•信息工程•

DOI:10.15961/j.jsuese.201601028

排序可验证的语义模糊可搜索加密方案

杨　旸[1,3], 杨书略[2,3], 蔡圣暐[1,3], 刘　佳[1,3], 李光澉[1,3]

(1.福州大学 数学与计算机科学学院, 福建 福州 350108; 2.福州大学 物理与信息工程学院, 福建 福州 350108;
3.网络系统信息安全福建省高校重点实验室, 福建 福州 350108)

摘　要:为解决现有语义模糊可搜索加密方案无法取得排序可验证的问题, 提出一种支持语义模糊搜索并能对搜索结果的排序进行验证的方案。首先引入相关度分数和域加权得分, 构造精确度更高的倒排索引结构。通过非线性保序加密, 云服务器可以对搜索结果进行高效排序, 减少了用户的计算开销和网络资源浪费。接着基于WordNet词典集对查询关键词进行语义拓展。在构造出语义拓展树后, 只选取最相关的语义拓展词进行查询。通过引入语义相似度, 设计双因子排序算法对搜索结果进行精确排序。然后为创建验证信息, 将关键词集合插入布隆过滤器中, 并计算布隆过滤器、安全索引、密文文档的消息认证码。在搜索阶段, 通过布隆过滤器可以验证索引中是否存在查询关键词, 并使用消息认证码对索引、文档的完整性和正确性进行验证。通过引入索引树结构进一步提高方案的搜索效率。最后, 将本方案与相关方案在功能、存储开销、搜索开销等方面进行比较, 并对创建索引、创建验证信息、语义拓展、加密查询、查询、验证等方面的开销进行仿真实验。方案对比分析显示本方案在语义搜索、排序、验证等功能上具有优势。实验结果表明, 本方案不仅实现了语义模糊搜索与结果排序, 而且实现了排序结果可验证功能。

关键词:云计算安全; 可搜索加密; 语义相似度; 域加权得分; 排序可验证

中图分类号:TP309　　**文献标志码**:A　　**文章编号**:2096-3246(2017)04-0119-10

Semantically Searchable Encryption Scheme Supporting Ranking Verification

YANG Yang[1,3], *YANG Shulue*[2,3], *CAI Shengwei*[1,3], *LIU Jia*[1,3], *LI Guangyan*[1,3]

(1.School of Mathematics and Computer Sci.,Fuzhou Univ.,Fuzhou 350108,China;
2.School of Physics and Info. Eng.,Fuzhou Univ.,Fuzhou 350108,China;
3.Key Lab. of Info. Security of Network System in Fujian Province,Fuzhou Univ.,Fuzhou 350108,China)

Abstract: In order to solve the problem that the existing semantically searchable encryption schemes could not realize ranking verification, a novel method was proposed in this paper which could not only support the semantic search,but also verify the ranking of the search results. Firstly,the relevant score and the weighted zone score were introduced to construct a more accurate inverted index structure.Using the non-linear order preserving encryption mechanism,the search results were sorted efficiently by the cloud server,which reduced the computation and communication overheads.Then,the query keywords were semantically extended based on the WordNet.Based on the semantically extended tree,the most relevant semantically extended words were selected.Combining the semantic similarity and relevance score,a two-factor ranking algorithm was designed to accurately sort the search results.Moreover,a set of keywords were inserted to the bloom filter.The message authentication codes of the bloom filter,security index and encrypted files were calculated to build the verifiable information.In the search phase,the bloom filter was used to verify the existence of the query keywords in the index.And the integrity and correctness of the index and files could be verified by the message authentication codes.The index tree was constructed to improve the search efficiency.Finally,the proposed scheme was compared with the related schemes in terms of functions,storage overheads and search overheads.And simulation experiments were carried out on the cost of building index and verifiable information,semantic extensions,encrypting queries,querying and verifying.The comparative analysis showed that the advantage of the proposed scheme in terms of semantic search,ranking and verifying.

收稿日期:2016 – 09 – 17

基金项目:国家自然科学基金资助项目(61402112; 61502086); 福建省教育厅科技项目资助(JA12028); 中央高校基本科研业务费资助(ZYGX2014J01)

作者简介:杨　旸(1984—), 女, 副教授, 博士.研究方向: 网络信息安全. E-mail: yang.yang.research@gmail.com

The experimental results demonstrated that the proposed scheme not only realizes the semantic search and the search results ranking,but also supports the ranking result verification.

Key words: cloud computing security;searchable encryption;semantic similarity;weighted zone score;ranking verification

随着云计算的发展，越来越多用户将数据外包给公有云服务器，极大减轻了本地计算和存储开销。如果将敏感数据，例如未发表的论文或专利、私人电子健康记录、公司财税报告等，以明文的形式存储在云服务器中，则有可能面临被云端管理员或攻击者窥视和窃取的风险。因此，为保障数据安全和用户隐私，许多敏感数据在外包之前需要进行加密。这又使得传统的明文搜索技术难以使用，给高效的数据利用带来挑战。为解决在密文中进行搜索的难题，Song等[1]率先开始进行可搜索加密技术的研究并提出可行的方案。

早先的可搜索加密方案主要致力于精确关键词搜索[2−7]，当用户输入的关键词匹配上预定义的关键词时，就能返回搜索结果。然而在用户输入搜索关键词时，由于粗心或遗忘等原因，很可能出现拼写错误或格式不匹配的情况。为此，Li等[8]率先提出模糊关键词可搜索加密方案，即使查询条件无法完全匹配预定义关键词，也能以较大的概率找到相关文档，极大改善用户的搜索体验。现有的模糊搜索方案中，大部分只考虑关键词字符上的模糊，忽视了关键词语义上的模糊。

为实现语义模糊搜索，Fu等[9]对文档关键词进行同义词拓展，通过计算内积的方式，实现支持同义词查询的多关键词排序搜索方案。Xia等[10]为文档集创建倒排索引，结合语义相似库对查询关键词进行语义拓展，实现了语义拓展的排序搜索方案。然而，这些方案无法对返回的结果进行验证。

