文章编号：2095-2163(2023)03-0077-06 中图分类号：TP309.7 文献标志码：A

云雾环境下可追责可撤销的属性基加密方案研究

苗志伟，巫朝霞
（新疆财经大学 统计与数据科学学院，乌鲁木齐 830011）

摘 要：随着物联网设备产生的数据激增，一种称为雾计算的新范式已经发展起来，其可以在边缘处理和分析数据。云计算和雾计算一起被用于巨大的存储和处理资源。而现有的 CP-ABE 方案不太适合云雾物联网环境，因为不能同时提供以下功能：抗密钥托管、属性撤销、恶意用户追责和复杂操作的外包。因此，本文提出了一种云雾环境下可追责可撤销的数据安全共享方案，该方案支持密钥抗托管、属性撤销和恶意用户追责功能。通过将复杂的加解密、属性撤销和恶意用户追责的任务外包给第三方，为物联网设备留下恒定且少量的计算量。同时，所提方案只更新那些与撤销属性相关联的关键组件和密文，实现多层次撤销，提高其撤销效率。该方案与现有的 CP-ABE 方案不同，用户持有一个恒定大小的密钥，该密钥在整个过程中保持不变。该方案性能分析表明，所提方案是安全高效的，适用于资源受限的物联网设备。

关键词：密文策略属性基加密；云计算；外包解密；可撤销；用户可追踪

Research on traceable and revocable attribute-based encryption scheme in cloud-fog environment

MIAO Zhiwei, WU Zhaoxia
(School of Statistics and Data Science, Xinjiang University of Finance and Economics, Urumqi 830011, China)

【Abstract】 With the explosion of data generated by IoT devices, a new paradigm called fog computing has been developed, which can process and analyze data at the edge. Cloud computing and fog computing are used together for huge storage and processing resources. However, the existing CP - ABE solutions are not suitable for the cloud IOT environment, because they cannot simultaneously provide the following functions: anti key escrow, attribute revocation, malicious user accountability, and outsourcing of complex operations. Therefore, this paper proposes a data security sharing scheme with accountability and revocation in cloud environment, which supports key anti trust, attribute revocation and malicious user accountability. By outsourcing complex encryption and decryption, attribute revocation, and malicious user accountability tasks to a third party, IoT devices are left with a constant and small amount of computing. At the same time, the proposed scheme only updates the key components and ciphertext associated with the revocation attribute to achieve multi - level revocation and improve its revocation efficiency. This scheme is different from the existing CP-ABE scheme. The user holds a constant size key, which remains unchanged throughout the process. The performance analysis of this scheme shows that the proposed scheme is safe, efficient, and suitable for resource constrained IoT devices.

【Key words】 ciphertext-policy attribute-based encryption; cloud computing; outsourcing decryption; revocable; user traceable

0 引 言

随着物联网设备产生的数据的激增，云服务器的系统响应时间也随之出现延迟，因此一种称为雾计算的新算法就应运而生，允许其在边缘处理和分析数据，雾节点与云服务器一样，都是以存储和计算巨大资源为目的。然而，不同层次的数据存储和计算增加了数据隐私的风险。因此，物联网环境中的雾云框架需要一种访问控制机制。

云雾计算不仅可以像传统的云计算一样提供存储和加解密服务，而且还可以通过雾节点将这些服务转移到离用户更近的地方，从而降低在资源受限设备上的加解密开销和系统的响应时间。2016 年，Peng 等学者[1]针对资源受限的物联网设备环境下，提出了第一个支持雾计算的 CP-ABE 方案，雾节点充当数据所有者、数据使用者和云服务提供商之间

基金项目：国家自然科学基金(61941205)。
作者简介：苗志伟(1999-)，男，硕士研究生，主要研究方向：数据安全加密；巫朝霞(1975-)，女，博士，教授，硕士生导师，主要研究方向：信息安全研究。
通讯作者：巫朝霞 Email：wuzhaoxia828@163.com
收稿日期：2022-12-09

的桥梁,以便于在本地设备上执行。随后,Fan 等学者[2]提出了一种高效且保护隐私的外包多权限访问控制方案,用户的所有属性都被转化为匿名的和可验证的,以实现隐私保护。Derki 等学者[3]提出了一种可验证机制和基于属性的密钥管理方案来维护细粒度的访问控制。该方案中的物联网设备的执行时间更短,并且可以容忍。Tu 等学者[4]提出了一种用于资源受限设备的安全数据共享方案。该方案设计了安全通信协议,利用混沌密码学中的逻辑映射来生成一次性加密密钥,并使用该密钥加密新属性。随着密文策略属性基加密的应用场景不断变化,其需求也是不断增长的。

