doi:10.11959/j.issn.1000-436x.2016158

支持关键词任意连接搜索的属性加密方案

宋衍[1,2]，韩臻[1]，陈栋[1,2]，赵进华[2]
(1.北京交通大学计算机与信息技术学院，北京 100044；2.信息保障技术重点实验室，北京 100072)

摘 要：构建一种基于素数阶双线性群的可搜索加密方案。基于属性加密，实现每个关键词密文能够被多个用户私钥搜索，显著降低细粒度访问控制带来的网络带宽和发送节点的处理开销。基于多项式方程，支持对关键词的任意连接搜索，显著提高连接搜索的灵活性。对方案的性能进行了分析，并与现有的连接关键词搜索方案进行了比较。

关键词：可搜索加密；属性加密；连接关键词；多项式方程
中图分类号：TP309.7 **文献标识码**：A

Attribute-based encryption supporting arbitrary conjunctive key word search

SONG Yan[1,2], HAN Zhen[1], CHEN Dong[1,2], ZHAO Jin-hua[2]
(1. School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China;
2. Technology on Information Assurance Key Laboratory, Beijing 100072, China)

Abstract: A new searchable encryption scheme was proposed in prime order bilinear groups based on the attribute-based encryption and polynomial equation. The scheme, in which each conjunctive-keyword ciphertext can be searched by a number of users, may significantly reduce the overhead of network and sending nodes' computation in the application of fine-grained access control.Meanwhile, the scheme facilitates the flexibility of conjunctive search by supporting arbitrary conjunctive search of the keywords. At last, the performance was analyzed and compared with some recent conjunctive search schemes.

Key words: searchable encryption, attribute-based encryption, conjunctive keyword, polynomial equation

1 引言

随着云计算的快速发展，数据外包成为一种广泛的应用形式。数据使用者只是享受云环境提供的数据服务，而不用购买和维护自己的信息系统。这样一是能够提高数据管理的专业化程度，改善数据应用效率，二是用户能够根据需要随意地增加和裁减服务的功能和容量，降低信息化投入。但是用户在享受云存储服务的同时，需要将数据交由云端的服务提供者来进行存储和处理。其最主要的特征是：数据与其所有者分离，处于所有者不可控的区域内[1]。

在不可控环境中，加密是保证数据安全的最有效方法。但是，加密破坏了原有数据的值和大小关系等，密文不再具有可供检索的语义和统计特性，因此，针对明文的检索技术并不能直接应用于密文。传统的处理方式是：服务提供者将所有满足访问控制策略的密文都返回给请求者，由请求者解密后再进行明文的检索。这种方法简单易行，但是，一方面，云端的计算和存储等资源优势无法得到有

收稿日期：2015-10-23；**修回日期**：2016-07-29
基金项目：国家自然科学基金资助项目（No.60973112）；北京市教育委员会学科建设与研究生培养基金资助项目（No.BMKY2011B06）
Foundation Items: The National Natural Science Foundation of China (No.60973112), Discipline Construction and Graduate Education Foundation of Beijing Municipal Commission of Education (No.BMKY2011B06)

效利用，另一方面，返回大量数据需要占用相当多的网络带宽，解密也会给请求者带来很大的计算量，因此，对客户机配置和网络质量有很高的要求，同时其效率也是不能被接受的。

为了更好地解决这个问题，可搜索加密（searchable encryption）应运而生，并在近几年中得到了广泛研究和快速发展[2]。2000 年，Song 等[3]提出了实用的可搜索对称加密方案，仅需要用户与服务器之间的一次交互，每项检索耗费的工作时间是线性的。对称可搜索加密方案的优点是效率高、安全性好，但缺点是仅适用于用户检索自己事先存储到云端服务器上的密文数据。2004 年，Boneh 等[4]利用公钥密码算法构造了一个可搜索加密方案（PEKS），解决了对其他用户的加密数据进行检索的难题。

鉴于一次搜索多个关键词能够缩小搜索范围改进搜索性能[5]，Golle 等[6]提出了关键词连接搜索的概念和安全模型，给出了 2 种连接关键词搜索方案；但是，该方案基于对称密钥加密和搜索关键词，应用环境受限。2005 年，Park 等[7]基于双线性映射提出了一种公钥密码系统连接关键词搜索方案（PKCKS）。Hwang 等[8]也基于双线性映射设计了一种 PKCKS 方案，并且将方案扩展到多用户环境；但是，采用多对公私钥的形式，给密钥管理带来较大负担。2007 年，Boneh 等[9]提出的基于谓词加密的概念，利用合数阶双线性群构造了一种支持连接关键词的可搜索加密方案，并扩展到子集和比较搜索；但是，方案需要较大的群空间。Katz 等[10]介绍了一种基于内积向量的谓词加密可搜索方案，并给出了在匿名身份加密、隐藏向量加密、关键词合取（连接关键词）、关键词析取和多项式方程搜索等方面的实现思路。Chen 等[11]提出了一种基于时间戳的 PKCKS，能够减小服务器在搜索时的运算量；但是，服务器需要利用密文生成时间戳，而数据请求者需要获取每个时间戳的一个组件来生成相应的陷门。Zhang 等[12]提出一种支持连接关键词子集搜索的 PKCKS，但是用户计算量较大。Yang 等[13]提出一种指定搜索者、支持代理重加密的 PKCKS 方案，能够抵御关键词猜测攻击（keyword guessing attack）。Chen 等[14]使用合数阶群和素数阶群分别提出了 2 种 PECKS 方案，但无法实现关键词域的子集搜索，如果需要支持所有可能的连接搜索，就会导致密文成指数增长。Wang 等[15]采用授权用户和云端服务器先后对关键词加密的方式，基于伪随机函数提出了一种连接关键词搜索方案，陷门大小固定，适用于多用户环境；但是，如果服务提供者与任意一个数据请求者共谋，则能够破解方案的主密钥，同时，基于对称密钥来加密和搜索关键词，应用环境受限。

目前，绝大多数的连接关键词搜索方案采用了 Golle[6]的假设，设置关键词域来对位搜索和被搜索的关键词。这种方式对于结构化、语义确定的数据具有较高的搜索效率，如邮件的首部信息；但是，对于非结构化的数据，则很难用固定的关键词域覆盖所有文件的准确内容[16]。因此，在实用性上有一定限制。

