可验证外包解密的离线/在线属性基加密方案

赵志远* 孙 磊 户家富 周时娥
(信息工程大学三院 郑州 450001)

摘 要：属性基加密可以为雾-云计算中的数据提供机密性保护和细粒度访问控制，但雾-云计算系统中的移动设备难以承担属性基加密的繁重计算负担。为解决该问题，该文提出一种可验证外包解密的离线/在线属性基加密方案。该方案能够实现离线/在线的密钥生成和数据加密，同时支持可验证外包解密。然后，给出方案的选择明文攻击的安全证明和可验证性的安全证明。之后，该文将转换阶段所需双线性对的计算量降为恒定常数。最后，从理论和实验两方面对所提方案进行性能分析，实验结果表明该方案是有效且实用的。

关键词：属性基加密；离线/在线；外包解密；可验证性
中图分类号：TP309 **文献标识码**：A **文章编号**：1009-5896(2018)12-2998-09
DOI: 10.11999/JEIT180122

Efficient Offline/Online Attribute Based Encryption with Verifiable Outsourced Decryption

ZHAO Zhiyuan SUN Lei HU Jiafu ZHOU Shie
(*The Third Institute, Information Engineering University, Zhengzhou* 450001, *China*)

Abstract: Attribute based encryption can provide data confidentiality protection and fine-grained access control for fog-cloud computing, however mobile devices in fog cloud computing system are difficult to bear the burdensome computing burden of attribute based encryption. In order to address this problem, an offline/online ciphertext-plicy attribute-based encryption scheme is presented with verifiable outsourced decryption based on the bilinear group of prime order. It can realize the offline/online key generation and data encryption. Simultaneously, it supports the verifiable outsourced decryption. Then, the formal security proofs of its selective chosen plaintext attack security and verifiability are provided. After that, the improved offline/online ciphertext-plicy attribute-based encryption scheme with verifiable outsourced decryption is presented, which reduces the number of bilinear pairings from linear to constant in the transformation phase. Finally, the efficiency of the proposed scheme is analyzed and verified through theoretical analysis and experimental simulation. The experimental results show that the proposed scheme is efficient and practical.

Key words: Attributed-based encryption; Offline/online; Outsourced decryption; Verifiability

1 引言

雾-云计算模式[1]有效解决了物联网中数据传输与存储问题。但是，数据安全和隐私保护仍是雾-云计算中的一个重要挑战。而属性基加密(Attribute-Based Encryption, ABE)是一种新颖的公钥加密原语，其能够为雾-云计算环境中的数据提供细粒度访问控制。Sahai和Waters[2]首先提出属性基加密方案。依据访问策略的位置，属性基加密有2种变体：密钥策略属性基加密(Key-Policy ABE, KP-ABE)[3]，其访问策略关联解密密钥；密文策略属性基加密(Ciphertext-Policy ABE, CP-ABE)[4]，其访问策略关联密文。自上述方案提出以来，相关科研人员进行了大量的研究工作。然而，CP-ABE计算过程所需的大量计算资源将给数据拥有者和用户带来严重的计算负担。因此，需要采用一些先进的技术来帮助物联网用户承担大部分属性基加密的计算任务。通常有2种方法来解决该问题。

第1种方法是将复杂的计算外包给第3方。有关学者提出了许多有价值的方法。Green等人[5]提出一种解密运算外包方案，该方案在解密过程中首先将密文传送给解密外包服务器，解密外包服务器对

收稿日期：2018-01-29；改回日期：2018-06-11；网络出版：2018-08-30
*通信作者： 赵志远 zzy_taurus@foxmail.com
基金项目：国家973计划(2013CB338000)，国家重点研发计划项目(2016YFB0501900)
Foundation Items: The National 973 Program of China (2013CB338000), The National Key Research Program of China (2016YFB0501900)

密文进行一次密文转换获得中间密文再传送给用户，达到降低本地解密计算量的目的。Lai等人[6]实现外包解密的同时，支持了对外包计算的正确性验证。Zhao等人[7]提出了面向移动云计算的外包ABE方案。该方案通过对加密过程中的随机盲化因子s进行秘密共享，构建混合访问策略，完成加密外包，同时该方案支持解密外包。Fan等人[8]提出了一种多授权中心的可验外包方案，该方案将大部分加密和解密计算外包给雾设备，并且可以验证计算结果的正确性。Li等人[9]提出一种新奇的可验证外包解密计算，并且密文长度不随着访问结构复杂度变化，但是该方案只实现了解密外包，而数据拥有者仍然需要大量计算完成数据加密任务。Zhang等人[10]提出一种完全外包ABE方案，即将密钥生成、加密和解密都外包给云服务商，并且完成了方案的安全性证明。但该方案没有提出如何验证中间密文的正确性，即无法完成外包计算正确性的验证，而可验证性对于正确计算至关重要。

第2种方法是采用离线/在线技术。即，离线阶段预处理大量繁重的工作，在线阶段可以快速响应密钥请求或加密任务。Even等人[11]第1次提出离线/在线的数字签名技术。Liu等人[12]在无线传感器网络环境中提出了一种基于身份的离线/在线签名方案。Guo等人[13]提出一种基于身份的离线/在线加密方案。离线阶段预处理大部分计算工作，在线阶段完成实际加密操作。文献[14]进一步提高了方案的效率，同时只需更短的密文。Chow等人[15]提出一种基于身份的离线/在线密钥封装机制，该方案将密钥封装过程分为离线阶段和在线阶段。基于文献[16]，Hohenberger等人[17]提出一种离线/在线ABE方案，该方案支持离线/在线加密和密钥生成。Liu等人[18]通过结合离线/在线技术和验证外包技术提出一种新的密文策略属性基加密方案。

上述方案中，大部分方案不能同时完成密钥生成、加密和解密的计算量预处理或外包工作。为解决上述问题，本文提出了一种可验证外包解密的离线/在线属性基加密方案(Offline/Online CP-ABE with Verifiable Outsourced Decryption, 3OVD-CP-ABE)。该方案采用离线/在线技术和验证外包技术相结合的方法，实现离线/在线密钥生成、离线/在线加密、解密外包及可验证性。然后，本文对方案进行了选择明文攻击的安全性证明和可验证性的安全证明。之后，本文基于3OVD-CP-ABE设计出I3OVD-CP-ABE (Improved I3OVD-CP-ABE)方案，该方案在密文转换阶段，将转换所需双线性对的计算数量降为恒定常数。最后，从理论和实验两方面对本文方案进行性能分析，实验结果表明所提方案是有效且实用的。

2　系统及安全模型

2.1　系统模型

本文提出方案系统模型中包括属性授权机构、云服务商、数据拥有者和数据用户4个实体。该方案包括以下8个算法：

Setup(ς, U)：属性授权机构运行该算法。其以隐含安全参数ς和属性集合U作为输入，输出系统公钥PK和系统主私钥MSK。

Offline.Encrypt(PK)：数据拥有者离线阶段运行该算法。其以公钥PK作为输入，输出中间密文IT。

Online.Encrypt(PK, IT, m, $(\boldsymbol{M}, \rho)$)：数据拥有者在线阶段运行该算法。该算法以公钥PK、中间密文IT、明文$m \in G_T$和访问策略$(\boldsymbol{M}, \rho)$作为输入，输出密文CT和验证标志VM_m。