基于此,本文提出了一种云雾环境下支持可追责和可撤销的属性基加密方案。该方案将物联网数据加解密时涉及的计算代价高昂的操作外包给雾节点进行运算,同时采用了部分策略隐藏以减少雾节点的计算量。此外,用户持有一个恒定大小的用户私钥,在任何更新期间都保持不变;引入了属性管理器,可多层次撤销,提高撤销效率;该方案通过给定格式良好的解密密钥绑定用户身份来跟踪恶意用户,无需初始化用户信息列表用户信息可以直接加密;该方案确保了单个机构不能代表任何用户解密密文,抵抗密钥托管问题。性能分析表明,该方案对于资源受限的物联网设备用户是高效的。

1 背景知识

1.1 相关定义

定义 1 素数阶群下的双线性映射[5]:p 表示大素数,G_1,G_2 与 G_T 表示 3 个 p 阶循环群,g 与 $\tilde{g}$ 则分别表示群 G_1 与 G_2 的生成元。$e:G\times G\rightarrow G_T$ 表示一个满足下列条件的双线性映射:

(1)双线性:对于任意的 $u\in G_1$,$\tilde{u}\in G_2$ 和 x,$y\in Z_p$,等式 $e(u^x,\tilde{u}^y)=e(u,\tilde{u})^{xy}$ 成立。

(2)非退化性:存在 $g\in G_1$ 与 $\tilde{g}\in G_2$,使得 $e(g,\tilde{g})$ 在 G_T 中的阶为 p。

(3)可计算性:对于任意 $u\in G_1$ 和 $\tilde{u}\in G_2$,存在计算 $e(u,\tilde{u})$ 的多项式时间算法。

需要注意的是,若 $G_1\neq G_2$,那么称其为非对称双线性映射,否则为对称双线性映射。

1.2 访问树

二叉树如图 1 所示。令 N 表示完全二叉树的叶子节点总数,则该二叉树的节点总数为 $2N-1$。假设系统用户集合 $U=\{u_1,u_2,\cdots,u_N\}$,用户的最大数量为 $N_{\max}=2^d$,其中 d 为完全二叉树的深度。系统的属性集合 $L=\{x_1,x_2,\cdots,x_n\}$,设 $G_i\subset U$,G_i 被看作是能够访问属性 x_i 的用户集合。

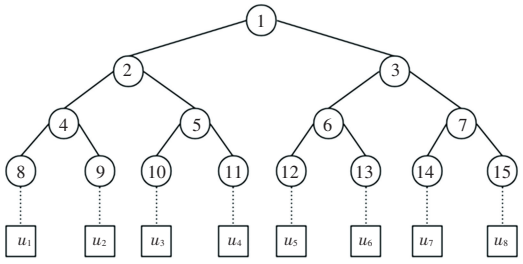

图 1 二叉树

Fig. 1 Binary tree

属性管理器按照如下过程根据完全二叉树为用户生成属性组密钥的相关参数:

(1)用户集合 U 中每一个用户被指定在二叉树的叶子节点中,每个节点 j 都存储一个随机值 θ_j。

(2)路径节点生成算法 $Path(u_i)$:对于每一个用户 u_i,从叶子节点到根节点上的所有节点被定义为用户 u_i 的路径节点。

(3)最小覆盖集算法 $Mincs(G_i)$:对于拥有相同属性的属性组 G_i,KEK 树中能覆盖 G_i 中所有用户的最小节点集合为最小覆盖集[6]。

(4)求 $Path(u_i)$ 与 $Mincs(G_i)$ 的交集:若用户 u_i 属性属于 G_i,则交集有且只有一个节点 j 存储随机值 θ_j。

当某一个用户 u_i 进行私钥申请时,需要完成以上 4 个步骤,且只针对未在撤销列表的用户 u_i 拥有的属性进行多次求交集,而不需要计算其它用户的交集情况。

2 方案设计

2.1 方案模型

本文方案系统一共包含 6 个实体。本文方案的系统模型如图 2 所示。由图 2 可见,对此方案模型拟做阐释分述如下。

(1)中央机构(Central Authority,CA):是一个完全可信的全局认证中心,负责承担本系统的初始化。

(2)属性机构(Attribute Authority,AA):主要负责授予不同用户的访问权限,生成数据用户的相关密钥。

(3)属性管理器(Attribute Manager,AM):主要负责生成和维护完全二叉树,生成用户的属性组密钥,也负责重新加密雾节点发送过来的密文生成最终密文。

