计 算 机 研 究 与 发 展
Journal of Computer Research and Development
DOI: 10.7544/issn1000-1239.202220133
61(12): 3088−3097, 2024

支持策略更新和即时密文验证的外包属性基加密方案

苏泽林　张文芳　王小敏
（西南交通大学信息科学与技术学院　成都　610756）
（1005719242@qq.com）

Outsourced Attribute-Based Encryption Scheme with Policy Updating and Verifiable Ciphertext

Su Zelin, Zhang Wenfang, and Wang Xiaomin
（*School of Information Science and Technology*, *Southwest Jiaotong University*, *Chengdu* 610756）

Abstract　Attribute-based encryption is a new access control scheme based on cryptography, which is suitable for data sharing. However, the large computational and communication costs of encryption and access policy updating limit the practical application of attribute-based encryption. Moreover, most of proposed outsourcing ABE schemes do not provide a ciphertext correctness verification method for data owners. Thus, an outsourced ABE scheme with dynamic policy updating and real-time verification of ciphertext correctness is proposed to further protect data privacy in an untrusted cloud environment. In the scheme, the design of policy updating references outsourced encryption, which reduces the computational cost of generating update key. The design of ciphertext correctness verification algorithm refers to decryption operation and introduces verification transformation key to make ciphertext verification more efficient. According to different cloud environment models, efficient verification algorithm and strict verification algorithm are designed, which are suitable for honest but curious cloud environment and untrustworthy cloud environment respectively. The scheme is secure against chosen plaintext attack under the standard model. Performance analysis and efficiency comparison show that the computation of local encryption, policy updating and ciphertext verification are reduced, and the scheme is more lightweight, which is suitable for the application of computation-constrained devices in access control scenarios.

Key words　attribute-based encryption; policy updating; ciphertext re-encryption; outsourced encryption; verifiable ciphertext

摘　要　属性基加密提供了全新的基于密码学的访问控制方案，适用于多用户数据共享场景，但由于加密阶段和访问策略更新过程的计算和通信开销较大，且现有的外包属性基加密方案大多数都没有提供面向数据拥有者的密文正确性验证方法，很大程度上限制了属性基加密的实际应用. 针对上述问题，提出了一种支持动态策略更新和即时验证密文正确性的属性基外包加密方案，能够在不可信云环境下有效地保护数据的隐私性. 方案根据外包加密原理设计策略更新过程，只需要完成少量计算即可生成更新密钥. 利用双线性对的运算特性和解密运算结构设计密文验证算法，通过引入验证转换密钥使密文验证效率明显高于解密运算效率. 方案根据不同的云环境模型设计了高效验证算法和严格验证算法，分别适用于诚实且好奇和不可信的云环境中. 方案在标准模型下被证明满足选择明文攻击安全性. 性能分析和效率对比表

收稿日期：2022-01-29；**修回日期**：2024-05-21
基金项目：国家自然科学基金项目（U2468201）；四川省科技计划项目（2024ZHCG0001）
This work was supported by the National Natural Science Foundation of China (U2468201) and the Key Program for Sichuan Science and Technology (2024ZHCG0001).
通信作者：张文芳（wfzhang@swjtu.edu.cn）

明，该方案的本地加密、策略更新和密文验证的计算量都有所减少，使得整体方案较现有方案更加轻量化，适用于资源受限设备的数据共享场景.

关键词 属性基加密；策略更新；密文重加密；外包加密；可验证密文

中图法分类号 TP30

随着知识经济和信息时代的到来，云计算成为大数据时代的关键技术，研究具备访问控制能力的加密方案具有良好的应用前景. 基于属性的加密（attribute-based encryption, ABE）体制由 Sahai 等人[1]在 2005 年提出，通过模糊身份的一对多加解密可以实现灵活细粒度的访问控制. 数据使用者的身份信息被一组描述性的属性代替，只有当用户的属性集合与数据的访问结构相匹配时，才能解密出明文. 根据密文的生成方式不同，属性基加密方案可分为密钥策略加密（KP-ABE）和密文策略加密（CP-ABE）. KP-ABE 由 Goyal 等人[2]提出，密钥对应访问结构，密文对应属性集合，其访问结构采用访问树，具有更强的逻辑表达性，因此被广泛采用. CP-ABE 方案由 Bethencourt 等人[3]在 2008 年提出，其密文对应访问结构，密钥对应属性集合，即用访问结构加密、用属性集合解密. 2008 年，Waters[4]将线性秘密共享技术引入了属性基方案中，提出了更加灵活的访问结构表达方式，并提高了系统运行效率.

在属性基加密方案中，加解密计算代价会随着属性数量和访问策略的复杂度线性增长，这无疑给终端设备带来庞大的计算负担，严重限制了属性基加密算法在资源受限设备上的应用. 为了解决加解密计算量过大的问题，Green 等人[5]提出了属性基外包解密方案. Li 等人[6]提出了外包加密方案，该方案借助 Map Reduce 模型实现，但是用户在加密时仍然需要进行大量的模幂运算. Zhang 等人[7]提出的外包加密方案中借助了 2 个不同的服务器完成外包计算，数据拥有者结合 2 个服务器生成的随机数作为陷门进行加密，该方案的外包计算效率有所提高，但需要确保 2 个服务器之间不能进行合谋，否则将导致加密陷门泄露. 文献 [8] 对文献 [7] 方案进行了改进，提出了混合访问策略，但是仍然难以抵抗服务器和数据使用者的合谋攻击. 文献 [9] 的外包方案有效减少了用户端的幂运算，用户只需执行一定数量的乘法运算，文献 [10] 在安全性方面对该方案进行了完善.

在外包计算过程中，负责计算的云服务器可能因为“偷懒”或受到敌手攻击，影响计算结果的准确性，因此数据拥有者需要具备验证外包加密密文正确性的能力. 文献 [8–10] 只设计了解密阶段密文正确性的检测算法，没有设计面向数据拥有者的检测算法，导致外包加密服务器的安全风险检测滞后. 文献 [11] 给出了一种针对数据拥有者的外包密文验证方法，但验证计算量很大. 文献 [12] 将模幂指数安全外包算法应用到属性基加密方案中，能够实现细粒度的外包计算，并且将外包加密和密文验证合并，但缺点是引入了过多冗余计算和交互，导致方案效率降低. 文献 [13] 将指数批量打包验证算法用于密文验证，该算法仅适用于幂次结构中底数相同而指数不同的形式，若底数不同则会增加大量指数运算，此类指数打包验证算法不适用于部分小属性集的属性基方案. 文献 [14–15] 提出的可验证外包加密方案架构中，设置了第三方可信实体作为外包加密密文的检测机构，但密文检测服务器易成为系统安全和效率瓶颈. 文献 [16] 提出基于 BLS 短签名的可验证外包方案，但密文的正确性需要数据使用者在解密和验签之后才能验证. 文献 [17] 给出一个基于边缘节点的外包解密方案，但密文的正确性验证仍需数据使用者完成. 2022 年，Hahn 等人[18]提出了外包解密验证的 2 种方案，使用验证密钥以防止伪造和重用有效承诺，但同样需要数据使用者完成验证. 所以，文献 [16–18] 也不能实现数据拥有着对密文的正确性检测.

