一种高效的基于聚合索引的可搜索加密方案

刘 政[1,2],王瑾璠[1,3],齐竹云[1],吕幸谕[1,4],杨镕玮[1,3]
(1. 鹏城实验室,广东 深圳 518055;
2. 深圳大学,广东 深圳 518061;
3. 南方科技大学,广东 深圳 518055;
4. 广州大学,广东 广州 510006)

摘 要:随着云存储技术的不断发展,用户在享受便捷存储服务的同时对存储数据的安全也有了更高的要求,可搜索加密技术是保障云计算场景中用户数据安全及隐私的重要技术手段。面向云存储的可搜索加密方案除了提供足够强度的安全性,还应具备良好的查询性能及动态更新索引的能力。当前已有的研究工作通常基于反向索引的思路来优化查询性能并实现对索引密文结构的更新。该文提出了一种基于聚合索引的可搜索加密方案。通过建立特定的聚合索引表快速定位关键词所在文件位置,减少关键词查询时的比较次数。实验结果表明,基于聚合索引的加密搜索方案在查询效率方面较经典的反向索引方案有明显优势。同时,该方案也提供了良好的动态索引更新能力和安全强度。

关键词:聚合索引;动态可搜索加密方案;索引表;隐私安全;哈希链表
中图分类号:TP301 **文献标识码**:A **文章编号**:1673-629X(2020)12-0112-06
doi:10.3969/j.issn.1673-629X.2020.12.020

An Efficient Searchable Aggregated-indexing-based Encryption Scheme

LIU Zheng[1,2],WANG Jin-fan[1,3],QI Zhu-yun[1],LYU Xing-yu[1,4],YANG Rong-wei[1,3]
(1. Pengcheng Laboratory,Shenzhen 518055,China;
2. Shenzhen University,Shenzhen 518061,China;
3. South University of Science and Technology of China,Shenzhen 518055,China;
4. Guangzhou University,Guangzhou 510006,China)

Abstract:The development of cloud storage technology enables users to enjoy more powerful cloud storage services. At the same time, users want to preserve the security of stored data as much as possible. Searchable encryption is an important technical means to guarantee user data security and privacy in cloud computing scenarios. In addition to providing sufficient security,the searchable encryption scheme for cloud storage should also have better query performance and the ability to dynamically update indexes. The existing research work is usually based on the idea of reverse indexing to optimize query performance and realize the update of index ciphertext structure. We propose a novel aggregated-index-based searchable encryption scheme, which realizes the fast positioning from keywords to file addresses,by dedicated aggregated indexing table. The experiment shows that the proposed solution has obvious advantages over the classical reverse index scheme in query efficiency. At the same time,it also provides better dynamic index update ability and security strength.

Key words:aggregated index;dynamic searchable encryption;index table;privacy-preserving;hash list

0 引 言

伴随着互联网应用的快速发展,用户在网络环境中所产生的数据迅猛增长,云存储[1]逐渐成为了一种常用的数据存储方式。云存储技术给用户带来极大的便利,而云上数据的安全却持续面临着威胁[2]。当用户的数据以明文形式存储在云上时,云存储提供商或黑客将很容易查看用户数据,用户的数据安全和隐私将无法保障。因此,云存储场景下的数据安全和隐私保护成为近年来的热点话题。作为一种可行的技术路线,云存储的可搜索加密技术[3-4]是指在不解密状态

收稿日期:2019-12-20 **修回日期**:2020-04-21
基金项目:国家自然科学基金面上项目(61872420)
作者简介:刘 政(1995-),男,硕士研究生,研究方向为可搜索加密技术;通信作者:王瑾璠(1990-),男,博士,助理研究员,硕导,研究方向为未来网络体系架构、信息中心网络、软件定义网络、智慧物流、区块链安全与隐私保护。

