基于属性加密的云存储方案研究

王光波*① 王建华②
①(解放军信息工程大学 郑州 450001)
②(空军电子技术研究所 北京 100195)

摘 要：云存储中往往采用属性加密方案来实现细粒度的访问控制，为了进一步保护访问控制策略中的敏感信息，并解决授权中心单独为用户生成密钥而产生的密钥托管问题。该文对访问控制策略中的属性进行重新映射，以实现其隐私性。另外在密钥生成算法中设计一个双方计算协议，由用户产生密钥的部分组件，与授权中心共同生成密钥以解决密钥托管问题。最后在标准模型下对方案进行了安全证明，并进行了性能分析与实验验证，实验结果表明，与已有相关方案相比，虽然为了实现访问控制策略隐藏并且解决密钥托管问题增加了额外的计算负载，但是由于该文将大部分解密工作授权给云存储中心来执行，因此数据访问者的计算负载较小。

关键词：属性加密；控制策略隐藏；密钥托管；外包解密
中图分类号： TP393.08 **文献标识码**：A **文章编号**：1009-5896(2016)11-2931-09
DOI: 10.11999/JEIT160064

Research on Cloud Storage Scheme with Attribute-based Encryption

WANG Guangbo① WANG Jianhua②
①(*PLA Information Engineering University*, *Zhengzhou* 450001, *China*)
②(*Electronic Technology Institute of Air Force*, *Beijing* 100195, *China*)

Abstract: Attribute-Based Encryption (ABE) is often used in cloud storage to achieve fine-grained access control. In order to further protect the sensitive information of access control policy and solve the key escrow caused by the authority center generating the private key for users alone. In this paper, the attributes of access control policy are remapped to achieve its privacy. Additionally, a two-party computing protocol in which the user generates partial private key component is devised to solve the problem of key escrow. At last, the security of this scheme is proved in the standard model, and the performance analysis and experiment validation are conducted, which show that although some additional computation overhead is added for achieving the privacy of access control policy and solving the problem of key escrow, the receiver in proposed scheme has smaller computation overhead compared with the existing related schemes because most of the decryption is delegated to the storage center to carry out.

Key words: Attribute-Based Encryption (ABE); Access control policy hidden; Key escrow; Outsource decryption

1 引言

随着大数据时代的到来，出现越来越多的用户数据，为了实现数据共享的同时降低成本，使用第三方的服务提供商成为外包数据的优先选择。云存储作为云计算的延伸与发展，其最大特点是存储即服务，用户可以在任何地点、任何时间，通过任何可连网设备方便地存取数据，因此得到了越来越广泛的应用。但云存储中用户数据存储在云服务器上，脱离了用户的实际控制，因此如何保证用户隐私和数据安全的同时尽可能地提高服务质量已经成为安全云存储的关键问题。

Sahai 等人在 2005 年提出了属性加密[1](Attribute-Based Encryption, ABE)的概念，将密文与密钥与一系列的属性相关联，通过定义访问结构，指定能够解密数据的属性集合，实现细粒度的访问控制，属性加密方案凭借其灵活的访问结构在云存储中得到了广泛的应用。最初的 ABE 只能实现门限操作，策略表达不够丰富。因此，有学者提出了基于密文策略[2–4](Ciphertext-Policy, CP)和密钥策略[5,6](Key-Policy, KP)的 ABE 机制，实现丰富的属性操作，从而支持灵活的访问控制策略。

收稿日期：2016-01-15；改回日期：2016-08-15；网络出版：2016-10-09
*通信作者：王光波 691759571@qq.com
基金项目：国家高技术研究发展计划(2012AA012704)，郑州市科技领军人才项目(131PLJRC644)
Foundation Items: The National High-tech R&D Program of China (2012AA012704), The Science and Technology Leading Talent Project of Zhengzhou (131PLJRC644)

然而，除了敏感数据需要保护外，访问控制策略也非常重要，它能够泄露用户的敏感信息。因此实际的应用中不仅要保护用户数据的安全性也要保证访问控制策略的隐私性。另外基本的 ABE 应用中，授权中心负责为用户生成属性相关的密钥，因此拥有所有用户密钥，产生密钥托管问题。

Kapadia 等人[7]提出了一种策略隐藏的 CP-ABE 方案，但是引入了一个在线的半可信服务器，为每个用户重加密密文，因此使服务器成为整个系统瓶颈。Nishide 等人[8]提出了 2 种实现策略隐藏的 CP-ABE 方案，通过多值属性之间的与逻辑来表示访问控制策略。Lai 等人[9]基于子群判定性假设，在合数阶双线性群上提出了一种适应性安全的策略隐藏 CP-ABE 方案。随后 Wang 等人[10]对文献[9]进行了改进，提出一种素数阶双线性群上的策略隐藏 CP-ABE 方案，使私钥规模与解密运算量成为常量，在大规模属性应用环境中具有很高的效率。但这几种方案只支持受限的访问结构，策略由所有的属性通过一个“与”门组合而成，可表达性非常弱。Hur 等人[11]等提出了一个支持任意访问机构的 CP-ABE 方案，对密文中的访问策略进行盲化，实现了访问策略隐私。而且该方案由云存储中心代为解密，只需要 1 个对操作，大大提高了访问者的解密效率。但是该方案被证明是通用群模型下安全的，通用群模型安全通常被认为是启发式的安全，而不是可证明安全。最近，宋衍等人[12]在文献[13]的基础上，对访问树进行相应的改进，通过秘密共享在“与”、“或”、“门限”中的应用，将具有权限的属性取值隐藏在系统所有的属性取值中，从而实现基于访问树的策略隐藏。

Chase 等人[14]提出了一种分布式的 ABE 方案，使用多属性授权中心模型来解决密钥托管问题。在该方法中，众多属性管理机构都参与到密钥生成的过程中。这种方式的缺点是系统的性能会随着属性数量的增长而不断退化。另外其访问结构的表达能力有限，仅仅支持与门，从而限制了数据拥有者对访问策略的制定。Yang 等人[15]提出了一个有效的 CP-ABE 方案，采用了一个新的访问控制策略来解决密钥托管问题。用户只有同时拥有来自属性授权与数据拥有者的密钥组件才能得到完整的密钥从而解密密文。然而，新的访问策略包含了用户的身份，这意味着当有新加入的用户时，数据拥有者必须保持在线状态来重新加密数据。这在实际应用时不太可行。

Liu 等人[16]通过借鉴文献[17]的签名方案，构造了一个可追踪的 CP-ABE，不仅支持任意单调访问机构，而且被证明是标准模型安全的，方案最后对密钥托管问题进行了研究，包含两个不同的机构：授权中心和追踪机构，授权中心负责为用户生成密钥，追踪机构则负责追踪，但是这种方法不能抵抗授权中心与追踪机构的合谋攻击，另外该方案也没有解决策略隐藏问题。

