文章编号: 1001-0920 (2015) 09-1630-05 DOI: 10.13195/j.kzyjc.2014.0997

# 一种无可信第三方的密文策略属性加密方案

王静宇[1,2], 涂春岩[2], 谭跃生[2], 郑雪峰[1]

(1. 北京科技大学 计算机与通信工程学院，北京 100083；2. 内蒙古科技大学 信息工程学院，内蒙古 包头 014010)

**摘 要:** 针对现有云环境中密文策略属性加密都依赖于一个可信密钥生成机构的问题, 提出一种无可信第三方的密文策略属性加密方案. 该方案使密钥生成机构在负责用户认证和属性管理时, 必须与另一方 (如云服务提供商) 通过安全双方计算协议来生成密钥, 而它们中的任何一方都没有能力单独解密密文. 安全性分析表明, 所提出的方案能够解决单独密钥生成机构所带来的安全性问题, 而且用户端仅需一次加法运算, 提高了计算效率.

**关键词:** 属性加密；安全双方计算；可信第三方；访问控制

**中图分类号:** TP393 **文献标志码:** A

# Removing trusted third party of ciphertext-policy attribute-based encryption scheme

*WANG Jing-yu*[1,2], *TU Chun-yan*[2], *TAN Yue-sheng*[2], *ZHENG Xue-feng*[1]

(1. School of Computer and Communication Engineering，University of Science and Technology Beijing，Beijing 100083, China；2. School of Information Engineering，Inner Mongolia University of Science and Technology，Baotou 014010, China．Correspondent：WANG Jing-yu，E-mail：btu_wjy@126.com)

**Abstract:** The existing ciphertext-policy attribute-based encryption(CP-ABE) schemes have to rely on a single trusted key generation center in cloud computing. Therefore, a removing trusted third party of CP-ABE scheme is proposed. The key generation center(KGC) is responsible for the user identity authentication and attributes management, which must be with the other party(cloud service providers) through secure two party computation protocol to generate keys, and neither of them has the ability to decrypt the ciphertexts. The security analysis shows that the proposed scheme can solve the security problem caused by the single trusted key generation center, and the end-user only needs once addition operation and the computional efficiency is greatly improved.

**Keywords:** attribute-based encryption; secure two-party computation; trusted third-party; access control

## 0 引 言

Shamir[1]提出了基于身份的加密 (IBE) 方法, 用以解决传统数据共享机制中 PKI 证书管理的难题. 它以用户的身份, 如邮箱、ID 等作为公钥, 数据提供方无需向 PKI 查询公钥证书. Boneh 等[2]在该方案的基础上使用了双线性对, 基于 Diffie-Hellman 假设和随机预言模型提高其安全性, 能够抵抗选择密文攻击. 其后 Sahai 等[3]在 Boneh 等的基础上又提出了基于属性的加密技术 (ABE), 用户被描述成属性的集合, 数据提供方不指定具体的数据接收对象, 仅根据属性利用秘密共享机制来制定策略, 以加密数据, 只要用户拥有的属性满足加密者制定的数量要求就可以解密密文.

为了实现细颗粒度的访问控制, Sahai 等[4]又进一步提出了基于密文策略的属性加密 CP-ABE. 在这个方案中, 每个用户都与一系列属性相关联, 并利用这些属性制定访问控制策略来加密数据; 只有当用户本身具有的属性满足加密者制定的访问策略时才能够解密密文; 访问控制策略以访问结构树实现, 访问树中的叶子节点表示属性, 非叶子节点表示与门、非门或门限值, 极大地提高了访问控制的细粒度; 解密阶段使用一个递归算法来验证用户的属性是否满足加密者制定的属性要求, 同时采用二级掩码来抵抗合谋攻击.

自 CP-ABE 方法提出以来, 研究者已经做了大量的工作, 提出了很多新方案来提高其效率并扩展功能.