在密文搜索的环境下，Chai等[11]提出“半诚实且好奇”的云服务器模型，服务器提供商为节省计算量和带宽资源，可能仅仅执行部分搜索操作或返回部分搜索结果，为此Chai等[11]提出基于单词查找树索引结构的可验证可搜索加密方案。Wang等[12]在关键词模糊搜索的基础上，通过构造符号索引树，提出可验证的关键词模糊搜索方案。Sun等[13]采用向量空间模型和MDB树结构，实现可验证的多关键词排序搜索方案。虽然这些方案都支持可验证功能，却无法支持语义模糊搜索。林柏钢等[14]基于构造校验和的方式，实现了可验证的语义模糊搜索。Fu等[15]通过TST树对关键词进行语义拓展，并同样借助符号索引树结构，提出可验证的语义模糊搜索方案。但以上可验证方案都无法验证搜索结果排序后的次序是否正确，即无法实现排序可验证的功能。

为满足用户日益增长的密文检索需求，提供一个较为完善的可搜索加密方案，提出一种排序可验证的语义模糊可搜索加密方案。本文的主要工作内容：

1）关键词语义模糊搜索：对查询关键词进行语义拓展并计算语义相似度，选取最相关的语义拓展词进行查询，实现语义模糊检索。

2）支持搜索结果的精确排序：对文档不同域中的关键词赋予不同的权重，并将其作为评估文档相关性的指标之一。通过双因子排序算法，云服务器能够对搜索结果进行精确的排序并返回给搜索用户。

3）排序可验证：通过引入布隆过滤器和消息认证码，能够对结果进行验证，特别是能够验证排序后的次序是否正确，实现了排序可验证功能。这在很多可验证方案中没有实现[11−15]。

4）方案对比与实验：通过方案对比及实验结果，展示并分析了本方案能够实现包括语义搜索、结果排序、结果精确性验证、完整性验证、排序可验证等功能。

1 问题模型与预备知识

1.1 系统模型

如图1所示，系统模型包含3个实体：数据拥有者、授权用户和云服务器。数据拥有者先对文件集 $F=f_1,f_2,\cdots,f_m$ 提取关键词 $W=\{w_1,w_2,\cdots,w_n\}$，并构造出安全索引 I。然后，将文件集加密为密文集 $C=\{c_1,c_2,\cdots,c_m\}$。最后将密文集 C 及安全索引 I 上传至云服务器。

在搜索阶段，授权用户先对搜索的关键词 $\delta=\{q_1,q_2,\cdots,q_u\}$ 进行语义拓展，得到语义拓展集合 $Q=\{q_1,q_2,\cdots,q_u,\alpha_1,\alpha_2,\cdots,\alpha_\sigma\}$；然后对 Q 构造陷门 T_Q 并上传。接收到陷门后，云服务器通过索引 I 进行搜索，将结果排序后返回。最后，授权用户解密密文文档。

1.2 威胁模型

在本文构建的威胁模型中，云服务器被认为是“半诚实且好奇”[11]的，云服务器和攻击者被视为潜在的威胁对象。这与之前大部分可搜索加密方案的威胁模型（“诚实且好奇”）有很大不同。确切说，“半诚实”是指云服务器为节省计算开销和传输带宽，或由于软件、硬件运作故障等原因，可能不会执行完整的搜索协议或返回全部的搜索结果，甚至会恶意地删除和篡改文件，导致用户的搜索结果丢失或错误；

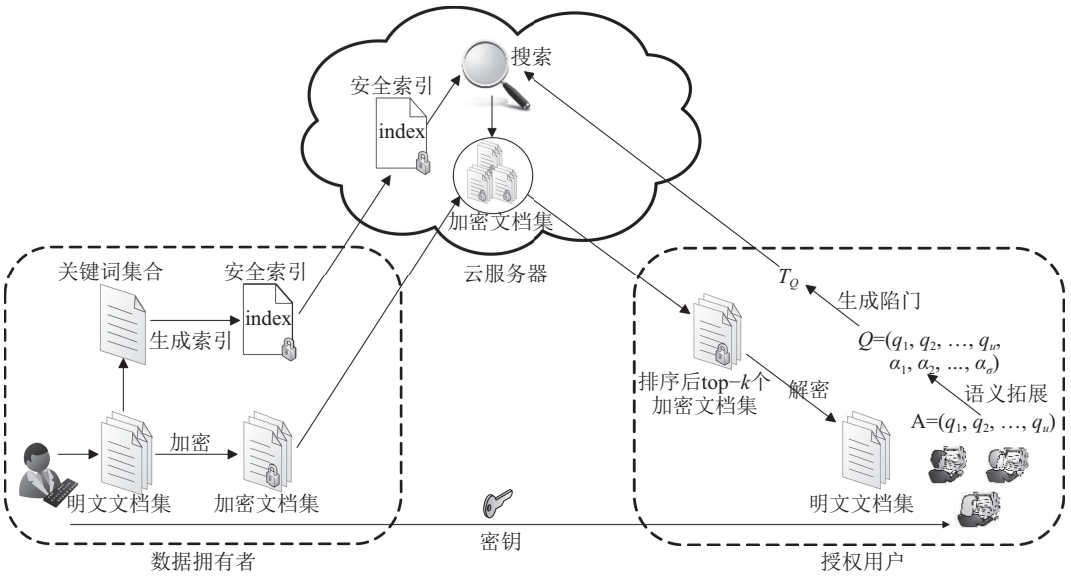

图 1　系统模型

Fig.1　System model

“好奇”是指云服务器出于某种目的，可能对存储在服务器上的文档或索引进行统计和分析以获取额外信息，这些信息可能会被攻击者利用并对系统构成安全隐患。因此在这种威胁模型下，所提方案一方面要能够对返回的搜索结果进行验证，另一方面要保障密文文档、索引和陷门等信息的隐私安全。

1.3　域加权评分

域（zone）的内容可以是任意的自由文本。例如，通常可以把文档标题、摘要或正文看作不同的域[16]。在不同域中的关键词通常具有不同的重要性，标题中的关键词重要性较大，摘要中的关键词重要性次之，正文中的关键词重要性最小。给定一系列文档，假定每篇文档有t个域，其对应权重系数分别为$k_1,k_2,\cdots,k_t\in[0,1]$，满足：

$$\sum_{i=1}^{t}k_i=1 \tag{1}$$

令s_i为查询和文档的第i个域的匹配得分（1和0分别表示匹配上和没匹配上），则域加权评分方法可以定义为：

$$Y=\sum_{i=1}^{t}k_i\,s_i \tag{2}$$

1.4　布隆过滤器

布隆过滤器（Bloom Filter）[17]是一种高效的数据结构，可以用于快速地判断某个元素和集合之间的隶属关系。通常可以将布隆过滤器初始化为每一位都为0，长度为m比特位的数组。布隆过滤器对应了r个独立同分布的哈希函数h_i：$\{0,1\}^*\to[1,m]$，$t\in[1,r]$，集合中的每个元素经过哈希函数计算后，映射到数组中的相应位置，被映射的比特位值设为1。当需要判断某个元素是否属于该集合时，同样将该元素通过哈希函数映射到数组，若对应的比特位的值全部为1，则该元素有较大概率存在于该集合中。