本文在方案[16]的基础上，通过构造用户的盲化名与盲因子，对密文中的访问控制策略进行盲化，实现策略隐私，同时在密钥生成算法中，使用双方计算协议，由用户选择进行追踪的身份标识，实现追踪的同时，解决密钥托管问题。

2 相关技术

在方案提出之前，首先对文中将用到的相关技术进行简单介绍，包括线性秘密分享方案、合数阶双线性群。

2.1 线性秘密分享方案

定义 1 (线性秘密分享方案(Linear Secret-Sharing Scheme, LSSS)[18])参与者集合 P 上的一个秘密分享方案 $\prod$ 如果满足以下条件，则称为(Z_p 上的)线性秘密分享方案：

(1)每个实体的秘密份额构成 Z_p 上的一个向量。

(2)对于每个秘密分享方案 $\prod$，存在一个生成矩阵 $\boldsymbol{A}(m\times n)$，对于矩阵 $\boldsymbol{A}$ 中的每一行 $i=1,2,\cdots,m$，映射 $\rho:\{1,2,\cdots,m\}\to P$ 把 $\boldsymbol{A}$ 的每一行映射到参与者集合 P 中。考虑向量 $\boldsymbol{v}=(s,r_2,\cdots,r_n)$，其中 $s\in Z_p$ 是共享密钥，$r_2,r_3,\cdots,r_n$ 随机选择用来隐藏 s，$\boldsymbol{Av}$ 是 m 个秘密份额形成的向量，令 $\lambda_i=(\boldsymbol{Av})_i$ 表示参与者 $\rho(i)$ 所持有的秘密份额。

2.2 合数阶双线性群

合数阶双线性群由 Boneh 等人[17]提出，之后被广泛应用到各种密码系统中。令 ψ 是一个群生成算法，以安全参数 λ 为输入，输出 (p_1,p_2,p_3,G,G_T,e)。其中 p_1,p_2,p_3 是 3 个不同的素数，其大小由安全参数 λ 决定，G 和 G_T 是两个 $N=p_1p_2p_3$ 阶循环群，$e:G\times G\to G_T$ 是一个满足下面条件的映射：

(1) 双线性：$\forall g,h\in G,a,b\in Z_N,e(g^a,h^b)=e(g,h)^{ab}$。

(2)非退化性：$\exists g\in G$ 使得 $e(g,g)$ 在 G_T 中的阶是 N。

3 属性加密方案

本节首先定义一个标准语义安全的安全游戏，即选择明文攻击下的密文不可区分性(ciphertext INDistinguishability under Chosen Plaintext Attacks, IND-CPA)[19]，接着对本文提出的属性加密方案进行构造。

3.1 安全性定义

初始化阶段：挑战者运行初始化 Setup 算法，并把产生的系统公开密钥参数 PK 传递给攻击者。

查询阶段 1：攻击者适应性的向挑战者提交一系列的身份-属性对集合，$(\mathrm{id}_1, S_1), \cdots, \left(\mathrm{id}_{q_1}, S_{q_1}\right)$，对于每个集合挑战者与攻击者进行密钥生成协议。协议过程中，由攻击者选择用于进行身份追踪的标识 c。

挑战阶段：攻击者提交一个身份 ID_j、长度相等的两个消息 M_1, M_2 和一个访问控制策略 $(\boldsymbol{A}, \rho')$ 给挑战者。挑战者随机选择 $\beta \in \{0,1\}$，并用矩阵 $\boldsymbol{A}$ 加密消息 M_β，产生密文 CT^*，然后挑战者盲化 $(\boldsymbol{A}, \rho')$ 生成新的访问控制策略 $(\boldsymbol{A}, \rho)$，并把 $(\boldsymbol{A}, \rho)$ 加进密文 CT^* 一同发送给攻击者。

查询阶段 2：如查询阶段 1，攻击者适应性地向挑战者提交一系列的身份-属性对集合 $\left(\mathrm{id}_{q_1+1}, S_{q_1+1}\right), \cdots, \left(\mathrm{id}_q, S_q\right)$。

猜测阶段：攻击者输出一个值 $\beta' \in \{0,1\}$ 作为对 β 的猜测。在“属性结合 $S_1, S_2, \cdots, S_q$ 都不满足访问控制策略 $(\boldsymbol{A}, \rho)$”的条件下，如果 $\beta' = \beta$，我们称攻击者赢得了该游戏。攻击者在该游戏中的优势定义为 $|\Pr[\beta' = \beta] - 1/2|$。

3.2 方案构造

3.2.1 系统初始化　系统初始化阶段，授权中心与用户生成各自的相关参数。

(1)授权中心初始化算法：$\mathrm{Setup}(\lambda, U) \to (\mathrm{PK}, \mathrm{MSK}, T)$。

授权中心以安全参数 λ 和属性集合 U 作为输入，运行群生成函数 ψ 获得系统参数 $(G, G_T, p_1, p_2, p_3, e)$，其中 p_1, p_2, p_3 是 3 个不同的素数，G 和 G_T 是 $N = p_1 p_2 p_3$ 阶循环群，$e: G \times G \to G_T$ 是一个双线性映射。令 G_{p_i} 表示 G 的 p_i 阶子群。$g \in G_{p_1}$，$X_3 \in G_{p_3}$ 分别是 G_{p_1} 和 G_{p_3} 的生成元。算法随机选择 α, $a \in Z_N$ 和 $h \in G_{p_1}$，并且对于每一个属性 $i \in U$，随机选择参数 $u_i \in Z_N$。系统公开密钥参数 PK 设置为：$\mathrm{PK}=(N, h, g, g^a, e(g,g)^\alpha, \{U_i = g^{u_i}\}_{i \in U})$，主密钥 MSK 设置为：$\mathrm{MSK} = (\alpha, a, X_3)$，身份追踪列表 T 初始化为空。

(2)用户盲因子生成算法：$\mathrm{BlindPolicy}(\mathrm{MSK}, \mathrm{ID}_i) \to \mathrm{BL}_i$　授权中心以主密钥 MSK 和用户身份 ID_i 作为输入，与用户进行隐私交互协议，产生对应于用户 ID_i 的盲因子 BL_i，BL_i 既可以实现用户的身份隐私，又可以用于盲化访问控制策略。该协议过程如下：