**收稿日期:** 2014-06-23；**修回日期:** 2014-09-05.

**基金项目:** 国家自然科学基金项目(61163025, 61462069)；内蒙古自然科学基金项目(2012MS0912).

**作者简介:** 王静宇(1976−), 男, 副教授, 博士生, 从事云计算与信息安全等研究；郑雪峰(1951−), 男, 教授, 博士生导师, 从事信息安全等研究.

Cheng 等[5]利用唯一分解定理提出了属性合并的方法, 以减少密文中属性字段的长度和计算量; Junbeom 等[6]利用树形二次加密方式提高了属性更新和用户权限撤销的效率; Wan 等[7]结合属性集和代理密钥生成算法提出了分层次的属性加密方案, 降低了密钥生成机构的负担. 然而, 上述 CP-ABE 方案都构建于单一的可信密钥生成机构之上, 该机构负责用自身的密钥算法生成全部的用户私钥, 使其有能力随时解密所有的密文, 因此必须由可信第三方负责密钥生成的问题是该系统的固有缺陷.

Chase 等[8]提出了一种分布式的 ABE 方案, 使用多属性授权机构模型来解决单一第三方密钥生成的问题. 在该方法中, 众多属性管理机构都参与到密钥生成的过程. 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, 因为所有的属性授权机构之间都要进行通信, 从而导致系统的通信开销随着授权机构数量的不断增加呈平方级 $O(N^2)$ 增长. 然而, 在用户端还需要存储这些通信信息来进行密钥的更新操作, 而且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.

Chow 等[9]提出了一个 IBE 模型下匿名的私钥生成方案. 在该方案中, 密钥生成中心 (KGC) 可以为已认证的用户授权私钥而不需要获得用户的身份信息, 实现了用户对 KGC 匿名这种新的安全理念. 从另一个角度看, KGC 不知晓密文的接收者是谁, 如果将这种思想应用于 ABE 系统中将会极大地提高系统的安全性. 然而, Chow 的方案[9]并不能直接应用于现存的 ABE 系统, 因为需要系统中所有的属性都被隐藏, 以防止 KGC 生成用户私钥.

Hur[10]提出将 KGC 拆分, 以交互式计算生成用户密钥来消除可信第三方问题. 然而, 该方法需要在用户端进行幂运算, 增加了用户的计算量. 本文在 Hur 的方案基础上设计一种无可信第三方密钥生成机构方案 (RTTP-CPABES), 利用安全多方计算实现无可信第三方, 以减少用户端的计算量, 并阐明如何将其应用于现存的 CP-ABE 系统.

## 1　预备知识

**定义 1** (双线性对)　设 $p$ 为大素数, $G_0$ 和 $G_1$ 是阶为 $p$ 的乘法循环群, $g$ 为群 $G_0$ 的生成元, 双线性对定义为映射 $e: G_0 \times G_0 \to G_1$, 它满足如下性质.

1) 双线性: $\forall g \in G_0, \forall a, b \in Z_p^*$, 有 $e(g^a, g^b) = e(g^b, g^a) = e(g, g)^{ab}$;

2) 非退化性: $\exists g \in G_0$, 使 $e(g, g) \neq 1$, 即 $e(g, g)$ 不是群 $G_0$ 的单位元;

3) 可计算性: 存在有效算法, 可计算 $e(g, g)$.

**定义 2** (访问结构)　设集合 $\{P_1, P_2, \cdots, P_n\}$ 由 $n$ 个参与者组成, $A \subseteq 2^{\{p_1, p_2, \cdots, p_n\}}$ 是一个非空子集. 如果 $\forall B, C$, 有 $(B \in A) \& (B \subseteq C) \to C \in A$, 则称 $A$ 是单调的, 是参与者集合 $P$ 上的一个访问结构; 若 $\exists A \subseteq 2^{\{p_1, p_2, \cdots, p_n\}} \backslash \{\varnothing\}$, 则称 $A$ 是一个授权的集合, 否则 $A$ 是一个非授权集合.

在 CP-ABE 方案中, 访问规则是根据属性制定的, 因此访问结构包含了经授权的属性集, 本文所讨论的都是按照上述定义的单调访问结构.

**定义 3** (拉格朗日插值)　设 $f(x)$ 为 $n$ 次多项式, 假如给定 $n+1$ 个不同点值 $(x_i, f(x_i))$, 则能唯一确定 $f(x)$ 为

$$f(x) = \sum_{i=1}^{n} f(x_i) \Big( \prod_{1 \leqslant k \neq i \leqslant n} \frac{x - x_k}{x_j - x_k} \Big).$$

定义拉格朗日系数 $\Delta_{i,s}(x) = \prod_{i \in s, i \neq j} \frac{x - j}{i - j}$, 其中 $i$ 和集合 $s$ 中的元素取自 $Z_p^*$.