Offline.KeyGen(MSK)：属性授权机构离线运行该算法。其以主私钥MSK作为输入，输出中间密钥IK。

Online.KeyGen(PK, IK, S)：属性授权机构在线运行该算法。其以公钥PK、中间密钥IK和属性集合S作为输入，输出私钥SK。

KeyGen.out(PK, SK)：属性授权机构运行该算法。其以公钥PK和私钥SK作为输入，输出转换密钥TK和取回密钥RK。

Transform(PK, CT, TK)：云服务商运行该算法。其以公钥PK、转换密钥TK和密文CT作为输入，输出转换密文CT′。

Decrypt(PK,CT′,RK,VK_m)：数据用户运行该算法。其以公钥PK、转换密文CT′、取回密钥RK和验证标志VM_m作为输入，输出明文消息m，或返回终止符$\perp$。

2.2　安全模型

本文假设未授权的用户能够访问云中资源。数据用户是不诚实的，并且用户之间允许进行合谋解密密文。假设对称加密算法和哈希函数是安全的。通过仿真者$\mathcal{B}$和敌手$\mathcal{A}$之间的博弈游戏描述所提方案的安全模型，具体过程如下：

系统初始化：$\mathcal{A}$将要挑战的访问结构$T^* = (\boldsymbol{M}^*, \rho^*)$传送给$\mathcal{B}$。

系统建立：$\mathcal{B}$执行Setup算法，然后将PK发送给$\mathcal{A}$。

查询阶段1：$\mathcal{B}$初始化空表T、空集合D和整数$j = 0$。$\mathcal{A}$可以适应性地做出下列任何查询：

(1) Create(S)：$\mathcal{B}$设置$j := j + 1$，然后运行

Online.KeyGen(PK, Offline.KeyGen(MSK), S)获得SK，运行KeyGen.out(PK, SK)获得TK和RK，最后在表T中存储元组(j, S, SK, TK, RK)。其中，属性集合S不满足访问策略T^*，即$f(T^*, S) \neq 1$。

(2) Corrupt.SK(i)：$\mathcal{B}$检查表T中是否存在第i^{th}个元组(i, S, SK)。若存在，则设置$D := D \cup \{S\}$并返回SK；若不存在，则返回终止符$\perp$。

(3) Corrupt.TK(i)：$\mathcal{B}$检查表T中是否存在第i^{th}个元组(i, S, TK)。若存在，则返回TK；否则返回终止符$\perp$。

挑战阶段：敌手$\mathcal{A}$提交2个等长的消息m_0和m_1，然后仿真者$\mathcal{B}$随机选择$b \in \{0,1\}$并运行Online.Encrypt(PK, Offline.Encrypt(PK), m_b, T^*)获得$(\text{CT}_b^*, \text{VK}_m^*)$，最后将$(\text{CT}_b^*, \text{VK}_m^*)$发送给敌手$\mathcal{A}$。

查询阶段2：类似查询阶段1，$\mathcal{A}$继续向$\mathcal{B}$提交一系列属性列表，其限制与查询阶段1相同。

猜测阶段：$\mathcal{A}$输出一个值$b' \in \{0,1\}$作为对b的猜测。如果$b' = b$，我们称$\mathcal{A}$赢得了该游戏。$\mathcal{A}$在该游戏中的优势定义为：$\text{Adv}_{\mathcal{A}} = |\Pr[b' = b] - 1/2|$。

定义 1 若无多项式时间敌手以不可忽略的优势攻破上述安全模型，那么说本文方案是选择性安全。

通过仿真者$\mathcal{B}$和敌手$\mathcal{A}$之间的博弈游戏描述所提方案的可验证性，具体过程如下：

系统建立：$\mathcal{B}$执行Setup算法，然后将PK发送给$\mathcal{A}$。

查询阶段1：$\mathcal{A}$按照上述查询阶段1方式适应性询问预言。

挑战阶段：$\mathcal{A}$提交明文m^*和挑战访问策略T^*。$\mathcal{B}$运行Online.Encrypt(PK, Offline.Encrypt(PK), m_b, T^*)获得$(\text{CT}^*, \text{VK}_m^*)$，然后将其发送给$\mathcal{A}$。

查询阶段2：$\mathcal{A}$按照上述查询阶段2方式适应性询问预言。

猜测阶段：敌手$\mathcal{A}$输出一个满足$f(T^*, S^*) = 1$的属性集合S^*和一个转换密文$\text{CT}' = (\text{CM}', C_{\text{SE}})$。若$\text{Decrypt}(\text{PK}, \text{CM}', C_{\text{SE}}, \text{RK}, \text{VK}_m^*) \notin \{m^*, \perp\}$，则敌手$\mathcal{A}$赢得了上述游戏。敌手$\mathcal{A}$在该游戏中的优势定义为：$\text{Adv}_{\mathcal{A}}^{\text{Ver}}(\lambda) := \Pr[\mathcal{A}\text{Wins}]$。

定义 2 若无多项式时间敌手以不可忽略的优势来攻破以上安全模型，那么本文方案具有可验证性。

3 3OVD-CP-ABE方案

3.1 具体方案

Setup(ς, U)：不失一般性，本文假设属性集合中的元素为Z_p中的元素。该算法通过群生成算法ψ获得元组(p, G, G_T, e)，其中G和G_T是阶为素数p的循环群，$e: G \times G \to G_T$为双线性映射。然后，该算法选择生成元$g, h, u, v, w \in G$，随机指数$\alpha \in Z_p$，2个哈希函数$H_0: G_T \to \{0,1\}^{l_{H_0}}$与$H_1: \{0,1\}^* \to \{0,1\}^{l_{H_1}}$，一个提取器$H \in \mathcal{H}(\mathcal{H}: G_T \to \{0,1\}^{l_{\text{SE}}}$是两两独立散列函数族)和一个对称加密方案SE。最后，输出系统主私钥为MSK$= (\alpha)$，系统公钥为PK$= (G, p, g, h, u, v, w, e(g,g)^{\alpha}, H_0, H_1, H, \text{SE})$。

加密阶段包括Offline.Encrypt和Online.Encrypt 2个算法，嵌入密文中的访问结构为LSSS(Linear Secret Sharing Scheme)，具体参考文献[5]。设计思路为：中间密文IT由主模块和属性模块2个逻辑类型的对象组成。离线阶段，任意数量的主模块和属性模块独立创建；在线阶段提供一个访问结构$(\boldsymbol{M}, \rho)$，然后一个主模块和l个属性模块将被选择使用，其中$\boldsymbol{M}$是一个$l \times n$矩阵。任何属性模块都可以搭配任何主模块。也就是说，本文创建了一个密文模块“池”，然后从中提取现成的密文组件。

Offline.Encrypt(PK)：该算法生成包含主模块和属性模块的中间密文IT。主模块计算过程如下：随机选择$s \in Z_p$并计算$\text{key} = e(g,g)^{\alpha s}$和$C_0 = g^s$，创建主模块为$\text{IT}_{ma} = (\text{key}, C_0, s)$；属性模块计算过程如下：随机选择$\lambda'_i, x_i, t_i \in Z_p$并计算$C_{i,1} = w^{\lambda'_i} v^{t_i}$，$C_{i,2} = (u^{x_i} h)^{-t_i}$，$C_{i,3} = g^{t_i}$，创建属性模块$\text{IT}_{at,i} = (\lambda'_i, x_i, t_i, C_{i,1}, C_{i,2}, C_{i,3})$。

