

云计算中支持有效用户撤销的多授权方基于属性加密方案

张亮轩^{1,2}, 李晖^{1,2}

(1. 西安电子科技大学综合业务网理论与关键技术国家重点实验室, 陕西 西安 710071;
2. 西安电子科技大学网络与信息安全学院, 陕西 西安 710071)

摘要: 提出了一个支持有效用户撤销的多授权方基于属性加密方案, 每个用户都拥有一个“混合解密密钥”用于实现有效的用户撤销。与现有的方案相比, 该方案在进行用户撤销时更加有效。除此以外, 该方案还能实现用户的动态权限管理, 在云端进行部分数据外包加密, 密文部分解密, 在期限过后已分享的数据可自我销毁。

关键词: 云计算; 访问控制; 基于属性加密; 多授权方; 用户撤销

中图分类号: TP309.7

文献标识码: A

doi: 10.11959/j.issn.2096-109x.2016.00023

Multi-authority attribute-based encryption with efficient user revocation in cloud computing

ZHANG Liang-xuan^{1,2}, LI Hui^{1,2}

(1. State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an 710071, China;
2. School of Cyber Engineering, Xidian University, Xi'an 710071, China)

Abstract: An efficient user revocable multi-authority ABE scheme was proposed, a “hybrid decryption key” for each user is used to achieve efficient user revocation. Comparing with the existing schemes, this scheme is more effective in user revocation. In addition, the scheme can also achieve user dynamic privileges management, partial data outsource encryption and ciphertext partially decrypted in the cloud server and shared data self-destruction after expiration.

Key words: cloud computing, access control, attribute-based encryption, multi-authority, user revocation

1 引言

云计算作为一种革命性的计算模式, 计算资源可通过互联网动态提供给用户, 用户也把数据的存储和计算任务外包给云服务商, 因此, 用户不再被有限的资源和设备的物理位置所限制。当数据的拥有者将他们的数据放在云端, 数据的安全, 特别是敏感数据的内容安全, 就成了人们关注的焦点。例如一个病人将他的医疗数据放在云端, 该病人就失去对数据的控制, 他会关注云服务商是否会非法访问他的数据, 除了他的主治医

师外, 其他用户是否也会非法访问该数据。

人们提出各种各样的访问控制技术用来保护数据的内容安全。基于属性加密 (ABE, attribute-based encryption)^[1~5]就是一种确保云中数据安全的访问控制技术。与传统的公钥加密相比, 基于属性加密是一种一对多的灵活的加密方式。它有 2 个基本的形式: 密钥策略基于属性加密 (KP-ABE, key-policy ABE) 和密文策略基于属性加密 (CP-ABE, ciphertext-policy ABE)。在 KP-ABE 中, 密钥与访问策略相关而密文与属性相关。相反, 在 CP-ABE 中, 密钥与属性相关而

收稿日期: 2016-01-20; 修回日期: 2016-02-08。通信作者: 李晖, lihui@mail.xidian.edu.cn

基金项目: 国家自然科学基金资助项目 (No.U1401251, No.61272457)

Foundation Item: The National Natural Science Foundation of China (No.U1401251, No.61272457)

密文与访问策略相关。

随着越来越多的个人和企业将他们的数据外包给云服务商,用户的加入和撤销也相应变得越来越频繁。如何进行有效的用户撤销成为一个公开的问题。然而,在现有的基于属性加密方案中该问题并没有被完全地解决。当用户的撤销发生时,必须保证该用户不再具备解密能力,即使他所拥有的属性满足密文的访问策略。因此,在现有的方案中,数据的拥有者需要更新访问策略,授权方需要更新未撤销用户的属性密钥,这就对数据的拥有者和授权方造成了巨大的计算和通信开销。用户的撤销也会引发新的安全问题,例如已撤销用户可能会通过与未撤销用户合谋来恢复他的解密能力。这些促使人们应该提出可有效解决用户撤销问题的方案。

支持有效用户撤销的方案应满足以下要求。

1) 正确性。已撤销用户不应再有访问数据的能力,即使他们拥有与访问策略匹配的属性。同时,未撤销用户仍应具有访问数据的能力。

2) 安全性。用户的撤销不应打破访问控制方案的安全性,也不应引入新的安全问题。

3) 有效性。用户的撤销不应给整个系统带来过大的计算负担和通信开销。

本文主要解决多授权方基于属性加密方案中有效用户撤销问题。为了实现有效的用户撤销,将时间期限嵌入密文和解密密钥中,并利用了“混合解密密钥”。它是通过一个“与”门将公开的解密密钥和秘密的解密密钥相连接。其中,公开的解密密钥中包含属性部分和时间部分。当用户的撤销发生后,授权方只需更新未撤销用户公开解密密钥中的时间部分,与此同时,云服务商也只需更新密文中的时间期限。因此,用户的撤销不会给未撤销用户和数据的拥有者带来任何计算负担。此方案还具有前向安全性和后向安全性,即新加入的用户能够解密之前分享的数据,已撤销的用户不能再具有数据解密能力。

该方案还可以实现用户动态特权管理。当用户增加了一个属性,授权方只需为该用户产生一个新的公开解密密钥即可;当用户减少一个属性,授权方应更新该用户的解密密钥并且更新其他用户公开解密密钥中的时间部分,同时云服务商应

更新密文中的时间期限。

另外,该方案通过将部分加密和解密任务外包给云服务商,使数据拥有者的加密开销和用户的解密开销都大大降低。同时,还实现了在数据拥有者制定的时间期限过后已分享数据的自我销毁。

2 相关的工作

基于属性加密技术^[1~3,5]被认为是最适合进行云中数据访问控制的技术。它有2个基本的形式:密钥策略基于属性加密^[1,6]和密文策略基于属性加密^[2,7]。

在密钥策略基于属性加密中,密文和属性相关而密钥和访问策略相关。第一个KP-ABE方案是由Goyal等^[1]提出的,该方案中,数据拥有者应向可信的授权方进行询问以确定哪些属性应出现在密文中。Ostrovsky等^[8]提出了第一个支持非单调访问策略的KP-ABE方案。Yu等^[9]将代理重加密技术与KP-ABE技术结合,使数据的拥有者将大部分的计算量外包给代理,并且支持细粒度的访问控制。Tysowski等^[10]修改了基于属性加密并且利用重加密算法提出了新的保护移动用户的数据方案。

在密文策略基于属性加密中,密文和访问策略相关而密钥和属性相关。Bethencourt等^[2]提出了第一个CP-ABE方案,他们方案的不足之处在于安全性证明仅仅是在一般双线性群模型下完成的。为了克服这个不足,Cheung等^[7]提出了标准模型下的另一个结构。Waters使用线性秘密共享(LSSS, linear secret sharing scheme)矩阵作为访问结构,提出了在标准模型下有效的和可证明安全的CP-ABE方案^[3]。

文献[11,12]提出了多授权方的方案,每个用户都有一个全局身份用于授权方产生密钥。Chase等^[12]提出的方案被认为是基于陷门的KP-ABE,但缺乏加密策略的共性表达。之后,许多授权方基于属性加密方案被提出^[13~16]。但是,它们或者使用陷门基于属性加密^[13],或者有一个半诚信中心授权方^[14~16],或者不能抵抗用户的合谋攻击^[13]。

最近,一些能够解决用户撤销问题的基于属性加密方案^[17~21]被提出。在文献[17]中,用户撤销需要改变公/私钥中的最小属性集。在文献[18]中,用户撤销需改变密文中的秘密值,这样会引

起密文一系列的改变。在文献[19]中,用户撤销会通过更新密钥更新未撤销用户密钥和数据拥有者的密文。在文献[20,21]中,用户撤销会通过代理重加密密钥来更新未撤销用户密钥中的属性和密文中的属性。本文不考虑这些方案是因为它们会给数据拥有者和授权方造成巨大的计算和通信开销。本文提出了一个支持有效用户撤销的多授权方基于属性加密方案。与上述方案相比,当用户撤销发生时,本文提出的方案不仅大大减小了授权方的计算量,而且不会给数据拥有者和用户造成任何计算和通信开销,因此更加有效。

3 准备工作

3.1 密码学背景知识

定义 1 双线性对映射。设有 3 个素数 p 阶乘法循环群 G_1 、 G_2 、 G_T , g_1 是群 G_1 的生成元, g_2 是群 G_2 的生成元。 $e: G_1 \times G_2 \rightarrow G_T$ 是它们之间的一个映射, 满足以下条件。

- 1) 双线性: 对所有 $g_1 \in G_1, g_2 \in G_2$, 满足 $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$, 其中, $a, b \in Z_p$ 。
- 2) 非退化性: 存在 $g_1 \in G_1, g_2 \in G_2$ 使 $e(g_1, g_2) \neq 1$, 换句话说, 该映射不会将 $G_1 \times G_2$ 中任何一对元素映射到 G_T 中的单位元。
- 3) 可计算性: 对所有的 $g_1 \in G_1, g_2 \in G_2$, 存在计算 $e(g_1, g_2)$ 的高效算法。