除此之外，已有可搜索加密方案大都针对数据拥有者与请求者是“一对一”的情形，在越来越多的信息共享的大背景下，这些方案显然不适合多方搜索的需求。属性加密将应用场景扩展到“一对多”的情况，一个密文可以选择性地由多个特定用户解密。Li 等[17]基于密钥策略的属性加密实现了一种单关键词的可搜索加密方案，但是在陷门中，关键词组件和访问树组件没有关联，容易被服务提供者和数据请求者的共谋所仿冒。Zheng 等[18]基于密钥策略和密文策略的属性加密分别提出了 2 种可搜索加密方案，并且支持搜索的可验证。到现在为止，还没有支持连接关键词搜索的属性加密方案。

针对以上 2 个问题，本文提出一种新的连接关键词可搜索属性加密方案，文档无需按照关键词域来设置关键词，搜索运算时无需对位密文和陷门中的关键词。方案基于属性关联实现搜索的细粒度访问控制，基于多项式方程实现关键词的任意连接搜索。具有较高的安全性，能够抵御选择关键词攻击（CKA, chosen keyword attack）。

2 预备知识

2.1 双线性群及复杂性假设

假设 p 为素数，G_1、G_2 和 G_T 为 p 阶循环群，g_1、g_2 和 g_T 分别为其生成元。e: $G_1\times G_2\rightarrow G_T$ 为双线性映射，则有：

1) $\forall h_1\in G_1, h_2\in G_2$，$\forall a, b\in Z_p$，满足 $e(h_1^a, h_2^b)=e(h_1, h_2)^{ab}$；

2) $e(g_1, g_2)\neq 1$；

3) G_1、G_2 和 G_T 中的运算以及双线性映射 e 都是在多项式时间内可完成的。

假设 1　m 维判定性 Diffie-Hellman（m-DDH）问题。随机选择 $a\in Z_p^*\backslash\{1\}$，$x\in G$，给定元组 $(g,g^a,g^{a^2},\cdots,g^{a^m},x)$，判断等式 $x=g^{a^m}$ 是否成立是困难的。

2.2　系统模型

系统包括 4 个参与角色，记为{UM, Serv, DO, DReq}。其中，UM 是授权管理机构，负责系统的初始化，并管理用户的属性及其密钥生成；Serv 是云端的服务提供者，负责数据的存储和搜索；DO 是数据拥有者，将自己的数据加密后交由 Serv 存储和管理；DReq 是数据请求者，向 Serv 请求数据搜索服务。UM 是完全可信的，并且 UM 与 DReq 的对话是安全的；Serv 是“诚实但狡猾”的，能够忠实地执行规定的操作，但可能会窃取数据的信息，或与 DReq 发起共谋攻击。方案的系统模型及工作流程如图 1 所示。

1) UM 管理所有用户的属性集合；UM 调用初始化算法生成公开参数和主私钥；公开参数用于数据加密、用户私钥生成、陷门生成和密文搜索算法，发布给系统的合法用户；主私钥用于生成用户私钥，由 UM 保存。

2) DO 向 UM 申请系统的公开参数。

3) UM 将公开参数返回给 DO。

4) DO 从文档中选出关键词集合，调用对称算法加密文档，对称加密算法选用现有的 3DES 或 AES 等即可。

5) DO 制定访问控制策略，调用属性加密公钥算法分别加密对称加密密钥和关键词集合。

6) DO 将文档密文、对称密钥密文和关键词集合密文组成数据密文，发送给 Serv 存储。

7) DReq 如果已经申请过用户私钥，则进入 10)，否则向 UM 申请用户私钥。

8) UM 调用私钥生成算法，生成与用户属性集匹配的用户私钥。

9) UM 将用户私钥通过安全方式返回给 DReq，如用 DReq 的公钥加密。

10) DReq 调用陷门生成算法，生成被搜索关键词集合相应的陷门；陷门是关键词集合的一个单向函数值，能够用于关键词的判断，而不会泄露关键词的信息。

11) DReq 将自身的属性集和陷门发送给 Serv。

12) Serv 首先根据 DReq 的属性集筛选满足访问控制策略的数据密文，然后调用搜索算法，对搜索关键词集合的密文和被搜索关键词集合的陷门进行运算，验证同时满足访问控制策略和搜索策略的数据密文；如果陷门所属的 DReq 满足密文的访问控制策略，并且陷门的关键词集合包含于密文所代表的关键词集合，则结果为一个固定值，否则结果为随机值。

13) 将满足要求的文档密文和对称密钥密文返回给 DReq。

14) DReq 利用自己掌握的用户私钥，调用属性加密公钥算法，解密对称密钥；如果 DReq 满足访问控制策略要求，则能够得出正确的对称密钥，接

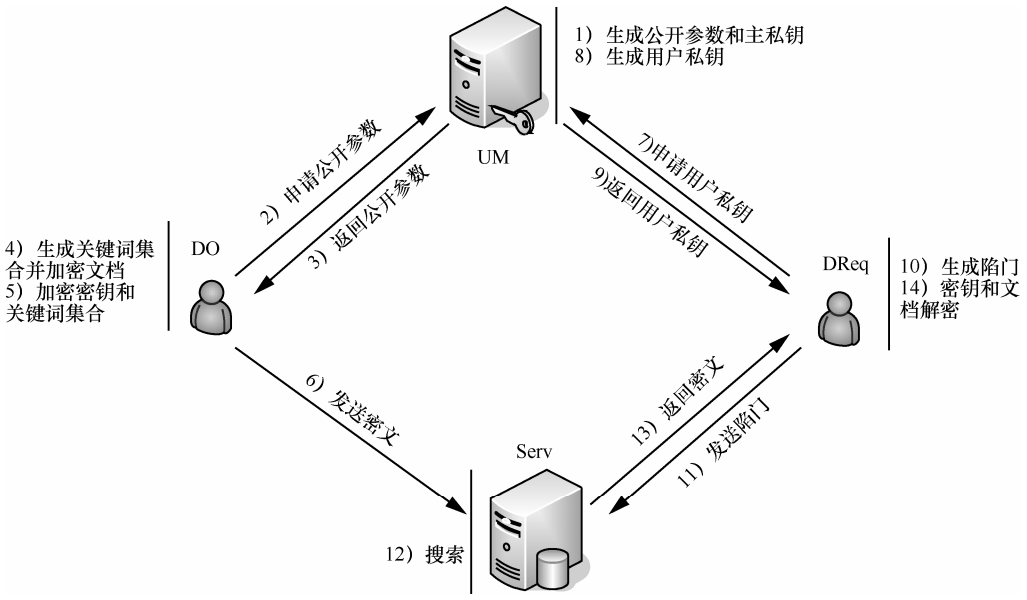


图 1　系统模型及工作流程

着调用对称加密算法解密出文档明文，否则，不能得出正确的对称密钥，无法解密出文档明文。

在 12) 中，虽然搜索算法能够辨别出密文是否同时满足访问控制策略和搜索策略，但是由 Serv 首先根据 DReq 的属性集筛选满足访问控制策略的数据密文，则可以在计算之前就过滤掉不满足访问控制策略的数据密文，从而减少搜索算法的调用次数，降低计算开销。

