中设置的密文子项可以决定是否能对重加密密文进行解密,实

现了可控性；初始密文中的验证子项可对重加密结果进行验证，实现了可验证性，提高了方案的可用性。

2) 在多属性授权中心下结合外包加解密技术。在加密阶段，把一部分配对运算转移到云端；在解密阶段，利用盲化后的私钥对外包在云端的数据文件进行解密，而用户端只需要进行一次指数运算就可以解密，降低了用户的计算开销。

1 相关研究

访问控制是对用户资源的访问进行监控，根据访问者身份来决定其是否有访问权限，是云数据安全的第一道屏障^[7]，但在云计算环境中，访问者的身份未知，这时就需要对数据进行加密。2007年，Bethencourt等^[8]提出CP-ABE方案，该方案引起了学者们广泛的研究。

代理重加密有许多方面的应用，如云计算、文件管理系统^[9]、电子邮件^[10]、医疗系统电子病历共享^[11-12]，为了抵抗量子攻击的基于格的代理重加密方案^[13-14]等，这些应用实现了数据安全、高效的共享。

代理重加密不是独立存在与发展的，2009年，Liang等^[15]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，解决了CP-ABE共享访问策略转换问题，但该方案仅具有单向性、非交互性和可重复性，只达到了选择明文安全。为提高方案的安全性，2015年，Liang^[16]提出了具有适应性选择密文安全的代理重加密方案。Kawai^[17]提出由授权中心外包生成重加密密钥，当访问结构改变时，用户执行很小计算并向授权中心提出生成重加密密钥请求，极大降低了用户的计算开销，但所有的重加密密钥都由一个授权中心生成，易造成授权中心的计算“瓶颈”，同时用户也失去了对重加密的掌控，此外，该方案仅具有单向性、非交互性。Ge等^[18]提出了一种具有选择密文安全的密钥策略基于属性代理重加密方案，其支持任意单调的访问结构，但同样仅具有单向性、非交互性。2019年，冯朝胜等^[5]提出了支持多种特性的基于属性代理重加密方案，该方案满足5种特性，但该方案只达到了选择明文安全。以上方案均是在单授权中心下

构建的，存在系统规模不易扩充、安全和性能瓶颈以及权限过大等问题。

Chase^[19]提出多授权机构属性基于加密方案，有效降低了用户密钥泄露的安全隐患。为了降低客户端计算开销，Chen等^[20]提出了带预解密功能的多属性授权中心属性加密方案，适用于移动云环境。关志涛等^[21]提出一种基于属性加密的多授权中心访问控制模型，降低了重加密属性数量，提高系统重加密效率，但仅具有单向性、非交互性。仲红等^[22]提出了一种高效的、可验证的多授权机构属性基于加密方案，通过在线-离线加密技术，加解密流程的开销下降，验证外包解密的正确性的同时保护用户隐私。为解决多授权中心下合谋攻击和不同域共享数据的问题，杨小东等^[23]将数据通过第三方处理降低了用户的计算开销，引入代理重加密方法实现了不同域之间以及相同域内用户数据分享，但该方案不支持可重复性、可控性。在移动网络社交中，罗思韬等^[24]提出跨境环境下的代理重加密社交隐私保护方案，利用代理重加密技术实现访问结构策略的变换，扩大了交友范围，但该方案只达到了选择明文安全，且只满足单向性、非交互性。Ge^[25]提出了一种可验证的、公平的基于属性的云数据共享代理重加密方案，但该代理重加密方案不满足可重复性与可控性。对于代理服务器是否存在数据泄露，Guo^[26]等提出可问责的代理重加密方案，通过给代理和授权者设置不同的解密能力，通过输出结果确定恶意方，该方案仅满足单向性、非交互性与可验证性，同时不是基于属性的代理重加密方案，不能实现细粒度访问控制。

2 代理重加密方案

2.1 算法特性

(1) 单向性

从Alice到Bob的解密不允许代理从Bob到Alice的重加密，即只允许一个方向的解密，不可逆向进行重加密。因为代理不知道重加密密钥中的随机因子，就不可以对数据执行解密操作，也就无法将Bob的访问策略 (M', ρ') 变为Alice的访问策略 (M, ρ) ，即代理不能逆向进行重加密，具有单向性。

(2) 非交互性

Alice 可以使用 Bob 的公钥生成重加密密钥 rk , 此过程在用户中心授权者客户端执行, 不需要任何受信任的第三方交互。

(3) 可重复性

PS 可以对相同密文进行多次重加密。将随机因子 d 嵌入重加密密钥组成部分中, 并将编码 $E(g^d)$ 与新的访问结构相关组成重加密密钥的一部分。利用重加密密钥对密文进行重加密, 生成的重加密密文由含有随机因子的密文和部分原始密文组成。若要对密文进行多次重加密, 对随机因子的部分密文 C_{rk} 进行重加密即可。

(4) 可控性

数据属主 (DO, data owner) 可以控制数据是否能被进行重加密。在密文解密过程中, 只有重加密密文解密时才需要原始密文子项 C_4 参与, 因此是否拥有该密文子项决定了密文是否能被重新加密。

(5) 可验证性

用户可以验证重加密后的密文是否正确。若 DO 要验证重加密结果是否正确, 则需要其结果 A 以及密文子项 C_0, C_4 发送给 DO, 验证:

$$V = \left(\frac{A}{e(C_4, g^d)} \right)^{H_3^{-1}(\delta)} = e(g^s, g^a) \quad (1)$$

若 $C_0 = g^{H_2(V)}$ 成立, 则 PS 重加密结果正确。

若 DO 要验证外包数据解密的结果是否正确, 则把密文子项 C_1, C_3 和预解密部分 W 进行解密并验证:

$$\text{key} \parallel \beta = H_2(W^{e^{-1}}) \oplus C_1 \quad (2)$$

若 $C_3 = g_1^{H_2(\text{key} \parallel \beta)}$ 成立, 则外包数据解密结果正确。

2.2 网络模型

多授权中心 CP-ABE 代理重加密方案网络模型如图 1 所示, 包括中央授权机构 (CA, central authority)、属性授权机构 (AA, attribute authority)、数据属主 (DO, data owner)、用户 (User)、云服务提供商 (CSP, Cloud Server Provider)。服务器主要包括内容管理服务器、代理加密服务器、代理解密服务器、代理重加密服务器以及数据管理服务器。内容管理服务器分析处理 User 客户端提交的访问要求; 代理加密服务器对 DO 的部分加密数据进行完整加密; 代理解密服务器对符合访问策略的 User 要求访问的数据进行部分解密; 代理重加密服务器可以在不解密出明文的前提下对加密的数据文件进行访问策略的转换, 实现被授权者可以访问, 达到了访问权限授权, 实现了数据分享的目的; 数据管理服务器用来存放加密后的数据文件。本文提出的支持重复可控特性的云计算多授