定义 2 离散对数 (DL) 假设。随机选择 $a \in Z_p$, 输入 $g, g^a \in G$, 对于任何多项式时间的敌手 A_{DL} 来说, 输出 a 是计算上不可行的。它被定义为

$$\Pr[A_{DL}(g, g^a) = (a) : a \leftarrow Z_p] \leq \varepsilon$$

其中, ε 是可忽略的。

定义 3 判定性双线性 Diffie-Hellman (DBDH) 假设。随机选择 $a, b, c \in Z_p$, 输入 $g, g^a, g^b, g^c \in G$ 和 $P \in G_T$, 对于任何多项式时间的敌手 A_{DBDH} 来说, 判定 P 是否为 $e(g, g)^{abc}$ 的优势是可忽略的。它被定义为

$$\begin{aligned} & \Pr[A_{DBDH}(g, g^a, g^b, g^c, P = e(g, g)^{abc}) \\ & = 0 : a, b, c \leftarrow Z_p] = \frac{1}{2} + \varepsilon \end{aligned}$$

其中, ε 是可忽略的。

3.2 线性秘密共享方案

P 是一个属性集, M 是一个 $n \times l$ 的访问矩阵, $\rho: \{1, 2, \dots, n\} \rightarrow P$ 将矩阵中的每一行都映射到属性集 P 中的属性。如果一个秘密共享方案 Π 是基于访问矩阵 M 和映射 ρ 时, 并且包括 2 个多项式时间算法, 则可以被称为线性秘密共享方案 (LSSS)。

1) $\text{Share}_{(M, \rho)}$ 。随机选择一个数 $s \in Z_p$ 和一个向量 $v = (s, y_2, \dots, y_l) \in Z_p^l$, 其中, $y_2, \dots, y_l$ 被用来分享 s 。对于 $i = 1, 2, \dots, n$, 计算 $\lambda_i = M_i v$, 其中, M_i 是矩阵 M 的第 i 行向量, λ_i 对应于 $\rho(i)$ 在 P 中的属性。

2) $\text{Recon}_{(M, \rho)}$ 。 S 是一个满足访问策略 (M, ρ) 的属性集。定义集合 $I = \{i | \rho(i) \in S\}$, 输出 $\{(i, \mu_i)\}_{i \in I}$ 满足线性重构特性 $\sum_{i \in I} \mu_i \lambda_i = s$ 。

3.3 系统模型

一个拥有多授权方的云存储系统如图 1 所示。系统模型包括下列实体: 多个授权方、云服务商、数据拥有者和用户。

授权方: 每一个授权方与其他的授权方都是相互独立的, 负责在其属性域内管理用户的属性。授权方可以为自己产生一个公钥/私钥对, 为每一个用户的每个属性产生一个公开/秘密的解密密钥对。当发生用户撤销时, 能够为每一个未撤销用户更新公开的解密密钥。

云服务商: 云服务商为数据的拥有者存储数据, 为用户提供数据访问服务。云服务商负责为数据拥有者承担部分加密工作, 为用户承担部分解密工作。

数据拥有者: 数据拥有者制定访问策略并在这些访问策略下进行数据的加密, 也可以要求云服务商为储存在云中的加密数据更新访问策略。之后, 可以检查云服务商更新的策略是否正确。

用户: 每个用户被分配一个全局的身份, 能够自由地从云服务商那里得到密文。当用户的属性满足访问策略, 且解密密钥中的时间部分满足密文中的时间期限, 就可以成功解密密文。

3.4 方案框架

基于系统模型, 定义一个支持有效用户撤销的多授权方基于属性加密方案。

定义 4 方案框架。本文提出的方案包括下列算法: GlobalSetup 、 AuthoritySetup 、 OwnerSetup 、

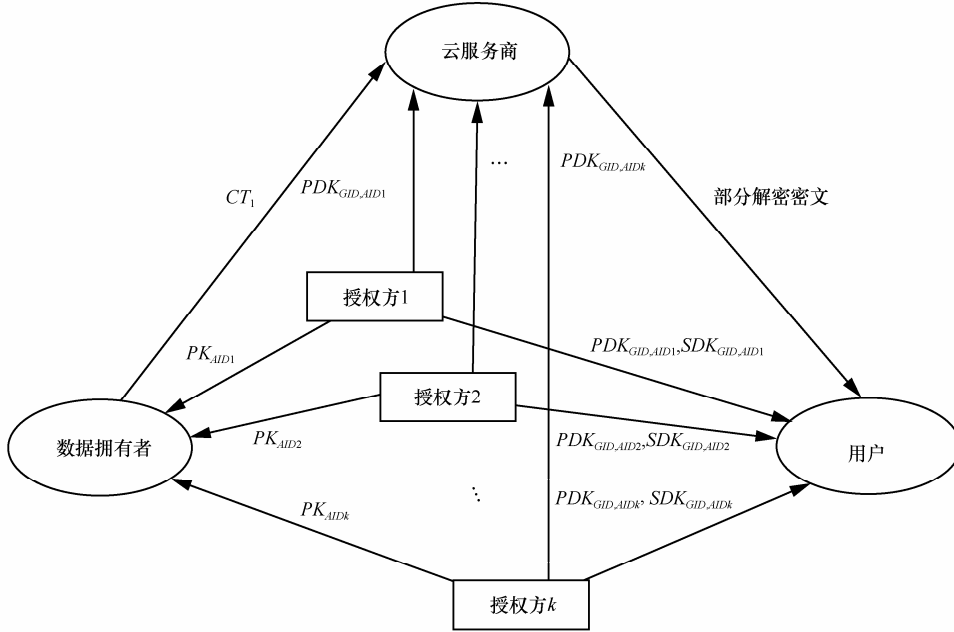


图1 拥有多授权方的云存储系统模型

UDKeyGen、Encrypt、Decrypt、UPDKeyUpdate、OCTUpdate、UKeyGen 和 CTUpdate。

1) GlobalSetup(λ) $\rightarrow$ GP 。全局的初始化算法输入一个安全参数 λ ，为整个系统输出一个全局参数 GP 。

2) AuthoritySetup(GP, S_{AID}) $\rightarrow$ (PK_{AID}, SK_{AID})。授权方的初始化算法在每一个授权方 AID 处运行。输入全局参数 GP 和授权方管理的属性域 S_{AID} ，输出其公/私钥对 (PK_{AID}, SK_{AID})。

3) OwnerSetup($GP, x_{GID'}$) $\rightarrow$ ($PK_{GID'}, SK_{GID'}$)。数据拥有者的初始化算法在每一个数据拥有者 GID' 处运行。输入全局参数 GP 和一个随机的元素 $x_{GID'} \in Z_p$ ，输出数据拥有者的公/私钥对 ($PK_{GID'}, SK_{GID'}$)。

4) UDKeyGen($GP, SK_{AID}, GID, S_{GID,AID}$) $\rightarrow$ ($PDK_{GID,AID}, SDK_{GID,AID}$)。每一个授权方 AID 输入全局的参数 GP 、私钥 SK_{AID} 、用户的全局身份 GID 和该用户在其属性管理域内的属性 $S_{GID,AID}$ 。每一个授权方 AID 分别运行公开/秘密的解密密钥产生算法用来为用户 GID 产生公开/秘密的解密密钥对 $\{PDK_{GID,AID}, SDK_{GID,AID}\}$ 。最后，每一个授权方 AID 发送 $\{GID, PDK_{GID,AID}\}$ 给云服务商，发送 $\{PDK_{GID,AID}, SDK_{GID,AID}\}$ 给用户 GID 。

5) Encrypt($PK_{AID}, GP, SK_{GID'}, m, (M, \rho), T_i$) $\rightarrow$ CT 。加密算法分为2个阶段。第1阶段是

数据拥有者 GID' 进行数据加密，输入相关授权方的公钥 PK_{AID} 、全局参数 GP 、自己的私钥 $SK_{GID'}$ 、消息 m 、还有访问策略 (M, ρ) 。输出加密信息 $EnInfo(m)$ 、部分密文 CT_1 和访问矩阵 M 。第2阶段是云服务商进行数据加密，输入全局参数 GP 、第1阶段的输出 $\{EnInfo(m), CT_1 \| M\}$ 、时间期限 T_i ；输出另一部分密文 CT_2 。最终，完整的密文 CT 是2阶段输出密文之和 ($CT_1 \| CT_2$)。

6) Decrypt($CT, GP, PK_{GID'}, \{GID, PDK_{GID,AID}\}, \{SDK_{GID,AID}\}$) $\rightarrow$ m 。解密算法也分为2个阶段。第1阶段是云服务商进行解密，输入密文 CT 、全局参数 GP 、数据拥有者的公钥 $PK_{GID'}$ 、用户公开的解密密钥集 $\{GID, PDK_{GID,AID}\}$ 。第2阶段是用户 GID 进行解密，输入秘密的解密密钥 $SDK_{GID,AID}$ 和第1阶段的输出。2个阶段解密完成后，如果用户解密密钥中的属性部分满足密文中的访问策略，解密密钥中的时间部分满足密文中的时间期限，就可以得到消息 m 。

