نشریه علمی "پدافند الکترونیکی و سایبری"
سال هشتم، شماره ۳، پاییز ۱۳۹۹، ص ۱۶۵-۱۵۱

علمی-پژوهشی

# طرح جستجوی کلیدواژه فازی بر روی پایگاه داده رمز شده در رایانش ابری مبتنی بر خوشه‌بندی کلمات

یحیی دهقانیان[۱]، مجید غیوری ثالث[۲*]، علیرضا رحیمی[۳]
۱- دانشجوی دکتری، ۲- استادیار، ۳- استادیار دانشگاه امام حسین (ع)

(دریافت: ۹۸/۰۸/۲۶، پذیرش: ۹۸/۱۱/۱۲)

## چکیده

برون‌سپاری پایگاه داده در رایانش ابری یکی از اصلی‌ترین راهکارها برای حفظ، نگهداری و دسترسی آسان بدون نیاز به سرمایه‌گذاری کلان، جهت تأمین زیرساخت است. معمولاً مالکان داده به سرویس دهندگان و تأمین کنندگان زیرساخت از نظر صحت نگهداری و دسترس پذیری، اطمینان دارند، ولی نگران حفظ حریم خصوصی و محرمانگی اطلاعات هستند و به همین دلیل ترجیح می‌دهند داده‌ها را به‌صورت رمز شده در سرورهای ابری نگهداری و بازیابی کنند. داده‌های رمز شده قابلیت جستجو ندارند و نیاز به راهکار، جستجو روی داده‌های رمز شده در سرور ابری است. یکی از راه‌حل‌ها، استفاده از شاخص‌دهی کلمات کلیدی در کنار پایگاه داده رمز شده است. برای استفاده از این راه‌حل‌ها چندین چالش اساسی وجود دارد که می‌توان به حجم بالای شاخص‌ها، مواجهه با خطای کاربران و سرعت جستجو اشاره کرد. در این تحقیق برای جبران خطای کاربران، از مجموعه کلیدواژه فازی به‌جای کلمات ثابت در هنگام جستجو استفاده می‌شود. همچنین برای کاهش فضای ذخیره‌سازی با استفاده از روش خوشه‌بندی کلمات کلیدی، مجموعه کلیدواژه فازی مناسب انتخاب شده و فراداده با حجم کمتر تولید و رمزگذاری می‌گردد. در اینجا با استفاده از روش‌های خوشه‌بندی سلسله مراتبی با سنجه‌های خاص، کلمات کلیدی مشابه در یک خوشه قرار گرفته و برای پیدا کردن کلیدواژه مورد نظر نیاز به جستجوی همه فراداده نیست و این روش باعث کاهش زمان جستجو می‌گردد. نتایج عملی و ارزیابی نشان می‌دهد که طرح پیشنهادی عملی، امن و کارآمد است.

**کلیدواژه‌ها:** رایانش ابری، برون‌سپاری پایگاه داده، رمزنگاری قابل جستجو، جستجوی فازی، خوشه‌بندی

## ۱- مقدمه

رایانش ابری مدلی برای دسترسی مناسب و بر اساس تقاضا به شبکه گسترده منابع محاسباتی قابل تنظیم (مانند تجهیزات شبکه‌ها، سرورها، برنامه‌ها و سرویس‌ها) است. این مدل می‌تواند به سرعت و با حداقل تلاش، منابع محاسباتی مقیاس پذیر، خدمات باکیفیت بالا و دسترسی به شبکه را فراهم کند؛ بنابراین به دلیل راحتی و انعطاف پذیری، مالکان پایگاه داده‌ها ترجیح می‌دهند که داده‌های خود را در سرورهای ابری برون‌سپاری کنند. با توجه به اینکه سرورهای ابری در فضای مورد اعتماد متفاوت با مالکان داده قرار دارند، بنابراین امنیت همواره دغدغه اصلی مالکان داده است. امنیت پایگاه داده شامل سه ویژگی اصلی، محرمانگی[۱]، تمامیت[۲] و دسترس پذیری[۳] است که معمولاً صاحبان پایگاه داده به سرویس دهندگان از لحاظ نگهداری صحیح (تمامیت) و در دسترس بودن اطمینان دارند ولی ازنظر محرمانگی اعتماد ندارند.

یکی از راهکارها برای حفظ محرمانگی پایگاه داده در رایانش ابری، ذخیره به‌صورت رمز شده است که در این صورت جستجوی اطلاعات برای کاربران با استفاده از روش‌های جستجوی سنتی، قابل انجام نیست. یک راه‌حل ساده برای این مسئله دریافت تمامی داده‌های رمز شده از سرور ابری و انجام عملیات جستجو بعد از رمزگشایی توسط کاربران می‌باشد. این روش به پهنای باند، فضای ذخیره‌سازی و سربار محاسباتی زیادی نیاز دارد و در عمل قابل‌اجرا نیست.

راه‌حل دیگر اینکه کلیدهای رمزگشایی و جستجو در اختیار سرور ابری قرار گیرد و سرور ابری عملیات رمزگشایی داده‌ها و