2　具体方案

2.1　关键词语义拓展方法

1）创建语义拓展树：给定关键词$\delta=\{q_1,q_2,\cdots,q_u\}$，利用WordNet进行语义拓展。在WordNet中，主要含有名词、动词、形容词、副词这4类实词，每个关键词可能包括这4种词性的一个或多个，每个词性下又有多种词义。因此，语义拓展要考虑各种词性和词义的情况。如：good在名称词性下的第1种意思为“好处”，因此可以拓展出单词advantage等。为语义拓展更为全面，还要考虑其第2种意思“美德”，并由此拓展出kindness等。good在形容词词性下则可以拓展出其他单词。另外，在语义研究中有一对概念是上义词（superordinate）和下义词（subordinate），上义词是对事物的概括性、抽象性说明，下义词是事物的具体表现形式，如color和white、black、blue等互为上下义关系。在语义拓展中考虑上下义关系可以使拓展结果更合理、全面。因此，可以构造出一棵语义拓展树，其结构见图2。

2）语义相似度计算：由于初步语义拓展的关键词可能较多，将其全部用于查询则查询范围太大，查询结果不够精确，因此计算原单词和拓展词之间的语义相似度Z，只选取最相关的前σ个拓展词，得到语义拓展集合$Q=\{q_1,q_2,\cdots,q_u,\alpha_1,\alpha_2,\cdots,\alpha_\sigma\}$。目前语义相似度分数的计算方法主要有两种，一种是基于语

义距离的方法，另一种是基于信息内容的方法。本文使用基于信息内容的Lin[18]方法计算2个关键词之间的相似度分数：

$$Z = sim(w_1, w_2) = \frac{2 \times IC(F(w_1) \cap F(w_2))}{IC(F(w_1)) + IC(F(w_2))} \tag{3}$$

式中：$F(w)$为关键词w包含的特征集合；$IC(S)$表示特征集合S中包含的信息内容，计算方法如下：

$$IC(S) = -\sum_{f \in S} \log P(f) \tag{4}$$

式中，$P(f)$为特征f的出现概率。通过计算含有特征f的关键词在语料库中的百分比，可得到$P(f)$。当2个关键词有完全相同的特征时，其相似度分数为最大值1；2个关键词无任何相同特征时，其相似度分数为0。具体的语义拓展流程参考算法一，其中$sim(\cdot,\cdot)$可换成其他语义相似度计算方法。

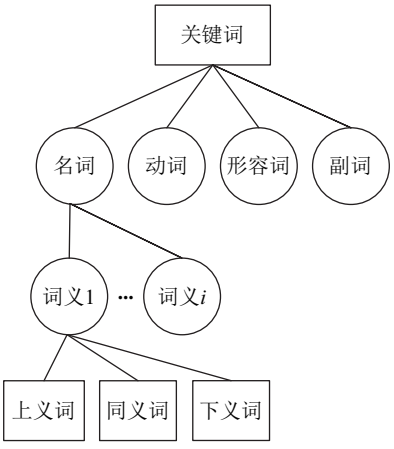

图 2　语义拓展树

Fig.2　Semantic expansion tree

算法一 语义拓展算法

输入：查询关键词$\delta = \{q_1, q_2, \cdots, q_u\}$

输出：拓展集合$Q = \{q_1, q_2, \cdots, q_u, \alpha_1, \alpha_2, \cdots, \alpha_\sigma\}$

1. for each $q_i \in \delta$ do
   for q_i的每一种词性do
   for q_i的每一种意思do
   if q_i有同义词||上位词||下位词
   $R=R\cup$同义词||上位词||下位词
   end if
   end for
   end for
   end for
2. for each $\alpha_{ij} \in R$ do
   $Z_{ij}=sim(q_i, \alpha_{ij})$
   end for
3. if $|R|>\sigma$
   根据Z_{ij}由大到小选σ个α_{ij}放入Q
   end if
4. $Q=Q\cup\delta$
5. return $Q = \{q_1, q_2, \cdots, q_u, \alpha_1, \alpha_2, \cdots, \alpha_\sigma\}$

2.2　方案结构

1）KeyGen（λ）：输入一个安全参数λ，生成文档加密密钥sk、单向哈希函数的密钥hk，保序加密函数的密钥ek。将密钥sk、hk发送给授权用户。

2）Index（F, hk, ek）：

a）抽取关键词：数据拥有者对文档集$F=\{f_1, f_2, \cdots, f_m\}$抽取关键词，得到关键词集合$W=\{w_1, w_2, \cdots, w_n\}$。

b）计算相关度分数：采用tf–idf权值计算方法[16]计算关键词相关度分数S。

$$S = \frac{1}{|f|} \times (1 + \ln tf_{t,f}) \times \ln\left(1 + \frac{N}{df_t}\right) \tag{5}$$

式中，$|f|$表示文档f的长度，$tf_{t,f}$表示关键词t在某篇文档f中出现的频率，N表示所有文档数量，df_t表示包含关键词t的文档数量。

c）计算域加权得分Y：在本方案中，假定每篇文档有3个域，标题为第1个域$zone_1$、摘要为第2个域$zone_2$，正文为第3个域$zone_3$。其对应的权重系数分别为k_1=0.5、k_2=0.3、k_3=0.2，满足公式（1）。令s_i为查询关键词和文档的第i个域的匹配得分，s_i=1表示匹配，s_i=0表示未匹配，可以根据式（2）计算出该关键词的域加权得分。例如，在某文档中，关键词computer没出现在标题中，但出现在了摘要和正文，则s_1=0、s_2=1、s_3=1；那么，计算出computer在该文档中域加权得分为Y=0.3×1+0.2×1=0.5。

d）创建倒排索引：为关键词$w_i \in W$创建索引$I_i = w_i \| \{id_j \| S_{i_j} \times Y_{i_j}\}$。其中，$id_j$表示包含$w_i$的文档$f_j$的标识符，$S_{ij}$表示$w_i$在$f_j$中的相关度分数，$Y_{ij}$表示$w_i$在$f_j$中的域加权得分。

