支持外包计算的属性基加密方案

常 晓 孙 瑾 付祎雪 苏文娟 宋娜娜
西安理工大学 陕西西安 710054

摘 要：属性基加密是云计算环境下解决数据隐私和细粒度访问控制的关键技术。考虑到传统的密文策略属性基加密 CP-ABE 方案中存在加解密运算成本高、属性更新困难、访问策略显式存储和恶意解密限制不足等挑战，我们的方案提出一种支持策略隐藏的可验证外包计算的属性基加密方案。该方案使用外包加解密技术，降低了用户端加解密成本。考虑到访问策略可能会包含数据所有者的隐私信息，我们提出了一种策略隐藏方案。使用星际文件系统（IPFS）存储密文，确保存储数据的安全性。使用可验证随机函数（VRF）实现有限的匿名访问控制，防止恶意用户发出大量外包解密请求，从而过度占用智能设备的计算资源。此外，该方案支持用户的属性更新。理论分析表明，该方案在标准模型下是选择明文攻击（CPA）安全的，实验结果表明，该方案能够减轻用户端的计算负担，并通过智能设备验证解密数据的正确性。

关键字：属性基加密；外包加解密；策略隐藏；访问控制；区块链

Attribute base encryption scheme that supports outsourcing calculation

Xiao Chang , Jin Sun, Yixue Fu, Wenjuan Su, Nana Song
Xi 'an University of Technology Xi' an, Shaanxi 710054

Abstract: Attribute-based encryption (ABE) is a critical technology for addressing data privacy and fine-grained access control in cloud computing environments. Traditional ciphertext-policy ABE (CP-ABE) schemes have faced challenges such as high computational costs for encryption and decryption, difficulty in updating attributes, explicit storage of access policies, and insufficient malicious decryption restrictions. In response to these challenges, our proposed solution is a verifiable outsourced computation ABE scheme that supports policy hiding. The scheme employs outsourced encryption and decryption techniques to reduce the computational costs for users. To protect the privacy of data owners, we propose a policy hiding scheme. The ciphertext is stored using the InterPlanetary File System (IPFS) to ensure data security. A verifiable random function (VRF) is used to implement limited anonymous access control, preventing malicious users from overwhelming smart device resources with excessive decryption requests. In addition, our scheme supports attribute updates for users. Theoretical analysis indicates that the proposed scheme is chosen plaintext attack (CPA) secure under the standard model, and experimental results demonstrate that the scheme reduces the computational burden on users and verifies the correctness of decrypted data using smart devices.

Keywords: attribute base encryption; outsourcing encryption and decryption; policy hiding; access control; blockchain

引言

随着信息技术和互联网技术的发展，信息数据的使用量呈现爆发式的增长，许多企业选择将大量的数据外包给云进行存储，以节省本地资源和降低数据的维护成本。然而，传统的云存储以集中存储的方式运行，导致数据的安全性很大程度上依赖于第三方云服务器的可信性。为了更好的保护隐私数据，我们将数据从集中式云存储系统转移到 IPFS 和区块链上，可以有效防止中心化存储由于网络或者物理原因造成的数据丢失风险。然而，直接将数据存储在 IPFS 中，会导致用户失去对数据的控制。密文策略属性基加密（CP-ABE）可以实现细粒度访问控制，数据拥有者可以灵活的设置访问策略，向指定的用户授予访问权限，保证存储数据的安全性。另外，许多现有的 CP-ABE 方案中，双线性对运算和解密时间随着访问策略的复杂性而增长，资源有限的用户将不能处理这种耗时的双线性对运算，或者需要花费很长的时间来解密。外包技术将大量的运算操作卸载至智能设备，极大的提高了用户的计算效率。但是，由于第三方智能设备不是完全可信，所以需要对外包解密结果的正确性进行验证。

本文构建了一种策略隐藏的可验证外包计算的属性基加密方案，主要工作内容如下：

（1） 安全的外包加解密：引入外包加解密技术，将部分复杂的运算外包给智能设备，降低了用户端加解密的计算量。同时，用户可以独立对第三方外包解密结果进行验证。

（2） 属性更新：支持数据用户的属性更新，保障了企业动态管理的安全性。

（3） 策略隐藏：使用策略隐藏算法，保护敏感的属性

信息，保障用户的隐私安全。

（4） 访问控制：我们提出一种限制用户恶意解密的细粒度访问控制机制，有效防止用户恶意浪费智能设备的计算资源，更加适用于实际的商业管理系统。其一，商业环境中使用的数据往往具有一定价值，数据的丢失和价值的流失都会对企业造成损失。其二，恶意用户为了占用资源可能会无限次的提出访问请求。

（5） 安全的数据存储：在 IPFS 存储大文件，并利用智能合约将验证参数存储在区块链上，显著提高了区块链的带宽。

一、基础知识

1.1 双线性映射

令 G_1， G_2 和 G_T 是三个乘法循环群，素数 p 是他们的阶。$e:G_1\times G_2\to G_T$ 是一个线性映射，满足以下性质：

1）双线性：$\forall g_1\in G_1,g_2\in G_2,\forall a,b\in_R z_p^*,e(g_1^a,g_2^b)=e(g_1,g_2)^{ab}$。

2）非退化性：$\exists g_1\in G_1,g_2\in G_2,e(g_1,g_2)\neq 1$。

3）可计算性：$\forall g_1\in G_1,g_2\in G_2$，可有效计算 $e(g_1,g_2)$。

1.2 可验证随机函数

可验证随机函数(Verifiable Random Function，简写 VRF)是一种将输入值映射为可验证的伪随机输出值的加密方案。VRF 算法由三个加密函数组成：*Keygen*、*Evaluate* 以及 *Verify*。