下,对云上的加密数据进行基于关键词检索的方法。可搜索加密技术的引入,使得用户可以更好地保护数据安全及隐私,同时云存储服务商依然可以提供相关云计算服务。

不同于以往的“先解密,后检索”的密文查询方式,面向云存储的可搜索加密的检索过程是将搜索关键词加密后发送至云端,在云端完成搜索过程后,将包含加密关键词的文件返回到客户端。在整个加密搜索过程中,用户的数据全程处于加密状态,用户的检索关键字也被加密保护,因此用户的数据安全和隐私得到了保障。

1　相关工作

可搜索加密技术最早是由 Song 等人[5]在 2000 年提出,最初的可搜索加密方案不仅无法证明其安全性而且检索性能与数据库内数据规模线性相关。后续经过众多学者的深入研究,使得可搜索加密模型不断发展和完善。2003 年,Goh 等人[6]提出了利用布隆过滤器对每一个文件做一个索引,之后通过多重哈希函数将单个文件中的关键词映射到较短的比特数组。相比较之前的方案,这种方法查询速度快,但是却引入了查询时误判率。并且当文件更新数据内容时索引更新代价较大,因此主要适用于静态文件可搜索加密方案上。之后,Chang 等[7]又提出了“数据字典”的设计以满足对文件内关键词的精确查找。2006 年,Curtmola 等[8]基于对数据大量检索后的状态分析提出了著名的反向索引(Inverted Index)结构。前期的可搜索加密研究工作主要聚焦于静态文件的可搜索加密,但是实际的应用环境中对于文件的修改、增加和删除等操作都是十分常见。因此,2012 年 Kamara[9]提出了动态可搜索加密方案,该方案支持关键词与文件的在线增加或删除。考虑到动态可搜索加密方案的性能,Kamara 等[10]又在原有工作的基础上采用了红黑树数据结构优化查询速度。之后,2014 年 Hahn 等[11]提出了新的可搜索加密方案,这种方案初始化时只构建基础的数据字典,后续伴随着更新和查询不断地更新可搜索密文结构。同时,Emil[12]和 Cash[13]等都提出了各自的动态可搜索加密方案。

在其他方面,Damiani 等[14-15]将可搜索加密技术应用与数据库上,Li 等[16-18]针对模糊语句的可搜索加密查询做了进一步的研究。Nepolean 等[19-20]又在可搜索加密方案上实现了多关键词排列查询和多维范围查询。Fu 等[21-22]又提出了一种有效解决同义词查询的多关键词排序搜索问题。Xia 等[23]针对文件动态增删做出改进,提出了一个可以动态进行文件增删的多关键词查询检索方案。

在上述研究的基础上,该文提出了一种基于聚合索引的云存储可搜索加密方案。该方案具有以下特点:

(1)动态性:支持文件的在线添加或者删除;

(2)高效性:即使在方案初始化阶段或者搜索最坏的情况下仍能保持着较好的查询效率;

(3)安全性:该方案在更新时仅透露一个查找集合。

引入了聚合索引提升关键词的查询效率,通过使用聚合索引可以将单个文件的多个关键词通过聚合函数累加在一起形成聚合索引值,后续按照文件-聚合索引值存储形成一个新的索引表。新的索引表可以在查找过程中快速地定位到查询关键词所在的文件位置,大量减少关键词的比对次数,提升查询效率。在云存储实际业务场景的运用中,该方案具有明显优势。

2　基本方案

可搜索加密技术按照不同的使用场景可能有多种定义和模型,该文的主要研究内容是基于对称可搜索加密模型进行的。一个常见的对称可搜索加密模型定义如下:

2.1　系统模型

在传统的对称可搜索加密方案中,参与者可抽象为数据拥有者、用户和云存储提供商,操作流程如下:

(1)数据加密上传阶段:数据拥有者在本地使用对称加密方式对明文数据进行加密,将加密后的数据上传至云端服务器。

(2)生成检索陷门阶段:用户使用密钥和关键字生成对应的检索陷门,并将该陷门发送给云端服务器。

(3)密文检索阶段:基于收到的关键字检索陷门,云端服务器在密文数据库上对密文进行检索,并将满足检索条件的密文发送给用户。

(4)密文解密阶段:用户从云端获得返回的密文后,使用密钥解密密文。

可搜索加密技术可以分为对称可搜索加密方案和非对称可搜索加密方案,对称可搜索加密方案[1]具有计算量小、算法简单、计算速度快等优点。相对而言,基于公钥密码体制的非对称可搜索加密方案[24-25]则无需事先建立安全信道即可实现用户在公共网络中的保密通信和安全检索,但加密性能较低。考虑到云存储可搜索加密的实际应用情况,文中的研究工作主要围绕对称可搜索加密方案展开。

2.2　符号与定义

方案定义:一个安全的基于索引的动态可搜索加密方案包含的算法具体如下: SUISE = (Gen,Enc,SearchToken,Search,AddToken,Add,Delete,Dec)

$(K,\gamma,\sigma) \leftarrow \mathrm{Gen}(1^{\lambda})$：一个概率算法，输入安全参数 λ，输出密钥 K，搜索索引 γ 和空的搜索历史 σ。

$c \leftarrow \mathrm{Enc}(K,f)$：一个概率算法，输入密钥 K 和文件集合 f，输出加密后的文件集合 c。

$(\sigma',\tau_w) \leftarrow \mathrm{SearchToken}(K,w,\sigma)$：一个概率算法，输入密钥 K，关键词 w 和搜索历史 σ，输出更新后的搜索历史 σ' 和搜索凭证 τ_w。

$(I_w,\gamma') \leftarrow \mathrm{Search}(\tau_w,\gamma)$：一个确定算法，输入搜索凭证 τ_w，文件密文集合 C 和搜索索引 γ，输出文件标志符集合 I_w 和更新后的搜索索引 γ'。

$\tau_a \leftarrow \mathrm{AddToken}(K,f,\sigma)$：一个概率算法，输入密钥 K，文件 f 和搜索历史 σ，输出添加凭证 τ_a。

$(C',\gamma') \leftarrow \mathrm{Add}(\tau_a,c,C,\gamma)$：一个确定算法，输入一个添加凭证 τ_a，一个加密文件 c，加密文件集合 C 和搜索索引 γ，输出更新后的密文集合 C' 和更新后的搜索索引 γ'。

$(C',\gamma') \leftarrow \mathrm{Delete}(\mathrm{ID}(f),C,\gamma)$：一个确定算法，输入一个文件标志符 $\mathrm{ID}(f)$，加密文件集合 C 和搜索索引 γ，输出更新后的密文集合 C' 和更新后的搜索索引 γ'。

$f \leftarrow \mathrm{Dec}(K,c)$：一个确定算法，输入密钥 K 和一个文件密文 c，输出解密后的文件 f。

3 基于聚合索引的可搜索加密方案

在本章节，将从采用的数据结构、索引信息和具体算法三个部分详细阐述基于聚合索引的可搜索加密方案。

3.1 使用数据结构

在本方案中，构建索引表的过程中采用的数据结构有链表和哈希表。对于链表 L，使用 $\mathrm{len}(L)$ 来表示列表的实际存储的元素个数，$x \in L$ 表示列表中包含 x 元素，位置在 i 的元素表示为 $L(i)$。对于哈希表 Map，其存储结构是 $k-v$ 形式的键值对，而存储在 Map 键值为 k 的元素表示为 $\mathrm{Map}[k]=v$，v 可能是一个元素，也有可能是一个链表。

3.2 索引信息

新方案采用了三个索引表来支持服务端的查询功能，三个索引表分别是数据字典、反向索引和聚合索引表。

数据字典 γ_f 用于记录每个文件存储的具体的关键词信息或者关键词加密后的信息。设初始情况下文件集合为 $f=(f_1,f_2,\cdots,f_{\mathrm{len}(f)})$，设 f_i 的关键词为 $f_i=(w_{i_1},w_{i_2},\cdots,w_{i_{\mathrm{len}(f_i)}})$，对应的关键词密文集合为 $c_i=(c_{i_1},c_{i_2},\cdots,c_{i_{len(c_i)}})$，文件标识符为 $\mathrm{ID}(f_i)$。以文件标识符的关键词作为哈希的键值，将文件内关键词密文集合经过哈希运算后以链表形式存储在数据字典中。