另外，2个算法用来实现有效的用户撤销。多授权方更新未撤销用户公开的解密密钥中的时间部分，云服务商更新密文中的时间期限。

1) UPDKeyUpdate($GP, \{GID_1, GID_2, \dots, GID_m\}, T_{i+1}$) $\rightarrow$ TK_{T_{i+1}, GID_j} 。用户公开的解密密钥更新算法在多授权方处运行。输入全局参数 GP 、未撤销用户的全局身份集 $\{GID_1, GID_2, \dots,$

$GID_m\}$ 、要更新的时间期限 T_{i+1} 。输出未撤销用户更新后的公开的解密密钥中时间部分 TK_{T_{i+1}, GID_j} , $j \in \{1, 2, \dots, m\}$ 。

2) OCTUpdate($GP, EnInfo(m), T_{i+1}$) $\rightarrow C_{T_{i+1}}$ 。数据拥有者的密文更新算法在云服务商处运行。输入全局参数 GP 、消息 m 的加密信息 $EnInfo(m)$ 、要更新的时间期限 T_{i+1} 。输出更新后的部分密文 $C_{T_{i+1}}$ 。

除此之外, 2 个算法用来实现访问策略的更新, 即将数据拥有者密文中之前的访问策略 (M, ρ) 更新到新的访问策略 (M', ρ') 。

1) UKeyGen($PK_{AID}, EnInfo(m), (M, \rho), (M', \rho')$) $\rightarrow UK_m$ 。更新密钥产生算法在数据拥有者处运行。输入相关授权方的公钥 PK_{AID} 、消息 m 的加密信息 $EnInfo(m)$ 、之前的访问策略 (M, ρ) 以及新的访问策略 (M', ρ') 。输出消息 m 的更新密钥 UK_m 。

2) CTUpdate(CT, UK_m) $\rightarrow CT'$ 。密文的更新算法在云服务商处运行。输入之前的密文 CT 和更新密钥 UK_m 。输出新的在访问策略 (M', ρ') 下的密文 CT' 。

因为方案中数据拥有者的访问策略更新和大多数现有的基于属性加密方案中数据拥有者访问策略更新相似, 更多的细节可以参考文献[22], 本文将不再进行介绍。

3.5 安全模型

假设在系统中云服务商是好奇的。具体而言, 好奇的云服务商可能基于其所拥有的资源去发现更多的信息, 并且可能与已撤销用户合谋。然而, 云服务商也会诚实地遵循协议运行。数据拥有者是完全可信的, 用户是不诚信的, 例如, 用户之间可能合谋去非法访问未经授权的数据。授权方可能会受到敌手的攻击, 假设敌手的攻击是静态的, 但是, 解密密钥请求是适应性的。

在本方案中考虑 2 类敌手。

类型 I 敌手: 定义为一个好奇的云服务商, 在时间期限 T^* 内从目标数据拥有者 GID' 的密文中获得有用的信息。这样的敌手拥有未撤销用户 GID 的公开解密密钥, 或通过已撤销用户合谋获得该用户秘密的解密密钥。为了使这样的攻击合理, 必须限制 $GID \neq GID'$ 。

类型 II 敌手: 定义为一个敌手能够在时间期限 T^* 内通过攻击一些授权方获得数据拥有者

GID' 的有用信息。这样的敌手能够得到被攻击的授权方给用户的公开/秘密解密密钥和未被攻击的授权方给用户的公开解密密钥。

3.5.1 类型 I 敌手的安全游戏

初始化阶段 运行全局的初始化算法, 授权方的初始化算法和数据拥有者的初始化算法。挑战者获得全局参数 GP , 多授权方的公/私钥对 $\{PK_{AID}, SK_{AID}\}$ 和数据拥有者 GID' 的公/私钥对 $\{PK_{GID'}, SK_{GID'}\}$ 。

第 1 阶段 挑战者初始化 2 个空的列表 S 和 S' , 提供给敌手下列谕言。

1) 公开解密密钥提取谕言。当收到全局身份 GID 和时间期限 T_j , 通过运行用户解密密钥产生算法获得 $PDK_{GID, AID}$ 和 $SDK_{GID, AID}$, 输出公开解密密钥 $PDK_{GID, AID}$, 并将全局身份 GID 加入列表 S 中 $S = S \cup \{GID\}$ 。

2) 公开解密密钥更新谕言。当收到全局身份 GID 和时间期限 T^* , 如果 $GID \in S$, 运行公开解密密钥更新算法获得并输出 TK_{GID, T^*} 。将全局身份 GID 加入列表 S' 中 $S' = S' \cup \{GID\}$ 。

3) 秘密解密密钥提取谕言。当收到全局身份 GID , 如果 $GID \in S - S'$, 输出用户的秘密解密密钥 $SDK_{GID, AID}$ 。

4) 解密谕言。当接收到 $PDK_{GID, AID}$ 、 $SDK_{GID, AID}$ 和 CT , 运行解密算法并输出结果 m 。

挑战 敌手给出 2 个等长的消息 m_0 、 m_1 和访问控制结构 (M^*, ρ^*, T^*) 。挑战者随机抛一枚硬币 $\mu \in \{0, 1\}$, 硬币正面朝上则加密消息 m_0 , 反面朝上则加密消息 m_1 。最终, 挑战者在敌手给出的访问控制结构 (M^*, ρ^*, T^*) 下加密消息 m_μ , 并将得到的密文 CT^* 发送给敌手。

第 2 阶段 重复第 1 阶段。

猜测 敌手输出对 μ 的猜测 μ' 。

在这个游戏中敌手 A 猜中的概率为

$$\Pr[\mu' = \mu] = \frac{1}{2}$$

3.5.2 类型 II 敌手的安全游戏

初始化阶段 运行全局的初始化算法和数据拥有者的初始化算法。敌手列出被攻击的授权方 $S'_A \subset S_A$, 挑战者通过运行授权方的初始化算法产生授权方的公/私钥对, 对于未被攻击的授权方 $S_A - S'_A$ 挑战者仅仅发送公钥给敌手, 对于被攻击的授权方 S'_A 挑战者发送公钥和私钥给敌手。

第1阶段 提供给敌手下列谕言。

1) 公开的解密密钥提取谕言。当收到全局身份 GID 、时间期限 T^* 还有属性集 $S_{GID,AID}$ 对, 其中, 属性集 $S_{GID,AID}$ 中的属性属于未被攻击的授权方 $S_A - S'_A$ 的属性域。挑战者将对应的公开解密密钥 $PDK_{GID,AID}$ 发送给敌手。

2) 解密谕言。当收到未被攻击的授权方产生的公开解密密钥 $PDK_{GID,AID}$ 、被攻击的授权方产生的公开/秘密解密密钥 $PDK_{GID,AID}$ 、 $SDK_{GID,AID}$ 和密文 CT 时, 运行解密算法并输出结果 m 。

挑战 敌手给出 2 个等长的消息 m_0, m_1 和访问控制结构 (M^*, ρ^*, T^*) 。挑战者随机抛一枚硬币 $\mu \in \{0, 1\}$, 硬币正面朝上则加密消息 m_0 , 反面朝上则加密消息 m_1 。最终, 挑战者在敌手给出的访问控制结构 (M^*, ρ^*, T^*) 下加密消息 m_μ , 并将得到的密文 CT^* 发送给敌手。

第2阶段 重复第1阶段。

猜测 敌手输出对 μ 的猜测 μ' 。

在这个游戏中敌手 A 猜中的概率为

$$\Pr[\mu' = \mu] = \frac{1}{2}$$

定义 5 支持有效用户撤销的多授权方基于属性加密方案对于适应性的选择密文攻击是语义安全的, 即在类型 I 和类型 II 敌手的安全游戏中, 没有多项式界的敌手对于挑战者有不可忽略的优势。

4 支持有效用户撤销的多授权方基于属性加密方案

4.1 简要描述

本文将采用密文策略基于属性加密构造支持有效用户撤销的多授权方基于属性加密方案。

为了实现有效的用户撤销: 1) 将时间期限嵌入解密密钥中, 并利用了“混合解密密钥”, 它是一个“与”门将公开的解密密钥和秘密的解密密钥相连接。公开的解密密钥中包含属性部分和时间部分, 秘密的解密密钥中只包含全局身份部分; 2) 在解密阶段, 用数据拥有者的访问策略和密文中的时间期限一起去限制数据的解密, 只有用户解密密钥中的属性部分满足访问策略, 解密密钥中时间部分满足密文中的时间期限, 该用户才能成功解密。这样, 当用户撤销发生时, 只需通过更新未撤销用户解密密钥