(4)云服务器(Cloud Server,CS):主要负责密文的存储,还可以将密文分享给雾节点。

(5)雾节点(Fog Node,FN):雾节点是放置在CSP 和物联网设备中的实体。是不完全可信的实体。

(6)数据拥有者(Data Owner,DO):DO 负责数据的部分加密,本系统中 DO 是不可信的。

(7)数据用户(Data User):当用户申请访问数据时,DU 向雾节点发送访问请求,将会得到雾节点返回的半解密文,用户利用个人私钥最终可以解密得到明文。

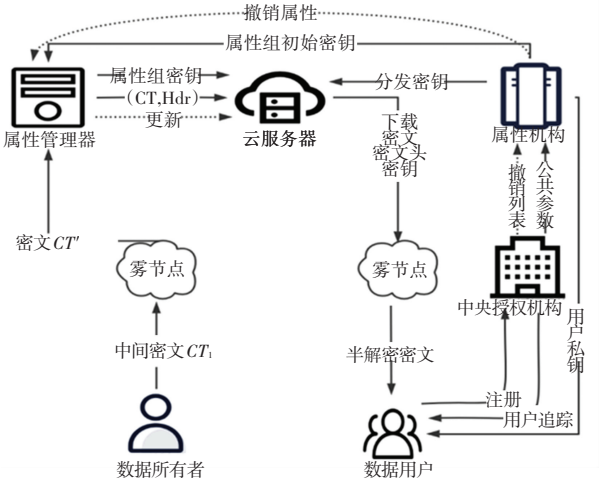

图 2　本文方案的系统模型

Fig. 2　System model of the proposed scheme

2.2　方案构造

(1)全局初始化 $GlobalSetup(\lambda) \rightarrow \{GP, uid, R\}$:算法输入一个安全参数 λ,生成全局公共参数 GP,其中 G 和 G_T 是2个阶为素数 p 的循环群,g 是 G 的生成元,$e: G \times G \rightarrow G_T$。CA 随机选取 $\alpha, a \in Z_p$,$h \in G$ 并计算 $Y = e(g,g)^{\alpha}$:$GP = \{G, G_T, g, Y, g^a, h, H\}$。选择一种对称加密方案(Enc,Dec)[7],随机选择对称加密密钥 $k \in Z_p$,加密任意长度比特串 $\{0,1\}^*$ 得到属于 Z_p 的密文。并且,中央机构为注册成功的用户生成全局唯一的身份标识 uid,初始化一个空的撤销列表 R。

(2)属性机构初始化 $AuthoritySetup(GP) \rightarrow \{APK, ASK, V, \{PK\}_{v_x \in V}\}$:属性机构 AA 将管理的属性集合划分为属性名集合以及属性值集合。这里,AA 定义属性名集合为 c_V,每个属性名对应的属性值集合定义为 $V_i = \{v_x\}$,$i \in c_V, x \in w_i$,其中 w_i 表示属性名 i 对应的属性值集合,属性值集合为 $V = U_{i \in c_V} V_i$。AA 选取随机数 $\beta \in Z_p$,对于任意属性值 $v_x \in V$,随机选取群元素 $h_{v_x}, h_{v_x} \in G$。最后生成属性密钥和 AA 的公私钥对:$\{PK\}_{v_x \in V} = h_{v_x}$,$APK = \{e(g,g)^{\beta}, h_{v_x}\}$,$ASK = \{a, \beta, \alpha\}$。

(3)属性管理器初始化 $AMSetup(T, U, L, GP) \rightarrow \{MPK, MSK\}$:该算法以二叉树 T,用户集合 U,系统属性集合 L 和公共参数 GP 为输入。其中,T 为完全二叉树,$u \in U$,完全二叉树的深度为 d。由完全二叉树的结构可知,用户数目最多为 $|U| = 2^d$,树种的节点数目总共为 $|L| = 2|U| - 1$。对于树中的每一个节点,随机选择 $\{t_i\}_{i=1}^{|L|} \in Z_p$,并计算 $T_i = g^{t_i}$。输出 AM 的公私钥:$MPK = \{T_i \mid 1 \leqslant i \leqslant |L|\}$,$MSK = \{t_i \mid 1 \leqslant i \leqslant |L|\}$