e）加密倒排索引：为保护相关度分数S_{ij}与域加权得分Y_{ij}的隐私性，需要进行加密操作。采用密钥为ek的非线性保序加密[19]函数$OPE(\cdot)$，将$S_{ij}\times Y_{ij}$加密为$OPE(S_{ij}\times Y_{ij})$。即使$S_{ij}\times Y_{ij}$处于密文状态，云服务器仍然可以对其进行高效排序，减少了用户的计算开销和带宽资源浪费。另外，为保障关键词$w_i \in W$的隐私安全，需要使用带密钥hk的哈希函数将其加密为$h(w_i)$。表1表示加密的倒排索引，左侧为密文关键词$h(w_i)$，右侧为文档标识符和加密分数$\{id_j \| OPE(S_{i_j} \times Y_{i_j})\}$，其中$1 \le i \le n$，$1 \le j \le m$。因此，加密后的索引可以表示为$I_i = h(w_i) \| \{id_j \| OPE(S_{i_j} \times Y_{i_j})\}$。最后，数据拥有者将索引集合$I = \{I_1, I_2, \cdots, I_n\}$上传给云服务器。

表 1　加密的倒排索引

Tab.1　Encrypted inverted index

密文关键词	文档标识符与加密分数
$h(w_i)$	$\{id_j \| OPE(S_{i_j} \times Y_{i_j})\}, 1 \le j \le m$

3）Encrypt(F, sk)：数据拥有者使用密钥为sk的对称加密算法$\varphi(\cdot)$对文档集合$F=\{f_1,f_2,\cdots,f_m\}$进行加密，得到密文集合$C=\{c_1,c_2,\cdots,c_m\}$并上传给云服务器。

4）Trapdoor(δ, hk)：当授权用户搜索时，首先输入u个感兴趣的关键词$\delta=\{q_1,q_2,\cdots,q_u\}$。接着，基于WordNet创建语义拓展树，由于考虑了关键词q_i多词性多词义的情况，并且进行了上义词、同义词、下义词的语义拓展，因此拓展出的关键词较为全面。但此时拓展词数量可能较多，将其全部用于查询则查询范围太大、查询结果不精确。因此根据算法一，使用基于信息内容的Lin[18]的方法，计算原单词q_i和拓展词α_{ij}之间的语义相似度Z_{ij}，只选取最相关的前σ个拓展词，最后得到语义拓展集合$Q=\{q_1,q_2,\cdots,q_u,\alpha_1,\alpha_2,\cdots,\alpha_\sigma\}$。为方便下文描述，将$Q$中的元素$q_i$（$1\le i\le u$）或$\alpha_j$（$1\le j\le\sigma$）统一表示为$\beta_z$（$1\le z\le u+\sigma$），即$Q=\{\beta_1,\beta_2,\cdots,\beta_{u+\sigma}\}$。然后，使用密钥$hk$生成陷门$T_Q=\{h(\beta_1),h(\beta_2),\cdots,h(\beta_{u+\sigma})\}$。最后，将$T_Q$上传到云服务器。

5）Query(T_Q, k, I)：

a）遍历倒排索引：云服务器接收到陷门T_Q后，依次将$h(\beta_z)\in T_Q$与倒排索引中的$h(w_i)$进行匹配。若$h(\beta_z)=h(w_i)$，则找到对应的索引$I_i=h(w_i)\|\{id_j\|OPE(S_{i_j}\times Y_{i_j})\}$。

b）双因子排序：为进行精确排序，设计了双因子排序算法。首先，根据查询关键词$h(\beta_z)$对应的语义相似度Z_z由大到小的顺序，对$h(\beta_z)$匹配到的索引I_i进行排序；再根据每个索引I_i中的加密分数$OPE(S_{ij}\times Y_{ij})$对文档标识符id_j进行排序；最后，根据id_j返回top-k篇密文$C'=\{c_1,c_2,\cdots,c_k\}$。其流程可参考算法二。

6）Decrypt(C', sk)：授权用户使用数据拥有者分发的密钥sk，将top-k篇密文$C'=\{c_1,c_2,\cdots,c_k\}$解密，获得所需的明文。

2.3　验证方法

为应对“半诚实且好奇”的云服务器威胁，用户希望确保搜索结果的精确性（返回的文档确实存在于数据集中且符合用户的搜索请求）和完整性（没有遗漏符合搜索请求的文档），以及实现排序可验证功能。因此，数据拥有者首先构造出验证信息，连同索引上传云服务器。接着，用户提交陷门进行搜索，云服务器返回搜索结果及相应验证信息。最后，用户在本地对搜索结果进行验证。具体的验证方案如下：

1）验证信息的准备：数据拥有者在构建完索引I后，为能让用户快速检测搜索集合$Q=\{\beta_1,\beta_2,\cdots,\beta_{u+\sigma}\}$中的查询关键词$\beta_z$（$1\le z\le u+\sigma$）是否存在于关键词集合$W=(w_1,w_2,\cdots,w_n)$中，还要将所有密文关键词$h(w_i)$插入布隆过滤器$BL$中，并使用密钥为$mk$的$Mac(\cdot)$计算$mac_{BL}=Mac(BL)$。

算法二 双因子排序算法

输入：倒排索引$I=\{I_1,I_2,\cdots,I_n\}$，陷门T_Q

输出：top-k篇密文文档$C'=\{c_1,c_2,\cdots,c_k\}$

```
1.  for each h(β_i)匹配到的I_i ∈ I
        将I_i按照h(β_z)对应的语义相似度Z_z从大到小
        进行排序，得到I' = (I_1, I_2, ···, I_ψ)
    end for
2.  初始化变量num=0
3.  for each I_i ∈ I'
        将id_j按OPE(S_ij×Y_ij)从大到小排序得到
        ID_i = {id_1, id_2, ···, id_j}
        for each id_j ∈ ID_i
            if num<k且id_j ∉ ID'
                将id_j添加到集合ID'中
                num++
            else if num >k
                return ID'
            end if
        end for
    end for
4.  找到ID' = {id_1, id_2, ···, id_k}对应的密文文档C' =
    {c_1, c_2, ···, c_k}
5.  return C' = {c_1, c_2, ···, c_k}
```