在 9)中对用户私钥的公钥算法加密，在 4)、14)中对文档的对称算法加解密以及第 5)、14)中对对称密钥的属性算法加解密，均有成熟技术可用，这里不再做研究。因此，本文之后的章节将忽略对文档及其对称密钥的处理，而只考虑关键词的处理。经过简化后，一个支持关键词任意连接搜索的基于属性加密方案由以下 5 种多项式时间算法组成。

1) $Setup(1^{\lambda})\rightarrow(pm,mk)$: UM 执行该算法以初始化系统；输入一个安全参数 1^{λ}，输出一个公开参数 pm 和一个主私钥 mk。

2) $KeyGen(Atts,mk,pm)\rightarrow sk$: UM 根据 DReq 提供的属性集为其生成相应的用户私钥；输入公开参数 pm、主私钥 mk 和 DReq 的属性集 $Atts$，输出对应的私钥 sk。

3) $Encrypt(pm,W,T)\rightarrow cph$: DO 根据访问控制策略加密文档的关键词集合；输入公开参数 pm、文档的关键词集合 W，以及访问结构 T，输出密文 cph。

4) $TokenGen(sk,pm,W')\rightarrow tk$: DReq 生成搜索陷门；输入公开参数 pm，用户私钥 sk，关键词集合 W'，输出关键词集合的陷门 tk。

5) $Search(tk,cph)\rightarrow 1$:Serv 根据陷门对密文进行搜索；输入陷门 tk 和密文 cph，若用户的属性集 $Atts$ 满足密文的访问结构 T，并且搜索的关键词集合 W' 包含于文档的关键词集合 W，则返回 1，否则返回随机值。

方案特点为：1)每个密文（本文以后提到的密文都指文档关键词集合的密文）都可以指定多个不同的用户私钥进行搜索；2)每个用户私钥都可以对不同的密文（这些密文可以由不同的 DO 生成）进行搜索；3)每个密文都支持对其中关键词的任意连接的搜索。

2.3 访问控制

使用树形结构表示访问控制策略，具有清晰、简便的优点[19]。访问树的内部节点代表关系（门），包括 and（与）、or（或）和 threshold（门限）；叶子节点代表属性。假设 num_v 为节点 v 下子节点的个数，k_v 为节点 v 的门限值，则有 $1\leqslant k_v\leqslant num_v$。2 种特别的情况是，当 $k_v=1$ 时，节点 v 代表关系 or；当 $k_v=num_v$ 时，节点 v 代表关系 and。使用 $parent(v)$ 表示节点 v 的父节点，$ind(v)$ 表示节点 v 在父节点下的索引号，$lvs(T)$ 表示访问树 T 中所有叶子节点组成的集合，$att(v)$ 表示叶子节点 v 所代表的属性，T_v 表示 T 中根为 v 的子树。

给定一个属性集合 $Atts$，$F(Atts,T_v)=1$ 表示 $Atts$ 满足树 T_v 代表的访问控制策略，$F(Atts,T_v)=0$ 表示 $Atts$ 不满足树 T_v 代表的访问控制策略。那么 $F(Atts,T_v)$ 的取值可以通过下列方式递归确定。

1) 当 v 是叶子节点时，如果 $att(v)\in Atts$，则设置 $F(Atts,T_v)=1$；否则 $F(Atts,T_v)=0$。

2) 当 v 是内部节点时，假设 $v_1,v_2,\cdots,v_{num}$ 为 v 的子节点，如果存在一个子集 $I\subseteq\{1,\cdots,num_v\}$ 使 $|I|\geqslant k_v$，并且对于 $\forall j\in I$，有 $F(Atts,T_{v_j})=1$，那么设置 $F(Atts,T_v)=1$；否则，设置 $F(Atts,T_v)=0$。

对于访问树 T，本文将 T 的秘密共享算法表示为

$$\{q_v(0)|v\in lvs(T)\}\leftarrow Share(T,q) \tag{1}$$

算法从上至下为 T 中的每个内部节点 v 构造一个 k_v-1 次一元多项式 q_v，并为每个内部节点和叶子节点赋值。

1) 如果 v 是 T 的根节点，设置 $q_v(0)=q$，并为多项式 q_v 随机选取 k_v-1 个系数。

2) 如果 v 是 T 中除根节点外的其他内部节点，设置 $q_v(0)=q_{parent(v)}(ind(v))$，并为多项式 q_v 随机选取 k_v-1 个系数。

3) 如果 v 是 T 的叶子节点，设置 $q_v(0)=q_{parent(v)}(ind(v))$。

给定访问树 T 和一个秘密值集合 $\{E_{u_1},E_{u_2},\cdots,E_{u_m}\}$，其中，$u_1,u_2,\cdots,u_m$ 为 T 的叶子节点，则满足 $F(att(u_1),\cdots,att(u_m),T)=1$。对于 $\forall j\in\{1,2,\cdots,m\}$，本文定义 $E_{u_j}=e(g_1,g_2)^{q_{u_j}(0)}$，并且将 $e(g_1,g_2)^q$ 的重构算法表示为

$$e(g_1,g_2)^q\leftarrow Combine(T,\{E_{u_1},E_{u_2},\cdots,E_{u_m}\}) \tag{2}$$

根据访问树 T 的结构，算法从底向上执行如下步骤。

对于节点 v，如果 $F(att(u_1),\cdots,att(u_m),T_v)=0$，则继续；

否则，$F(att(u_1),\cdots,att(u_m),T_v)=1$，则

如果 v 是叶子节点，则存在一个 u_j，使 u_j=v；设置 $E_v = E_{u_j}(0) = e(g_1,g_2)^{q_{u_j}(0)}$；

否则，v 是内部节点；那么对于 v 的子节点集合 $\{v_1,v_2,\cdots,v_{num_v}\}$，存在一个子节点索引的集合 S，使 $|S|=k_v$，并且对于 $\forall j \in S$，有 $F(att(u_1),\cdots, att(u_m),T_{v_j})=1$；使用 $\Delta u_j = \prod_{l\in S,l\neq j}\frac{-j}{l-j}$ 表示系数，则 $E_v = \prod_{l\in S}E_{v_j}^{\Delta v_j} = \prod_{l\in S}\left[e(g_1,g_2)^{q_{v_j}(0)}\right]^{\Delta v_j(0)} = e(g_1,g_2)^{q_v(0)}$。