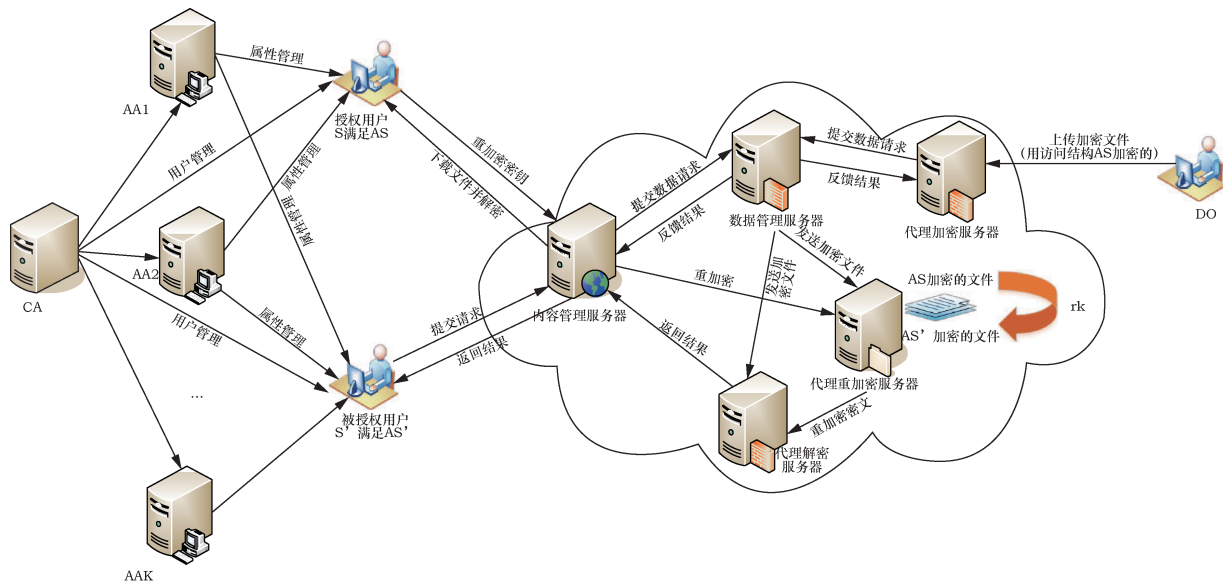


图 1 多授权中心 CP-ABE 代理重加密方案网络模型

Figure 1 The model of CP-ABE proxy re-encryption scheme for cloud computing based on multi-authority

权中心 CP-ABE 代理重加密方案流程如下。

(1) 数据生成

Step1: 数据属主将数据用经典对称加密算法 AES 加密, 得到数据密文 C_{AES} 。

Step 2: 将对称加密的密钥 key 用访问结构加密, 得到密钥密文 m , 上传到代理加密服务器之后得到完整的数据密文并存储在数据管理服务器中。

(2) 数据访问

授权用户与被授权用户均可访问数据。授权用户访问数据的步骤如下。

Step 1: 授权用户向内容管理服务器发起访问请求, 请求 C_{AES} 和 m 。

Step 2: 内容管理服务器分析并提交请求至数据管理服务器, 若用户私钥属性满足访问结构, 判定其为合法用户, 将 m 发送至代理解密服务器进行预解密, 并将预解密结果通过内容管理服务器返回给授权用户。

被授权用户访问数据的步骤如下。

Step 1: 授权用户运行重加密密钥生成算法,

将自身私钥和新的访问结构生成重加密密钥, 发送给内容管理服务器, 内容管理服务器分析再提交至代理重加密服务器, 在此将利用重加密密钥与 m 生成重加密密文 (此时的重加密密文是用新的访问结构加密的)。

Step 2: 生成的重加密密文被发送至代理解密服务器进行预解密, 然后发送至内容管理服务器。被授权用户若访问数据, 则可以通过自身私钥进行解密, 因为被授权用户私钥属性满足新访问结构。

2.3 算法执行

本文所提算法执行流程如下, 其关系如图 2 所示。

- 1) $\text{Setup}(k) \rightarrow (\text{MSK}, \text{PK}, \{s\})$
- 2) $\text{AASetup}(\text{PK}, s, i) \rightarrow (\text{APK}_i, \text{ASK}_i)$
- 3) $\text{IDKeyGen}(\text{PK}, \text{MSK}, \text{ID}_u) \rightarrow (\text{PK}_u, \text{SK}_u)$
- 4) $\text{AttrKeyGen}(\text{PK}, S, \{\text{ASK}_i\}, \text{PK}_u) \rightarrow \text{SK}_{S,u}$
- 5) $\text{Enc}(\text{PK}, \{\text{APK}_i\}, (M, \rho), m) \rightarrow C_{(M, \rho)}^{\text{part}}$

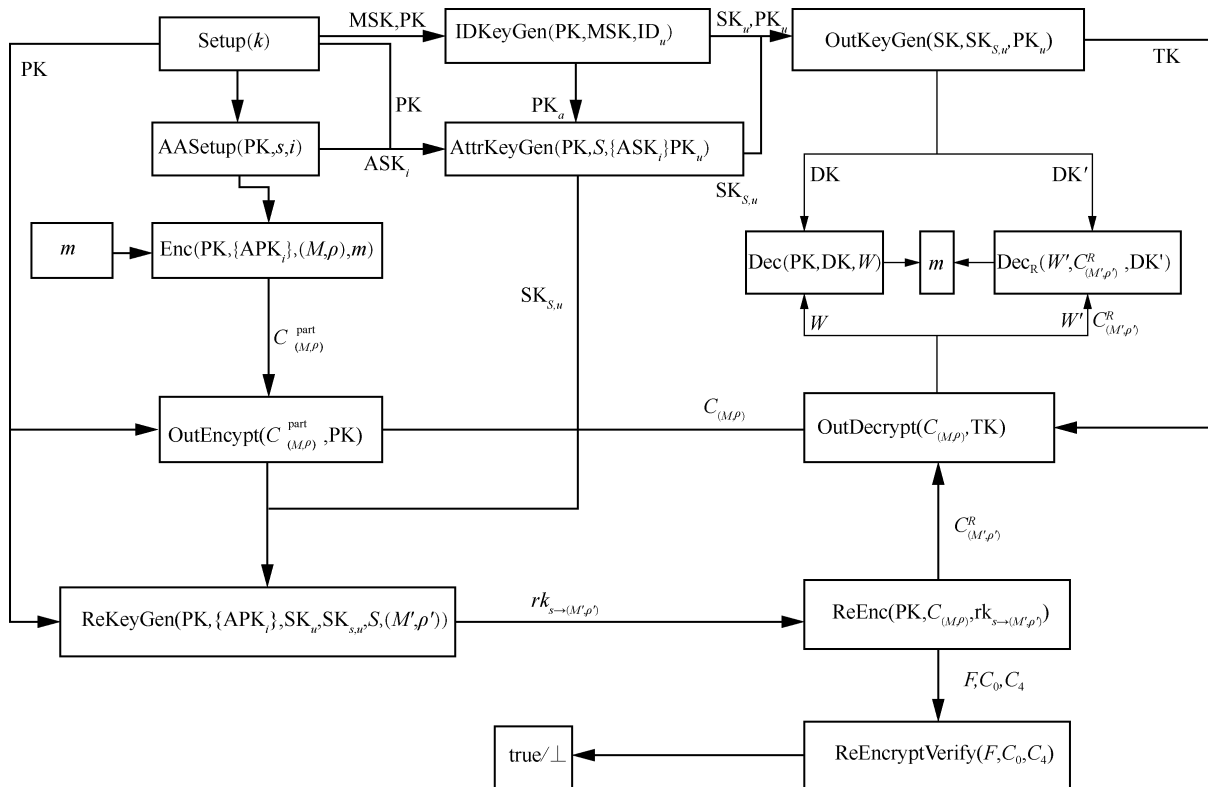


图 2 各算法之间的关系

Figure 2 The relation between algorithms

- 6) $\text{OutEncrypt}(C_{(M,\rho)}^{\text{part}}, \text{PK}) \rightarrow C_{(M,\rho)}$
- 7) $\text{ReKeyGen}(\text{PK}, \{\text{APK}_i\}, \text{SK}_u, \text{SK}_{S,u}, S, (M', \rho')) \rightarrow \text{rk}_{S \rightarrow (M', \rho')}$
- 8) $\text{ReEnc}(\text{PK}, C_{(M,\rho)}, \text{rk}_{S \rightarrow (M', \rho')}) \rightarrow C_{(M', \rho')}^R$
- 9) $\text{ReEncryptVerify}(F, C_0, C_4) \rightarrow (\text{true} / \perp)$
- 10) $\text{OutKeyGen}(\text{SK}, \text{PK}_u) \rightarrow (\text{TK}, \text{DK})$
- 11) $\text{OutDecrypt}(C_{(M,\rho)}, \text{TK}) \rightarrow Z$
- 12) $\text{Dec}(\text{PK}, \text{DK}, Z) \rightarrow m$
- 13) $\text{Dec}_R(Z', C_{(M', \rho')}^R, \text{DK}') \rightarrow m$

对算法按照执行主体进行分类如下。

(1) 授权中心执行

1) $\text{Setup}(k) \rightarrow (\text{MSK}, \text{PK}, \{s\})$, 初始化算法由授权中心执行, 输入安全参数 k , 输出主密钥 MSK, 系统公钥 PK, 各属性管理中心的标识 s 。