* رایانامه نویسنده مسئول: Ghayoori@ihu.ac.ir

[1] Confidentiality
[2] Integrity
[3] Availability

جستجو بر روی متن اصلی را انجام دهد. بدیهی است که در این روش سرور ابری غیر قابل اعتماد به اصل داده‌ها دسترسی خواهد داشت و محرمانگی اطلاعات که هدف رمزگذاری بوده، به مخاطره خواهد افتاد. برای حل این مسئله، طرح‌های مختلف رمزنگاری قابل جستجوی (SE[1]) ارائه شده است. این طرح‌ها به کاربران اجازه می‌دهد تا به‌طور انتخابی اطلاعات رمز شده روی سرور ابری را با جستجوی کلمه کلیدی مورد نظر بازیابی کند. روش‌های رمزنگاری قابل جستجو در کل بر سه مبنا استوار است؛ مبنای اول جستجو روی خود داده‌های رمز شده است که می‌توان به طرح‌هایی مانند روش سانگ و همکاران [۱]، رمزنگاری هم‌ریخت [۲] و رمزنگاری مبتنی بر حفظ ترتیب[2] [۳] اشاره کرد. روش دومی بر مبنای تسهیم راز[3] [۴] جستجو می‌کند. درنهایت سومی عملیات جستجو را با ساختن دریچه‌های مرتبط با داده‌های اصلی انجام می‌دهد [۵ و ۶]. با توجه به ملاحظات رایانش ابری از نظر حجم پردازش، پهنای باند، نوع کاربرد و نوع داده، روش سوم برای برون‌سپاری امن پایگاه داده مناسب‌تر می‌باشد و طرح‌های بیشتری در این زمینه ارائه شده است. در این روش در زمان برون‌سپاری، کلمات کلیدی استخراج شده و بعد از شاخص‌دهی به همراه داده‌های اصلی به‌صورت رمز شده برون‌سپاری می‌شوند. این داده‌های رمز شده از این پس به‌عنوان فراداده[4] در عملیات جستجو و بازیابی مورد استفاده قرار می‌گیرد. بدین صورت که در هنگام بازیابی، ابتدا جستجو در داخل فراداده انجام گرفته و بعد از انطباق الگوی جستجو مطابق شاخص‌دهی، اطلاعات رمز شده استخراج و به کاربر ارسال می‌شود. در اینجا هر چه تعداد کلمات کلیدی از پیش تعریف شده بیشتر باشد، احتمال موفقیت پرس‌وجو بالا می‌رود؛ زیرا شاخص‌ها به‌صورت رمز شده ذخیره می‌شوند و کاربر می‌بایست کلیدواژه را عیناً وارد کند و در صورت خطا، نتیجه پرس‌وجو موفق نخواهد بود. از طرفی پرس‌وجوی تمامی حالت‌های یک کلمه برای جبران خطای کاربر، سرعت پرس‌وجو را به شدت کاهش می‌دهد. یک روش بهبود و افزایش سرعت، جستجوی بر مبنای شباهت[5] کلمات و یا جستجوی فازی[6] می‌باشد. در این تحقیق به دلیل نوع داده (کلمات اصلی پایگاه داده)، پرس‌وجو و بازیابی بر اساس شباهت ظاهری کلمات انجام می‌گیرد، بنابراین ارتباط کلمات از نظر ریشه‌ای، معنایی و مفهومی کمکی به جستجو نمی‌کند، بنابراین از روش جستجوی فازی بر مبنای فاصله ویرایش[7] خطای ورودی کاربران را جبران می‌شود [۷]. در این روش هم نیاز است که مجموعه کلیدواژه فازی کلمات، به‌صورت رمز شده برون‌سپاری شود که در مقیاس بالا سبب افزایش حجم فراداده ذخیره شده و کاهش سرعت جستجو می‌گردد. در این تحقیق برای مقابله با این مشکل از روش خوشه‌بندی کلمات کلیدی استفاده می‌شود. بدین ترتیب که به‌جای ذخیره مجموعه کلیدواژه فازی تمامی کلمات، ابتدا آن‌ها را خوشه‌بندی کرده و سپس مجموعه کلیدواژه فازی مراکز خوشه‌ها و شاخص‌ها را به‌صورت رمز شده برون‌سپاری می‌شود.

در ادامه مقاله، در بخش دوم کارهای پیشین مرتبط با این تحقیق را بیان می‌گردد. برخی مفاهیم پایه و مقدمات پیش نیاز را در بخش سوم این تحقیق بررسی خواهد شد. در بخش چهارم، توضیح دقیق الگوریتم پیشنهادی و مراحل اجرایی آن به‌طور کامل ارائه خواهد شد. در بخش پنجم، تحلیل امنیت طرح پیشنهادی را بررسی می‌شود. در ادامه در بخش ششم ضمن مقایسه با سایر طرح‌ها، نتایج شبیه‌سازی مربوط به الگوریتم پیشنهادی مورد بررسی قرار گرفته و درنهایت در بخش هفتم نتیجه و جمع‌بندی این تحقیق ارائه خواهد شد.

## ۲- کارهای پیشین

### ۲-۱- رمزنگاری قابل جستجو

در زمینه رمزنگاری قابل جستجو مطالعات زیادی انجام گرفته است، اغلب طرح‌ها بر روی بهبود کارایی، فرموله سازی و امنیت تمرکز دارند که به برخی از آن‌ها اشاره می‌شود. سانگ و همکاران [۱] برای اولین بار در سال ۲۰۰۰، طرح رمزگذاری متقارن قابل جستجو را ارائه کردند. در این طرح، یک ساختار رمزنگاری دولایه ویژه برای رمزگذاری هر کلمه کلیدی ساخته می‌شود و در فاز جستجو سرور ابری تمام اسناد را به‌صورت ترتیبی پویش می‌کند. بدین ترتیب زمان جستجو با اندازه مجموعه اسناد رابطه خطی دارد. در سال ۲۰۰۴، طرح [۸] استفاده از فیلترهای بلوم و توابع شبه تصادفی برای ساخت شاخص‌های امن ارائه کرد. ۳ سال بعد کورت‌مولا و همکارانش یک طرح امن و کارآمد، ارائه کردند. [۹] این طرح از چند کاربر برای ارسال درخواست جستجو پشتیبانی می‌کند و هزینه جستجو متناسب با تعداد اسناد حاوی کلمات کلیدی مورد نظر است. در سال ۲۰۱۱ کاو و همکاران [۱۰] برای اولین بار، یک طرح جستجوی چند کلمه‌ای با حفظ حریم خصوصی بر روی داده‌های ابری رمزگذاری شده ارائه کردند. در این طرح از تطبیق مختصات برای اندازه‌گیری شباهت بین اسناد با کلمات کلیدی مورد پرسش، استفاده می‌شود. شیا و همکاران [۱۱] در سال ۲۰۱۶، یک طرح جستجوی امن و پویا شامل چندین کلمه کلیدی روی محیط ابری، ارائه کردند. این طرح از

[1] Searchable Encryption
[2] Order Preserving Encryption Scheme
[3] Secret Sharing
[4] Metadata
[5] Like
[6] Fuzzy
[7] Distance Edit

یک شاخص مبتنی بر درخت که خود از ترکیب مدل فضای برداری (VSM[1]) و فرکانس اصطلاح و معکوس فرکانس سند ($TF\ x\ IDF$[2]) به‌دست آمده بهره می‌برد. فو و همکاران [۱۲] یک مدل با کاربری آسان که با تجزیه و تحلیل تاریخچه جستجو برای بازیابی اسناد مربوطه از سرور ابری بر اساس علاقه‌مندی مالکان داده عمل می‌کرد را ارائه دادند. البته تمامی روش‌های فوق فقط جستجوی دقیق کلیدواژه را پشتیبانی می‌کنند و مناسب رایانش ابری نیستند.