(a)拥有身份标识 ID_i 的用户计算 $\mathrm{BL}_i^{'} = h^{\mathrm{ID}_i}$ 并将其发送给授权中心。

(b)用户与授权中心关于 $\mathrm{BL}_i^{'}$ 进行交互式零知识证明。

(c)授权中心计算盲因子 $\mathrm{BL}_i = (\mathrm{BL}_i^{'})^a$ 并将其发送给用户。

(3)盲化名生成算法：$\mathrm{PseudoGen}(\mathrm{PK}, \mathrm{ID}_i) \to \mathrm{PS}_i$　算法在数据外包前由数据拥有者执行，以系统公开参数 PK 和数据拥有者的 ID_i 为输入，然后随机选择 $y \in Z_p$ 并计算 $\mathrm{PS}_i = (h^{\mathrm{ID}_i})^y$，并将其公开作为自己的盲化名，$\mathrm{PS}_i$ 可以用于实现访问策略的盲化。

3.2.2 密钥生成　密钥生成算法：$\mathrm{KeyGen}(\mathrm{PK}, \mathrm{MSK}, \mathrm{ID}_i, S, T) \to (\mathrm{SK}_{\mathrm{ID}_i, S}, T')$。

算法以系统公开密钥 PK、主密钥 MSK、用户身份 ID_i、用户属性集合 S 和追踪列表 T 为输入，生成相应的用户密钥 $\mathrm{SK}_{\mathrm{ID}_i, S}$ 和更新后的追踪列表 T'。该算法使用双方交互协议产生密钥中的组件，实现可追踪的同时防止授权中心完全掌握所有用户的密钥，解决密钥托管问题。

令 KeyGen, Enc 和 Dec 表示加法同态且语义安全的加密方案，“$\oplus$”表示对密文进行同态加法操作；令 e 代表一个密文，r 为一个整数，$e \otimes r$ 表示对 e 进行 r 次自“加”操作。授权中心以主密钥 MSK 中的 a 为输入，用户则随机选择追踪标识 $c \in Z_N^*$，然后双方通过运行以下交互协议生成用户密钥。

(1)授权中心运行 $(\mathrm{sk}_{\mathrm{hom}}, \mathrm{pk}_{\mathrm{hom}}) \leftarrow \mathrm{KeyGen}(1^\lambda)$，产生系统参数 $\mathrm{sk}_{\mathrm{hom}}$ 和 $\mathrm{pk}_{\mathrm{hom}}$，然后计算 $e_1 = \mathrm{Enc}(\mathrm{pk}_{\mathrm{hom}}, a)$，并将 e_1 和 $\mathrm{pk}_{\mathrm{hom}}$ 发送给用户，并与其进行交互式的零知识证明，证明 e_1 是对 $[0, N]$ 中消息的加密。

(2)用户随机选择用来进行追踪的身份标识 $c \in Z_N^*$，并计算 $e_2 = g^c$ 后发送给授权中心，并对其进行零知识证明。如果 e_2 已经出现在追踪列表 T 中，则协议终止，用户重新选择身份标识 c。否则将 (ID_i, e_2) 放入 T。

(3)用户随机选择 $r_1 \leftarrow Z_N^*$ 和 $r_2 \leftarrow \{0,1,\cdots, 2^\lambda N\}$ 后计算 $e_3 = ((e_1 \oplus \mathrm{Enc}(\mathrm{pk}_{\mathrm{hom}}, c)) \otimes r_1) \oplus \mathrm{Enc}(\mathrm{pk}_{\mathrm{hom}}, r_2 N)$，并将 e_3 发送给授权中心。

(4)授权中心与用户进行交互式的零知识证明，证明 e_3 的正确性，并且证明 r_1 和 r_2 在正确的区间内。

(5)授权中心对 e_3 进行解密：$x = \mathrm{Dec}(\mathrm{sk}_{\mathrm{hom}}, e_3)$，解密后随机选择 $t \in Z_N$，$R, R_0, R_0^{'} \in G_{p_3}$，$\left\{R_i \in G_{p_3}\right\}_{i \in S}$，并计算用户密钥如下：

$$\mathrm{SK}_{\mathrm{ID}_i, S}^{''} = \Big(K' = (g^{1/x})^\alpha h^t R = g^{\alpha/((a+c)r_1)} h^t R,\ L' = g^t R_0, L_0^{'} = g^{at} R_0^{'},\ \left\{K_{i,1}^{'} = U_i^{xt} = U_i^{(a+c)r_1 t} R_i, K_{i,2}^{'} = U_i^a\right\}_{i \in S}\Big)$$

并将$\mathrm{SK}''_{\mathrm{ID}_i,S}$发送给用户。

(6) 用户收到$\mathrm{SK}''_{\mathrm{ID}_i,S}$后将追踪标记$c$加入$\mathrm{SK}''_{\mathrm{ID}_i,S}$并计算：$\mathrm{SK}'_{\mathrm{ID}_i,S}=\left(K=K',K_0=c,L=L',L_0=L'_0,\left\{K_{i,1}=(K'_{i,1})^{1/r_1}=U_i^{(a+c)t}R_i^{1/r_1},K_{i,2}=K'_{i,2}\right\}_{i\in S}\right)$。

最后将密钥设置为$\mathrm{SK}_{\mathrm{ID}_i,S}=\left(r_1,\mathrm{SK}'_{\mathrm{ID}_i,S}\right)$，其中密钥$\mathrm{SK}'_{\mathrm{ID}_i,S}$可以发送给云存储中心，由云存储中心进行部分解密，然后再使用r_1进行最后的解密，这样可以大大用户的计算负载。

3.2.3 数据加密 当用户ID_j想要将数据M放到云存储中心时，他首先定义访问控制策略$(\boldsymbol{A},\rho')$，其中$\boldsymbol{A}$是一个$m\times n$矩阵，映射函数ρ'把$\boldsymbol{A}$的每行$\boldsymbol{A}_i$映射到一个属性$\rho'(i)$。如文献[20]要求ρ'不会把两个不同的行映射到同一个属性。然后运行算法$\mathrm{Encrypt}(\mathrm{PK},M,(\boldsymbol{A},\rho'))$对$M$进行加密。

(1)加密算法：$\mathrm{Encrypt}(\mathrm{PK},M,(\boldsymbol{A},\rho'))\to \mathrm{CT}'$

算法以系统公开密钥PK、明文消息M和访问控制策略$(\boldsymbol{A},\rho')$为输入，随机选择一个向量$\boldsymbol{v}=(s,v_2,\cdots,v_n)\in Z_p^n$，对$\boldsymbol{A}$的每一行$\boldsymbol{A}_i$，计算内积$\lambda_i=\boldsymbol{A}_i\cdot\boldsymbol{v}$，并随机选择$r_i\in Z_p$，算法输出密文：

$$\mathrm{CT}'=\left((\boldsymbol{A},\rho'),C=M\cdot e(g,g)^{\alpha s},C_0=g^s,C'_0=g^{as},\left\{C_{i,1}=h^{\lambda_i}U_{\rho(i)}^{-r_i},C_{i,2}=g^{r_i}\right\}_{i=1}^{m}\right)$$