**定义 4** (安全多方计算 (SMC[11])　安全多方计算是指在无第三方参与的情况下, 对于 $n$ 个参与者 $\{A_1, A_2, \cdots, A_n\}$, 每一位参与者 $A_i$ 拥有一个不想让其他人知道的保密输入值 $x_i$, 希望共同计算出某个函数 $f(x_1, x_2, \cdots, x_n) = (y_1, y_2, \cdots, y_n)$, 执行安全多方计算后每个参与者 $A_i$ 得到相应的 $y_i$, 并且在这个过程中保密值 $x_i$ 并不会泄露给其他参与者, 也不会得到除 $y_i$ 之外的其他信息.

## 2　无可信第三方的密文策略属性加密 (RTTP-CPABES)

### 2.1　安全双方乘法计算模型

AA 和 CSP 各有一个实数 $x$ 和 $y$, 它们之间进行交互式计算后 AA 获得 $u$, CSP 获得 $v$, $u$ 和 $v$ 满足 $u+v = xy$.

1) AA 和 CSP 约定一个整数 $m$, 使计算 $2^m$ 次加法是不可能的. AA 随机生成 $m$ 个实数 $x_1, x_2, \cdots, x_m$, 令 $x = \sum_{j=1}^{m} x_j$.

2) 对于每一个 $j = 1, 2, \cdots, m$, AA 生成一个保密的整数 $1 \leqslant k \leqslant 2$, 给 CSP 发送 $h_1, h_2 (h_k = x_k)$, 余下的 $h_i$ 为随机产生的实数. CSP 不知道哪一个 $h_i$ 是 $x_j$.

3) 对于 $i = 1, 2$, CSP 随机生成实数 $r_j$, 计算 $h_i y - r_j$, AA 获得 $h_k y - r_j = x_j y - r_j$.

4) AA 获得 $u = \sum_{j=1}^{m} (x_j y - r_j) = xy - \sum_{j=1}^{m} r_j$, CSP 获得 $v = \sum_{j=1}^{m} r_j$.

安全多方计算能够在解决参与方协同计算的同时有效保护用户数据隐私的问题, 同时保证输入的独

立性、计算的正确性和数据的保密性. 该概念自提出以来, 研究者已给出了大量成熟、高效的算法, 本节给出的安全双方计算乘法模型便参考了文献[12].

### 2.2 密钥生成方案

属性授权机构(AA)和云服务提供商(CSP)是本方案中的密钥生成机构, AA负责用户的身份认证和属性授权, 它为每个认证后的用户随机生成一个秘密的整数来唯一标识该用户, 且对其他参与方保密, 并用它来生成用户的属性密钥, 用户对CSP是匿名的. CSP不负责管理任何属性, 但它会为用户生成部分密钥, 用户利用它来获得最终的密钥. AA与CSP之间会用各自的主密钥进行交互式的安全双方计算, 然后分别得出用户私钥的一部分, 在这个过程中各自的主密钥不会泄露. 用户需要与两方通信才能够最终获得完整密钥. 在本方案中, AA和CSP都参与密钥生成的过程, 但安全双方计算协议会确保一方不会知晓另一方的主密钥, 所以它们中单独的任何一方都不可能获得全部的用户私钥, 因此假设在AA与CSP不会合谋串通时, 用户的密文数据无法被第三方解密.

在本文所提出的无可信第三方的密钥生成方案中, 系统的主密钥由AA和CSP分别生成, 方案过程描述如下:

1) $\mathrm{Setup}(1^{\lambda}) \rightarrow (\mathrm{param})$. 由CSP或其他初始化机构生成系统公共参数param.

2) $\mathrm{AAKeyGen} \rightarrow (\mathrm{PK_{AA}}, \mathrm{MK_{AA}})$. AA生成其私有的主密钥$\mathrm{MK_{AA}}$和相应的公钥$\mathrm{PK_{AA}}$.

3) $\mathrm{CSPKeyGen} \rightarrow (\mathrm{PK_{CSP}}, \mathrm{MK_{CSP}})$. CSP生成其私有的主密钥$\mathrm{MK_{CSP}}$和相应的公钥$\mathrm{PK_{CSP}}$.