①$Keygen(r)\to(PK,SK)$：对任意随机输入，*Keygen* 产生非对称密钥对：公钥 PK 和私钥 SK。

②$Evaluate(SK,X)\to(result,proof)$：求值函数 *Evaluate* 输入私钥 SK、消息 X，输出伪随机字符串 $result$ 和证明 $proof$。

③$Verify(PK,X,result,proof)\to 0/1$：验证函数 *Verify* 输入验证密钥 PK、消息 X、伪随机字符串 $result$ 和证明 $proof$。输出结果 0/1：只有该函数验证了证明 $proof$ 是根据 X 生成的，且根据证明 $proof$ 可以推导出 $result$，才会输出 1，也就是说该函数验证 X 与 $proof$ 是否存在唯一的对应关系。

1.3 线性秘密共享方案

线性秘密共享方案可以用 $(\mathbb{M},\rho)$ 来表示，令 $P=\{P_1,P_2,\cdots,P_n\}$ 为参与者集合，$(\mathbb{M},\rho)$ 为访问控制策略，其中 M 是 $l\times n$ 的矩阵，ρ 是一个单射，它将矩阵 M 的行映射为相关的属性，线性秘密共享方案包括两个有效算法：

1) 共享：假设共享秘密为 $r\in z_p$，任意选择 $y_2,y_3,\cdots,y_n\in z_p$，构造向量 $v=(r,y_2,y_3,\cdots,y_n)$。计算 $\lambda_i=(W,v)_i$，其中 λ_i 是通过分量 $\rho(i)$ 得到的秘密共享值。

2) 重建：令 $S\in A$ 是一个属性集合，以及集合 $I=\{i:\rho(i)\in S\}\subseteq\{1,2,\cdots,l\}$。有一个常数 $\{w_i\in z_p\}_{i\in I}$ 满足 $\sum w_iM_i=(1,0,\cdots,0)$ 且使得 $\sum w_i\lambda_i=r$。对于任意的非授权集合 S'，常数 $\{w_i\}$ 不存在。

1.4 DBDH 假设

判定性双线性 Diffie–Hellman 问题（DBDH 问题）：G 和 G_1 是两个阶为素数 p 的乘法循环群，且满足双线性映射 $e:G\times G\to G_1$，g 是 G 的一个生成元，选取随机数 $a,b,c,z\in z_p$。给定五元组 (g,g^a,g^b,g^c,Z)，其中 $Z\in G_1$，如果没有一个算法能够以不可忽略的优势在多项式时间内区分 $Z=e(g,g)^{abc}$ 或 $Z=e(g,g)^z$，那么认为 DBDH 问题是难解的。

二、方案模型

2.1 系统模型

如图 1 所示，我们的方案由 6 个实体组成：企业系统（ES）、智能设备（SD）、数据拥有者（DO）、数据用户（DU）、星际文件系统（IPFS）和区块链系统（BS）。

区块链：不可更改的分布式账本。部署用户合约和授权合约，用户合约维护合法的用户列表；授权合约判断用户的访问权限。当用户发送访问请求时，授权合约判断用户是否为授权用户，若验证成功，则向用户发送相关的存储参数，否则拒绝访问。

IPFS: 内容寻址的数据存储协议，它会为每个文件生成唯一的哈希值。IPFS 作为分布式的存储系统，存储数 DO 上传的加密文件并返回哈希地址。

数据拥有者：定义基于属性的访问策略，并在 SD 的辅助下完成数据加密并上传到 IPFS 上。DO 在区块链上部署授权合约，将相关的验证参数存储在区块链上。企业内部的员工具有双重身份，既可以是数据拥有者也可以是数据用户。

