软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2020,31(6):1817−1828 [doi: 10.13328/j.cnki.jos.005704]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

面向公有云的支持快速解密的 CP-ABE 方案*

邹莉萍[1], 冯朝胜[1], 秦志光[2,3], 袁 丁[1], 罗王平[1], 李 敏[1,3]

[1](四川师范大学 计算机科学学院,四川 成都 610101)

[2](电子科技大学 信息与软件工程学院,四川 成都 610054)

[3](网络与数据安全四川省重点实验室(电子科技大学),四川 成都 610054)

通讯作者: 冯朝胜, E-mail: csfenggy@126.com

摘 要: 现有的密文策略基于属性加密 CP-ABE(ciphertext-policy attribute-based encryption)算法普遍在解密时存在计算量过大、计算时间过长的问题,该问题造成 CP-ABE 难以应用和实施.针对该问题,将计算外包引入到方案的设计之中,提出一种面向公有云的基于 Spark 大数据平台的 CP-ABE 快速解密方案.在该方案中,专门根据 CP-ABE 的解密特点设计了解密并行化算法;利用并行化算法,将计算量较大的叶子节点解密和根节点解密并行化;之后,将并行化任务交给 Spark 集群进行处理.计算外包使得绝大多数解密工作由云服务器完成,用户客户端只需进行一次指数运算;而并行化处理则提高了解密速度.安全性分析表明,所提出的方案在一般群模型和随机预言模型下能对抗选择明文攻击.

关键词: 快速解密;解密外包;密文策略基于属性加密;访问树;Spark 平台

中图法分类号: TP309

中文引用格式: 邹莉萍,冯朝胜,秦志光,袁丁,罗王平,李敏.面向公有云的支持快速解密的 CP-ABE 方案.软件学报,2020,31(6):1817−1828. http://www.jos.org.cn/1000-9825/5704.htm

英文引用格式: Zou LP, Feng CS, Qin ZG, Yuan D, Luo WP, Li M. CP-ABE scheme with fast decryption for public cloud. Ruan Jian Xue Bao/Journal of Software, 2020,31(6):1817−1828 (in Chinese). http://www.jos.org.cn/1000-9825/5704.htm

CP-ABE Scheme with Fast Decryption for Public Cloud

ZOU Li-Ping[1], FENG Chao-Sheng[1], QIN Zhi-Guang[2,3], YUAN Ding[1], LUO Wang-Ping[1], LI Min[1,3]

[1](School of Computer Science, Sichuan Normal University, Chengdu 610101, China)

[2](School of Information & Software Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China)

[3](Network and Data Security Key Laboratory of Sichuan Province (University of Electronic Science and Technology of China), Chengdu 610054, China)

Abstract: Most of existing CP-ABE (ciphertext-policy attribute-based encryption) schemes have such problems as over-computation and a long calculation time in decryption, which make them difficult to be applied and implemented. To solve this problem, the computation outsourcing is introduced into the design of CP-ABE scheme, a Spark-platform-based CP-ABE scheme with fast decryption for public cloud is proposed. In this scheme, the decryption parallelization algorithm is designed based on the decryption feature of CP-ABE, with which, decryption at both leaf node and root node with over-computation is parallelized. Then, the parallelization tasks are handed over to the Spark cluster. The computation outsourcing makes the most decryption computation done by cloud servers, while the

* 基金项目: 国家自然科学基金(61373163); 国家科技支撑计划(2014BAH11F02, 2014BAH11F01); 四川省科技支撑计划(2015GZ079); 网络与数据安全四川省重点实验室开放课题(NDSMS201606); 四川省教育厅重点项目(17ZA0322)

Foundation item: National Natural Science Foundation of China (61373163); National Key Technology Research and Development Program of theMinistry of Science and Technology of China (2014BAH11F02, 2014BAH11F01); Science and Technology Support Program of Sichuan Province (2015GZ079); Opening Foundation for the Key Laboratory of Sichuan Province (NDSMS201606); Key Project for Education Department of Sichuan Province (17ZA0322)

收稿时间: 2018-01-06; 修改时间: 2018-05-08; 采用时间: 2018-10-17

user client only needs an exponential operation, and parallelization greatly improves the speed of decryption. Security analysis shows that the proposed scheme can fight against chosen plaintext attack under both the generic group model and the random oracle model.

Key words: fast decryption; decryption outsourcing; CP-ABE; access tree; Spark platform

大数据的出现,使得企业特别是中小型企业将数据外包给云服务商[1]变得紧迫,而数据外包首先要解决好数据机密性和隐私性[2]问题.加密是确保数据机密性和隐私性的有效途径.如果只是进行加密保存,传统的加密方法即可胜任;如果还要实现密文共享,虽然传统的加密方法也可以实现,但只有采取效率低下的"一对一"加密方式和粗粒度的访问控制[3]方式.为了实现密文共享的高效性和访问控制的细粒度,基于属性的加密方法被提出[4].基于属性的加密方法可以分为两种:密文策略基于属性加密方法 CP-ABE[5]和密钥策略基于属性加密方法 KP-ABE(key-policy attribute-based encryption)[6].由于 CP-ABE 的访问控制方法和企业 IT 系统在明文访问控制上广泛采用的基于角色的访问控制方法 RABC(role-based access control)相似,更加受到企业的关注.然而,现有的大部分 CP-ABE 方案在解密时所用的时间随着匹配访问策略的属性数量呈线性增长,用户客户端计算量急剧增长,对用户终端设备要求较高,常用的用户终端特别是平板、手机难以胜任.针对这一问题,利用 Spark 集群中的 Map 运算和 Reduce 运算,提出一种基于 Spark 的支持快速解密的 CP-ABE 方案.本文具体贡献包括:

(1) 提出一种基于 Spark 集群的解密外包方案.在该方案中,云服务器对大量密文进行存储,授权中心生成仅由用户存储的最终解密密钥 *DK* 和用于云服务器解密的转换密钥 *TK*,降低用户客户端对于存储性能的需求.利用转换密钥 *TK* 对上传到云服务器端密文使用 Spark 集群中的 Map 运算与 Reduce 运算进行半解密,将用户客户端解密代价降低到一次指数运算,显著减少云服务器计算时间;

(2) 设计了一种快速求解共享访问树根节点值的算法.该策略的核心是:将求解任务分成若干个可并行执行的子任务,然后将子任务交由 Spark 集群中的 Map 节点去完成,Reduce 节点通过汇聚 Map 节点的输出结果计算出目标值.现有的包括 Bethencourt 等人[5]的方案(以下简称 BSW 方案)在内的用树来表示访问策略 CP-ABE 方案,几乎所有的方案求解根节点所需的指数运算次数都和共享访问树的节点数量成正比,而基于所设计的快速求解算法求解根节点值所需的指数运算时间仅和叶子节点的数量成正比;

(3) 证明方案的有效性.安全性分析表明,在一般群模型和随机预言模型下可抵御选择明文攻击.性能分析表明,该方案在计算上对用户客户端要求较低,手机、平板、电脑等移动设备亦可胜任.