4) $\mathrm{Key\,SMC_{AA}}(\mathrm{MK_{AA}}, R_u) \leftrightarrow \mathrm{Key\,SMC_{CSP}}(\mathrm{MK_{CSP}})$. AA用$\mathrm{MK_{AA}}$和用户认证后生成的唯一标识符$R_u$与CSP的$\mathrm{MK_{CSP}}$进行安全双方计算, AA得到计算结果$\mathrm{SK}_A$, CSP得到计算结果$\mathrm{SK}_C$, 并将$\mathrm{SK}_C$发送给用户.

5) $\mathrm{AAKeyGen}(R_u, S) \rightarrow (\mathrm{SK}_u)$. AA用$R_u$和用户的属性集$S$生成用户的属性私钥$\mathrm{SK}_u$, 并将$\mathrm{SK}_u$和$\mathrm{SK}_A$发送给用户.

6) $\mathrm{KeyIssue}(\mathrm{SK}_u, \mathrm{SK}_A, \mathrm{SK}_C) \rightarrow (\mathrm{SK})$. 用户利用得到的$\mathrm{SK}_u$、$\mathrm{SK}_A$、$\mathrm{SK}_C$计算得出最终私钥SK.

### 2.3 RTTP-CPABES方案描述

1) Setup.

由CSP或其他可信任的初始化机构选择一个阶为大素数$p$的双线性群$G_0$, 其生成元为$g$. 选取一个哈希函数$H: \{0,1\}^* \rightarrow G_0$, 该函数会将描述属性的字符串映射成双线性群中的元素. 系统中的公共参数为$\{G_0, g, H\}$.

2) KeyGen.

密钥生成过程由AA和CSP共同完成.

① $\mathrm{AAKeyGen} \rightarrow (\mathrm{PK_{AA}}, \mathrm{MK_{AA}})$. AA随机选取数值$\beta \in Z_p^*$, 计算$g^{\beta}$, 并生成密钥对为$(\mathrm{PK_{AA}} = g^{\beta}, \mathrm{MK_{AA}} = \beta)$. $\mathrm{PK_{AA}}$作为系统公钥的一部分, 并设$h = \mathrm{PK_{AA}}$, $\mathrm{MK_{AA}}$为AA的主密钥.

② $\mathrm{CSPKeyGen} \rightarrow (\mathrm{PK_{CSP}}, \mathrm{MK_{CSP}})$. CSP随机选取数值$\alpha \in Z_p^*$, 计算$e(g,g)^{\alpha}$, 生成密钥对为$(\mathrm{PK_{CSP}} = e(g,g)^{\alpha}, \mathrm{MK_{CSP}} = \alpha)$. $\mathrm{PK_{CSP}}$作为系统公钥的一部分, $\mathrm{MK_{CSP}}$为CSP的主密钥.

③ $\mathrm{KeySMC_{AA}}(\mathrm{MK_{AA}}) \leftrightarrow \mathrm{KeySMC_{CSP}}(\mathrm{MK_{CSP}})$. AA与CSP之间利用$\alpha$和$1/\beta$进行安全双方乘法计算, AA得到$\gamma$, CSP得到$\eta$, 其中$\gamma + \eta = \alpha/\beta$(在此过程中$\alpha$和$\eta$不会泄露给AA, $\beta$和$\gamma$不会泄露给CSP); 然后CSP计算出$g^{\eta}$.

④ $\mathrm{KeySMC_{AA}}(\mathrm{MK_{AA}}, R_u, \gamma) \leftrightarrow \mathrm{KeySMC_{CSP}}(\eta)$. 当AA认证了一位用户之后, 会为该用户随机选取$R_u = r \in Z_p^*$作为该用户唯一的秘密值, 对CSP和用户保密, 并计算$g^{r/\beta+\gamma}$. AA与CSP之间利用$g^{r/\beta+\gamma}$与$g^{\eta}$进行安全双方乘法计算, 分别得到$\mathrm{SK}_A$和$\mathrm{SK}_C$, 其中$\mathrm{SK}_A + \mathrm{SK}_C = g^{r/\beta+\gamma} \times g^{\eta} = g^{(r+\alpha)/\beta}$(在此过程中$g^{r/\beta+\gamma}$和$\mathrm{SK}_A$不会泄露给CSP, $g^{\eta}$和$\mathrm{SK}_C$不会泄露给AA). CSP将$\mathrm{SK}_C$发送给用户.