除了上述外包计算结果的验证需求之外，数据拥有者在将数据存储到云端后，可能会动态更改密文中的访问策略，以进一步保护数据的隐私和安全. 然而，数据拥有者对云端密文进行访问策略更新时，往往存在计算开销高、存储消耗大、更新后的密文难以验证等问题. 为了解决计算和存储效率问题，可以利用初始加密生成的密文参数来更新访问策略[11,19-23]. 2014 年，Yang 等人[11]提出了支持密文访问策略更新的属性基加密方案，还提出了更新后密文的正确性检测算法，云服务器根据用户生成的更新密钥和新的访问结构，对旧密文进行重新计算从而完成访问策略更新. 仿真实验表明，相较于重新加密明文的方案，该策略更新方案降低了 20%~50% 的计算量，但在策略更新和密文检测的计算效率上还有进一步提升的空间. 文献 [19] 将半策略隐藏与策略更新方案

相结合，避免了在策略更新时泄露策略中的敏感信息. Sethi 等人[21]引入了密钥可追踪机制，能够检测到参与泄露解密密钥的恶意用户. 文献 [23] 引入匿名密钥分发协议，提出了一种支持策略更新的多机构方案. 在文献 [11] 的基础上，后续研究[19-23]虽在安全性上进行了改进，但更新密钥的生成阶段仍然保留了大部分指数运算，导致其策略更新效率难以提升，并且这些方案均未给出更新后密文的正确性检测方法.

针对上述问题，本文提出一个支持策略更新和即时密文验证的高效属性基外包加密方案. 首先，将策略更新与外包加密相结合，通过拆分外包参数，将策略更新阶段的指数运算安全地外包给云端，数据拥有者仅需付出少量代价就生成密文更新密钥. 其次，针对外包加密密文和策略更新后密文的正确性检测效率和精度问题，设计了面向数据拥有者的密文正确性即时验证算法(包括高效验证和严格验证 2 个算法)，该算法将解密阶段的部分运算结构和双线性对的运算特性进行结合，同时引入验证转换密钥以简化运算过程，不需要在线交互，也不依赖可信第三方，使得数据拥有者在无需恢复秘密值的前提下具备高效的密文验证能力和严格的错误检查能力，使之能及时发现云服务器在外包加密和密文更新过程中的无意或有意的密文构造错误，提高了可验证外包加密的整体安全性和效率. 经验证，所提方案在标准模型下被证明满足选择明文攻击安全性，并且在本地加密、策略更新和密文验证效率方面较现有方案有所提升，同时对密文组件结构类似的属性基方案都具有适用性.

1 相关知识

1.1 双线性对

令G和G_T为 2 个阶是素数p的乘法循环群，g为群G的生成元，双线性映射$e:G\times G\to G_T$. 双线性映射e存在 3 个属性.

1)双线性. 对于任意元素$u,v\in G$和$a,b\in\mathbb{Z}_p$，有$e(u^a,v^b)=e(u,v)^{ab}$.

2)非退化性. $e(g,g)\neq 1$.

3)可计算性. 对于任意元素$u,v\in G$，存在有效算法计算$e(u,v)$.

1.2 访问结构

令$\{P_1,P_2,\cdots,P_n\}$为参与方集合. 令集合$A\subseteq 2^{\{P_1,P_2,\cdots,P_n\}}$是单调的，即满足对于任意集合$B,C$，若$B\in A$并且$B\subseteq C$，那么$C\in A$. 若访问结构(集合)$A$是$\{P_1,P_2,\cdots,P_n\}$的非空子集，即$A\subseteq 2^{\{P_1,P_2,\cdots,P_n\}}\setminus\varnothing$，则集合$A$是授权集合，而不包含$A$的集合是非授权集合.

1.3 线性秘密共享(LSSS)

若一个在集合$\mathcal{P}$上的秘密共享方案是线性的，则需要满足 2 个条件：

1)集合中的每个元素的共享份额可以形成一个$\mathbb{Z}_p$上的向量.

2)存在一个$l\times n$份额的生成矩阵$\boldsymbol{M}$，对于所有的$i=1,2,\cdots,l$，矩阵$\boldsymbol{M}$的第i行表示集合中的一个元素. 定义一个映射函数$\rho(i)$，可以将矩阵$\boldsymbol{M}$的任意一行映射为集合中的一个元素. 选择一个向量$\boldsymbol{v}=(s,r_2,\cdots,r_n)$，其中$s\in\mathbb{Z}_p$表示被共享的秘密，随机选取$r_2,r_3,\cdots,r_n\in\mathbb{Z}_p$，则$\boldsymbol{M}_i\boldsymbol{v}$为秘密份额，其中$\boldsymbol{M}_i$为矩阵$\boldsymbol{M}$的第$i$行.

线性秘密共享方案具有秘密线性重构功能. 假设访问结构为A，令任意属性集合$S\in A$，定义集合$I=\{i:\rho(i)\in S\}$且$I\subset\{1,2,\cdots,l\}$，若$\{\lambda_i\}$是共享秘密值s的有效份额，那么存在常数$\omega_i\in\mathbb{Z}_p$，$i\in I$满足$\sum_{i\in I}\omega_i\lambda_i=s$.

1.4 BDHE 假设(q-bilinear Diffie-Hellman exponent assumption)

根据安全参数，选择阶为素数p的群G，g为群G的生成元，存在双线性映射$e:G\times G\to G_T$. 随机选择数$a,s\in\mathbb{Z}_p$，$T\in G_T$. 若敌手给定$\boldsymbol{y}=\left(g,g^a,\cdots,g^{a^q},g^{a^{q+2}},\cdots,g^{a^{2q}},g^s\right)$，使敌手判断$e(g,g)^{a^{q+1}s}=T$是否成立.

如果对于任何多项式时间的敌手算法$\mathcal{B}$区分$\left(\boldsymbol{y},e\left(g,g\right)^{a^{q+1}s}\right)$和$(\boldsymbol{y},T)$，所具备的优势$Adv_{\mathcal{B}}=\left|Pr\left[B\left(\boldsymbol{y},e\left(g,g\right)^{a^{q+1}s}\right)=1\right]-Pr\left[B\left(\boldsymbol{y},T\right)=1\right]\right|\geqslant\epsilon$是可忽略的，则解决群$G$上的$q$-BDHE 问题是困难的.

2 算法定义与安全模型

2.1 算法定义

本文方案包括 12 个算法，分别为系统初始化$Setup(k,U)$、属性密钥生成$KeyGen(PP,MSK,S)$、加密$Encrypt(PP,m,A)$、解密$Decrypt(CT,SK)$、加密转换密钥生成$ETKGen(PP,U)$、本地加密$Encrypt_{\text{Local}}(PP,m,A,ETK)$、外包加密$Encrypt_{\text{Outsource}}(PP,OP)$、更新密钥生成$UpdateKeyGen(PP,A,A',ETK)$、密文更新$CTUpdate(PP,A,A',UK,CT)$、验证转换密钥生成$VTKGen(PP,U)$、验证$Verify(PP,CT)$、严格验证$Verify_{\text{strict}}(PP,CT,VTK)$. 具体算法定义有：

1)系统初始化$Setup(k,U)\to(PP,MSK)$. 算法由密钥生成中心执行，输入安全参数k和全体属性集合U，输出系统公共参数PP和系统主密钥MSK.

2）属性密钥生成$KeyGen(PP,MSK,S)\rightarrow SK$. 算法由密钥生成中心执行，输入公共参数$PP$、系统主密钥$MSK$、用户属性集合$S$，输出用户属性密钥$SK$.

3）加密$Encrypt(PP,m,A)\rightarrow CT$. 算法由数据拥有者执行，输入公共参数$PP$、明文$m$、访问结构$A$，输出密文$CT$.

4）解密$Decrypt(CT,SK)\rightarrow m$. 算法由数据使用者执行，输入密文$CT$、属性密钥$SK$，输出明文$m$.

5）加密转换密钥生成$ETKGen(PP,U)\rightarrow ETK$. 算法由数据拥有者执行，输入公共参数$PP$、全体属性集合$U$，输出加密转换密钥$ETK$.

6）本地加密$Encrypt_{\text{Local}}(PP,m,A,ETK)\rightarrow(CT_{\text{part}},OP)$. 算法由数据拥有者执行，输入公共参数$PP$、明文$m$、访问结构$A$、加密转换密钥$ETK$，输出部分密文$CT_{\text{part}}$和外包参数$OP$.