1 相关研究

2005 年,Sahai 和 Waters[4]在欧洲密码学年度会议上首先提出了一种基于属性的加密算法(attribute-based encryption,简称 ABE),并在学术界受到了广泛的关注与研究.在该方案中,将每一个标识视为一组描述属性,并且将用户私钥和密文都与属性相关联,只有当用户私钥的属性集合满足密文属性集合时,才能解密出正确的明文数据.之后,Bethencourt 和 Goyal 分别提出了密文策略基于属性加密 CP-ABE[5]和密钥策略基于属性加密 KP-ABE[6].在 CP-ABE 中,密文策略与访问属性相关联;在 KP-ABE 中,密钥策略与访问属性相关联.2007 年,Ostrovsky 等人[7]提出了一种允许使用任意访问结构的 ABE 方案,并且基于 DBDH 假设(decisional bilinear diffie-Hellman assumption)进行了安全性证明.同年,Ling 等人[8]提出一种支持访问结构包含正属性和负属性的 CP-ABE 方案,在 DBDH 假设下,证明了该论文提出的基本方案满足 CPA 安全,然后使用 Canetti-Halevi-Katz 技术进行一次性签名,使得改进方案满足 CCA 安全.但是该方案只适用于仅限于门限与("AND")的共享访问策略.2008 年,Goyal 等人[9]首次提出了一种基于数论理论假设且支持高级访问结构的 CP-ABE 方案,并且在 DBDH 假设下给出安全性证明.2009 年,Li 等人[10]提出了一种具有用户责任的 ABE 方案,在该方案中,给每个用户私钥中嵌入额外的用户特定信息,以防止用户间非法密钥共享问题,进一步解决用户访问隐私问题.2011 年,Waters[11]提出一种更加安全的易于实现的高效 CP-ABE 方案,在该方案中,提出了 3 种不同的结构,这些结构允许在系统效率和不同的安全假设间进行不同的权衡,使其比基本的 CP-ABE 方案拥有更好的实用效率.但是其解密时间还是随着密文大小增长,随着访问结构复杂度增加而增加.同年, Green 等人[12]在 Water 等人[11]的方案上提出了

一种实现了 RCCA 安全并且同时适用于 CP-ABE 和 KP-ABE 的解密外包方案.该方案将大量解密计算外包给云服务器,减少了用户客户端解密计算量,但是没有给出云服务器端进行解密的具体操作.2012 年,Li 等人[13]基于 Zhou 等人[14]提出的外包加密 CP-ABE 方案,提出了一种基于 MapReduce 的外包加密方案.但是在该方案中,没有提出相应的使用 MapReduce 的快速解密方案,且 Hadoop 中的 MapReduce 存在高延迟性的特性,在实际应用中较难实现.2013 年,Lai 等人[15]基于 Green 等人[12]的方案,提出了一种可验证的外包解密方案,并且该方案的安全性不依赖于随机预言模型.在该方案中,除了对明文数据进行加密,还需要对 G_T 域中的一个随机元素进行加密处理.由于加密工作量的增加,导致解密时计算代价增加一倍. 2015 年,Qin 等人[16]基于 Green 等人[12]的方案,提出了一个高效的可验证的解密外包方案,与 Lai 等人[15]方案相比,在该方案中引入了哈希映射来验证第三方解密的正确性.同年,Lin 等人[17]也提出了一种可验证的外包解密方案,并且证明其在标准模型下是安全的.与 Lai 等人[15]方案相比,密文长度减少,解密计算代价减少接近一半. 2016 年,Mao 等人[18]提出一种 CPA 安全的可验证的外包解密方案.与其他可验证的 CPA 安全相比,该方案不再单独加密一个额外的随机消息,而是采用同时对一个消息和一个随机值进行加密,然后通过该随机值来将消息提交给云服务器,使得其拥有比一般 CPA 安全外包方案更短的密文长度.同年,Zhang 等人[19]提出一种具有可验证性外包功能的自适应安全的多授权 CP-ABE 方案.该方案不仅实现了解密外包验证,并且降低了公钥大小.但是在该方案中,每个属性不是由唯一的控制权限控制,所以仅能被视为适当的多授权 ABE 方案,且其仅支持单调访问结构.2017 年,Liu 等人[20]提出一种将离线与在线技术相结合的密文策略 ABE 方案,并且支持可验证的外包解密.该方案将密钥生成和大部分加密计算都进行离线计算,减少在线计算时间,并且将大部分解密工作量都外包给第三方服务器,但是仍然没有对云服务器端解密方式进行快速处理.

由以上分析可知:现有的 ABE 方案,无论是在用户端还是云服务器端,都存在计算量过大、计算时间过长的问题.

2 系统模型与算法定义

2.1 系统模型

支持快速解密的 CP-ABE 系统模型如图 1 所示,包括授权中心、云服务器、数据所有者和数据消费者.

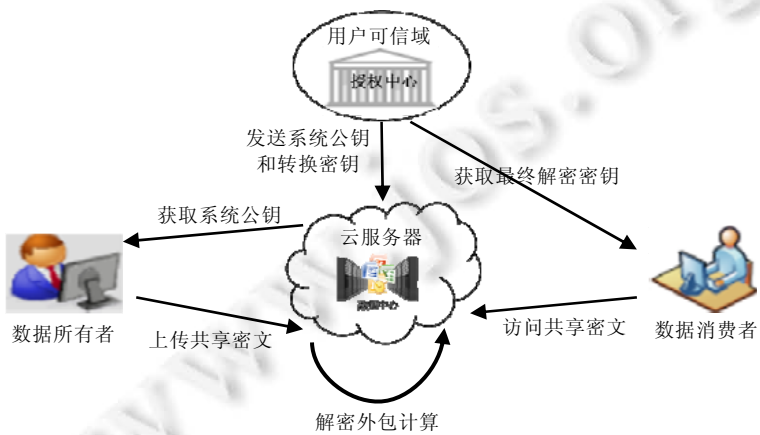

Fig.1 Ciphertext-policy attribute-based encryption system model with fast decryption

图 1 支持快速解密的密文共享系统模型

(1) 授权中心:通常在企业可信域内,负责生成系统公钥 *PK*、系统主密钥 *MK*、用户最终解密密钥 *DK* 和云服务器转换密钥 *TK*;

(2) 云服务器:云服务器是模型中半可信的第三方,用于存储大量密文数据,并且从授权中心获取转换密

钥 *TK* 对密文进行半解密;

(3) 数据所有者:数据所有者是模型中数据的原始拥有者,将数据进行加密并上传至云服务器进行存储;

(4) 数据消费者:数据消费者是模型中数据的使用者,使用由授权中心获取到的最终解密密钥 *DK*,将从云服务器获取到的半解密密文进行解密,以获得共享数据.