(4)私钥生成阶段。首先属性机构通过算法生成与属性集合相关的属性私钥,然后属性管理器通过算法生成属性组密钥。

① 属性密钥生成 $AAKeyGen(GP, ASK, \{PK\}_{v_x \in V}, MPK, S_{uid}) \rightarrow \{PxK, SK, KEK'\}$:假设用户拥有属性集 $S_{uid} = (c_{uid}, v^*)$,其中 $c_{uid} \subseteq c_V, v^* = \{v_i \mid v_i \in V_i, i \in c_{uid}\}$ 分别表示用户属性名和需要隐藏的属性值。AA 选取一个全局唯一的随机数 $z_{uid} \in Z_p^*$,计算 $kek_i = T_i^{z_{uid}}$ 和 $c = Enc_k(id)$,令 c 为追踪因子。其中,id 是与用户 u 在二叉树上相关联的节点值。对于 $i \in c_{uid}$ 计算:$PxK = (k' = c, k = g^{\frac{\alpha + z_{uid}\beta c}{z_{uid}(a+c)}}, L = g^{\beta} h^{\varepsilon}, L' = g^{\varepsilon}, D_i = \{g^{c\beta} h_{v_i}^{z_{uid}}\}_{i \in c_{uid}}, D' = g^{z_{uid}})$,$KEK' = \{att_i, kek_i\}_{i \in [1, |L|]}$,$SK = z_{uid}$。

其中,属性机构 AA 将代理密钥 PxK 发送给雾节点,并通过安全信道将个人私钥 SK 发送给用户。

② 属性组密钥生成 $FNKeyGen(GP, S_{uid}, KEK') \rightarrow KEK$:对于每一个属性 $x_i \in S_{uid}$,雾节点计算 $\eta_i \in Path(u) \cap Mincs(G_i)$,然后判断 η_i 是否为空。如果 $\eta_i = \emptyset$,雾节点停止计算;如果 $\eta_i \neq \emptyset$,雾节点计算 $KEK_i = kek_i^{1/\theta_j} = g^{z_{uid} t_i / \theta_j}$,其中节点随机值 θ_j,$Path(u) = \{i_1, \cdots, i_j, \cdots, i_{id}\}$,$i_1$ 为二叉树的根节点,i_{id} 是完全二叉树与数据用户 u 相关联的叶子节点值,$i_j = j$。最后输出属性组密钥 $KEK = \{x_i, j, kek_i, KEK_i\}_{i \in [0,d]}$

(5)数据加密阶段:首先,数据所有者通过对称密钥加密算法,使用内容密钥 ck 加密文件 M,生成 $E_{ck}(M)$。然后,定义一个应用于其上的访问策略 W,并将其发送到指定的雾节点中,雾节点在其中重新加密密文策略下的部分密文。

① $DO.Encrypt(GP, ck, APK, W) \rightarrow CT_1$:数据拥有者根据访问控制策略 $\boldsymbol{W} = (\boldsymbol{A}, \rho, \tau)$,其中 $\boldsymbol{A}$ 为 $l \times n$ 的矩阵,l 为访问策略中涉及的属性总数。定义一

个函数 ρ 将矩阵中每一行都映射到的属性名 c_{uid} 中，每个属性名只能出现一次。$\tau=(t_{\rho(1)},t_{\rho(2)},t_{\rho(3)},\cdots,t_{\rho(l)})\in Z_p^l$ 表示属性名对应的属性值。数据拥有者随机随机选择 $s_1\in Z_p$，计算 $CT_1=(W,E_{ck}(M),C=ck\cdot Y^{s_1},C_0=g^{as_1},C_0'=g^{s_1},C_0''=h^{s_1})$。

② $FN.Encrypt(GP,\{PK\}_{v_x\in V},M,W)\rightarrow CT'$：雾节点随机选择秘密值 $s\in Z_p$ 和一个随机向量 $\vec{\mu}=(s,v_2,v_3,\cdots,v_n)\in Z_p^n$，其中随机数 $v_2,v_3,\cdots,v_n$ 是用来共享秘密值 s。对于 $i\in[1,l]$，雾节点分别计算 $\lambda_i=A_i\cdot\vec{\mu}$，其中 A_i 表示矩阵 $\boldsymbol{A}$ 的第 i 行。然后，计算密文组件 $C_1=g^{s_1}g^s,C_1'=h^{s_1}h^s$。最后，输出密文 $CT'=(\bar{W},E_{ck}(M),C=ck\cdot Y^{s_1},C_0=g^{as_1},C_0'=g^{s_1},C_1=g^{s_1}g^s,C_1'=h^{s_1}h^s,\{C_i=g^{\lambda_i},C_i'=h_{t_{\rho(i)}}^{-s}\}_{i\in[1,l]})$。