当算法结束时，得到 T 的根节点的秘密值 $E_{\text{root}} = e(g_1,g_2)^{q_{\text{root}}(0)} = e(g_1,g_2)^q$。

3　方案设计

3.1　方案构造

DO 可以任意设置文档的关键词集合，没有域的划分，没有关键词个数、位置的限制。假设 H:$\{0,1\}^*\rightarrow Z_p$ 是一个安全的单向散列函数，将位串随机的映射到群 Z_p 中。G_1、G_2 和 G_T 为 p 阶循环群，g_1、g_2 和 g_T 分别为其生成元，e:$G_1\times G_2\rightarrow G_T$ 为双线性映射。方案的详细描述如下。

1) *Setup*(1^λ)。运行算法 $\mathcal{G}$ (1^λ) 获得 (p,G_1,G_2, G_T,e)。随机选取 $b,c,d\in Z_p^*$，发布公开参数为

$$pm=(e,p,g_1,g_2,g_1^b,g_1^c,g_1^d,g_2^c,g_2^d,G_1,G_2,G_T) \tag{3}$$

保存主密钥为

$$mk=(b,c,d) \tag{4}$$

2) *KeyGen*(*Atts*,*mk*,*pm*)。构造属性集 *Atts* 对应的私钥。随机选择 $r\in Z_p^*$，计算 $K=g_2^{\frac{bc-r}{d}}$。对于每个 $att_j\in Atts$，随机选择 $r_j\in Z_p^*$，计算 $A_j = g_2^r g_2^{H(att_j)r_j}$，$B_j = g_2^{r_j}$。构造私钥如下

$$sk=(Atts,K,\{(A_j,B_j)|att_j\in Atts\}) \tag{5}$$

3) *Encrypt*(*pm*,*W*,*T*)。加密关键词集合 W=$\{w_1,\cdots,w_m\}$。随机选取 $q\in Z_p$ 作为访问树 T 的秘密共享值，执行$\{q_v(0)|v\in lvs(T)\}\leftarrow Share(T,q)$，使 T 中的每个叶子节点 v 得到一个关于 q 的秘密共享值 $q_v(0)$，并为每个叶子节点计算 $C_v = g_1^{q_v(0)}$ 和 $D_v = g_1^{H(att(v))q_v(0)}$。随机选择 $a,k\in Z_p^*$，构造一个 m 次多项式为

$$\begin{aligned} f(x)&=a(x-H(w_1))(x-H(w_2))\cdots(x-H(w_m))+k\\ &=a_m x^m+\cdots+a_1x+a_0 \end{aligned} \tag{6}$$

对于每个 $i\in\{0,1,\cdots,m\}$，计算 $F_i = g_1^{ca_i}$，并计算 $W_0 = g_1^{bq}g_1^{dk}$ 和 $W_1 = g_1^{dq}$。构造密文为

$$\begin{aligned} cph=&(T,W_0,W_1,\{F_i|i\in\{0,1,2,\cdots,m\}\},\\ &\{(C_v,D_v)|v\in lvs(T)\}) \end{aligned} \tag{7}$$

4) *TokenGen*(*Atts*,*sk*,*pm*,W')。构造关键词集合 $W'=\{w_1',\cdots,w_t'\}$ 对应的陷门，其中，$t\leqslant m$。随机选取 $s\in Z_p^*$，计算 $tok_0 = g_2^{cs}$，tok_1=K^s，对于每个 $att_j\in Atts$，计算 $A_j' = A_j^s, B_j' = B_j^s$，这部分工作在线下提前完成。对于每个 $i\in\{0,1,\cdots,m\}$，计算 $H_i = g_2^{\frac{H(w_1')^i+\cdots+H(w_t')^i}{t}}$。构造关键词陷门为

$$\begin{aligned} tk=&(Atts,tok_0,tok_1,\{H_i\,|\,i\in\{0,1,\cdots,m\}\},\\ &\{(A_j',B_j')\,|\,att_j\in Atts\}) \end{aligned} \tag{8}$$

5) *Search*(*tk*,*cph*)。根据密文中的访问树 T 和陷门 *tk* 中的属性集 *Atts*，服务器从 *Atts* 中选择一个满足 T 的属性子集 S。如果 S 不存在，则返回 0；否则，对于每个 $att_j\in S$，找到与之相应的 $att(v)=att_j$，计算 T 中叶子节点的秘密值为

$$Ev = \frac{e(A_j',C_v)}{e(B_j',D_v)} = \frac{e(g_2^{rs}g_2^{H(att_j)r_js},g_1^{q_v(0)})}{e(g_2^{r_js},g_1^{H(att(v))q_v(0)})} = e(g_1,g_2)^{rsq_v(0)} \tag{9}$$

然后计算根节点的秘密值

$$\begin{aligned} E_{\text{root}} &= Combine(T,\{E_v\,|\,att_v\in S\})\\ &= e(g_1,g_2)^{rsq_{\text{root}}(0)} = e(g_1,g_2)^{rqs} \end{aligned} \tag{10}$$

如果 $e(W_0,tok_0) = e(W_1,tok_1)E_{\text{root}}\prod_{i=0}^m e(F_i,H_i)$，则返回 1，否则返回 0。

