

云计算下支持语义的可搜索加密方法研究

杨清琳¹, 黄治国², 钱文标¹, 杨晓雷¹

(1. 广西财经学院 现代教育技术部, 广西 南宁 530003;

2. 河南工程学院 计算机学院, 河南 郑州 451191)

摘要:传统的云计算下的可搜索加密算法没有对查询关键词进行语义扩展,导致了用户查询意图与返回结果存在语义偏差,并且对检索结果的相关度排序不够合理,无法满足用户对智能搜索的需求。对此,提出了一种支持语义的可搜索加密方法。该方法利用本体知识库实现了用户查询的语义拓展,并通过语义相似度来控制扩展词的个数,防止因拓展词过多影响检索的精确度。同时,该方法利用文档向量、查询向量分块技术构造出对应的标记向量,以过滤无关文档,并在查询-文档的相似度得分中引入了语义相似度、关键词位置加权评分及关键词-文档相关度等影响因子,实现了检索结果的有效排序。实验结果表明,该方法在提高检索效率的基础上显著改善了检索结果的排序效果,提高了用户满意度。

关键词:云计算;可搜索加密;语义扩展;本体知识库;语义相似度;标记向量

中图分类号:TP391

文献标识码:A

文章编号:1673-629X(2020)03-0111-06

doi:10.3969/j.issn.1673-629X.2020.03.021

Research on Searchable Encryption Method with Supporting Semantics in Cloud Computing

YANG Qing-lin¹, HUANG Zhi-guo², QIAN Wen-biao¹, YANG Xiao-lei¹

(1. Department of Modern Education Technology, Guangxi University of Finance and Economics,

Nanning 530003, China;

2. School of Computer Science, Henan University of Engineering, Zhengzhou 451191, China)

Abstract: The traditional searchable encryption (SE) algorithm in cloud computing doesn't extend the query keywords semantically, which leads to the semantic deviation between the query intention and the returned result, and the inordinate relevance of the search results is not reasonable enough to meet the needs of users for intelligent search. In this study, a SE method of supporting semantics is proposed. The ontology knowledge base is used to realize the semantic extension of user query, and the number of extended words by semantic similarity is seriously controlled to improve the retrieval accuracy without overextending words. Furthermore, by using document vector and query vector block technology, the corresponding marker vector is constructed to filter irrelevant documents, and semantic similarity, keyword location weighted score and keyword-document relevancy are introduced into query-document similarity score, so as to effectively sort the retrieval results. The experiment shows that the proposed method can significantly increase the retrieval efficiency and improve the user's satisfaction.

Key words: cloud computing; searchable encryption; semantic expansion; ontology knowledge base; semantic similarity mark vector

0 引言

随着云计算技术的日趋成熟和快速发展,把数据存储在云端,只需较小的代价就能享受高效快捷的文件存储和处理服务已经成为当下数据存储的一种重要选择。然而,数据安全和隐私保护如何有效地得到保障仍然是很多用户所担心的问题。常见的解决方法是将用户数据先进行加密,再将加密后的密文数据上传

到云端,当用户需要使用某个数据时,再从用户密文中检索出需要的文档。如何对加密数据执行检索等操作,是近年来可搜索加密(searchable encryption, SE)技术^[1-2]研究的内容。

针对国内外学者可搜索加密技术的相关研究,大致可以分为单关键词可搜索加密、多关键词可搜索加密和模糊加密搜索。单关键词可搜索加密^[3-5]是基于

收稿日期:2019-03-12

修回日期:2019-07-15

网络出版时间:2019-11-07

基金项目:河南省高等学校重点科研资助项目(17A520027);广西自然科学基金(2014GXNSFAA118259)

作者简介:杨清琳(1983-),女,硕士,工程师,研究方向为智能信息处理、信息安全。

网络出版地址: <http://kns.cnki.net/kcms/detail/61.1450.TP.20191107.0908.004.html>

给公有云,由公有云执行进一步的检索操作。公有云服务器的主要责任是存储密文文档及安全索引,并根据授权用户的查询请求执行检索服务。公有云通过私有云提交的候选索引标识符找出与之相应的安全索引,并用找出的安全索引与授权用户提交的陷门来计算查询-文档相关度得分,返回得分最高的前 Top-k 篇文档给授权用户。

