一种基于云存储的多服务器多关键词可搜索加密方案

黄海平*①②　杜建澎①②　戴　华①　王汝传①②
①(南京邮电大学计算机学院　南京　210003)
②(江苏省无线传感网高技术研究重点实验室　南京　210003)

摘　要：在可搜索加密的云服务中，数据拥有者往往更希望将数据文件以密文的形式分别存储到多个云服务器，从而提高授权用户对云端数据的检索效率以及对大型数据的处理能力。基于此，该文提出一种基于云存储的多服务器多关键词多用户可搜索加密方案，该方案被证明是 IND-CKA(adaptive Chosen Keyword Attack)安全的，且同时具备关键词陷门的安全性。相对于单服务器可搜索加密，该方案在保证数据机密性的前提下能够对其进行高效检索，并能够在关键字索引中不完全包含所检索的多个关键词或者不存在某个文件包含所有被检索的多个关键词的情况下，更精确地进行检索。

关键词：可搜索加密；云存储；多服务器；多关键词
中图分类号： TP393　　**文献标识码**： A　　**文章编号**：1009-5896(2017)02-0389-08
DOI: 10.11999/JEIT160338

Multi-sever Multi-keyword Searchable Encryption Scheme Based on Cloud Storage

HUANG Haiping①②　DU Jianpeng①②　DAI Hua①　WANG Ruchuan①②
①(*Institute of Computer*, *Nanjing University of Posts and Telecommunications*, *Nanjing* 210003, *China*)
②(*High Technology Research Key Laboratory of Wireless Sensor Network of Jiangsu Province*, *Nanjing* 210003, *China*)

Abstract: In the searchable encryption services provided by cloud severs, data owners expect that their data files can be partitioned and stored to multiple cloud severs with the form of ciphertext, so as to improve the searching efficiency of authorized users and the processing ability of big data. For this issue, a multi-sever multi-keyword searchable encryption scheme is proposed based on cloud storage, and the proposed scheme is proved to be IND-CKA (adaptive Chosen Keyword Attack) secure coexisting with the secure trapdoor. Compared with the single sever searchable encryption, the proposed scheme can not only guarantee the data security, but also provide more accurate retrieval service when the keyword index or any one file does not contain all of the searching keywords.

Key words: Searchable encryption; Cloud storage; Multi-sever; Multi-keyword

1　引言

随着云计算技术的发展，云服务商为人们提供了更高效和更快捷的文件存储和处理服务，因此，越来越多的用户将他们的数据存储在云端。然而，用户在享受便捷的云存储服务的同时，也在担心存储在云端的私密数据得不到安全保障。常见的解决方法是使用可搜索加密技术，Song 等人[1]最早提出的不可信赖服务器问题，该问题基于密文扫描思想和对称密钥机制提出了第 1 个 SSE(Symmetric Searchable Encryption)方案，命名为 SWP 方案。

Boneh 等人[2]最早提出 PEKS(Public Key Encryption with Keyword Search)的概念，并基于 BF-IBE 加密方案(Boneh 和 Franklin 设计出的 IBE 方案)构造了第 1 个基于公钥的可搜索加密方案 BDOP-PEKS[3]。近年来，提出了各种类型的 SSE[4,5] 方案和 PEKS[6,7]方案。其中，文献[4]基于索引思想提出了 Z-IDX 方案，每个文件使用布隆过滤器作为

收稿日期：2016-04-07；改回日期：2016-09-01；网络出版：2016-11-14
*通信作者：黄海平　hhp@njupt.edu.cn
基金项目：国家自然科学基金(61373017, 61373138, 61300240, 61672297)，国家博士后基金(2015M570468, 2016T90485)，江苏省自然科学基金(BK20151511)，江苏省六大人才高峰项目(DZXX-017)，江苏省无线传感网高技术研究重点实验室基金(WSNLBZY 201516)，江苏省研究生培养创新工程项目(KYLX15_0853)
Foundation Items: The National Natural Science Foundation of China (61373017, 61373138, 61300240, 61672297), The National Postdoctoral Foundation of China (2015M570468, 2016T90485), The Natural Science Foundation of Jiangsu Province (BK20151511), The Six Major Talent Peak Foundation in Jiangsu Province (DZXX-017), The High Technology Research Key Laboratory of Wireless Sensor Network Foundation of Jiangsu Province (WSNLBZY 201516), The Graduate Education Innovation Project of Jiangsu Province (KYLX15_0853)

索引结构，每个文件所包含的关键词被映射为码字并存储在该文件的索引中，通过运算布隆过滤器，就能判定某关键字是否存在于密文文件中；文献[6]定义了一种安全模型，在具有恶意服务器和恶意接收者的内部攻击环境下, 提出了 dPEKS 方案(PEKS scheme with designated server)，在随机语言机模型下被证明是安全的, 并且能够抵御关键字猜测攻击。

与单用户模型相比，在实际的云存储环境中，多用户模型下的可搜索加密(Multi-User Searchable Encryption, MUSE)是更具实用性的应用。多个授权用户可共享外包至云服务器的文件，其经典方案包括在文献[8,9]中。其中，文献[8]基于一般访问结构提出一种多用户可搜索加密方案，该方案中任何用户都可以向外包数据库添加加密数据，且任何授权用户都可以检索并解密接收到的密文；文献[9]基于混合结构提出一个实现关键字精确检索和细粒度访问控制加密数据的可搜索加密系统，然而该方案允许服务器获得关键词信息，有较高的安全风险。因此，在 MUSE 模型中，现存的主要问题是如何控制哪个用户可以访问哪些文件。