中的时间部分和密文中的时间期限即可完成有效的用户撤销。

然而, 已撤销用户可能通过与未撤销用户合谋重建他的解密能力。具体而言, 已撤销用户希望得到未撤销用户解密密钥中的时间部分, 使其在新的时间期限内仍具有解密能力。为了抵抗这种合谋, 在本文提出的方案中, 多授权方会为每一个用户 GID 分别选择随机数 $r_{GID,1} \in Z_p$, 并计算 $r_{GID,2} = 1 - r_{GID,1} \bmod (p-1)$ 。 $r_{GID,1}$ 被嵌入该用户公开解密密钥中的属性部分, 而 $r_{GID,2}$ 被嵌入该用户公开解密密钥中的时间部分, 它们确定了一种“对应关系”。假设用户在时间 T_i 内撤销, 即使他能获得未撤销用户 GID 公开解密密钥中的时间部分 $TK_{T_{i+1}, GID}$ 和该用户秘密的解密密钥, 他仍然不能在时间期限 T_{i+1} 内解密密文。

综上, 已撤销用户不可能通过与未撤销用户合谋重新获得解密能力。

4.2 具体的方案

4.2.1 系统的初始化

系统的初始化包括 3 个阶段: 全局的初始化、授权方的初始化和数据拥有者的初始化。

1) 全局的初始化

在全局的初始化阶段, 选择 3 个素数 p 阶乘法群 G_1 、 G_2 和 G_T , g_1 、 g_2 分别是群 G_1 、 G_2 的生成元。3 个群之间存在一个双线性对运算 $e: G_1 \times G_2 \rightarrow G_T$ 。再选择 2 个散列函数 $H_1: \{0, 1\}^* \rightarrow G_1$, $H_2: \{0, 1\}^* \rightarrow G_2$ 。全局参数为 $GP = \{p, g_1, g_2, H_1, H_2\}$ 。

2) 授权方的初始化

每个授权方 AID 运行授权方的初始化算法产生他的公/私钥对。 S_{AID} 定义为授权方 AID 管理域内的所有属性, 对每一个属性 $x \in S_{AID}$, 授权方选择 2 个随机的数 $\alpha_x, \beta_x \in Z_p$, 公开其公钥 $PK_{AID} = \{x, e(g_1, g_2)^{\alpha_x}, g_2^{\beta_x}\}_{x \in S_{AID}}$, 保留其私钥 $SK_{AID} = \{\alpha_x, \beta_x\}_{x \in S_{AID}}$ 。

3) 数据拥有者的初始化

每个数据拥有者 GID' 运行数据拥有者的初始化算法去产生他的公/私钥对。数据拥有者 GID' 选择一个随机数 $x_{GID'} \in Z_p$, 公布其公钥 $PK_{GID'} = g_2^{x_{GID'}}$, 保留其私钥 $SK_{GID'} = x_{GID'}$ 。

4.2.2 解密密钥的产生

对于用户 GID , 每个授权方 AID 首先选择 2

个随机数 r_{GID} 、 $r_{GID,1} \in Z_p$ ，并计算 $r_{GID,2} = 1 - r_{GID,1} \bmod (p-1)$ 。接下来，为该用户在其属性域内分配属性 $S_{GID,AID}$ ，并运行解密密钥产生算法产生公开/秘密的解密密钥对

$$\begin{aligned} PDK_{GID,AID} &= \{K_{x,GID} = (k_{x,1}, k_{x,2}), TK_{T_i,GID}\} \\ &= \{k_{x,1} = g_1^{\beta_x}, k_{x,2} = g_1^{\alpha_x} g_1^{r_{GID,1}} (H_1(GID))^{\beta_x} \\ TK_{T_i,GID} &= g_2^{r_{GID,2}} (H_2(T_i))^{r_{GID}}\} \\ SDK_{GID,AID} &= \{K_{GID} = g_1^{r_{GID}}\} \end{aligned}$$

其中，公开的解密密钥 $PDK_{GID,AID}$ 中包含属性部分 $K_{x,GID}$ 和时间部分 $TK_{T_i,GID}$ ，秘密的解密密钥 $SDK_{GID,AID}$ 中只包含全局身份部分 K_{GID} 。每个授权方 AID 为用户 GID 产生的公开的解密密钥 $PDK_{GID,AID}$ 和秘密的解密密钥 $SDK_{GID,AID}$ 通过随机数 r_{GID} 将其对应起来，它们共同组成一个解密密钥集合。

最后，授权方 AID 发送 $\{GID, PDK_{GID,AID}\}$ 给云服务商，发送 $\{PDK_{GID,AID}, SDK_{GID,AID}\}$ 给用户 GID 。

4.2.3 数据加密

数据的加密分为2个阶段。

第1阶段是数据所有者 GID' 对数据进行加密，输入全局参数 GP 、私钥 $SK_{GID'}$ 、相关授权方的公钥 PK_{AID} 、数据 m 、一个 $n \times l$ 访问矩阵 M 和映射 ρ 。接下来，选择一个随机的加密指数 $s \in Z_p$ 和向量 $v = (s, y_2, \dots, y_l) \in Z_p^l$ 并计算 $\lambda_i = M_i \cdot v$ ，其中， $y_2, \dots, y_l$ 被用于分享加密指数 s ， M_i 表示矩阵 M 的第 i 行。然后，又选择一个向量 $w \in Z_p^l$ 并计算 $w_i = M_i \cdot w$ ，其中，向量 w 需满足它的第一项为0，其余 $(l-1)$ 项任意。最后，数据所有者选择一个随机数 $r \in Z_p$ 作为数据 m 的加密信息 $EnInfo(m)$ ，计算密文

$$\begin{aligned} CT_1 &= (C = me(g_1, g_2)^s, \forall i = 1, \dots, n, \\ C_{1,i} &= e(g_1, g_2)^{\lambda_i} e(g_1, g_2)^{\alpha_{\rho(i)} r}, \\ C_{2,i} &= g_2^{w_i} g_2^{\beta_{\rho(i)} r}, C_{3,i} = g_2^{\beta_{\rho(i)} x_{GID'}}) \end{aligned}$$

并发送 $\{EnInfo(m), CT_1 \| M\}$ 给云服务商。

第2阶段是云服务商对数据进行加密。收到第1阶段的输出后，云服务商计算另一部分密文

$$CT_2 = (C_4 = g_1^r, C_5 = g_2^r, C_{T_i} = (H_2(T_i))^r)$$

最终的密文为

$$\begin{aligned} CT &= (C = me(g_1, g_2)^s, \forall i = 1, \dots, n, \\ C_{1,i} &= e(g_1, g_2)^{\lambda_i} e(g_1, g_2)^{\alpha_{\rho(i)} r}, \\ C_{2,i} &= g_2^{w_i} g_2^{\beta_{\rho(i)} r}, C_{3,i} = g_2^{\beta_{\rho(i)} x_{GID'}}, \\ C_4 &= g_1^r, C_5 = g_2^r, C_{T_i} = (H_2(T_i))^r) \end{aligned}$$

4.2.4 数据解密

数据的解密分为2个阶段。

第1阶段是云服务商进行数据解密，首先用户 GID 发送解密请求给云服务商。接下来云服务商从随机谕言中获得 $H_1(GID)$ ，找到属于用户 GID 的公开解密密钥 $\{GID, PDK_{GID,AID}\}$ ，数据所有者 GID' 的公钥 $PK_{GID'}$ 。接下来，云服务商通过检验 $e(k_{\rho(i),1}, pk_{GID'}) = e(g_1, C_{3,i})$ 是否成立，确定用户 GID 的属性是否满足密文中的访问策略。若成立，云服务商计算

$$\begin{aligned} &\frac{C_{1,i} e(H_1(GID), C_{2,i}) e(C_4, g_2)}{e(k_{\rho(i),2}, C_5) e(C_4, TK_{T_i,GID})} \\ &= \frac{e(g_1, g_2)^{\lambda_i} e(H_1(GID), g_2)^{w_i}}{e(g_1^r, (H_2(T_i))^{r_{GID}})} = CI_i \end{aligned}$$

并发送 $\{(C, C_{T_i}) \| M, \{TK_{T_i,GID}, CI_i\}_{i=1,2,\dots,n}\}$ 给用户 GID 。若不成立，云服务商输出 $\perp$ ，并发送给用户 GID 。

第2阶段是用户 GID 进行解密，收到第1阶段的输出后，用户首先通过 $TK_{T_i,GID}$ 找到与之对应的秘密的解密密钥 $SDK_{GID,AID} = \{K_{GID}\}$ ，并计算

$$CI_i e(K_{GID}, C_{T_i}) = e(g_1, g_2)^{\lambda_i} e(H_1(GID), g_2)^{w_i}$$

接着，选择常数 $c_i \in Z_p$ 满足 $\sum_i c_i M_i = (1, 0, \dots, 0)$ ，并计算

$$\prod_i (e(g_1, g_2)^{\lambda_i} e(H_1(GID), g_2)^{w_i})^{c_i} = e(g_1, g_2)^s$$