2.2 算法定义

定义 1. 一种基于 Spark 集群的解密外包方案由以下 5 个算法构成.

- *Setup*(λ,*U*)→(*PK*,*MK*):由授权中心执行,输入安全参数λ、属性空间 *U*,输出系统公钥 *PK* 和系统主密钥 *MK*;
- *Encrypt*(*PK*,*M*,*T*)→(*CT*):由用户客户端执行,输入访问策略 *T*、明文 *M* 和系统公钥 *PK*,输出密文 *CT*;
- *KeyGen*(*MK*,*S*)→(*DK*,*TK*):由授权中心执行,输入用户属性集合 *S*、系统主密钥 *MK*,输出用户最终解密密钥 *DK* 和转换密钥 *TK*;
- *Transform*(*CT*,*TK*)→(*CT*′):由云服务器执行,输入密文 *CT* 和转换密钥 *TK*,输出半解密密文 *CT*′;
- *Decrypt*(*DK*,*CT*′)→(*M*):由用户客户端执行,输入用户最终解密密钥 *DK* 和半解密密文 *CT*′,输出解密密文 *M*.

3 支持快速解密的 CP-ABE 方案

3.1 预备知识

3.1.1 双线性映射

双线性映射(bilinear map)[21,22]因其在密钥生成上具有高效、精确和安全的特点,使其成为 ABE 的数学基础之一.其具体定义如下.

设 G_1,G_2 和 G_T 都是阶为素数 p 的乘法循环群,记 G_1 的生成元为 g_1,G_2 的生成元为 g_2.形如 e:$G_1\times G_2\to G_T$ 的映射就是双线性映射.双线性映射具有以下特性.

(1) 双线性.对于$\forall a,b\in Z_p$,$\forall u\in G_1$,$\forall v\in G_2$ 有 $e(u^a,v^b)=e(u^b,v^a)=e(u,v)^{ab}$,当 $G_1=G_2$ 时,该映射被称为对称双线性映射;

(2) 可计算性.对于任意取值,双线性映射的计算都是高效的;

(3) 非退化性.$e(g,g)\neq 1$,即不会将所有的数对都映射为 G_T 中的同一元素.

3.1.2 拉格朗日插值系数

拉格朗日插值法[23]是以法国数学家约瑟夫·拉格朗日命名的一种多项式插值方法.假设平面上有(x_0,y_0),(x_1,y_1),...,(x_{n-1},y_{n-1})共 n 个点,令函数 $f(x)$经过这 n 个点,设 $D_n=\{0,1,...,n-1\}$,存在 n 个多项式 $p_j(x),j\in D_n$.则对于任意 $k\in D_n$,都有 $p_k(x)$,$B_k=\{i|i\neq k,i\in D_n\}$,使得:

$$p_k(x)=\prod_{i\in B_k}\frac{x-x_i}{x_k-x_i} \tag{1}$$

$p_k(x)$就是拉格朗日系数.拉格朗日插值公式的一般表达形式如下:

$$L_n(x)=\sum_{j=0}^{n-1}y_i p_j(x) \tag{2}$$

在本文方案中,令 $i\in Z_p^*$,S 表示一个节点集合,$\Delta_{i,S}(x)$表示拉格朗日系数,则由公式(1)可得:

$$\Delta_{i,S}(x)=\prod_{j\in S,j\neq i}\frac{x-j}{i-j}.$$

3.1.3 共享访问树

为实现秘密共享,访问策略可以用共享访问树表达.设共享访问树为 *T*,且其根节点为 *R*.共享访问树 *T* 的每一个非叶子节点都代表一个“AND”门或“OR”门.为实现基于属性的访问控制,共享访问数 *T* 的每一个叶子节点都和一个属性相对应.

令共享访问树 T 的每个节点 x 的阈值为 k_x(当 x 为叶子节点时,设 k_x=1),为 x 随机选择一元 k_x–1 次多项式 q_x.秘密共享从 T 的根节点 R 开始,采用自上而下的方式进行.选择随机数 $s \in Z_p^*$ (p 为大素数),令根节点的秘密共享数 $q_R(0)=s$,加密共享数为 $e(g,g)^{\alpha s}$($\alpha \in Z_p^*$);令其他节点的秘密共享数为 $q_x(0)=q_{parent}(x)(index(x))$ ($parent(x)$表示节点 x 的父节点;$index(x)$表示节点 x 在兄弟节点中的序号,按共享访问树结构从左往右依次进行编号),加密共享数为 $e(g,g)^{\alpha q_x(0)}$.

如果属性集 S 满足 T_x(以 x 为根的树),记作 $T_x(S)$=1.计算 $T_x(S)$按照递归方式进行.如果 x 不是叶子节点,则为 x 的每个孩子节点 x'计算 $T_{x'}(S)$;如果 x 代表“AND”门,只有所有孩子的返回值都为 1 时,才有 $T_x(S)$=1;如果 x 代表“OR”门,只要有一个孩子的返回值为 1,就有 $T_x(S)$=1.如果 x 是叶子节点且其关联的属性属于 S,则 $T_x(S)$=1.

3.2 最优子树与加密共享数

3.2.1 共享访问树的最优子树

如果某个用户属性集满足共享访问树 T,就可以用用户密钥和叶子节点的共享数计算出根节点 R 对应的共享数 $e(g,g)^{\alpha s}$,其中,s 为秘密共享数.显然,满足共享访问树并不一定要求属性集中的属性和每个叶子节点关联的属性都匹配,可能只需匹配部分叶子节点就能满足访问树.

定义 2. 如果属性集 S 只和 T 的叶子节点集 L_T 的某个子集 SL_T 匹配就能实现对 T 的满足,即 $T(S)$=1,那么由从 T 的根节点到叶子节点集 SL_T 形成的 T 的子树,就被称作共享访问树 T 的解密子树.

定义 3. 共享访问树 T 的解密子树中叶子节点最少的子树,被称作共享访问树 T 的最优解密子树.

定义 4. 将共享访问树 T 的最优解密子树中所有的“OR”节点删除后所形成的树,称作共享访问树 T 的最优简化解密子树.