2) $\text{IDKeyGen}(\text{PK}, \text{MSK}, \text{ID}_u) \rightarrow (\text{PK}_u, \text{SK}_u)$, 用户身份私钥生成算法由 CA 执行。输入系统公钥 PK, 主密钥 MSK, 用户身份标识 ID_u , 输出用户身份公钥 PK_u 和用户身份私钥 SK_u 。

(2) 属性管理中心执行

1) $\text{AASetup}(\text{PK}, s, i) \rightarrow (\text{APK}_i, \text{ASK}_i)$, 属性管理中心初始化算法由各个属性管理中心执行。输入系统公钥 PK, 该 AA 的标识 s , 以及该 AA 管理的属性 i , 输出属性 i 对应的属性公钥 APK_i 和属性私钥 ASK_i 。

2) $\text{AttrKeyGen}(\text{PK}, S, \{\text{ASK}_i\}, \text{PK}_u) \rightarrow \text{SK}_{S,u}$, 用户属性私钥生成算法由属性管理中心执行。输入系统公钥 PK, 用户的属性集 S , 属性私钥 ASK_i , 用户身份公钥 PK_u , 输出用户属性私钥 $\text{SK}_{S,u}$ 。

(3) 数据属主执行

1) $\text{Enc}(\text{PK}, \{\text{APK}_i\}, (M, \rho), m) \rightarrow C_{(M,\rho)}^{\text{part}}$, 加密算法由数据属主执行。输入系统公钥 PK, 属性公钥 APK_i , LSSS 访问结构 (M, ρ) , 待加密的数据 m , 输出 (M, ρ) 与 m 相关联的部分密文 $C_{(M,\rho)}^{\text{part}}$ 。

2) $\text{ReEncryptVerify}(F, C_0, C_4) \rightarrow (\text{true} / \perp)$, 重加密验证算法由用户执行。输入 $C_{(M', \rho')}$ 和密文组件 C_0 、 C_4 , 若验证通过, 则输出 true, 否则输出 $\perp$ 。

(4) CSP 加密代理服务器执行

1) $\text{OutEncrypt}(C_{(M,\rho)}^{\text{part}}, \text{PK}) \rightarrow C_{(M,\rho)}$, 外包加密算法由加密代理服务器执行, 输入用户计算后的部分密文 $C_{(M,\rho)}^{\text{part}}$ 和系统公钥 PK, 输出数据 m 的完整密文 $C_{(M,\rho)}$ 。

2) $\text{ReEnc}(\text{PK}, C_{(M,\rho)}, \text{rk}_{S \rightarrow (M', \rho')}) \rightarrow C_{(M', \rho')}^R$, 密文重加密算法由 CSP 执行, 输入系统公钥 PK, 与访问结构 (M, ρ) 相关的密文 $C_{(M,\rho)}$, 与 (M', ρ') 相关联的重加密密钥 $\text{rk}_{S \rightarrow (M', \rho')}$, 当密文 $C_{(M,\rho)}$ 被设置为不能重加密或者 $S \neq (M, \rho)$ 时, 输出 $\perp$, 否则输出与访问结构 (M', ρ') 相关联的密文 $C_{(M', \rho')}^R$ 。

3) $\text{OutDecrypt}(C_{(M,\rho)}, \text{TK}) \rightarrow Z$, 外包解密算法由 CSP 执行, 输入密文 $C_{(M,\rho)}$ 和转换密钥 TK, 当 $S \models (M, \rho)$ 时, 输出部分解密密文 Z , 否则输出 $\perp$ 。

(5) 授权用户执行

1) $\text{ReKeyGen}(\text{PK}, \{\text{APK}_i\}, \text{SK}_u, \text{SK}_{S,u}, S, (M', \rho')) \rightarrow \text{rk}_{S \rightarrow (M', \rho')}$, 重加密密钥生成算法由授权用户 (delegator) 执行, 输入系统公钥 PK, 属性公钥 APK_i , 授权用户身份私钥 SK_u , 用户属性私钥 $\text{SK}_{S,u}$, 属性集 S 和新的 LSSS 访问结构 (M', ρ') , 输出一个重加密密钥 $\text{rk}_{S \rightarrow (M', \rho')}$ 。若重加密密钥中, 授权用户属性私钥 $\text{SK}_{S,u}$ 对应的属性集 S , 满足密文的访问结构 (M, ρ) 时, 代理服务器可以利用该重加密密钥用于将访问结构 (M, ρ) 对应的密文向访问结构 (M', ρ') 对应的密文的替换。

2) $\text{OutKeyGen}(\text{SK}, \text{PK}_u) \rightarrow (\text{TK}, \text{DK})$, 转换密钥生成算法由用户执行。输入用户私钥 SK 和身份公钥 PK_u , 输出外包至 CSP 解密的转换密钥 TK, 以及最终用户解密时的密钥 DK。

3) $\text{Dec}(\text{PK}, \text{DK}, Z) \rightarrow m$, 解密算法由授权用户执行, 输入部分解密密文 Z , 用户保留的密钥 DK, 输出数据 m 。

(6) 被授权用户执行

$\text{Dec}_R(Z', C_{(M', \rho')}^R, \text{DK}') \rightarrow m$ 重加密密文解密算法由被授权用户执行, 如果被授权用户属性集满足新的访问结构 (M', ρ') , 即 $S \models (M', \rho')$, 则输出明文 m , 否则输出 $\perp$ 。

正确性: 对于任意属性集 $S \in U$, 任意访问结构 (M, P) 和任意明文 m , 若 $S \models (M, P)$, $S' \models (M', P')$, 则有:

$$\text{OutEncrypt}(\text{Encrypt}(\text{PK}, \{\text{APK}_i\}, (M, \rho)), \text{PK}) \rightarrow C_{(M, \rho)}$$

$$\text{Decrypt}(\text{PK}, \text{DK}, \text{OutDecrypt}(C_{(M, \rho)}, \text{TK})) \rightarrow m$$

$$\text{Decrypt}(C_{(M', \rho')}^R, \text{OutDecrypt}(C_{(M', \rho')}^R, \text{TK}), \text{DK}') \rightarrow m$$

$$\text{ReEncrypt}(\text{TK}, C_{(M, \rho)}, \text{ReKeyGen}(\{\text{APK}_i\}, \text{SK}_u, \text{SK}_{S, u}, (M', \rho'))) \rightarrow C_{(M', \rho')}^R \quad (3)$$

3 方案构造

本文提出的方案包括 CA 建立、AA 建立、用户私钥生成、数据加密、外包加密、重加密密钥生成、密文重加密、重加密验证、转换密钥生成、外包解密以及解密 (原始密文解密和重加密密文解密)。

(1) CA 建立: Setup(k)

k 为系统安全参数, CA 选择阶为素数 p 的乘法循环群 G 和 G_T , 生成元 $g, g_1 \in G$ 。哈希函数 $H_1: \{0, 1\}^* \rightarrow \mathbb{Z}_p^*$, $H_2: G_T \rightarrow \{0, 1\}^{2k}$, $H_3: \{0, 1\}^k \rightarrow \mathbb{Z}_p^*$ 。选择经典的对称密码算法 $\text{AES} = (\text{AES.Enc}(\cdot), \text{AES.Dec}(\cdot))$ 。设属性空间为 U , 属性最大个数为 n , 属性域为 $U = \{1, 2, \dots, n\}$ 。系统中有 k 个属性管理中心 AA_k , 每个 AA 管理互不交叉的属性组, 一个 AA 可管理多个属性。

CA 随机选择元素 $P, g_2 \in G, \alpha \in \mathbb{Z}_p$, 利用伪随机函数为 k 个属性管理中心 $\{\text{AA}_k\}$ 生成标识 $s_1, s_2, \dots, s_k$ 。CA 生成系统公钥如下: $\text{PK} = \langle P, g_1, g_2, e(g, g)^\alpha, H_1, H_2, H_3, \text{AES} \rangle$, 主密钥 $\text{MSK} = (g^\alpha)$ 。