企业系统：负责企业中所有人员的身份注册及职位标识，并为其生成相关密钥。

智能设备：位于网络边缘的可信实体，具有计算、存储和网络服务的能力。负责数据的外包加解密工作。

数据用户：需要访问企业数据的用户。DU 发送外包解密请求，在智能设备的辅助下完成解密。

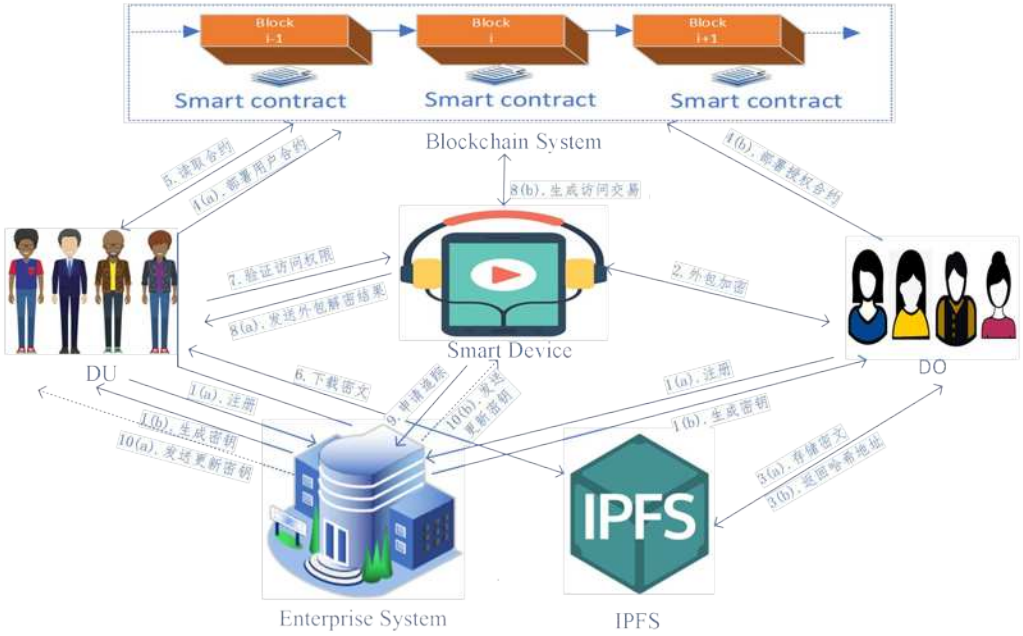

图 1 方案模型

2.2 安全模型

本节基于文献[6, 4]中的定义，我们提出该方案的选择性（CPA）安全模型。通过概率多项式敌手 A 和规约算法 B 之间的游戏来描述，安全游戏如下。

Init：A 设置一个挑战访问策略 $M*$ 并将其发送给 B。

Setup：B 运行 Setup 算法，生成公共参数 mpk 和公私钥对 Y_{SD}, sk_{SD}，并将其发送给 A。

Phase1：此阶段允许 A 以自适应的方式发出以下类型的质询：

–Create(S):A 可以质询一系列与属性集合 S 有关的密钥，即 A 将属性集合 S 发送给 B，然后 B 将与属性集合 S 有关的私钥 sk_s 和公私钥对 Y_{DU}, sk_{DU} 返回给 A。B 设置 $tk_s = (sk_s, tk_1)$，$j = j+1$ 并将 $(j, S, Y_{DU}, sk_{DU}, sk_s, tk_s)$ 记录在表 T 中。该阶段需要满足限制：任何查询的属性集合 S 都不能满足挑战的访问策略 $M*$。

–Corrupt.SK(i):当从 A 处接收到关于内容 i 的质询，B 检查第 i 个元组 $(i, S, Y_{DU}, sk_{DU}, sk_s, tk_s)$ 是否在列表 T 中存在。若不是，B 会给 A 返回终止。否则，B 设置 $D[i] \leftarrow D[i] \bigcup \{S\}$ 并给 A 返回 (Y_{SD}, sk_{SD}, sk_s)。

–Corrupt.TK(i):当接收到关于内容 i 的质询，B 检查第 i 个元组 $(i, S, Y_{DU}, sk_{DU}, sk_s, tk_s)$ 是否在列表 T 中存在。若不是，B 会给 A 返回终止。否则，B 返回 tk_S 给 A。

–AUpdate (i, γ, ψ) :当从 A 处接收到关于内容 (i, γ, ψ) 的质询，B 检查第 i 个元组 $(i, S, Y_{DU}, sk_{DU}, sk_s, tk_s)$ 是否在列表 T 中存在并且 $\gamma \in S$。若不是，B 会给 A 返回终止。否则，B 设置 $S' \leftarrow S / \{\gamma\}, S' \leftarrow S' \bigcup \{\psi\}$，$D[i] \leftarrow S'$。然后，B 检查是否 $S' | \neq (M*, \rho*)$。若满足，B 发送属性更新质询 (γ, ψ) 给 C，并使用后者返回的 $UK_{\gamma \to \psi}$ 更新属性密钥为 $sk_{s'}$。同时，用 $sk_{s'}$ 重新计算 $tk_{s'}$。最后，B 将元组更新为 $(i, S', Y_{DU}, sk_{DU}, sk_{s'}, tk_{s'})$。

Challenge：敌手 A 提交两个相等长度的消息 m_0 和 m_1。B 检查 $S \in D$ 时，是否满足 $S | \neq (M*, \rho*)$。若满足，B 随机掷一个硬币 $\theta \in [0,1]$，并基于访问策略 $M*$ 加密 m_θ 获得挑战密文 $CT_\theta*$。最后，B 将密文 $CT_\theta*$ 发送给 A。

Phase2：与 Phase1 类似。

Guess：A 输出 θ 的一个猜测值 θ'。如果 $\theta = \theta'$，那么称 A 赢得了该游戏。

定义 2 若无多项式时间内攻击者 A 能以不可忽略的优势来攻破上述安全模型，那么可以认为方案是 IND–CPA 安全的。

三、具体方案

3.1 方案构造

本文提出的方案具有策略隐藏，外包解密可验证，用户属性可更新，恶意解密可限制等特点，具体如下：

1）$Setup(1^\lambda) \to (mpk, msk)$：$G$ 和 G_1 为两个阶为素数 p 的乘法循环群，g 是 G 的一个生成元，$e: G \times G \to G_1$ 是一个双线性映射。定义两个抗碰撞的哈希函数：$H: \{0,1\}^* \to z_p^*, H_1: \{0,1\}^* \to G$。$A$ 是系统中的属性域。对于每个属性 $i \in A$，系统随机选取 $t_i \in z_p^*$ 并计算 $T_i = H_1^{t_i}(i)$。给定外包解密次数 N，初始化外包解密计数器 $ctr = 0$，构建用户的访问列表 AL_{DU} 记录相关的访问参数。系统主私钥为 $msk = \{\alpha, \beta, \{t_i\}_{i \in A}\}$，系统公共参数为

$mpk=\left\{G,G_1,p,e,g,g^{\alpha},e(g,g)^{\beta},\left\{T_i=H_1^{t_i}(i)\mid i\in A\right\},H,H_1,AL_{DU},N\right\}$。

2）$SDKeyGen(mpk)\rightarrow(Y_{SD},sk_{SD})$：SD 输入公共参数 mpk，并随机选择 $z_{SD}\in z_p^*$，计算公私钥对：$Y_{SD}=g^{z_{SD}},sk_{SD}=z_{SD}$。

$DOKeyGen(mpk)\rightarrow(Y_{DO},sk_{DO})$：DO 输入系统公共参数 mpk。系统随机选择 $z_{DO}\in z_p^*$，生成公私钥对:$Y_{DO}=g^{Z_{DO}},sk_{DO}=z_{DO}$。