3.2　正确性验证

做如下计算

$$e(W_0,tok_0) = e(g_1^{bq}g_1^{dk},g_2^{cs}) = e(g_1,g_2)^{bqcs+cdks} \tag{11}$$

$$e(W_1,tok_1) = e(g_1^{bq},g_2^{\frac{(bc-r)s}{d}}) = e(g_1,g_2)^{bqcs-rqs} \tag{12}$$

$$\begin{aligned} \prod_{i=0}^m e(F_i,H_i) &= \prod_{i=0}^m e(g_1^{ca_i},g_2^{ds\frac{H(w_1')^i+\cdots+H(w_t')^i}{t}})\\ &= \prod_{i=1}^t e(g1,g2)^{cds\frac{amH(w_i')^m+\cdots+a_1H(w_i')+a_0}{t}} \end{aligned} \tag{13}$$

当搜索的关键词集合包含于文档的关键词集合，即 $W'\in W$ 时，有 $\prod_{i=0}^m e(F_i,H_i) = e(g_1,g_2)^{cdsk}$；当属性集 *Atts* 满足访问树 T，即 $F(Atts,T)$=1 时，有 $E_{\text{root}} = e(g_1,g_2)^{rqs}$。因此

$$\frac{e(W_0,tok_0)}{e(W_1,tok_1)E_{\text{root}}\prod_{i=0}^m e(F_i,H_i)} = 1 \tag{14}$$

4 方案分析

4.1 安全性分析

实现访问控制的基本思想是：将每个陷门和密文分为 2 部分：1) 与关键词明文相关联；2) 与属性相关联，陷门中的属性表示 DReq 的属性集，密文中的属性表示 DO 的访问控制策略。如果 DReq 的属性集满足 DO 的访问控制策略，服务器能够通过双线性映射算法确定密文的关键词集合是否包含陷门的关键词集合。实现任意连接搜索的基本思想是：利用文档的所有关键词和 2 个随机数构造一个次数等于关键词个数的多项式，将多项式的系数以幂的形式发布；如果所搜索的关键词都包含在文档的关键词集合中，则能够利用拉格朗日插值公式还原出该随机数，否则，返回结果不正确。

方案在私钥生成、密文生成和陷门生成阶段，分别使用随机数将各组件关联化和随机化。以下从 3 个方面来说明或证明方案的安全性。

1) 陷门不可伪造。DReq 随机选择大数 s，使陷门的每个组件都是以 s 为指数的幂，如此以来，每个组件都通过 s 相关联。在离散对数困难问题假设下，攻击者无法通过已用陷门中的组件 $tok_0 = g_2^{cs}$ 求解得到随机数 s 的值，并伪造新陷门。当攻击者随机选择 s 来伪造陷门时，只能构造陷门的组件 H，还必须获得用户私钥来构造陷门的组件 tok_0、tok_1、A 和 B。

2) 用户私钥不可伪造。用户私钥中的组件 B_j 通过随机数 r_j 与组件 A_j 相关联，组件 K 通过随机数 r 与 A_j 相关联。由于不同攻击者的用户私钥所选取的随机数不同，所以，即使攻击者们的属性组合能够包含某 DReq 的属性集合，也无法以共谋的方式通过组件组合获得该 DReq 的私钥。用户私钥中的组件在用于陷门时，都以随机数 s 为指数进行了幂运算，在离散对数困难问题假设下，保证了用户私钥的安全性。

3) CKA 安全性。在 m-DDH 假设下，本文方案是 CKA 安全的。下面用反证法予以证明。

证明 假设本文方案在下面的 CKA 安全游戏中是不安全的，那么存在一个 PPT 算法 $\mathcal{A}$ 能够赢得安全游戏，而本文可以构造出一个 PPT 算法 $\mathcal{S}$ 利用 $\mathcal{A}$ 攻破 m-DDH 假设。假设 g_β 为群 G_β 的生成元，随机取 $a \in Z_p^* \backslash \{1\}$，$x \in G_\beta$，生成 $\mathcal{S}$ 的挑战元组 $(g_\beta, g_\beta^a, \cdots, g_\beta^{a^{m-1}}, y)$，其中，$y = g_\beta^{a^m}$ 或 $y=x$。

1) 初始化。$\mathcal{S}$ 取 $G_2=G_\beta$ 和 $g_2 = g_\beta$，随机选取 $b,c,d \in Z_p^*$，保存主密钥 $mk=(b,c,d)$ 和公开参数 $pm = (e, p, g_1, g_2, g_1^b, g_1^c, g_1^d, g_2^c, g_2^d, G_1, G_2, G_T)$，并生成用户私钥。为了保证询问的一致性，$\mathcal{S}$ 保存元组 $(w_{i,j},x_{i,j})$的列表，记为 L。$\mathcal{A}$ 选择多项式数量的关键词集合，向 $\mathcal{S}$ 询问相应的密文。假设其中一个关键词集合为 $W_i=(w_{i,1},\cdots,w_{i,m})$，对于每个 $w_{i,j} \in W_i$，如果未被询问过，$\mathcal{S}$ 随机选择 $x_{i,j} \in Z_p$ 作为 $H(w_{i,j})$，并将元组$(w_{i,j},x_{i,j})$加入到列表 L 中；如果已经被询问过，则 $\mathcal{S}$ 返回列表 L 中相应元组的 $x_{i,j}$。接着，$\mathcal{S}$ 按照 3.1 节中的 *Encrypt* 算法生成关键词集合 W_i 的密文，并发送给 $\mathcal{A}$。

2) 询问阶段 1。$\mathcal{A}$ 询问关键词集合的陷门。假设其中一个关键词集合为 $W'_q = (w'_{q,1},\cdots,w'_{q,t})$，对于每个 $w'_{q,t} \in W'_q$，如果已经在之前询问的密文或陷门中出现过，则 $\mathcal{S}$ 返回列表 L 中相应元组的 $x_{i,j}$ 作为其 $H(w'_{q,i})$；否则随机选择 $x_{q,i} \in Z_p$ 作为其 $H(w'_{q,i})$，同样，元组 $(w'_{q,i}, x_{q,i})$ 被加入到列表 L 中保存。接下来，$\mathcal{S}$ 按照 3.1 节中的 *TokenGen* 算法生成关键词集合 W'_q 的陷门。由于当关键词 $w'_{q,i}$ 出现在不同的询问中时，$\mathcal{S}$ 前后使用同一个散列值；因此，生成的陷门是关键词集合 W'_q 的正确陷门。在收到陷门后，$\mathcal{A}$ 调用搜索算法 *Search* 对每个密文进行搜索，看关键词集合 W'_q 是否包含于某些密文的关键词集合中。