(2) AA 建立: AASetup(PK, s, a)

对 AA_k 管理的每一个属性 i , 生成随机值 $t_i = \text{PRF}_{\text{sk}}(i) \in \mathbb{Z}_p$, 生成属性 i 的私钥 $\text{ASK}_i = t_i$, 属性 i 的公钥 $\text{APK}_i = g^{t_i}$ 。

(3) 用户私钥生成

1) 用户身份私钥生成: IDKeyGen($\text{PK}, \text{MSK}, \text{ID}$)。

CA 对注册的每一个新用户, 分配一个唯一身份标识符 $u \in \mathbb{Z}_p$, CA 随机选择一个对应值 $\text{mk}_u \in \mathbb{Z}_p$, 生成用户的身份公钥 $\text{PK}_u = g^{\text{mk}_u}$, 用

户身份私钥 $\text{SK}_u = (\text{MSK} \cdot P^{\text{mk}_u}) = g^\alpha \cdot P^{\text{mk}_u}$ 。

2) 用户属性私钥生成: AttrKeyGen($\text{PK}, S, \{\text{ASK}_i\}, \text{PK}_u$)

AA 为合法用户输出属性 i 对应的属性私钥 $\text{SK}_{i, u} = (\text{PK}_u)^{t_i} = g^{\text{mk}_u \cdot t_i}$ 。

用户的属性集私钥 $\text{SK}_{S, u} = \{\text{SK}_{i, u}, \forall i \in S\}$ 。

(4) 数据加密: Enc($\text{PK}, m, \{\text{APK}_i\}, (M, \rho)$)

DO 根据对称加密算法 $\text{AES} = (\text{AES.Enc}(\cdot), \text{AES.Dec}(\cdot))$, 产生对称密钥 $\text{key} \in \{0, 1\}^k$, 选择随机值 $\beta \in \{0, 1\}^k$, 令 $s = H_1(\text{key}, \beta)$ 。

DO 对 key 进行加密, 定义访问结构 (M, ρ) (M 是一个 $l \times n$ 的矩阵, ρ 是矩阵每一行 M_i 到属性 $\rho(i)$ 的映射关系)。随机选择一个向量 $\mathbf{v} = (s, v_2, v_3, \dots, v_n)$, 其中, $v_2, v_3, \dots, v_n \in \mathbb{Z}_p^*$ 。随机选择 $z \in \mathbb{Z}_p^*$, $Z = P^z$, $C_0 = g^{H_2(e(g, g)^{\alpha s})}$, $C_1 = H_2(e(g, g)^{\alpha s}) \oplus (\text{key} \parallel \beta)$ 。随机选择 $r_1, r_2, \dots, r_n \in \mathbb{Z}_p^*$, $C_2 = g^s$, $C_3 = g_1^s$, $\forall i \in [1, l]$, $(C_{i,1}^{\text{part}} = \text{APK}_{\rho(i)}^{-r_i}, C_{i,2} = g^{r_i}, C_i^* = \lambda_i / z)$, $C_4 = g_2^s$ 。将 $C_{(M, \rho)}^{\text{part}}$, Z , C_i^* 以及数据密文 C_{AES} 发送给 CSP。部分密文: $C_{(M, \rho)}^{\text{part}} = \langle (M, \rho), C_0, C_1, C_2, C_3, C_4, \forall i \in [1, l](C_{i,1}^{\text{part}}, C_{i,2}) \rangle$ 。

(5) 外包加密: OutEncrypt($C_{(M, \rho)}^{\text{part}}, \text{PK}$)

计算 $\forall i \in I = [1, l]$, $C_i = Z^{r_i} = P^{r_i}, C_{i,1} = C_i \cdot C_{i,1}^{\text{part}}$, $C_{(M, \rho)} = \langle (M, \rho), C_0, C_1, C_2, C_3, C_4, \forall i \in [1, l](C_{i,1}, C_{i,2}) \rangle$ 。

(6) 重加密密钥生成: ReKeyGen($\text{PK}, \text{SK}, (M', \rho')$)

随机选择 $\beta', \delta \in \{0, 1\}^k$, 令 $s' = H_1(\beta', \delta)$, 选择随机向量 $\mathbf{v}' = (s', v'_2, v'_3, \dots, v'_n)$, $v'_2, v'_3, \dots, v'_n \in \mathbb{Z}_p^*$, 对 $i = 1 \dots l'$, 令 $\lambda'_i = \mathbf{v}' \cdot \mathbf{M}'_i$ ($\mathbf{M}'_i$ 是 $l' \times n'$ 的矩阵 $\mathbf{M}'$ 第 i 行的向量), $C'_1 = H_2((e(g, g)^{\alpha s'}) \oplus (\delta \parallel \beta'))$, $C'_2 = g^{s'}$, $C'_3 = g_1^{s'}$, $C'_4 = g_2^{s'}$, $\forall i \in [1, l']$ ($C'_{i,1} = P^{\lambda'_i} \text{APK}_{\rho'(i)}^{-r'_i}$, $C'_{i,2} = g^{r'_i}$), $C_{(M', \rho')} = (C'_1, C'_2, C'_3, C'_4, C'_{i,1}, C'_{i,2})$ 。

选择随机因子 $d \in \mathbb{Z}_p^*$, 计算 g^d , 编码得 $E(g^d)$ 。选择随机值 $\theta \in \mathbb{Z}_p^*$, $\text{rk}_1 = (\text{SK}_u)^{H_3(\delta)} \cdot g_1^\theta \cdot g_2^d$, $\text{rk}_2 = g^\theta$, $\text{rk}_3 = (\text{PK}_u)^{H_3(\delta)}$, $\text{rk}_4 = C_{(M', \rho')}$, $\forall i \in S$, $R_{i, u} = (\text{SK}_{i, u})^{H_3(\delta)} \cdot C'_{\text{rk}} = \text{Encrypt}(\text{PK},$

$E(g^d), \{APK_i\}, (M', \rho')$ 。

输出重加密密钥： $rk_{S \rightarrow (M', \rho')} = (S, rk_1, rk_2, rk_3, rk_4, C'_{rk}, R_{i,u})$ 。

(7) 密文重加密： $ReEncrypt(PK, C_{(M, \rho)},$

$$A = \frac{e(C_2, rk_1) / e(C_3, rk_2)}{\prod_{i \in I} (e(C_{i,1}, rk_3) \cdot e(C_{i,2}, R_{\rho(i)}))^{\omega_i}} = \frac{e(g^s, (g^\alpha \cdot P^{mk_u})^{H_3(\delta)} \cdot g_2^d \cdot g_1^\theta)}{e(g_1^s, g^\theta) \cdot \prod_{i \in I} (e(P^{\lambda_i} \cdot g^{-t_i r_i}, g^{mk_u H_3(\delta)}) \cdot e(g^{r_i}, (g^{mk_u t_i})^{H_3(\delta)}))^{\omega_i}} =$$

$$\frac{e(g^s, g^{\alpha H_3(\delta)}) \cdot e(g^s, P^{mk_u H_3(\delta)}) \cdot e(g^s, g_2^d) \cdot e(g^s, g_1^\theta)}{e(g_1^s, g^\theta) \cdot e(P^s, g^{mk_u H_3(\delta)})} = e(g^s, g^{\alpha H_3(\delta)}) \cdot e(g^s, g_2^d) \quad (4)$$

重加密密钥密文： $C_{(M', \rho')}^R = \langle S, (M', \rho'), C_1, C_3, \forall i \in [1, l] (C_{i,1}, C_{i,2}), rk_4, A, C_{rk} \rangle$ 。

(8) 重加密验证

User 使用重加密后的 A ，验证因子 C_0, C_4 对重加密结果进行正确性验证。

$$V = \left(\frac{A}{e(C_4, g^d)} \right)^{H_3^{-1}(\delta)} = e(g^s, g^\alpha) \quad (5)$$