文献[10]提出了一种密钥融合的可搜索加密(Key-Aggregate Searchable Encryption, KASE)方案，该方案不同于以往方案中数据所有者需要向用户分发大量密钥，它只需要向授权用户分配一个密钥就能进行加密和检索等操作。文献[11]基于身份加密机制设计了一种新的公钥可搜索加密方案，加密方通过传给服务器一个密钥，使得服务器仅能识别加密方期望服务器识别的关键词而无法获得更多的信息。然而上述两个方案均只支持单关键词检索。

相较于单关键词检索, Golle 等人[12]最早研究多关键词检索(Multi-Keyword Searchable Encryption, MKSE)问题, 并提出 GSW-1 方案。近几年的 MKSE 方案有文献[13-15]。其中，文献[13]基于代理重加密提出指定检验者的 RE-DPECK 方案，能够抵御关键词猜测攻击，且可使用代理检索功能。文献[14]首次提出非结构化文本的多关键词可搜索加密方案，其无需指定关键词的位置，但每次搜索均针对文件中的所有关键词，而不是有选择地进行；文献[15]基于特定树形索引结构设计“贪心深度优先搜索”算法提供高效的多关键词排序检索，使文件的删除和插入更加灵活。然而，现有的 MKSE 方案大多集中于“逻辑与”连接词。

出于安全性和效能的考虑，大数据的拥有者更加期望能将海量数据分开存储到多个云服务器，这对于单服务器模型提出了巨大的挑战。因此，如何在多服务器模型中实现多授权用户使用不同的多关键字进行密文检索，同时保障安全和隐私成为本文要解决的关键性问题。针对此，本文方案的主要贡献可阐述如下：

(1) 提出了一种多服务器可搜索加密方案，使得多个服务器可以并行执行密文检索的任务。相比单服务器模型，提高了对大型数据的处理能力和用户存储在云端的数据的安全性。

(2) 提供更精确的检索服务，在关键字索引中不完全包含所检索的多个关键词或者不存在某个文件包含所有被检索的多个关键词的情况下，仍然能够为用户检索到正确的数据。

(3) 更灵活地控制用户对密文文件的访问权限，不同用户对于同一文件的访问权限不相同，并支持动态更新用户的访问权限，进一步解决了多用户在云端共享数据的安全问题。

2 准备工作

2.1 系统模型

如图 1 所示，本文的系统模型包括 4 个部分：数据拥有者、授权用户、云端服务器 S 和服务器 $S_1, S_2, \cdots, S_N$。其中，数据拥有者是数据的加密方，他将数据加密后上传至云端，并且可以授权用户拥有检索的权限；用户获得授权后可根据选定的关键词从云端检索相应的密文文件；云端服务器 S 专用于存储数据拥有者加密后上传的关键词索引，同时为用户提供部分检索服务；$S_1, S_2, \cdots, S_N$ 为各云存储服务商提供的服务器，用来存储数据拥有者上传的密文文件和服务器 S 重加密后的密文索引，同时提供部分检索服务。

此处, 我们假设云存储服务器是诚实且好奇的, 服务器会正确的执行检索操作并且返回全部检索结果，但无时无刻均想获取数据隐私。假设云端服务器有 m 个用户拥有访问权限，即 $\text{Users} = \{u_1, u_2, \cdots, u_m\}$，被加密并上传云端的文件有 n 个，即 $\text{Files} = \{f_1, f_2, \cdots, f_n\}$。文件上传服务器之前，数据拥有者首先需要为 Files 创建其关键词索引 I，从而方

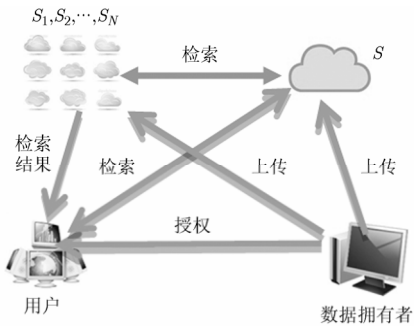

图 1 多服务器可搜索加密系统模型

便密文检索，假设文件的关键词为 $W=\{w_1, w_2,\cdots,w_d\}$，这些关键词分别来自 Files 中的文件，数据拥有者将加密后的索引 I 发送给服务器 S，加密后的 Files 按照某种规则等分为 N 份分别发送给服务器 $S_1,S_2,\cdots,S_N$；服务器 S 对索引 I 进行重加密后，将生成的文件关键词索引 S 结构(如图 2)再发送给 $S_1,S_2,\cdots,S_N$。

2.2 安全性定义

系统安全性定义为：在知道索引和关键词陷门的前提下，云端服务器无法获得索引中关键词的其它任何信息，即服务器无法利用关键词 w_i ($w_i\in\{w_1,w_2,\cdots,w_d\}$, $1\le i\le d$)的陷门构造出一个新的关键词 w_j 的陷门；另外，即使服务器可以欺骗授权用户为其生成任何关键词的陷门，此安全性也成立，称为自适应选择关键词攻击下的不可区分性(IND-CKA)[4]。

定义 1　不可忽略函数：　对于任意的多项式 $f(t)$，总存在一个自然数 M 使对于所有 $t>M$ 都有 $p(t)<1/f(t)$ 则称 $p(t)$ 为可忽略函数。反之，则称 $p(t)$ 为不可忽略函数。

定义 2　选择关键词攻击 IND-CKA：　挑战者 C 和攻击者 A 之间的安全性游戏如下：

系统建立：运行初始化函数 Setup(k)，将辅助密钥 s' 发给挑战者 C，值 h' 发给攻击者 A。

挑战前查询：攻击者 A 向挑战者 C 自适应地选择多项式个关键词 W_i 的密文索引 I_i。

挑战：攻击者 A 随机选择两个未向挑战者 C 查询过的关键词 w_0', w_1' 并发送给 C。C 随机选择 $t'\in\{0,1\}$，返回 $w_{t'}'$ 的密文索引 I_b 给 A。