7）外包加密$Encrypt_{\text{Outsource}}(PP,OP)\rightarrow CT_{\text{out}}$. 算法由外包加密服务执行，输入公共参数$PP$、外包参数$OP$，输出密文组件$CT_{\text{out}}$.

8）更新密钥生成$UpdateKeyGen(PP,A,A',ETK)\rightarrow UK$. 算法由数据拥有者执行，输入公共参数$PP$、旧的访问结构为$A$，新的访问结构$A'$，加密转换密钥$ETK$，输出更新密钥$UK$.

9）密文更新$CTUpdate(PP,A,A',UK,CT)\rightarrow CT'$. 算法由策略更新服务执行，输入公共参数$PP$、旧的访问结构为$A$，新的访问结构$A'$、更新密钥$UK$、旧密文$CT$，输出新密文$CT'$.

10）验证转换密钥生成$VTKGen(PP,U)\rightarrow VTK$. 算法由数据拥有者执行，输入公共参数$PP$、全体属性集合$U$，输出验证转换密钥$VTK$.

11）验证$Verify(PP,CT)\rightarrow Result$. 算法由数据拥有者执行，输入公共参数$PP$、密文$CT$，输出验证结果$Result$.

12）严格验证$Verify_{\text{strict}}(PP,CT,VTK)\rightarrow Result$. 算法由数据拥有者执行，输入公共参数$PP$、密文$CT$、验证转换密钥$VTK$，输出验证结果$Result$.

2.2 安全模型

本文方案的安全模型是选择明文攻击下的不可区分性（indistinguishability against under chose-plaintext-attack, IND-CPA）游戏，游戏中包含一个挑战者和一个敌手，其中挑战者需要模拟系统运行并回答敌手的询问，具体游戏模型为：

1）Init. 敌手提交要挑战的访问结构$(\boldsymbol{M}^*,\rho^*)$给挑战者.

2）Setup. 挑战者运行系统初始化$Setup$算法，生成公共参数PP，并发送给敌手.

在Init阶段中，敌手向挑战者发起私钥请求，但要求私钥对应的属性集合S不能满足访问结构$(\boldsymbol{M}^*,\rho^*)$.

3）Challenge. 敌手发送2个等长的明文消息$\mathcal{M}_0$，$\mathcal{M}_1$给挑战者，挑战者随机选择$\beta\in\{0,1\}$，并使用旧的访问结构$(\boldsymbol{M}^*,\rho^*)$对明文消息$\mathcal{M}_\beta$运行$Encrypt$算法进行加密，得到加密后的密文$CT^*$.

Setup阶段重复Init阶段.

4）Guess. 敌手输出对β的猜测$\beta'\in\{0,1\}$. 如果存在任意多项式时间的攻击者攻击IND-CPA游戏的优势$\epsilon=\left|Pr\left[\beta=\beta'\right]-\frac{1}{2}\right|$是可忽略的，则本文方案应对选择明文攻击是安全的.

3 方案构造

本文方案共包含4个部分共12个算法，其中系统初始化、属性密钥生成、加密、解密等算法为常规部分；加密转换密钥生成、本地加密、外包加密等算法为外包加密部分；更新密钥生成、密文更新等算法为策略更新部分；验证转换密钥生成、验证、严格验证等算法为密文验证部分. 在外包加密部分，数据拥有者将随着访问结构大小而线性增长的指数计算需求交给外包加密服务完成. 在策略更新部分，结合外包加密算法结构，根据访问结构中出现的情况对属性做了区分，进一步降低了策略更新部分的计算量. 在密文验证部分分别提供了高效的验证算法和准确的严格验证算法，可以由用户根据自身需求选择算法. 方案具体算法构造如下：

1）系统初始化$Setup(k,U)$. 输入安全参数k和全体属性集合$U=\{1,2,\cdots,u\}$. 生成阶为素数p的乘法循环群G_1，满足双线性映射$e:G_1\times G_1\rightarrow G_2$. G_1的生成元为g. 随机选择$\alpha,a\in\mathbb{Z}_p^*$，$h_1,h_2,\cdots,h_u\in G_1$. 输出系统公共参数$PP=(G_1,G_2,g,e(g,g)^\alpha,g^a,h_1,h_2,\cdots,h_u)$，系统主密钥$MSK=(\alpha,a,g^\alpha)$.

2）属性密钥生成$KeyGen(PP,MSK,S)$. 定义S为用户属性集合，且满足$S\subseteq U$. 随机选择$t\in\mathbb{Z}_p$，对于所有属性$x\in S$，计算属性密钥$SK=\left\{K=g^\alpha g^{at},K_0=g^t,K_x=h_x^t\right\}$.

3）加密$Encrypt(PP,m,A)$. 明文消息$m\in G_2$，访问结构$A=(\boldsymbol{M},\rho)$，其中$\boldsymbol{M}$是$l\times n$的矩阵而ρ是将矩阵$\boldsymbol{M}$中的每一行映射到一个属性的映射函数. 随机选择$s\in\mathbb{Z}_p$作为秘密值，随机选择向量$\boldsymbol{v}=(s,r_2,r_3,\cdots,r_n)^{\mathrm{T}}\in\mathbb{Z}_p^n$. 计算秘密份额$\lambda_i=\boldsymbol{M}_i\cdot\boldsymbol{v}$，其中$i\in\{1,2,\cdots,l\}$，$\boldsymbol{M}_i$表

示矩阵$\boldsymbol{M}$的第i行. 计算密文$CT=\left(C=m\cdot e(g,g)^{\alpha s},C'=g^{s},\left(C_i=g^{a\lambda_i}h_{\rho(i)}^{-s}\right)_{i\in\{1,2,\cdots,l\}}\right)$.

4)解密$Decrypt(CT,SK)$. 令属性集合S对应属性密钥SK, 假设属性集合S满足访问结构. 令$I\subset\{1,2,\cdots,l\}$, 定义集合$I=\{i:\rho(i)\in S\}$. 令常数集合$\omega_i\in\mathbb{Z}_p$, $i\in I$, 当λ_i是矩阵$\boldsymbol{M}$的有效份额时, 满足$\sum_{i\in I}\omega_i\lambda_i=s$. 计算$\dfrac{e(C',K)}{\prod_{i\in I}(e(C_i,K_0)e(C',K_x))^{\omega_i}}=e(g,g)^{\alpha s}$, 得到明文$m=\dfrac{C}{e(g,g)^{\alpha s}}$.

5)加密转换密钥生成$ETKGen(PP,U)$. 随机选择$\gamma\in\mathbb{Z}_p$, 对于所有属性的$x\in U$, 计算加密转换密钥$ETK=(\gamma,ETK_x=h_x^{\gamma})$.

6)本地加密$Encrypt_{\text{Local}}(PP,m,A,ETK)$. 随机选择$s\in\mathbb{Z}_p$作为秘密值, 随机选择向量$\boldsymbol{v}=(s,r_2,r_3,\cdots,r_n)^{\mathrm{T}}\in\mathbb{Z}_p^n$. 计算秘密份额$\lambda_i=\boldsymbol{M}_i\cdot\boldsymbol{v}$, 其中$i\in\{1,2,\cdots,l\}$, $\boldsymbol{M}_i$表示矩阵$\boldsymbol{M}$的第i行. 计算部分密文$CT_{\text{part}}=(C=m\times e(g,g)^{\alpha s},C'=g^{s})$. 随机选择$d\in\mathbb{Z}_p$, 计算外包参数$OP=\left(-s-\gamma,\left(\lambda_i-d,g^{ad}ETK_{\rho(i)}\right)_{i\in\{1,2,\cdots,l\}}\right)=\left(-s-\gamma,\left(\lambda_i-d,g^{ad}h_{\rho(i)}^{\gamma}\right)_{i\in\{1,2,\cdots,l\}}\right)$.