3.2.2 加密共享数

观察如图 2 所示共享访问树,该树的所有非叶子节点都代表“AND”门,s_i 和 $F = E^{s_i}$ 分别表示节点 i 的秘密共享数和加密共享数,l_{ij} 表示 i 的第 j 个孩子的拉格朗日系数,其中, $E=e(g,g)^{\alpha}$.

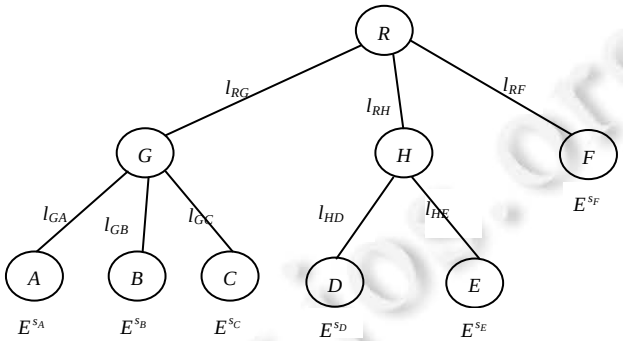

Fig.2 Sharing access tree

图 2 共享访问树

根据共享访问树秘密共享方法和拉格朗日插值法,计算根节点加密共享数的过程如下:

$$
\begin{aligned}
F_G &= (E^{s_A})^{l_{GA}} \cdot (E^{s_B})^{l_{GB}} \cdot (E^{s_C})^{l_{GC}} = E^{s_G},\\
F_H &= (E^{s_D})^{l_{HD}} \cdot (E^{s_E})^{l_{HE}} = E^{s_H},\\
F_R &= (E^{s_G})^{l_{RG}} \cdot (E^{s_H})^{l_{RH}} \cdot (E^{s_F})^{l_{RF}}\\
&= ((E^{s_A})^{l_{GA}} \cdot (E^{s_B})^{l_{GB}} \cdot (E^{s_C})^{l_{GC}})^{l_{RG}} \cdot ((E^{s_D})^{l_{HD}} \cdot (E^{s_E})^{l_{HE}})^{l_{RH}} \cdot (E^{s_F})^{l_{RF}}\\
&= E^{s_A l_{GA} l_{RG}} \cdot E^{s_B l_{GB} l_{RG}} \cdot E^{s_C l_{GC} l_{RG}} \cdot E^{s_D l_{HD} l_{RH}} \cdot E^{s_E l_{HE} l_{RH}} \cdot E^{s_F l_{RF}}\\
&= F_A^{\,l_{GA} l_{RG}} \cdot F_B^{\,l_{GB} l_{RG}} \cdot F_C^{\,l_{GC} l_{RG}} \cdot F_D^{\,l_{HD} l_{RH}} \cdot F_E^{\,l_{HE} l_{RH}} \cdot F_F^{\,l_{RF}}.
\end{aligned}
$$

根据以上观察,得到定理 1.

定理 1. 共享访问树的最优简化子树根节点对应的加密共享数等于以每个叶子节点对应的加密共享数为底、以从叶子节点到根节点路径上所有节点对应的拉格朗日系数的积为指数的幂的乘积.

证明:用数学归纳法容易证明,限于篇幅考虑,省略证明过程. □

3.3 方案构造

方案包括系统初始化、数据加密、密钥生成、数据转换和数据解密这 5 个模块.

(1) 系统初始化:*Setup*(λ,*U*)→(*PK*,*MK*)

系统初始化算法由授权中心执行,该算法以系统空间 *U* 和安全参数λ作为输入,选择一个阶为大素数 *p* 的双线性群 G_0,记 G_0 的生成元为 *g*,定义双线性映射 $e:G_0\times G_0\to G_T$.设定哈希函数 $H_1:G_T\to\{0,1\}^l$,选择两个随机数 $\alpha,\beta\in Z_p^*$,输出系统公钥 *PK* 信息如下:

$$PK=(G_0,g,h=g^{\beta},e(g,g)^{\alpha},H_1).$$

将其上传至云服务器,并向云服务器及所有用户公开.

授权中心保存系统主密钥:$MK=(\beta,g^{\alpha})$.

(2) 数据加密:*Encrypt*(*PK*,*M*,*T*)→(*CT*)

数据加密算法由用户客户端执行,该算法以系统公钥 *PK*、明文数据 *M*、密文共享访问树 *T* 作为输入,输出明文数据 *M* 与共享访问树 *T* 相关联的密文数据 *CT*.

随机选择秘密共享数 $s\in Z_p^*$,令 $q_R(0)=s$(*R* 表示共享访问树 *T* 的根节点).按照第 3.1.3 节介绍的方法进行秘密共享.令 *Y* 是共享访问树 *T* 中的所有叶子节点的集合,输出如下密文:

$$CT=(T,\tilde{C}=M\cdot e(g,g)^{\alpha s},C=h^s,\hat{C}=H_1(M),\forall y\in Y:C_y=g^{q_y(0)},C_y'=H(att(y))^{q_y(0)}).$$

(3) 密钥生成:*KeyGen*(*MK*,*S*)→(*DK*,*TK*)

密钥生成算法由授权中心执行,该算法以系统主密钥 *MK* 和属性集合 *S* 作为输入,选择一个随机数 $r\in Z_p^*$,对于任意属性 $j\in S$,选择随机数 $r_j\in Z_p^*$.在授权中心输出用户私钥 *SK* 如下:

$$SK=\left(D=g^{\frac{\alpha+r}{\beta}},\forall j\in S:D_j=g^r\cdot H(j)^{r_j},D_j'=g^{r_j}\right).$$

选择随机数 $n\in Z_p^*$,计算用于云服务器外包解密的转换密钥 *TK*,并将其发送给云服务器.

转换密钥 *TK* 的计算过程如下:

$$TK=\left(D^{TK}=\left(g^{\frac{\alpha+r}{\beta}}\right)^n,\forall j\in S:D_j^{TK}=(g^r\cdot H(j)^{r_j})^n,D_j'^{(TK)}=(g^{r_j})^n\right).$$

最后,通过安全信道将最终解密密钥 *DK*=*n* 发送给用户客户端.

(4) 数据转换:*Transform*(*CT*,*TK*)→(*CT*′)

数据转换算法由云服务器执行,该算法以密文消息 *CT* 和转换密钥 *TK* 作为输入,根据定理 1,使用 Spark 集群快速计算半解密密文 *CT*′的模型如图 3 所示.

首先检测用户属性是否满足共享访问树:若不满足,则直接输出⊥;否则,针对最优简化解密子树中的节点 *x*(包括叶子节点)启动 Spark 中的 Map 运算,对最优解密子树中的叶子节点 *y* 启动 Spark 中的 Reduce 运算.整个算法包括 *Transform*(*Map*),*Transform*(*Reduce*)和 *Transform*(*Multiple*)这 3 个重要模块.