挑战后查询：挑战者 A 向攻击者 C 适应性地选择查询关键词的密文索引，但不能查询关键词 w_0', w_1' 的密文索引，查询次数为 k 的多项式次，其中 k 为安全参数。

输出：挑战者 C 输出密文索引 I_b 的 b 的猜测 $b_{\mathrm{A}}\in\{0,1\}$。如果 $b_{\mathrm{A}}=b$，则称攻击者 A 给出正确猜测并赢得游戏。A 赢得游戏的概率定义为 $\mathrm{Adv}_{\mathrm{A}}(k)=\left|\Pr[b_{\mathrm{A}}=b]-1/2\right|$。

定义 3　如果对于任意的多项式时间攻击者 A, $\mathrm{Adv}_{\mathrm{A}}(k)$ 是安全参数 k 的一个可忽略函数，则称基于多关键词的可搜索加密方案是 IND-CKA 安全的。

本方案的安全性证明是基于确定性 Diffe-Hellman 问题(DDHP)，定义如下：

定义 4　确定性 Diffe-Hellman 问题(DDHP)假设：G_1 是由 g 生成的阶为 p 的循环群，给定 (g^a,g^b,g^{ab}) 和 (g^a,g^b,g^c)，其中 $a,b,c\in Z_p^*$，DDHP 问题就是判断 c 是否等于 ab。

定义 5　自适应 DDH 假设：对于任意概率多项式时间的攻击者 A，存在一个可忽略函数 u 满足 $\left|\Pr_{a,b\in Z_p^*}[\mathrm{A}(G_1,g,g^a,g^b,g^{ab})=1]-\Pr_{a,b,c\in Z_p^*}[\mathrm{A}(G_1,g,g^a,g^b,g^c)=1]\right|<u(k)$，$k$ 是安全参数，则说明群 G_1 中的 DDHP 是困难的。

3　多服务器模型下多关键词可搜索加密方案

3.1 数据结构

本文中多用户可搜索加密方案所用到的数据结构如下：

(1)用户表：数据拥有者有权添加和删除用户，其执行算法 AddUser(u_i)，将 $u_i=\{a_1,a_2,\cdots,a_x\}$ 发送到云端服务器 $S_1,S_2,\cdots,S_N$ 保存的用户表(如表 1)，完成授权用户添加工作。如果需要改变用户访问权限，数据拥有者只需添加或者删除用户表中用户的相关属性 a_x。

表 1 用户表

User	Attribute
u_1	a_1,a_2,a_5,a_9
⋮	⋮
u_m	$a_1,a_2,\cdots,a_x$

(2)文件访问权限表：数据拥有者通过表 1 中的用户属性和表 2 所示的文件访问权限表中的文件属性来控制用户对文件的访问权限，其中 X_i 是事先设定好的某个值。如果用户 u_1 要检索文件 f_1，我们首先找到文件访问权限表中的 f_1 项和用户表中的 u_1 项，然后对比 u_1 拥有的属性和访问 f_1 需要的属性，计算两者交集($A(u_1)\cap A(f_1)$)的属性个数，如果大于某个设定的值，即 $A(u_1)\cap A(f_1)\ge X_1$，说明 u_1 有权访问 f_1；否则无权访问 f_1。

(3)文件关键词索引结构：如图 2 所示，$\mathrm{id}(f_i)$ 表示文件 f_i 的标识符，标识符节点后的列表是该文件所包含的关键词，服务器在图 2 中检索，找出用户想要检索的文件。

表 2 文件访问权限表

文件	阈值	属性
f_1	X_1	a_1,a_2,a_5,a_8
⋮	⋮	⋮
f_n	X_n	$a_1,a_2,\cdots,a_z$

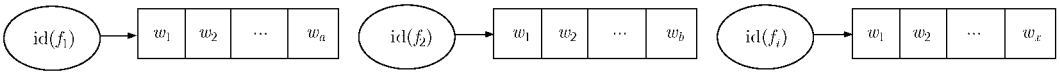

图 2 文件关键词索引结构

3.2 算法描述

多服务器模型下多用户多关键词可搜索加密方案包括以下 8 个步骤：

(1) Setup(k)：数据拥有者执行该算法，输入安全参数 k，输出一个由 g 生成的阶为 p 的循环群 G，且 G 的 DDHP 是困难的。随机选取主密钥 $s \in Z_p^*$ 和辅助密钥 $s' \in Z_p^*$，计算 $h = g^s$，$h' = g^{s-s'}$；随机选择伪随机函数 $f: \{0,1\}^k \times Z_p^* \to Z_p^*$ 及随机参数 $t \in \{0,1\}^k$；选择一个哈希函数 $H(x): \{0,1\}^* \to Z_p^*$ 和另一个哈希函数 $H'(x): \{0,1\}^* \to Z_p^*$；选择分组加密算法 $\mathrm{En}(\cdot)$ 的对称密钥 K，发布系统参数 $\mathrm{params} = (G, g, p, f, H, H', h, \mathrm{En}(\cdot))$。

(2) AddUser(u_i)：数据拥有者执行该算法以添加用户，根据用户的访问权限生成 $u_i = \{a_1, a_2, \cdots, a_x\}$ 并发送到云端服务器 $S_1, S_2, \cdots, S_N$ 的用户表，其中 a_x 是用户的属性(可以是一个数值)，服务器通过对比用户拥有的属性和访问文件所需要的属性，从而判断用户是否有权访问文件，然后将 (t, K, s') 安全地发送给 u_i。