接下来用户ID_j对CT'中的访问控制策略$(\boldsymbol{A},\rho')$进行盲化以实现其隐私性。

(2)盲化算法：$\mathrm{BlindAtt}(\mathrm{BL}_j,(\boldsymbol{A},\rho'),\mathrm{PK})\to(\boldsymbol{A},\rho)$

算法以用户的盲因子BL_j、访问控制策略$(\boldsymbol{A},\rho')$和系统公开参数PK为输入，输出盲化后的访问控制策略$(\boldsymbol{A},\rho)$。计算如下：

对于所有的属性值$\rho'(i)$，用户ID_j计算$s_{\rho'(i)}=e\left(\left(\mathrm{BL}_j\right)^y,U_{\rho'(i)}\right)=e\left(\left(h^{\mathrm{ID}_j}\right)^{ay},U_{\rho'(i)}\right)$进行盲化，即将访问控制矩阵中的每行从映射到具体的属性进行重新映射到盲化后的属性。需要注意的是，对操作可以提前计算一次并且以后继续使用。用ρ表示盲化后的映射函数，然后更新密文为$\mathrm{CT}=((\boldsymbol{A},\rho),C,C_0,C'_0,\{C_{i,1},C_{i,2}\}_{i=1}^m)$。密文生成后，用户将数据$(\mathrm{ID}_j,\mathrm{PS}_j,\mathrm{CT})$发送到云存储中心。

3.2.4 令牌生成 当拥有属性集合S的用户ID_i需要访问云存储中用户ID_j的数据时，用户首先从存储中心获得相应的盲化名PS_j，然后通过运行算法$\mathrm{GenToken}\left(S,\mathrm{SK}_{\mathrm{ID}_i,S},\mathrm{PS}_j\right)$产生访问令牌。

令牌生成算法：$\mathrm{GenToken}\left(S,\mathrm{SK}_{\mathrm{ID}_i,S},\mathrm{PS}_j\right)\to\mathrm{TK}_{\mathrm{ID}_i,S}$。

算法以用户ID_i的属性集合S、密钥$\mathrm{SK}_{\mathrm{ID}_i,S}=\left(r_1,\mathrm{SK}'_{\mathrm{ID}_i,S}\right)$和用户$\mathrm{ID}_j$的盲化名$\mathrm{PS}_j$为输入，输出访问令牌$\mathrm{TK}_{\mathrm{ID}_i,S}$如下：

对于所有的$i\in S$，计算$s_i=e\left(\mathrm{PS}_j,K_{i,2}\right)=e\left(\left(h^{\mathrm{ID}_j}\right)^y,U_i^a\right)$，然后用户$\mathrm{ID}_i$构造访问令牌如下：$\mathrm{TK}_{\mathrm{ID}_i,S}=\left(I=\{s_i\}_{i\in S},\mathrm{SK}'_{\mathrm{ID}_i,S}\right)$。

令牌生成后，用户将其发送给云存储中心，请求云存储中心进行部分解密的密文PCT。令牌中的集合I用来作为盲化后属性的索引。虽然生成令牌需要进行$|S|$个对操作，但是这一计算可以在发送查询请求之前进行，并且计算一次以后可以继续使用。

3.2.5 部分解密 云存储中心一旦收到用户ID_i对用户ID_j密文的查询请求，就会检查索引集合I中的属性是否满足密文中的访问控制策略。检查只需进行简单的对比操作，不会泄露密文和令牌中的属性。如果索引集合I中的属性满足访问控制策略，云存储中心则运行算法$\mathrm{PDecrypt}\left(\mathrm{CT},\mathrm{TK}_{\mathrm{ID}_i,S}\right)$将密文进行部分解密，然后将生成的密文发送给用户$\mathrm{ID}_i$。这样可以将大部分复杂的解密操作授权给计算能力更强的云存储中心执行，大大降低数据访问者的计算负载。

部分解密算法：$\mathrm{PDecrypt}\left(\mathrm{CT},\mathrm{TK}_{\mathrm{ID}_i,S}\right)\to\mathrm{PCT}$：

算法以用户ID_j的密文CT和用户ID_i的访问令牌$\mathrm{TK}_{\mathrm{ID}_i,S}$为输入，输出部分解密密文PCT。算法执行之前将实施一个简单的比较来确定索引集合I是否满足密文中的访问控制策略$(\boldsymbol{A},\rho)$：将$(\boldsymbol{A},\rho)$中的盲化属性集合$\{\rho(i)\}_{i=1}^m$与令牌中的索引集合I进行比较，找出同时属于二者的属性集合$V=\{i:\rho(i)\in I\}$。如果V中相应行的线性组合都不等于向量$[1,0,0,\cdots,0]$，那么算法输出$\perp$。否则计算$\{w_i\in Z_p\}_{i\in V}$使以下等式成立：$\sum_{i\in V}w_i\boldsymbol{A}_i=(1,0,\cdots,0)$。然后计算：

$$D=\prod_{i\in V}\left(e\left(L^{K_0}L_0,C_{i,1}\right)\cdot e\left(K_{\rho(i),1},C_{i,2}\right)\right)^{w_i}=\prod_{i\in V}e(g,h)^{(a+c)tw_i\lambda_i}=e(g,h)^{(a+c)ts}$$

$$E=e\left(K,C_0^{K_0}C'_0\right)=e\left(g^{\alpha/((a+c)r_1)}h^tR,g^{(a+c)s}\right)=e\,(g,g)^{\alpha s/r_1}\,e(g,h)^{(a+c)ts}$$

最后计算$B=D/E=e\,(g,g)^{\alpha s/r_1}$后，存储中心将部分解密密文$\mathrm{PCT}=(B,C)$发送给用户$\mathrm{ID}_i$。

3.2.6 数据解密 解密算法：$\mathrm{Decrypt}\left(\mathrm{PCT},\mathrm{SK}_{\mathrm{ID}_i,S}\right)\to M$。

算法以部分解密密文 PCT 和用户密钥 $\mathrm{SK}_{\mathrm{ID}_i,S}$ 为输入，输出明文消息 M。当用户 ID_i 收到部分解密密文 PCT 后，运行解密算法计算最终明文消息如下：$C/B^{r_1}=M\cdot e(g,g)^{as}/(e(g,g)^{as/r_1})^{r_1}=M$。