另外，为实现对安全索引的验证，需对安全索引计算消息认证码$mac_{I_i}=Mac(I_i)$。最后，为实现对密文文档的验证，还需结合文档标识符集合$ID=(id_1,id_2,\cdots,id_m)$，对密文文档集$C=(c_1,c_2,\cdots,c_m)$中每篇密文文档计算消息认证码，生成$mac_{c_i}=Mac(id_i\|c_i)$。最后将$BL$、$mac_{BL}$及所有$mac_{I_i}$、$mac_{c_i}$，连同索引$I=(I_1,I_2,\cdots,I_n)$、密文集合$C=(c_1,c_2,\cdots,c_m)$一起上传云服务器。

2）搜索过程：用户的搜索方法和2.2节中Query的步骤基本相同，但有一些改动。注意到2.2节Query中根据文档得分大小排序后，云服务器只返回了top-k篇密文文档给用户，这与验证搜索结果完整性的目的相矛盾。因此，在启动本方案的可验证功能后，需要取消只返回top-k篇密文文档的设定。为配合之后的验证操作，此时云服务器应该返回的内容包括：

a）布隆过滤器$\widetilde{BL}$及mac_{BL}。

b）所有查询词$h(\beta_z)$匹配到的安全索引$\widetilde{I_i}=$

$h(w_i)\|\{id_j\|OPE(S_{i_j}\times Y_{i_j})\}$及$mac_{I_i}$。

c)在匹配到的安全索引$\widetilde{I_i}$中，经过双因子排序后的id_j所对应的所有密文文档集$C''=\{\tilde{c}_1,\tilde{c}_2,\cdots,\tilde{c}_x\}$及$mac_{c_i}$。

注意到，由于云服务器返回的布隆过滤器$\widetilde{BL}$、安全索引$\widetilde{I_i}$、密文集$C''=\{\tilde{c}_1,\tilde{c}_2,\cdots,\tilde{c}_x\}$都有待验证，因此在其指代符号之上添加波浪号，区别于数据拥有者上传到云服务器的相应原始数据。

3)验证过程：

a)验证索引中是否包含查询关键词：首先用户使用密钥mk对返回的$\widetilde{BL}$计算$mac_{\widetilde{BL}}=Mac(\widetilde{BL})$，若$mac_{\widetilde{BL}}=mac_{BL}$，则证明返回的$\widetilde{BL}$没有被篡改。然后，通过$\widetilde{BL}$可快速检测查询关键词$\beta_z\in Q(1\le z\le u+\sigma)$是否存在于关键词集合$W=\{w_1,w_2,\cdots,w_n\}$中。如果$h(\beta_z)$属于$\widetilde{BL}$，则$\beta_z\in W$。令所有属于$\widetilde{BL}$的$h(\beta_z)$构成的集合为$T_Q{}'$。该验证过程可以有效防止云服务器对用户的欺骗。例如，云服务器的原始安全索引I中含有用户查询的关键词，但云服务器却欺骗用户没有找到相关文档且不返回任何搜索结果，通过该验证即可很容易判断搜索结果是假的。

b)验证返回的索引是否完整且正确：首先，用户使用密钥mk对返回的安全索引$\widetilde{I_i}=h(w_i)\|\{id_j\|OPE(S_{i_j}\times Y_{i_j})\}$计算得到$mac_{\widetilde{I_i}}=Mac(\widetilde{I_i})$，若所有$mac_{\widetilde{I_i}}=mac_{I_i}$，则证明返回的$\widetilde{I_i}$没有被篡改。然后，用户使用$T_Q'$中的查询关键词$h(\beta_z)$在返回的安全索引$\widetilde{I_i}$中进行本地搜索。若每个查询关键词$h(\beta_z)$都能在返回的$\widetilde{I_i}$中找到对应相等的$h(w_i)$，则说明云服务器返回的安全索引正确。最后执行双因子排序算法，获得经过排序的文档标识符集$FID=\{id_1,id_2,\cdots,id_y\}$。

c)验证返回的密文是否完整且正确：至此，用户获得了经过本地排序后的文档标识符集$FID=\{id_1,id_2,\cdots,id_y\}$，也获得了经过云服务器排序后的密文文档集$C''=\{\tilde{c}_1,\tilde{c}_2,\cdots,\tilde{c}_x\}$，因此可以使用密钥$mk$依照顺序计算$mac_{\tilde{c}_i}=Mac(id_i\|\tilde{c}_i)$。若$x=y$且所有$mac_{\tilde{c}_i}=mac_{c_i}$，则证明返回的密文文档完整且正确，并且同时证明了经过云服务器排序后的密文文档的次序也是正确的，即实现了排序可验证。以上验证方案流程可以参考图3。

如果用户第1次进行查询并希望验证搜索结果，云服务器将返回所有用于验证的数据结构给用户。用户可以选择保存，避免之后搜索时的通信开销。如果用户使用了之前的关键词进行重复查询，云服务器就没有必要返回相应的验证结构，用户只需要将搜索结果和之前保存的验证结构比较，可以极大节省通信开销。

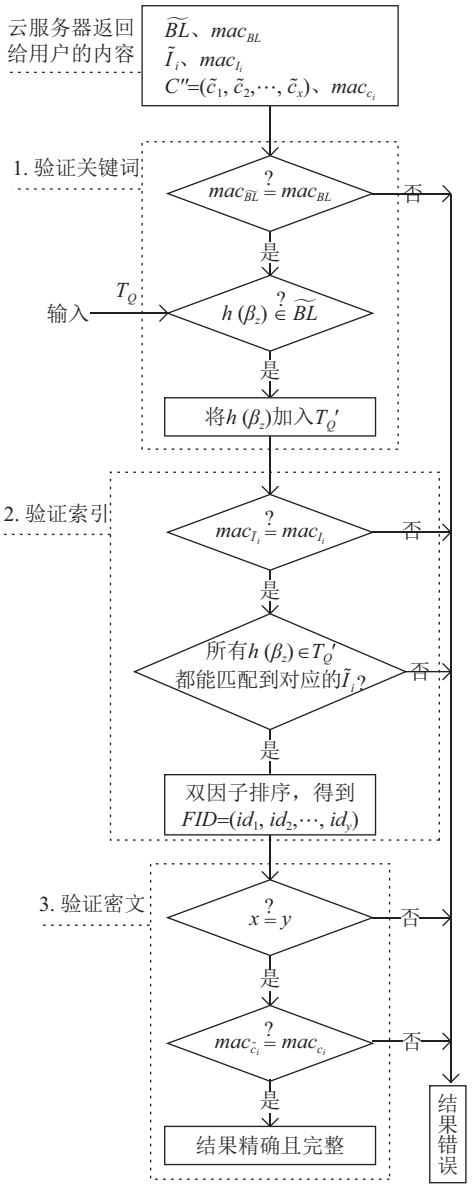

图 3 验证流程

Fig.3 Process of verification

第2.3节通过倒排索引、哈希函数、布隆过滤器和消息认证技术，创建相关验证信息。通过验证mac_{BL}、mac_{I_i}和mac_{c_i}等结构，用户能够确保返回文档的精确性和完整性。因此，提出的语义模糊搜索方案取得了完备的搜索结果可验证功能。此外，如果有必要对搜索结果的时效性进行验证，则在相应消息认证码中添加时间戳即可。