反向索引表 γ_w 用于记录重复搜索关键词所对应的文件集合，对于每一个已经搜索过的关键词，可以通过查询 γ_w 提升对重复关键词的查找效率。

最后，在本方案中引入了新的聚合索引表 γ_c 来优化查询性能，具体过程为在数据字典的基础上对于每一个文件所有的密文关键词利用质数哈希函数生成唯一的大质数，并且将单个文件中生成的所有质数累乘后以文件 ID 为键保存在哈希链表 γ_c 中。在后续检索过程中，通过使用质数哈希函数将关键词哈希成质数，再通过聚合索引表 γ_c 快速判断关键词是否在文件中。

引理：给定一个质数集合的连乘积 $M_A=\prod_{i=1}^{n}a_i$，$A=(a_1,a_2,\cdots,a_n)$，对于一个质数 w，如果质数不在集合 A 中，则 $M_A \bmod w > 0$。

证明：假设质数 w 不在集合 A 中并且 $M_A \bmod w = 0$，则 w 必定是 M_A 的一个因数，而 w 又是一个质数，因此 w 必定是 M_A 的一个质因数。假设中质数 w 不在集合 A 中，与假设矛盾，因此 w 一定不在集合 A 中。

聚合索引原理：

(1)对于给定的聚合索引值 $A_c=\prod_{i=1}^{n}p_i$，其中 p_i 为单个文件中基于唯一关键词生成的 τ_w 再经过哈希后生成的质数；

(2)对于新的查询关键词 w，其搜索陷门为 τ_w。令 $P_w=H'(\tau_w)$，P_w 为经过质数哈希函数后生成的质数；

(3)如果 P_w 属于聚合索引值中的一部分，即 $P_w \in (p_1,p_2,\cdots,p_n)$，则 $A_c \bmod P_w = 0$，否则 $A_c \bmod P_w > 0$。

3.3 具体算法

假设有抗不可区分选择明文攻击的对称加密方案 $\mathrm{SKE}=\{\mathrm{Gen},\mathrm{Enc},\mathrm{Dec}\}$，一个输入长度为 λ，输出长度任意的伪随机数生成器 G，一个伪随机函数 $F:\{0,1\}^{\lambda}\times\{0,1\}^{*}\rightarrow\{0,1\}^{\lambda}$，一个随机数生成器 $H:\{0,1\}^{\lambda}\times\{0,1\}^{*}\rightarrow\{0,1\}^{\lambda}$，一个质数哈希函数 $H':\{0,1\}^{\lambda}\times\{0,1\}^{*}\rightarrow\{0,1\}^{\lambda}$。

文中构建的方案 SUISE = (Gen, Enc, SearchToken, Search, AddToken, Add, Delete, Dec) 描述如下：

(1) $(K,\gamma,\sigma) \leftarrow \mathrm{Gen}(1^{\lambda})$。

初始化三个哈希表 γ_f，γ_w，γ_c。生成两个 λ 比特长的二进制串 $k_1 \leftarrow \mathrm{Gen}(1^{\lambda})$，$k_2 \leftarrow \{0,1\}^{\lambda}$，初始化一个集合 σ，返回 (K,γ,σ)，其中 $K=(k_1,k_2)$，$\gamma=(\gamma_f,\gamma_w,\gamma_c)$。

(2) $c \leftarrow \mathrm{Enc}(K,f)$。

对输入的文件 f，输出 $c=\mathrm{SKE}.\mathrm{Enc}_{k_1}(f)$。

(3) $(\tau_w,\sigma') \leftarrow \text{SearchToken}(K,w,\sigma)$。

$K=(k_1,k_2)$,对于输入的关键词 w,计算 w 的搜索凭证,将 $\tau_w = F_{k_1}(w)$ 添加到 σ 中,最后返回 τ_w 和更新后的 σ'。