⑤ $\mathrm{AAKeyGen}(R_u, S) \rightarrow (\mathrm{SK}_u)$. 生成用户的属性密钥. AA为用户所拥有的每一个属性, 随机选取$r_j \in Z_p^*, j \in S$, $S$为用户的属性集. 计算属性密钥$\mathrm{SK}_u = (\forall j \in S: D_j = g^r \cdot H(i)^{r_j}, D'_j = g^{r_j})$; 然后将$\mathrm{SK}_u$和$\mathrm{SK}_A$发送给用户.

⑥ $\mathrm{KeyIssue}(\mathrm{SK}_u, \mathrm{SK}_A, \mathrm{SK}_C) \rightarrow (\mathrm{SK})$. 用户得到$\mathrm{SK}_u$、$\mathrm{SK}_A$和$\mathrm{SK}_C$后计算出完整密钥$\mathrm{SK} = (D = \mathrm{SK}_A + \mathrm{SK}_C = g^{(r+\alpha)/\beta}, \forall j \in S: D_j = g^r \cdot H(i)^{r_j}, D'_j = g^{r_j})$.

至此, 本方案所生成的用户私钥与Sahai等的方案完全一致, 可直接应用于加密、解密算法, 其安全性和执行效率不变, 同时解决了不可信第三方的问题.

3) Encrypt.

加密算法将利用数据拥有者制定的访问结构$T$[13]对明文$M$进行加密. 该算法首先会对访问结构树$T$中的每一个节点(包含叶子节点)选取一个多项式$q_x$, 多项式的构造将从根节点自上而下采取如下方式: 设节点的门限值为$k_x$, 多项式的度$d_x = k_x - 1$. 对于根节点$R$, 算法随机选取$s \in Z_p^*$, 然后设置$q_R(0) = s$, $q_R$的其他系数随机选取. 对于其余节点$x$, 设$q_x(0) = q_{\mathrm{parent}(x)}(\mathrm{index}(x))$, $q_x$的其他系数随机选取. 设$Y$为访问结构树$T$的叶子节点, 然后给出该访问结构下的密文为$\mathrm{CT} = (T, \tilde{C} = Me(g,g)^{\alpha s}, C = h^s, \forall y \in Y: C_y = g^{q_y(0)}, C'_y = H(\mathrm{att}(y))^{q_y(0)})$.

4) Decrypt.

解密过程首先定义如下的一个循环算法:

$$\text{DecryptNode}(\text{CT}, \text{SK}, x),$$

它以用户的属性集 $S$, 密文 $\text{CT}=(T, \tilde{C}, C, \forall y \in Y : C_y, C'_y)$ 和 $T$ 上的一个节点 $x$ 作为输入. 若 $x$ 为叶子节点, 则设 $i=\text{att}(x)$. 若 $i \in S$, 则计算

$$\text{DecryptNode}(\text{CT}, \text{SK}, x) =$$

$$\frac{e(D_i, C_x)}{e(D'_i, C'_x)} = \frac{e(g^r \cdot H(i)^{r_i}, h^{q_x(0)})}{e(g^{r_i}, H(i)^{q_x(0)})} = e(g,g)^{rq_x(0)};$$

若 $i \notin S$, 则定义 $\text{DecryptNode}(\text{CT}, \text{SK}, x) = \perp$. 若 $x$ 为非叶子节点, 则设 $z$ 为 $x$ 的孩子节点, 计算

$$\text{DecryptNode}(\text{CT}, \text{SK}, z) = F_z.$$

设 $S_x$ 为任意 $k_x$ 个 $F_z \neq \perp$ 的集合, 然后计算

$$F_x = \prod_{z \in S_x} F_z^{\Delta_{i, S'_x}(0)} =$$

$$\prod_{z \in S_x} (e(g,g)^{r \cdot q_z(0)})^{\Delta_{i, S'_x}(0)} =$$

$$\prod_{z \in S_x} (e(g,g)^{r \cdot q_{\text{parent}(z)}(\text{index}(z))})^{\Delta_{i, S'_x}(0)} =$$

$$\prod_{z \in S_x} e(g,g)^{r \cdot q_x(i) \cdot \Delta_{i, S'_x}(0)} = e(g,g)^{r \cdot q_x(0)}.$$