最后，用户通过计算 $m = \frac{C}{e(g_1, g_2)^s}$ 成功解密数据。

4.2.5 用户撤销

有效的用户撤销包括2个阶段：未撤销用户公开解密密钥的更新和数据所有者密文的更新。

1) 未撤销用户公开解密密钥的更新

当用户撤销发生时，多授权方运行未撤销用

户公开的解密密钥更新算法。输入未撤销用户的全局身份集 $\{GID_1, GID_2, \dots, GID_m\}$ ，要更新的时间期限 T_{i+1} 。接下来，授权方为每一个未撤销用户更新公开解密密钥中的时间部分 $TK_{T_{i+1}, GID_j} = g_2^{r_{GID_j,2}} (H_2(T_{i+1}))^{r_{GID_j}}, j \in \{1, 2, \dots, m\}$ ，并发送 $\{GID_j, TK_{T_{i+1}, GID_j}\}$ 给云服务商，发送 TK_{T_{i+1}, GID_j} 给用户 GID_j 。

2) 数据拥有者密文的更新

与此同时，云服务商输入所有密文的加密信息 $\{EnInfo(m_1), EnInfo(m_2), \dots, EnInfo(m_n)\}$ 和需要更新的时间期限 T_{i+1} ，更新所有密文的时间期限 $C_{T_{i+1}} = (H_2(T_{i+1}))^{r_j}, j \in \{1, 2, \dots, n\}$ 。

5 方案分析

该方案可实现用户动态特权管理，在时间期限过后已分享数据的自我销毁。另外，本文将对方案的正确性、安全性进行证明，对方案的性能进行分析。

5.1 用户动态特权管理

本文提出的方案可以实现用户动态特权管理。

当用户 GID 增加一个属性 $x \in S_{AID}$ ，授权方 AID 只需运行解密密钥产生算法，为这个用户的属性 x 产生一对新的公开/秘密解密密钥对

$$\begin{aligned} PDK_{GID, AID} &= \{K_{x, GID}, TK_{T_i, GID}\} \\ &= \{k_{x,1} = g_1^{\beta_x}, k_{x,2} = g_1^{\alpha_x} g_1^{r_{GID,1}} (H_1(GID))^{\beta_x} \\ TK_{T_i, GID} &= g_2^{r_{GID,2}} (H_2(T_i))^{r_{GID}}\} \\ SDK_{GID, AID} &= \{K_{GID} = g_1^{r_{GID}}\} \end{aligned}$$

当用户 GID 减少或删除一个属性 $x' \in S_{AID}$ ，授权方 AID 首先选择一个随机的数 $r'_{GID,1} \in Z_p$ ，并计算 $r'_{GID,2} = 1 - r'_{GID,1} \bmod (p-1)$ 。接着为用户计算 $k_{x',2} = g_1^{\alpha_{x'}} g_1^{r'_{GID,1}} (H_1(GID))^{\beta_{x'}}$ 和 $TK_{T_{i+1}, GID} = g_2^{r'_{GID,2}} (H_2(T_{i+1}))^{r'_{GID}}$ ，其中 $\alpha_{x'}, \beta_{x'}$ 对应于 $\{S_{AID} - x'\}$ 中的属性。其他授权方只需将该用户公开解密密钥中的时间部分更新到时间期限 T_{i+1} 。对于其他的用户 GID_j ，所有授权方都应为他们更新公开解密密钥中的时间部分 $TK_{T_{i+1}, GID_j} = g_2^{r_{GID_j,2}} (H_2(T_{i+1}))^{r_{GID_j}}$ 。与此同时，云服务商应更新数据拥有者密文的时间期限 $C_{T_{i+1}} = (H_2(T_{i+1}))^r$ 。

通过以上步骤，用户的动态特权管理就可以实现了。

5.2 时间期限过后已分享数据的自我销毁

因为在密文和解密密钥中都有时间期限，所以可以利用它来控制数据的自我销毁。数据的拥有者可以提前确定数据销毁的时间期限 T_i ，一旦超过该时间期限云服务商将不再对密文进行更新，该密文将自我销毁。

5.3 正确性证明

定理 1 本文提出的访问控制方案是正确的。

证明 为了解密密文，云服务商首先检验 $e(k_{\rho(i),1}, PK_{GID'}) = e(g_1, C_{3,i})$ 是否正确。若不正确，则输出 $\perp$ 。

$$\begin{aligned} e(k_{\rho(i),1}, PK_{GID'}) &= e(g_1^{\beta_{\rho(i)}}, g_2^{x_{GID'}}) \\ &= e(g_1, g_2^{\beta_{\rho(i)} x_{GID'}}) \\ &= e(g_1, C_{3,i}) \end{aligned}$$

接下来，云服务商计算

$$\begin{aligned} &\frac{C_{1,i} e(H_1(GID), C_{2,i}) e(C_4, g_2)}{e(k_{\rho(i),2}, C_5) e(C_4, TK_{T_i, GID})} \\ &= \frac{e(g_1, g_2)^{\lambda_i} e(g_1, g_2)^{\alpha_{\rho(i)} r} e(H_1(GID), g_2^{w_i} g_2^{\beta_{\rho(i)} r}) e(g_1^r, g_2)}{e(g_1^{\alpha_{\rho(i)}} g_1^{r_{GID,1}} H_1(GID)^{\beta_{\rho(i)}}, g_2^r) e(g_1^r, g_2^{r_{GID,2}} (H_2(T_i))^{r_{GID}})} \\ &= \frac{e(g_1, g_2)^{\lambda_i} e(g_1, g_2)^{\alpha_{\rho(i)} r} e(H_1(GID), g_2^{w_i}) e(H_1(GID), g_2^{\beta_{\rho(i)} r}) e(g_1^r, g_2)}{e(g_1^{\alpha_{\rho(i)}} g_2^r) e(g_1^{r_{GID,1}}, g_2^r) e(H_1(GID)^{\beta_{\rho(i)}}, g_2^r) e(g_1^r, g_2^{r_{GID,2}} (H_2(T_i))^{r_{GID}})} \\ &= \frac{e(g_1, g_2)^{\lambda_i} e(H_1(GID), g_2)^{w_i}}{e(g_1^r, (H_2(T_i))^{r_{GID}})} \end{aligned}$$

接收到云服务商的输出后，用户计算

$$\begin{aligned}
& \frac{e(g_1, g_2)^{\lambda_i} e(H_1(GID), g_2)^{w_i}}{e(g_1^r, (H_2(T_i))^{r_{GID}})} e(K_{GID}, C_{T_i}) \\
&= \frac{e(g_1, g_2)^{\lambda_i} e(H_1(GID), g_2)^{w_i}}{e(g_1^r, (H_2(T_i))^{r_{GID}})} e(g_1^{r_{GID}}, (H_2(T_i))^r) \\
&= e(g_1, g_2)^{\lambda_i} e(H_1(GID), g_2)^{w_i}
\end{aligned}$$

接下来, 用户选择常数 $c_i \in Z_p$ 满足 $\sum_i c_i M_i = (1, 0, \dots, 0)$ 。又由于 $\lambda_i = M_i v$, $w_i = M_i w$, 故

$$\begin{aligned}
& \prod_i (e(g_1, g_2)^{\lambda_i} e(H_1(GID), g_2)^{w_i})^{c_i} \\
&= e(g_1, g_2)^{\sum_i c_i M_i v} \cdot e(H_1(GID), g_2)^{\sum_i c_i M_i w} \\
&= e(g_1, g_2)^s \\
& \frac{C}{e(g_1, g_2)^s} = \frac{m e(g_1, g_2)^s}{e(g_1, g_2)^s} = m
\end{aligned}$$

定理 2 当用户撤销发生后, 本文提出的访问控制方案仍然是正确的。

证明 当用户撤销发生后, 未撤销用户可以更新他们的公开解密密钥

$$\begin{aligned}
PDK_{GID, AID} &= \{K_{x, GID}, TK_{T_{i+1}, GID}\} \\
&= \{k_{x,1} = g_1^{\beta_x}, k_{x,2} = g_1^{\alpha_x} g_1^{r_{GID,1}} (H_1(GID))^{\beta_x} \\
TK_{T_{i+1}, GID} &= g_2^{r_{GID,2}} (H_2(T_{i+1}))^{r_{GID}}\}
\end{aligned}$$

但是, 已撤销用户不能够更新他们的公开解密密钥。

由于, 云服务商会更新数据拥有者的密文

$$CT = (C = me(g_1, g_2)^s, \forall i = 1, \dots, n,$$

$$C_{1,i} = e(g_1, g_2)^{\lambda_i} e(g_1, g_2)^{\alpha_{\rho(i)} r},$$

$$C_{2,i} = g_2^{w_i} g_2^{\beta_{\rho(i)} r}, C_{3,i} = g_2^{\beta_{\rho(i)} x_{GID}},$$

$$C_4 = g_1^r, C_5 = g_2^r, C_{T_{i+1}} = (H_2(T_{i+1}))^r)$$

对于未撤销用户, 他们仍然可以如定理 1 那样解密密文。

但是, 对于已撤销用户, 他们不能够再解密密文。因为他们接收到云服务商的输出

$$\begin{aligned}
& \frac{e(g_1, g_2)^{\lambda_i} e(H_1(GID), g_2)^{w_i}}{e(g_1^r, (H_2(T_i))^{r_{GID}})} \text{ 后, 接下来计算} \\
& \frac{e(g_1, g_2)^{\lambda_i} e(H_1(GID), g_2)^{w_i}}{e(g_1^r, (H_2(T_i))^{r_{GID}})} e(K_{GID}, C_{T_{i+1}}) \\
&= e(g_1, g_2)^{\lambda_i} e(H_1(GID), g_2)^{w_i} \frac{e(g_1^{r_{GID}}, (H_2(T_{i+1}))^r)}{e(g_1^r, (H_2(T_i))^{r_{GID}})}
\end{aligned}$$