(3) Encrypt(K, s', t, D, W)：数据拥有者执行该算法加密关键词和文件，输入分组加密密钥 K，辅助密钥 s'，随机参数 t，文档 D 及其关键词列表 $W = \{w_1, w_2, \cdots, w_d\}$，随机选择 $r \in Z_p^*$，计算 $g^r, h'' = (h')^r$ 和 h^r，将 h'' 发送给服务器 S。计算 $\delta_i = f[t, H(w_i)]$，$E(w_i) = g^{r(s'+\delta_i)}$，其中 $1 \le i \le d$，令 $I = (g^r, h^r, E(w_1), E(w_2), \cdots, E(w_d))$，计算 $C = \mathrm{En}_K(D)$。将 I 和加密后的文件关键词索引结构发送给服务器 S，并将密文文档 C 按照某种规则等分成 N 份分别发送给服务器 $S_1, S_2, \cdots, S_N$。

(4) S-Encrypt(I)：服务器 S 执行该算法对索引 I 进行重加密，输入索引 I 和接收到的 h''，计算 $E'(w_1) = g^{r(s'+\delta_1)} \cdot h'' = (h \cdot g^{\delta_1})^r, \cdots, E'(w_d) = g^{r(s'+\delta_d)} \cdot h'' = (h \cdot g^{\delta_d})^r$，$I' = (g^r, h^r, E'(w_1), E'(w_2), \cdots, E'(w_d))$，计算 $H'[E(w_1)], \cdots, H'[E(w_d)]$ 替换图 2 中对应的关键词，将新生成的文件关键词索引结构发送给服务器 $S_1, S_2, \cdots, S_N$。

(5) GenerateTrapdoor($s', t, w_1', w_2', \cdots, w_\omega'$)：授权用户执行该算法生成关键词陷门，输入 s'，t 和要检索的关键词 $w_1', w_2', \cdots, w_\omega'$，随机选择 $t'' \in Z_p^*$，计算 $Y = (g^r)^{t''}$。对每个关键词 w_i'，计算 $T_i = t'' + f(t, H(w_i')) + s'$，其中 $1 \le i \le \omega$。将陷门 $T = (T_1, T_2, \cdots, T_\omega, Y)$ 发送给服务器 S。

(6) Search(T, I, C)：服务器 S 执行该算法，输入 h''，陷门 $T = (T_1, T_2, \cdots, T_\omega, Y)$ 和索引 I，对于 $1 \le i \le \omega$，计算 $(g^r)^{T_i} \cdot h'' / Y = g^{r[s+f(t,H(w_i'))]} = (h \cdot g^{\delta_i'})^r$，$\delta_i' = f[t, H(w_i')]$ 和 $H_i' = H'[(h \cdot g^{\delta_i'})^r]$，将 $I'' = (H_1', H_2', \cdots, H_\omega')$ 发送给服务器 $S_1, S_2, \cdots, S_N$。

然后，服务器 S_j ($j = 1, 2, \cdots, N$)分别将 I'' 中的哈希值与文件关键词索引结构中每个 id(f_i) 节点中关键词的哈希值($H'[E'(w_i)]$)进行比较，每匹配成功一次则该文件匹配度加 1。S_j 选择匹配度最大的文件(S_j 中包含授权用户检索关键词数目最多的文件)，记为 f_x (可能是多个文件)，然后查询 f_x 的文件访问权限表和用户表中的 u_i，计算 $m = A(u_i) \cap A(f_x)$ (如果 f_x 是多个文件则需计算多次)，如果 $m \ge X_i$，则将 f_x 加入到返回结果 Result 中；否则 u_i 没有权限访问 f_x，S_j 继续验证匹配度次高的文件，直到找到为止；如果没有任何匹配的文件则返回 False。然后将最高的匹配值发送给服务器 S，因此如果关键字索引中不完全包含所检索的多个关键词或者不存在某个文件包含所有被检索的多个关键词，服务器仍能够提供精确检索。

其余($N-1$)个服务器并行重复上述工作，S 比较后通知拥有匹配度最大的服务器 S_j 将其相对应的文件 C' 发送给用户；如果没有任何匹配的文件则服务器 S 给用户返回检索失败。

(7) Decrypt(K, C')：用户执行该算法解密密文，输入分组密钥 K 和接收到的密文 C'，对于 $\forall f_i \in C'$，计算 $f_i = \mathrm{Dec}_K(\mathrm{En}_K(f_i))$。

(8) UpdateUser(u_i)：通过添加或者删除用户表中 u_i 的属性来改变用户的访问权限，如果要撤销用户可以直接将其从用户表中删除。

4 安全和效率分析

4.1 方案的安全性分析

定理 1 上述方案满足完备性。

证明 如果所有数据都按照方案中所描述生成，并且 $f(t, H(w_i')) = \delta_i, 1 \le i \le d$，那么 $(g^r)^{T_i} \cdot h'' / (g^r)^{t''} = g^{r[t''+s+f(t,H(w_i'))]} \Big/ g^{rt''} = \left(h \cdot g^{f(t,H(w_i'))}\right)^r = (h \cdot g^{\delta_i})r$，

即有 $H'\left[\left(h \cdot g^{\delta_i}\right)^r\right] = H'\left[E'\left(w_i'\right)\right]$。 证毕