其中: $i=\text{index}(z)$, $S'_x = \{\text{index}(z) : z \in S_x\}$. $R$ 为访问结构树 $T$ 的根节点, 如果用户的属性集 $S$ 满足访问结构, 则设

$$A = \text{DecryptNode}(\text{CT}, \text{SK}, r) =$$

$$e(g,g)^{r \cdot q_R(0)} = e(g,g)^{rs},$$

可解密密文为 $\tilde{C}/(e(C,D)/A) = \tilde{C}/(e(h^s, g^{(r+\alpha)/\beta})/e(g,g)^{rs}) = M$.

## 3　安全性分析

对于 $\text{KeySMC}_{\text{AA}} \leftrightarrow \text{KeySMC}_{\text{CSP}}$ 的交互式安全双方计算过程, 要求一个恶意的 AA(CSP) 在计算结束后只能够获得用户密钥的组成部分 $\text{SK}_A(\text{SK}_C)$, 而无法获知 $\text{SK}_C(\text{SK}_A)$ 的信息. 为了保护用户密钥的安全性, 要求 AA 不能获知 CSP 的主密钥, 同样 CSP 也不能获知 AA 的主密钥.

**定义 5**　存在一个模拟器 $\text{SimSMC}_{\text{CSP}}$[14] 对于所有敌手 $(A_1, A_2)$, 有

$|\Pr[\text{Setup}(1^\lambda) \rightarrow \text{param};$

$\text{CSPKeyGen}(\text{param}) \rightarrow (\text{PK}_{\text{CSP}}, \text{MK}_{\text{CSP}});$

$\text{AAKeyGen}(\text{param}) \rightarrow (\text{PK}_{\text{AA}}, \text{MK}_{\text{AA}});$

$A_1(\text{param}, \text{PK}_{\text{CSP}}, \text{MK}_{\text{CSP}}) \rightarrow (R_u, \text{st});$

$\text{KeySMC}_{\text{CSP}}(\text{param}, \text{MK}_{\text{CSP}}, R_u) \leftrightarrow$

$A_2(\text{st}) \rightarrow b : b=1] - \Pr[\text{Setup}(1^\lambda) \rightarrow \text{param};$

$\text{CSPKeyGen}(\text{param}) \rightarrow (\text{PK}_{\text{CSP}}, \text{MK}_{\text{CSP}});$

$\text{AAKeyGen}(\text{param}) \rightarrow (\text{PK}_{\text{AA}}, \text{MK}_{\text{AA}});$

$A_1(\text{param}, \text{PK}_{\text{CSP}}, \text{MK}_{\text{CSP}}) \rightarrow (R_u, \text{st});$

$\text{SimSMC}_{\text{CSP}}(\text{param}, \text{KeyGen}(\text{MK}_{\text{CSP}}, \text{MK}_{\text{AA}}), R_u) \leftrightarrow A_2(\text{st}) \rightarrow b : b=1]| < \text{negl}(\lambda).$

st 表示敌手保持的状态信息, 对于恶意的 AA (或 CSP), 除一部分密钥外无法获得其他信息. 假设在安全双方计算中, AA 和 CSP 所提供的参与计算的信息是真实的, 否则生成的密钥是无意义的. 如果敌手只知道 $\text{MK}_{\text{CSP}}$ 的部分信息, 则他仍然无法区分与其交换信息的是模拟器还是真正的密钥生成方.