(4) $(I_w,\gamma') \leftarrow \text{Search}(\tau_w,\gamma)$。

根据采用的三个索引 $\gamma=(\gamma_w,\gamma_f,\gamma_c)$,然后判断 γ_w 中是否含有键值 τ_w。

(a)如果有 $I_w=\gamma_w[\tau_w]$,$\gamma'=\gamma$。

(b)否则令 $a_c = H'(\tau_w)$,γ_c 中逐条记录即 $\gamma_c[\text{ID}(f_i)]$ 对 a_c 取余,判断结果是否为 0。如果取余结果为 0,则对于密文集合 $c'=\gamma_f[\text{ID}(f_i)]$,其中对于每一个密文关键词 $c_i \in c'$,$c_i = l_i \| r_i$ 验证 $H_{\tau_w}(r_i)$ 是否和 l_i 相同,相同则将 $\text{ID}(f_i)$ 加入到 I_w 中。

(c)更新 $\gamma'_w = I_w$,返回 I_w 和 $\gamma'=(\gamma'_w,\gamma_f,\gamma_c)$。

(5) $\alpha_f \leftarrow \text{AddToken}(K,f,\sigma)$。

对于给定的 $K=(k_1,k_2)$,f 是一个包含了文件中唯一关键词的集合,即 $f \subseteq f'$,$f'=(w_1,w_2,\cdots,w_{\text{len}(f)})$。使用伪随机函数 G 生成一系列伪随机值 $s_1,s_2,\cdots,s_{\text{len}(f)}$,同时生成一个空的列表 x 和一个初始化为 1 的数字 index,对于每一个 $w_i \in f'$ 进行如下操作:

(a)计算 w_i 对应的搜索陷门,即 $\tau_{w_i} = F_{k_1}(w_i)$;

(b)如果 τ_{w_i} 在之前检索历史集合 σ 中,则将 τ_{w_i} 加入到 x 中;

(c)令 $c_i = H_{\tau_w}(s_i) \| s_i$;

(d) index = index * $H'(\tau_w)$。

最后添加 $\gamma_c[\text{ID}(f)]$ = index,对集合 $c'=(c_1,c_2,\cdots,c_{\text{len}(f)})$ 按照字典序排序,然后返回 $\alpha_f=(\text{ID}(f),c',X)$。

(6) $(C',\gamma') \leftarrow \text{Add}(\tau_a,c,C,\gamma)$。

按照 AddToken 函数得知 $\alpha_f=(\text{ID}(f),c',X)$,$\gamma=(\gamma_w,\gamma_f,\gamma_c)$。首先向 γ_f 中添加 $\gamma_f[\text{ID}(f)]=c'$。此外对于每一个 $x_i \in X$,将 $\text{ID}(f)$ 添加到 $\gamma_w[x_i]$,再将密文 c 添加到现有密文集合 C 中。返回 C' 和 $\gamma'=(\gamma'_w,\gamma'_f,\gamma_c)$。

(7) $(C',\gamma') \leftarrow \text{Delete}(\text{ID}(f),C,\gamma)$:

给定 $\gamma=(\gamma_w,\gamma_f,\gamma_c)$

(a)对于存储在 γ_w 的每一条链表记录 e,如果 $\text{ID}(f) \in e$,则从 e 中移除 $\text{ID}(f)$ 对应的记录节点;

(b)之后再从密文数据库中移除密文 c,密文数据库更新为 C';

(c)移除 γ_f 中键为 $\text{ID}(f)$ 的链表并删除键值;