③ $AM.Encrypt(GP,FSK,CT',R)\rightarrow\{CT,Hdr\}$：属性管理器对于 $\forall i\in[1,l]$，随机选择 $r_i\in Z_p$ 并调用 $Mincs(G_i)$ 算法计算该属性对应属性组的最小覆盖集，然后重新加密密文 CT' 获得密文 CT。另外，其计算密文头 Hdr，最后 AM 将 (CT,Hdr) 上传至云服务器进行存储，即：

$$CT=\{\bar{W},E_{ck}(M),C,C_0,C_0',C_1,C_1',C_i,C_i'=g^{r_i}h_{t_{\rho(i)}}^{-s},R\}$$

$$Hdr=\{j,E(r_i)=g^{r_i\theta_j/t_i}\}_{j\in Mincs(G_i),i\in[1,l]}$$

（6）雾节点解密 $FN.Decrypt(PxK,CT,Hdr,KEK)\rightarrow B$：当数据用户发出访问数据的请求时，雾节点判断用户是否满足部分隐藏后的访问控制策略。如果满足，雾节点利用代理密钥为其完成属性名认证操作，输出半解密密文，否则返回 $\perp$。该算法的输出存在以下 2 种情况：

① 情况 1。若用户的身份 $u\in R$，则输出 $\perp$。

② 情况 2。若用户的身份 $u\notin R$，对于满足部分隐藏访问策略 $\bar{W}$ 的用户，可以借助矩阵 $\boldsymbol{A}$ 的逆，求出一组常数量 $w_i\in Z_p$，使得 $\sum w_i\lambda_i=s$，其中 $i\in[1,l]$。输出的半解密密文如下：

$$F=\frac{\prod_{i\in c_{uid}}(e(D_i,C_i)e(g^{r_i}C_i',D'))^{w_i}}{e(KEK,E(r_i))}=\frac{\prod_{i\in c_{uid}}(e(g^{c\beta}h_{v_i}^{z_{uid}},g^{\lambda_i})e(g^{r_i}h_{t_{\rho(i)}}^{-s},g^{z_{uid}}))^{w_i}}{e(g^{z_{uid}t_i/\theta_j},g^{r_i\theta_j/t_i})}=e(g,g)^{c\beta s}$$

$$A=\frac{e(L^{k'},C_1)}{e(L'^{k'},C_1')}=\frac{e(g^{c\beta}h^{c\varepsilon},g^{s_1}g^s)}{e(g^{c\varepsilon},h^{s_1}h^s)}=e(g,g)^{c\beta(s_1+s)}$$

$$B=\frac{e(K,C_0C_0'^{k'})}{\frac{A}{F}}=\frac{e(g^{\frac{\alpha+z_{uid}\beta c}{z_{uid}(a+c)}},g^{(a+c)s_1})}{e(g,g)^{c\beta s_1}}=e(g,g)^{\frac{\alpha s_1}{z_{uid}}}$$

（7）用户解密 $(Decrypt(CT,B,SK)\rightarrow M)$：身份认证成功以后，数据用户 DU 可以利用用户私钥执行 $\frac{C}{B^{sk}}=\frac{ck\cdot e(g,g)^{\alpha s_1}}{e(g,g)^{\alpha s_1}}=ck$，然后使用相同的 ck 来解密 $E_{ck}(M)$ 获得消息 M。

（8）密钥完整性检查 $KeySanityCheck((GP,APK,PxK)\rightarrow True\ or\ False)$：中央权威机构运行该算法，评估解密密钥是否需要跟踪。如果怀疑解密密钥 PxK，则算法将检查解密密钥是否满足 $KeySanityCheck$，该算法由 3 部分组成：$k'\in Z_p,k,L,L',D_i,D'\in G,e(g,L)=e(g,g)^{\beta}\cdot e(h,L')\neq 1,\exists i\in c_{uid},\text{s.t.}\ e(h_{v_i},D')=e(g,D_i)\cdot e(g,g)^{\beta k'}\neq 1$。如果解密密钥 PxK 满足上述 3 个公式，则算法输出 $True$；否则，算法输出 $False$。