定理 2 如果 DDH 假设成立，那么不存在一个攻击者可以选择关键词攻击破坏该系统，因此上述方案是 IND-CKA 安全的。

证明　若攻击者 A(存储服务器)以不可忽略的概率 ε 赢得游戏 IND-CKA(3-B 节定义)，则挑战者 C 利用攻击者 A 能以不可忽略的概率 ε 解决挑战 DDHP。挑战者 C 开始第 2 节定义的挑战游戏，进行方式如下：

系统建立：选择 DDHP 参数为 $(h_1=g^a,h_2=g^b,h_3=g^c)$，选择伪随机函数 $f:\{0,1\}^k\times Z_p^*\to Z_p^*$ 及随机参数 $t\in\{0,1\}^k$；选择一个哈希函数 $H(x):\{0,1\}^*\to Z_p^*$，计算 $h=h_1=g^a$，将 $\text{params}=(G,g,p,f,h,H)$ 作为公共参数。

挑战前查询：攻击者 A 向挑战者 C 询问关键词列表 $W_i=\{w_1',w_2',\cdots,w_m'\}$ 的密文索引，挑战者 C 将 W_i 加密后发送给 A。加密方式如下：挑战者 C 随机选择 $r\in Z_p^*$，计算 g^r 和 h^r，对 $\forall w_j'\in W_i$，$1\le j\le m$，计算 $\delta_j'=f\left[t,H\left(w_j'\right)\right]$，$E(w_j')=\left(h\cdot g^{\delta_j'}\right)^r$，输出 $I=\left(g^r,h^r,E\left(w_1'\right),E\left(w_1'\right),\cdots,E\left(w_m'\right)\right)$，将 I 发送给 A。

挑战：攻击者 A 随机选择两个未向挑战者 C 查询过的关键词 w_0'，w_1' 并发送给 C。C 随机选择 $t'\in\{0,1\}$，计算 $\delta_{t'}'=f[t,H(w_{t'})]$，同时随机选择 $r'\in Z_p^*$，计算 $E\left(w_{t'}'\right)=\left(h_3\cdot h_2^{\delta_{t'}'}\right)^{r'}$，输出 $w_{t'}'$ 的加密索引 $I_{t'}=\left(h_2^{r'},h_3^{r'},E\left(w_{t'}'\right)\right)$ 并返回给 A。

挑战后查询：挑战者 A 向攻击者 C 适应性地选择查询关键词的密文索引，但不能查询关键词 w_0'，w_1' 的密文索引，查询次数为 k 的多项式次。

输出：攻击者 A 输出密文索引 I_b 中对 b 的猜测 $b_A\in\{0,1\}$。如果 $b_A=b$，则称 A 猜测正确并赢得游戏，此时 C 回答 DDHP 挑战中 $c=ab$；否则 A 失败，则 C 回答 $c\ne ab$。

若 $c=ab$，则 $w_{t'}'$ 的关键词索引密文 $I_{t'}=\left(h_2^{r'},h_3^{r'},\left(h_3\cdot h_2^{\delta_{t'}'}\right)^{r'}\right)=\left(g^{br'},h^{br'},\left(h\cdot g^{\delta_{t'}'}\right)^{br'}\right)$ 是一个正确的关键词密文，这里 $h=h_1=g^a$，$h_3=g^c=h^b$；否则如果 $c\ne ab$，则 $I_{t'}$ 不是一个正确关键词列表的密文。因此，如果 A 赢得游戏，则 $I_{t'}$ 一定是正确关键词列表的密文，从而一定会有 $c=ab$。如果 $c\ne ab$，则 A 没有概率赢得游戏。

综上所述，如果 A 赢得游戏的不可忽略概率为 ε，则挑战者 C 能够以不可忽略概率解决 DDHP 挑战。所以，如果 G_1 群中的 DDH 假设成立，则本文方案满足 IND-CKA 安全。　　证毕

另外，文件采用对称分组加密算法(如 DES)来保障机密性。关键词采用伪随机函数 f' 和随机参数 t 加密，由于攻击者不知道 t，其即使获得关键词密文也无法得知明文的任何信息，因此生成的陷门是安全的。最后，由于文件索引和文件本身被分开存储在服务器 S 和 $S_1,S_2,\cdots,S_N$ 中，因此服务器在图 2 中分别检索每个关键词时，并不会得到比检索结果多的额外信息(如哪些密文包含哪些关键词)。此外，在算法 Search(·) 中可设计一种安全多方计算协议，服务器 $S_1,S_2,\cdots,S_N$ 将各自求得的最大度值按照此协议加密(或混淆)后再将密文传给服务器 S，S 通过安全多方计算协议仍然能够比较出最大值，但无法获知各服务器传来的具体数值，因此在多服务器模型下，用户存储在云端的数据比单服务器模型的安全性要更高。

4.2 方案的效率分析

(1) 算法性能评价：针对本文方案中的 Encrypt(·)，GenerateTrapdoor(·) 和 Search(·) 算法进行仿真实验分析，具体如图 3 所示。

(a)加密算法中对关键词的加密需要一次指数运算，其它运算量与之相比可以忽略不计。这里加密关键字 $W=\{w_1,w_2,\cdots,w_d\}$ 以生成关键字密文索引，首先，数据拥有者计算 $\delta_i=f[t,H(w_i)]$，$E(w_i)=g^{r(s'+\delta_i)}$ 生成关键字索 $I=(g^r,h^r,E(w_1),E(w_2),\cdots,E(w_d))$ 并发送给服务器，服务器执行重加密算法对关键词索引 I 进行重加密，计算 $E'(w_1)=g^{r(s'+\delta_1)}\cdot h''=\left(h\cdot g^{\delta_1}\right)^r,\cdots,E'(w_d)=g^{r(s'+\delta_d)}\cdot h''=\left(h\cdot g^{\delta_d}\right)^r$，最终生成密文索引 $I'=(g^r,h^r,E'(w_1),E'(w_2),\cdots,E'(w_d))$。客户端在关键字加密过程中只负责一部分计算，其余部分由服务器端进行，因此，关键词加密的计算复杂度是一次指数运算。如图 3(a)所示，由实验结果可见，关键词加密时间与关键词数量呈线性关系，因此，加密关键词的时间消耗随着索引中关键词数量的增加呈线性增长。