7)外包加密$Encrypt_{\text{Outsource}}(PP,OP)$. 根据公共参数$PP$和外包参数$OP$, 对于所有$i\in\{1,2,\cdots,l\}$, 计算$C_i=(g^{a})^{\lambda_i-d}\times g^{ad}h_{\rho(i)}^{\gamma}\times h_{\rho(i)}^{-s-\gamma}=g^{a\lambda_i}h_{\rho(i)}^{-s}$, 得到密文组件$CT_{\text{out}}=(C_i)_{i\in\{1,2,\cdots,l\}}$.

8)更新密钥生成$UpdateKeyGen(PP,A,A',ETK)$. 旧的访问结构为$A=(\boldsymbol{M},\rho)$, 更新为新的访问结构$A'=(\boldsymbol{M}',\rho')$, 其中$\boldsymbol{M}'$为新的$l'\times n'$份额生成矩阵. 旧的随机向量为$\boldsymbol{v}=(s,r_2,r_3,\cdots,r_n)^{\mathrm{T}}\in\mathbb{Z}_p^n$, 随机选择新向量$\boldsymbol{v}'=(s,r_2',r_3',\cdots,r_n')^{\mathrm{T}}\in\mathbb{Z}_p^n$, 其中第一项仍为旧的秘密值$s$. 计算新的秘密份额$\lambda_j'=\boldsymbol{M}_j'\cdot\boldsymbol{v}'$, 其中$j\in\{1,2,\cdots,l'\}$, $\boldsymbol{M}_j'$表示矩阵$\boldsymbol{M}'$的第j行. 对于$j\in\{1,2,\cdots,l'\}$, 计算更新密钥UK. 若属性$\rho'(j)$在旧的访问结构A中出现过(类型1), 则计算$UK_{j,i}=\lambda_j'-\lambda_i$; 若属性$\rho'(j)$在旧的访问结构$A$中未出现过(类型2), 则计算$UK_j=\left(UK_j^{(1)}=\lambda_j'-d,UK_j^{(2)}=-s-\gamma,UK_j^{(3)}=g^{ad}h_{\rho(j)}^{\gamma}\right)$.

9)密文更新$CTUpdate(PP,A,A',UK,CT)$. 对于$j\in\{1,2,\cdots,l'\}$, 计算新密文CT'. 若属性$\rho'(j)$在旧的访问结构A中出现过(类型1), 计算$C_j'=C_i\cdot(g^{a})^{UK_{j,i}}=g^{a\lambda_j'}h_{\rho(j)}^{-s}$; 若属性$\rho'(j)$在旧的访问结构$A$中未出现过(类型2), 计算$C_j'=(g^{a})^{UK_j^{(1)}}\times UK_j^{(3)}\times h_{\rho(j)}^{UK_j^{(2)}}=g^{a\lambda_j'}\times h_{\rho(j)}^{-s}$. 最终, 新密文$CT'$形式为$CT'=\left(C_1=m\times e(g,g)^{\alpha s},C_2=g^{s},\left(C_j'=g^{a\lambda_j'}\times h_{\rho(j)}^{-s}\right)_{i\in\{1,2,\cdots,l'\}}\right)$.

外包加密服务器处于不可信的云环境中, 可能存在“懒惰”的情况, 即外包加密服务器可能不会严格执行算法, 只执行部分计算或者故意返回错误的结果. 外包加密服务器也可能由于程序漏洞、遭受网络入侵等原因, 没有计算出正确的结果. 如果数据拥有者无法对外包加密密文进行正确性检测, 会导致错误及不安全的数据共享. 因此, 本文提出了2个验证算法, 使数据拥有者有验证外包加密计算正确性的能力. 验证算法$Verify$为高效验证算法, 能够以极少的计算代价, 确保外包加密服务器的计算没有故意出现错误. 如果外包加密服务器故意制造错误结果, 要使故意制造的错误结果能够通过$Verify$算法的验证, 其付出的计算代价和遵循算法得到正确密文结果的计算代价是一样的, 即云服务器不能以较少的代价制造满足要求的错误结果, 进而防止服务器倾向于“偷懒”. 而针对云服务器的恶意出错问题, 严格验证算法$Verify_{\text{strict}}$则能够完全检测出来.

10)验证转换密钥生成$VTKGen(PP,U)$. 选择安全参数θ, 对于所有的$x\in U$, 选择随机数$b_x\in\{0,1\}^{\theta}$, 计算验证转换密钥$VTK=\left(b_x,VTK_x=h_x^{b_x}\right)$.

11)验证$Verify(PP,CT)$. 根据公共参数PP和密文CT, 计算验证信息$\mathcal{P}=e\left(\prod_{i\in\{1,2,\cdots,l\}}C_i,g\right)e\left(C',\prod_{i\in\{1,2,\cdots,l\}}h_{\rho(i)}\right)$, 若$e(g^{a},g)^{\sum_{i\in\{1,2,\cdots,l\}}\lambda_i}=\mathcal{P}$, 则表示验证通过, 输出1, 否则输出0.

12)严格验证$Verify_{\text{strict}}(PP,CT,VTK)$. 验证算法$Verify$无法检验密文组件$C_i$顺序上的颠倒错误(如: 云服务器故意以$C_2,C_1,\cdots$的顺序返回计算结果)、$C_i$的生成因子$g^{a\lambda_i}$和$h_{\rho(i)}^{-s}$的交换错位(如: 云服务器故意返回$C_1=g^{a\lambda_1}h_{\rho(2)}^{-s}$, 其中$i$为不同值), 所以需要引入验证转换密钥$VTK$和严格验证算法进行验证, 需要更高的计算量. 根据公共参数PP、密文CT和转换密钥VTK, 计算验证信息$\mathcal{P}=e\left(\prod_{i\in\{1,2,\cdots,l\}}C_i,g\right)e\left(C',\prod_{i\in\{1,2,\cdots,l\}}VTK_{\rho(i)}\right)=e(g,g)^{a\sum_{i\in\{1,2,\cdots,l\}}\lambda_i b_{\rho(i)}}$, 若$e(g^{a},g)^{\sum_{i\in\{1,2,\cdots,l\}}\lambda_i b_{\rho(i)}}=\mathcal{P}$, 则表示验证通过, 输出1, 否则输出0.

4 方案分析

4.1 正确性分析

数据拥有者能够利用加密过程中的已知数据$\boldsymbol{\lambda}_i$和转换密钥VTK, 通过验证算法$Verify(PP,CT)$和严格验证算法$Verify_{\text{strict}}(PP,CT,VTK)$检测密文的正确性, 具体推导过程有:

1)验证算法$Verify(PP,CT)$

因为

$$\mathcal{P}=e\left(\prod_{i\in\{1,2,\cdots,l\}}C_i,g\right)e\left(C',\prod_{i\in\{1,2,\cdots,l\}}h_{\rho(i)}\right)=$$
$$e\left(\prod_{i\in\{1,2,\cdots,l\}}g^{a\lambda_i}h_{\rho(i)}^{-s},g\right)e\left(g^s,\prod_{i\in\{1,2,\cdots,l\}}h_{\rho(i)}\right)=$$
$$e\left(\prod_{i\in\{1,2,\cdots,l\}}g^{a\lambda_i}\prod_{i\in\{1,2,\cdots,l\}}h_{\rho(i)}^{-s},g\right)e\left(g^s,\prod_{i\in\{1,2,\cdots,l\}}h_{\rho(i)}\right)=$$
$$e\left(g^{a\sum\limits_{i\in\{1,2,\cdots,l\}}\lambda_i}\left(\prod_{i\in\{1,2,\cdots,l\}}h_{\rho(i)}\right)^{-s},g\right)e\left(g^s,\prod_{i\in\{1,2,\cdots,l\}}h_{\rho(i)}\right)=$$
$$e\left(g^{a\sum\limits_{i\in\{1,2,\cdots,l\}}\lambda_i},g\right)=e(g,g)^{a\sum\limits_{i\in\{1,2,\cdots,l\}}\lambda_i}.$$

由双线性对的性质可知：

$e(g^a,g)^{\sum\limits_{i\in\{1,2,\cdots,l\}}\lambda_i}=e(g,g)^{a\sum\limits_{i\in\{1,2,\cdots,l\}}\lambda_i}$，所以

$e(g^a,g)^{\sum\limits_{i\in\{1,2,\cdots,l\}}\lambda_i}=\mathcal{P}$成立.