因为云存储中心对密文进行了大部分解密工作，访问者只需要进行一个简单的指数操作，大大降低了访问者的计算负载。

4　安全性证明

4.1 IND-CPA 安全性

本文提出的属性加密方案(记为 $\Sigma_{\mathrm{ntwcpabe}}$)的安全性可以用文献[20]中相似的证明方法来证明。因此我们把 $\Sigma_{\mathrm{ntwcpabe}}$ 的安全性规约到文献[20]中属性加密方案(记为 Σ_{cpabe})的安全性。由下面的引理 1 来证明方案的安全性。

引理 1　如果方案 Σ_{cpabe} 在文献[20]中第 2.1.1 节中的安全游戏是安全的，那么 $\Sigma_{\mathrm{ntwcpabe}}$ 在本文中的安全游戏也是安全的。

证明　假设存在一个多项式时间的攻击者 A 能够以优势 $\mathrm{Adv}_A\Sigma_{\mathrm{ntwcpabe}}$ 攻破方案 $\Sigma_{\mathrm{ntwcpabe}}$，那么可以构造一个多项式时间算法 B 以优势 $\mathrm{Adv}_B\Sigma_{\mathrm{cpabe}}$ 攻破方案 Σ_{cpabe}，且 $\mathrm{Adv}_B\Sigma_{\mathrm{cpabe}}$ 等于 $\mathrm{Adv}_A\Sigma_{\mathrm{ntwcpabe}}$。

初始化阶段：Σ_{cpabe} 发送给 B 公开参数 $\mathrm{PK}'=(N,X_3,g,g^{\beta},e(g,g)^{\alpha},\{U_i=g^{u_i}\}_{i\in U})$，B 随机选择参数 $a\in Z_N$，然后将公开参数 $\mathrm{PK}=(N,X_3,g,h=g^{\beta},g^{a},e(g,g)^{\alpha},\{U_i=g^{u_i}\}_{i\in U})$ 发送给 A。并初始化追踪列表 $T=\Phi$。

查询阶段 1：当 A 向 B 提交 (id,S) 查询密钥时，B 将 (id,S) 提交给 Σ_{cpabe}，并得到如下形式的解密密钥：

$$\overline{SK}_{\mathrm{id},S}=\left(\overline{K}=g^{\alpha}g^{\beta t'}R,\overline{L}=g^{t'}R',\{\overline{K}_i=U_i^{t'}R_i\}_{i\in S}\right)$$

然后 B 与 A 进行如 3.2.2 节所示的密钥生成协议，在协议中 A 随机选择 $c\in Z_N^*$ 和 $r_1\in Z_N^*$，协议执行后 B 得到值 $x=(a+c)r_1$，且已经将对应的 (g^c,id) 放入追踪列表 T。B 利用 X_3 随机选择 $R''\in G_{p_3}$，并计算：

$$\begin{aligned}\mathrm{SK}'_{\mathrm{id},S}=&\Bigg(K'=(\overline{K})^{1/x}=(g^{\alpha}g^{\beta t'}R)^{1/x}=(g^{\alpha}g^{\beta t'}R)^{\frac{1}{(a+c)r_1}}\\&=g^{\frac{\alpha}{(a+c)r_1}}h^{\frac{t'}{(a+c)r_1}}R^{\frac{1}{(a+c)r_1}},L'=(\overline{L})^{1/x}=(g^{t'}R')^{\frac{1}{(a+c)r_1}}\\&=g^{\frac{t'}{(a+c)r_1}}R'^{\frac{1}{(a+c)r_1}},L'_0=(\overline{L})^{a/x}R''=(g^{t'}R')^{\frac{a}{(a+c)r_1}}R''\\&=g^{\frac{at'}{(a+c)r_1}}R'^{\frac{a}{(a+c)r_1}}R'',\left\{K'_{i,1}=\overline{K}_i=U_i^{t'}R_i,\right.\\&\left.\quad K'_{i,2}=(U_i)^{a}\right\}_{i\in S}\Bigg)\end{aligned}$$

然后 B 将 $\mathrm{SK}'_{\mathrm{id},S}$ 作为解密密钥发送给 A，需要注意的是：R'' 使 L'_0 的 G_{p_3} 部分与 L' 的 G_{p_3} 部分无关，因此需要对方案 Σ_{cpabe} 进行修改：B 需要使用 X_3，因此需要公开参数中获得 X_3。

A 获得解密密钥 $\mathrm{SK}'_{\mathrm{id},S}=(K',L',L'_0,\{K'_{i,1},K'_{i,2}\}_{i\in S})$ 后，隐式地令 $t=t'/((a+c)r_1)$，然后计算解密密钥 $\mathrm{SK}_{\mathrm{id},S}$ 如下：

$$\begin{aligned}\mathrm{SK}_{\mathrm{id},S}=&\Bigg(K=K'=g^{\frac{\alpha}{(a+c)r_1}}h^{\frac{t'}{(a+c)r_1}}R^{\frac{1}{(a+c)r_1}}\\&=g^{\frac{\alpha}{(a+c)r_1}}h^{t}R^{\frac{1}{(a+c)r_1}},K_0=c,\ L=L'\\&=g^{\frac{t'}{(a+c)r_1}}R'^{\frac{1}{(a+c)r_1}}=g^{t}R'^{\frac{1}{(a+c)r_1}},\ L_0=L'_0\\&=g^{\frac{at'}{(a+c)r_1}}R'^{\frac{a}{(a+c)r_1}}R''=g^{at}R'^{\frac{a}{(a+c)r_1}}R'',\\&\quad\left\{K_{i,1}=\left(K'_{i,1}\right)^{1/r_1}=\left(U_i^{t'}R_i\right)^{1/r_1}\right.\\&\left.=\left(U_i^{(a+c)r_1t}R_i\right)^{1/r_1}=U_i^{(a+c)t}R_i^{1/r_1},K'_i=(U_i)^{a}\right\}_{i\in S}\Bigg)\end{aligned}$$

挑战阶段：A 向 B 提交一个身份 ID_j、一个访问控制策略 $(\boldsymbol{A},\rho)$ 和两个长度相等的消息 M_0 和 M_1。然后 B 向 Σ_{cpabe} 提交 $((\boldsymbol{A},\rho),M_0,M_1)$，并得到如下形式的挑战密文：

$$\overline{\mathrm{CT}}=\left((\boldsymbol{A},\rho),\overline{C}=M_b\cdot e(g,g)^{\alpha s},\overline{C}_0=g^{s},\left\{\overline{C}_{i,1}=g^{\beta\lambda_i}U_{\rho(i)}^{-r_i},\overline{C}_{i,2}=g^{r_i}\right\}\right)$$

得到 $\overline{\mathrm{CT}}$ 后 B 计算：

$$\mathrm{CT}'=\left(C=\overline{C},C_0=\overline{C}_0,C'_0=\overline{C}_0^{a}=g^{as},\left\{C_{i,1}=\overline{C}_{i,1},C_{i,2}=\overline{C}_{i,2}\right\}_{i=1}^{m}\right)$$