## ۲-۲- طرح جستجو کلیدواژه فازی

طرح‌های جستجوی با کلیدواژه دقیق نمی‌توانند نتیجه مورد انتظار را در حالتی که کلمه ورودی اشتباه تایپی دارد، بازگردانند و این موضوع کارایی سامانه را به شدت تحت تأثیر قرار می‌دهد. در اینجا هدف از جستجو، قابلیت پیدا کردن کلمات مشابه از نظر املایی می‌باشد. برای حل این مشکل، لی و همکاران [۷] طرح جستجوی کلیدواژه فازی را بر روی داده‌های ابری رمزنگاری شده، با استفاده از فاصله ویرایش برای اندازه‌گیری شباهت دو کلمه، ارائه کردند. این طرح از روش "نشانه عام[3]" برای ساخت مجموعه‌های کلیدواژه فازی استفاده می‌کند. لیو و همکاران [۱۳] با استفاده از روش مبتنی بر لغت‌نامه‌ای[4] طرح فوق را از نظر اندازه شاخص بهبود بخشیدند. کوزو و همکاران [۱۴] از توابع LSH[5] برای تولید شاخص فایل جهت جستجوی سریع شباهت استفاده کردند. چوا و همکاران [۱۵] طرح [۷] را با معرفی یک نمایه ساختار درخت بهبود یافته و با جستجوی عبارات از پیش تعریف شده، به‌عنوان یک کلمه کلیدی، قابلیت جستجو را بهبود داد. وانگ و همکاران [۱۶] طرح جستجوی کلیدواژه فازی چند کلمه‌ای را بر اساس فیلتر بلوم و تابع LSH پیشنهاد کردند که در آن کلمات کلیدی به وسیله مجموعه دو-گرم[6] انتخاب می‌شوند.

## ۲-۳- طرح جستجو کلیدواژه فازی کارآمد

روش‌های جستجوی رمز شده، تک کلمه‌ای و چند کلمه‌ای فازی در فضای ابری به نتایج قابل قبولی دست یافته است، اما کارایی جستجو، هزینه ذخیره‌سازی، دقت و سازگاری این روش‌ها همچنان به تحقیقات بیشتری نیاز دارد و طرح‌های مختلفی برای بهبود هر کدام از پارامترهای فوق ارائه شده است. لی و همکاران [۱۷] با استفاده از جستجو درخت پیشوندی مبتنی بر نماد، سرعت جستجوی طرح خودشان را بهبود دادند. وانگ و همکاران [۱۸] قابلیت تصدیق[7] نتایج جستجو را به طرح فوق اضافه کردند. فو و همکاران [۱۹] دقت طرح جستجوی چند کلمه‌ای رجبندی شده [۱۶] را بهبود دادند. ماهاجان و همکاران [۲۰] از روش خوشه‌بندی سلسله مرتبی[8] و حداکثر انتظار (EM[9]) برای ذخیره و جستجو در سروری ابری استفاده کردند. مرجع [۲۱] از روش تخصیص یک بردار به کل مجموعه کلیدواژه فازی برای کاهش حجم محاسبه و فضای ذخیره‌سازی استفاده کرده است. ایکسایی و همکاران [۲۲] با استفاده از الگوریتم خوشه‌بندی سلسله مراتبی دقت جستجو را بهبود دادند و روش k-نزدیک‌ترین همسایه را برای جستجوی کلمات مشابه انتخاب کرده‌اند. فنگ و همکاران [۲۳] طرح جستجوی فازی چند کلمه‌ای خاص زبان چینی ارائه کردند.

## ۳- مفاهیم پایه

در اکثر طرح‌های جستجوی فازی بر روی داده‌های رمز شده در محیط رایانش ابری، حجم فراداده و سرعت جستجو به‌عنوان چالش اصلی مطرح است. یکی از روش‌ها برای کاهش حجم فراداده، خوشه‌بندی کلمات کلیدی است. در اصل خوشه‌بندی به فرایندی اطلاق می‌شود که در آن مجموعه‌ای از اشیاء به چندین دسته یا خوشه گروه‌بندی می‌شوند، به ترتیبی که اشیاء درون یک خوشه تا حد ممکن شبیه به یکدیگر و اشیاء خوشه‌های مختلف، با هم متفاوت باشند. تشابه و عدم تشابه میان اشیاء بر اساس مقادیر صفات خاصه[10] آن‌ها تعیین می‌شود و اغلب برای محاسبه آن، از سنجه فاصله استفاده می‌شود.

## ۳-۱- خوشه‌بندی سلسله مراتبی

به‌طور کلی روش‌های پایه خوشه‌بندی شامل روش‌های افراز[11]، روش‌های سلسله مراتبی[12]، روش‌های مبتنی بر چگالی[13] و روش‌های مبتنی بر توری[14] هست [۲۴]. در روش خوشه‌بندی سلسله مراتبی یک درخت سلسله مراتبی از مجموعه اشیاء ایجاد می‌شود و به دو دسته تجمیعی[15] و تقسیمی[16] گروه‌بندی می‌شود.

[1] Vector Space Model
[2] Term Frequency-Inverse Document Frequency
[3] Wildcard
[4] Dictionary-based
[5] Locality Sensitive Hashing
[6] Bi-Gram
[7] Verifiable
[8] Hierarchical Clustering
[9] Expectation Maximization
[10] Special Attributes
[11] Partitioning Methods
[12] Hierarchical Methods
[13] Density Based Methods
[14] Grid Based Methods
[15] Agglomerative
[16] Divisive

در نوع تجمیعی (AGNES[1]) که با نام رویکرد پایین به بالا نیز شناخته می‌شود، ابتدا هر شیء در یک گروه قرار می‌گیرد و در ادامه گروه‌هایی که به یکدیگر شباهت دارند، ادغام می‌شوند و این فرایند تا زمانی که کلیه اشیاء در یک گروه قرار بگیرند ادامه می‌یابد. البته می‌توان با کمک شرط پایانی نیز الگوریتم را متوقف کرد. در نوع تقسیمی (DIANA[2]) که نام رویکرد بالا به پایین نیز شناخته می‌شود، ابتدا کلیه اشیاء در یک خوشه قرار می‌گیرند. در هر بار تکرار، یک خوشه به خوشه‌های کوچک‌تر شکسته می‌شود. این کار تا زمانی که هر شیء به تنهایی در یک خوشه قرار بگیرد و یا یک شرط پایانی رخ بدهد، ادامه پیدا می‌کند. البته برای بهبود خوشه‌بندی سلسله مراتبی می‌توان از خوشه‌بندی چند مرحله‌ای نیز استفاده کرد که در آن از روش‌های خوشه‌بندی سلسله مراتبی به همراه روش‌های دیگر خوشه‌بندی به‌صورت یکپارچه استفاده می‌شود. از معروف‌ترین روش‌ها می‌توان به روش‌های BIRCH و Chameleon اشاره کرد. الگوریتم BIRCH در ابتدا با کمک ساختارهای درختی، اشیاء را به‌صورت سلسله مراتبی افراز می‌کند. در این ساختار درختی گره‌های برگ یا گره‌های غیر برگ سطوح پایین‌تر، بر اساس مقیاس دقت، به‌عنوان زیرخوشه‌ها در نظر گرفته می‌شوند. سپس در ادامه با کمک الگوریتم‌های دیگر خوشه‌بندی این زیرخوشه‌ها انجام می‌شود. در الگوریتم Chameleon به کمک مدل‌سازی پویا شباهت میان زوج خوشه‌ها تعیین می‌شود. در واقع به‌طور دقیق‌تر می‌توان گفت در این الگوریتم شباهت خوشه بر اساس چگونگی اتصال اشیاء درون خوشه و مجاورت خوشه‌ها ارزیابی می‌شود [۲۴].