从上述推导过程可以看出，本文的高效验证算法$Verify(PP,CT)$不需要转换密钥VTK，仅需要输入已知数据λ_i. 该算法使用与解密算法$Decrypt(CT,SK)$中类似的计算结构进行构造，并用g和$h_{\rho(i)}$替换了解密算法中属性密钥SK的K_0,K_x参数，使得数据拥有者在无需恢复秘密值s的前提下，仅利用$\sum\limits_{i\in\{1,2,\cdots,l\}}\lambda_i$即可完成密文的正确性验证，较已有方案更高效.

2) 严格验证算法$Verify_{\text{strict}}(PP,CT,VTK)$

因为

$$\mathcal{P}=e\left(\prod_{i\in\{1,2,\cdots,l\}}C_i^{b_x},g\right)e\left(C',\prod_{i\in\{1,2,\cdots,l\}}VTK_{\rho(i)}\right)=$$
$$e\left(\prod_{i\in\{1,2,\cdots,l\}}\left(g^{a\lambda_i}h_{\rho(i)}^{-s}\right)^{b_{\rho(i)}},g\right)e\left(g^s,\prod_{i\in\{1,2,\cdots,l\}}h_{\rho(i)}^{b_{\rho(i)}}\right)=$$
$$e\left(\prod_{i\in\{1,2,\cdots,l\}}\left(g^{a\lambda_i}\right)^{b_{\rho(i)}}\prod_{i\in\{1,2,\cdots,l\}}\left(h_{\rho(i)}^{-s}\right)^{b_{\rho(i)}},g\right)e\left(g^s,\right.$$
$$\left.\prod_{i\in\{1,2,\cdots,l\}}h_{\rho(i)}^{b_{\rho(i)}}\right)=e\left(\prod_{i\in\{1,2,\cdots,l\}}g^{a\lambda_i b_{\rho(i)}}\prod_{i\in\{1,2,\cdots,l\}}\left(h_{\rho(i)}^{b_{\rho(i)}}\right)^{-s},g\right)\times$$
$$e\left(g^s,\prod_{i\in\{1,2,\cdots,l\}}h_{\rho(i)}^{b_{\rho(i)}}\right)=$$
$$e\left(\prod_{i\in\{1,2,\cdots,l\}}g^{a\lambda_i b_{\rho(i)}}\left(\prod_{i\in\{1,2,\cdots,l\}}h_{\rho(i)}^{b_{\rho(i)}}\right)^{-s},g\right)\times$$
$$e\left(g^s,\prod_{i\in\{1,2,\cdots,l\}}h_{\rho(i)}^{b_{\rho(i)}}\right)=e\left(\prod_{i\in\{1,2,\cdots,l\}}g^{a\lambda_i b_{\rho(i)}},g\right)=$$
$$e\left(g^{a\sum\limits_{i\in\{1,2,\cdots,l\}}\lambda_i b_{\rho(i)}},g\right)=e(g,g)^{a\sum\limits_{i\in\{1,2,\cdots,l\}}\lambda_i b_{\rho(i)}}.$$

由双线性对的性质可知：$e(g^a,g)^{\sum\limits_{i\in\{1,2,\cdots,l\}}\lambda_i b_{\rho(i)}}=e(g,g)^{a\sum\limits_{i\in\{1,2,\cdots,l\}}\lambda_i b_{\rho(i)}}$，所以$e(g^a,g)^{\sum\limits_{i\in\{1,2,\cdots,l\}}\lambda_i b_{\rho(i)}}=\mathcal{P}$成立.

本文的严格验证算法$Verify_{\text{strict}}(PP,CT,VTK)$与高效验证算法$Verify(PP,CT)$在输入参数上的区别在于使用了额外的转换密钥$VTK$参数. 严格验证算法的计算结构与高效验证算法相同，同样参考了解密算法$Decrypt(CT,SK)$中的部分运算. 但是严格验证算法将C_i替换为$C_i^{b_x}$，将$h_{\rho(i)}$替换为$VTK_{\rho(i)}$，实际上是引入了随机数b_x. 由于随机数b_x作用于C_i，因此严格验证算法具备发现C_i顺序错乱的能力. 由于随机数b_x作用于$h_{\rho(i)}$，因此严格验证算法还具备发现$g^{a\lambda_i}$结构和$h_{\rho(i)}^{-s}$结构不匹配的能力. 以上2种错误都是在验证$\sum\limits_{i\in\{1,2,\cdots,l\}}\lambda_i b_{\rho(i)}$时被发现的，而高效验证算法无法检测到. 当然，引入$C_i^{b_x}$也使得严格验证算法需要进行复杂度$O(n)$的指数运算，在效率上低于高效验证算法.

4.2 安全证明

定理 1. 若q-BDHE假设成立，不存在多项式时间的敌手可以选择访问结构$(\boldsymbol{M}^*,\rho^*)$，在安全游戏中对支持策略更新的属性基外包加密方案存在不可忽略的优势ϵ，那么该方案是IND-CPA安全的.

证明.

1) Init. 挑战者生成挑战向量$\boldsymbol{y}=(g,g^s,g^a,\cdots,g^{a^q},g^{a^{q+2}},\cdots,g^{a^{2q}})$，选择$T\in G_T$组成挑战结构$(\boldsymbol{y},T)$. 敌手发送要挑战的访问结构$(\boldsymbol{M}^*,\rho^*)$，其中矩阵$\boldsymbol{M}^*$大小为$l^*\times n^*$，且$n^*\leqslant q$.

2) Setup. 挑战者随机选择$\alpha'\in\mathbb{Z}_p$，并且设置$\alpha=\alpha'+a^{q+1}$，使得$e(g,g)^{\alpha}=e\left(g^a,g^{a^q}\right)\cdot e(g,g)^{\alpha'}$. 对于每个属性$x\in\{1,2,\cdots,U\}$，随机选择$z_x\in\mathbb{Z}_p$. 如果$i$满足$\rho^*(i)=x$，即$x$出现在访问结构$(\boldsymbol{M}^*,\rho^*)$中，那么令$h_x=g^{z_x}g^{a\boldsymbol{M}_{i,1}^*}g^{a^2\boldsymbol{M}_{i,2}^*},\cdots,g^{a^{n^*}\boldsymbol{M}_{i,n^*}^*}$，否则令$h_x=g^{z_x}$. 由于$h_x$中存在$g^{z_x}$项，所以$h_x$参数是随机分布的，且由于$\rho^*$是一个单射函数，每个$i$只有1个对应的$x$，所以$h_x$参数的值是明确的. 在此阶段，挑战者可以对挑战访问结构的每个属性x设置对应的h_x参数.