① *Transform*(*Map*)

对每个叶子节点 *y*,使用转换密钥 *TK*,通过如下计算获得该节点的部分秘密值 CT_y':

$$CT_y'=Transform(CT,TK,y)=\frac{e(D_i^{TK},C_y)}{e(D_i'^{(TK)},C_y')}=\frac{e((g^r\cdot H(j)^{r_j})^n,g^{q_y(0)})}{e((g^{r_j})^n,H(i)^{q_y(0)})}=e(g,g)^{nrq_y(0)}.$$

令 $NodeLeaf_y$ 为叶子节点标识符,将每个叶子节点的标识符和部分秘密值 CT_y' 以($NodeLeaf_y$, CT_y')键值对分

别发送到不同的 Reduce.

对于从叶子节点 y 到根节点路径上的每个非叶子节点 x,计算该节点在该路径上的孩子节点 sx 对应的拉格朗日系数值$\Delta index(sx),S(0)$.对于 x 的每个子孙叶子节点 y,将$(NodeLeaf_y,\Delta index(sx),S(0))$键值对发送给 y 所对应的 Reduce.

② *Transform*(*Reduce*)

对每一个 Reduce 运算,首先判断传入的值是叶子节点的值还是非叶子节点的值:如果是两个非叶子节点的值,则将两个节点的拉格朗日系数值相乘;如果是一个叶子节点的值、一个非叶子节点的值且该轮 Reduce 运算所需要的所有非叶子节点的值还未全部传入,则将非叶子节点的拉格朗日系数值保存起来,直到所有节点的值传入完毕.在 Reduce 节点接收到所有值后,根据定理 1 计算叶子节点 y 对应的幂值:

$$Y_y = e(g,g)^{nrq_y(0)\cdot\prod \Delta index(sx),S(0)}.$$

③ *Transform*(*Multiply*)

令共享访问树 T 的最优简化解密子树 ST 的叶子节点的总数为 $num(ST)$,将每一个 Reduce 的结果收集起来进行连乘运算,得到根节点的秘密共享数:

$$CT'_R = \prod_{y=1}^{num(ST)} Y_y = \prod_{y=1}^{num(ST)} e(g,g)^{nrq_y(0)\cdot\prod \Delta index(sx),S(0)} = e(g,g)^{nrq_R(0)} = e(g,g)^{nrs}.$$

然后进行如下计算,得到半解密密文 CT'.

$$CT' = \frac{e(C,D^{TK})}{CT'_R} = \frac{e\left(h^s,\left(g^{\frac{\alpha+r}{\beta}}\right)^n\right)}{e(g,g)^{nrs}} = e(g,g)^{n\alpha s}.$$

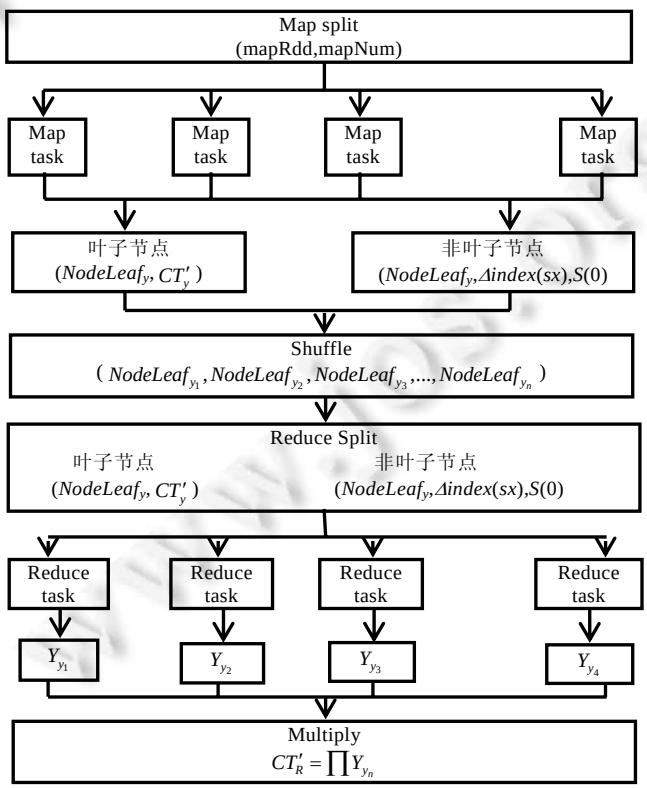

Fig.3 Spark-based semi-decryption model

图 3 基于 Spark 平台进行半解密模型图

(5) 数据解密:$Decrypt(DK,CT')\rightarrow(M)$

数据解密算法由用户客户端执行,以用户最终解密密钥 DK 和云服务器传回的半解密密文 CT'作为输入,输出明文 M.

首先,使用最终解密密钥 DK 对从云服务器传回的半解密密文 CT'进行开方处理,得到密文解密参数 $A=e(g,g)^{\alpha s}$,再通过如下计算得到明文 M':

$$M'=\frac{\tilde{C}}{A}.$$

如果 $H_1(M')$等于 $\hat{C}$,则表明 M'等于 M,输出明文 M;否则,输出⊥.

4 安全性与性能分析

4.1 安全模型

实现 CPA 安全的 CP-ABE 安全模型如下.

- 初始化:挑战者运行 $Setup$ 算法,产生系统公钥 PK,并且将公钥 PK 公开给敌手;
- 第 1 阶段:敌手 A 查询属性集 $S_1,...,S_{q_1}$ 分别对应的私钥;
- 挑战:敌手提交要挑战的访问结构 A^*和两个等长的明文消息 M_0 和 M_1.选择挑战的访问结构时,应保证第 1 阶段查询过的任意属性集都不满足该访问结构.挑战者选择一个随机数 b,并且基于访问结构 A^*加密数据 M_b 形成密文 CT^*,然后将密文 CT^*提供给敌手;
- 第 2 阶段:选择一组都不满足访问结构的属性 $S_{q_1+1},...,S_q$,重复第 1 阶段的操作;
- 猜测:敌手 A 输出与随机数 b 对应的猜测值 b'.

敌手能够赢得上述挑战游戏的优势为 $\Pr[b'=b]-1/2$.在上述游戏中,如果敌手在多项式时间内赢得游戏的优势是可忽略不计的,就可以认为对应的 CP-ABE 方案是安全的.

4.2 安全性分析

定理 2. 本文所提出的方案,在一般群模型和随机预言模型下可抵御选择明文攻击.