در یک دسته‌بندی دیگر می‌توان روش‌های خوشه‌بندی سلسله مراتبی را به سه گروه: روش‌های الگوریتمی، روش‌های احتمالی و روش‌های بیزی طبقه‌بندی نمود. گروه روش‌های الگوریتمی را می‌توان شامل روش‌های تجمیعی، تقسیمی و چند مرحله‌ای در نظر گرفت و این بدین معنی است که این روش‌ها خوشه‌ها را طبق فواصل قطعی و معین میان اشیاء به‌دست می‌آورند. روش‌های احتمالاتی از مدل‌های احتمالاتی برای یافتن خوشه‌ها استفاده و کیفیت خوشه‌ها را با کمک برازندگی[3] مدل‌ها اندازه‌گیری می‌کنند. در روش‌های بیزی، یک توزیع از خوشه‌بندی ممکن محاسبه می‌شود و به‌جای نشان دادن یک خوشه‌بندی قطعی در خروجی، گروهی از ساختارهای خوشه‌بندی همراه با احتمالات آن‌ها نشان داده می‌شود [۲۴].

خوشه‌بندی کلمات کاربرد زیادی در پردازش اسناد و متون وب، مانند بازیابی سریع اطلاعات، سازمان‌دهی بدون ناظر و افزایش کارایی در موتورهای جستجو دارد. بسیاری از الگوریتم‌های موجود در خوشه‌بندی از مدل فضای برداری برای نمایش استفاده می‌کنند. یک چالش مهم در استفاده از این مدل‌ها، زیاد بودن تعداد ویژگی‌ها (بزرگ بودن ابعاد داده) است که بر کارایی الگوریتم اثرگذار است. یک راه‌حل برای این مشکل که اغلب به نتایج بهتری منجر می‌شود، استفاده از روش‌های انتخاب ویژگی به‌منظور انتخاب زیرمجموعه‌ای از ویژگی‌ها به نمایندگی از کل ویژگی‌ها است؛ به عبارت دیگر باید بتوان ویژگی‌هایی را انتخاب کرد که حاوی بیشترین اطلاعات در مجموعه داده باشند. چانگ و همکاران [۲۵] برای بالا بردن سرعت خوشه‌بندی در بازیابی اسناد، ابتدا کلمات کلیدی را شناسایی و استخراج کرده و سپس محتوی اسناد با محوریت کلمات کلیدی خوشه‌بندی کردند. جانگیر و همکاران [۲۶] جستجو در اسناد را بر اساس خوشه‌بندی کلمات نسبت به حالت معمولی مقایسه کردند و نتیجه گرفتند که حالت خوشه‌بندی نیاز به فضای حافظه بیشتری دارد و نیاز است که الگوریتم‌ها بهینه شوند. هاندا و همکاران [۲۷] طرح جستجوی چند کلیدواژه بر روی داده‌های رمز شده در ابر را ارائه کردند که تعداد مقایسه‌ها را کاهش داده و زمان جستجو را در مقایسه با فن‌های موجود کاهش می‌دهد. این طرح، اسناد را بر اساس رابطه بین کلمات کلیدی خوشه‌بندی کرده و روش جستجو شامل جستجوی اسناد درون خوشه مربوطه در مقابل جستجوی کل مجموعه اسناد می‌باشد. سامانتری و همکاران [۲۸] یک طرح جستجوی چند کلمه‌ای کارآمد در متن رمز شده، برون‌سپاری شده در ابر با استفاده از یک ساختار داده فهرست شده مبتنی بر درخت ارائه کردند. در طرح فوق، مجموعه اسناد با استفاده از روش سلسله مراتبی k-Means خوشه‌بندی شده و از مدل فضای برداری برای ایجاد یک شاخص رمز شده و بردارهای پرس‌وجو استفاده شده است.

### ۳-۲- معیار تشابه

اندازه‌گیری شباهت به‌عنوان محاسبه فاصله بین نقاط مختلف داده تعریف می‌شود. عملکرد بسیاری از الگوریتم‌ها بستگی به انتخاب عملکرد مناسب فاصله نسبت به مجموعه داده‌های ورودی دارد. در واقع، شباهت مقداری است که قدرت رابطه بین دو مورد داده را بیان می‌کند. در اینجا، به برخی از توابع اندازه‌گیری شباهت مرتبط اشاره می‌شود:

**۱- فاصله اقلیدسی**[4]: فاصله اقلیدسی ریشه اختلاف مربع بین مختصات یک جفت از اشیاء را تعیین می‌کند. برای بردارهای x و y فاصله $d(x, y)$ توسط رابطه (۱) محاسبه می‌شود.

[1] Agglomerative Nesting
[2] Divisive Analysis
[3] Fitness
[4] Euclidean distance

$$d = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (۱)$$

در اینجا x و y بردارهای n-بعدی هستند [۲۹].

**۲- فاصله کوساین**[1]: فاصله کوساین تعیین کننده کسینوس زاویه بین دو بردار است که توسط رابطه (۲) محاسبه می‌شود.

$$d = cos\theta = \frac{(x_i.x_j)}{(|x_i| \times |x_j|)} \quad (۲)$$

در اینجا $\theta$ زاویه بین دو بردار $x_i$ و $x_j$ بردارهای n-بعدی هستند [۲۹].