他们仍然可以选择常数 $c_i \in Z_p$, 满足 $\sum_i c_i M_i = (1, 0, \dots, 0)$, 并计算

$$\begin{aligned}
& \prod_i (e(g_1, g_2)^{\lambda_i} e(H_1(GID), g_2)^{w_i} \frac{e(g_1^{r_{GID}}, (H_2(T_{i+1}))^r)}{e(g_1^r, (H_2(T_i))^{r_{GID}})})^{c_i} \\
&= e(g_1, g_2)^s \frac{e(g_1^{r_{GID}}, (H_2(T_{i+1}))^r)^{c_i}}{e(g_1^r, (H_2(T_i))^{r_{GID}})^{c_i}} \\
& \quad \frac{C}{e(g_1, g_2)^s \frac{e(g_1^{r_{GID}}, (H_2(T_{i+1}))^r)^{c_i}}{e(g_1^r, (H_2(T_i))^{r_{GID}})^{c_i}}} \\
&= m \frac{e(g_1^{r_{GID}}, (H_2(T_{i+1}))^r)^{c_i}}{e(g_1^r, (H_2(T_i))^{r_{GID}})^{c_i}} \neq m
\end{aligned}$$

从计算的结果可以得出, 已撤销用户不能够正确地解密密文。

5.4 安全性证明

本文提出的方案不仅满足正确性要求还满足安全性要求。它不仅不会破坏现有访问控制系统的安全性, 也不会引入新的安全问题。

接下来, 将证明该方案不会给云服务商泄露任何数据拥有者的访问策略, 实现了访问策略隐藏, 也不会给云服务商泄露任何用户的属性信息。它还能够阻止好奇的云服务商基于他所拥有的未撤销用户的公开解密密钥, 或通过与已撤销用户合谋非法访问数据。当授权方遭受到敌手的攻击时, 本文的方案仍然是安全的。

定理 3 本文提出的访问控制方案不会给云服务商泄露任何数据拥有者的访问策略, 实现了访问策略隐藏, 也不会给云服务商泄露任何用户的属性信息。

证明 在本文的访问控制方案中, 好奇的云服务商拥有数据拥有者的密文

$$CT = (C = me(g_1, g_2)^s, \forall i = 1, \dots, n,$$

$$C_{1,i} = e(g_1, g_2)^{\lambda_i} e(g_1, g_2)^{\alpha_{\rho(i)} r}$$

$$C_{2,i} = g_2^{w_i} g_2^{\beta_{\rho(i)} r}, C_{3,i} = g_2^{\beta_{\rho(i)} x_{GID}},$$

$$C_4 = g_1^r, C_5 = g_2^r, C_{T_i} = (H_2(T_i))^r)$$

以及访问矩阵 M 。

然而, 好奇的云服务商却不能从密文中计算出 $\alpha_{\rho(i)}$ 和 $\beta_{\rho(i)}$, 并且云服务商只拥有访问矩阵 M , 并不知道完整的访问策略 (M, ρ) 。故本文的访问控制方案没有给云服务商泄露任何数据拥有者的访问策略, 实现了访问策略的隐藏。

好奇的云服务商还拥有用户的公开的解密密钥

$$PDK_{GID, AID} = \{K_{x, GID}, TK_{T_i, GID}\}$$

$$= \{k_{x,1} = g_1^{\beta_x}, k_{x,2} = g_1^{\alpha_x} g_1^{r_{GID,1}} (H_1(GID))^{\beta_x},$$

$$TK_{T_i, GID} = g_2^{r_{GID,2}} (H_2(T_i))^{r_{GID}}\}$$

然而，好奇的云服务商仍不能从中计算出 $\alpha_{\rho(i)}$ 和 $\beta_{\rho(i)}$ ，故本文的访问控制方案没有给云服务商泄露任何用户的属性信息。

定理 4 本文提出的访问控制方案能够阻止好奇的云服务商基于他所拥有的未撤销用户的公开解密密钥或通过已与撤销用户合谋非法访问数据。这就意味着在类型 I 敌手的安全游戏中，没有多项式界的敌手对于挑战者有不可忽略的优势。

证明 假设在类型 I 敌手的安全游戏中，敌手 A 的优势为 ε 。本文将建立一个模拟器 S_1 ，它将 A 作为它的子程序，以优势 $\frac{\varepsilon}{2}$ 解决判定性 BDH 问题。

假设在 DBDH 问题中，挑战者抛一枚均匀的硬币 $\sigma \in \{0,1\}$ ，如果正面朝上 ($\sigma=0$)，则将 $(X = g^a, Y = g^b, Z = g^c, P = e(g, g)^{abc})$ 给 S_1 ；否则 ($\sigma=1$)，则将 $(X = g^a, Y = g^b, Z = g^c, P = e(g, g)^v)$ 给 S_1 。

S_1 模拟对类型 I 敌手 A 。

初始化阶段 S_1 将全局参数 $G = \{p, g_1 = X, g_2 = Y\}$ ，授权方公钥 $PK_{AID} = \{x, e(g_1, g_2)^{\alpha_x}, g_2^{\beta_x}\}_{x \in S_{AID}}$ 和数据拥有者公钥 $PK_{GID} = g_2^{x_{GID}}$ 给 A 。

第 1 阶段 S_1 初始化一个空的列表 S ， A 可以请求下列谕言。

1) 公开解密密钥提取谕言。当收到全局的身份 GID_i 和时间期限 T_j 。 S_1 首先通过以下 H_1 -谕言和 H_2 -谕言得到散列函数 $H_1(GID_i)$ 和 $H_2(T_j)$ 的散列值。

H_1 -谕言：当收到用户的全局身份 GID_i ， $1 \leq i \leq q_{H_1}$ 。 S_1 随机地选择 $u_i \in Z_p$ 并计算 $H_1(GID_i) = g_1^{u_i}$ ，将散列的结果给 A 。

H_2 -谕言：当收到时间期限 T_j ， $1 \leq j \leq q_{H_2}$ 。 S_1 随机地选择 $v_j \in Z_p$ 并计算 $H_2(T_j) = g_2^{v_j}$ 。

接下来， S_1 将全局身份 GID_i 加入列表 S 中 $S = S \cup \{GID_i\}$ ，并将用户 GID_i 的公开解密密钥 $PDK_{GID_i, AID} = \{k_{x,1} = g_1^{\beta_x}, k_{x,2} = g_1^{\alpha_x} g_1^{r_{GID,1}} g_1^{u_i \beta_x}, TK_{T_j, GID_i} = g_2^{r_{GID,2}} g_2^{v_j r_{GID_i}}\}$ 给 A 。

2) 公开解密密钥更新谕言。当收到全局身份 GID_i 和时间期限 T^* 。如果 $GID_i \in S$ ， S_1 将更新后

的 $TK_{T^*, GID_i} = g_2^{r_{GID,2}} g_2^{v^* r_{GID_i}}$ 给 A ，并将全局身份 GID_i 加入列表 S' 中 $S' = S' \cup \{GID_i\}$ 。

3) 秘密解密密钥提取谕言。当收到全局身份 GID_i 。如果 $GID_i \in S - S'$ ， S_1 将用户 GID_i 的秘密解密密钥 $SDK_{GID_i, AID} = \{g^{r_{GID_i}}\}$ 给 A 。

挑战 A 给出 2 个等长的消息 m_0, m_1 和访问控制结构 (M^*, ρ^*, T^*) 。 S_1 选择加密指数 $s = c$ 和向量 $v = (s, y_2, \dots, y_l) \in Z_p^l$ ，并计算 $\lambda_i = M_i^* v$ ，其中 $y_2, \dots, y_l$ 被用来分享加密指数 s ， M_i^* 表示矩阵 M^* 的第 i 行。又选择一个向量 $w = (0, y'_2, \dots, y'_l) \in Z_p^l$ 和加密信息 $r \in Z_p$ ，其中 $y'_2, \dots, y'_l$ 需满足 $(0M_{i1}^* + y'_2 M_{i2}^* + y'_3 M_{i3}^* + \dots + y'_l M_{il}^* + \beta_{\rho(i)} r) \cdot b = s$ ，其中 $(M_{i1}^*, M_{i2}^*, \dots, M_{il}^*)$ 是矩阵 M^* 的任意第 i 行的元素。然后，计算 $w_i = M_i w$ 。最后， S_1 随机抛一枚硬币 $\mu \in \{0,1\}$ ，输出模拟密文

$$CT^* = (C = m_\mu P, \forall i = 1, \dots, n,$$

$$C_{1,i} = e(g_1, g_2)^{\lambda_i} e(g_1, g_2)^{\alpha_{\rho(i)} r},$$

$$C_{2,i} = g_2^{w_i} g_2^{\beta_{\rho(i)} r}, C_{3,i} = g_2^{\beta_{\rho(i)} x_{GID}},$$

$$C_4 = g_1^r, C_5 = g_2^r, C_{T^*} = g_2^{v^* r})$$

第 2 阶段 重复第 1 阶段。

猜测 A 将会输出对 μ 的猜测 μ' 。如果 $\mu' = \mu$ ， S_1 猜测 $P = e(g, g)^{abc}$ ，输出 $\sigma = 0$ ；否则， S_1 猜测 $P = e(g, g)^v$ ，输出 $\sigma = 1$ 。