$DUKeyGen(mpk,msk,ID,S)\rightarrow(ID',Y_{DU},sk_{DU},sk_s)$：①ES 输入系统公共参数 mpk、系统主私钥 msk、身份 ID 和用户属性集 S，系统随机选择 $z_{DU}\in z_p^*$，计算用户公私钥对：$Y_{DU}=g^{z_{DU}},sk_{DU}=z_{DU}$，并计算协商密钥 $ck_{DU}=(g^{z_{DU}})^{\alpha}$，生成匿名身份 $ID^{'}=ID\oplus H(ck_{DU})$。②给定属性集 $S=\left\{S_1,S_2,\cdots,S_n\right\}$，系统进行盲化操作，并返回盲化后属性集 $R_{att}=\{T_j\mid 1\le j\le n\}$。随机选择 $h,\pi,\{h_j\}_{j=1}^{|S|}\in z_p^*$，计算 $K_0=Y_{DU}^{\beta}Y_{SD}^{\pi}g^{\alpha h}$，$K_1=g^{\pi}$，$\{K_{2,j}=g^hT_j^{h_j}=g^hH_1^{t_jh_j}(S_j),K_{2,j}^{'}=g^{\alpha h_j}\}_{j\in[1,n]}$，得到属性密钥：$sk_s=(R_{att},K_0,K_1,\left\{K_{2,j},K_{2,j}^{'}\mid j\in[1,n]\right\})$。

3）*Encrypt*：包含策略隐藏算法、预加密算法和外包加密算法。

①策略隐藏算法：ES 首先对系统属性域中的属性进行盲化处理。为每个属性 i 选取随机数 T_i 作为盲化值，并建立表单 L_T 将属性和盲化值一一对应存储。当收到 DO 的访问策略 (M,ρ)，ES 选择一个随机数 Θ，并将单射函数 ρ 转换为 $\rho':M_i\rightarrow H(T_{att}\|\Theta)$。然后，ES 将访问策略 (M,ρ') 公开，将随机数 Θ 发送给 DO，并上传至区块链。

② 预加密算法：首先，DO 运行对称加密算法加密 m，计算 $\overline{C}=E_{KEY}(m)$ 为对称加密密文。然后，DO 使用定义访问策略 (M,ρ')，其中 M 是一个 $l\times k$ 的矩阵，函数 ρ' 是将矩阵 M 的行关联到 $H(T_{att}\|\Theta)$ 的一个映射。DO 选择随机向量 $\vec{v}=(r,y_2,\cdots,y_k)$，其中 r 是秘密分享值，$y_2,\cdots,y_k$ 是随机选取的，对于 $j\in[1,l]$，计算 $\lambda_j=\vec{v}\cdot M_j$，其中 M_j 是 M 的第 j 行。最后计算：$\overline{C}=E_{KEY}(m),\hat{C}=KEY\cdot e(g,g)^{\beta r},C=g^r$,并将访问策略 (M,ρ')，秘密份额 λ_j 和预加密密文 $CT_{pre}=\{\overline{C},\hat{C},C,\}$ 发送给 SD。

③ 外包加密算法：对于 $j\in[1,l]$，SD 计算：$C_j=g^{\alpha\lambda_j}$，$C_j^{'}=T_j^{\lambda_j}=H^{t_j\lambda_j}{}_1(S_j)$。输出完整密文 $CT=((M,\rho'),\overline{C},\hat{C},C,\{C_j,C_j^{'}\}_{j\in[1,l]})$。

DO 将密文 CT 上传到 IPFS，并得到 IPFS 返回的哈希地址 h_{addr}。DO 计算哈希值 $h^{'}=H(e(g,g)^{\beta r})$，并将 h_{addr}，$h^{'}$，(M,ρ') 和 Θ 作交易通过授权合约上传到以太坊区块链上。区块链上主要包含合约如下：

①ES 通过部署用户合约管理系统中的合法用户，主要包含两个函数。

upUser()，此函数负责更新用户列表，将合法的用户地址放入授权列表中。

getUser()，此函数通过返回 true 和 false，判断用户是否为系统合法用户。

②DO 通过部署授权合约，实现相关参数的存储，主要包含三个函数。

setStorage()，此函数函数只有 DO 才能调用，DO 调用这个函数将相关参数存储在区块链上，此函数会记录下 DO 的以太坊地址，当 DO 想更新参数时，只需调用该函数，进行简单修改即可；

judge()，此函数通过调用用户合约中的 getUser()函数，判断用户是否拥有数据的读取权限；

getStorage()，DU 调用该函数来读取存储在区块链上的参数。

Algorithm 1: Storage Transaction

Input: The identifier S of the storage transaction;
The storage address h_{addr} of the ciphertext;
The access policy (M,ρ') of the ciphertext;
The random number Θ of the access policy;
The hash value $h^{'}$ of the result verification;
The private key SK_{DO} of the DO;

Output: The storage transaction $Tx_{storage}$;

1 /*Compute the message digest for transaction*/

$MD=H(S,h_{addr},h^{'},(M,\rho'),\Theta)$;

2 /*Encrypt the message digest with the SK_{DO} */

$sign=Sign_{SK_{DO}}(MD)$;

3 /*Generate the storage transaction*/

$Tx_{storage}=\{S,h_{addr},h^{'},(M,\rho'),\Theta,sign\}$;

4 **return** $Tx_{storage}$

如算法 2 为存储交易的生成过程。$Tx_{storage}$ 被广播到区块链上，用户可以通过签名来验证交易的有效性。

Algorithm 2: Storage verify Transaction.