**۳- فاصله جاکارد**[2]: فاصله جاکارد برای اندازه‌گیری شباهت از رابطه (۳) استفاده می‌کند [۲۹].

$$d = cos\theta = \frac{(x_i.x_j)}{(|x_i|^2 + |x_j|^2 + x_i.x_j)} \quad (۳)$$

**۴- فاصله ویرایش**

اگر دو مجموعه رشته‌ای[3] داشته باشید، می‌توان با یک تشابه پیوسته، همه جفت رشته‌های مشابه هر دو مجموعه را پیدا کرد. از فاصله ویرایش برای کمی سازی شباهت بین دو رشته استفاده می‌شود. به‌طور کلی، فاصله ویرایش بین دو رشته $x$ و $y$ را با $Ed(x,y)$ نشان داده و شامل حداقل تعداد عملیات ویرایش کاراکتر (درج، حذف و جانشینی) است که برای تبدیل $x$ به $y$ مورد نیاز است. برای مثال، $Ed(algorithm, algortm) =$ 2 می‌باشد. در اینجا دو رشته مشابه هستند اگر فاصله ویرایش آن‌ها، از یک آستانه تعریف شده بزرگ‌تر نباشد. در بیشتر موارد برای بررسی شباهت دو رشته، لازم نیست فاصله ویرایش بین دو رشته را تا انتها محاسبه شود. بلکه می‌توان به محض رسیدن به آستانه تعریف شده محاسبه را خاتمه دهید. با این روش حجم محاسبات کاهش پیدا می‌کند [۳۰].

## ۳-۳- خوشه‌بندی بر اساس فاصله ویرایش

اکثر تحقیقات انجام گرفته در زمینه خوشه‌بندی برای برون‌سپاری و بازیابی اسناد بر اساس تک کلمه‌ای، چند کلمه‌ای، کلمات هم‌خانواده و یا کلمات مرتبط در اسناد می‌باشد. در این تحقیق هدف برون‌سپاری و جستجو بر روی پایگاه داده رمز شده است، بنابراین فرض می‌شود که فیلدهای پایگاه داده برون‌سپاری شده دارای کلمات اصلی تک کلمه‌ای بوده و خوشه‌بندی بر اساس شباهت ظاهری (املایی) کلمات اصلی انجام می‌گیرد و معنای کلمات و یا هم‌ریشه بودن در نظر گرفته نمی‌شود؛ بنابراین تشابه کلمات بر مبنای فاصله ویرایش یا سنجه فاصله کلمات محاسبه می‌شود و تشابه ازنظر معنایی کمکی برای جستجو نمی‌کند.

به‌عنوان مثال اگر کلمه $y =$ "algorithm" به‌عنوان مرکز خوشه و کلمات $z =$ "algoithm" و $x =$ "algothm" را به‌عنوان کلمات کلیدی در نظر گرفته شود و $Ed(x,y)$ را به‌عنوان فاصله ویرایش دو کلمه محاسبه شود (اختلاف بین کلمات از نظر کاراکتری)، آنگاه دارید:

$$Ed(x,y) = 2 \ , Ed(y,z) = 1 \quad (۴)$$

در این معیار تشابه مطابق رابطه (۴) اگر فاصله ویرایش را عدد یک در نظر بگیرید، کلمه z در خوشه y قرار می‌گیرد و اگر فاصله ویرایش را عدد دو در نظر بگیرید، هر دو کلمه x و z در خوشه y قرار می‌گیرند.

تغییرات کلمات کلیدی شامل جایگزینی، حذف و یا اضافه شدن کاراکترها می‌باشد. در این روش هر چقدر سنجه فاصله بزرگ‌تر در نظر گرفته شود، تعداد اعضای خوشه‌ها بیشتر می‌شود و در جستجوی یک کلمه، کلمات مشابه بیشتری نمایش داده می‌شود. تعداد نتایج جستجوی که در اختیار کاربر قرار می‌گیرد در میزان رضایت کاربر تأثیر به سزایی دارد؛ یعنی اگر نتایج جستجو از یک حدی بیشتر باشد ممکن است خوشایند کاربر نباشد. در مقابل اگر در انتخاب سنجه فاصله محدودیت بیشتری اعمال شود تعداد اعضای خوشه‌ها کمتر و در نتیجه تعداد کلمات مشابه کمتری در اختیار کاربر قرار می‌گیرد؛ بنابراین پارامترها باید با توجه به نوع کلمات، به‌صورت مناسب انتخاب شود. البته ذکر این نکته ضروری است که در یک جستجوی موفق، تمامی اعضای یک خوشه به‌عنوان نتیجه جستجو اعلام می‌شود، اما به کمک تلفیق نتایج فیلدهای مختلف بعد از جستجو، نتیجه نهایی منطبق بر درخواست کاربر می‌شود.

## ۴- طرح پیشنهادی

طرح پیشنهادی می‌بایست الزامات زیر را برآورده کند:

- **جستجوی کلیدواژه فازی**. این طرح باید قادر به بازیابی همه رکوردهای حاوی کلمه کلیدی مورد جستجو با جبران خطای املایی با آستانه تعریف شده باشد.
- **کارایی**. این طرح باید به‌طور مؤثر نتیجه جستجو را بازگرداند و سربار کمتری از نظر حجم محاسبات و اندازه حافظه برای ذخیره دریچه‌های جستجو را حاصل نماید.
- **حفظ حریم خصوصی**. این طرح نباید هیچ اطلاعات داده‌ای را به سرور ابری مانند متن آشکار کلمات کلیدی و رکوردها خارج از نشت محدود تعریف شده، فاش کند.

[1] Cosine Distance
[2] Jaccard Distance
[3] String

معمولاً، نشت محدود از سه بعد زیر را در نظر گرفته می‌شود:

۱. حریم خصوصی پایگاه داده. نباید محتوی پایگاه داده را به جز شماره رکورد فاش شود.

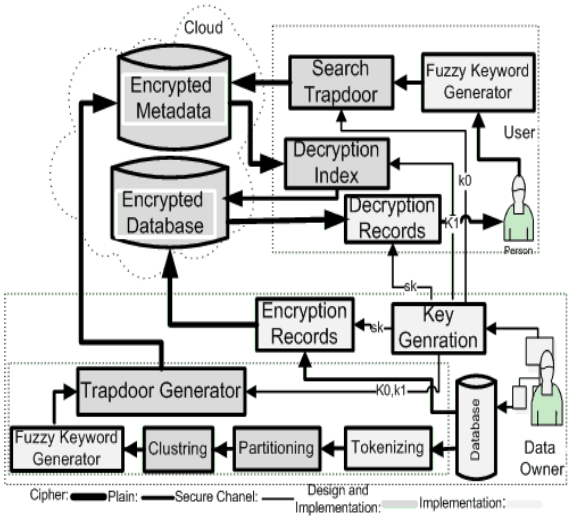
۲. حریم خصوصی شاخص. سرور ابر نتواند ارتباط بین کلمات کلیدی و رکوردها را از روی شاخص امن به دست آورد.

۳. حریم خصوصی دریچه. نباید ارتباط بین دریچه و کلمه کلیدی مورد جستجو فاش شود.