1.2 系统威胁分析

从图1的系统模型中可以看出,用户数据通过网络上传到云服务器中储存,存在网络攻击者截取窃听用户数据的可能,但由于用户数据上传前是经过加密的,即使攻击者盗取用户数据,如果没有获取解密密钥,也不能破解加密文件,因此该类攻击对系统的威胁较弱。

另外公有云服务器具有“诚实而好奇”的特征^[13-14],它会正确地执行授权用户提交的陷门完成查询请求,并会根据私有云上传的候选索引标识符找出与之相对的安全索引,完成检索服务后会如实返回检索结果,但是无法保证公有云不会采取一些措施去分析用户数据、索引及查询请求以获取额外的用户隐私信息。

文中探讨的为已知密文模型,假设公有云在知道数据用户上传的加密文档、加密索引和授权用户发送的查询陷门的前提下,无法利用查询陷门构造出一个新的查询陷门,并且私有云不会对外泄露标记向量信息,会如实地完成授权用户提交的查询请求并提交给公有云。并假定数据用户和授权用户是完全可信的,对存储在公有云上的加密文档有权限共享,不会通过其已了解的部分数据、索引、陷门、密钥等信息来试图攻击其他用户隐私信息,且密钥已经安全分发,即数据用户与授权用户之间的密钥授权已经完成。

2 云计算下支持语义的可搜索加密方法

2.1 关键词-位置加权评分

关键词在文档中出现的位置,可以反映出该关键词在这篇文档中的重要程度。例如,关键词出现在标题中,则比其出现在正文中对文档的区分度更高。如不做关键词位置区分,即关键词出现在标题和正文赋予同样的权重,则构造出的索引无法准确地反映关键词对文档的重要程度,导致检索结果不精确。文中设定每篇文档有 s 个不同的位置,每个不同的位置被赋予不同的权重系数 $w_1, \dots, w_i, \dots, w_s (0 \leq w_i \leq 1)$ 并且满足 $\sum_{i=1}^s w_i = 1$,如果关键词 t 出现在文档的第 i 个位置,则设置 m_i 为 1,否则设置为 0,关键词-位置加权评分计算公式定义为:

$$L = \sum_{i=1}^s w_i m_i \quad (1)$$

2.2 基于领域本体的语义相似度计算

本体是以树状层次结构对概念进行描述和组织,从而构建出概念的语义空间,并使得概念树中任意两个概念节点之间的语义距离具有可计算性。

定义1:文中使用两个节点间的语义距离来计算其语义相似度,对于已知的有向边 $A \rightarrow B$,节点 A 的深度 $\text{Deep}(A)$ 代表有向边 $A \rightarrow B$ 的深度,则该 $A \rightarrow B$ 的语义距离定义为:

$$\text{Dsit}(A, B) = \frac{1}{\text{Weight}(A, B)} \quad (2)$$

$$\text{其中, } \text{Weight}(A, B) = \frac{1}{2^{\text{Deep}(A)}} + \frac{1}{2^{\text{Deep}(A)-1}} + \dots + \frac{1}{2}$$

为该有向边的权重。

定义2:两个概念节点 V_i, V_j 之间的语义距离定义为:

$$\text{Dist}(V_i, V_j) = \text{Dist}(V_i, \text{Can}(V_i, V_j)) + \text{Dist}(V_j, \text{Can}(V_i, V_j)) \quad (3)$$

其中, $\text{Dist}(V_i, \text{Can}(V_i, V_j)) = \sum_{n \in \text{path}(V_i, \text{Can}(V_i, V_j))} \text{Dist}(n, \text{parent}(n))$, $\text{Can}(V_i, V_j)$ 代表距离两个概念节点 V_i, V_j 最近的公共祖先节点, $\text{path}(V_i, \text{Can}(V_i, V_j))$ 代表在本体层次结构中概念节点 V_i 到最近公共祖先节点 $\text{Can}(V_i, V_j)$ 的最短路径上所有有向边的集合。根据语义距离计算语义相似度公式为:

$$\text{Sim}(V_i, V_j) = 1 - \frac{\text{Dist}(V_i, V_j)}{2(\text{MaxDeep} - 1)} \quad (4)$$