Input: $Tx_{storage}=\{S,h_{addr},h^{'},(M,\rho'),\Theta,sign\}$;

The public key PK_{DO} of the DO registered in the CB;

Output: The validation of ; $Tx_{storage}$

1 /*Compute the message digest for transaction*/

$MD^{'}=H(S,h_{addr},h^{'},(M,\rho'),\Theta)$;

2 /*Verify the sign with the public key*/

$MD=Compute_{PK_{DO}}(sign)$;

3 **if** $MD^{'}=MD$ **then**

4 **return** True;

4）*Decrypt*：若 DU 想要访问数据，他首先从区块链上获取随机数 Θ ,并计算 $H(T_{att}\|\Theta)$。如果 DU 的属性满足访问策略 (M,ρ')，DU 向 SD 发送外包密钥 tk_S，SD 检查 DU 的访问次数是否超过 N 次，若访问次数超过，则向系统发送追踪申请；否则继续检查 DU 属性是否满足访问策略，若用户不满足，则拒绝其访问请求；否则执行外包解密并将部分解密结果返回给 DU，DU 验证外包解密结果的正确性，并恢复得到对称密钥。

① *OutDec* 算法由数据用户和智能设备执行。

(i) $KeyGen_{out}(mpk,sk_s,sk_{DU},Nonce)\rightarrow(tk_S)$：外包密钥生成算法由为 DU 运行。首先，DU 计算一次性随机数 $Nonce=H(ctr+1\|Timestamp)\in\{0,1\}$，$K_r=e(g,g)^{\frac{1}{(z_{DU}+\mathrm{Nonce})}}$，$K_p=g^{\frac{1}{(z_{DU}+Nonce)}}$,生成验证密钥 $tk_1=(Nonce,K_r,K_p)$。然后，输出外包解密密钥 $tk_S=(sk_s,tk_1)$。

(ii) $PreDec(CT,tk_S,sk_{SD},AL_{DU})\rightarrow(PCT)$：SD 首先验证 DU 的访问次数，即是否满足以下条件：（1）$e(g^{H(\mathrm{Nonce})}\cdot Y_{DU},K_p)=e(g,g)$ 和 $K_r=e(g,K_p)$ 成立。（2）$ctr+1\le N$。（3）$K_r\notin AL_{DU}$。其中 $AL_{DU}=\{ctr,K_r\}$。如果满足，则更新 $ctr\leftarrow ctr+1$ 并存储 K_r 在 AL_{DU} 中用于下一次解密，并进行如下计算：

定义 $I=\{M_j:M_j\subseteq M,\rho'(M_j)=H(T_{att}\|s)\}$，则在一个多项式时间内可以计算出系数 w_j，使得 $\sum_{j\in I}w_j\lambda_j=r$。

$$PCT=\frac{e(K_0,C)}{e(K_1^{sk_{SD}},C)\prod_{j\in I}(\frac{e(K_j,C_j,)}{e(K_j^{'},C_j^{'})})^{w_j}}=(Y_{DU},g)^{\beta r}$$

。SD 将部分解密结果 PCT 发送给 DU，之后，SD 生成关于 DU 访问次数的承诺 $com(ctr,\tau)=g^{H(ctr)}h^{H(\tau)}$，其中 τ 为随机安全参数并将 τ 发送给 DU。SD 生成访问交易将承诺上传到区块链中作为 DU 访问次数的效验证据，并返回 $TxID$，如算法 3 所述。DU 利用 $TxID$ 检查计数器是否被 SD 恶意更改。

Algorithm 3: Access Transaction

Input: The identifier A of the access transaction;

The storage address h_{addr} of the ciphertext;

The commitment $com(ctr,\tau)$ of the access times;

The random result K_c of VRF calculation;

The private key SK_{SD} of the SD;

Output: The storage transaction Tx_{access};

1 Get the current time $time$;

2 /*Compute the message digest for transaction*/

$MD=H(A,PK_{DU},h_{addr},K_c,time)$;

3 /*The SD signs the transaction */

$sign = Sign_{SK_{SD}}(MD)$;

4 $Tx_{access} = \{A, PK_{DU}, h_{addr}, K_c, time, sign\}$;

5 **return** Tx_{access}

②$DUDec(PCT, sk_{DU}) \to (KEY)$：DU 接收到 PCT 用自己的私钥 sk_{DU} 计算 $CT' = PCT^{-sk_{DU}} = PCT^{-z_{DU}} = e(g,g)^{\beta r}$。计算结果与区块链中存储的哈希值对比，若 $h' = H(CT')$，则外包解密结果验证正确。然后，计算 $KEY = \frac{\hat{C}}{CT'}$，并使用密钥 KEY 计算 $m = D_{KEY}(\bar{C})$，得到明文 m。若不相等，则说明外包计算有误。

5）$Trace(ID', msk) \to (ID)$：当检测到用户恶意发起大量外包请求，SD 会向 ES 发送对该用户的追踪请求，ES 首先计算协商密钥 $ck_{DU} = (g^{z_{DU}})^{\alpha}$，计算 $ID = ID' \oplus H(ck_{DU})$，得到用户的真实身份 ID。

6）$Update(S_j \to S_\psi) \to (UK_{j\to\psi}, UK_j', UK_j'')$：当 DU 请求变更属性时，ES 运行更新算法。假设 DU 申请将属性值 S_j 更新为 S_ψ，则 ES 生成如下更新密钥：

（1）选择 $\tilde{t_j} \neq t_j \in z_p, UK_{j\to\psi} = H_1^{-t_j\beta}(S_j) H_1^{t_\psi \beta}(S_\psi)$

（2）$UK_j' = H_1^{\beta(\tilde{t_j} - t_j)}(S_j), UK_j'' = H_1^{h_j(t_j - \tilde{t_j})}(S_j)$

（3）输出 $UK_{j\to\psi}$，UK_j'，UK_j''