همان‌طوری که در شکل (۱) نشان داده شده است، طرح پیشنهادی شامل سه ماهیت، مالک داده، کاربر داده و سرور ابری است. در مرحله برون‌سپاری، ابتدا مالک داده، پایگاه داده آشکار $P = \{R_j, 1 \le j \le N\}, R_j = \{F_i, 1 \le i \le M\}$ را به یک پایگاه داده رمز شده $Ep = \{E_{R_j}, 1 \le j \le N\}$ تبدیل می‌کند که در این نام‌گذاری $P$ به‌کل پایگاه داده، $R_j$ رکورد شماره j، $N$ تعداد رکورد، $F_i$ مشخصه[1] شماره i و $M$ تعداد مشخصه (تعداد ستون‌ها) پایگاه داده می‌باشد. در هنگام ذخیره‌سازی، مالک داده دریچه جستجو و شاخص امن را که شامل شماره رکوردهای اعضای هر خوشه است را برای تمامی رکوردها استخراج می‌کند. این شاخص‌ها پس از رمزگذاری به همراه پایگاه داده رمز شده، در سرور ابری برون‌سپاری می‌شوند. هنگام جستجو سازوکار بدین صورت است که کاربر دریچه مربوط به کلمه کلیدی مورد نظر را محاسبه و به سرور ابری می‌فرستد، سرور ابری پس از دریافت دریچه از کاربر، عملیات جستجو را روی دریچه‌های جستجو ذخیره شده انجام داده و در صورت انطباق، شاخص‌های امن مربوطه را به کاربر برمی‌گرداند. کاربر پس از رمزگشایی شاخص‌ها، شماره رکوردهای مورد نظر را از سرور درخواست می‌کند. سرور ابری محتوای رکوردها را به‌صورت رمز شده به کاربر ارسال و کاربر پس از رمزگشایی، نتیجه جستجو را مورد استفاده قرار می‌دهد. مراحل اجرایی، نوع اطلاعات تبادل شده (رمز یا آشکار) و بخش‌های طراحی و پیاده‌سازی شده، طرح پیشنهادی در شکل (۱) مشخص می‌باشد. مطابق معماری ارائه شده فرایند انجام کار در دو بخش مالک داده و تعامل با سرور (کاربران)، به‌صورت شکل (۱) می‌باشد.

در ادامه به بررسی هر یک از اجزای این معماری پرداخته می‌شود:

[1] Field




**شکل (۱):** معماری پیشنهادی.

### ۴-۱- مالک داده

مالک داده به فرد یا سازمانی اطلاق می‌شود که در ابتدا پایگاه داده رمز شده و شاخص‌ها را به‌صورت امن برون‌سپاری می‌کند. توضیح دقیق ریز عملیات‌هایی که باید برای برون‌سپاری انجام شود، به شرح زیر است:

#### ۴-۱-۱- تولید کلید

در این مرحله مالک داده الگوریتم تولید کلید[2] احتمالی $K \leftarrow Gen(1^k)$ را فراخوانی می‌کند. به‌عنوان مثال می‌توان از توابع تولید کلید شبه تصادفی با بذر[3] تصادفی استفاده کرد که خروجی آن غیر قابل تمیز[4] از یک مقدار تصادفی واقعی باشد. خروجی الگوریتم مجموعه کلید مخفی $K = (sk, k_0, k_1)$ است که $sk$ کلید رمزنگاری محتوای رکوردها، $k_0$ کلید رمزنگاری دریچه‌های پرس‌وجو و $k_1$ کلید رمزنگاری شاخص‌ها است. فرض می‌شود کلیدهای تولید شده از طریق یک کانال امن توزیع و در اختیار کاربران مجاز قرار می‌گیرد. همچنین با توجه به متقارن بودن الگوریتم‌های رمزنگاری، کلیدهای توزیع شده قابل استفاده در هر دو مرحله رمزگذاری و رمزگشایی می‌باشد.

#### ۴-۱-۲- رمزگذاری رکوردهای پایگاه داده[5]

در این بخش تمامی رکوردهای پایگاه داده با استفاده از تابع $E_{R_j} \leftarrow f_{sk}(R_j)\,(1 \le j \le N)$ رمزگذاری شده و برون‌سپاری می‌شود. در این رابطه، $N$ تعداد رکوردهای پایگاه داده و $f$ تابع

[2] Key Generation
[3] Seed
[4] indistinguishable
[5] Encryption Records

رمزنگاری است. به‌عنوان مثال برای تابع رمزنگاری می‌توان از الگوریتم رمزنگاری بلوکی متقارن AES[1] استفاده کرد. برای جلوگیری از یکسان بودن خروجی رمز شده رکوردها با محتویات یکسان، از مد عملکردی CBC[2] الگوریتم رمزنگاری مطابق رابطه (۵) استفاده می‌شود.

محتویات اصلی هر رکورد به i بلوک تقسیم شده و برای شروع از یک بردار IV برای ترکیب با قالب متن اصلی استفاده می‌شود.

$$C_0 = IV = H(j), C_i = E_{sk}(C_{i-1} \oplus P_i), 1 \le i \le l \qquad (۵)$$
$$f_{sk}(R_j) = C_1 || \dots || C_i \,, (1 \le \mathrm{i} \le \mathrm{l})$$

در اینجا H یک تابع درهم‌ساز خالی از تصادم، $P_i$ بلوک‌های محتوی رکورد، $l$ تعداد بلوک‌ها و $j$ شمار رکورد می‌باشد.

### ۴-۱-۳- ساخت شاخص و دریچه پرس‌وجوی

این مرحله یکی از اساسی‌ترین مراحل برون‌سپاری پایگاه داده است که خود شامل ۵ گام می‌باشد:

**الف) برچسب‌گذاری[3] و استخراج کلمات کلیدی**: صاحب داده تمامی کلمات کلیدی برای هر فیلد (ستون) پایگاه داده را استخراج و مجموعه کلمه کلیدی $W$ با تعداد $n$ را مطابق جدول (۱) تشکیل می‌دهد. فرایند عملیات برای سایر فیلدها مشابه می‌باشد، بنابراین در اینجا جریان کار برای یک فیلد ارائه می‌شود. برای استخراج کلمات کلیدی متمایز و انجام عملیات پیش‌پردازش مانند حذف ایست واژه‌ها[4]، ابزارهای مختلفی وجود دارد. به‌عنوان مثال می‌توان از ابزار جی‌هزم[5] استفاده کرد. مالک داده برای هر کلمه کلیدی $w_i \in W, 1 \le i \le n\}$ استخراج شده، بردار شاخص $v(w_i)$ را ایجاد می‌کند که مجموعه‌ای از رکوردهای حاوی کلمه کلیدی $w_i$ است. بدین ترتیب اگر کلمه $w_i$ در رکورد شماره j باشد $v(w_i)[j] = 1$ و در غیر این صورت $v(w_i)[j] = 0$. در حقیقت هدف این بخش تشکیل ماتریس ارتباط[6] برای تمامی کلمات کلیدی رکوردها است. قابل توجه است که ماتریس تشکیل شده برای داده‌های بزرگ، دارای حجم بسیار بالایی است که از نظر تخصیص حافظه برای انجام محاسبات، نیاز به راه‌حل‌های پیاده‌سازی خاصی دارد.