其中, MaxDeep 为本体层次结构的最大深度。

2.3 计算关键词-文档相关度得分

文中使用关键词词频 * 逆文档率来计算关键词与文档的相关度,计算公式为:

$$\text{tf-idf} = \text{tf}(f, t) * \text{idf}(t) \quad (5)$$

其中, $\text{tf}(f, t)$ 为关键词 t 在文档 f 中出现的频率, idf 的计算公式为:

$$\text{idf}(t) = 1 + \log \frac{N}{n + 1} \quad (6)$$

其中, N 表示整个文档集 D 中包含的所有文档个数, n 表示整个文档集 D 中包含有关键词 t 的文档个数。

不难发现上述公式存在如下问题:一般情况下,文档集中会包含长短不一的各类文档,而长文档(即大文档)所包含的词条个数通常会大于短文档(小文档)中所包含的词条个数,如果同一个关键词在长文档(10 000个分词)和短文档(100个分词)中的词频都为 10 次,显然上述公式的计算方法不利于短文档中关键词对文档的相关度得分计算。为公平起见,将对词频

做归一化处理,以保证词频计算权重公式更加合理。

$$gf(f, t) = \frac{1 + \log \text{tf}(f, t)}{1 + \log(\sum_{i=1}^N \text{tf}(f_i, t) / N)} \quad (7)$$

其中, $\sum_{i=1}^N \text{tf}(f_i, t)$ 为关键词 t 在整个文档集中出现的总词频数之和。则关键词-文档相关度计算公式为:

$$gf_idf = gf(f, t) * idf(t) \quad (8)$$

2.4 算法描述

文中基于向量空间模型实现文档检索,每个文档构造一个文档向量,其维度为文档集抽取的关键词集合的个数,从文档中抽取的每个关键词对应向量的每一维,每一维的值设置为相应关键词的权重。用户的查询也被构造成一个查询向量,维度与文档向量一致。用查询向量与文档向量进行内积来计算查询-文档的相关度得分并用该相关度分数对文档进行排序。文中支持语义的可搜索加密算法描述如下:

Step1:数据用户从整个文档集 $D = (d_1, d_2, \dots, d_i, \dots, d_m)$ 中使用分词工具提取出关键词集合 $T = (t_1, t_2, \dots, t_j, \dots, t_n)$ 。

Step2:KMS生成对称密钥 K ,生成的密钥用三元组形式表示: $\{S, M_1, M_2\}$,其中 S 为 n 维随机比特指示向量, M_1, M_2 为 n 维可逆矩阵,并将密钥 K 安全分发到数据用户和授权用户手中。

Step3:数据用户用式(1)计算关键词的位置加权评分 L ,用式(8)计算出关键词在文档中的相关度得分 gf_idf 。

Step4:数据用户为每个文档 d_i 构造出相应的文档向量 DV_i ,若 $t_j \in d_i$,则 $DV_i[j] = 1$,否则为0。

Step5:数据用户将文档向量 DV_i 平均分成 u 块, u 满足条件 $u \mid n$,生成文档标记向量 $DMV_i = (DMV_{i1}, DMV_{i2}, \dots, DMV_{ij}, \dots, DMV_{iu})$,方法是如果某块全为0,则 $DMV_{ij} = 0$,否则为1。并生成对应索引标识符 $IM_i = (DMV_i, id_i)$,后将文档标记向量 DMV_i 及其对应的索引标识符 IM_i 上传到私有云。

Step6:数据用户将文档向量中 $DV_i[j] = 1$ 的值对应替换成 $L * (gf_idf)$,并用 n 维随机比特指示向量 S 将文档向量 DMV_i 分解成 DV_i^1 和 DV_i^2 ,方法是当 $S_j = 0$ 时, $DV_i[j] = DV_i[j]^1 = DV_i[j]^2$;当 $S_j = 1$ 时, $DV_i[j] = DV_i[j]^1 + DV_i[j]^2$,其中 $DV_i[j]^1$ 和 $DV_i[j]^2$ 为随机数。用 M_1 和 M_2 加密 DV_i^1 和 DV_i^2 后得到 $SV_i = \{M_1^T DV_i^1, M_2^T DV_i^2\}$,并生成对应的安全索引 $I_i = (SV_i, id_i)$,后将安全索引 I_i 提交至公有云。