证明:假设敌手(算法)A 在一般群模型和随机预言模型能以不可忽略优势攻破本文所提出的方案,那么可以基于 A 构建敌手(算法)B,使得其可以在同样模型下攻破 BSW 方案,这与在一般群模型和随机预言模型下 BSW 方案可以抵御选择明文攻击矛盾,故本文所提出的方案在一般群模型和随机预言模型下可抵御选择明文攻击.下面说明敌手 B 的构建过程.

- 初始化阶段:敌手 B 获取 BSW 方案的公钥 $PK=(G_0,g,h=g^{\beta},e(g,g)^{\alpha})$,并将其发送给 A;
- 第 1 阶段:敌手 B 建立空表 T,敌手 A 可重复发出查询请求.A 发出一次查询后,B 将要查询属性集 S 发送给 BSW 方案挑战者(模拟器),由 BSW 方案挑战者利用密钥生成算法生成与 S 对应的私钥 SK'并返回给 B.B 选择一个随机数 $n\in Z_p^*$,由 SK'计算出转换密钥 TK,计算 $SK=(n,TK)$.将 SK 返回给 A.将(S,SK,TK)存入到表 T 中;
- 挑战阶段:当 A 确定第一个阶段结束后,向 B 提交要挑战的访问结构树 T 和两个明文消息 $M_0,M_1\in G_1$.提交 T 时,要确保已经查询的属性集中没有一个满足 T.B 将 T 和两个消息发送给 BSW 方案挑战者以获得密文 CT;
- 第 2 阶段:B 继续接受 A 的查询,但查询分为如下两种情况.
  - S 不满足 T.重复第 1 阶段的查询过程;
  - S 满足 T.该情况下,无法查询 S 对应私钥,故只能按如下方法生成伪转换密钥:B 随机选择 $d\in Z_p^*$,$t\in G_0$,运行 $KeyGen((d,t,PK),S)$生成密钥 SK',令 $TK=SK'$,$SK=(d,TK)$.将 TK 返回给 A,将(S,SK,TK)存入到表 T 中;
- 猜测:如果 A 输出的猜测为 b',那么 B 输出的猜测也为 b'.因此,如果 A 能以不可忽略优势攻破本文所提

出的方案,那么 B 也能以不可忽略优势攻破 BSW 方案.

5 性能分析

5.1 理论分析

在进行数据处理时,双线性运算 B 和指数运算 E 所需要的时间占 CP-ABE 方案绝大部分时间,故以这两个指标作为衡量计算性能的指标.令$|T_L|$为 T 的叶子节点个数(树结构)或矩阵的行数(LSSS 结构),则针对本文方案,在解密时,用户客户端仅需对从云服务器端获取到的半解密密文 CT'进行一次指数运算,即用户客户端解密代价仅为 $1E$.云服务器端需要对每一个叶子节点进行一次双线性运算和一次指数运算,最后在求解半解密密文时再进行一次双线性运算,所以在云端的解密代价为$(|T_L|+1)B+|T_L|E$,且其运行在 Spark 集群上,将云服务器的解密代价分给集群中各个节点,从而进一步减少了云服务器解密时间.故本文方案解密总代价为$(|T_L|+1)B+(|T_L|+1)E$.

本文方案与 BSW 方案、文献[12]中的方案、文献[18]中的方案在用户端解密计算开销、服务器端解密计算开销以及解密计算总开销对比见表 1~表 3.

Table 1 Comparison of decryption time in user client

表 1 用户客户端解密计算开销对比

方案	解密时间
BSW 方案	$(2\|T_L\|+1)B+(2\|T_L\|-2)E$
文献[12]中的方案	$1E$
文献[18]中的方案	$1E$
本文方案	$1E$

Table 2 Comparison of decryption time in cloud server

表 2 云服务器计算开销对比

方案	解密时间
文献[12]中的方案	$(\|T_L\|+2)B+(2\|T_L\|-1)E$
文献[18]中的方案	$(2\|T_L\|+1)B+\|T_L\|E$
本文方案	$(\|T_L\|+1)B+\|T_L\|E$

Table 3 Comparison of total decryption time

表 3 解密计算总开销对比

方案	解密时间
BSW 方案	$(2\|T_L\|+1)B+(2\|T_L\|-2)E$
文献[12]中的方案	$(\|T_L\|+2)B+2\|T_L\|E$
文献[18]中的方案	$(2\|T_L\|+1)B+(\|T_L\|+1)E$
本文方案	$(\|T_L\|+1)B+(\|T_L\|+1)E$

5.2 实验分析

实验采用 Eclipse 开发工具,利用双线性对加密库(JPBC)(http://crypto.stanford.edu/pbc/)和 CP-ABE 开发工具包(http://acsc.csl.sri.com/cpabe/),基于本文提出的方案,在 Spark 集群环境下进行.双线性运算和指数运算皆采用 JPBC 中原有的操作,从素数阶群 $y^2=x^3+x$ 中选取群 G 和 G_T,群 G 和 G_T 中的元素长度为 512 位.实验使用的虚拟 PC 机用户客户端配置为:1 个 Intel(R) Xeon(R) CPU(E5-2620 2.0GHZ),内存 1GB,系统 CentOS6.5 64 位;实验使用的虚拟移动设备用户客户端配置为:1 个 Google APIs Intel Atom(x86_64),内存 2GB,系统 Android7.0;实验使用的虚拟云服务器端配置为:4 个 Intel(R) Xeon(R) CPU(E5-2620 2.0GHZ),内存 8GB,系统 CentOS6.5 64 位;实验使用的虚拟云服务器端 Spark 集群配置为:11 台虚拟机,其中 1 台为 master 节点,另外 10 台均为 node 节点.每台配置均为:2 个 Intel(R) Xeon(R) CPU(E5-2620 2.0GHZ),内存 8GB,系统 CentOS6.5 64 位.

实验时,分别使用 PC 机和手机作为客户端进行对比实验.为保证数据的可靠性,每组对比实验中每个方案每轮进行 50 次运算并取其平均值.针对 PC 机用户客户端解密时间和总解密时间,将本文方案分别与 BSW 方

案、文献[12]中的方案、文献[18]中的方案进行对比;针对移动设备用户客户端解密时间和总解密时间,将本文方案分别与文献[12]中的方案、文献[18]中的方案进行对比;针对云服务器端解密时间,将本文方案分别和基于BSW 方案的 Proxy 方案(即将解密工作外包给云服务器但未进行并行化处理)、文献[12]中的方案、文献[18]中的方案进行对比.为方便进行实验比较,密文共享访问结构之间的逻辑关系均为逻辑与(即“AND”关系).每组对比实验进行 25 轮测试,每轮增加 4 个共享访问策略属性,为保证实验变量的唯一性,每一轮所使用的数据及共享访问策略相同.