接着盲化密文中的访问控制策略 $(\boldsymbol{A},\rho)$，即对于每个 $\{\rho(i)\}_{i=1}^{m}$ 中的属性，B 随机选择参数 $y\in Z_p$ 并计算 $s_{\rho(i)}=e\left(\left(h^{\mathrm{ID}_j}\right)^{ay},U_{\rho(i)}\right),\mathrm{PS}_j=\left(h^{\mathrm{ID}_j}\right)^{y}$。

然后将访问控制矩阵中的每一行重新映射到 $\{s_{\rho(i)}\}_{i=1}^{m}$ 中的向量，获得新的映射函数 ρ'，用新的访问控制策略 $(\boldsymbol{A},\rho')$ 代替 $(\boldsymbol{A},\rho)$。因此新生成的密文为：$\mathrm{CT}=((\boldsymbol{A},\rho'),C,C_0,C'_0,\{C_{i,1},C_{i,2}\}_{i=1}^{m})$，最后将 $(\mathrm{ID}_j,\mathrm{PS}_j,\mathrm{CT})$ 发送给 A。

查询阶段 2：与查询阶段 1 一样。

猜测阶段：A 给 B 一个 $b'\in\{0,1\}$ 作为对 b 的猜测，然后 B 把 b' 发送给 Σ_{cpabe} 作为输出。

由于公开参数、解密密钥和挑战密文与文献[20]中的方案都具有一样的分布，因此得出 $\mathrm{Adv}_B\Sigma_{\mathrm{cpabe}}=\mathrm{Adv}_A\Sigma_{\mathrm{ntwcpabe}}$。　证毕

4.2 密钥生成协议的安全性

3.2.2 节的密钥生成协议中，由用户产生用于追踪的身份标识信息 c，对授权中心是隐藏的，因此解决了密钥托管问题。下面对该密钥生成协议进行安全性证明。

首先对恶意用户进行模拟，按照零知识证明的过程，定义一个仿真器 S，该仿真器以系统参数、追踪身份信息 c 的承诺 C 和在理想功能下的密钥 $(K,L,L_0,\{K_{i,1},K_{i,2}\})$ 为输入，因此仿真器 S 必须在不知道系统主密钥 MSK 的情况下，模拟一个诚实授权中心的功能。运行以下仿真过程：

(1) S 诚实地产生密钥对 $(\mathrm{sk}_{\mathrm{hom}},\mathrm{pk}_{\mathrm{hom}})\leftarrow \mathrm{KeyGen}(1^{\lambda})$，然后计算 $e_1=\mathrm{Enc}(\mathrm{pk}_{\mathrm{hom}},0)$ 并将 $\mathrm{pk}_{\mathrm{hom}}$，$e_1$ 发送给恶意用户；

(2) S 从恶意用户处得到 e_2；

(3) S 作为验证者验证 e_2 计算的正确性。S 运行知识抽取算法得到 r_1。最后计算 $(K^{1/r_1},L^{1/r_1},L'^{1/r_1},\{K_i^{1/r_1},K_i'\})$ 并将其发送给恶意用户。

接下来对恶意授权中心进行模拟。同样定义一个仿真器 S，该仿真器以系统参数、授权中心的公开密钥 PK 和追踪身份信息 c 的承诺 C 作为输入。仿真器 S 必须能够在不知道身份信息 c 的情况下来模拟诚实用户。运行以下仿真过程。

(1) S 从恶意授权中心得到 $\mathrm{pk}_{\mathrm{hom}}$ 和 e_1；

(2) S 随机选择 $t\leftarrow[0,2^{\lambda}N^2]$，计算 $e_2=e_1\oplus \mathrm{Enc}(\mathrm{pk}_{\mathrm{hom}},t)$ 并将其发送给恶意授权中心；

(3) S 用零知识证明的仿真器与恶意授权中心进行交互；

(4) S 从恶意授权中心得到如下所示的密钥 $K=(g^{\alpha/(a+t)})h^{r_1t'},L=g^{r_1t'},L'=g^{a\cdot r_1t'},\{Ki=U_i^{(a+t)r_1t'},K_i'=U_i^a\}_{i\in S}$，并验证其是对追踪身份信息 $c=t$ 下的正确密钥。

为了论文的简洁性，我们可以从文献[21]中得到该仿真器 S 成功的对协议进行了仿真。

5 方案分析与实验验证

5.1 方案分析

现将本文提出的方案与原始方案及已有几种策略隐藏的方案进行性能与功能比较，主要考虑群阶的性质、策略隐私性、密钥托管、安全性及密文长度、用户密钥长度、解密运算量。具体比较结果如表 1 所示。表中所使用的描述符号如下：C_0 表示 G 中数据元素的长度；C_1 表示 G_T 中数据元素的长度；C_N 表示 Z_N^* 中数据元素的长度；t 表示与密文有关的属性个数；k 表示用户密钥中属性的个数；u 表示整个属性集合 U 中属性的总个数；u_i 表示属性 U_i 的取值个数；$|\alpha|$ 表示文献[12]中访问结构中末端内部节点的个数。

功能方面：从表 1 可以看出，本文方案中的存储中心通过比较属性盲化后的索引来确定用户提交的查询令牌是否满足密文中的访问控制策略，因此密文与令牌中有关属性的任何信息对存储中心来说都是隐藏的。另外，本文方案在密钥生成算法中，使用双方计算协议，由用户选择追踪身份标识，实现追踪的同时解决了密钥托管问题。最后本文方案在适应性模型下进行了安全证明。而其他方案只实现了其中的某项功能。

性能方面：从表 1 可以看出，本文方案的密文大小与原始 CP-ABE 相同，比文献[11]方案增加了

表 1 本文方案与不同方案之间的比较

方案	群阶	策略隐私	密钥托管	追踪性	安全性	密文	密钥	解密(对)
原始方案[16]	合数	否	是	是	适应性	$(2t+2)C_0+C_1$	$(k+3)C_0+C_N$	$2k+1$
文献[8]方案	素数	是	是	否	选择性	$\left(u+1+\sum_{i=1}^{u}u_i\right)C_0+C_1$	$(2u+1)C_0$	$3u+1$
文献[9]方案	合数	是	是	否	适应性	$\left(1+\sum_{i=1}^{u}u_i\right)C_0+C_1$	$(u+1)C_0$	$u+1$
文献[11]方案	素数	是	是	否	通用群	$(2t+1)C_0+C_1$	$(3k+1)C_0$	1
文献[12]方案	合数	是	是	否	适应性	$\left(1+\lvert\alpha\rvert+\lvert\alpha\rvert\sum_{i=1}^{u}u_i\right)C_0+(1+\lvert\alpha\rvert)C_1$	$(k+2)C_0$	$1+\lvert\alpha\rvert+k\lvert\alpha\rvert$
本文方案	合数	是	否	是	适应性	$(2t+2)C_0+C_1$	$(2k+3)C_0+2C_N$	0