3) 挑战。在经过多项式数量的询问后，$\mathcal{A}$ 开始挑战。$\mathcal{A}$ 选择 2 个关键词集合 $W_0=\{w_{0,1},\cdots,w_{0,m}\}$ 和 $W_1=\{w_{1,1},\cdots,w_{1,m}\}$发送给 $\mathcal{S}$，要求 $\mathcal{A}$ 没有询问过 $(W_0 \backslash W_1) \cup (W_1 \backslash W_0)$中任何关键词子集的陷门。$\mathcal{S}$ 随机选择 $\beta \in \{0,1\}$，构造关键词集合 W_β 的密文。对于每个 $w_{\beta,i}(1 \leqslant i \leqslant m)$，如果已经被询问过，则 $\mathcal{S}$ 返回列表 L 中相应元组的 $x_{i,j}$ 作为其散列值，否则随机选择 $x_{\beta,i} \in Z_p$ 作为其散列值，并加入到列表 L 中保存。$\mathcal{S}$ 按照 3.1 节中的 *Encrypt* 算法生成关键词集合 W_β 的密文 C_β，并发送给 $\mathcal{A}$。

4) 询问阶段 2。$\mathcal{A}$ 继续询问关键词集合的陷门，只是不能询问$(W_0 \backslash W_1) \cup (W_1 \backslash W_0)$中任何关键词子集的陷门。假设其中一个关键词集合为 $W'_q = (w'_{q,1},\cdots,w'_{q,t})$，其中，$t \leqslant m$。$\mathcal{S}$ 随机选取 $s \in Z_p^*$，计算 $tok_0 = g_2^{cs}$，$tok_1=K^s$，对于每个 $att_j \in Atts$，计算 $A'_j = A_j^s, B'_j = B_j^s$。对于每个 $j \in \{1,\cdots,t\}$，随机

选择 $h_j \in Z_p$；对于每个 $i \in \{0,1,\cdots,m\}$，计算 $r_i = \sum_{j=1}^{t} h_j^i$；$\mathcal{S}$ 利用挑战元组 $(g_2, g_2^q, \cdots, g_2^{a_{m-1}}, y)$，对于每个 $i \in \{0,1,\cdots,m-1\}$，计算 $H_i = (g_2^{a_i})^{\frac{dsr_i}{t}}$，并计算 $H_m = y^{\frac{dst_m}{t}}$。$\mathcal{S}$ 将陷门发送给 $\mathcal{A}$，$\mathcal{A}$ 调用搜索算法 *Search* 对密文 C_β 进行搜索，看关键词集合 W_q' 是否是 W_β 的子集。

5) 猜测。最终，$\mathcal{A}$ 输出一个猜测 $\beta' \in \{0,1\}$。如果 $\beta'=\beta$，$\mathcal{S}$ 猜测 $y = g_2^{a_m}$，否则 $\mathcal{S}$ 猜测 $y=x$。

在挑战阶段，由于 $\mathcal{A}$ 不知道主密钥，并且主密钥的选择与所有的关键词无关，因此，从 $\mathcal{A}$ 的角度看来，密文中的组件 F_i 是均匀分布的，即构造的 m 次多项式 $f(x)$ 的系数是均匀分布的。在离散对数难题假设下，$\mathcal{A}$ 无法从已经询问过的密文和陷门中计算出主密钥或者 $f(x)$ 的系数。因此，在挑战阶段，$\mathcal{A}$ 无法分辨出 β 是 0 还是 1。

在询问阶段 2，如果 $y = g_2^{a_m}$，则对于 $\forall i \in \{0,1,\cdots,m\}$，有 $H_i = (g_2)^{\frac{dsr_i a^i}{t}}$，即 $H_i = (g_2)^{\frac{ds(ah_1)^i+\cdots+(ah_t)^i}{t}}$，生成的是其他某个关键词集合的陷门；否则，生成的不是正确的陷门。这就使在不询问 $(W_0 \backslash W_1) \cup (W_1 \backslash W_0)$ 中关键词子集的陷门的要求下，当 $y = g_2^{a_m}$ 时，可能生成 $(W_0 \backslash W_1) \cup (W_1 \backslash W_0)$ 中关键词子集的陷门，从而通过搜索算法分辨出 β 是 0 还是 1。因此，当 $\mathcal{A}$ 能够以一定优势分辨出 β 是 0 还是 1 时，必然是生成了关键词集合的正确陷门，有 $y = g_2^{a_m}$。所以，如果 $\mathcal{A}$ 具有分辨出密文的优势，则 $\mathcal{S}$ 具有判断 $y = g_2^{a_m}$ 是否成立的优势。

4.2　性能分析

本节将构造的方案与现有的支持连接关键词子集搜索的加密方案在通信开销、存储开销和计算开销方面做详细比较。其中，性能指标包括每个密文的长度和加密计算量，每个陷门的长度和计算量，以及 DSP 使用一个陷门针对一个密文进行搜索时的计算量。

密文长度关系到 DO 与 DSP 之间的通信开销以及 DSP 的存储开销；陷门长度关系到 DReq 与 DSP 之间的通信开销。加密计算量关系到 DO 的计算开销，陷门计算量关系到 DReq 的计算开销，搜索计算量关系到 DSP 的计算开销。

在云计算的数据外包环境中，DO 和 DReq 所在终端的计算能力、通信能力和存储能力可能受限。由于公开参数和用户私钥等信息的存储量并不大，因此，公开参数长度和用户私钥长度对用户终端的影响很小，可以忽略。但是，要求密文长度、密文计算量、陷门长度和陷门计算量尽可能小。由于所有的密文都存储在 DSP 端，并由 DSP 进行搜索计算，因此，密文长度大时，导致 DSP 存储开销大，DSP 的存储能力需要相应提高；而搜索的计算量大时，则对于 DSP 固定的计算能力来说，将带来响应时间长的问题。