Online.Encrypt(PK, IT, m, $(\boldsymbol{M}, \rho)$)：首先选择一个访问策略$(\boldsymbol{M}, \rho)$，其中$\boldsymbol{M}$是一个$l \times n$的矩阵。该算法从“池”中选择一个主模块$\text{IT}_{ma} = (\text{key}, C_0, s)$和任意$l$个属性模块$\text{IT}_{at,j} = (\lambda'_j, x_j, t_j, C_{j,1}, C_{j,2}, C_{j,3})$。然后随机选择$y_2, y_3, \cdots, y_n \in Z_p$并设置向量$\boldsymbol{y} = (s, y_2, y_3, \cdots, y_n)^{\text{T}}$，计算秘密值$s$的共享份额$\{\lambda_i = \boldsymbol{M}_i \cdot \boldsymbol{y}\}_{1 \le i \le l}$。对于$1 \le j \le l$，计算$C_{j,4} = \lambda_j - \lambda'_j$和$C_{j,5} = t_j(x_j - \rho(j))$，并设置$\text{CM} = ((\boldsymbol{M}, \rho), C_0, \{C_{j,1}, C_{j,2}, C_{j,3}, C_{j,4}, C_{j,5}\}_{j \in [1,l]})$。然后计算$\text{Tag}_0 = H_0(\text{key})$，$K_{\text{SE}} = H(\text{key})$，$C_{\text{SE}} = \text{SE.Enc}(K_{\text{SE}}, m)$，$\text{Tag} = H_1(\text{Tag}_0 || C_{\text{SE}})$。最后，输出$\text{CT} = (\text{CM}, C_{\text{SE}})$和$\text{VK}_m = \text{Tag}$。

密钥生成包括Offline.KeyGen和Online.KeyGen 2个算法。设计思路为：中间密钥IK由主模块和属性模块2个逻辑类型的对象组成。离线阶段，任意数量的主模块和属性模块独立创建；在线阶段提供一个属性集合$S = \{A_1, A_2, \cdots, A_k\} \subseteq Z_p$，然后一个主模块和$k$个属性模块将被选择使用。任何属性模块都可以搭配任何主模块。也就是说，本文创建了一个密钥模块“池”，然后从中提取现成的密钥组件。

Offline.KeyGen(MSK)：该算法生成包含主模块和属性模块的中间密钥IK。主模块计算过程如下：随机选择$r \in Z_p$并计算$K_0 = g^{\alpha} w^r$，$K_1 = g^r$，

$K_v = v^{-r}$，创建主模块$\mathrm{IK}_{ma} = (K_0, K_1, K_v)$；属性模块计算过程如下：随机选择$r_i, b_i \in Z_p$并计算$K'_{i,1} = g^{r_i}$和$K'_{i,2} = (u^{b_i}h)^{r_i}$，创建属性模块$\mathrm{IK}_{at,i} = (r_i, b_i, K'_{i,1}, K'_{i,2})$。

Online.KeyGen(PK, IK, S)：假设属性集合为$S = \{A_1, A_2, \cdots, A_k\} \subseteq Z_p$。该算法从“池”中选择一个主模块$\mathrm{IK}_{ma} = (K_0, K_1, K_v)$和任意$k$个属性模块$\mathrm{IK}_{at,i} = (r_i, b_i, K'_{i,1}, K'_{i,2})$。然后计算$K_{i,1} = K'_{i,1} = g^{r_i}$，$K_{i,2} = K'_{i,2} \cdot K_v = (u^{b_i}h)^{r_i} \cdot v^{-r}$，$K_{i,3} = r_i(A_i - b_i)$。最后，输出私钥为$\mathrm{SK} = (S, K_0, K_1, \{K_{i,1}, K_{i,2}, K_{i,3}\}_{i\in[1,k]})$。

KeyGen.out(PK, SK)：该算法以公钥PK和私钥$\mathrm{SK} = (S, K_0, K_1, \{K_{i,1}, K_{i,2}, K_{i,3}\}_{i\in[1,k]})$作为输入，然后随机选择$z \in Z_p$并计算$\bar{K}_0 = K_0^z$，$\bar{K}_1 = K_1^z$，$\bar{K}_{i,1} = K_{i,1}^z$，$\bar{K}_{i,2} = K_{i,2}^z$，$\bar{K}_{i,3} = K_{i,3}^z$。最后，输出转换密钥$\mathrm{TK} = (S, \bar{K}_0, \bar{K}_1, \{\bar{K}_{i,1}, \bar{K}_{i,2}, \bar{K}_{i,3}\}_{i\in[1,k]})$和取回密钥$\mathrm{RK} = z$。

Transform(PK, CT, TK)：该算法首先判断S是否满足$(\boldsymbol{M}, \rho)$。若不满足，则返回终止符$\perp$；否则，设置$I = \{i : \rho(i) \in S\}$，然后计算常数$w_i \in Z_p$且满足$\sum_{i\in I} w_i \cdot M_i = (1, 0, \cdots, 0)$。然后计算$\mathrm{CM}'$：

$$CM' = e(C_0, \bar{K}_0) \Big/ \Big[e(w^{\sum_{i\in I} C_{i,4} w_i}, \bar{K}_1) \cdot \prod_{i\in I} \Big(e(C_{i,1}, \bar{K}_1) \cdot e(C_{i,2} \cdot u^{C_{i,5}}, K_{j,1}) \cdot e(C_{i,3}, u^{\bar{K}_{j,3}} \cdot \bar{K}_{j,2}) \Big)^{w_i} \Big] = e(g, g)^{\alpha s z} \tag{1}$$

其中，j是集合S中的属性$\rho(i)$的索引(其独立于i)。最后，输出转换密文$\mathrm{CT}' = (\mathrm{CM}', C_{\mathrm{SE}})$。

Decrypt$(\mathrm{PK}, \mathrm{CT}', \mathrm{RK}, \mathrm{VK}_m)$：该算法首先计算$\mathrm{key} = (\mathrm{CM}')^{1/\mathrm{RK}} = e(g, g)^{\alpha s}$和$\mathrm{Tag}_0 = H_0(\mathrm{key})$。若$H_1(\mathrm{Tag}_0 || C_{\mathrm{SE}}) \neq \mathrm{VK}_m$，则返回终止符$\perp$；否则，计算$K_{\mathrm{SE}} = H(\mathrm{key})$并返回$m = \mathrm{SE.Dec}(K_{\mathrm{SE}}, C_{\mathrm{SE}})$。

3.2 安全证明

定理1　假设RW [16, Sec. 4]的CP-ABE方案是选择性CPA安全，$\mathcal{H}$是一个两两独立散列函数族，SE是语义安全的一次性对称加密方案且涉及参数满足$0 < l_{\mathrm{SE}} \leq (\lg|\mathcal{K}| - l_{H_0}) - 2\lg(1/\varepsilon_H)$。那么本文所提3OVD-CP-ABE方案是选择性CPA安全。

证明　首先，定义以下3个游戏：

Game0：原始CPA安全游戏。$\mathcal{A}$选择一个挑战访问策略$(\boldsymbol{M}^*, \rho^*)$，然后根据该访问策略生成挑战密文和验证标志$(\mathrm{CT}^*, \mathrm{VM}_m^*) = ((\mathrm{CM}^*, C_{\mathrm{SE}}^*), \mathrm{Tag}^*)$。$R^* \in G_T$表示密文$\mathrm{CT}_{\mathrm{RW}}^*$所对应的明文消息，$K_{\mathrm{SE}}^* = H^*(\mathrm{key}^*)$表示密文$C_{\mathrm{SE}}^*$所对应的对称密钥，其中$\mathrm{key}^* = C^*/R^*$。

Game1：在Game0中，通过执行$\mathrm{Encrypt}_{\mathrm{RW}}(\mathrm{PK}_{\mathrm{RW}}^*, R^*, (\boldsymbol{M}^*, \rho^*))$获得密文$\mathrm{CT}_{\mathrm{RW}}^*$，其中$R^* \in G_T$是明文消息，设置$\mathrm{key}^* = C^*/R^*$并计算$\mathrm{Tag}_0^* = H_0^*(\mathrm{key}^*)$和$K_{\mathrm{SE}}^* = H^*(\mathrm{key}^*)$；在Game1中，$\mathrm{CT}_{\mathrm{RW}}^*$仍然是$R^* \in G_T$对应的密文，但是$\mathrm{key}^* = C^*/K^* \in G_T$，计算$\mathrm{Tag}_0^* = H_0^*(\mathrm{key}^*)$和$K_{\mathrm{SE}}^* = H^*(\mathrm{key}^*)$。此外，2个游戏完全相同。