当用户客户端分别为 PC 机和移动设备时,本文方案与 BSW 方案、文献[12]中的方案、文献[18]中的方案在用户客户端解密时间对比分别如图 4、图 5 所示.当用户客户端为 PC 机时,本文方案、文献[12]中的方案、文献[12]中的方案在用户客户端解密时间在 3ms~5ms 之间,而 BSW 方案的解密时间随着访问策略属性数量的增加呈急剧上升趋势;当用户客户端为移动设备时,本文方案、文献[12]中的方案和文献[18]中的方案这 3 种方案在用户客户端解密时间均在 4ms~8ms 之间.

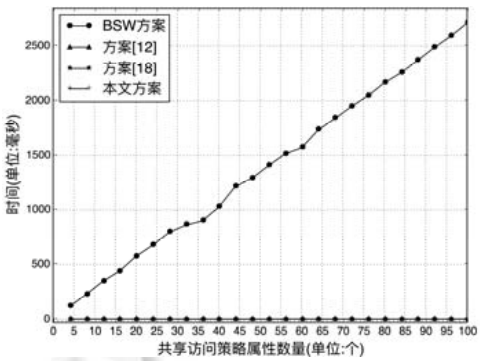

Fig.4 Comparison of decryption time in user client (PC)

图 4 用户端解密时间对比图(PC)

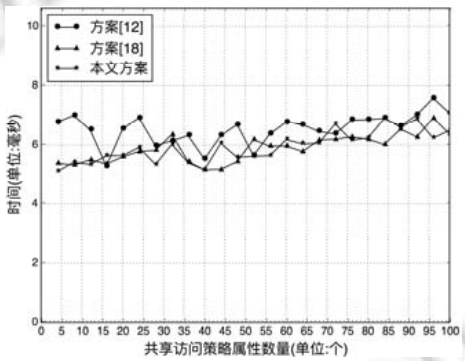

Fig.5 Comparison of decryption time in user client (Android)

图 5 用户端解密时间对比图(安卓)

本文方案与 Proxy 方案、文献[12]中的方案、文献[18]中的方案在云服务器端解密时间对比如图 6 所示,几种方案在云服务器端的解密时间都随着访问策略属性的增加而逐渐增长,但是当共享访问策略属性数量增加时,本文方案具有明显优势.

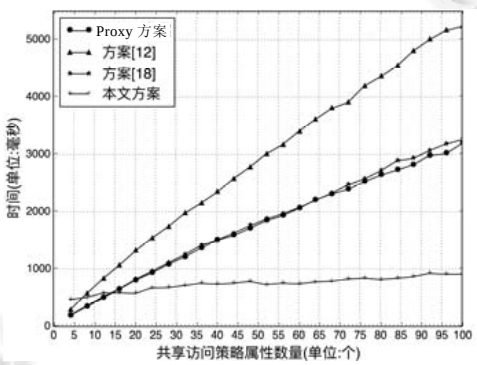

Fig.6 Comparison of decryption time in cloud server

图 6 云服务器端解密时间对比图

当用户客户端为 PC 机时,本文方案与 BSW 方案、文献[12]中的方案、文献[18]中的方案总解密时间对比如图 7 所示;当用户客户端为移动设备时,本文方案与文献[12]中的方案、文献[18]中的方案总解密时间对比如图 8 所示.无论是使用 PC 机还是移动设备,几种方案的解密时间都随着访问策略属性的增加呈现上升趋势,但是在两种环境下,本文方案在共享访问策略数量增加的情况下都具有明显优势,且当访问策略属性的数量达到

100 个时,本文方案总解密时间在 1s 左右,满足用户响应时间要求.

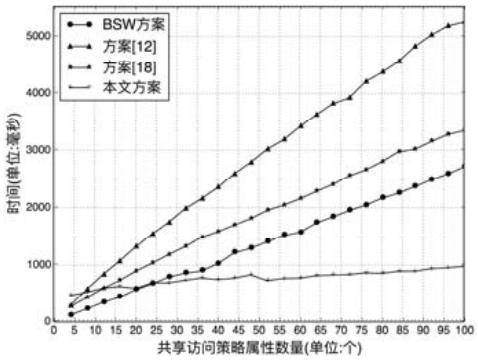

Fig.7 Comparison of total decryption time (PC)

图 7 总解密时间对比图(PC)

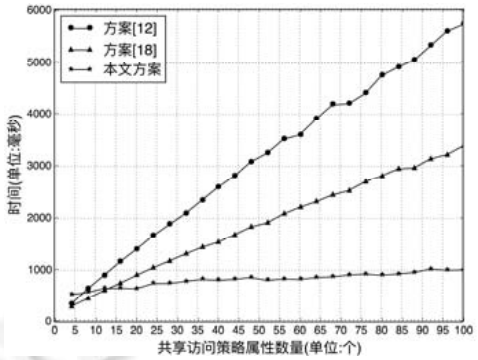

Fig.8 Comparison of total decryption time (Android)

图 8 总解密时间对比图(安卓)

6 结束语

现有的 CP-ABE 方案在解密时普遍存在计算量过大、计算时间过长等问题,难以在用户终端特别是运算能力较弱的移动终端上实施.为了减少用户客户端的负担,一些将解密外包给云服务器的 CP-ABE 方案被提出.这些方案虽然明显降低了用户终端解密时的计算量,但计算时间过长问题并没有得到有效解决.针对该问题,将 Spark 集群技术和并行计算技术引入到 CP-ABE 方案的设计之中,提出了一种面向公有云的支持快速解密的 CP-ABE 方案.在该方案中,将计算量较大的求解共享访问树根节点对应的加密共享数的任务分解为若干个可并行执行的 Map 任务和 Reduce 任务.Map 任务负责拉格朗日系数的计算或叶子节点加密共享数的计算,Reduce 节点负责计算出叶子节点对应的幂值.最后,计算出的幂值的乘积即为共享访问树根节点对应的加密共享数.性能分析表明,与 BSW 方案、文献[12]中的方案、文献[18]中的方案相比,所提出的方案在解密速度上有显著提高;而安全性分析表明,所提出方案在一般群模型和随机预言模型下能对抗选择明文攻击.