若 $C_0 = g^{H_2(V)}$ ，则重加密正确，输出 true，否则输出 $\perp$ 。

(9) 转换密钥生成： $OutKeyGen(SK)$

选择随机数 $\varepsilon \in \mathbb{Z}_p^*$ ， $TK = \langle tk_{D_{1,1}} = (\forall i \in S, (SK_u)^\varepsilon), tk_{D_{1,2}} = (PK_u)^\varepsilon, tk_{D_{1,3}} = (SK_{i,u})^\varepsilon \rangle$ 用户保存 $DK = \langle \varepsilon \rangle$ 。

(10) 外包解密： $OutDecrypt(C_{(M, \rho)}, TK)$

选择 ω_i ，使 $\sum_{\rho(i) \in S} \omega_i \cdot M_i = (1, 0, \dots, 0)$ 。

$$W = \frac{(C_2, tk_{D_{1,1}})}{\prod_{i \in I} (e(C_{i,1}, tk_{D_{1,2}}) \cdot e(C_{i,2}, tk_{D_{1,3}}))^{\omega_i}} =$$

$$\frac{e(g^s, (g^\alpha \cdot P^{mk_u})^\varepsilon)}{\prod_{i \in I} (e(P^{\lambda_i} \cdot g^{-t_i r_i}, g^{mk_u \varepsilon}) \cdot e(g^{r_i}, g^{mk_u t_i \varepsilon}))^{\omega_i}} =$$

$$\frac{e(g^s, g^\alpha) \cdot e(g^s, P^{mk_u \varepsilon})}{\prod_{i \in I} (e(P^{\lambda_i}, g^{mk_u \varepsilon}) \cdot e(g^{-t_i r_i}, g^{mk_u \varepsilon}) \cdot e(g^{t_i r_i}, g^{mk_u \varepsilon}))^{\omega_i}} =$$

$$\frac{e(g^s, g^{\alpha \varepsilon}) e(g^s, P^{mk_u \varepsilon})}{e(P^s, g^{mk_u \varepsilon})} = e(g^s, g^{\alpha \varepsilon}) \quad (6)$$

(11) 解密

对于初始密文解密： $key \parallel \beta = H_2(W^{\varepsilon^{-1}}) \oplus C_1$ 。

$rk_{S \rightarrow (M', \rho')}$

$C_{rk} = OutEncrypt(C'_{rk}, PK)$ ，选择常数 ω_i ，使

$\sum_{\rho(i) \in S} \omega_i \cdot M_i = (1, 0, \dots, 0)$ 。计算：

$$e(g^s, (g^\alpha \cdot P^{mk_u})^{H_3(\delta)} \cdot g_2^d \cdot g_1^\theta) / (e(g_1^s, g^\theta) \cdot \prod_{i \in I} (e(P^{\lambda_i} \cdot g^{-t_i r_i}, g^{mk_u H_3(\delta)}) \cdot e(g^{r_i}, (g^{mk_u t_i})^{H_3(\delta)}))^{\omega_i})$$

验证 $C_3 = g_1^{H_2(key \parallel \beta)}$ 是否成立，若成立，得到 key，计算数据明文 $m = AES.Dec(C_{AES}, key)$ ，否则输出 $\perp$ 。

对于被重加密的密文，按照上面的解密算法对 C_{rk} 解密得到 $E(g^d)$ ，解码得 g^d 。设 $I' \subset \{1, 2, \dots, l\}$ 被定义为 $I' = \{i : \rho'(i) \in S'\}$ ， $\{\omega'_i \in \mathbb{Z}_p^*\}_{i \in I'}$ 是一个常数集， $\{\lambda'_i\}$ 是 M' 对秘密 s' 的共享秘密份额， $\sum_{i \in I'} \omega'_i \cdot \lambda'_i = s'$ 。代理解密服务器计算：

$$W' = \frac{(C'_2, tk'_{D_{1,1}})}{\prod_{i \in I} (e(C'_{i,1}, tk'_{D_{1,2}}) \cdot e(C'_{i,2}, tk'_{D_{1,3}}))^{\omega'_i}} = e(g, g)^{\alpha s' \varepsilon'} \quad (7)$$

$$\delta \parallel \beta' = H_2((W')^{1/\varepsilon'}) \oplus C'_1 \quad (8)$$

$$key \parallel \beta = H_2\left(\left(\frac{A}{e(C_4, g^d)}\right)^{H_3^{-1}(\delta)}\right) \oplus C_1 \quad (9)$$

验证 $C_3 = g_1^{H_2(key \parallel \beta)}$ 是否成立，若成立，输出 key，即对称加密方法的密钥。对于被多次重加密的密文，采用第一种解密方法解密出最后一次重加密的数据，再重复使用第二种解密方法，得到 key，使用 key 进行对称解密， $m = AES.Dec(C_{AES}, key)$ ，得出明文。

4 安全性分析

判定性 q -parallel BDHE (bilinear diffie-Hellman exponent) 假设。给定阶为素数 p 的群 G ， g 是 G 的一个生成元，随机选取 $a, s, b_1, \dots, b_q \in \mathbb{Z}_p$ 和一个随机元素 $T \in G_T$ 。判定性 q -parallel BDHE 问题描述为：如果给定敌手 γ ，

$$\begin{aligned} \gamma &= g, g^s, g^a, \dots, g^{a^q}, g^{a^{q+2}}, \dots, g^{a^{2q}} \\ \forall_{1 \leq j < q} g^{sb_j}, g^{a/b_j}, \dots, g^{a^q/b_j}, g^{a^{q+2}/b_j}, \dots, g^{a^{2q}/b_j} \\ \forall_{1 \leq j, k \leq q, k \neq j} g^{asb_k/b_j}, \dots, g^{a^qsb_k/b_j} \end{aligned} \quad (10)$$

判断 T 是 $e(g, g)^{a^{q+1}s}$ 还是群 G_T 中的一个随机元素 R 。敌手使用多项式算法 B 攻破上述假设的优势为： $|\Pr[B(\gamma, T = e(g, g)^{a^{q+1}s}) = 0] - \Pr[\gamma, T = R) = 0] \geq \varepsilon$ 。

定义 1 如果不存在概率多项式时间 (PPT) 算法以不可忽略的优势解决判定性 q -parallel BDHE 问题, 则称 q -parallel BDHE 问题是困难的。

定理 1 假设判定性 q -parallel BDHE 假设在 (G, G_T) 上成立, 那么不存在敌手 $\mathcal{A}$ 能在一个多项式时间内选择访问结构 (M^*, ρ^*) 攻破本文方案, 本文方案在标准模型下可证 IND-sAS-CCA-Or (selective access structure and choose ciphertext security at original ciphertext) 安全。

证明 假设存在 $\mathcal{A}$ 能在 IND-sAS-CCA-Or 中以 $\varepsilon = \text{Adv}_{\mathcal{A}}$ 的优势攻破本文方案, 则可根据判定性 q -parallel BDHE 假设, 判断出 $T = e(g, g)^{a^{q+1}s}$ 还是属于 G_T 的随机值。挑战者 $\mathcal{C}$ 和敌手 $\mathcal{A}$ 按下面方式进行游戏。

(1) 初始化阶段

$\mathcal{A}$ 将要挑战的访问结构 (M^*, ρ^*) 发送给挑战者 $\mathcal{C}$, M^* 是一个 $l^* \times n^*$ 的矩阵, $l^*, n^* \leq q$ 。 $\mathcal{A}$ 公布攻破的属性管理中心 W'_{AA} 。 $\mathcal{C}$ 输入参数 γ 和 T 。

$$\begin{aligned} \gamma &= g, g^s, g^a, \dots, g^{a^q}, g^{a^{q+2}}, \dots, g^{a^{2q}} \\ \forall_{1 \leq j < q} g^{sb_j}, g^{a/b_j}, \dots, g^{a^q/b_j}, g^{a^{q+2}/b_j}, \dots, g^{a^{2q}/b_j} \\ \forall_{1 \leq j, k \leq q, k \neq j} g^{asb_k/b_j}, \dots, g^{a^qsb_k/b_j} \end{aligned} \quad (11)$$