Game2：在Game2中用随机串$\mathrm{Ra}_{\mathrm{SE}}^* \in \{0,1\}^{l_{\mathrm{SE}}}$代替$K_{\mathrm{SE}}^*$，其余与Game1完全相同。

通过引理1证明Game0与Game1不可区分；通过引理2证明Game1与Game2不可区分；通过引理3证明$\mathcal{A}$在Game2中的优势可以忽略不计。因此，可以推导出$\mathcal{A}$在Game0中的优势可以忽略不计。

引理1　假设RW [16,Sec.4]的CP-ABE方案是选择性CPA安全，那么$\mathcal{A}$区分Game0和Game1的优势可以忽略不计。

引理1证明过程　假设存在$\mathcal{A}$可以以不可忽略的优势ε区分Game0和Game1，那么可以构建一个仿真者$\mathcal{B}$在多项式时间内以不可忽略的优势ε攻破RW [16,Sec.4]的CP-ABE方案的CPA安全。$\mathcal{C}$是RW [16,Sec.4]的CP-ABE方案的挑战者。$\mathcal{B}$基于挑战密文模拟敌手$\mathcal{A}$在Game0和Game1的视角。$\mathcal{B}$同时扮演RW [16,Sec.4]的CP-ABE方案的敌手，与本文$\mathcal{A}$交互过程如下：

系统初始化：$\mathcal{A}$选择一个挑战访问策略$T^* = (\boldsymbol{M}^*, \rho^*)$并发送给$\mathcal{B}$，然后$\mathcal{B}$将$T^* = (\boldsymbol{M}^*, \rho^*)$发送给$\mathcal{C}$。

系统建立：$\mathcal{B}$与$\mathcal{C}$交互，从$\mathcal{C}$处获得挑战公共参数$\mathrm{PK}_{\mathrm{RW}}^*$，然后选择2个抵抗合谋攻击的哈希函数$H_0^*$和$H_1^*$，一个随机提取器$H^* \in \mathcal{H}$和一个语义安全的一次对称加密方案$\mathrm{SE}^*$。最后，将$(\mathrm{PK}_{\mathrm{RW}}^*, H_0^*, H_1^*, H^*, \mathrm{SE}^*)$发送给$\mathcal{A}$作为最终的挑战公钥PK。

询问阶段1：$\mathcal{B}$初始化空表T，空集合D和整数$j = 0$。$\mathcal{A}$可以适应性地做出下列任何查询：

(1) Create(S)：$\mathcal{B}$设置$j := j + 1$，并从$\mathcal{A}$处获得私钥询问请求$S = \{A_1, A_2, \cdots, A_k\} \subseteq Z_p$，然后将$S$发送给$\mathcal{C}$，从$\mathcal{C}$处获得私钥$\mathrm{SK}_{\mathrm{RW}}^* = (S^*, K_0, K_1, \{K_{i,1}, K_{i,2}\}_{i\in[1,k]})$，其中$K_0 = g^{\alpha} w^r$，$K_1 = g^r$，$K_{i,1} = g^{r_i}$，$K_{i,2} = (u^{A_i}h)^{r_i} \cdot v^{-r}$。然后$\mathcal{B}$选择随机盲化值$b_1, b_2, \cdots, b_k \in Z_p$并计算$K'_{i,1} = K_{i,1} = g^{r_i}$，$K'_{i,2} = K_{i,2} \cdot u^{-b_i} = (u^{A_i}h)^{r_i} \cdot v^{-r} \cdot u^{-b_i}$，$K'_{i,3} = b_i$，生成私钥为$\mathrm{SK} = (S, K_0, K_1, \{K_{i,1}, K_{i,2}, K_{i,3}\}_{i\in[1,k]})$。最后，$\mathcal{B}$运行KeyGen.out获得TK和RK，并将元组$(j, S,$

SK, TK, RK)存储与表T中。注意，属性集合S不满足访问策略T^*，即$f(T^*, S) \neq 1$。

(2) Corrupt.SK(i)：$\mathcal{B}$检查表T中是否存在第i^{th}个元组(i, S, SK)。若存在，则设置$D := D \cup \{S\}$并返回SK；若不存在，则返回终止符$\perp$。

(3) Corrupt.TK(i)：$\mathcal{B}$检查表T中是否存在第i^{th}个元组(i, S, TK)。若存在，则返回TK；否则返回终止符$\perp$。

挑战阶段：敌手$\mathcal{A}$提交2个等长的消息m_0和m_1。$\mathcal{B}$随机选择2个独立的值$R^*, K^* \in G_T$，然后向$\mathcal{C}$请求加密$((R^*, K^*), T^*)$，$\mathcal{C}$将挑战密文$\text{CT}^*_{\text{RW}} = ((\boldsymbol{M}^*, \rho^*), C, C_0, \{C_{j,1}, C_{j,2}, C_{j,3}\}_{j\in[1,l]})$返回给$\mathcal{B}$，其中，$C = R^* e(g,g)^{\alpha s}$或者$C = K^* e(g,g)^{\alpha s}$，$C_0 = g^s$；对于$\boldsymbol{M}^*$的每一行，$C_{j,1} = w^{\lambda_j} v^{t_j}$，$C_{j,2} = (u^{\rho^*(j)} h)^{-t_j}$，$C_{j,3} = g^{t_j}$。$\mathcal{B}$随机选择$z_1, z_2, \cdots, z_l, {z'}_1, {z'}_2, \cdots, {z'}_l \in Z_p$并计算$\text{CM}^* = ((\boldsymbol{M}^*, \rho^*), C^*, C_0, \{C'_{j,1}, C'_{j,2}, C'_{j,3}, C'_{j,4}, C'_{j,5}\}_{j\in[1,l]})$，其中$C^* = C$，$C'_{j,1} = C_{j,1} \cdot w^{-z_j} = w^{\lambda_j - z_j} v^{t_j}$，$C'_{j,2} = C_{j,2} \cdot u^{-z'_j} = (u^{\rho^*(j)} h)^{-t_j} \cdot u^{-z'_j}$，$C'_{j,3} = C_{j,3} = g^{t_j}$，$C'_{j,4} = z_j$，$C'_{j,5} = z'_j$。然后$\mathcal{B}$计算$K^*_{\text{SE}} = H^*(\text{key}^*)$和$\text{Tag}^*_0 = H^*_0(\text{key}^*)$，其中$\text{key}^* = C^*/R^* \in G_T$。$\mathcal{B}$随机选择$b \in \{0,1\}$并计算$C^*_{\text{SE}} = \text{SE}^*.\text{Enc}(K^*_{\text{SE}}, m_b)$和$\text{Tag}^* = H^*_1(\text{Tag}^*_0 || C^*_{\text{SE}})$。最后，$\mathcal{B}$将$\text{CT}^*_b = \{\text{CM}^*, C^*_{\text{SE}}\}$和$\text{VK}^*_m = \text{Tag}^*$发送给$\mathcal{A}$。若$\text{CT}^*_{\text{RW}}$是$R^*$的密文，则$\text{CT}^*_b$是Game0中的挑战密文；若$\text{CT}^*_{\text{RW}}$是$K^*$的密文，则$\text{CT}^*_b$是Game1中的挑战密文。