2.4 提升搜索效率

本方案基于倒排索引的搜索开销为$O(N)$，为进一步提高搜索效率，在不改变其他操作的前提下，只需将原有的倒排索引更改为符号索引树结构[12]，即可将搜索开销降低为$O(1)$。如图4所示，在构建索引树G_W时，首先产生根节点$\varnothing$，其为空集。然后计算

$h(w_i)$，若$h(w_i)$为τ个比特长，可以将$h(w_i)$分成τ/λ段，每段都用α_ρ进行表示，则$h(w_i)$可表示为$\alpha_1\alpha_2\cdots\alpha_{\tau/\lambda}$。接着，用$\alpha_\rho$代表一个节点，相同的$\alpha_\rho$为同一个节点。当$\alpha_\rho$为叶子节点时，在叶子节点中插入$I_i=h(w_i)\|\{id_j\|OPE(S_{ij}\times Y_{ij})\}$与其对应的消息认证码$mac_{I_i}$。每一条从根节点到叶子节点的路径表示$h(w_i)$。对每个关键词都进行上述操作，直至一棵完整的符号索引树构造完成。云服务器在Query阶段，能从匹配到的叶子节点中获取相应的索引I_i和mac_{I_i}。经过双因子排序后，即可将2.3(2)中提到的内容返回给搜索用户。

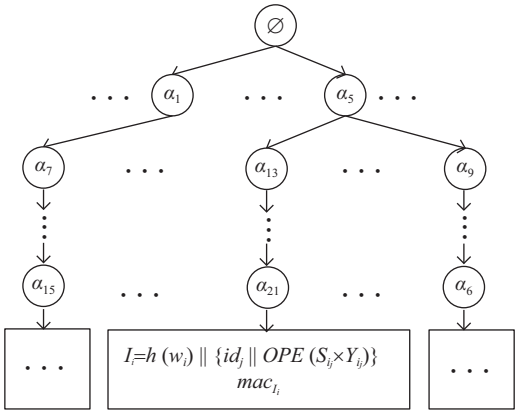

图 4　符号索引树

Fig.4　Symbol-tree

3　实验结果与分析

仿真实验使用Java语言编写，以IEEE等英文论文作为测试数据集。共提取关键词数量5 242个。环境为Windows 7, 64位台式机，处理器主频为2.80 GHz，内存4 GB。基于WordNet 3.0，采用JAR包JWS(Java WordNet Similarity)进行语义拓展，使用WordNet-InfoContent-2.1[20]，配合WordNet语料库计算语义相似度分数。

3.1　方案比较

表2为相关方案的对比情况。其中，M为文档数量，N为关键词数量，M'为最大模糊集数量，L为查询关键词的长度。

1)实现的功能比较：如表2所示，现有可验证方案大多仅支持单关键词搜索。Sun等[13]虽然实现了多关键词搜索，但产生的存储开销远大于本方案。Chai[11]、Wang[12]、Sun[13]等实现了验证功能，却不支持语义搜索；Xia等[10]能够进行关键词语义排序搜索，但不支持结果验证；林柏钢等[14]提出可验证的语义搜索方案，但无法对结果进行排序；Fu等[15]的语义搜索方案支持排序功能，但无法实现排序可验证。本方案基于WordNet实现了关键词语义搜索，引入域加权得分等对结果排序，借助布隆过滤器和消息认证等技术设计新的验证方案，能够验证结果完整性、精确性，并实现了排序可验证功能。

2)存储开销、搜索开销的比较：各个方案的存储开销、搜索开销主要受到索引结构的影响。由于Chai[11]、Wang[12]、Sun[13]等不支持语义搜索，其搜索功能与本方案有较大差异，因此不对其开销做进一步分析。Xia等[10]采用倒排索引结构，存储开销和搜索开销都为$O(N)$；林柏钢等[14]针对每篇文档构建布隆过滤器，存储开销和搜索开销都为$O(M)$；Fu等[15]与本方案都采用了符号索引树结构，由于不需要像Wang等[12]一样构造模糊集合，因此Fu等[15]同本方案的存储开销都为$O(N)$，搜索开销都为$O(1)$。由于在海量数据存储中，$M\gg N$，因此本方案在相关方案的比较中，其存储开销、搜索开销方面都比较有优势。

相比不含可验证功能的方案，由于可验证方案使得系统具备了更高级别的安全性能，能够应对“半诚实且好奇”云服务器的威胁，因此需要额外空间来存储验证信息。如果对系统的安全级别有较高的要求，这些额外的存储开销是值得且难以避免的。本方案运用布隆过滤器和消息认证码，在尽量节约存储开销的基础上，实现了对结果精确性、完整性的验证，特别是补充了排序可验证功能，使方案的安全体系更加完善。

表 2　方案对比

Tab.2　Comparison of schemes

内容	Chai[11]	Wang[12]	Sun[13]	Xia[10]	Lin[14]	Fu[15]	本方案
存储开销	$O(N)$	$O(M'N)$	$O(MN)$	$O(N)$	$O(M)$	$O(N)$	$O(N)$
搜索开销	$O(L)$	$O(1)$	$O(1)$	$O(N)$	$O(M)$	$O(1)$	$O(1)$
多关键词搜索	否	否	是	否	否	否	否
语义搜索	否	否	否	是	是	是	是
排序	否	否	是	是	否	是	是
验证完整性	是	是	是	否	是	是	是
验证精确性	是	是	是	否	是	是	是
排序可验证	否	否	否	否	否	否	是