Step7:数据用户使用加密密钥 K 对文档集合进行 $D = (d_1, d_2, \dots, d_i, \dots, d_m)$ 加密,生成密文文档集合

$C = (c_1, c_2, \dots, c_i, \dots, c_m)$,然后提交至公有云上存储。

Step8:授权用户输入初始查询关键字 $Q = (q_1, q_2, \dots, q_i, \dots, q_n)$,再使用本体知识库进行语义扩展,并使用语义相似度公式(4)计算初始关键字与扩展词的语义相似度,选择相似度排前的 s 个扩展词(c_1, c_2, \dots, c_s)加入到初始查询来构成最终的查询关键字集合 $Q' = (q_1, q_2, \dots, q_i, \dots, q_n, c_1, c_2, \dots, c_s)$,其对应的语义相似度为 $SC = (sc_1, sc_2, \dots, sc_n, sc_{n+1}, \dots, sc_{n+s})$ 。

Step9:根据语义扩展后的查询关键字集合 Q' 构造查询向量 QV ,方法是若 $t_i \in Q'$,则 $QV[i] = 1$,否则为0。将查询向量 QV 平均分成 u 块, u 满足条件 $u \mid n$,生成查询标记向量 $QMV = (QMV_1, QMV_2, \dots, QMV_i, \dots, QMV_u)$,方法是如果某块全为0,则 $QMV_i = 0$,否则为1。将查询标记向量上传至私有云。

Step10:将 $QV[i] = 1$ 的值替换成 sc_i ,并用指示向量 S 将查询向量 QV 分解成 QV^1 和 QV^2 。方法是当 $S_j = 0$ 时, $QV[i] = QV[i]^1 = QV[i]^2$;当 $S_j = 1$ 时, $QV[i] = QV[i]^1 + QV[i]^2$,其中 $QV[i]^1$ 和 $QV[i]^2$ 为随机数。用 M_1 和 M_2 加密 QV^1 和 QV^2 后得到陷门 $T = \{M_1^{-1} QV^1, M_2^{-1} QV^2\}$,并将陷门 T 提交至公有云。

Step11:私有云对查询标记向量 QMV 中的每一位1的值与文档标记向量 DMV_i 进行匹配,相同位置上都为1,则说明该文档对应的块包含有查询关键词,则将文档标记向量 DMV_i 对应的索引标识符上的 id_i 记录下来,最终得到所有可能包含查询关键词的候选索引标识集合 $ID' = \{\dots, id_i, \dots, id_j\}$,并将其提交至公有云。公有云根据 ID' 找到对应的安全索引 I_i ,将对应的 SV_i 与陷门 T 进行内积来计算查询与文档的相似度得分。

$$\begin{aligned} SV_i \cdot T &= \{M_1^T DV_i^1, M_2^T DV_i^2\} \cdot \{M_1^{-1} QV^1, M_2^{-1} QV^2\} = \\ &DV_i^1 \cdot QV^1 + DV_i^2 \cdot QV^2 = DV_i \cdot QV = \\ &\sum_{j=1}^n L_j \cdot gf_idf \cdot sc_j \end{aligned}$$

Step12:对查询-文档的相似度得分进行排序,将前 Top_k 篇文档返回给授权用户。

Step13:授权用户使用密钥 K 对返回的文档进行解密,得到原始的明文文档。

3 实验结果分析

文中实验环境为:Windows 7, CPU: i7 四核 3.4 GHz, 内存:8 GB, 硬盘空间:1 TB, 仿真实验编程语言:Java。建立了一个计算机领域本体知识库,并从Web上抓取计算机领域网页3 000个作为测试数据集,使用ICTCLAS分词系统抽取出文档关键词。实验的目的是验证在用户初始查询中加入了本体语义推理的扩

展词及在查询-文档的相关度分数中引入了语义相似度、关键词不同位置加权评分及关键词-文档相关度得分影响因子后,密文检索效率(用 F_{measure} 来衡量)是否被提高。

$$F_{\text{measure}} = \frac{(\alpha^2 + 1)\text{precision} * \text{recall}}{\alpha^2(\text{precision} + \text{recall})} \quad (9)$$