在Init阶段中，挑战者需要响应敌手的私钥查询请求. 假设挑战者收到的私钥请求对应的属性集合是S，且集合S不满足挑战访问结构$(\boldsymbol{M}^*,\rho^*)$. 挑战者首先随机选择$r\in\mathbb{Z}_p$. 找到一个向量$\boldsymbol{w}=(w_1,w_2,\cdots,w_{n^*})\in\mathbb{Z}_p^{n^*}$，满足$w_1=-1$且对于所有的$\rho^*(i)\in S$，满足$\boldsymbol{w}\cdot\boldsymbol{M}_i^*=0$. 根据线性秘密共享方案的定义，由于集合$S$不满足访问结构$(\boldsymbol{M}^*,\rho^*)$，这样的向量$\boldsymbol{w}$一定存在. 挑战者设置$t=r+w_1a^q+w_2a^{q-1}+\cdots+w_{n^*}a^{q-n^*+1}$，得$K_0=g^t=g^r\prod\limits_{i=1,2,\cdots,n^*}\times$

$(g^{a^{q+1-i}})^{w_i}$. 计算 $K=g^{\alpha}g^{at}=g^{\alpha'+a^{q+1}}g^{ar}\prod_{i=1,2,\cdots,n^*}(g^{a^{q+2-i}})^{w_i}$, 其中连乘的第 1 项与 $g^{a^{q+1}}$ 相抵消, 得到 $K=g^{\alpha'}g^{ar}\prod_{i=1,2,\cdots,n^*}(g^{a^{q+2-i}})^{w_i}$. 下来对任意 $x\in S$, 计算 K_x. 若 x 出现在挑战访问结构 $(\boldsymbol{M}^*,\rho^*)$ 中, 考虑 Setup 阶段设置的 h_x 参数的值, 计算 $K_x=K_0^{z_x}\prod_{j=1,2,\cdots,n^*}\left(g^{a^j r}\prod_{\substack{k=1,2,\cdots,n^*\\k\neq j}}(g^{a^{q+1+j-k}})^{w_k}\right)^{M^*_{i,j}}$; 否则设 $K_x=K_0^{z_x}$.

3) Challenge. 敌手选择 2 个等长的明文消息 $\mathcal{M}_0$, $\mathcal{M}_1$ 发送给挑战者. 挑战者随机选择 $\beta\in\{0,1\}$. 计算密文 $C=\mathcal{M}_\beta T\times e\left(g^s,g^{\alpha'}\right)$, $C'=g^s$, 得到 $CT_{\text{part}}=(C,C')$. 挑战者考虑访问结构 $(\boldsymbol{M}^*,\rho^*)$, 随机选择 $y_2^*,y_3^*,\cdots,y_{n^*}^*\in\mathbb{Z}_p$, 构造共享秘密值为 s 的向量 $\boldsymbol{v}^*=\left(s,sa+y_2^*,sa^2+y_3^*,\cdots,sa^{n^*-1}+y_{n^*}^*\right)\in\mathbb{Z}_p^{n^*}$. 对于 $i=1,2,\cdots,l^*$, 计算 $C_i^*=g^{-sZ\rho^*(i)}\times\prod_{j=2,3,\cdots,n^*}(g^a)^{M^*_{i,j}y_j^*}$, 得到密文 $CT^*=(C,C',\left(C_i^*\right)_{i\in\{1,2,\cdots,l^*\}})$.

Setup 阶段重复 Init 阶段.

4) Guess. 敌手最终输出一个对 β 的猜测值 $\beta'\in\{0,1\}$. 若 $\beta=\beta'$, 则挑战者输出 $\theta=0$, 表示 $T=e(g,g)^{a^{q+1}s}$; 否则输出 $\theta=1$ 表示 T 是群 G_T 上的随机元素. 当 $T=e(g,g)^{a^{q+1}s}$ 时, 敌手的优势是 $\epsilon=Pr\left[\beta=\beta'|\theta=0\right]-\frac{1}{2}$. 当 T 是群 G_T 上的随机元素时, 消息 $\mathcal{M}_\beta$ 对于敌手被完全隐藏, 敌手的优势是 $Pr\left[\beta=\beta'|\theta=1\right]=\frac{1}{2}$. 因此, 任何多项式时间内的敌手赢得 q-BDHE 假设的 IND-CPA 游戏的优势是可以忽略的. 证毕.

定理 2. 若 q-BDHE 假设成立, 对于本文 2.2 节定义的安全游戏, 在本文方案的策略更新过程中敌手无法增加其优势.

证明. 考虑 2 个更新密钥查询请求 $UK(\mathcal{M}_0,(\boldsymbol{M}^*,\rho^*),(\boldsymbol{M}^{\#},\rho^{\#}))$ 和 $UK(\mathcal{M}_1,(\boldsymbol{M}^*,\rho^*),(\boldsymbol{M}^{\#},\rho^{\#}))$, 由于加密 $\mathcal{M}_0$ 和 $\mathcal{M}_1$ 的随机数是相同的(这是因为挑战者将随机选择其中一个明文进行加密, 只有 1 个明文消息会被挑战者选中), 挑战者向敌手返回的更新密钥完全相同, 且不包含任何挑战信息. 因此, 挑战者在选择挑战消息时, 更新密钥不会透露任何信息. 在定理 1 证明的 Challenge 阶段, 使用访问策略 $(\boldsymbol{M}^*,\rho^*)$ 加密得到的密文 $CT^*=(C,C',(C_i^*)_{i\in\{1,2,\cdots,l^*\}})$. 考虑在策略更新过程中选择新的访问策略 $(M^{\#},\rho^{\#})$, 根据更新密钥生成算法 $UpdateKeyGen(PP,A,A',ETK)$ 和密文更新算法 $CTUpdate(PP,A,A',UK,CT)$ 计算得到新密文 $CT^{\#}=(C,C',(C_k^{\#})_{k\in\{1,2,\cdots,l^{\#}\}})$, 其中 $C_k^{\#}=g^{-sZ\rho^{\#}(i)}\left(\prod_{j=2,3,\cdots,n^{\#}}(g^a)^{M^{\#}_{k,j}y_j^{\#}}\right)$, 而 $CT^{\#}$ 中的 $C=\mathcal{M}_\beta T\times e\left(g^s,g^{\alpha'}\right)$ 与 CT^* 中的 C 相同. 因此, 由定理 1 可知, 若 q-BDHE 假设成立, 敌手在区分 $CT^{\#}$ 中的 C 与群 G_T 上的随机元素时, 不具备不可忽略的优势, 并且密文验证过程不会为敌手增加任何优势. 证毕.

4.3 安全性对比

本文从策略更新能力、密文验证能力、安全模型和安全假设 4 个方面对本文方案与现有方案的安全性进行对比分析, 如表 1 所示.

Table 1 Comparison of Schemes for Security

表 1 方案安全性对比

方案	外包加密	策略更新	密文验证	安全模型	安全假设
文献 [10]	√	×	×	随机预言机	q-parallel BDHE
文献 [12]	√	×	√	随机预言机	q-parallel BDHE
文献 [8]	√	×	×	随机预言机	q-BDHE
文献 [7]	√	×	×	标准模型	RCCA-secure
文献 [13]	√	×	√		
文献 [11]	×	√	√	随机预言机	GSGDA
文献 [15]	√	×	√	随机预言机	DBDH
本文方案	√	√	√	标准模型	q-BDHE

从表 1 可以看出, 仅有文献 [11] 方案和本文方案同时具备策略更新和密文验证能力. 但文献 [11] 方案仅具备常规的密文正确性验证能力, 即仅能发现云服务器的无意错误, 而本文方案中数据拥有者不仅具备高效的密文验证能力还具备严格的错误检查能力, 既能发现云服务器在外包加密及密文更新过程中的无意错误, 还能发现其有意构造的密文错误. 此外, 本文方案的密文验证效率较文献 [11] 方案更高, 同时还支持外包加密. 在安全模型方面, 也只有本文方案和文献 [7] 方案是在标准模型下证明的.

4.4 理论性能分析

本文从功能、加密时间、密文长度和交互长度等方面与相关方案进行比较, 具体对比参数如表 2 所示. 对比方案的功能主要考虑是否支持外包加密功能、策略更新功能和密文正确性验证功能; 加密效率比较主要考虑外包加密方案中的本地加密时间(此处将验证计算时间计入本地加密时间中)、本地和外包的合计加密时间. 如果方案支持密文正确性验证, 则对比不同方案在验证密文时的计算时间; 密文长度和交互长度主要比较的是方案的通信代价, 交互长度指的是在外包加密过程中, 数据拥有者和外包服务之间的交互数据的长度.