查询阶段2：类似查询阶段1，$\mathcal{A}$继续向$\mathcal{B}$提交一系列属性列表，其限制与查询阶段1相同。

猜测阶段：$\mathcal{A}$输出一个值$b' \in \{0,1\}$作为对b的猜测。如果$b' = b$，$\mathcal{B}$输出0，即CT^*_{RW}是R^*的密文；如果$b' \neq b$，$\mathcal{B}$输出1，即CT^*_{RW}是K^*的密文。

从上述游戏分析，$\mathcal{B}$能够完美模拟$\mathcal{A}$在Game0和Game1的视角。通过本文假设，$\mathcal{A}$在Game0中正确猜测出b的概率与$\mathcal{A}$在Game1中正确猜测出b的概率相差一个不可忽略的值ε，因此$\mathcal{A}$能够以不可忽略的优势ε区分Game0和Game1。当$\mathcal{A}$在Game0中，CT^*_{RW}是R^*的密文；当$\mathcal{A}$在Game1中，CT^*_{RW}是K^*的密文。因此，$\mathcal{B}$可以以不可忽略的优势ε攻破RW [16,Sec.4]的CP-ABE方案的安全性。

引理2 假设$\mathcal{H}$是一个两两独立散列函数族，那么$\mathcal{A}$在Game1和Game2中的视角是静态不可区分的。

引理2证明过程 在Game1和Game2中，K^*完全独立于CT^*_{RW}，H^*，PK^*_{RW}。另外，$\text{Tag}^*_0 = H^*_0(C^*/K^*)$至多有$2^{l_{H_0}}$种可能的值。然后可以导出：

$$\begin{aligned}&\tilde{H}'_\infty(K^*|(\text{PK}, \text{CT}^*_{\text{RW}}, H^*, \text{Tag}^*_0))\\ &\geq \tilde{H}'_\infty(K^*|(\text{PK}, \text{CT}^*_{\text{RW}}, H^*)) - l_{H_0}\\ &= \lg|\mathcal{K}| - l_{H_0}\end{aligned} \tag{2}$$

注：X, Y, Z是随机值，若Y至多有2^r种可能的值，那么可以得出$\tilde{H}'_\infty(X|(Y,Z)) \geq \tilde{H}'_\infty(X|Z) - r$。

因为$0 < l_{\text{SE}} \leq (\lg|\mathcal{K}| - l_{H_0}) - 2\lg(1/\varepsilon_{H'})$，所以从$\mathcal{A}$的视角(除去$C^*_{\text{SE}}$)观察，Game1中的对称密钥$K^*_{\text{SE}} = H^*(C^*/K^*)$与Game2中真实随机的对称密钥$\text{Ra}^*_{\text{SE}} \in \{0,1\}^{l_{\text{SE}}}$有$\varepsilon_{H'}$静态不可区分的概率。即，$K^*_{\text{SE}}$与$\text{Ra}^*_{\text{SE}}$是不可区分的。另外，$C^*_{\text{SE}}$是$K^*_{\text{SE}}$的函数，$\text{Tag}^*$是$\text{Tag}^*_0$和$C^*_{\text{SE}}$的函数。因此，它们不会增加上述2个分布之间的距离。也就是说，Game1和Game2中一系列的变化不能改变上述2个分布之间的距离。因此，Game1和Game2之间的不可区分性与K^*_{SE}和Ra^*_{SE}之间的不可区分性相同。即，在$\mathcal{A}$的视角中，Game1和Game2是不可区分的。

引理3 假设对称加密方案SE^*是语义安全的，那么$\mathcal{A}$在Game2中的优势可以忽略不计。

引理3证明过程 在Game2中，对称密钥Ra^*_{SE}是一个随机串，$\mathcal{A}$完全不知道该值。因此，我们可以根据$\mathcal{A}$直接构建一个$\mathcal{B}$去攻击SE^*的语义安全，然后可以得出$|\Pr[\text{Game2}] - 1/2| \leq \text{Adv}^{\text{SE}^*}_{\mathcal{B}}(\lambda)$。因为对称加密方案$\text{SE}^*$是语义安全的，因此$\mathcal{A}$在Game2中的优势可以忽略不计。

通过引理1，引理2和引理3可以联合推导出3OVD-CP-ABE方案是选择性CPA安全。 证毕

定理2 假设H_0和H_1是抵抗合谋攻击的哈希函数，那么3OVD-CP-ABE方案具有可验证性。

证明 假设$\mathcal{A}$可以攻破可验证性，那么可以构建一个$\mathcal{B}$打破底层哈希函数H_0和H_1的抗合谋攻击能力。$\mathcal{A}$提交2个挑战哈希函数(H^*_0, H^*_1)，然后$\mathcal{B}$仿真实验过程如下：

系统建立：$\mathcal{B}$执行Setup算法获得公钥PK和主私钥MSK，并用H^*_0和H^*_1替换公钥PK中的哈希函数。

查询阶段1：$\mathcal{B}$按照上述查询阶段1方式适应性回答$\mathcal{A}$的询问。

挑战阶段：$\mathcal{A}$提交一个挑战明文m^*和一个访问策略T^*。$\mathcal{B}$调用$\text{Encrypt}_{\text{RW}}(\text{PK}^*_{\text{RW}}, R^*, (\boldsymbol{M}^*, \rho^*))$获得$R^* \in G_T$的挑战密文$\text{CM}^* = ((\boldsymbol{M}^*, \rho^*), C, C_0, \{C'_{j,1}, C'_{j,2}, C'_{j,3}, C'_{j,4}, C'_{j,5}\}_{j\in[1,l]})$，其中$C = R^* e(g,g)^{\alpha s}$，$C_0 = g^s$；对于$\boldsymbol{M}^*$的每一行，计算$C_{j,1} = w^{\lambda_j} v^{t_j}$，$C_{j,2} = (u^{\rho^*(j)} h)^{-t_j}$，$C_{j,3} = g^{t_j}$。然后，$\mathcal{B}$随机选择$z_1, z_2, \cdots, z_l, {z'}_1, {z'}_2, \cdots, {z'}_l \in Z_p$并计算$\text{CM}^* = ((\boldsymbol{M}^*, \rho^*), C^*, C_0, \{C'_{j,1}, C'_{j,2}, C'_{j,3}, C'_{j,4}, C'_{j,5}\}_{j\in[1,l]})$，其中

$C^* = C$，$C'_{j,1} = C_{j,1} \cdot w^{-z_j} = w^{\lambda_j - z_j} v^{t_j}$，$C'_{j,2} = C_{j,2} \cdot u^{-z'_j} = (u^{\rho^*(j)} h)^{-t_j} \cdot u^{-z'_j}$，$C'_{j,3} = C_{j,3} = g^{t_j}$，$C'_{j,4} = z_j$，$C'_{j,5} = z'_j$。$\mathcal{B}$ 计算 $K^*_{\mathrm{SE}} = H^*(\mathrm{key}^*)$ 和 $\mathrm{Tag}^*_0 = H^*_0(\mathrm{key}^*)$，其中 $\mathrm{key}^* = C^*/R^* \in G_T$，然后计算 $C^*_{\mathrm{SE}} = \mathrm{SE}^*.\mathrm{Enc}(K^*_{\mathrm{SE}}, m_b)$ 和 $\mathrm{Tag}^* = H^*_1(\mathrm{Tag}^*_0 || C^*_{\mathrm{SE}})$。最后，$\mathcal{B}$ 将 $\mathrm{CT}^* = \{CM^*, C^*_{\mathrm{SE}}\}$ 和 $\mathrm{VK}^*_m = \mathrm{Tag}^*$ 发送给 $\mathcal{A}$。$\mathcal{B}$ 同时自己保留 VK^*_m 和 $\{R^*, C^*_{\mathrm{SE}}\}$。