（9）追踪认责 $(Trace(GP,R,PxK)\rightarrow u\ or\ \perp)$：该算法由权威机构执行。如果解密密钥 PxK 不能通过 $KeySanityCheck$，输出 $\perp$。否则，算法执行如下操作：首先计算 $Dec_k(id)$ 来恢复与用户 u 相关联的叶子节点值 id。然后，在二叉树中搜索值为 id 的叶子节点，检索属性机构输出与 id 相关联的用户 u。如果不存在这样的节点，则输出 $\perp$。

（10）用户撤销：当接收到更新的撤销列表 R' 后，根据用户 uid 检索代理密钥列表，删除被撤销用户对应的代理密钥和个人私钥。雾节点就不能进行部分解密工作，用户也不能解密恢复内容密钥。

①属性撤销：当用户 u_i 的属性 x_x 被撤销后，可以更新属性组为 G_x' 对应的其他未撤销用户的相应 KEK，并且与撤销属性相关联的密文也将重新加密。

② $UpKEK(MSK,KEK,x_x)\rightarrow\overline{KEK}$：当用户 u 的属性 att_x 被撤销时，更新用户属性组为 G_x'，雾节点随机选择 ϵ_x 并计算 $T_x'=T_x^{\epsilon_x}$ 和 $t_x'=t_x\epsilon_x$，然后用 T_x' 和 t_x' 代替 MPK 和 MSK 中的 T_x 和 t_x，获得新的 $\overline{MPK}$ 和 $\overline{MSK}$。雾节点更新用户属性群 G_x' 并计算其最小覆盖集。对于每一个用户 $u\in G_x'$，雾节点计算 $\eta_x'=Path(u)\cap Mincs(G_x')$，计算 $\overline{kek_x}=(kek_x)^{\epsilon_x}$ 和

$\overline{KEK_x} = \overline{kek_x}^{1/\theta_j'}$，其中，随机值 $\theta_{j'}$ 所对应的节点是 $j' \in \eta_x'$。

③ $CTUpdate(CT, Hdr) \rightarrow \{\widetilde{CT}, \widetilde{Hdr}\}$：雾节点随机选择 s'，$r_i' \in Z_p^*$，重新加密密文 $\widetilde{C} = C \cdot e(g,g)^{\alpha s'}$，$\widetilde{C_0} = C_0 \cdot g^{s'}$，$\widetilde{C_0'} = C_0' \cdot g^{s'}$，$\widetilde{C_0''} = C_0'' \cdot g^{s'}$，$\widetilde{C_1} = C_i$，$\{\widetilde{C_x'} = \widetilde{C_x'} h_{t_{\rho(x)}}^{-s'} g^{r_x' - r_x}\}$，$\{\widetilde{C_1'} = \widetilde{C_1'} h_{t_{\rho(i)}}^{-s'}\}_{i \neq x}$。

④更新密文头：

$$\widetilde{Hdr} = \left(\frac{\{j', E(r_x') = g^{r_i' \theta_{j'}' / t_x'}\}_{j \in Mincs(G_x')}}{\{j, E(r_i) = g^{r_i \theta_j / t_i}\}_{j \in Mincs(G_x'), i \in [1,l], i \neq x}} \right)$$

最后输出更新后的密文和密文头，并存储到云服务器上。

3　方案分析

3.1　抗共谋攻击

在 ABE 方案中，2 个或多个数据用户可能会尝试对其属性键进行分组，以恢复各用户无法单独解锁的文件。因此，在本文方案中，AA 为每个用户生成密钥，该密钥由用户特定的随机值 z_{uid} 随机化。即使 2 个用户尝试组合其属性键，也无法获得 $e(g,g)^{\alpha s_1 / z_{uid}}$，因为这 2 个用户的 z_{uid} 都是唯一的。可能发生共谋攻击的另一个例子是被撤销的用户和不具有足够属性的现有用户共同攻击，在该方案中，AA 和 AM 分别生成的属性密钥 ASK 和 KEK，这使得 2 个不同用户的 2 个不同密钥的组合没有意义。因此，本文方案是抗共谋攻击的。