(2) 系统建立阶段

$\mathcal{A}$ 随机选择 $\alpha' \in \mathbb{Z}_p$, 设置 $\alpha = \alpha' + \alpha^{q+1}$, 则有 $e(g, g)^\alpha = e(g^\alpha, g^{\alpha'})e(g, g^{\alpha'})$ 。 $\mathcal{C}$ 运行初始化算法将 $PK = e(g, g)$ 发送给 $\mathcal{A}$ 。 $\mathcal{C}$ 运行 AASetup 算法, 分为两种情况。若管理属性 x 被攻破, 则 $\mathcal{C}$ 随机选择 z_x , $APK_x = g^{z_x}$, 否则 $APK_x = g^{z_x} \cdot \prod_{i \in X} g^{a \cdot M_{i,1}^* / b_1} \cdot g^{a^2 \cdot M_{i,2}^* / b_1} \dots g^{a^n \cdot M_{i,n^*}^* / b_1}$ 。

对于未被攻破的 AA, $\mathcal{C}$ 仅将对应的 APK 发

送给 $\mathcal{A}$; 对于已被攻破的属性管理中心, $\mathcal{C}$ 将该属性管理中心的 APK 和 ASK 都发送给 $\mathcal{A}$ 。

$\mathcal{A}$ 进行如下询问。

1) $H_1(m, \beta)$: 若在表 ListH₁ 中存在 (m, β, s) , 则返回 s 给 $\mathcal{A}$; 否则选择随机数 $s \in \mathbb{Z}_p$, 将 (m, β, s) 记录在表 ListH₁ 中并返回 s 给 $\mathcal{A}$ 。

2) $H_2(R)$: 若在表 ListH₂ 中存在 (R, r) , 则返回 r 给 $\mathcal{A}$; 否则选择随机数 $r \in \{0, 1\}^{2k}$, 将 (R, r) 记录在表 ListH₂ 中并返回 r 给 $\mathcal{A}$ 。

3) $H_3(\delta)$: 若在表 ListH₃ 中存在 (δ, ζ) , 则返回 r 给 $\mathcal{A}$; 否则选择随机数 $\zeta \in \mathbb{Z}_p^*$, 将 (δ, ζ) 记录在表 ListH₃ 中并返回 ζ 给 $\mathcal{A}$ 。

(3) 查询阶段 1

1) $\mathcal{A}$ 向 $\mathcal{C}$ 提出请求 $Q_{sk}(u, S)$, 且对同一用户 u 只能进行一次查询。一个用户的私钥由 SK_u 和 $SK_{a,u}$ 两部分组成。

由于用户 $S \models (M^*, \rho^*)$, 因此式(12)、式(13)构造 PK_u 和 SK_u :

$$\begin{aligned} PK_u &= g^r \cdot \prod_{i=1, \dots, n^*} g^{a^{q+1-i} \cdot \omega_i} = g^{mk_u} \\ SK_u &= g^{\alpha'} \cdot g^{\alpha} \cdot \prod_{i=2, \dots, n^*} (g^{q+2-i})^{w_i} \end{aligned} \quad (12)$$

其中, $mk_u = r + w_1 \cdot a^q + w_2 \cdot a^{q-1} + \dots + w_{n^*} \cdot a^{q-n^*+1}$, $SK_{a,u}$ 构造有两种情况, 不能包含不可模拟项 $g^{-a^{q+1}}$, 因此当 $\rho^*(i) \neq a(i = i, \dots, l^*)$ 时, 按真实方案构建 $SK_u = PK_u^{z_a}$, z_a 是随机数; 否则按式(13)构建。

$$SK_{a,u} = PK_u^{z_a}.$$

$$\prod_{i \in X} \prod_{j=1, \dots, n^*} (g^{(a^j/b_1) \cdot r} \prod_{k=1, \dots, n^*, k \neq j} (g^{a^{q+1+j-k}/b_1})^{\omega_k})^{M_{i,j}^*} \quad (13)$$

2) $\mathcal{A}$ 向 $\mathcal{C}$ 提出查询请求 $Q_{rk}(u, S, (M', \rho'))$, 若元组已经存在于对应的表中, $\mathcal{C}$ 则把 $rk_{S \rightarrow (M', \rho')}$ 返回给 $\mathcal{A}$ 。否则, 若 $S \models (M^*, \rho^*)$, 且 $S' \models (M', \rho')$, 已被询问过的对应私钥不能再次询问, 若没有询问过且对应的私钥不存在于重加密密钥列表里, 构建 $rk_{S \rightarrow (M', \rho')}$ 如下: $\mathcal{A}$ 随机选择 $\theta, \sigma \in \mathbb{Z}_p^*, \delta, \beta' \in \{0, 1\}^k, \tilde{K} \in G$, $rk_1 = \tilde{K} \cdot g_1^\theta$, $rk_2 = g^\theta, rk_3 = g^\sigma, R_x = APK_x^\delta (\forall x \in S), rk_4 = C_{(M', \rho')}$, $rk_{S \rightarrow (M', \rho')} = (S, rk_1, rk_2, rk_3, rk_4, R_{u,S})$ 记录到表

ListRK 中并返回给 $\mathcal{C}$ 。

3) $\mathcal{A}$ 向 $\mathcal{C}$ 提出重加密密文查询： $Q_{re}(u, S, (M', \rho'), C_{(M, \rho)})$ 。

4) $\mathcal{A}$ 向 $\mathcal{C}$ 提出原始密文解密查询： $Q_{d_1}(u, S, C_{(M, \rho)})$ 。

5) $\mathcal{A}$ 向 $\mathcal{C}$ 提出私钥查询请求重加密密文解密查询： $Q_{d_2}(u', S', C_{(M', \rho')})$ 。

(4) 挑战阶段

$\mathcal{A}$ 给 $\mathcal{C}$ 提交消息 m_0 和 m_1 ， $\mathcal{C}$ 随机选择一个 m_b ($b=0$ 或 1)，加密并返回密文 $C_{(M^*, \rho^*)}^* = ((M^*, \rho^*), C_0^*, C_1^*, C_2^*, C_3^*, C_4^*(C_{i,1}^*, C_{i,2}^*))$ 给 $\mathcal{A}$ 。

$$C_{i,1}^* = (\text{APK}_{\rho^*(i)})^{-r'_i} \left(\prod_{j=1, \dots, n^*} (g^a)^{M_{i,j}^* y'_j} \right) (g^{b_i s})^{-z_{\rho^*(i)}} \quad (14)$$

$$\left(\prod_{k \in R_i} \prod_{j=1, \dots, n^*} (g^{a^j \cdot s \cdot (b_i / b_k)})^{M_{k,j}^*} \right)^{-1}$$

如果 $T = e(g, g)^{a^{q+1}s}$ ，则密文 $C_{(M^*, \rho^*)}^*$ 有效。

$$C_1^* = H_2(T \cdot e(g^s, g^{a'})) \oplus (M_b \parallel \beta^*) =$$

$$H_2(e(g, g)^{a^{q+1}s} \cdot e(g^s, g^{a'})) \oplus (M_b \parallel \beta^*) =$$

$$H_2(e(g^s, g^{a'})) \oplus (M_b \parallel \beta^*) \quad (15)$$

(5) 查询阶段 2

继续查询阶段 1 中的询问。

(6) 猜测阶段

$\mathcal{A}$ 对 b 进行猜测输出 $b' \in \{0, 1\}$ ，若猜测正确，即 $b' = b$ ，得到 $T = e(g, g)^{a^{q+1}s}$ ；否则得到的是 G_T 中的随机 T 。

当 $\mathcal{C}$ 输出为 1 时， $T = e(g, g)^{a^{q+1}s}$ ， $\mathcal{A}$ 得到了关于 m_b 的有效密文， $\Pr[b' = b | (\gamma, T = e(g,$

$g)^{a^{q+1}s}) = 0] = \frac{1}{2} + \varepsilon$ ；当输出 0 时，得到的 T 是随机值，此时得不到任何关于 m_b 的消息，无法恢复明文， $\Pr[b' \neq b | (\gamma, T = R) = 0] = \frac{1}{2}$ 。因此挑战者 $\mathcal{C}$ 在判定性 q -parallel BDHE 中有不可忽略的优势。