查询阶段2：$\mathcal{B}$ 按照上述查询阶段2方式适应性回答 $\mathcal{A}$ 的询问。

猜测阶段：$\mathcal{A}$ 输出一个属性集合 $S^*(f(T^*, S^*) = 1)$，中间解密密文 CM' 和 C_{SE}。

若 $\mathcal{A}$ 攻破可验证性，那么 $\mathcal{B}$ 将通过 $\mathrm{Decrypt}(\mathrm{PK}, \mathrm{CM}', C_{\mathrm{SE}}, \mathrm{RK}, \mathrm{VK}^*_m)$ 恢复出明文 $m \notin \{m^*, \perp\}$。现在分析 $\mathcal{A}$ 成功的可能性。若 $H^*_1(\mathrm{Tag}_0 || C_{\mathrm{SE}}) \neq \mathrm{Tag}^*$，则解密算法输出终止符 $\perp$，其中，$\mathrm{Tag}_0 = H^*_0(C^*/R)$ 和 $R = \mathrm{Decrypt}_{\mathrm{RW}}(\mathrm{RK}, \mathrm{CM}')$。因此，本文只需考虑以下2种情况：(1) $(\mathrm{Tag}_0, C_{\mathrm{SE}}) \neq (\mathrm{Tag}^*_0, C^*_{\mathrm{SE}})$。因为 $\mathcal{B}$ 知道 $(\mathrm{Tag}^*_0, C^*_{\mathrm{SE}})$，若这种情况发生，则 $\mathcal{B}$ 立即得到哈希函数 H^*_1 的碰撞；(2) $(\mathrm{Tag}_0, C_{\mathrm{SE}}) = (\mathrm{Tag}^*_0, C^*_{\mathrm{SE}})$，但 $R \neq R^*$。因为 $H^*_0(C^*/R) = \mathrm{Tag}_0 = \mathrm{Tag}^*_0 = H^*_0(C^*/R^*)$，所以这将打破 H^*_0 的抗合谋攻击能力。

通过上述2种情况分析，完成了定理2的安全证明。注意：在证明过程中，CM^* 和 CM' 都包含 $C = Xe(g,g)^{\alpha s}$ 这种形式的组件。但是实际方案中，CM和 CM' 不包含 $C = Xe(g,g)^{\alpha s}$。

4　I3OVD-CP-ABE方案

本节基于3OVD-CP-ABE方案提出I3OVD-CP-ABE方案，与3OVD-CP-ABE方案相比，I3OVD-CP-ABE在密文转换阶段需要恒定常数个双线性对计算。

Setup(ς, U)：该算法与3OVD-CP-ABE方案的Setup(ς, U)算法相同。

Offline.Encrypt(PK)：该算法设置 $C_1 = g^{T_1}$ 和 $C_{i,3} = (t_i - T_1) \bmod p$，其中 $T_1 \in Z_p$ 是一个随机指数。其余设置与3OVD-CP-ABE方案的Offline.Encrypt(PK)算法相同，即 $\mathrm{IT}_{at,i} = (\lambda'_i, x_i, t_i, C_1, C_{i,1}, C_{i,2}, C_{i,3})$。

Online.Encrypt($\mathrm{PK}, \mathrm{IT}, m, (\boldsymbol{M}, \rho)$)：该算法的CM包含 $C_1 = g^{T_1}$ 和 $C_{j,3} = (t_j - T_1) \bmod p$。其余设置与3OVD-CP-ABE方案的Online.Encrypt($\mathrm{PK}, \mathrm{IT}, m, (\boldsymbol{M}, \rho)$)算法相同。

Offline.KeyGen(MSK)：该算法设置 $K'_1 = g^{T_2}$ 和 $K'_{i,1} = (r_i - T_2) \bmod p$，其中 $T_2 \in Z_p$ 是一个随机指数。其余设置与3OVD-CP-ABE方案的Offline.KeyGen(MSK)算法相同，即 $\mathrm{IK}_{at,i} = (r_i, b_i, K'_1, K'_{i,1}, K'_{i,2})$。

Online.KeyGen(PK, IK, S)：该算法的SK包含 $K'_1 = g^{T_2}$ 和 $K_{i,1} = K'_{i,1} = (r_i - T_2) \bmod p$。其余设置与3OVD-CP-ABE方案的Online.KeyGen(PK, IK, S)算法相同。

KeyGen.out(PK, SK)：该算法的TK包含 $\bar{K}_1{}' = K_1{}'^z$。其余设置与3OVD-CP-ABE方案的KeyGen.out(PK, SK)算法相同。

Transform(PK,CT,TK)：该算法的 CM' 按下述公式计算。其余设置与3OVD-CP-ABE方案的Transform(PK, CT, TK)算法相同。

$$\begin{aligned}\mathrm{CM}' = e(C_0, \bar{K}_0) \Bigg/ \Bigg[e\Big(w^{\sum_{i\in I} C_{i,4} w_i}, \bar{K}_1\Big) \cdot \prod_{i\in I} \Big(e(C_{i,1}, \bar{K}_1) \\ \cdot e(C_{i,2} \cdot u^{C_{i,5}}, \bar{K}'_1 \cdot g^{\bar{K}_{j,1}}) \cdot e(C_1 \cdot g^{C_{i,3}}, u^{\bar{K}_{j,3}} \cdot \bar{K}_{j,2}) \Big)^{w_i} \Bigg] \\ = e(g,g)^{\alpha s z} \end{aligned} \tag{3}$$

Decrypt($\mathrm{PK}, \mathrm{CT}', \mathrm{RK}, \mathrm{VK}_m$)：该算法与3OVD-CP-ABE方案的Decrypt($\mathrm{PK}, \mathrm{CT}', \mathrm{RK}, \mathrm{VK}_m$)算法相同。

5　方案分析及实验验证

5.1　效率分析

本节从原理层面将本文方案与其它ABE方案[17,18]在密钥生成、加密和解密方面进行对比。在对比过程中，s 表示系统属性集合的大小，l 表示LSSS中矩阵 $\boldsymbol{M}$ 的行数，y 表示满足访问策略 $(\boldsymbol{M}, \rho)$ 的属性集合的大小，E_G 和 E_{G_T} 分别表示 G 和 G_T 中模指数计算，Mul表示群中的乘法计算，H 表示哈希运算时间，P 表示双线性对计算。模指数、群中乘法运算和双线性对的计算量相对于其它计算需要更多的计算时间，而 Z_p 中的计算时间非常快，因此本文忽略了次要因素。原理对比分析如表1所示。

如表1所示，密钥生成、数据加密和数据解密阶段所需计算量与参与计算的属性数量成线性正相关。文献[17,18]和本文方案采用离线/在线密钥生成和离线/在线数据加密技术，因此，密钥生成和数据加密过程中大部分的计算在离线阶段执行，而在线阶段只需要少量的计算就能完成密钥生成和数据加密工作。此外，文献[18]和本文方案在解密过程中的大部分计算可以外包到雾节点，用户只需少量计算即可完成数据解密。而文献[17]方案中，数据用户需要大量计算才能完成数据解密，这对于资源有限的智能终端是不可承受的。