若 DU 需要将更新属性，ES 为其发送密钥 $UK_{j\to\psi}$，则 DU 执行 $K_\psi \leftarrow K_j \cdot UK_{j\to\psi}$。

若 DU' 具有属性 S_j 但不需要更新属性，ES 为其发送密钥 UK_j'，则 DU' 执行 $K_j \leftarrow K_j \cdot UK_j'$。

ES 向 SD 发送密钥收到 UK_j''，SD 执行 $C_j' \leftarrow C_j' \cdot UK_j''$。

3.3 正确性分析

解密阶段正确性：

$$
\begin{aligned}
PCT &= \frac{e(K_0, C)}{e(K_1^{sk_{SD}}, C) \prod_{j\in I} \left(\frac{e(K_j, C_j)}{e(K_j', C_j')}\right)^{w_j}} \\
&= \frac{e(Y_{DU}^{\beta} Y_{SD}^{\pi} g^{\alpha h}, g^r))}{e(g^{\pi Z_{SD}}, g^r) \prod_{j\in I} \left(\frac{e(g^h H_1^{t_j h_j}(S_j), g^{\alpha\lambda_j})}{e(g^{\alpha h_j}, H_1^{t_j\lambda_j}(S_j))}\right)^{w_j}} \\
&= \frac{e(Y_{DU}^{\beta} g^{\alpha h}, g^r)}{\prod_{j\in I} \left(\frac{e(g^h, g^{\alpha\lambda_j}) \cdot e(H_1^{t_j h_j}(S_j), g^{\alpha\lambda_j})}{e(g^{\alpha h_j}, H_1^{t_j\lambda_j}(S_j))}\right)^{w_j}} \\
&= \frac{e(Y_{DU}^{\beta} g^{\alpha h}, g^r)}{\prod_{j\in I} (e(g^h, g^{\alpha\lambda_j}))^{w_j}} \\
&= \frac{e(Y_{DU}^{\beta} g^{\alpha h}, g^r)}{e(g,g)^{h\alpha r}} \\
&= (Y_{DU}, g)^{\beta r}
\end{aligned}
$$

计算 $\frac{\hat{C}}{PCT^{-sk_{DU}}} = \frac{\hat{C}}{PCT^{-Z_{DU}}} = \frac{KEY \cdot e(g,g)^{\beta r}}{e(g,g)^{\beta r}} = KEY$，得出对称密钥，因此解密算法满足正确性。

四、安全性分析

定理 1：在标准模型下，假设张等人的 CP-ABE 方案[4]（定义为 $\sum_{CP-ABE}$）基于 DBDH 假设满足选择性 CPA 安全，则我们的方案满足选择性 CPA 安全，即不存在多项式敌手 A 能够以不可忽略的优势攻破我们的方案。

证明：假设 A 是我们方案的敌手，C 是 $\sum_{CP-ABE}$ 方案的挑战者，B 是构造的规约算法，同时参与两个游戏：在我们的案中 B 模拟挑战者和 A 执行选择性 CPA 安全游戏；在 $\sum_{CP-ABE}$ 方案中 B 模拟敌手和 C 执行选择性 CPA 安全游戏。游戏过程如下：

Init：A 选择一个挑战访问策略 (M^*, ρ^*)，并将其发送给 B。B 再将 (M^*, ρ^*) 发送给 C。

Setup：当接收到 C 发送给 B 的 $\sum_{CP-ABE}$ 方案的公共参数 mpk，B 选择 $z_{SD} \in z_p^*$，并设置 $Y_{SD} = g^{z_{SD}}$，$sk_{SD} = z_{SD}$。然后 B 发送 (mpk, Y_{SD}, sk_{SD}) 给 A。

Phase1：在此阶段，允许 A 自适应的进行以下质询：

–Create(S):A 可以查询一系列与属性集合 S 有关的密钥，即 A 将属性集合 S 发送给 B，B 检查是否 $S \models (M^*, \rho^*)$。如果是，则 B 发送属性集合 S 给 C，并接收 C 返回的 $\overline{sk}_s = (\overline{K}_0, \overline{K}_1, \overline{K}_2, \{\overline{K}_{3,i}, \overline{K}'_{3,i}\}_{i \in S}) = (Y_{DU}^{\beta} g^{\alpha h}, g^{(\alpha+\beta+\alpha h)ID}, g^{\alpha+\beta+\alpha h}, g^h H_1^{t_i h_i}(i), g^{\alpha h_i})$。B 选择 $z'_{DU}, \pi \in z_p^*$，计算 $Y_{DU} = g^{Z'_{DU}}$，$sk_{DU} = z'_{DU}$ 然后设置

$$sk_s = (S, K_0 = \overline{K}_0^{Z'_{DU}} Y_{SD}^{\pi}, K_1 = g^{\pi}, \{K_{2,i} = (\overline{K}_{3,i})^{Z'_{DU}}, K'_{2,i} = (\overline{K}'_{3,i})^{Z'_{DU}}\}_{i \in S},)$$
$$= (S, K_0 = Y_{DU}^{\beta} Y_{SD}^{\pi} g^{\alpha h Z'_{DU}}, K_1 = g^{\pi}, \{K_{2,i} = g^{h Z'_{DU}} H_1^{t_i h_i Z'_{DU}}(S_i), K'_{2,i} = g^{\alpha h_i Z'_{DU}}\}_{i \in S},)。$$

B 设置 $tk_s = (sk_s, tk_1(Nonce))$，$j = j+1$ 并将 $(j, S, Y_{DU}, sk_{DU}, sk_s, tk_s)$ 记录在表 T 中。

–Corrupt.SK(i):当 B 从 A 处接收到关于内容 i 的质询，B 检查第 i 个元组 $(i, S, Y_{DU}, sk_{DU}, sk_s, tk_s)$ 是否在列表 T 中存在。若不是，B 会给 A 返回终止。否则，B 设置 $D[i] \leftarrow D[i] \cup \{S\}$ 并给 A 返回 (Y_{SD}, sk_{SD}, sk_s)。