3.2 索引和验证信息

图5为创建索引时间随关键词数量的变换趋势。关键词越多，创建索引耗时越长。图中显示创建符号索引树比创建倒排索引耗时多。注意到索引只需数据拥有者创建1次，而查询操作的执行更为频繁；由于符号索引树能极大提高查询效率，因此其更有利于提高系统的整体效率。

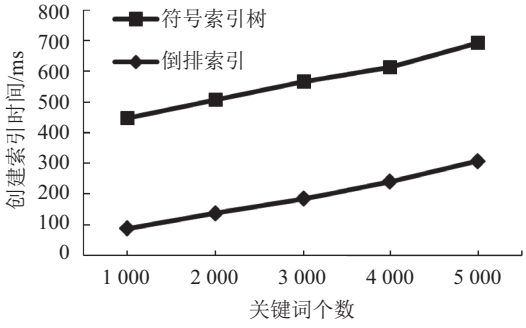

图 5 创建索引时间

Fig.5 Time of building index

图6为创建验证信息的时间随关键词数量的变换趋势。其中，验证信息是指在验证阶段需要验证的相关数据，如BL、mac_{BL}及所有mac_{I_i}、mac_{c_i}等。显然，关键词越多，索引I_i、密文c_i越多，因此创建验证信息的时间越长。验证信息与索引结构一样，只需创建1次。

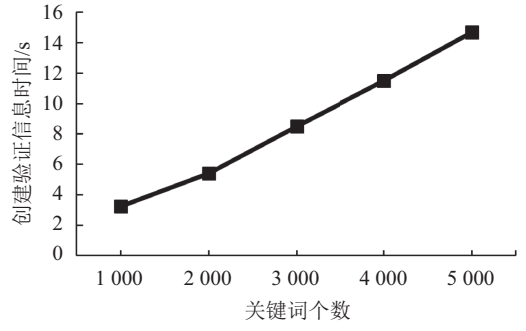

图 6 创建验证信息时间

Fig.6 Time of building verification information

3.3 语义拓展与加密查询

图7为语义拓展时间随查询关键词数量的变换趋势。查询词越多，就要对越多词汇进行语义拓展操作，拓展出的词汇随之增加，需要计算的语义相似度也越多，因此语义拓展的时间就越多。图8为加密查询的时间随查询关键词数量的变化趋势，查询词越多，语义拓展集合就越大，因此对查询进行加密的时间越多。

3.4 查 询

图9为查询时间随关键词数量的增长趋势。随着关键词数量的增加，基于倒排索引的查询时间增加较明显，基于符号索引树的查询时间增加不明显。基于符号索引树的查询时间明显小于倒排索引，这是由于符号索引树将含有相同前缀的哈希段合并为一个节点，能够更高效地淘汰无关信息，极大地提高查询效率。

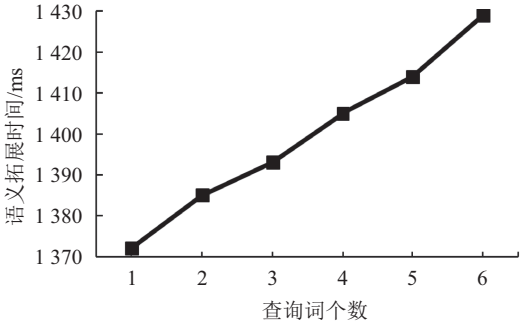

图 7 语义拓展时间

Fig.7 Time of semantic expansion

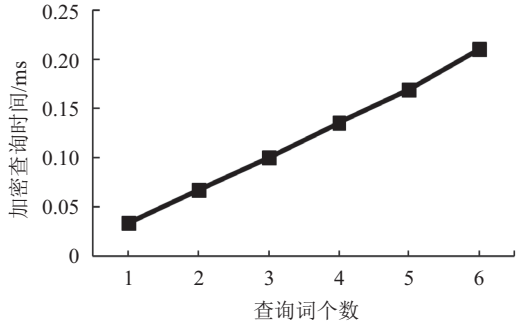

图 8 加密查询时间

Fig.8 Time of encrypting query

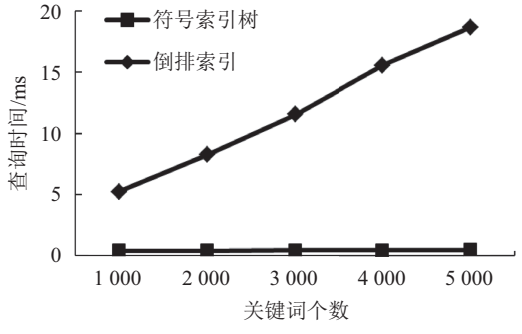

图 9 查询时间

Fig.9 Time of searching

3.5 验 证

图10为查询词数量分别为q=1、q=2、q=3时，验证时间随关键词数量的增长趋势。随着关键词数量的增加，云服务器返回的mac_{I_i}、mac_{c_i}的数量也越多，即用户需要验证更多的索引和密文的精确性和完整性，因此耗时越多。此外，随着查询词数量q的增加，会匹配到更多索引和密文，返回的mac_{I_i}、mac_{c_i}的数量越多，所以耗时也会相应增加。

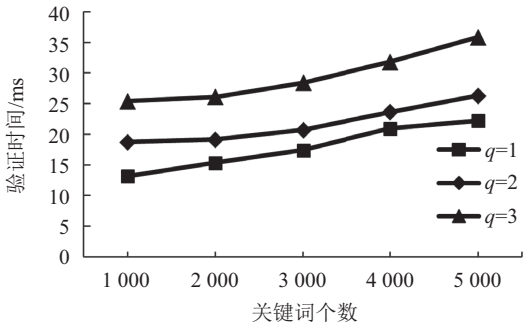

图 10　验证时间

Fig.10　Time of verifying

4　安全分析

在本方案中，云服务器能够获得密文文档C、安全索引I、查询陷门T_Q、相关验证信息BL、mac_{BL}、mac_{I_i}、mac_{c_i}，本方案应保障这些信息不会泄漏任何明文关键词或明文文档的相关信息。

1）文档：数据拥有者外包文件之前，使用传统对称加密方法（AES或3DES）$\varphi(\cdot)$对明文文档进行加密，保障文档的隐私性，且会通过安全信道将密钥发送给授权用户。在没有密钥的情况下，云服务器或攻击者难以对密文文档进行解密。