(b)加密算法中生成陷门需要一次哈希函数运算和一次伪随机函数运算，相比之下，其余运算的时间消耗可忽略不计。这里要检索的关键词为 $w_1',w_2',\cdots,w_\omega'$，用户首先计算 $Y=(g^r)^{t''}$，然后对每个关键词 w_i' 计算 $T_i=t''+f(t,H(w_i'))+s'$，最后将陷门 $T=(T_1,T_2,\cdots,T_\omega,Y)$ 发送给服务器 S。因此，生成陷门的计算复杂度是一次哈希函数和一次伪随机函数运算。本文分别针对“文件数量不同而关键词数量相同”和“文件数量相同而关键词数量不同”的两种场景分别做了仿真实验。在图 3(b)中，所生成陷门的关键词序列包含 10 个关键词，根据实验结果可以看到关键词陷门的计算量只与生成陷门的关键词个数呈线性关系，因此，随着文件数量的增加，生成陷门的关键词个数并没有增加，时间消耗基本保持一致；在图 3(c)中，生成陷门的时间消耗随着陷门中关键词数量的增加呈线性增长。

(c)加密算法中对于关键词检索需要一次指数运算，而其它运算与之相比可以忽略不计。服务器 S 根据用户发送过来的陷门 $T=(T_1,T_2,\cdots,T_\omega,Y)$，计算

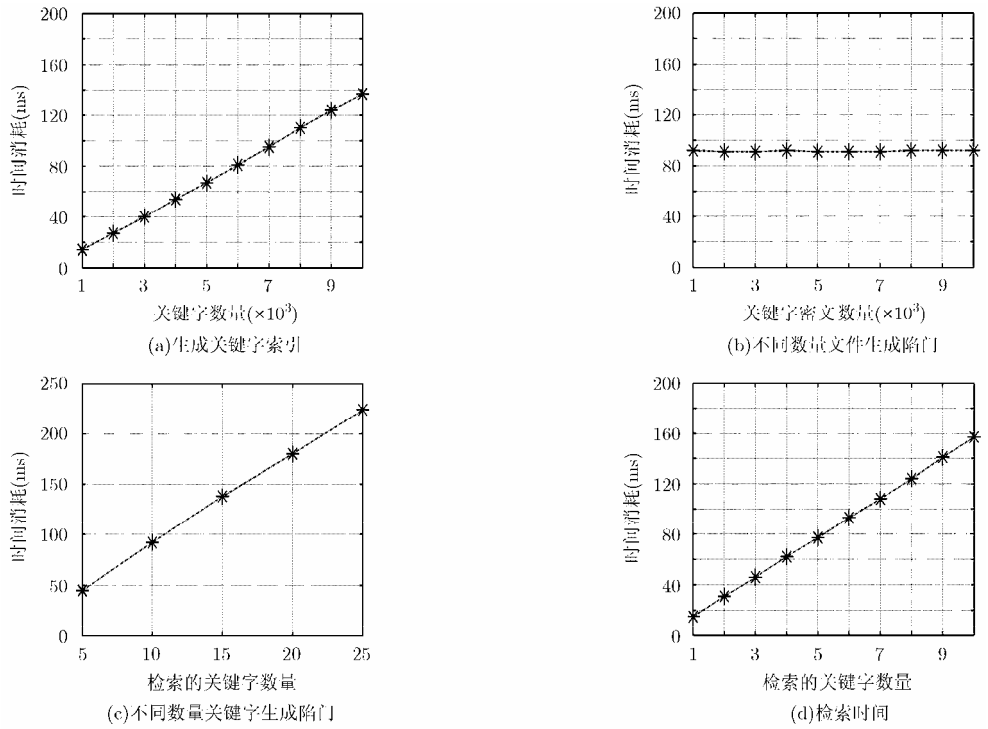

图 3 算法的时间消耗性能分析

$\left(g^{r}\right)^{T_i}\cdot h''/Y=g^{r[s+f(t,H(w_i'))]}=\left(h\cdot g^{\delta_i'}\right)^{r}$，$\delta_i'=f\left[t,H\left(w_i'\right)\right]$ 和 $H_i'=H'\left[\left(h\cdot g^{\delta_i'}\right)^{r}\right]$，并将最终生成的关键字陷门 $I''=\left(H_1',H_2',\cdots,H_\omega'\right)$ 发送给服务器 $S_1,S_2,\cdots,S_N$。因此，关键词检索的计算复杂度是一次指数运算，并且所有计算工作全部是在服务器端进行。如图 3(d)所示，由于检索需要匹配每个关键词，实验结果得出，文件的检索时间与所检索的关键词数量呈线性关系，因此，关键词检索的时间消耗随着索引中关键词数量的增加呈线性增长。

(2)性能比较： 表 3 列出了本文方案与其它经典方案的性能比较。由于 KASE[10]和 CLPEKS[11]不支持多关键词检索，因此此处仅比较生成单关键词陷门的运算量。由表 3 可知，本文方案生成单关键词陷门的效率与 KASE 相近，比其余方案均要高；由于本文方案提供多关键词的精确检索，因此检索运算量较大，但如果仅检索单个关键词，本文方案的检索量仅是指数运算，效率高于其它方案；本文方案中，对关键词加密只需要 $(m+3)$ 次指数计算，其余计算都由存储服务器执行，故本文方案的关键词加密效率要远高于其它方案；此外，其它方案均不支持多服务器和多用户模型，而本文方案可将大型数据文件分割存储在不同的云服务器，相比单服务器模型，多个服务器可并行执行密文检索任务，提高了对大型数据文件的处理能力。