**定义 6**　存在一个模拟器 $\text{SimSMC}_{\text{AA}}$ 对于所有敌手 $(A_1, A_2)$, 有

$|\Pr[\text{Setup}(1^\lambda) \rightarrow \text{param};$

$\text{AAKeyGen}(\text{param}) \rightarrow (\text{PK}_{\text{AA}}, \text{MK}_{\text{AA}});$

$A_1(\text{param}) \rightarrow (R_u, \text{st});$

$\text{KeySMC}_{\text{AA}}(\text{param}, \text{MK}_{\text{AA}}, R_u) \leftrightarrow$

$A_2(\text{st}) \rightarrow b : b=1] - \Pr[\text{Setup}(1^\lambda) \rightarrow \text{param};$

$\text{AAKeyGen}(\text{param}) \rightarrow (\text{PK}_{\text{AA}}, \text{MK}_{\text{AA}});$

$A_1(\text{param}) \rightarrow (R_u, \text{st});$

$\text{SimSMC}_{\text{AA}}(\text{param}, \text{MK}_{\text{AA}}) \leftrightarrow$

$A_2(\text{st}) \rightarrow b : b=1]| < \text{negl}(\lambda).$

对于 CSP (或 AA) 的安全性, 需要构造以 param 为输入的模拟器算法 $\text{SimSMC}_{\text{CSP}}$, $\text{KeyGen}(\text{MK}_{\text{CSP}}, \text{MK}_{\text{AA}})$ 和 $R_u$ 必须模拟对手的视角, $\text{SimSMC}_{\text{CSP}}$ 调用安全双方计算的模拟器来利用对手的输入 $(r, \beta)$, 以便得出计算结果. 假如敌手能够判定与之交互的是模拟器, 则一定是因为敌手输入的数据是虚假的, 这样便违反了安全双方协议的规则.

对于交互式的安全双方计算过程, 不需要可信第三方的参与, 这样便避免了敏感信息泄露. 可使用不经意传输协议 (OT)[15], 以保证传输过程的安全性, 从而 CSP 不会泄露 $y$ 的任何组成部分 $r_j$, 而 CSP 只能猜测 AA 的 $x$ 的组成部分 $x_j$. 因为 CSP 不知道哪一个 $h_i$ 是 $x_j$, 所以猜对一个 $x_j$ 的概率为 1/2, CSP 猜测出 AA 的 $x$ 的概率为 $1/2^m$. 因此, 只要 $2^m$ 足够大, 攻击者不具备无限的计算能力 (否则不经意传输协议 (OT) 是不安全的), 则该安全双方计算模型是安全的.

## 4　效率分析

效率分析的参考对象是原始 CP-ABE 和 Hur 的方案, 所使用的描述符号如下: $C_0$ 表示 $G_0$ 中数据元素的长度; $C_1$ 表示 $G_1$ 中数据元素的长度; $C_T$ 表示密文中访问控制树 $T$ 的长度; $t$ 表示 $T$ 中属性出现的次数; $k$ 表示用户私钥中属性的个数; Exp 表示幂运算的计算量; Add 表示加法运算的计算量.

表 1 显示了密文长度、私钥长度、公钥长度的对

比结果, 本文方案与初始的CP-ABE和Hur的方案一样, 在保证无可信第三方的情况下三方面的数据尺寸不变. 这是因为本文的密钥生成方案所生成的私钥参数与原CP-ABE方案完全相同.

**表 1 数据长度对比**

| 方案 | 密文 | 私钥 | 公钥 | 可信第三方 |
|---|---|---|---|---|
| CP-ABE | $(2t+1)C_0+C_1+C_T$ | $(2k+1)C_0$ | $C_0+C_1$ | 是 |
| Hur | $(2t+1)C_0+C_1+C_T$ | $(2k+1)C_0$ | $C_0+C_1$ | 否 |
| RTTP-CPABES | $(2t+1)C_0+C_1+C_T$ | $(2k+1)C_0$ | $C_0+C_1$ | 否 |

然而, 与原方案相比, 本文在密钥生成阶段需要CSP与用户进行额外的计算任务. 作者分析了原始CP-ABE方案在私钥生成过程中的计算量, 并与Hur的方案和本文方案进行了对比. 表2展示了在密钥生成过程中的计算量和在群$G_0$中的幂运算次数对比. 考虑到本文无可信第三方私钥生成方案由相互独立的两方(AA、CSP)共同完成, 所以CSP与用户相比, 原方案增加了计算量, 但在可接受的范围内. 在用户私钥生成的过程中共进行了2次交互式的安全双方计算, 第1次乘法计算只需在AA和CSP建立初始计算一次即可, 第2次乘法计算在每次新用户认证后进行. 前文给出了计算过程的一种参考模型, 更高效的算法有待进一步研究. 相比Hur的方案, 本文减少了用户端的计算量, 无需幂运算, 相比之下仅有的一次加法运算的时间消耗可忽略. 这样, 可依托云服务端强大的运算能力使本文方案更易于实施.