References:

[1] Feng CS, Qin ZG, Ding Y, Yu Q. Key techniques of access control for cloud computing. Acta Electronica Sinica, 2015,43(2): 312−319 (in Chinese with English abstract). [doi: 10.3969/j.issn.0372-2112.2015.02.017]

[2] Feng DG, Zhang M, Zhang Y, Xu Z. Study on cloud computing security. Ruan Jian Xue Bao/Journal of Software, 2011,22(1): 71−83 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/3958.htm [doi: 10.3724/SP.J.1001.2011.03958]

[3] Feng CS, Qin ZG, Yuan D. Techniques of secure storage for cloud data. Chinese Journal of Computers, 2015,38(1):150−163 (in Chinese with English abstract). [doi: 10.3724/SP.J.1016.2015.00150]

[4] Sahai A, Waters B. Fuzzy identity based encryption. In: Proc. of the Advances in Cryptology, Eurocrypt. LNCS, Springer-Verlag, 2005. 457−473. [doi: 10.1007/11426639_27]

[5] Bethencourt J, Sahai A, Waters B. Ciphertext-Policy attribute-based encryption. In: Proc. of the 2007 IEEE Symp. on Security and Privacy. Washington: IEEE Computer Society, 2007. 321−334. [doi: 10.1109/SP.2007.11]

[6] Goyal V, Pandey A, Sahai A, Waters B. Attribute-Based encryption for fine-grained access control of encrypted data. In: Juels A, Wright RN, Vimercati SDC, eds. Proc. of the 13th ACM Conf. on Computer and Communications Security (CCS 2006). Alexandria: ACM, 2006. 89−98. [doi: 10.1145/1180405.1180418]

[7] Ostrovsky R, Sahai A, Waters B. Attribute-Based encryption with non-monotonic access structures. In: Proc. of the 14th ACM Conf. on Computer and Communications Security. New York: ACM, 2007. 1−17. [doi: 10.1145/1315245.1315270]

[8] Cheung L, Newport C. Provably secure ciphertext policy ABE. In: Proc. of the 14th ACM Conf. on Computer and Communications Security. New York: ACM, 2007. 456−465. [doi: 10.1145/1315245.1315302]

[9] Goyal V, Jain A, Pandey O, Sahai A. Bounded ciphertext policy attribute based encryption. In: Proc. of the 35th Int'l Colloquium on Automata, Languages and Programming. Berlin: Spring-Verlag, 2008. 579−591. [doi: 10.1007/978-3-540-70583-3_47]

[10] Li J, Ren K, Zhu B, Wan Z. Privacy-Aware attribute-based encryption with user accountability. In: Proc. of the Int'l Conf. on Information Security. Berlin: Springer-Verlag, 2009. 347−362. [doi: 10.1007/978-3-642-04474-8_28]

[11] Waters B. Ciphertext-Policy attribute-based encryption: An expressive, efficient, and provably secure realization. In: Proc. of the Public Key Cryptography (PKC 2011). Berlin: Springer-Verlag, 2011. 53−70. [doi: 10.1007/978-3-642-19379-8_4]

[12] Green M, Hohenberger S, Waters B. Outsourcing the decryption of ABE ciphertexts. In: Proc. of the 20th Usenix Conf. on Security. San Francisco: ACM, 2011. 34−34.

[13] Li J, Jia C, Li J, *et al*. Outsourcing encryption of attribute-based encryption with MapReduce. In: Proc. of the 14th Int'l Conf. on Information and Communications Security. Berlin: Springer-Verlag, 2012. 191−201. [doi: 10.1007/978-3-642-34129-8_17]

[14] Zhou Z, Huang D. Efficient and secure data storage operations for mobile cloud computing. In: Proc. of the 8th Int'l Conf. on Network and Service Management. Austria: IEEE, 2012. 37−45.

[15] Lai J, Deng RH, Guan C, Weng J. Attribute-based encryption with verifiable outsourced decryption. IEEE Trans. on Information Forensics and Security, 2013,8(8):1343−1354. [doi: 10.1109/TIFS.2013.2271848]

[16] Qin B, Deng R, Liu S, Ma S. Attribute-Based encryption with efficient verifiable outsourced decryption. IEEE Trans. on Information Forensics and Security, 2015,10(7):1384−1393. [doi: 10.1109/TIFS.2015.2410137]

[17] Lin S, Zhang R, Ma H, Wang M. Revisiting attribute-based encryption with verifiable outsourced decryption. IEEE Trans. on Information Forensics & Security, 2015,10(10):2119−2130. [doi: 10.1109/TIFS.2015.2449264]

[18] Mao X, Lai J, Mei Q, *et al*. Generic and efficient constructions of attribute-based encryption with verifiable outsourced decryption. IEEE Trans. on Dependable & Secure Computing, 2016,13(5):533−546. [doi: 10.1109/TDSC.2015.2423669]

[19] Zhang K, Ma J, Liu J, *et al*. Adaptively secure multi-authority attribute-based encryption with verifiable outsourced decryption. Science China Information Sciences, 2016,59(9):99−105. [doi: 10.1007/s11432-016-0012-9]

[20] Liu Z, Jiang ZL, Wang X, *et al*. Offline/Online attribute-based encryption with verifiable outsourced decryption. Concurrency & Computation Practice & Experience, 2017. [doi: 10.1002/cpe.3915]

[21] Boneh D, Franklin M. Identity-Based encryption from the Weil pairing. Siam Journal on Computing, 2001,32(3):213−229. [doi: 10.1137/S0097539701398521]

[22] Dan B, Lynn B, Shacham H. Short signatures from the Weil pairing. Journal of Cryptology, 2004,17(4):297−319. [doi: 10.1007/s00145-004-0314-9]

[23] Rockafellar RT. Lagrange multipliers and optimality. Siam Review, 1993,35(2):183−238. [doi: 10.1137/1035044]

附中文参考文献:

[1] 冯朝胜,秦志光,袁丁,卿昱.云计算环境下访问控制关键技术.电子学报,2015,43(2):312−319. [doi: 10.3969/j.issn.0372-2112.2015.02.017]

[2] 冯登国,张敏,张妍,徐震.云计算安全研究.软件学报,2011,22(1):71−83. http://www.jos.org.cn/1000-9825/3958.htm [doi: 10.3724/SP.J.1001.2011.03958]

[3] 冯朝胜,秦志光,袁丁.云数据安全存储技术.计算机学报,2015,38(1):150−163. [doi: 10.3724/SP.J.1016.201§.00150]

邹莉萍(1994－),女,四川乐山人,硕士生,主要研究领域为信息安全,云计算,大数据安全.

袁丁(1967－),男,博士,教授,主要研究领域为密码学,信息安全.

冯朝胜(1971－),男,博士,教授,CCF 高级会员,主要研究领域为网络与信息安全,云计算,大数据安全.

罗王平(1993－),男,硕士生,主要研究领域为信息安全,云计算,大数据安全.

秦志光(1956－),男,博士,教授,博士生导师,CCF 杰出会员,主要研究领域为信息安全,分布式计算.

李敏(1978－),女,博士,副教授,主要研究领域为隐私保护,智能学习.