**ب) دسته‌بندی کلمات**[7]: در این مرحله کلمات کلیدی استخراج شده با توجه به تعداد کاراکتر در دسته‌های مجزا قرار می‌گیرد. فرض می‌شود $l$ حداکثر طول کلمات کلیدی باشد، خروجی این مرحله مطابق جدول (۲) تشکیل $l$ دسته کلمات کلیدی به همراه شاخص‌های مربوط به رکوردها است. در اینجا طول کلمات کلیدی هر دسته، یکسان ولی تعداد اعضای آن‌ها متفاوت است، یعنی کلمات کلیدی $\{w_i \in W, 1 \le i \le n\}$ در $l$ دسته قرار می‌گیرند. در اینجا دارید:

$S_j = \{D_{r,j} \in w_i \,, 1 \le r \le, m_l, 1 \le j \le l, 1 \le i \le, n\}$ که $n = \sum_{k=1}^{l} m_k$ می‌باشد.

در حقیقت در این بخش ماتریس ارتباط را به چندین زیرماتریس که دارای اعضاء با طول یکسان هستند تجزیه می‌شود.

**جدول (۱):** شاخص گذاری کلمات کلیدی (ماتریس ارتباط)

| $W$ \ $R$ | $R_1$ | $R_2$ | $R_3$ | ... | $R_N$ |
|---|---|---|---|---|---|
| $w_1$ | 1 | 0 | 1 | ... | 0 |
| $w_2$ | 0 | 1 | 0 | ... | 1 |
| $w_3$ | 1 | 1 | 0 | ... | 1 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| $w_n$ | 0 | 0 | 1 | ... | 1 |
| | | | $v(w_i)$ | | |

**ج) خوشه‌بندی**[8]: خوشه‌بندی با استفاده از روش خوشه‌بندی سلسله مراتبی تجمیعی انجام می‌گیرد. معیار شباهت کلمات با استفاده از فاصله ویرایش کلمات با آستانه عدد ۱ در نظر گرفته می‌شود. با توجه به اینکه کلمات در دسته‌هایی با طول یکسان کلاس‌بندی شده‌اند، اعضای داخل هر دسته علاوه بر شباهت با اعضای داخل دسته مربوطه، حداکثر با دسته بعدی شباهت خواهند داشت. توضیح بیشتر اینکه به دلیل بررسی شباهت کلمات دسته‌ها، از طول کم به طول زیاد، در صورت تشابه دو کلمه با طول متفاوت، در حین خوشه‌بندی کلمه با طول کم، در یک خوشه قرار می‌گیرند. بنابراین، در خوشه‌بندی نیازی به بررسی اعضای دسته قبلی نیست؛ خوشه‌بندی مطابق شبه کد الگوریتم (۱) انجام می‌گیرد.

**جدول (۲)**: دسته‌بندی کلمات (ماتریس ارتباط چندبخشی)

| $S_1$ | $R_1$ | $R_2$ | $R_3$ | ... | $R_N$ |
|---|---|---|---|---|---|
| $D_{1,1} = W_3$ | 1 | 0 | 1 | ... | 0 |
| $D_{2,1} = W_9$ | 0 | 0 | 0 | ... | 0 |
| $D_{3,1} = W_1$ | 1 | 0 | 0 | ... | 1 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| $D_{m1,1} = W_6$ | 0 | 0 | 1 | ... | 1 |

...

| $S_l$ | $R_1$ | $R_2$ | $R_3$ | ... | $R_N$ |
|---|---|---|---|---|---|
| $D_{1,l} = W_7$ | 1 | 0 | 1 | ... | 0 |
| $D_{2,l} = W_4$ | 0 | 0 | 0 | ... | 0 |
| $D_{3,l} = W_5$ | 1 | 0 | 0 | ... | 1 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| $D_{ml,l} = W_2$ | 0 | 0 | 1 | ... | 1 |

توضیح مراحل الگوریتم (۱) به شرح ذیل است:

[1] Advanced Encryption Standard
[2] Cipher Block Chaining
[3] Tokenizing
[4] Stop Words
[5] Jhazm
[6] Affinty Matrux
[7] Word Classification
[8] Clustering

۱. انتخاب یک کلمه در داخل دسته به‌عنوان مرکز خوشه؛

۲. مقایسه تشابه مرکز خوشه با کلمات داخل دسته فعلی، و دسته بعدی با بهره‌گیری از معیار تشابه که توسط الگوریتم‌های (۲ و ۳) معرفی شده است. البته در دسته مرزی آخر، فقط با کلمات دسته فعلی مقایسه انجام می‌گیرد. در صورت شباهت کلمه انتخاب شده با مرکز خوشه مطابق معیار شباهت، کلمه در خوشه مربوطه قرار می‌گیرد.

۳. مراحل فوق برای همه کلمات خوشه‌بندی نشده همه دسته‌ها اعمال می‌گردد و کلمات کلیدی جدول (۲)، مطابق جدول (۳) خوشه‌بندی می‌شود. در این جدول کلمه کلیدی $C_j$ مرکز خوشه و $c$ نشان دهنده تعداد خوشه‌ها است.

**جدول (۳)**: خوشه‌بندی کلمات

| $C$ \ $R$ | $R_1$ | $R_2$ | $R_3$ | ... | $R_N$ |
|---|---|---|---|---|---|
| $C_1$ | 1 | 1 | 1 | ... | 0 |
| $C_2$ | 0 | 1 | 0 | ... | 0 |
| $C_3$ | 0 | 0 | 1 | ... | 1 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| $C_c$ | 1 | 0 | 1 | ... | 1 |
| $Cluster$ | $v(C_i)$ | | | | |

همان‌طور که در بالا گفته شد، دو کلیدواژه می‌توانند با جایگزینی و یا اضافه شدن کاراکتر به یکدیگر شباهت داشته باشند. به همین منظور، برای بررسی شباهت دو کلیدواژه، الگوریتم (۲ و ۳) برای بررسی هر یک از این سنجه تشابه‌ها، مورد استفاده قرار گرفته است.