C_0。而其他方案的密文长度均与整个属性集合中的属性总个数有关，这大大增加了密文长度。为了实现策略隐私，本文方案需要在密钥中额外增加 k 个 $K'_{i,2}$ 与 1 个 r_1 参数，因此密钥长度相比原始方案增加了 $(kC_0 + C_N)$。解密过程中，本文方案与文献[11]方案将部分解密操作交由存储中心代为执行，因此文献[11]方案只需要 1 个对操作而本文方案不需要进行对操作，大大提高了访问者的解密效率。

5.2 实验验证

本文通过实验分析执行效率，为了衡量可追踪并解决策略隐藏与密钥托管问题带来的计算开销，本文主要对以下过程进行分析：

(1)数据拥有者加密数据；(2)数据访问者解密数据。需要注意的是，在本文中，为了实现访问控制策略隐藏并且解决密钥托管问题，访问者需要进行额外的计算，因此实验时将其一并考虑，归为功能操作。

实验环境为 64 bit Ubuntu 14.04 操作系统、Intel® Core™ i7-3770CPU (3.4 GHz)、内存 4 G，实验代码基于 Pairing-based Cryptography Library (PBC-0.5.14)[22]与 cpabe-0.11[23]进行修改与编写，并且实现 128 bit 的安全级别。实验数据取运行 20 次所得的平均值。

将本文方案与同样采用合数阶双线性群的原始 CP-ABE 方案[16], 文献[9]方案、文献[12]方案进行实验验证并比较，主要考虑对运算与群 G 与 G_T 中的指数运算，在合数阶双线性群中，运行一次对运算需要的时间大约为 1.26 s, G 中的指数运算大约为 0.53 s, G_T 中的指数运算大约为 0.18 s。其计算时间对比如表 2 所示。

如表 2 所示，相比于原始 CP-ABE 方案，在加密过程中，为了实现密文中的访问控制策略隐藏，本文提出的方案需要数据拥有者额外计算 t 个对操作和 2 个 G 中的指数操作。而在访问者解密数据时，需要 k 个对操作来产生属性隐藏后的信息索引。但是需要主要的是这些额外的对操作与指数操作，可以提前被计算，以后继续使用，因此表 2 列出的是本文方案在最差情况下的计算负载。另外数据访问者需要额外计算(k+1)个 G 中的指数操作来实现密钥托管问题。虽然，为了实现策略隐藏并且解决密钥托管问题，访问者需要计算额外的对操作和指数操作，但是由于本文方案将大部分解密操作授权给存储中心进行，而访问者只需要执行 1 个 G 中的指数操作即可。从表 2 可以得出，在本文方案中访问者的计算时间为 (1.79k+0.53+0.53) ms=(1.79k+1.06) s，而原始 CP-ABE 方案需要(2.52k+2.32) s，文献[9]方案需要(1.26u+1.26) s，文献[12]方案则需要 $(1.26+1.26\,|\,\alpha\,|+1.26k\,|\,\alpha\,|)$ s，而 u 代表的是整个属性集合 U 中属性的总个数，因此本文方案最终访问者解密所需计算时间略多于文献[12]的最优方案($|\,\alpha\,|=1$)，但是小于所有其他方案，包括文献[12]的次优方案($|\,\alpha\,|=2$)。

不失一般性，本文假设用户拥有的属性数目为 5，系统的属性数目在 5~50 之间，与密文有关的属性数目为系统数目的一半，并假设文献[9]方案与文献[12]方案中的属性平均拥有 4 个不同的属性取值，且文献[12]方案访问结构中末端内部节点的个数取最优方案 1 与次优方案 2。

表 2 计算时间对比

操作	时间 (s)	原始 CP-ABE[16]		文献[9]方案		文献[12]方案		本文方案		
		加密	解密	加密	解密	加密	解密	加密	功能	解密
Pairing	1.26	0	$2k+1$	0	$u+1$	0	$1+\|\alpha\|+k\|\alpha\|$	t	k	0
Exp.in$G$	0.53	3t+2	2	$1+\sum_{i=1}^{u}u_i$	0	$1+\|\alpha\|+\|\alpha\|\sum_{i=1}^{u}u_i$	0	3t+4	k+1	1
Exp.in$G$	0.18	1	0	1	0	$1+\|\alpha\|$	0	1	0	0
计算时间		$1.59t+1.24$	$2.52k+2.32$	$0.53\sum_{i=1}^{u}u_i+0.71$	$1.26u+1.26$	$0.71+0.71\|\alpha\|+0.53\|\alpha\|\sum_{i=1}^{u}u_i$	$1.26+1.26\|\alpha\|+1.26k\|\alpha\|$	$2.85t+2.3$	$1.79k+0.53$	0.53

图 1 给出了 5 种方案在不同系统属性数目下的加密耗时对比，由此可见，本文为了实现密文中的访问控制策略隐藏，需要额外计算 t 个对操作和 2 个 G 中的指数操作，因此加密耗时要大于原始方案，但是优于其他 3 种方案。特别是文献[12]方案，随着访问结构中末端内部节点个数地增加，加密耗时大大增加。

图 2 给出了 5 种方案在不同系统属性数目下的解密耗时对比， 可见本文方案、原始 CP-ABE 方案与文献[12]方案的解密耗时与系统属性数目无关，而文献[9]方案中，随着系统属性数目的增加，解密耗时不断增加。另外，本文方案为了实现功能操作，一定程度上增加了访问者的解密耗时，但是由于本文将大部分解密操作交由存储中心执行，因此总的解密耗时小于原始 CP-ABE 方案。 而文献[12]方案的解密耗时虽然与系统属性数目无关，但是与访问结构中末端内部节点个数有关，内部节点数越多，解密耗时越大，从图中可以看出，本文方案的解密耗时要略大于文献[12]的最优方案 1，但小于文献[12]的次优方案 2。

6 结束语

本文从用户角度出发，提出了一种访问控制策略隐藏的标准模型安全的属性加密方案，并且在密钥生成算法中设计一个双方计算协议，解决密钥托管问题。本文对方案在标准模型下进行了安全证明，并且进行了性能分析与实验验证，实验结果表明，与已有相关方案相比，虽然为了实现访问控制策略隐藏并且解决密钥托管问题增加了额外的计算负载，但是由于本文将大部分解密工作授权给云存储中心来执行，因此数据访问者的计算负载较小，降低了其成本。