如果 $P = e(g, g)^{abc}$ ，则 $C = m_\mu e(g, g)^{abc} = m_\mu e(g_1, g_2)^s$ 。根据假设，在类型 I 敌手的安全游戏中，敌手 A 的优势为 ε 。因此

$$\Pr[\mu' = \mu | e(g, g)^{abc}] = \frac{1}{2} + \varepsilon$$

反之，如果 $P = e(g, g)^v$ ，则 $C = m_\mu e(g, g)^v$ 。对于敌手 A 而言， m_μ 完全被隐藏，没有任何优势区分 μ 。因此

$$\Pr[\mu' = \mu | e(g, g)^v] = \frac{1}{2}$$

对于 S_1 在 DBDH 问题中，输出 $\sigma = 0$ 的概率为

$$\begin{aligned} & \Pr[\sigma = 0 | (g, g^a, g^b, g^c, P)] \\ &= \frac{1}{2} \Pr[\mu' = \mu | e(g, g)^{abc}] + \frac{1}{2} \Pr[\mu' = \mu | e(g, g)^v] \\ &= \frac{1}{2} \times \left(\frac{1}{2} + \varepsilon\right) + \frac{1}{2} \times \frac{1}{2} = \frac{1}{2} + \frac{\varepsilon}{2} \end{aligned}$$

根据 DBDH 假设, 对于任何多项式时间的敌手来说, 判定 P 是否为 $e(g, g)^{abc}$ 的优势是可忽略的。所以, S_1 在解决 DBDH 问题中的优势 $\frac{\varepsilon}{2}$ 是可忽略的。因此, 在类型 I 敌手的安全游戏中, 敌手 A 的优势 ε 也是可忽略的。

故在类型 I 敌手的安全游戏中, 敌手 A 猜中的概率为 $\Pr[\mu' = \mu] = \frac{1}{2}$ 。

定理 5 本文提出的访问控制方案能够抵抗敌手对部分授权方的攻击。这就意味着在类型 II 敌手的安全游戏中没有多项式界的敌手对于挑战者有不可忽略的优势。

证明 假设在类型 II 敌手的安全游戏中, 敌手 A 的优势为 ε 。建立一个模拟器 S_2 , 将 A 作为它的子程序, 以优势 $\frac{\varepsilon}{2}$ 解决判定性 BDH 问题。

假设在 DBDH 问题中, 挑战者抛一枚均匀的硬币 $\sigma \in \{0, 1\}$, 如果正面朝上 ($\sigma = 0$), 则将 $X = g^a, Y = g^b, Z = g^c, P = e(g, g)^{abc}$ 给 S_2 ; 否则 $\sigma = 1$, 则将 $X = g^a, Y = g^b, Z = g^c, P = e(g, g)^v$ 给 S_2 。

S_2 模拟对类型 II 敌手 A 。

初始化阶段 S_2 将全局参数 $G = \{p, g_1 = X, g_2 = Y\}$ 和数据拥有者公钥 $PK_{GID} = g_2^{x_{GID}}$ 给 A 。对于被攻击的授权方 $S'_A \subset S_A$, S_2 将这些授权方的私钥 $SK_{AID} = \{\alpha_x, \beta_x\}_{x \in AID}$ 和公钥 $PK_{AID} = \{x, e(g_1, g_2)^{\alpha_x}, g_2^{\beta_x}\}_{x \in S_{AID}}$ 都给 A 。对于未被攻击的授权方 $S_A - S'_A$, S_2 只需将这些授权方的公钥 $PK_{AID} = \{x, e(g_1, g_2)^{\alpha_x}, g_2^{\beta_x}\}_{x \in S_{AID}}$ 给 A 。

第 1 阶段 A 可以请求下列谕言。

公开解密密钥提取谕言: 当收到全局身份 GID_i , 时间期限 T^* 还有属性集 $S_{GID_i, AID}$, 其中, 属性集 $S_{GID_i, AID}$ 中的属性属于未被攻击的授权方 $S_A - S'_A$ 的属性域。 S_2 首先通过以下 H_1 -谕言和 H_2 -谕言得到散列函数 $H_1(GID_i)$ 和 $H_2(T^*)$ 的散列值。

H_1 -谕言: 当收到用户的全局身份 GID_i , $1 \leq i \leq q_{H_1}$ 。 S_2 随机的选择 $u_i \in Z_p$ 并计算 $H_1(GID_i) = g_1^{u_i}$, 将散列的结果给 A 。

H_2 -谕言: 当收到时间期限 T^* 。 S_2 随机的选择 $v^* \in Z_p$ 并计算 $H_2(T^*) = g_2^{v^*}$ 。

接下来, S_2 将用户 GID_i 对应于授权方 $S_A - S'_A$ 的公开解密密钥 $PDK_{GID_i, AID} = \{k_{x,1} = g_1^{\beta_x}, k_{x,2} = g_1^{\alpha_x} g_2^{r_{GID_i,1}} g_1^{u_i \beta_x}, TK_{T^*, GID_i} = g_2^{r_{GID_i,2}} g_2^{v^* r_{GID_i}}\}$ 给 A 。

挑战 A 给出 2 个等长的消息 m_0, m_1 和访问控制结构 (M^*, ρ^*, T^*) 。与 S_1 相同, S_2 最终也输出模拟密文

$$CT^* = (C = m_\mu P, \forall i = 1, \dots, n,$$

$$C_{1,i} = e(g_1, g_2)^{\lambda_i} e(g_1, g_2)^{\alpha_{\rho(i)} r},$$

$$C_{2,i} = g_2^{w_i} g_2^{\beta_{\rho(i)} r}, C_{3,i} = g_2^{\beta_{\rho(i)} x_{GID}},$$

$$C_4 = g_1^r, C_5 = g_2^r, C_{T^*} = g_2^{v^* r}).$$

第 2 阶段 重复第 1 阶段。

猜测 A 将会输出对 μ 的猜测 μ' 。如果 $\mu' = \mu$, S_2 猜测 $P = e(g, g)^{abc}$, 输出 $\sigma = 0$; 否则, S_2 猜测 $P = e(g, g)^v$, 输出 $\sigma = 1$ 。

如果, $P = e(g, g)^{abc}$, 则 $C = m_\mu e(g, g)^{abc} = m_\mu e(g_1, g_2)^s$ 。根据假设, 在类型 I 敌手的安全游戏中, 敌手 A 的优势为 ε 。因此

$$\Pr[\mu' = \mu | e(g, g)^{abc}] = \frac{1}{2} + \varepsilon$$

反之, 如果 $P = e(g, g)^v$, 则 $C = m_\mu e(g, g)^v$ 。对于敌手 A 而言, m_μ 完全被隐藏, 没有任何优势区分 μ 。因此

$$\Pr[\mu' = \mu | e(g, g)^v] = \frac{1}{2}$$

S_2 在 DBDH 问题中, 输出 $\sigma = 0$ 的概率为

$$\begin{aligned} \Pr[\sigma = 0 | (g, g^a, g^b, g^c, P)] \\ &= \frac{1}{2} \Pr[\mu' = \mu | e(g, g)^{abc}] + \frac{1}{2} \Pr[\mu' = \mu | e(g, g)^v] \\ &= \frac{1}{2} \times \left(\frac{1}{2} + \varepsilon\right) + \frac{1}{2} \times \frac{1}{2} = \frac{1}{2} + \frac{\varepsilon}{2} \end{aligned}$$

根据 DBDH 假设, 对于任何多项式时间的敌手来说, 判定 P 是否为 $e(g, g)^{abc}$ 的优势是可忽略的。所以, S_2 在解决 DBDH 问题中的优势 $\frac{\varepsilon}{2}$ 是可忽略的。因此, 在类型 II 敌手的安全游戏中, 敌手 A 的优势 ε 也是可忽略的。

故在类型 II 敌手的安全游戏中, 敌手 A 猜中的概率为 $\Pr[\mu' = \mu] = \frac{1}{2}$ 。

5.5 性能分析

下文中以 n, l 表示访问矩阵的行数和列数, C_1 表示指数运算的计算量, C_2 表示乘法运算的计算量, C_3 表示双线性对运算的计算量。加解密时数据拥有者, 用户和云服务商的计算量如表 1 所示。