5 性能分析

下面从计算开销和存储开销两方面对本文所提方案进行分析。假设系统中的属性个数为 N ，属性管理中心 AA 的数量为 k ， N_p 为共享访问策略中的属性数。其中，双线性运算 B 和指数运算 E 运行时间最长（用 E_T 、 E_G 分别表示 G_T 群和 G 群的指数运算），其他计算开销远小于这两个运算的计算开销可被忽略为 0，所以用 B 和 E 来衡量性能。

5.1 计算开销

表 1 列举并对比了几种经典方案的用户客户端计算开销，下面进行详细分析。

用户客户端加密数据，包括用对称密钥 key 加密数据文件，以及使用基于密文策略的属性加密算法对 key 进行加密得到密钥密文。前者的计算复杂度与数据文件的大小有关，而后者的计算开销如下：密文中 $Z, C_0, C_1, C_2, C_3, C_4$ 需执行 6 次指数运算， ρ 中每个属性 $\rho(i)$ 需执行两次指数运算，计算开销为 $(2N_p + 5)E_G + E_T$ 。云服务器为 LSSS 共享访问结构中每个属性 $\rho(i)$ 执行一次指数运算，开销为 $N_p E_G$ 。

生成重加密密钥时，加密随机值 δ 并生成重

表 1 各阶段用户客户端计算开销
Table 1 Comparison of computing overhead on user's client

方案	加密	原始密文解密	重加密密钥	重加密密文解密
Liang ^[5]	$(3 + 4N_p)E_G + 2E_T$	$(2 + 3N_p)E_G + (1 + N_p)E_T + (7 + 3N_p)B$	$(6 + 4N_p + N_s)E_G + 2E_T$	$4E_G + (3 + N_p)E_T + (3 + 2N_p)B$
Liang ^[15]	$(5 + 3N_p)E_G + E_T + \text{sig}$	$(1 + 2N_p)E_G + N_p E_T + (10 + 3N_p)B + \text{sig}$	$(14 + 3N_p + 2N_s)E_G + E_T + \text{sig}$	$(2 + 6N_p)E_G + E_T + (16 + 6N_p)B + 2\text{sig}$
Luo ^[23]	$(2 + 2N_p)E_G + E_T$	$N_p E_T + (1 + 2N_p)B$	$(4 + 2N_p)E_G + E_T$	$(1 + N_p)E_T + (1 + 2N_p)B$
Yang ^[22]	$(2 + 2N_p)E_G + E_T$	—	$(4 + 2N_p)E_G + E_T$	E_T
Feng ^[4]	$(7 + N_p)E_G + E_T$	$3E_T$	$(10 + N_p)E_G + E_T$	$4E_T + B$
Ge ^[24]	$(4 + 2N_p)E_G + E_T$	$(1 + 2N)B$	$(5 + 3N_p)E_G + E_T$	$(1 + 2N)B + E_G$
本文方案	$(5 + 2N_p)E_G + E_T$	$E_G + E_T$	$(8 + 2N_p)E_G + E_T$	$2E_T + E_G + B$

加密密钥。主要计算开销为 $(2N_p + 8)B + E_T$ ，还需执行一次对编码 $E(g^d)$ 的加密过程。

CSP 重加密时，计算共享策略的每个属性 $\rho(i)$ 都参与两次 B 运算和一次 E 运算，计算 A 时，分子还需要进行两次双线性运算，计算开销为 $(2N_p + 2)B + N_p E_T$ 。CSP 为新访问结构中的每个属性 $\rho(i)$ 执行一次 E 运算的开销为 $N_p E_G$ ，PS 的计算代价为 $(2N_p + 2)B + N_p E_T + N_p E_G$ 。User 对重加密结果进行正确性验证，计算开销为 $B + E_T + E_G$ 。

文件解密时使用 CP-ABE 解密算法解密密钥密文，得到对称密钥 key ，再利用 key 进行对称解密即可得到数据文件。对称解密的复杂度与数据文件的大小有关，而前者的计算开销分为原始密文的解密和重加密密文解密两种情况。

原始密文解密时，User 对 Z 执行一次 E 运算，验证需要一次 E 运算，故总代价为 $E_T + E_G$ 。

重加密密文解密时，对随机因子进行一次解密，User 需进行两次 G_T 上的 E 运算，一次 B 运算，验证需再进行一次 G 上的 E 运算，所以在用户客户端的计算开销为 $2E_T + E_G + B$ 。CSP 解密流程与重加密流程类似，计算开销为 $N_p E_T + 2N_p B$ 。

5.2 存储开销

存储开销主要包括以下 3 个阶段。

用户私钥生成：包括用户身份私钥和用户属性私钥。身份私钥和属性私钥分别由一个 G 群元素和 N_s 个 G 群元素组成。

文件加密：DO 将数据文件用对称密钥 key 进行加密后，将密钥密文和数据密文传送给 CSP。密钥密文由 $2N_p + 4$ 个 G 群元素构成。方案中需要增加验证消息，故增加了一个 G 群上的元素。

重加密密文：PS 对密文进行重加密生成的重加密密文由两个 G 群元素和一个 G_T 群元素组成，由于需要验证正确性增加了一个 G 群元素。

私钥与密文存储开销对比如表 2 所示。

表 2 私钥与密文存储开销对比
Table 2 Comparison of storage efficiency of private key and ciphertext

方案	私钥	原始密文	重加密密文
Liang ^[5]	$(2 + N_s) G $	$(3 + 2N_p) G + 2k$	$(6 + 4N_p) G + 4k$
Liang ^[15]	$(3 + N_s) G $	$(4 + 2N_p) G + G_T + \text{sig} $	$(3 + 2N_p) G + G_T + \text{sig} + \text{sym} $
Luo ^[23]	$(2 + 3N_s) G $	$(2 + 2N_p) G + G_T $	$(1 + 2N_p) G + 2 G_T + N_s G $
Yang ^[22]	$(2 + 2N_s) G $	$4 G + G_T $	$(1 + 2N_p) G + 2 G_T $
Feng ^[4]	$(4 + 2N_s) G $	$(5 + 2N_p) G + G_T $	$(7 + 2N_p) G + 3 G_T $
Ge ^[24]	$(2 + N_s) G $	$(3 + 2N_p) G $	$(2 + 2N_p) G + 3 G_T $
本文方案	$(1 + N_s) G $	$(4 + 2N_p) G $	$(5 + 2N_p) G + G_T $

6 结束语

云计算环境存在大量密文流通、转换的场景，代理重加密技术可以在不解密密文的情况下实现访问策略的转换，从而实现对数据访问权限的传递。基于密文策略属性代理重加密方案大多只满足理想方案特性中的部分特性如：单向性、非交互性，而对可重复性、可控性和可验证性的研究不足。基于密文策略属性代理多重加密方法，一般只能解决理想方案特性中的一部分特征，如单向性、非交互性等，但对可重复性、可控性和可

验证性方面的研究还不够。管理方案一般在单属性授权中心下，面临着密钥泄露、中央授权机构权限过大且易出现性能瓶颈等问题，为分散中央授权机构的管理权力，减少工作量，引入了多属性管理中心管理用户不同的属性组，用户私钥由各个属性管理中心发放，在一定限度上增强了稳定性。因此，本文提出了支持重复可控特性的云计算多授权中心 CP-ABE 代理重加密方案，该方案在多属性授权中心框架下构建且支持大多数代理重加密方案不满足的重复性、可控性与可验证性，同时，为了降低客户端计算开销使用了外包

加解密技术, 利用云计算强大的计算能力, 分担了客户端的计算代价。且通过安全性证明, 本文方案达到了选择密文安全。

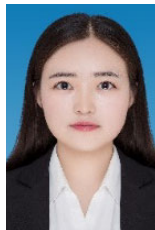
参考文献:

- [1] 张玉清, 王晓菲, 刘雪峰, 等. 云计算环境安全综述[J]. 软件学报, 2016, 27(6): 1328-1348.
ZHANG Y Q, WANG X F, LIU X F, et al. Survey on cloud computing security[J]. Journal of Software, 2016, 27(6): 1328-1348.
- [2] SUN P J. Privacy protection and data security in cloud computing: a survey, challenges and solutions [J]. IEEE Access, 2019, (99): 1.
- [3] BLAZE M. Divertible protocols and atomic proxy cryptography[J]. EUROCRYPT, 1998: 127-144.
- [4] 王悦, 樊凯. 隐藏访问策略的高效 CP-ABE 方案[J]. 计算机研究与发展, 2019, 56(10): 2151-2159.
WANG Y, FAN K. Effective CP-ABE with hidden access policy[J]. Journal of Computer Research and Development, 2019, 56(10): 2151-2159.
- [5] 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.
FENG CS, 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.
- [6] LIANG K, FANG L, SUSILO W, et al. A ciphertext-policy attribute-based proxy re-encryption with chosen-ciphertext security[C]//Proceedings of the 2013 5th International Conference on Intelligent Networking and Collaborative Systems, 2013.
- [7] 沈剑, 周天祺, 曹珍富. 云数据安全保护方法综述[J]. 计算机研究与发展, 2021, 58(10): 2079-2098.
SHEN J, ZHOU T Q, CAO Z F. Protection methods for cloud data security [J]. Journal of Computer Research and Development, 2021, 58(10): 2079-2098.
- [8] BETHENCOURT J, SAHAI A, WATERS B. Ciphertext-policy attribute-based encryption[C]//Proceedings of the 2007 IEEE Symposium on Security and Privacy (SP'07). 2007.
- [9] 苏锐, 吴彬, 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.
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.
- [10] 牛淑芬, 陈俐霞, 刘文科, 等. 电子邮件系统中支持关键字搜索的代理重加密方案[J]. 计算机工程, 2020, 46(6): 136-143.
NIU S F, CHEN L X, LIU W K, et al. Proxy re-encryption scheme supporting keyword search in email system[J]. Computer Engineering, 2020, 46(6): 136-143.
- [11] 牛淑芬, 刘文科, 陈俐霞, 等. 基于联盟链的可搜索加密电子病历数据共享方案 [J]. 通信学报, 2020, 41(8): 204-214.
NIU S F, LIU W K, CHEN L X, et al. Electronic medical record data sharing scheme based on searchable encryption via consortium blockchain[J]. Journal on Communications, 2020, 41(8): 204-214.
- [12] FUGKEAW S. A lightweight policy update scheme for outsourced personal health records sharing [J]. IEEE Access, 2021, 9: 54862-54871.
- [13] LUO F, AL-KUWARI S, WANG F, et al. Attribute-based proxy re-encryption from standard lattices[J]. Theoretical Computer Science, 2021, 865: 52-62.
- [14] 江明明, 郭宇燕, 余磊, 等. 有效的标准模型下格上基于身份的代理重加密[J]. 电子与信息学报, 2019, 41(1): 61-66.
JIANG M M, GUO Y Y, YU L, et al. Efficient identity-based proxy re-encryption on lattice in the standard model[J]. Journal of Electronics & Information Technology, 2019, 41(1): 61-66.
- [15] LIANG X H C Z F, LIN H, ET AL. Attribute based proxy re-encryption with delegating capabilities[C]//Proceedings of the ACM Symposium on Information, Computer and Communications Security. 2009.
- [16] LIANG K, 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 Computer Systems, 2015, 52: 95-108.
- [17] KAWAI Y. Outsourcing the re-encryption key generation: flexible ciphertext-policy attribute-based proxy re-encryption[C]//Information Security Practice and Experience. 2015: 301-315.
- [18] 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.
- [19] CHASE M. Multi-authority attribute based encryption[C]//Proceedings of the Lecture Notes in Computer Science. 2007.
- [20] CHEN D, WAN L, WANG C, et al. A Multi-authority attribute-based encryption scheme with pre-decryption[Z]. 2015 Seventh International Symposium on Parallel Architectures, Algorithms and Programming (PAAP). 2015: 223-228.
- [21] 关志涛, 杨亭亭, 徐茹枝, 等. 面向云存储的基于属性加密的多授权中心访问控制方案[J]. 通信学报, 2015, 36(6): 120-130.
GUAN Z T, YANG T T, XU R Z, et al. Multi-authority attribute-based encryption access control model for cloud storage[J]. Journal on Communications, 2015, 36(6): 120-130.
- [22] 仲红, 崔杰, 朱文龙, 等. 高效且可验证的多授权机构属性基加密方案[J]. 软件学报, 2018, 29(7): 2006-2017.
ZHONG H, CUI J, ZHU W L, et al. Efficient and verifiable multi-authority attribute based encryption scheme[J]. Journal of Software, 2018, 29(7): 2006-2017.
- [23] 杨小东, 杨苗苗, 刘婷婷, 等. 基于多授权中心属性基加密的多域云访问控制方案[J]. 计算机工程与科学, 2018, 40(7): 1192-1198.
YANG X D, YANG M M, LIU T T, et al. A multi-domain access

control scheme based on multi-authority attribute encryption for cloud storage[J]. Computer Engineering & Science, 2018, 40(7): 1192-1198.

- [24] 罗恩韬, 王国军, 陈淑红, 等. 移动社交网络中跨域代理重加密朋友发现隐私保护方案研究[J]. 通信学报, 2017, 38(10): 81-93.
- LUO E T, WANG G J, CHEN S H, et al. Privacy preserving friend discovery cross domain scheme using re-encryption in mobile social networks [J]. Journal on Communications, 2017, 38(10): 81-93.
- [25] GE C, SUSILO W, BAEK J, et al. A verifiable and fair attribute-based proxy re-encryption scheme for data sharing in clouds[J]. IEEE Transactions on Dependable and Secure Computing, 2021: 1.
- [26] GUO H, ZHANG Z, XU J, et al. Accountable proxy re-encryption for secure data sharing [J]. IEEE Transactions on Dependable and Secure Computing, 2021, 18(1): 145-159.

[作者简介]



刘尚(1995-), 女, 河北石家庄人, 太原科技大学硕士生, 主要研究方向为云计算与系统安全。



郭银章(1968-), 男, 山西原平人, 博士, 太原科技大学教授, 主要研究方向为协同计算与云计算、系统安全与隐私保护。



科技期刊世界影响力指数 (WJCI) 报告 (2021)

收录证书



This is certificate for

网络与信息安全学报

(ISSN: 2096-109X CN: 10-1366/TP)

to be indexed in

*World Journal Clout Index(WJCI) Report of
Scientific and Technological Periodicals(2021)*



项目联合研发单位 Project research units:

中国科学技术信息研究所 Institute of Scientific and Technical Information of China

《中国学术期刊 (光盘版)》电子杂志社有限公司 Chinese Academic Journal(CD Edition)

Electronic Publishing House Co.ltd

清华大学图书馆 Library of Tsinghua University

万方数据有限公司 Wanfang Data Co.Ltd

中国高校科技期刊研究会 Society of China University Journals

中国科学技术期刊编辑学会 China Editology Society of Science Periodicals

收录声明

本刊对发表的文章,拥有出版电子版、网络版版权,并拥有和其他网站交换信息的权利。本刊支付的稿酬中已经包含上述费用。

Chinese Journal of Network and Information Security has the copyright to publish electronic edition, online edition of the published articles, and has the right to exchange information with other sites. The expenses have been included in the fee paid by editorial department.

道德声明

本刊发表的论文是作者独立取得的原创性研究成果,无一稿多投;论文内容不涉及国家机密;未曾以任何形式用任何文种在国内外公开发表过;论文内容不侵犯他人著作权和其他权利。若发生一稿多投、侵权、泄密等问题,论文作者将承担全部责任。

The authors of *Chinese Journal of Network and Information Security* guarantee that their submitted articles are original and contain nothing confidential. The said article is only submitted to *Chinese Journal of Network and Information Security*. The said article has not been published before and has not been submitted elsewhere for print or electronic publication consideration. The said article is no way whatever a violation or an infringement of any existing copyright or license from the third party. Otherwise, the authors of the said article shall take the blame for the violation or infringement of the related copyright and the leakage of secrets.