算法建立、用户私钥生成是由 UM 预先实现的，它们同公开参数长度、用户私钥长度一样，与用户体验无太大关系，这里就不做比较了。

记 P 和 M 分别为循环群中的指数运算（椭圆曲线的点乘运算）和乘法运算（椭圆曲线的点加运算），E 为双线性映射运算，H 为从群 Z_p 到群 G 和 G_T（或从群 G 和 G_T 到群 Z_p）的编码运算。$|G|$ 表示群 G 和 G_T 中元素的长度，$|p|$ 表示域 Z_p 中元素的长度。m 为每个文档的关键词数量，t 为搜索的关键词的数量。v 表示属性个数，u 表示具有访问权限的 DReq 个数，i 表示系统中的用户属性总数。

连接搜索时，搜索条件是文档的关键词集合包含陷门的关键词集合。当搜索的关键词的数量大于文档的关键词的数量时，必然没有文档满足搜索条件。因此，搜索的关键词数量和文档的关键数量之间存在关系 $t \leqslant m$。

在访问树中，叶子节点代表属性，通过代表关系的内部节点进行组合，确定具有访问权限的 DReq 集合。这使访问树可以通过规模很小、数量固定的属性集来表示规模巨大、不断增加的授权用户集。访问树中的用户属性个数与属性总数之间有关系 $v \leqslant i$。i 个用户属性可以定义 2^i 种用户，v 个用户属性在通过 and 关系组合时，定义最少的允许访问用户数量，这时 $u=2^{i-v}$，通过 or 关系组合时，定义最多的允许访问用户数量，这时 $u=v2^{i-1}$。因此有 $2^{i-v} \leqslant u \leqslant v2^{i-1}$，即在大多数情况下，$u$ 远大于或者大于 v，并且随着 u 的增加，v 并不增加。

采用对称密钥加密关键词时，虽然单次加密和搜索的效率较高，但在多用户场景中，访问控制实现困难，当更改用户权限时，需要花费大量的通信开销和计算开销来更新密文和分发密钥，因此一般只适用于搜索自己存储在服务器中的数据。所以，这里没有把这类方案[6,15]列入性能对比，如表 1 所示。

表 1 多用户环境下连接关键词搜索方案的性能对比

方案	密文长度	陷门长度	加密计算量	陷门计算量	搜索计算量
文献[7]的 PKL-1	$u(m+2)\|G\|$	$\|G\|+\|p\|+\text{lb}\ m$	$mH+umE+(m+u+1)P$	$tH+P+M$	$(t-1)M+E$
文献[8]的 HL-2	$(1+m+u)\|G\|$	$3\|G\|+\text{lb}\ m$	$(2m+u+1)P+2mH+mM$	$(2t-2)M+3P$	$tM+3E$
文献[9]的 BW	$(2m+2)u\|G\|$	$(2t+1)\|G\|+\text{lb}\ m$	$(6m+3)uP+(4m+2)uM$	$(4t+1)P+3tM$	$(2t+1)E+(t+2)M$
文献[11]的 CH	$(um+1)\|G\|$	$3\|G\|+\text{lb}\ m$	$(um+1)P$	$2E+(t-1)M$	$2E+(t-1)M$
文献[12]的 ZZ	$(um+u+m+2)\|G\|+\|p\|$	$(m+3)\|G\|$	$(um+u+m+2)P+H+E$	$(2m+3)E+2M+H$	$(2m+3)E+2M+H$
本文	$(2v+m+3)\|G\|$	$(2v+m+3)\|G\|$	$(2v+m+3)P$	$(m+1)P$	$(2v+m+3)E+(v+m+1)M$

从表 1 可以看出，在密文长度和运算量上，其他方案（除了 HL-2 的密文运算量）是基于授权用户数量 u 和文档关键词数量 m 的二次函数，而本文方案是基于属性数量 v 和文档关键词数量 m 的一次函数，由于访问树中的属性数量与授权访问的用户数量没有直接关系，因此，随着用户数量的增长，本文方案并不增长，而其他方案则会线性增长；随着用户数量和关键词数量的增长，本文方案的增长率远小于其他方案。在陷门长度上，3 种方案为固定值，BW 方案是搜索关键词数量 t 的一次函数，ZZ 方案是文档关键词数量 m 的一次函数；本文方案是属性数量 v 和文档关键词数量 m 的一次函数，随着属性数量和关键词数量的增加，将大于其他方案。在陷门运算量上，前 4 种方案是搜索关键词数量 t 的一次函数，ZZ 方案和本文方案是文档关键词数量 m 的一次函数，但本文方案对每个关键词处理的操作较少，体现在函数计算上，即系数较小，因此同其他方案相比各有优势。在搜索运算量上，前 4 种方案是搜索关键词数量 t 的一次函数，ZZ 方案是文档关键词数量 m 的一次函数，本文方案是属性数量 v 和文档关键词数量 m 的一次函数，在 v 较大时，具有一定的劣势。

相比于其他方案，本文方案在陷门计算量上处于中等水平。搜索运算量和陷门长度与属性数量和关键词数量成线性关系，但属性数量相比于其可以定义的用户数量，可以忽略，一般情况下有限，为个位数；并且每个文档的关键词数量也不可能太多，因此，陷门长度和搜索运算量的开销增加有限。而在云存储环境中，如果使用其他方案，DO 需要基于所有授权用户的公钥，将关键词集合分别加密成相应的密文版本，造成密文长度和加密运算量线性增长，由于用户数量通常较大，所以 DO 的加密负担、通信负担、Serv 的存储负担以及文档内容修改所带来的重新加密负担，是不可接受的。因此，本文方案能够显著减少多用户环境下的密文长度和加密运算量，更适合于大用户量的云存储环境。其原因是在搜索机制中融入了基于属性的访问控制，使一个公钥对应多个不同的私钥，DO 通过相同公钥加密形成的密文，能够授权给不同的 DReq 进行搜索，避免了多次加密和多个密文版本。但本文方案的代价是除陷门运算量外，各项开销有所增加，主要用于属性相关的表示和运算，陷门长度和每次搜索的运算量比大多数方案稍大。但是在方案实现时，DReq 可以提前将陷门中的属性部分 $(Atts,\{(A'_j,B'_j)\,|\,att_j\in Atts\})$ 发送给 Serv，由 Serv 在空闲时段计算出权限分量 $e(g_1,g_2)^{tqs}$；在 DReq 有搜索需求并发送陷门的关键词部分 $(tok_0,tok_1,\{H_i\,|\,i\in\{0,1,\cdots,m\}\})$ 后，Serv 再将 $e(g_1,g_2)^{tqs}$ 分量、陷门的关键词部分，连同密文输入到搜索算法中，验证是否满足访问控制策略和搜索策略，从而降低搜索进行时的运算量，减小搜索响应时间。