当用户撤销发生时, 首先多授权方应该为 $|U|$ 个未撤销用户更新他们公开解密密钥中的时间部分, 因此, 在授权方处的计算量大约是 $|U|(2C_1 + C_2)$ 。接下来云服务商需要去更新 $|CT|$ 个密文中的时间期限, 因此, 在云服务商处的计算量大约是 $|CT|C_1$ 。故当用户撤销发生时, 此方案的全部计算量为 $|U|(2C_1 + C_2) + |CT|C_1$ 。通过以上分析可知, 当用户撤销发生时, 本文提出方案的全部计算量和系统中用户的多少和密文的多少有关, 与属性域中的属性的多少无关。另外, 整个用户撤销过程中不会给数据拥有者和未撤销用户带来额外的计算和通信开销。

接下来, 将对本文提出的方案与现有的 CP-ABE 方案^[2,3,5]在加密、解密还有用户撤销发生时的计算量进行比较。为了简洁, 只考虑各个

方案中指数运算和双线性对运算的计算量, 忽略乘法运算的计算量。

表 2 列出了加密、解密时的计算量, 其中, n 表示密文访问策略中属性的数量。表 3 列出了当用户撤销发生时的计算量, 其中, $|R|$ 表示撤销用户所拥有的属性数量, $|R_i|$ 表示第 i 个密文访问策略中包含撤销用户所拥有属性的数量, $|U|$ 表示未撤销用户的数量, $|CT|$ 表示需要更新的密文的数量。

通过表 2 的比较可以看出, 与现有的 CP-ABE 方案相比, 虽然本文提出的方案在密文中加入了时间期限, 但并没有因此增加数据拥有者在加密过程中的计算量, 因为这些计算量由云服务商承担。在解密过程中, 由于云服务商承担了更多的解密计算量, 因此, 与现有的 CP-ABE 方案相比, 本文提出的方案更具有优势。

通过表 3 的比较可以看出, 当用户撤销发生时, 现有的 CP-ABE 方案不仅会给数据拥有者带来额外的计算量, 而且授权方和数据拥有者的计算量与撤销用户所拥有的属性数量 $|R|$ 相关。当 $|R|$ 较大时, 整个系统的计算量将大大增加。然而,

表 1 加解密时的运算量

加密		解密	
数据拥有者	云服务商	用户	云服务商
$(5n+1)C_1 + [(2l+5)n+1]C_2 + C_3$	$3C_1$	$nC_1 + 4nC_2 + nC_3 + C_4$	$4nC_2 + 6nC_3$

表 2 本文方案与现有 CP-ABE 方案加密、解密时的性能比较

算法	加密		解密	
	数据拥有者	云服务商	云服务商	用户
BSW ^[2] 方案	$(2n+2)C_1 + C_3$	0	0	$2nC_1 + (2n+1)C_3$
W ^[3] 方案	$(3n+2)C_1 + C_3$	0	0	$nC_1 + (2n+1)C_3$
LW ^[5] 方案	$(5n+1)C_1 + C_3$	0	0	$nC_1 + 2nC_3$
本文的方案	$(5n+1)C_1 + C_3$	$3C_1$	$6nC_3$	$nC_1 + nC_3$

表 3 本文方案与现有 CP-ABE 方案用户撤销时的性能比较

算法	授权方	云服务商	数据拥有者
BSW ^[2] 方案	$(2 R +1) U C_1$	0	$\sum_{i=1}^{ CT } 2 R_i C_1$
W ^[3] 方案	$ R U C_1$	0	$\sum_{i=1}^{ CT } 3 R_i C_1$
LW ^[5] 方案	$2 R U C_1$	0	$\sum_{i=1}^{ CT } [5 R_i C_1 + C_3]$
本文的方案	$2 U C_1$	$ CT C_1$	0

本文提出的方案不仅不会给数据拥有者带来额外的计算量,而且授权方和云服务商的计算量与撤销用户所拥有的属性数量无关,大大减小了整个系统的计算量。

通过以上的分析和比较,可以得出结论:与现有的基于属性的访问控制方案相比,本文提出的方案更具有优势,更加有效。

6 结束语

本文主要解决了在多授权方基于属性加密中进行有效用户撤销的问题。本文将时间期限加入密文和解密密钥中,并利用“混合解密密钥”实现了有效的用户撤销。本文还分析了该方案的正确性、安全性和有效性,最终得出该方案是安全、有效的。除此之外,该方案还能实现密文中访问策略的更新,用户动态特权管理和已共享数据的自我销毁。接下来,将进行进一步优化,减少加密、解密过程中的计算量,使整个方案更加满足现实的要求。

参考文献:

- [1] GOYAL V, PANDEY O, SAHAI A, et al. Attribute-based encryption for fine-grained access control of encrypted data[C]/CCS'06, ACM. c2006: 89-98.
- [2] BETHENCOURT J, SAHAI A, WATERS B. Ciphertext-policy attribute-based encryption[C]/S&P'07, IEEE. c2007: 321-334.
- [3] WATERS B. Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization[C]/PKC'11, Berlin: Springer. c2011: 53-70.
- [4] LEWKO A B, OKAMOTO T, SAHAI A, et al. Fully secure functional encryption: attribute-based encryption and (hierarchical) inner product encryption[C]/EUROCRYPT'10, Berlin: Springer. c2010: 62-91.
- [5] LEWKO A B, WATERS B. Decentralizing attribute-based encryption[C]/EUROCRYPT'11, Berlin: Springer. c2011: 568-588.
- [6] LI Q, MA J, LI R, et al. Large universe decentralized key-policy attribute-based encryption[J]. Security and communication Networks, 2014, 8(3): 501-509.
- [7] CHEUNG L, NEWPORT C C. Provably secure ciphertext policy ABE[C]/The 14th ACM Conference on Computer and Communications Security ACM. c2007: 456-465.
- [8] OSTROVSKY R, SAHAI A, WATERS B. Attribute-based encryption with non-monotonic access structures[C]/The 14th ACM on Computer and Communications Security. c2007: 195-203.
- [9] YU S, WANG C, REN K, et al. Achieving secure, scalable, and fine-grained data access control in cloud computing[C]/The 29th IEEE Internet Conference Computer Communication. c2010: 1-9.
- [10] TYSOWSKI P, HASAN M. Hybrid attribute and re-encryption-based key management for secure and scalable mobile applications in clouds[C]/IEEE Transactions Cloud Computing. c2013: 172-186.
- [11] CHASE M. Multi-authority attribute based encryption[C]/Theory of Cryptography, Berlin: Springer-Verlag. 2007: 515-534.
- [12] CHASE M, CHOW S S M. Improving privacy and security in multi-authority attribute-based encryption[C]/The 16th CCS. c2009: 121-130.
- [13] LIN H, CAO Z, LIANG X, et al. Secure threshold multi authority attribute based encryption without a central authority[J]. Information Sciences an International Journal, 2010, 180(13): 2618-2632.
- [14] BOŽOVIĆ V, SOCEK D, STEINWANDT R, et al. Multi-authority attribute-based encryption with honest-but-curious central authority[J]. International Journal of Computer Mathematics, 2011, 90(3): 268-283.
- [15] LI F, RAHULAMATHAVAN Y, RAJARAMAN M, et al. Low complexity multi-authority attribute based encryption scheme for mobile cloud computing[C]/The 7th SOSE, IEEE. c2013: 573-577.
- [16] YANG K, JIA X, REN K, et al. DAC-MACS: Effective data access control for multi-authority cloud storage systems[C]/The IEEE INFOCOM. c2013: 2895-2903.
- [17] YU S, WANG C, REN K, et al. Attribute based data sharing with attribute revocation[C]/The ACM Symposium Information, Computer and Communication Security. c2010: 261-270.
- [18] RUJ S, STOJMENOVIC M, NAYAK A. Decentralized access control with anonymous authentication of data stored in clouds[C]/IEEE Trans. Parallel and Distributed Systems. c2014: 384-394.
- [19] YANG K, JIA X H. Expressive, efficient, and revocable data access control for multi-authority cloud storage[C]/IEEE Trans Parallel and Distributed Systems. c2014: 1735-1744.
- [20] HE H, LI R X, DONG X H, et al. Secure, efficient and fine-grained data access control mechanism for p2p storage cloud[C]/IEEE Transactions on Cloud Computing. c2014: 471-484.
- [21] BINBUSAYYIS A, ZHANG N. Decentralized attribute-based encryption scheme with scalable revocation for sharing data in public cloud servers[C]/Cloud Technologies and Applications. c2015: 1-8.
- [22] YANG K, JIA X H, REN K. Secure and verifiable policy update outsourcing for big data access control in the cloud[C]/IEEE Trans. Parallel and Distributed Systems. c2014: 3461-3470.

作者简介:



张亮轩(1991-),男,陕西西安人,西安电子科技大学硕士生,主要研究方向为密码学、云计算安全。



李晖(1968-),男,河南灵宝人,博士,西安电子科技大学教授、博士生导师,主要研究方向为密码学、无线网络安全、云计算安全、信息论与编码理论。