(d)移除 γ_c 中键为 $\text{ID}(f)$ 的记录。

最后返回更新后的密文数据库 C' 和更新后的索引 $\gamma'=(\gamma'_w,\gamma'_f,\gamma'_c)$。

(8) $f \leftarrow \text{Dec}(K,c)$。

对于给定的 $K=(k_1,k_2)$,数据输出解密后的文件 $f=\text{SKE.Dec}_{k_2}(c)$。

4　安全性能分析

在本方案中,希望尽量降低在查询、更新和删除操作中泄露原有数据信息给服务端的可能性。因此,详细地评估新方案中这些操作过程的安全性。

做出如下定义:假设有文件集合 F,基于该文件集合的查询历史可以定义为一个二元组(F,phrase),F 代表查询文件集合,phrase 代表历史查询集合。

(1)索引表的隐私性。

为了提供高效率的查询性能,采用了多个基于哈希链表的数据结构存储相关索引信息。索引表存储的信息都进行了加密处理,这样可以避免明文的索引泄露 F 及 phrase 的特征信息(包括文件的内容或用户隐私)。因此,云服务提供者或者对手在没有获取密钥的情况下是无法得到索引存储内容的原有信息。

证明:假设这里有一个非统一对手 $A=(A_1,A_2)$。$k \in N$,k 是一个安全参数。根据 k 生成密钥 K,$K=\text{KeyGen}(1^k)$。之后,A_1 根据密钥参数作为输入生成 (st_A,H_0,H_1),st_A 是一个根据 A 状态生成的字符串。下一步,选择 $\beta \leftarrow \{0,1\}$,令 $H_\beta=(F_\beta,\text{phrase}_\beta)$,生成方案中的索引表集合 I_β 和密文结构 C_β。对于 $2 \leqslant i \leqslant q$,通过搜索陷门生成函数生成 $T_{\beta,i}$ 并作为查询凭证。同样的,令 $\beta' \leftarrow A_2(\text{st}_A,I_\beta,C_\beta,T_\beta)$。

通过保证 $\Pr[\beta'=\beta] \leqslant 1/2+\text{negl}(k)$,而 $\text{negl}(k)$ 可以忽略不计,可以确保新方案是语义安全的。同时新方案能够很好地保证文件关键词及索引信息不会泄露给服务端,从而保护了数据安全及用户的隐私。

(2)搜索陷门生成的安全性。

在新方案的查询过程中,为了确保原有查询关键词不会被用户泄露,使用了伪随机的加密和哈希算法对原有查询关键词进行混淆。因此,服务端无法通过基于关键词生成的搜索陷门猜解原有查询关键词的内容信息。

证明:假设存在一个竞争对手 A 和一个模拟器 S。首先模拟器对于关键词查询集合 p 生成搜索陷门集合 T_p。之后模拟器选择一个关键词 w' 并生成 w' 的搜索陷门 τ'_w,再将 τ'_w 提交给 A,A 根据 τ'_w 输出对原有关键词的猜测 w''。这里对于一个在多项式时间给出结果的竞争对手 A 给出,我们有 $\Pr[w''=w'] \leqslant 1/(|W|-m)+\text{negl}(k)$。因此,新方案可以保证搜索关键词的安全。

(3)云端存储文件的安全性。

在本方案中,用户存储在云上的数据全程均处于加密状态。文件拥有者或者得到文件拥有者授权的用

户通过密钥对加密存储的文件进行解密而获得明文，但密钥的生成、分发和加密、解密过程，均不是在云上完成的。因此，云服务提供商并不掌握用户对文件进行加密的密钥，无法解密用户文件。由此可以得出结论，在本方案中，用户的文件在云端存储的过程中是安全的。

5 性能分析

5.1 空间复杂度分析

本方案中的存储主要包括存储用户数据和相关文件索引信息，这里只分析相关索引信息所占用的空间大小。用 n 表示文件的个数，m 为文本数据库中所有关键词的总量，则 γ_f 索引表所占用的空间大小为 $O(m)$。γ_w 索引表用于存储重复关键词的搜索结果，则 γ_w 的空间消耗为 $\sum_{w}\text{len}(f_w)$。γ_c 索引表用于存储不同文件的聚合索引值，其空间消耗为 $n*p$，其中 p 为单个文件聚合索引值长度。σ 用于存储每个文件都被搜索过哪些关键词，最坏情况为每个文件每个关键词都被搜索过，空间复杂度为 $O(m)$。整体方案的空间占用为 $O(m+n*p+\sum_{n}\text{len}(f_i))$，而 $m>n*p$，$m>\sum_{n}\text{len}(f_i)$，因此整体方案的空间复杂度为 $O(m)$。