从表 2 可以看出本文方案的本地加密计算开销为 $2T_1+lt_1$, 若不使用外包计算, 则加密开销上升为

Table 2 Comparison of Outsourced Encryption Attribute-Based Schemes

表 2 外包加密属性基方案对比

方案	本地加密时间$-T_2-t_2$	合计加密时间$-T_2-t_2$	验证计算时间	密文长度	交互长度
文献 [10]	$3T_1+lt_1$	$(2l+3)T_1+3lt_1$		$lL_1+(l+2)L_3$	$2lL_1+(l+2)L_3$
文献 [12]	$5T_1+22lt_1$	$(14l+5)T_1+22lt_1$		$(2l+1)L_1+L_m$	$28lL_1+14lL_3$
文献 [8]	$3T_1+t_1$	$(2l+4)T_1+(l+1)$		$(l+3)L_1+L_m$	$(l+1)L_1+L_3+L_A$
文献 [7]	$(4l+1)t_1$	$(10l+2)T_1+(8l+1)t_1$		$(3l+1)L_1+2lL_3$	$(3l+1)L_1+(3l+1)L_3$
文献 [13]	T_2+t_2	$(3l+1)T_1+lt_1$	$(l+1)t_1+(l+2)T_1$	$(2l+1)L_1+(2l+1)L_3$	$2lL_1+3lL_3$
文献 [11]			$l(T_1+T_2+T_3)$	$2lL_1+lL_2$	lL_1+lL_2
文献 [15]	$T_2+t_1+t_2$	$lT_1+T_2+2t_1+t_2$	lT_1+2lT_3	$(l+2)L_1+L_2$	$l(L_1+L_3)$
本文	$2T_1+lt_1$	$(2l+2)T_1+3lt_1$	$2lt_1+t_2+T_2$	$(l+1)L_1$	$2lL_1+(l+1)L_3$

注：其中T_1,T_2分别表示群G_1,G_2指数运算时间；T_3表示双线性对运算时间；t_1, t_2分别表示群G_1,G_2乘法运算时间；l表示线性秘密共享访问结构中矩阵$\boldsymbol{M}$的行数；L_1,L_2,L_3分别表示群$G_1,G_2,\mathbb{Z}_p$上的元素长度；L_m表示明文长度；L_A表示访问结构长度.

$(2l+1)T_1+lt_1$. 可见通过外包计算，数据拥有者不需要承担随着访问结构线性增长的指数运算，缓解了本地终端的计算压力. 本文方案的合计加密时间为$(2l+2)T_1+3lt_1$，略大于不使用外包加密计算的本地开销，但额外开销基本只有乘法运算，没有过多的冗余计算量，对整体效率影响不大. 现有的支持外包加密功能的属性基方案都不支持策略更新功能，本文方案采用相似的计算结构，构造了支持外包机密和策略更新功能的属性基方案. 对于密文验证功能，现有的部分属性基方案也没有很好的实现方式，文献 [7] 方案不支持密文验证功能，文献 [8] 和文献 [10] 方案仅支持数据使用者在解密时验证密文，对于数据拥有者来说，无法得到密文正确性的信息. 文献 [12] 方案支持在外包加密过程中进行验证，外包过程和验证过程不可拆分，导致引入了过多的冗余计算量. 文献 [13] 方案引入指数打包验证算法，能够实现特定结构密文的正确性验证. 本文方案的本地加密时间相较于文献 [7–8, 10, 12] 方案均有所缩短. 文献 [13] 方案通过构造更长的密文结构，将部分加密阶段的计算开销转移到解密阶段，使得本地加密时间大大缩短. 对于密文验证计算时间，本文方案中的验证算法效率高于文献 [13] 方案，严格验证算法与文献 [13] 方案效率相当，且高于文献 [11, 15] 方案. 在通信开销方面，本文方案的密文长度最短，交互长度仅长于文献 [8] 方案. 文献 [8, 13] 方案虽然具有本地加密效率高、验证算法效率较高的优点，但是没有解决外包服务器和数据使用者的合谋攻击问题，不具备抗合谋攻击的能力，而本文方案不仅具备抗合谋攻击的能力，并且支持高效的密文正确性验证方式.

4.5 实验分析

本文通过仿真实验来分析对比属性基方案的实际性能表现，使用 Charm 密码学框架对本文方案和其他相关方案进行仿真实现. 实验中使用 512 b 椭圆曲线，运行环境为 Ubuntu 16.04 和 Python 3.5. 在实验中分别实现了不同方案的本地加密部分、外包加密部分、验证计算部分和策略更新部分，记录不同属性个数情况下各方案的 CPU 运行时间，具体结果如图 1 所示.

由图 1(a)可以看出，本文方案与相关方案在属性个数较少时，本地加密时间的差异并不明显，当属性个数增加时，本文方案的本地加密时间相比文献 [7] 和文献 [10] 方案分别降低 35.5% 和 22.9%. 由图 1(b)可以看出，外包加密承担的计算量通常是本地加密的 10 倍以上，而本文方案也很好地控制了外包计算量，分别为文献 [7] 和文献 [10] 方案的 19.3% 和 65.2%，提高了系统的整体效率. 在验证计算方面，本文方案中的严格验证算法计算时间降低为文献 [11] 方案的 22.3% 和文献 [15] 方案的 18.9%. 文献 [13] 方案将指数打包验证用于密文验证，其验证算法的计算量与本文方案中的严格验证算法计算量相当，而本文提出的高效验证算法的效率为文献 [13] 方案的 6 倍，大大降低了密文验证所需要的计算开销. 在策略更新方面，由于本文方案将策略更新与外包加密相结合，使得数据拥有者在策略更新阶段的计算开销降低至本地加密水平. 由图 1(d)可以看出，同为类型 3 的属性策略更新，本文方案的计算时间仅为文献 [11] 方案的 3.5%，在策略更新阶段的大部分计算量都由负责密文更新的外包服务器承担.