在I3OVD-CP-ABE中，CM' 和 CM'' 可以按照式(4)，式(5)计算：

表 1 计算效率对比分析

算法		文献[17]方案	文献[18]方案	3OVD-CP-ABE	I3OVD-CP-ABE
密钥生成	离线	$(3s+4)E_G+(s+1)$Mul	$(3s+4)E_G+(s+1)$Mul	$(3s+4)E_G+(s+1)$Mul	$(2s+5)E_G+(s+1)$Mul
	在线	sMul	sMul	sMul	sMul
加密	离线	$(5l+1)E_G+1E_{G_T}+2l$Mul	$(5l+1)E_G+1E_{G_T}+2l$Mul	$(5l+1)E_G+1E_{G_T}+2l$Mul	$(4l+2)E_G+1E_{G_T}+2l$Mul
	在线	0	$2E_G$+Mul	$3H$	$3H$
解密	外包	—	$(2y+1)E_{G_T}+yE_{G_T}$ $+(3y+2)P+2y$Mul	$(2y+1)E_{G_T}+yE_{G_T}$ $+(3y+2)P+2y$Mul	$(5y+5)E_G+10P+4y$Mul
	用户	$(2y+1)E_{G_T}+yE_{G_T}$ $+(3y+2)P+2y$Mul	$2E_G+1E_{G_T}+$ Mul	$1E_{G_T}+3H$	$1E_{G_T}+3H$

$$\begin{aligned}\mathrm{CM}''=&\prod_{i\in I}\Big(e(C_{i,1},\bar{K}_1)\cdot e(C_{i,2}\cdot u^{C_{i,5}},\bar{K}'_1\cdot g^{\bar{K}_{j,1}})\\&\cdot e(C_1\cdot g^{C_{i,3}},u^{\bar{K}_{j,3}}\cdot\bar{K}_{j,2})\Big)^{w_i}\\=&\,e\left(\prod_{i\in I}C_{i,1}^{w_i},\bar{K}_1\right)\cdot e\left(\prod_{i\in I}C_{i,2}^{w_i},\bar{K}'_1\right)\\&\cdot e\left(\prod_{i\in I}C_{i,2}^{w_i\bar{K}_{j,1}},g\right)\cdot e\left(u^{\sum_{i\in I}w_iC_{i,5}},\bar{K}'_1\cdot\right)\\&\cdot e\left(u^{\sum_{i\in I}w_iC_{i,5}(\bar{K}_{j,1}+\bar{K}_{j,3})},g\right)\cdot e\left(C_1,\prod_{i\in I}\bar{K}_{j,2}^{w_i}\right)\\&\cdot e\left(C_1,u^{\sum_{i\in I}w_i\bar{K}_{j,3}}\right)\cdot e\left(g,\prod_{i\in I}\bar{K}_{j,2}^{w_iC_{i,3}}\right)\end{aligned}\tag{4}$$

$$\mathrm{CM}'=\frac{e(C_0,\bar{K}_0)}{e\left(w^{\sum_{i\in I}C_{i,4}w_i},\bar{K}_1\right)\cdot\mathrm{CM}''}\tag{5}$$

在计算转换密文过程中，3OVD-CP-ABE需要$3y$+2个双线性对计算，I3OVD-CP-ABE只需要10个双线性对计算；3OVD-CP-ABE需要$(2y+1)E_{G_T}+yE_{G_T}$个模指数计算，而I3OVD-CP-ABE需要$(5y+5)E_G$个模指数计算。综合分析，相比于3OVD-CP-ABE, I3OVD-CP-ABE仍然是高效的。因此，I3OVD-CP-ABE也比文献[17,18]方案更加高效。

5.2 实验分析

实验环境为64 bit Ubuntu 14.04操作系统，Intel® CoreTM i5-6200U(2.3 GHz)，内存8G，实验代码基于Pairing-based Cryptography Library (PBC-0.5.14)与cpabe-0.11进行修改与编写，并且使用224位MNT的椭圆曲线。对于$\lambda=80$bit安全参数，本文选择$l_{H_0}=l_{H_1}=160$，并封装了一个随机128位对称密钥l_{SE}。在本文方案中，首先将群G_T中的元素C_T散列成一个随机的“种子”，然后用一个伪随机数生成器(例如，AES方案)将其扩展到512位密钥K。这足以保证$\lg|\mathcal{K}|-l_{H_1}\geq l_{\mathrm{SE}}+2\lg 2^{\lambda}$。

实验设置：本文在密钥生成、数据加密和数据解密阶段中，将其各自所涉及的属性数量以10为增量，从10增加到100，以这种方式完成实验仿真。对于每种类型的仿真实验，本文重复20次实验且每次实验完全独立，然后取平均值作为最终实验结果。仿真实验结果如图1所示。

图1中每个子图给出本文方案1(3OVD-CP-ABE)，本文方案2(I3OVD-CP-ABE)与文献[17,18]方案执行时间的对比情况。文献[17]方案没有采用解密外包技术，所以在图1(e)中没有其相应曲线，而图1(f)中为方便且不丢失细节的展示用户解密时间，本文在图1(f)中将文献[17]的解密时间缩小100倍。即，文献[17]的实际解密时间为图1(f)中相应曲线值乘以100。

图1(a)和图1(b)说明离线密钥生成阶段承担大部分密钥生成工作，密钥生成时间与属性数量成线性关系。图1(c)和图1(d)说明离线加密阶段承担大部分加密工作，加密时间与访问策略的复杂度成线性关系。图1(e)和图1(f)说明雾节点承担大部分解密工作。密文转换时间与访问策略的复杂度成线性关系。用户解密只需要常量计算，与访问策略的复杂性无关。但文献[17]方案没有采用解密外包技术，其用户需要计算大量双线性对运行。

图1(a)，图1(c)和图1(e)说明文献[17,18]与3OVD-CP-ABE方案的计算效率是相似的。而I3OVD-CP-ABE方案更有效。特别地，在密文转换阶段，I3OVD-CP-ABE方案的效率是其它方案的2倍多。由上述实验过程得到的实验数据与5.1节的理论分析相吻合。图1(b)和图1(d)说明在线阶段需要很少的计算。在5.1节中，本文分析了在线密钥生成时间和在线加密时间，在分析过程中忽略了Z_p中的运算，所以出现在线加密时间为零或恒定。但实际仿真过程中，会累积这些运算量极小的计算时间，所以存在理论分析与实际仿真时间存在微小差异的情况。图1(f)说明用户解密时间非常少，且与访问策略的复杂度无关。但文献[17]没有采用解