当参数 α 取值为 1 时,就是常用的 F_1 调和平均考虑了 precision 和 recall 的结果, F_1 值较高则证明文中

方法的有效性。

图2给出了文中方法与 MRSE 方法创建索引所花费时间的变化趋势,可以看出随着文档关键词数量的增加,两种方法创建索引的时间都与之成正比,但由于文中方法需要对文档向量进行分块标记,因此创建索引的时间略有增加。但是对数据用户来说创建索引是外源到云端前的操作,因此对时间上的要求不是很高。

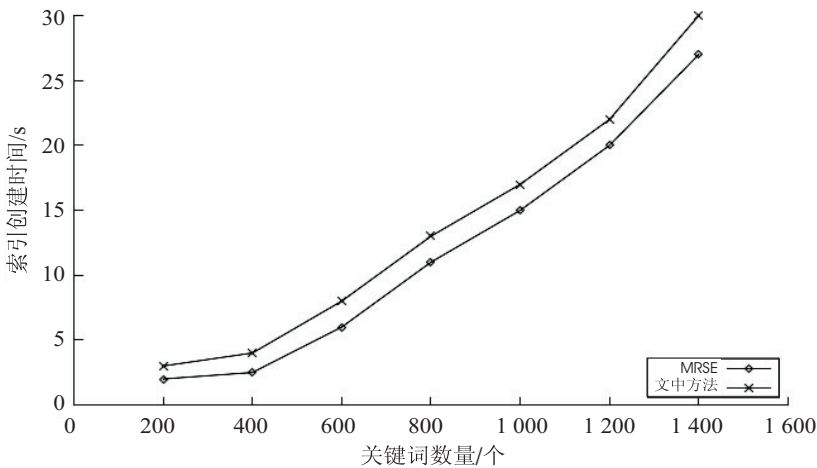


图2 索引创建时间对比分析

由于文中方法基于文档向量和查询向量的分块技术,因此所分块的数量 u 的取值直接决定了标记向量的维度, u 的取值与检索时间成反比。因为 u 值越大,那么文档向量所分的块数越多,每块的维度“ $u \times n$ ”越小,块全为 0 的概率越大,私有云对查询标记向量与文档标记向量匹配时,能过滤掉更多的无关文档,从而会

节省花费在计算不相关文档相关度分数和文档排序上的时间。但是查询标记向量与文档标记向量是明文保存在私有云上的, u 值越大,每块的维度“ $u \times n$ ”越小,越会暴露查询向量与文档向量的信息,因此,需要在考虑检索时间和保护隐私两者间权衡 u 的取值。

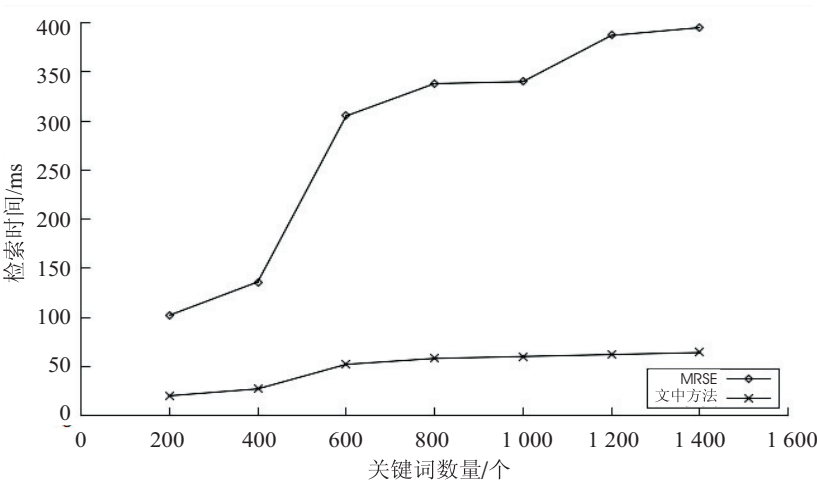
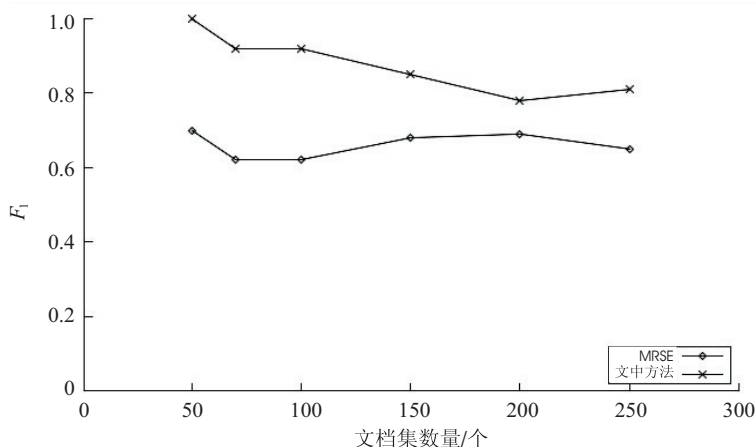