(3)多服务器检索效率分析： 针对本文方案中算法 Search(·) 的多服务器并行检索，仿真实验结果如图 4 和图 5 所示。

图 4(a)分别针对单服务器模型和多服务器模型

表 3 方案的效率比较

方案	单关键词生成陷门运算量	检索运算量	关键词加密运算量	支持多用户	支持多关键词	支持多服务器
FK[14]	3 exp+ 1pr	2pr	$3m$ exp	否	是	否
RE-DPECK[13]	1 exp	2pr	$(2m+2)$ exp+ 3pr	否	是	否
CLPEKS[11]	3M	1pr	$(3m+4)$pr	否	否	否
KASE[10]	1MU	2pr	n exp+ $(2m+n)$pr	否	否	否
本文方案	1MU	2ω exp	$(m+3)$ exp	是	是	是

注：n 是文件数目；m 是文件的关键词数量；ω 是用户检索的关键词数目；M 是群 G 中的乘法运算；MU 是乘法运算；pr 是双线性运算；exp 是指数运算，这里 pr > exp。

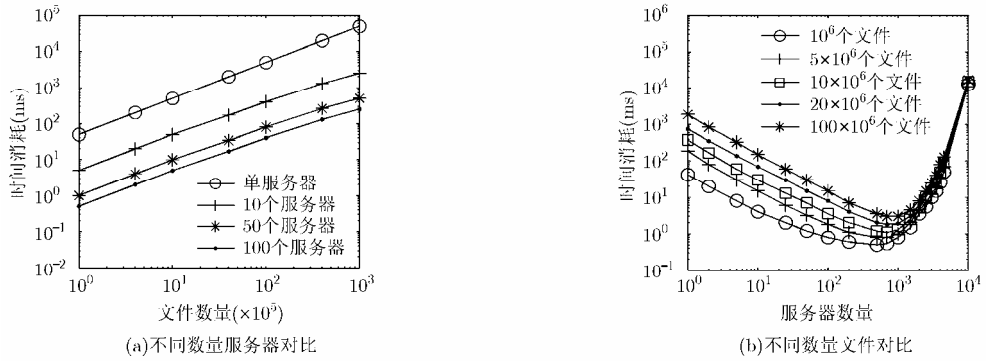

图 4 单服务器/多服务器检索效率

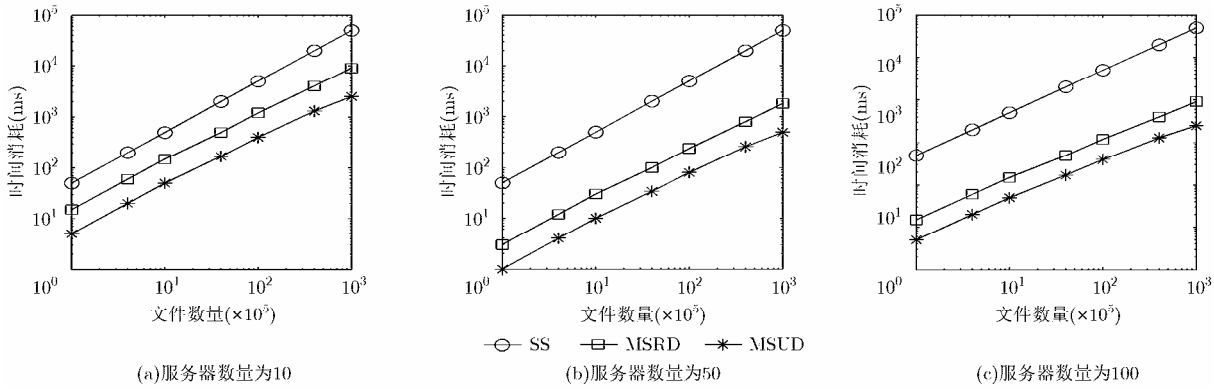

图 5 单服务器/不同分割存储方式下多服务器检索效率

进行比较, 服务器数量在图中从上往下分别为 1 个、10 个、50 个和 100 个；图 4(b)在不同数量文件和不同数量服务器情况下，针对服务器对密文文件的检索效率做了实验分析。假设文件等分存储于每个服务器上，每个文件有 20 个关键字，用户用于检索的关键字陷门有 10 个关键字。

如图 4(a)所示, 在密文文件数量相同的情况下，参与检索的服务器数量越多，则检索效率越高，尤其当文件数量较大时，检索效率有较显著提高。图 4(b)展示了 5 种不同数量文件的多服务器效率比较，检索时间首先随着参与检索的服务器数量的增加而递减，当服务器数量超过某个临界值时，所需的检索时间反而随其数量的增加而递增。可见，检索时间并非总是随着服务器数量的增加而递减，而是有一个临界值，因为服务器 S 需要将服务器 $S_1, S_2, \cdots, S_N$ 传来的检索结果进行排序，并最终选出检索结果最优的几个服务器。

因此，从上述实验可知，在确保服务器数量不超过临界值的情况下，多服务器模型在对密文文件进行检索时，其效率会随着参与检索的服务器数量的增加而有较显著的提高，尤其是在文件数量巨大的情况下，检索效率的提高更为显著。