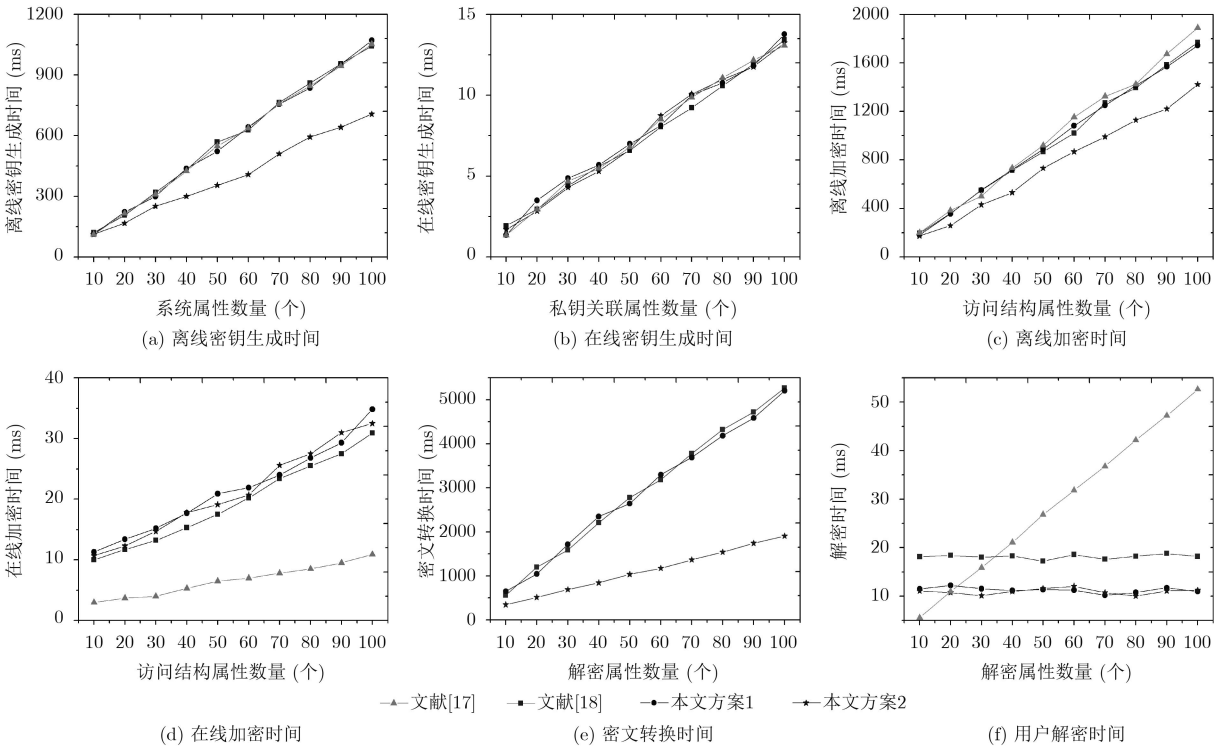

图 1 本文方案与文献[17,18]方案仿真时间对比

密外包技术，所以其用户需要承担繁重的计算任务。

综上所述，离线/在线技术和外包技术对于应用于雾-云计算环境和物联网的ABE至关重要。总之，本文方案采用离线/在线技术，可以将大部分密钥生成和加密工作转移到离线阶段；同时本文方案采用了外包技术，可以将大部分的解密计算外包到雾节点。与其它方案相比，本文方案在效率方面有一定优势。

6　结束语

本文提出了一种可验证外包解密的离线/在线属性基加密方案3OVD-CP-ABE。该方案能够将私钥生成阶段分为离线和在线阶段，即属性授权机构在闲时运行离线私钥生成算法，预计算私钥生成所需组件，当数据用户提交属性集合申请私钥时，属性授权机构再运行在线私钥生成算法通过预计算的私钥组件最终生成用户私钥；该系统同时将加密阶段分为离线和在线阶段，即数据拥有者在闲时运行离线加密算法，预计算加密所需组件，当数据拥有者需要加密明文时，其再运行在线加密算法通过预计算的密文组件最终完成明文的加密工作；该系统还能在解密阶段将部分解密计算外包给云服务商以减少数据用户的计算量，同时还能够验证云服务商计算的正确性。然后，本文对方案进行了选择明文攻击的安全性证明和可验证性的安全证明。之后，本文改进了3OVD-CP-ABE，提出I3OVD-CP-ABE算法，提高了3OVD-CP-ABE的计算效率，在密文转换阶段，将转换所需双线性对的数量降为恒定常数。最后，从理论和实验两方面对本文方案进行性能分析，实验结果表明该方案是有效且实用的。

参 考 文 献

[1] KHAN S, PARKINSON S, and QIN Yongrui. Fog computing security: A review of current applications and security solutions[J]. *Journal of Cloud Computing*, 2017, 6(1): 19–41. doi: 10.1186/s13677-017-0090-3.

[2] SAHAI A and WATERS B. Fuzzy identity-based encryption[C]. Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, 2005: 457–473. doi: 10.1007/11426639_27.

[3] GOYAL V, PANDEY O, SAHAI A, *et al*. Attribute-based encryption for fine-grained access control of encrypted data[C]. Proceedings of the 13th ACM Conference on Computer and Communications Security, Alexandria, USA, 2006: 89–98. doi: 10.1145/1180405.1180418.

[4] BETHENCOURT J, SAHAI A, and WATERS B. Ciphertext-policy attribute-based encryption[C]. Proceedings of the IEEE Symposium on Security and Privacy, Berkeley, USA, 2007: 321–334. doi: 10.1109/SP.2007.11.

[5] GREEN M, HOHENBERGER S, and WATERS B. Outsourcing the decryption of ABE ciphertexts[C].

Proceedings of the 20th USENIX Conference on Security, San Francisco, USA, 2011: 34.

[6] LAI Junzuo, DENG R H, GUAN Chaowen, *et al*. Attribute-based encryption with verifiable outsourced decryption[J]. *IEEE Transactions on Information Forensics and Security*, 2013, 8(8): 1343–1354. doi: 10.1109/TIFS.2013.2271848.

[7] ZHAO Zhiyuan and WANG Jianhua. Verifiable outsourced ciphertext-policy attribute-based encryption for mobile cloud computing[J]. *KSII Transactions on Internet and Information Systems*, 2017, 11(6): 3254–3272. doi: 10.3837/tiis.2017.06.024.

[8] FAN Kai, WANG Junxiong, WANG Xin, *et al*. A secure and verifiable outsourced access control scheme in fog-cloud computing[J]. *Sensors*, 2017, 17(7): 1695–1710. doi: 10.3390/s17071695.

[9] LI Jiguo, SHA Fengjie, ZHANG Yichen, *et al*. Verifiable outsourced decryption of attribute-based encryption with constant ciphertext length[J]. *Security and Communication Networks*, 2017, 2017: 1–11. doi: 10.1155/2017/3596205.

[10] ZHANG Rui, MA Hui, and LU Yao. Fine-grained access control system based on fully outsourced attribute-based encryption[J]. *Journal of Systems and Software*, 2017, 125(3): 344–353. doi: 10.1016/j.jss.2016.12.018.

[11] EVEN S, GOLDREICH O, and MICALI S. On-line/off-line digital signatures[C]. Proceedings of the Conference on the Theory and Application of Cryptology, Santa Barbara, USA, 1989: 263–275. doi: 10.1007/0-387-34805-0_24.

[12] LIU J K, BAEK J, ZHOU Jianying, *et al*. Efficient online/offline identity-based signature for wireless sensor network[J]. *International Journal of Information Security*, 2010, 9(4): 287–296. doi: 10.1007/s10207-010-0109-y.

[13] GUO Fuchun, MU Yi, and CHEN Zhide. Identity-based online/offline encryption[C]. Proceedings of the International Conference on Financial Cryptography and Data Security, Cozumel, Mexico, 2008: 247–261. doi: 10.1007/978-3-540-85230-8_22.

[14] LIU J K and ZHOU Jianying. An efficient identity-based online/offline encryption scheme[C]. Proceedings of the International Conference on Applied Cryptography and Network Security, Paris-Rocquencourt, France, 2009: 156–167. doi: 10.1007/978-3-642-01957-9_10.

[15] CHOW S S M, LIU J K, and ZHOU Jianying. Identity-based online/offline key encapsulation and encryption[C]. Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security, Hong Kong, China, 2011: 52–60. doi: 10.1145/1966913.1966922.

[16] ROUSELAKIS Y and WATERS B. Practical constructions and new proof methods for large universe attribute-based encryption[C]. Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, Berlin, Germany, 2013: 463–474. doi: 10.1145/2508859.2516672.

[17] HOHENBERGER S and WATERS B. Online/offline attribute-based encryption[C]. Proceedings of the International Workshop on Public Key Cryptography, Buenos Aires, Argentina, 2014: 293–310. doi: 10.1007/978-3-642-54631-0_17.

[18] LIU Zechao, JIANG Z L, WANG Xuan, *et al*. Offline/online attribute-based encryption with verifiable outsourced decryption[J]. *Concurrency and Computation: Practice and Experience*, 2017, 29(7): 1–17. doi: 10.1002/cpe.3915.

赵志远：男，1989年生，博士生，研究方向为云安全与属性加密.
孙 磊：男，1973年生，教授，研究方向为云计算与信息安全.
户家富：男，1981年生，讲师，研究方向为网络计算与信息安全.
周时娥：女，1978年生，讲师，研究方向为科技英语与云计算.