图3 检索时间对比分析

图3给出了文中方法与 MRSE 方法检索时间的变化趋势,可以看出随着文档关键词数量的增加,两种方法的查询时间都与之成正比,但是文中方法比 MRSE 方法大大降低了检索时间。因为文中方法基于标记向量,提前过滤了不相关文档,从而节省了花费在计算不相关文档相关度分数和文档排序上的时间。

图4给出了文中方法与 MRSE 方法的 F_1 比较,其中相关性的判断采用专家评判的方法。可以看出文中方法取得了显著结果,获得了更高的 F_1 值,也即相关度得分更高。因为 MRSE 采用的是关键词精确匹配,没有上升到语义层面的检索,必然不会检索出与关键词语义上相似的文档。

图4 F-measure(F_1)对比分析

4 结束语

针对传统的可搜索加密算法基于关键词精确匹配,查全查准率低,对检索结果没有进行合理的相关度排序,无法满足用户对智能搜索的要求,提出了一种支持语义的可搜索加密方法。该方法利用本体知识库对用户查询进行了语义拓展,通过语义相似度来控制扩展词的规模,防止因拓展词过多产生“查询漂移”。结合向量空间模型实现文档检索,通过文档向量、查询向量分块技术构造出对应的标记向量,过滤无关文档,提高了检索效率,并通过引入语义相似度、关键词不同位置加权评分及关键词-文档相关度得分来构造索引,在查询-文档的相似度得分函数中也引入这三个影响因子,进一步改善了排序效果,从而满足了授权用户对检索结果排序的需求。

参考文献:

- [1] 李经纬,贾春福,刘哲理,等.可搜索加密技术研究综述[J].软件学报,2015,26(1):109-128.
- [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.
- [3] CURTMOLA R, GARAY J A, KAMARA S, et al. Searchable symmetric encryption: improved definitions and efficient constructions[C]//ACM conference on computer and communications security. [s. l.]: ACM, 2006: 79-88.
- [4] WANG C, CAO N, LI J, et al. Secure ranked keyword search over encrypted cloud data[C]//2010 IEEE 30th international conference on distributed computing systems. Genova: IEEE, 2010: 253-262.
- [5] REVATHY B, ANBUMANI A, RAVISHANKAR M. Enab-

ling secure and efficient keyword ranked search over encrypted data in the cloud[J]. International Journal of Recent Advances in Science & Engineering, 2015, 1(1): 28-32.

- [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): 222-233.
- [7] XIA Z H, WANG X H, SUN X M, et al. A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data[J]. IEEE Transactions on Parallel and Distributed Systems, 2016, 27(2): 340-352.
- [8] 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.
- [9] 严小龙,庞晓琼,任孟琦.支持动态更新的多关键词密文排序检索[J].计算机工程与设计,2018,39(4):901-906.
- [10] FU Z, WU X, GUAN C, et al. Toward efficient multi-keyword fuzzy search over encrypted outsourced data with accuracy improvement[J]. IEEE Transactions on Information Forensics and Security, 2016, 11(12): 2706-2716.
- [11] 王恺璇,李宇溪,周福才,等.面向多关键字的模糊密文搜索方法[J].计算机研究与发展,2017,54(2):348-360.
- [12] 何亨,夏薇,张继,等.一种云环境中密文数据的模糊多关键词检索方案[J].计算机科学,2017,44(5):146-152.
- [13] 杨畅,刘佳,蔡圣曜,等.云计算中保护数据隐私的快速多关键词语义排序搜索方案[J].计算机学报,2018,41(6):1346-1359.
- [14] 李陶深,王翼,黄汝维.云环境下支持多用户模糊检索加密算法研究[J].小型微型计算机系统,2016,37(10):2244-2248.