5.2 时间复杂度分析

假设 m 为 γ_f 中存储的所有关键词的总量，m' 为 γ_f 中存储的唯一关键词总量。在本方案中，在经过一段时间运行后所有关键词都被检索过，此时时间复杂度经过均摊后为 $O(m/m')$。

考虑到数据库所有关键词都是独一无二的情况下，新的方案相比较之前反向索引的方案，在整体方案初始化运行情况下使单个关键词的查询时间复杂度从 $O(n*\text{len}(f_i))$ 降到 $O(n+\text{len}(f_i))$，其中 $\text{len}(f_i)$ 代表了单个文件平均关键词的个数。

5.3 查询时间测试

测试数据集包括 50 个英文文档，从 100 个文档中剔除英文标点符号后用空格将单词隔开。其中每个文件只选出前 100 到 900 个不同关键词作为文件内容。每次实验中对于所有文件中每个文件选出的所有关键词逐个进行检索以确保整体方案性能的稳定性。

实验环境在 Intel Core i7-8565U CPU 1,99 GHz，16 GB 内存，windows10 professional 系统环境下进行，开发环境为 java JDK1.8，开发工具为 Eclipse(版本为 2019-06(4.12.0))。实验过程中选取了基础的数据字典、反向索引和提出的聚合索引方案进行单个关键词查询时间对比，结果如表 1 所示。

表 1 不同方案在不同文件关键词个数下的查询时间 s

文件关键词个数	数据字典	反向索引	聚合索引
100	32.9	28.9	4.59
200	135.36	118.06	9.27
300	503.83	435.3	14.67
400	1 166.48	976.19	20.59
500	2 250.89	1 867.7	28.5
600	3 862.33	3 161.54	38.83
700	6 080.4	4 954.58	52.06
800	9 316.1	7 486.85	72.31
900	13 164.76	10 549.43	103.23

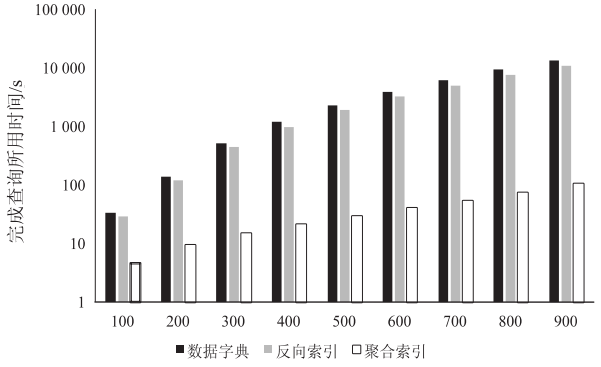

图 1 不同方案的查询时间和文件关键词个数的关系

图 1 是根据单次查询所消耗时间进行绘制的折线图，横轴为给定不同的关键词数量，纵轴为单次查询所消耗的时间经过以 10 为底对数运算后的结果，不同标注的线代表了不同的方案。

6 结束语

针对云存储场景中的数据安全和隐私保护问题，提出了一种高效的基于聚合索引的加密搜索方案。在该方案中，查询过程中的关键词比对次数大幅度减少，查询性能得到优化。对该方案进行了理论分析和实现，实验结果表明，该方案相较于现有方案展示出了更

佳的查询性能,同时提供了良好的动态更新能力和安全性能。

参考文献:

[1] 梁知音,段　镭,韦　韬,等. 云存储安全技术综述[J]. 电子技术应用,2013,39(4):130-132.

[2] 郭禹彤. 云存储环境下个人用户隐私信息保护研究[D]. 长春:吉林大学,2019.