–Corrupt.TK(i):当接收到关于内容 i 的质询，B 检查第 i 个元组 $(i, S, Y_{DU}, sk_{DU}, sk_s, tk_s)$ 是否在列表 T 中存在。若不是，B 会给 A 返回终止。否则，B 返回 tk_S 给 A。

–AUpdate(i, γ, ψ):当从 A 处接收到关于内容 (i, γ, ψ) 的质询，B 检查在列表 T 中是否存在第 i 个元组 $(i, S, Y_{DU}, sk_{DU}, sk_s, tk_s)$ 使得 $\gamma \in S$。若没有，B 会给 A 返回终止。否则，B 设置 $S' \leftarrow S / \{\gamma\}, S' \leftarrow S' \cup \{\psi\}$，$D[i] \leftarrow S'$。然后，B 检查是否 $S' \models (M^*, \rho^*)$。若满足，B 发送属性更新质询 (γ, ψ) 给 C，并使用 C 返回的 $UK_{\gamma \to \psi}$ 更新属性密钥为 $sk_{s'}$。同时，B 用 $sk_{s'}$ 重新计算 $tk_{s'}$。最后，B 将元组更新为 $(i, S', Y_{DU}, sk_{DU}, sk_{s'}, tk_{s'})$。

Challenge：A 递交两个长度相等的消息 m_0 和 m_1 给 B，B 检查 $S \in D$ 时，是否满足 $S \models (M^*, \rho^*)$。若满足，则 B 将 m_0 和 m_1 发送给 C。C 随机掷一个硬币 θ，$\theta \in [0,1]$，并使用使用 (M^*, ρ^*) 加密 m_θ，最后 B 将挑战密文 $CT_\theta^* = ((M^*, \rho^*), \hat{C}, C, \{C_i, C'_i\}_{i \in [1,l]})$ 发送给 A。

Phase2：与 Phase1 相同。

Guess：A 输出 θ 的一个猜测值 θ'，B 将 θ' 作为猜测发送给 C。

显然，如果 A 能够以不可忽略的优势攻破本文提出的方案，那么 B 也能以不可忽略的优势攻破 $\sum_{CP-ABE}$ 方案。换句话说，B 在 A 的帮助下打破了 $\sum_{CP-ABE}$ 方案的选择性 CPA 安全。

文献[2]中的选择性 CPA 安全被分为两种类型：针对 type Ⅰ敌手的选择性 CPA 安全和针对 type Ⅱ敌手的选择性 CPA 安全。上述证明等价于证明我们的方案满足对 type Ⅰ敌手的选择性 CPA 安全。事实上，我们的方案还满足针对 type Ⅱ敌手的选择性 CPA 安全，证明方法类似，除了在设置阶段，A 需要自己生成 $Y_{SD} = g^{z_{SD}}, sk_{SD} = z_{SD}$ 并发送 Y_{SD} 给 B。

根据 DBDH 假设任何攻击者无法以不可忽略的概率优势攻破 DBDH 问题。因此，该方案中不存在敌手 A 能够以不可忽略的优势在多项式时间内打破 IND–CPA 模型。

五、性能分析

本章节通过功能比较和计算开销将现有方案和本方案进行了对比分析。利用 PBC 库对方案进行仿真实验。

5.1 功能比较

将方案[2][3][1][4][5]与我们的方案进行比较。从表 2 可以看出，只有方案[1]和我们的方案实现了策略隐藏，保护用户隐私信息；方案[2][3]和我们的方案限制了用户的恶意访问；方案[2][3][4]和我们的方案实现了用户的属性更新；此外，只有我们的方案同时实现了外包加解密计算，并且具有用户可以独立验证外包解密。方案[1][5]和我们的方案运用了区块链技术，确保数据的正确存储。

表 2 功能比较

方案	访问策略	策略隐藏	恶意访问限制	属性更新	外包加密	用户验证外包解密	区块链
文献[2]	LSSS	×	√	√	×	×	×
文献[3]	LSSS	×	√	√	×	√	×
文献[1]	LSSS	√	×	×	×	√	√
文献[4]	Tree	×	×	√	×	×	×
文献[5]	LSSS	×	×	×	×	×	√
本方案	LSSS	√	√	√	√	√	√

5.2 计算开销

我们方案的实验环境为 64 bit Ubuntu 14.04 操作系统、

Intel Core i7-3770CPU (3.4GHz)、内存 4GB，实验代码基于 Pairing-based Cryptography Library (PBC-0.5.14)和 cpabe-0.11 进行修改与编写，并且使用基于 512 bit 有限域上的超奇异曲线 $y^2 = x^3 + x$ 中的 160 bit 椭圆曲线群。为了方便描述，我们用 T_G 表示一次 G 域指数运算，用 T_{G_1} 表示一次 G_1 域指数运算，用 T_P 表示一次双线性对运算。

表 3 计算代价比较

方案	KeyGen	Encrypt(DO)	Decrypt(DS)	Decrypt(U)
文献[2]	$(4n+6)T_G$	$(5n+2)T_G$	$T_G + 3nT_{G_1} + (3n+2)T_P$	T_{G_1}
文献[1]	$(3n+3)T_G$	$(3n+2)T_G + T_{G_1} + T_P$	$nT_{G_1} + (2n+2)T_P$	T_{G_1}
文献[5]	$(n+7)T_G$	$(3n+1)T_G + T_{G_1} + T_P$	–	$(2n+2)T_G + nT_{G_1} + (2n+2)T_P$
本方案	$(2n+5)T_G$	$T_G + T_{G_1} + T_P$	$T_G + nT_{G_1} + (2n+2)T_P$	T_{G_1}