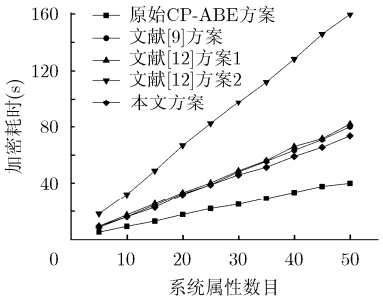

图 1 加密耗时与系统属性数目的关系

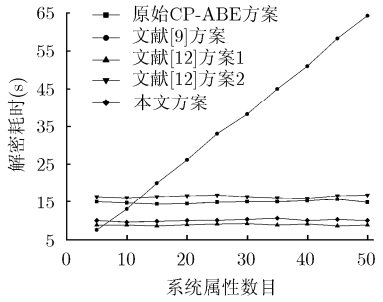

图 2 解密耗时与系统属性数目的关系

参 考 文 献

[1] SAHAI A and WATERS B. Fuzzy Identity-Based Encryption [M]. Heidelberg, Berlin, Springer, 2005: 457–473. doi: 10.1007/11426639_27.

[2] YADAV U C. Ciphertext-policy attribute-based encryption with hiding access structure[C]. 2015 IEEE International Advance Computing Conference (IACC), Bangalore, India, 2015: 6–10. doi: 10.1109/IADCC.2015.7154664.

[3] NARUSE T, MOHRI M, and SHIRAISHI Y. Provably secure attribute-based encryption with attribute revocation and grant function using proxy re-encryption and attribute key for updating[J]. *Human-centric Computing and Information Sciences*, 2015, 5(1): 1–13.

[4] WANG H, YANG B, and WANG Y. Server aided ciphertext-policy attribute-based encryption[C]. IEEE International Conference on Advanced Information Networking & Applications Workshops, Gwangju, Korea, 2015: 440–444. doi: 10.1109/WAINA.2015.11.

[5] QI L, MA J, RUI L, *et al*. Large universe decentralized key-policy attribute-based encryption[J]. *Security & Communication Networks*, 2015, 8(3): 501–509.

[6] WANG X, ZHANG J, SCHOOLER E M, *et al*. Performance evaluation of attribute-based encryption: Toward data privacy in the IoT[C]. IEEE International Conference on Communications (ICC), Sydney, Australia, 2014: 725–730.

[7] KAPADIA A, TSANG P P, and SMITH S W. Attribute-based publishing with hidden credentials and hidden policies [C]. Network and Distributed System Security Symposium, NDSS 2007, San Diego, CA, USA, 2007: 179–192.

[8] NISHIDE T, YONEYAMA K, and OHTA K. Attribute-based Encryption with Partially Hidden Encryptor-specified Access Structures[M]. Heidelberg, Berlin, Springer, 2008: 111–129. doi: 10.1007/978-3-540-68914-0_7.

[9] LAI J, DENG R H, and LI Y. Fully secure cipertext-policy hiding CP-ABE[J]. *Lecture Notes in Computer Science*, 2011, 6672: 24–39.

[10] 王海斌, 陈少真. 隐藏访问结构的基于属性加密方案[J]. 电子与信息学报, 2012, 34(2): 457–461.
WANG Haibin and CHEN Shaozhen. Attribute-based encryption with hidden access structures[J]. *Journal of Electronics & Information Technology*, 2012, 34(2): 457–461.

[11] HUR J. Attribute-based secure data sharing with hidden policies in smart grid[J]. *IEEE Transactions on Parallel & Distributed Systems*, 2013, 24(11): 2171–2180. doi: 10.1109/TPDS.2012.61.

[12] 宋衍, 韩臻, 刘凤梅, 等. 基于访问树的策略隐藏属性加密方案[J]. 通信学报, 2015, 36(9): 119–126.
SONG Yan, HAN Zhen, LIU Fengmei, *et al*. Attribute-based encryption with hidden policies in the access tree[J]. *Journal on Communications*, 2015, 36(9): 119–126.

[13] LUAN Ibraimi, QIANG Tang, PITER Hartel, *et al*. Efficient and Provable Secure Ciphertext-policy Attribute-Based Encryption Schemes. Information Security Practice and Experience[M]. Heidelberg, Berlin, Springer, 2009: 1–12.

[14] CHASE M and CHOW S S M. Improving privacy and security in multi-authority attribute-based encryption[C]. ACM Conference on Computer and Communications Security, Chicago, IL, USA, 2009: 121–130. doi: 10.1145/1653662.1653678.

[15] YANG M, LIU F, HAN J L, *et al*. An efficient attribute based encryption scheme with revocation for outsourced data sharing control[C]. 2011 First International Conference on Instrumentation, Measurement, Computer, Communication and Control, Beijing, China, 2011: 516–520.

[16] LIU Z, CAO Z, and WONG D. Traceable ciphertext-policy attribute-based encryption supporting any monotone access structures[J]. *IEEE Transactions on Information Forensics and Security*, 2013, 8(1): 76–88.

[17] BONEH D and BOYEN X. Short signatures without random oracles[C]. Advances in Cryptology-EUROCRYPT 2004, Switzerland, 2004: 56–73.

[18] ZAVATTONI E, PEREZ L J D, MITSUNARI S, *et al*. Software implementation of an attribute-based encryption scheme[J]. *IEEE Transactions on Computers*, 2015, 64(5): 1429–1441.

[19] CHEUNG L and NEWPORT C. Provably secure ciphertext policy ABE[C]. Proceedings of the 14th ACM Conference on Computer and Communications Security, New York, USA, 2007: 456–465. doi: 10.1145/1315245.1315302.

[20] LEWKO A, OKAMOTO T, SAHAI A, *et al*. Fully Secure Functional Encryption: Attribute-based Encryption and (Hierarchical) Inner Product Encryption[M]. Heidelberg, Berlin, Springer, 2010: 62–91. doi: 10.1007/978-3-642-13190-5_4.

[21] BELENKIY M, CAMENISCH J, CHASE M, *et al*. Randomizable Proofs and Delegatable Anonymous Credentials[M]. Heidelberg, Berlin, Springer, 2009: 108–125. doi: 10.1007/978-3-642-03356-8_7.

[22] LYNN B. The pairing-based cryptography (PBC) library[OL]. http://crypto.stanford.edu/pbc, 2006.

[23] BETHENCOURT J, SAHAI A, and WATERS B. Advanced crypto software collection: The cpabetoolkit[OL]. http://acsc.cs.utexas.edu/cpabe,2011.

王光波： 男，1987 年生，博士生，研究方向为属性加密、网络信息安全.

王建华： 男，1962 年生，教授，博士生导师，研究方向为信息安全、安全管理.