[3] 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.

[4] 李　颖,马春光. 可搜索加密研究进展综述[J]. 网络与信息安全学报,2018,4(7):13-21.

[5] SONG D X,WAGNER D,PERRIG A. Practical techniques for searches on encrypted data[C]//Proceeding 2000 IEEE symposium on security and privacy. Berkeley, CA: IEEE, 2000:44-55.

[6] GOH E J. Secure indexes[R/OL]. 2003. http://eprint. iacr. org/2003/216.

[7] CHANG Y C, MITZENMACHER M. Privacy preserving keyword searches on remote encrypted data[C]//International conference on applied cryptography and network security. Berlin,Heidelberg:Springer,2005:442-455.

[8] CURTMOLA R,GARAY J,KAMARA S,et al. Searchable symmetric encryption: improved definitions and efficient constructions[J]. Journal of Computer Security, 2011, 19(5):79-88.

[9] KAMARA S,PAPAMANTHOU C,ROEDER T. Dynamic searchable symmetric encryption[C]//Proceedings of the 2012 ACM conference on computer and communications security. [s. l.]:ACM,2012:965-976.

[10] KAMARA S, PAPAMANTHOU C. Parallel and dynamic searchable symmetric encryption[M]//Financial cryptography and data security. Berlin,Heidelberg:Springer,2013:258-274.

[11] HAHN F,KERSCHBAUM F. Searchable encryption with secure and efficient updates[C]//Proceedings of the 2014 ACM SIGSAC conference on computer and communications security. [s. l.]:ACM,2014:310-320.

[12] STEFANOV E,PAPAMANTHOU C,SHI E. Practical dynamic searchable encryption with small leakage[C]//Network and distributed system security symposium. Virginia: ISOC,2014:72-75.

[13] CASH D,JAEGER J,JARECKI S,et al. Dynamic searchable encryption in very-large databases: data structures and implementation[C]//Network and distributed system security symposium. Virginia:ISOC,2014:23-26.

[14] DAMIANI E,VIMERCATI S D C,JAJODIA S,et al. Balancing confidentiality and efficiency in untrusted relational DBMSs[C]//Proceedings of the 10th ACM conference on computer and communications security. [s. l.]:ACM,2003: 93-102.

[15] HACIGÜMÜŞ H,IYER B,MEHROTRA S. Efficient execution of aggregation queries over encrypted relational databases[J]. Lecture Notes in Computer Science,2004,2973:125-136.

[16] LI J,WANG Q,WANG C,et al. Fuzzy keyword search over encrypted data in cloud computing[C]//Proceedings IEEE INFOCOM. San Diego,CA:IEEE,2012:1-5.

[17] LIU Z,MA H,LI J,et al. Secure storage and fuzzy query over encrypted databases[C]//International conference on network and system security. Berlin, Heidelberg: Springer, 2013:439-450.

[18] 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.

[19] NEPOLEAN D, KARTHIK I, PREETHI M, et al. Privacy pre-serving ranked keyword search over encrypted cloud data[M]//Recent trends in computer networks and distributed systems security. Berlin, Heidelberg: Springer, 2014: 396-403.

[20] 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,Italy:IEEE,2010:253-262.

[21] FU Z,SUN X,LINHE 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.

[22] 裴树军,易　鑫,陈彦橦,等. 可问责的多关键字可搜索加密方案[J]. 计算机科学与探索,2019,13(10):1710-1720.

[23] XIA Z, WANG X, SUN X, et al. A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data[J]. IEEE Transactions on Parallel & Distributed Systems,2016,27(2):340-352.

[24] DAN B,CRESCENZO G D,OSTROVSKY R,et al. Public key encryption with keyword search[M]//Advances in cryptology - Eurocrypt. Berlin, Heidelberg: Springer, 2003: 506-522.

[25] 秦志光,徐　骏,聂旭云,等. 公钥可搜索加密体制综述[J]. 信息安全学报,2017,2(3):1-12.