该阶段仿真了密钥生成、用户加密、外包解密、用户解密阶段的计算开销。如图 3(a-d)所示，测试属性数量从 0 增加到 60 的条件下，各算法操作时间的比较。如图 3 所示，除了密钥生成算法外，我们的计算成本都低于方案[2][1][5]。这是因为我们的方案同时使用了外包加密和解密技术，而密钥生成算法的计算成本仍然在可接受的范围内。我们的方案在用户加密算法方面具有明显的优势。如图 3(b)所示，我们方案的计算成本是很小的常数，但方案[2][1]和[5]的计算成本随属性数量呈线性增长。如图 3(c)所示，由于方案[5]不使用外包解密算法，因此不参与外包解密时间的比较。如图 3(d)所示，我们的用户端解密计算成本与方案[2][1]的性能相当。综上所述，我们的方案在实际应用中是高效的、可行的。

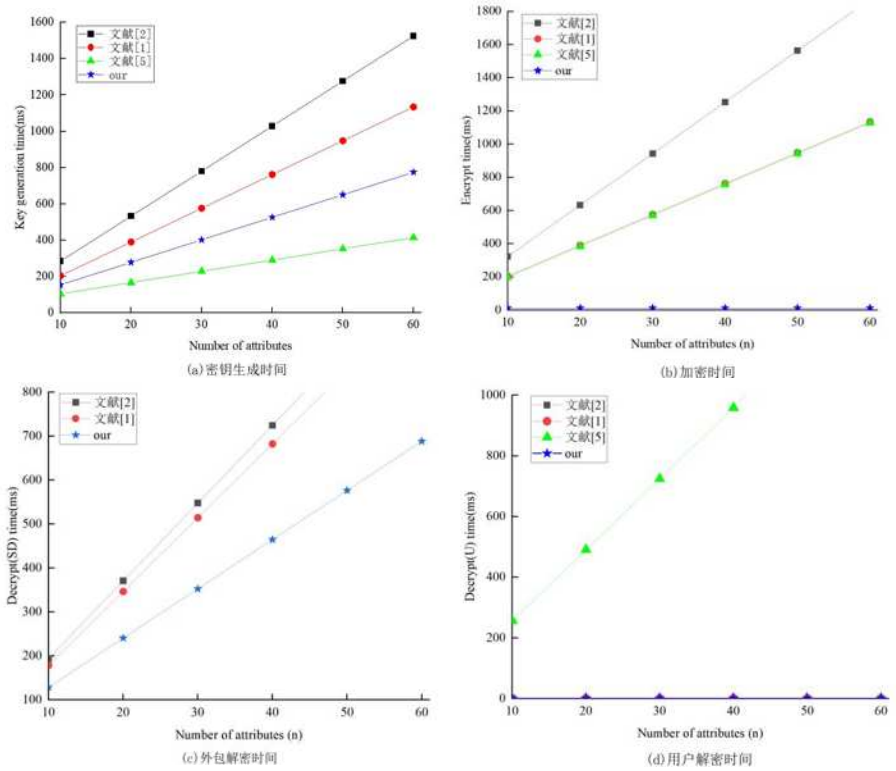

图 3 计算代价与仿真时间对比

六、结束语

我们提出了一种支持策略隐藏的可验证外包解密访问控制方案。该方案使用智能设备实现外包加解密技术，并引入了区块链记录验证参数，实现了用户的自认证和智能设备的不否定，保障了解密结果的安全性问题。分析结果表明，与现有的 CP-ABE 方案相比，该方案减少了用户端的计算负担，并通过转变线性秘密共享矩阵的映射函数实现策略隐藏，避免了用户敏感信息的泄漏问题。安全性分析证明了在实际应用中的安全性，仿真实验证明了该方案的有效性。如何实现高效的用户撤销和属性撤销方案，将是我们下一步的重点研究工作。

参考文献

[1] K.Fan, Q.Pan, K.Zhang, et al. A secure and verifiable data sharing scheme based on blockchain in vehicular social

networks[J]. IEEE Transactions on Vehicular Technology, 2020, 69(6): 5826–5835.

[2] J. Ning, Z. Cao, X. Dong, et al, Auditable σ –time outsourced attribute–based encryption for access control in cloud computing, IEEE Trans. Inf. Forensics Secur. 13 (1) (2018) 94 – 105.

[3] X.Liu, H.Wang, B.Zhang, et al. An efficient fine–grained data access control system with a bounded service number[J]. Information Sciences, 2022, 584: 536–563.

[4] P. Zhang, Z. Chen, K. Liang, et al. A cloud–based access control scheme with user revocation and attribute update. In: Proceedings of ACISP 2016, LNCS 9722, Springer, Cham, 2016: 525–540.

[5] Y.Zuo, Z.Kang, J.Xu, et al. BCAS: A blockchain–based ciphertext–policy attribute–based encryption scheme for cloud data security sharing[J]. International Journal of Distributed Sensor Networks, 2021, 17(3): 1550147721999616.

[6] X. Mao, J. Lai, Q. Mei, K. Chen, J. Weng, Generic and efficient constructions of attribute–based encryption with verifiable outsourced decryption, IEEE Trans. Dependable Secure Comput. 13 (5) (2016) 533 – 546.

作者简介

常晓（1996 年 10 月—），女，汉族，陕西人，西安理工大学硕士研究生，研究方向：密码理论与信息安全。

孙瑾（1977 年 6 月—），女，汉族，安徽人，西安理工大学副教授，博士，研究方向：密码理论与信息安全。

基金项目: Natural Science Basic Research Program of Shaanxi (Program No. 2021JM-341) （Program No.2021JQ-485）