图 5 针对算法Search(·)中的单服务器模型和两种多服务器模型(不同的文件分割存储方式 —— 随机分割和平均分割)分别做了实验分析。此处，假设每个文件有 20 个关键字, 用户用于检索的关键字陷门包含 10 个关键字。

在多服务器模型中，图 5(a)至图 5(c)中分别有 10 个、50 个和 100 个服务器同时进行并行检索(均未超过服务器数量的门限)，文件可能随机分别存储于这些服务器(MSRD)，或者等分存储于这些服务器(MSUD)，图 5 中所示均为 10 次实验的平均值。由实验结果可知，随着存储于服务器上文件数量的增加，多服务器检索比单服务器(SS)检索更有优势，且随着服务器数量的增加，检索效率也有较显著的提升；相比于单服务器模型，多服务器模型对文件检索的开销明显减少。同时，由于所有服务器并行检索，检索的最大时间由花费检索时间最长的那个服务器所决定，而服务器上存储的文件越多则所需检索的时间越多，因此，文件等分存储的方式往往比随机存储的方式拥有更高的检索效率。

5　结束语

本文提出了一种基于云存储的多服务器多关键词可搜索加密方案，可在多个服务器相互不知道检

索结果的情况下，共同完成检索任务，并将最终的检索结果返回给用户。该方案能够更精确地进行检索，并且被证明是 IND-CKA 安全的，同时也证明了关键词陷门的安全性。此外，该方案可以更灵活地控制用户对密文文件的访问权限。

然而，本文方案仍存在一些问题有待解决：例如本文假设服务器是诚实且好奇的，但在恶意服务器模型中，如何分辨服务器返回的检索结果是否真实，具有很大的挑战性。此外，一个不诚实的授权用户会导致许多安全隐患，他可能把检索返回并解密的文件发送给没有权限的其他用户，甚至将密钥信息发送给他人，从而导致数据隐私的泄露和其它的安全隐患。在未来的工作中，我们将尝试改进本文的可搜索加密方案，从而克服这些难题。

参 考 文 献

[1] SONG X D, WAGNER D, and PERRIG A. Practical techniques for searches on encrypted data[C]. IEEE Symposium on Security and Privacy, Berkeley, USA, 2000: 44-55.

[2] BONEH D and FRANKLIN M. Identity-based encryption from the weil pairing[C]. Advances in Cryptology-CRYPTO 2001-21st Annual International Cryptology Conference, California, USA, 2001: 213-229.

[3] BONEH D, CRESCENZO G, OSTROVSKY R, *et al*. Public key encryption with keyword search[C]. Proceedings of EUROCRYPT 2004, Interlaken, Switzerland, 2004: 506-522.

[4] GOH E. Secure indexes[OL]. http://crypto.stanford.edu/~eujin/papers/secureindex, 2003.

[5] KAMARA S, PAPAMANTHOU C, and ROEDER T. Dynamic searchable symmetric encryption[C]. CCS 2012 19th ACM Conference on Computer and Communications Security, Raleigh, USA, 2012: 965-976.

[6] RHEE H S, PARK J H, SUSILO W, *et al*. Improved searchable public key encryption with designated tester[C]. ASIACCS'09 Proceedings of the 4th International Symposium on Information, Computer, and Communications Security, Sydney, Australia, 2009: 376-379.

[7] HU C and LIU P. A secure searchable public key encryption scheme with a designated tester against keyword guessing attacks and its extension[C]. Communications in Computer and Information Science, Jinan, China, 2011: 131-136.

[8] ZIRTOL KOBRA AMIRI, NOROOZI MAHNAZ, and ESLAMI ZIBA. Multi-user searchable encryption scheme with general access structure[C]. 2015 2nd International Conference on Knowledge-Based Engineering and Innovation (KBEI), Tehran, Iran, 2015: 399-404.

[9] LI J, LI J, CHEN X, *et al*. Privacy-Preserving data utilization in hybrid clouds[J]. *Future Generation Computer Systems*, 2014, 30(1): 98-106. doi: 10.1016/j.future.2013.06.011.

[10] CUI B, LIU Z, and WANG L. Key-Aggregate Searchable Encryption (KASE) for group data sharing via cloud storage[J]. *IEEE Transactions on Computers*, 2015, 65(8): 2374-2385. doi: 10.1109/TC.2015.2389959.

[11] PENG Yanguo, CUI Jiangtao, PENG Changgen, *et al*. Certificateless public key encryption with keyword search[J]. *China Communications*, 2014, 11(11): 100-113. doi: 10.1109/CC.2014.7004528.

[12] GOLLE P, STADDON J, and WATERS B. Secure conjunctive keyword search over encrypted data[C]. International Conference on Applied Cryptography and Network Security, Huangshan, China, 2004: 31-45.

[13] YANG Yang, MA Maode, and LIN Bogang. Proxy re-encryption conjunctive keyword search against keyword guessing attack[C]. Computing, Communications and IT Applications Conference (ComComAp), Hong Kong, China, 2013: 125-130.

[14] KERSCHBAUM F. Secure conjunctive keyword searches for unstructured text[C]. International Conference on Network and System Security, Milan, Italia, 2011: 285-289.

[15] XIA Zhihua, WANG Xinhui, SUN Xingming, *el 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. doi: 10.1109/TPDS.2015.2401003.

黄海平： 男，1981 年生，教授，研究方向为信息网络安全和数据隐私保护技术.

杜建澎： 男，1989 年生，硕士生，研究方向为信息网络安全和数据隐私保护技术.

戴 华： 男，1982 年生，副教授，研究方向为信息网络安全和数据隐私保护技术.

王汝传： 男，1943 年生，教授，研究方向为计算机软件、信息安全和无线传感器网络技术等.