3.2　前后向安全性

在该方案中，一旦从用户中删除属性，AM 将重建主密钥的组件和被撤销属性公钥的相应组件。其重新定义了 KEK 和组件 r_i，通过随机选择的秘密 s' 重新加密用秘密 s 加密的相关密文，并且还用更新的 r_i 和相关联的密文头 Hdr。在这种情况下，被撤销的用户将无法恢复消息，因为密文被 s' 重新加密，并且该方案被构造为使得被撤销用户不可能确定 $e(g,g)^{\alpha(s+s')}$。因此，被撤销的用户决不可能进一步解密出由被撤销属性组成的访问策略加密的消息，从而实现了前向保密。当新用户加入系统时，用户的属性集中将会包含一个新属性。AM 根据更新后属性组信息去为该特定用户生成与新添加的属性相关的关键组件，从而保证了该方案的后向安全性。

3.3　抗密钥托管

用户生成密钥的机构恶意使用这些密钥并代表用户解密消息（称为密钥托管问题）。因此，在该方案中，用户的解密密钥是由 AA 和 AM 分别生成的，因此都不知道完整的密钥。实体 AA 生成解密密钥 APK 和 ASK，而 AM 生成 KEK。在该方案中，密钥 APK、ASK 和 KEK 是获得内容密钥 ck 所必需的，因此 AA 或 AM 不能单独解密消息，保证了针对实体 AA、AM、雾节点、CSP 和未授权用户的数据保密性。

4　性能分析

对本文所提出方案进行了全面的性能分析，并在功能、计算和存储开销方面与文献[6]、[8]、[9]、[10]的相关工作进行了讨论。不同方案系统功能对比结果见表 1。

在表 1 中，讨论了功能的对比，文献[8]和文献[9]所提出的方案支持访问策略隐藏。因此，由访问策略导致的隐私泄漏被减少到最小。文献[8]支持可追溯性和撤销，但与本文方案相比该方案不支持雾计算环境和抗密钥托管性，因为被撤销的用户和授权用户可以通过组合相关的密钥来解密密文，而单独地却无法解密。文献[9]和文献[10]均实现了抗密钥托管，但没有实现可追溯性，当密钥泄露给第三方时无法追踪到恶意用户。此外，本文方案和文献[6]支持外包解密期间涉及的所有昂贵操作，但文献[6]的访问策略任务仍会给资源受限的设备带来沉重的计算负担。由此可以观察到，仅有本文方案同时解决了云雾系统中多个方面的问题。

表 1　不同方案系统功能对比

Tab. 1　System function comparison of different schemes

方案	撤销	访问策略	外包解密	更新策略	可追踪	策略隐藏	抗密钥托管	雾计算
文献[8]方案	用户	LSSS	×	密文更新	√	√	×	×
文献[9]方案	属性	Tree	×	密文更新	×	×	√	×
文献[10]方案	×	LSSS	×	×	×	√	√	×
文献[6]方案	用户和属性	Tree	√	密文更新	×	×	√	√
本文方案	用户和属性	LSSS	√	密文更新	√	√	√	√

所提方案和文献[8-10]方案中 Encrypt、Decrypt 算法的性能和时间成本比较。图 3、图 4 比较了加、解密算法的时间成本。由图 3、图 4 可以看到,在本文方案中,无论属性的数量如何,所有者的加密时间和用户的解密时间都基本上更少且恒定,而在文献[8-10]中,随着属性的数量线性增加。

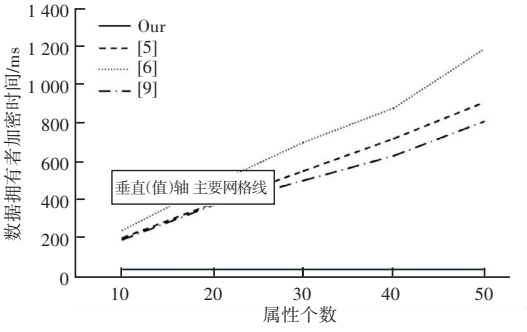

图 3 加密时间分析

Fig. 3 Encryption time analysis

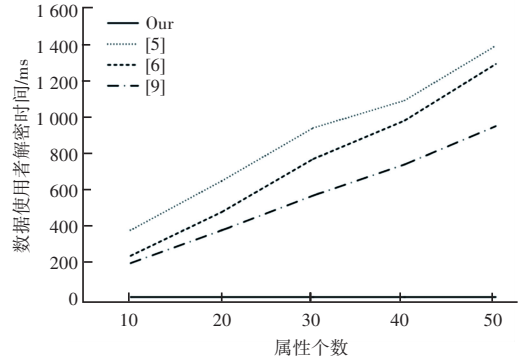

图 4 解密时间分析

Fig. 4 Decryption time analysis

从以上讨论可以推断,在数据所有者加密、数据用户解密、属性撤销和恶意用户追踪期间,所提方案要比其余方案更为高效。此外,本文方案对资源受限的数据用户分配了非常小且恒定的存储开销。