5 结 论

综上所述，本文方案在保证安全性的前提下，提

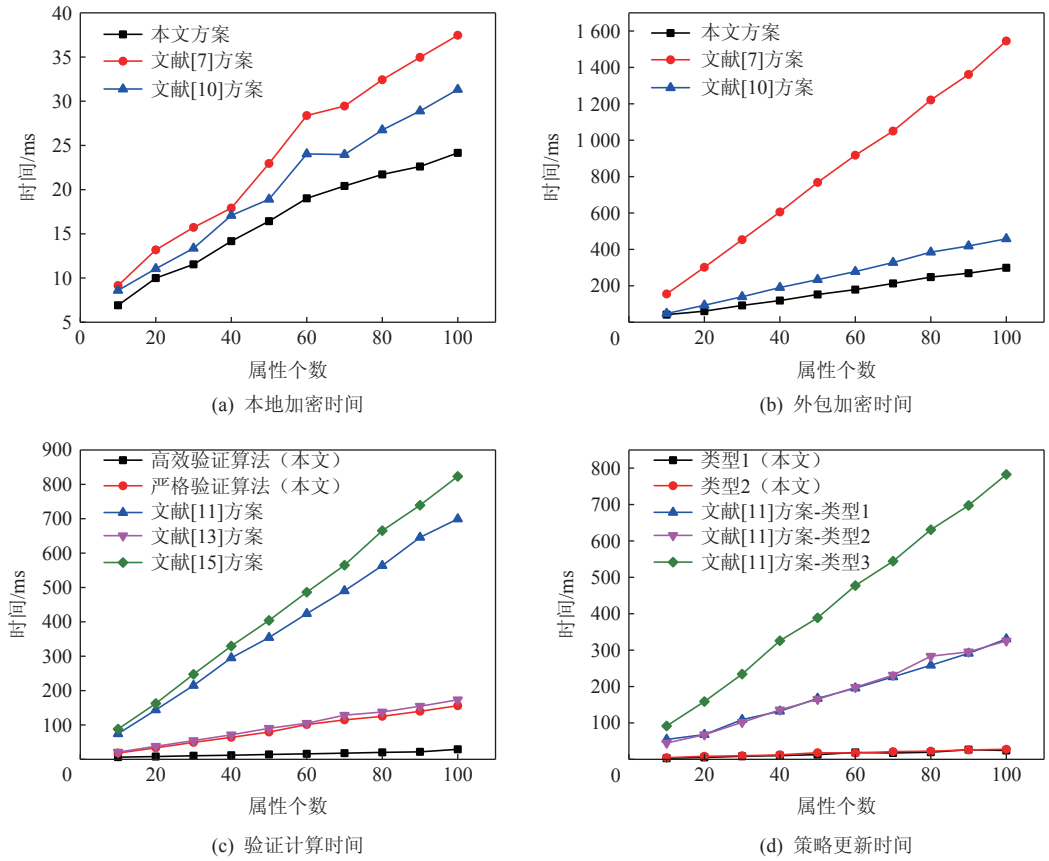

Fig. 1 Experimental results comparison

图 1 实验结果对比

出一种支持策略更新的属性基外包加密方案，还提出了 2 种用于验证密文正确性的方式，有效避免云计算环境下外包计算过程中的安全隐患. 性能分析表明，本文方案在密文长度、策略更新计算量、密文验证计算量上具有一定优势.

作者贡献声明：苏泽林提出了算法思路和实验方案，完成实验并撰写论文；张文芳和王小敏提出指导意见并修改论文.

参 考 文 献

[1] Sahai A, Waters B. Fuzzy identity-based encryption[C]//Proc of the 24th Annual Int Conf on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 2005: 457−473

[2] Goyal V, Pandey O, Sahai A, et al. Attribute-based encryption for fine grained access control of encrypted data[C]//Proc of the 13th ACM Conf on Computer and Communications Security. New York: ACM, 2006: 89–98

[3] Bethencourt J, Sahai A, Waters B. Ciphertext-policy attribute-based encryption[C]//Proc of the 23rd IEEE Symp on Security and Privacy. Piscataway, NJ: IEEE, 2008: 321–334

[4] Waters B. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization[C]//Proc of the 14th Int Workshop on Public Key Cryptography. Berlin: Springer, 2008: 53−70

[5] Green M, Hohenberger S, Waters B. Outsourcing the decryption of ABE ciphertexts[C]//Proc of the 20th USENIX Conf on Security. Berkeley, CA: USENIX Association, 2011: 34−49

[6] Li Jingwei, Jia Chunfu, Li Jin, et al. Outsourcing encryption of attribute-based encryption with MapReduce[C]//Proc of the 24th Int Conf on Information and Communications Security. Berlin: Springer, 2012: 191−201

[7] Zhang Rui, Ma Hui, Lu Yao. Fine-grained access control system based on fully outsourced attribute-based encryption[J]. Journal of Systems & Software, 2017, 125: 344−353

[8] Zhao Zhiyuan, Wang Jianhua, Xu Kaiyong, et al. Fully outsourced attribute-based encryption with verifiability for cloud storage[J]. Journal of Computer Research and Development, 2019, 56(2): 218−228 (in Chinese)
(赵志远，王建华，徐开勇，等. 面向云存储的支持完全外包属性基加密方案[J]. 计算机研究与发展，2019，56(2)：218−228)

[9] Li Jing, Li Xiong, Wang Licheng, et al. Fuzzy encryption in cloud computation: efficient verifiable outsourced attribute-based encryption[J]. Soft Computing, 2018, 22(3): 707−714

[10] Chen Hongjie, Liao Yongjian. Improvement of an outsourced attribute-based encryption scheme[J]. Soft Computing, 2019, 23(22):

11409−11417

[11] Yang Kun, Jia Xiaohua, Ren Kui. Secure and verifiable policy update outsourcing for big data access control in the cloud[J]. IEEE Transactions on Parallel and Distributed Systems, 2014, 26(12): 3461−3470

[12] Yan Xixi, He Guanghui, Yu Jinxia. Secure and verifiable outsourced ciphertext policy attribute base encryption[J]. Journal of Cryptologic Research, 2020, 7(5): 628−642 (in Chinese)
(闫玺玺, 何广辉, 于金霞. 可验证的密文策略属性基加密安全外包方案[J]. 密码学报, 2020, 7(5): 628−642)

[13] Fan Kai, Wang Junyong, Wang Xin, et al. A secure and verifiable outsourced access control scheme in fog-cloud computing[J]. Sensors, 2017, 17(7): 1695

[14] Wang Hao, He Debiao, Shen Jian, et al. Verifiable outsourced ciphertext-policy attribute-based encryption in cloud computing[J]. Soft Computing, 2017, 21(24): 7325−7335

[15] Premkamal P K, Pasupuleti S K, Alphonse P J A. A new verifiable outsourced ciphertext-policy attribute based encryption for big data privacy and access control in cloud[J]. Journal of Ambient Intelligence and Humanized Computing, 2019, 10(7): 2693−2707

[16] Wang Shulan, Wang Haiyan, Li Jianqiang, et al. A fast CP-ABE system for cyber-physical security and privacy in mobile healthcare network[J]. IEEE Transactions on Industry Applications, 2020, 56(4): 4467−4477

[17] Li Xiong, Liu Tian, Chen Chaoyang, et al. A lightweight and verifiable access control scheme with constant size ciphertext in edge computing assisted IoT[J]. IEEE Internet of Things Journal, 2022, 9(19): 19227−19237

[18] Hahn C, Kim J. Verifiable outsourced decryption of encrypted data from heterogeneous trust networks[J]. IEEE Internet of Things Journal, 2022, 9(22): 22559−22570

[19] Ying Zuobin, Ma Jianfeng, Cui Jiangtao. Partially policy hidden CP-ABE supporting dynamic policy updating[J]. Journal on Communications, 2015, 36(12): 178−189 (in Chinese)
(应作斌, 马建峰, 崔江涛. 支持动态策略更新的半策略隐藏属性加密方案[J]. 通信学报, 2015, 36(12): 178−189)

[20] Ying Zuobin, Li Hui, Ma Jianfeng, et al. Adaptively secure ciphertext-policy attribute-based encryption with dynamic policy updating[J]. Science China Information Sciences, 2016, 59(4): 1−16

[21] Sethi K, Pradhan A, Bera P. Practical traceable multi-authority CP-ABE with outsourcing decryption and access policy updation[J]. Journal of Information Security and Applications, 2020, 51: 102435

[22] Li Jianqiang, Wang Shulan, Li Yuan, et al. An efficient attribute-based encryption scheme with policy update and file update in cloud computing[J]. IEEE Transactions on Industrial Informatics, 2019, 15(12): 6500−6509

[23] Yan Xixi, Liu Yuan, Li Zichen, et al. Multi-authority attribute-based encryption scheme with policy dynamic updating[J]. Journal on Communications, 2017, 38(10): 94−101(in Chinese)
(闫玺玺, 刘媛, 李子臣, 等. 支持策略动态更新的多机构属性基加密方案[J]. 通信学报, 2017, 38(10): 94−101)

Su Zelin, born in 1997. Master. His main research interests include cryptography and attribute-based encryption.
苏泽林, 1997 年生. 硕士. 主要研究方向为密码学、属性基加密.

Zhang Wenfang, born in 1978. PhD, professor, PhD supervisor. Her main research interests include cryptography and information security.
张文芳, 1978 年生. 博士, 教授, 博士生导师. 主要研究方向为密码学、信息安全.

Wang Xiaomin, born in 1974. PhD, professor, PhD supervisor. His main research interests include information security and rail transit safety engineering.
王小敏, 1974 年生. 博士, 教授, 博士生导师. 主要研究方向为信息安全、轨道交通安全工程.