此外，本文方案支持非域形式的关键词连接搜索，在多文档应用场景中，当文档不断增加、文档类型不断丰富时，无需根据文档语义调整关键词域，重新加密关键词，即能够支持非结构化数据的密文检索，因而具有更广阔的应用前景。

5 结束语

本文基于属性加密和多项式方程提出了一种连接关键词可搜索加密方案。方案实现了“一对多”通信模式，支持多用户环境下的细粒度访问控制，显著减小了大用户量时的密文长度和加密计算量；实现了关键词的任意连接搜索模式，支持针对非结构化数据的连接关键词检索。只是在陷门长度和搜索运算量上，增加了相应的开销。下一步将重点研究通过融入代理加密机制，使首次搜索时权限分量的计算结果，能够用于以后的搜索请求，避免每次搜索时重复发送属性部分和计算权限分量，从而减

小陷门的计算量和长度，达到降低数据请求者的计算和通信开销，以及减少搜索时运算量的目的。

参考文献：

[1] 项菲, 刘川意, 方滨兴, 等. 云计算环境下密文搜索算法的研究[J]. 通信学报, 2013, 34(7): 143-153.
XIANG F, LIU C Y, FANG B X, et al. Research on ciphertext search for the cloud environment[J]. Journal on Communications, 2013, 34(7): 143-153.

[2] 沈志荣, 薛巍, 舒继武. 可搜索加密机制研究与进展[J]. 软件学报, 2014,25(4): 880-895.
SHEN Z R, XUE W, SHU J W. Survey on the research and development of searchable encryption schemes[J]. Journal of Software, 2014, 25(4): 880-895.

[3] SONG D, WAGNER D, PERRIG A. Practical techniques for searches on encrypted data[C]//The IEEE Symposium on Security and Privacy(S&P'00). c2000:44-55.

[4] BONEH D, CRESCENZOM G D, OSTROVSKY R, et al. Public key encryption with keywordsearch[C]//The International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT2004). Interlaken, Switzerland, c2004:506-522.

[5] LEE C C, HSU S T, H M S. A study of conjunctive keyword searchable schemes[J]. International Journal of Network Security, 2013, 15(5): 321-330.

[6] GOLLE P, STADDON J, WATERS B. Secure conjunctive keyword search over encrypted data[C]//Applied Cryptography and Network Security Conference (ACNS 2004). Yellow Mountain, China, c2004:31-45.

[7] PARK D J, KIM K, LEE P J. Public key encryption with conjunctive-field keyword search[C]//The 5th Information Security Applications International Workshop (WISA 2004).Jeju Island, Korea, c2004:73-86.

[8] HWANG Y H, LEE P J. Public key encryption with conjunctive keyword search and its extension to a multi-user system[C]//The first International Conference of Pairing-BasedCryptography (Pairing 2007). Tokyo, Japan, c2007:2-22.

[9] BONEH D, WATERS B. Conjunctive, subset, and range queries on encrypted data[C]//The 4th Theory of Cryptography conference (TCC 2007). Amsterdam, The Netherlands, c2007: 535-554.

[10] KATZ J, SAHAI A, WATERS B. Predicate encryption supporting disjunctions, polynomial equations, and inner products[C]//The 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 2008). Istanbul, Turkey, c2008:146-162.

[11] CHEN Y C, HORNG G. Timestamped conjunctive keyword searchable public key encryption[C]//Fourth International Conference on Innovation Computing Information and Control (ICICIC). Kaohsiung, c2009:729-932.

[12] ZHANG B, ZHANG F G. An efficient public key encryption with conjunctive-subset keywords search[J]. Journal of Network and Computer Applications, 2011, 34(1):262-267.

[13] YANG Y, MA G D. Proxy re-encryption conjunctive keyword search against keyword guessing attack[C]//The IEEE International Conference on Computers, Communications and IT Applications. Hongkong, China, c2013:125-130.

[14] CHEN Z H, WU C Y, WANG D S, et al. Conjunctive keywords searchable encryption with efficient pairing, constant ciphertext and short trapdoor[C]//Intelligence and Security InformaticsPacific Asia Workshop (PAISI 2012). Kuala Lumpur, Malaysia, c2012:176-189.

[15] 王尚平, 刘利军, 刘亚玲. 一个高效的基于连接关键词的可搜索加密方案[J]. 电子与信息学报, 2013, 35(9): 2266-2271.
WANG S P, LIU L J, LIU Y L. An efficient conjunctive keyword searchable encryption scheme[J]. Journal of Electronics and Information Technology, 2013, 35(9): 2266-2271.

[16] KERSCHBAUM F. Secure conjunctive keyword searches for unstructured text[C]//The 5th International Conference on Network and System Security (NSS). c2011: 285-289.

[17] 李双, 徐茂智. 基于属性的可搜索加密方案[J]. 计算机学报, 2014,37(5): 1017-1024.
LI S, XU M Z. Attribute-based public encryption with keyword search[J]. Chinese Journal of Computers, 2014, 37(5):1017-1024.

[18] ZHENG Q, XU S H, ATENIESE G. VABKS: verifiable attribute-based keyword search over outsourced encrypted data[C]//Proceeding-IEEE INFOCOM. c2014:522-530.

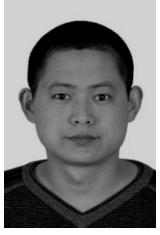
[19] GOYAL V, PANDEY O, SAHAI A, et al. Attribute-based encryption for fine-grained access control of encrypted data[C]//The 13th ACM Conference on Computer and Communications Security. Alexandria, VA, USA, c2006: 89-98.

作者简介：

宋衍（1982-），男，湖北老河口人，北京交通大学博士生，主要研究方向为密态计算、安全数据库等。

韩臻（1962-），男，浙江宁波人，博士，北京交通大学教授、博士生导师，主要研究方向为信息安全体系结构、可信计算等。

陈栋（1982-），男，山西运城人，北京交通大学博士生，主要研究方向为网络安全、软件工程等。



赵进华（1981-），男，山东菏泽人，信息保障技术重点实验室副研究员，主要研究方向为密码理论与应用。