5 结束语

本文提出了一种云雾环境下可追责可撤销的数据安全共享方案,用于使用 CP-ABE 的物联网环境。将LSSS用作访问策略,用户属性表示为属性名和属性值,其中属性值用于加密,因此密文相关访问策略仅包含属性名称以满足部分隐藏策略。本文方案可以基于用户解密密钥中的追踪因子来跟踪用户,确保了该方案实现白盒的可追溯性,然后通过使用与用户信息相关联的二叉树的叶节点值来撤销用户。本文所提出的方案安全且高效,适用于资源受限的物联网设备。

参考文献

[1] PENG Zhang, CHEN Zehong, LIU J K, et al. An efficient access control scheme with outsourcing capability and attribute update for fog computing[J]. Future Generation Computer Systems, 2016, 78(PT.2):753-762.

[2] FAN Kai, XU Huiyue, GAO Longxiang, et al. Efficient and privacy preserving access control scheme for fog-enabled IoT[J]. Future Generation Computer Systems, 2019,99:134-142.

[3] DERKI M S,TABOUDJEMAT-NOUALI N,NOUALI O. A fine-grained access control scheme in fog - IoT based environment [C]// International Conference on Advanced Intelligent Systems for Sustainable Development. Cham:Springer, 2020: 465-474.

[4] TU Yuanfei, YANG Geng, WANG Jing, et al. A secure, efficient and verifiable multimedia data sharing scheme in fog networking system[J]. Cluster Computing, 2021, 24(1): 225-247.

[5] BONEH D, FRANKLIN M. Identity-based encryption from the Weil pairing[J]. Siam Journal on Computing, 2003, 32(3): 586-615.

[6] SARMA R, KUMAR C, BARBHUIYA F A. ACS-FIT: A secure and efficient access control scheme for fog-enabled IoT [C]// 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC). Toronto, on, Canada: IEEE, 2020: 2782-2789.

[7] 杜瑞忠, 闫沛文, 刘妍. 雾计算中细粒度属性更新的外包计算访问控制方案[J]. 通信学报, 2021,42(03):160-170.

[8] HAN Dezhi, PAN Nannan, LI K C. A traceable and revocable ciphertext-policy attribute-based encryption scheme based on privacy protection [J]. IEEE Transactions on Dependable and Secure Computing, 2022,19(1):316 - 327.

[9] LI Jiguo, YAO Wei, HAN Jinguang, et al. User collusion avoidance CP-ABE with efficient attribute revocation for cloud storage[J]. IEEE Systems Journal, 2018,12(2):1767-1777.

[10] ZHANG Yinghui, ZHENG Dong, DENG R H, Security and privacy in smart health: Efficient policy-hiding attribute-based access control [J]. IEEE Internet of Things Journal, 2018, 7: 2130-2145.

(上接第 76 页)

[24] ADERGHAL K, AFDEL K, BENOIS-PINEAU J, et al. Improving Alzheimer's stage categorization with Convolutional Neural Network using transfer learning and different magnetic resonance imaging modalities[J]. Heliyon, 2020, 6(12): 5652.

[25] ZHANG Yuanpeng, WANG Shuihua, XIA Kaijian, et al. Alzheimer's disease multiclass diagnosis via multimodal neuroimaging embedding feature selection and fusion [J]. Information Fusion, 2021, 66: 170-183.

[26] KANG Wenjie, LIN Lan, ZHANG Baiwen, et al. Multi-model and multi-slice ensemble learning architecture based on 2D convolutional neural networks for Alzheimer's disease diagnosis [J]. Computers in Biology & Medicine, 2021, 136: 104678.

[27] LIU Sheng, YADAV C, FERMANDEZ-GRANDA C, et al. On the design of convolutional neural networks for automatic detection of Alzheimer's disease [C]//Machine Learning for Health Workshop. NIPS Foundation, 2020: 184-201.