**表 2 运算量对比**

| 方案 | AA | CSP | User |
|---|---|---|---|
| CP-ABE | $(2k+2)\mathrm{Exp.In}G_0$ | | |
| Hur | $(2k+2)\mathrm{Exp.In}G_0$ | $1\mathrm{Exp.In}G_0$ | $1\mathrm{Exp.In}G_0$ |
| RTTP-CPABES | $(2k+2)\mathrm{Exp.In}G_0$ | $1\mathrm{Exp.In}G_0$ | 1Add. |

## 5 结 论

基于密文策略的属性加密在分布式数据系统有着很多的实际应用, 然而必须由可信任的第三方机构来生成用户私钥是原始CP-ABE系统存在的固有缺陷. 本文提出的RTTP-CPABES方案利用安全双方计算解决了必须依赖可信任的第三方机构问题, 提高了系统的安全性, 并减少了用户端的计算量, 在移动互联网迅速发展的今天, 所提出的方案更适合应用在计算能力有限的便携式客户端上, 推动云计算的应用更加广泛.

## 参考文献(References)


[1] Shamir. Identity-based cryptosystems and signature schemes[C]. Proc of CRYPTO'84 on Advances in Cryptology. Heidelberg, 1985: 47-53.

[2] Boneh Dan, Matt Franklin. Identity-based encryption from the weil pairing[C]. Proc of the 21st Annual Int Cryptology Conf on Advances in Cryptology. California, 2001: 213-229.

[3] Sahai A, Waters B. Fuzzy identity based encryption[C]. Advances in Cryptology-Eurocrypt. Aarhus, 2005: 457-473.

[4] Sahai A, Waters B. Ciphertext-policy attribute based encryption[C]. Proc IEEE Symposium on Security and Privacy. California: IEEE Computer Society, 2007: 321-334.

[5] Yong Cheng, Jiangchun Ren. Attributes union in CP-ABE algorithm for large universe cryptographic access control[C]. 2012 the 2nd Int Conf on Cloud and Greed Computing. Xiangtan: IEEE Computer Society, 2012: 181-185.

[6] Junbeom Hur, Dong Kun Noh. Attribute-based access control with efficient revocation in data outsourcing systems[J]. IEEE Trans on Parallel and Distributed Systems, 2011, 22(7): 1214-1221.

[7] Zhiguo Wan, Jun'e Liu, Robert H Deng. HASBE: A hierarchical attribute-based solution for flexible and scalable acce[J]. IEEE Trans on Information Forensics and Security, 2012, 7(2): 743-754.

[8] Chase M, Chow S S M. Improving privacy and security in multi-authority attribute-based encryption[C]. Proc ACM Conf on Computer and Communications Security. Chicago: ACM, 2009: 121-130.

[9] Chow S S M. Removing escrow from identity-based encryption[C]. Proc PKC2009. Irvine, 2009, 5443: 256-276.

[10] Hur Junbeom. Removing escrow from ciphertext policy attribute-based encryption[J]. Computers and Mathematics with Applications, 2013, 65(9): 1310-1317.

[11] Yao Andrew C. Protocols for secure computations[C]. The 23rd Annual Symposium on Digital Object Identifier. Chicago: IEEE Computer Society, 1982: 160-164.

[12] 李禾, 王述洋. 关于除法的安全双方计算协议[J]. 计算机工程与应用, 2010, 46(6): 86-88.
(Li H, Wang S Y. The secure two party computation about division[J]. Computer Engineering and Applications, 2010, 46(6): 86-88.)

[13] Goyal V, Pandey O, Sahai A, et al. Attribute based encryption for fine-grained access conrol of encrypted data[C]. ACM Conf on Computer and Communications Security. Alexandria: ACM, 2006: 89-98.

[14] Belenkiy M, Lysyanskaya A. P-signatures and noninteractive anonymous credentials[C]. TCC 2008 LNCS 4948. California, 2008: 356-374.

[15] Naor M, Pinkas B. Oblivious transfer with adaptive queries[C]. Proc of Cryto'99. Santa Barbara, 1999: 573-590.


(责任编辑: 曹洪武)