2）索引：在每条索引$I_i = h(w_i)\|\{id_j\|OPE(S_{i_j} \times Y_{i_j})\}$中，带密钥的哈希函数$h(\cdot)$保障了关键词$w_i \in W$的隐私安全；保序加密函数$OPE(\cdot)$保障了相关度分数$S$和域加权得分$Y$的隐私安全；文档标识符$id_j \in ID$只是关联密文文档的指示符号，并不会泄露相关信息。因此，索引的隐私安全能够得到保障。

3）查询：在陷门T_Q中，每个查询关键词$\beta_z \in Q$都由带密钥的哈希函数$h(\cdot)$加密，因此其隐私安全得到保障。

4）相关验证信息：本方案在布隆过滤器BL中插入加密的关键词$h(w_i)$，云服务器或攻击者在没有密钥hk的情况下无法得知$h(w_i)$的相关信息，也无法对BL进行分析，因此BL是隐私安全的。消息认证码mac_{BL}、mac_{I_i}、mac_{c_i}都是通过函数$Mac(\cdot)$得到的，只有拥有密钥mk的用户能使用其进行结果验证。

5　结　论

本文基于WordNet构建语义拓展树，实现了语义相似度计算与语义拓展；引入域加权评分反映不同域中关键词的权重差异，并设计双因子排序算法实现排序功能；运用布隆过滤器和消息认证技术，对搜索结果进行了有效的验证；最后通过索引树结构进一步提升搜索效率。对比分析和实验结果表明，本方案不但能够进行语义模糊搜索并对结果进行排序，还可以验证搜索结果的精确性和完整性，特别是实现了排序可验证功能。在下一步工作中，可以尝试使用其他索引结构实现排序可验证的可搜索加密方案，减少创建验证信息时产生的存储开销并支持多关键词搜索。

参考文献：

[1] Song D X,Wagner D,Perrig A.Practical techniques for searches on encrypted data[C]//Proceedings of the IEEE Symposium on Security and Privacy.Los Alamitos:IEEE Computer Society,2000:44–55.

[2] Chang Y C,Mitzenmacher M.Privacy preserving keyword searches on remote encrypted data[C]//Proceedings of the Applied Cryptography and Network Security.Berlin:Springer,2005:442–455.

[3] Curtmola R,Garay J,Kamara S,et al.Searchable symmetric encryption:Improved definitions and efficient constructions[J].Journal of Computer Security,2011,19(5):895–934.

[4] Wang C,Cao N,Ren K,et al.Enabling secure and efficient ranked keyword search over outsourced cloud data[J].IEEE Transactions on Parallel and Distributed Systems,2012,23(8):1467–1479.

[5] Wang C,Cao N,Li J,et al.Secure ranked keyword search over encrypted cloud data[C]//Proceedings of the IEEE International Conference on Distributed Computing Systems (ICDCS).Los Alamitos:IEEE Computer Society,2010:253–262.

[6] Cao N,Wang C,Li M,et al.Privacy-preserving multi-keyword ranked search over encrypted cloud data[J].IEEE Transactions on Parallel and Distributed Systems,2014,25(1):829–837.

[7] Boneh D,Crescenzo G D,et al,Public key encryption with keyword search[C]//Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques.Berlin:Springer,2004:506–522.

[8] Li J,Wang Q,Wang C,et al.Fuzzy keyword search over encrypted data in cloud computing[C]//Proceedings of the IEEE INFOCOM.Piscataway,New Jersey:IEEE,2010:1–5.

[9] Fu Z,Sun X,Linge N,et al.Achieving effective cloud search services:Multi-keyword ranked search over encrypted cloud data supporting synonym query[J].IEEE Transactions on Consumer Electronics,2014,60(1):164–172.

[10] Xia Z,Zhu Y,Sun X,et al.Secure semantic expansion based search over encrypted cloud data supporting similarity rank-

ing[J].Journal of Cloud Computing,2014,3(1):1–11.

[11] Chai Q,Gong G.Verifiable symmetric searchable encryption for semi honest but curious cloud servers[C]//Proceedings of the 2012 IEEE International Conference on Communications.Piscataway,NJ:IEEE,2012:917–922.

[12] Wang J,Ma H,Tang Q,et al.A new efficient verifiable fuzzy keyword search scheme[J].Journal of Wireless Mobile Networks,Ubiquitous Computing and Dependable Applications,2012,3(4):61–71.

[13] Sun W,Wang B,Cao N,et al.Verifiable privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking[J].IEEE Transactions on Parallel & Distributed Systems,2014,25(11):71–82.

[14] Lin Bogang,Wu Yang,Yang Yang,et al.Verifiable semantic fuzzy searchable encryption scheme in cloud computing[J].Journal of Sichuan University(Engineering Science Edition),2014,46(6):1–6.[林柏钢,吴阳,杨旸,等.云计算中可验证的语义模糊可搜索加密方案[J].四川大学学报(工程科学版),2014,46(6):1–6.]

[15] Fu Z,Shu J,Sun X,et al.Smart cloud search services:Verifiable keyword-based semantic search over encrypted cloud data[J].IEEE Transactions on Consumer Electronics,2014,60(4):762–770.

[16] Manning C D,Raghavan P,Schütze H.Introduction to information retrieval[M].Cambridge:Cambridge University Press,2008:76–78.

[17] Bloom B H.Space/time trade-offs in hash coding with allowable errors[J].Communications of the ACM,1970,13(7):422–426.

[18] Lin D.An information-theoretic definition of similarity[C]//Proceedings of the 15th International Conference on Machine Learning.San Francisco:Morgan Kaufmann,1998:296–304.

[19] Agrawal R,Kiernan J,Srikant R,et al.Order-preserving encryption for numeric data[C]//Proceedings of the 2004 ACM SIGMOD International Conference on Management of data.ACM,2004:563–574.

[20] Ted Pedersen.Information content computed on various corpora [EB/OL].(2013-5-19) [2016-08-15].http://www.d.umn.edu/~tpederse/similarity.html.

（编辑　李轶楠）

引用格式:Yang Yang,Yang Shulue,Cai Shengwei,et al.Semantic searchable encryption scheme supporting ranking verification[J].Advanced Engineering Sciences,2017,49(4):119–128.[杨旸, 杨书略, 蔡圣暐, 等. 排序可验证的语义模糊可搜索加密方案[J].工程科学与技术, 2017,49(4):119–128.]