**د) مجموعه کلیدواژه فازی:** مالک داده، مجموعه کلیدواژه فازی مرکز هر خوشه را تولید می‌کند. فرض می‌شود $S_{C_i}$ مجموعه کلیدواژه فازی کلمه کلیدی $C_i$، با فاصله ویرایش $d$ باشد. جدول (۴) مجموعه کلید $S_{C_i} = \{C_{i,t}, 1 \le i \le c, 1 \le t \le |C_i|\}$ شامل مجموعه کلیدواژه فازی تولید شده مراکز خوشه، مطابق الگوریتم (۴) است [۷].

**ه) دریچه امن**[1]: مالک داده، مجموعه کلیدواژه مراکز خوشه، یعنی $S_{C_{i,t}}$ را با محاسبه $ES_{C_{i,t}} \leftarrow f_{k_0}\left(S_{C_{i,t}}\right)$ و شاخص‌های هر خوشه، یعنی $v(C_i)$ را با محاسبه $Ev(C_i) \leftarrow f_{k_1}(v(C_i))$ رمزگذاری کرده و در یک لیست دو ستونی برون‌سپاری می‌کند. برای رمزگذاری می‌توان از هر الگوریتم رمز متقارن بلوکی مانند AES استفاده می‌شود.

---

Generate_Clustering$(S, j, N[j], d)$

**Input**: Set of Clustering Step One $S_j$
, Number of Members $N[j]$ and d=Edit Distance
**Output**: Clustering Sets $O_n$

```
1.  for i ← 1 to j do
2.    for k ← 1 to N[j] do
3.      F[i][k] ←0;
4.    end for
5.  end for
6.  n←0;
7.  for k ← 1 to j do
8.    for i ← 1 to N[k] do
9.      if F[k][i]=0 then
10.       F[k][i] ←1;
11.       n ← n + 1;
12.       Set O_n ← S_k[i];
13.       for m ← i + 1 to N do
14.         if F[k][m] = 0 then
15.           if Stimulatory_R(S_k[i], S_k[m]) ≤ d then
16.             Set O_n ← S_k[i]US_k[m];
17.             F[k][m] ←1;
18.           end if
19.         end if
20.       end for
21.       for m ← 1 to N[k + 1] do
22.         if F[k + 1][m] = 0 then
23.           if Stimulatory_I(S_k[i], S_{k+1}[m]) ≤ d then
24.             Set O_n ← S_k[i] U S_{k+1}[m];
25.             F[k][m] ←1;
26.           end if
27.         end if
28.       end for
29.     end for
30. end for
31. return O_n;
```

**الگوریتم (۱)**: الگوریتم خوشه‌بندی

---

Stimulatory_$R(x, y)$

**Input**: x, y input string, *Length(x) =Length(y)*
**Output**: d Number deferent characters

```
1.  d ← 0;
2.  n = Lenght(x);
3.  for i ← 1 to n do
4.    if x_i ≠ y_i then
5.      d ← d + 1;
6.    end if
7.  end for
8.  return d;
```

**الگوریتم (۲)**: سنجه تشابه جایگزینی کاراکتر.

[1] Secure Trapdoor

Stimulatory_$I(x,y)$

**Input**: x, y input string, *Length(x) <Length(y)*
**Output**: d Number deferent characters

1. $i \leftarrow 1$;
2. $d \leftarrow 0$;
3. $n = \textbf{Lenght}(y)$;
4. **for** $j \leftarrow 1$ to $n$ **do**
5. **if** $x_i \neq y_j$ **then**
6. $d \leftarrow d + 1$;
7. $i \leftarrow i - 1$;
8. **end if**
9. $i \leftarrow i + 1$;
10. **end for**
11. **return** $d$;

**الگوریتم (۳):** سنجه تشابه اضافه شدن کاراکتر

**جدول(۴):** مجموعه کلیدواژه فازی

| $S_{C_i}$ \ $R$ | $v(C)$ |
|---|---|
| $(C_{1,1}, C_{2,1}, \dots, C_{1,\lvert S_{C_1}\rvert})$ | $v(C_1)$ |
| $(C_{2,1}, C_{2,1}, \dots, C_{2,\lvert S_{C_2}\rvert})$ | $v(C_2)$ |
| $(C_{3,1}, C_{3,1}, \dots, C_{3,\lvert S_{C_3}\rvert})$ | $v(C_3)$ |
| ⋮ | ⋮ |
| $(C_{c,1}, C_{c,1}, \dots, C_{c,\lvert S_{C_c}\rvert})$ | $v(C_c)$ |

Generate Fuzzy Set Gen_Fuzzy_Set ($W_i$, d)

Input: Keyword $W_i$ and Edit Distance d
Output: Fuzzy keyword Set $S_{wi,d}$

1. **if** $d > 1$ **then**
2. Gen_Fuzzy_Set ($W_i$, d-1);
3. **end if**
4. **if** $d = 0$ **then**
5. Set $S_{w_i,d} = \{Wi\}$;
6. **else**
7. **for** $k \leftarrow 1$ to $\lvert S_{w_i,d}\rvert$ **do**
8. **for** $j \leftarrow 1$ to $2 \times \lvert S_{w_i,d}[k]\rvert + 1$ **do**
9. **if** $j$ is odd **then**
10. Set $Temp = \lvert S_{w_i,d}[k]\rvert$;
11. Insert * at position$j + 1/2$ ;
12. **else**
13. *Set* $Temp = \lvert S_{w_i,d}[k]\rvert$;
14. Replace $\frac{j}{2} - th$ character with *;
15. **end if**
16. **if** $Temp$ is not in $S_{w_i,d-1}$ then
17. Set $S_{w_i,d} = S_{w_i,d} U \{Temp\}$ ;
18. **end if**
19. **end for**
20. **end for**
21. **end if**
22. **return** $S_{w_i,d}$

**الگوریتم (۴):** تولید مجموعه کلیدواژه فازی[۷].

## ۴-۲- تعامل کاربر با سرور ابری

### ۴-۲-۱- دریچه جستجو

در مرحله جستجو، کاربر مجموعه کلیدواژه فازی $S_{w'}$ را تولید می‌کند ($w'$ کلمه برای جستجو). سپس دریچه‌های جستجو را مطابق الگوریتم رمز مرحله ذخیره، با کلید $k_0$ طبق رابطه $T_{w'} = f_{k_0}(S_{w'})$ تولید می‌کند. درنهایت دریچه‌های تولید شده را جهت جستجو به سرور ابری ارسال می‌کند.

### ۴-۲-۲- پرس‌وجو

وقتی سرور ابری دریچه‌های جستجو را دریافت می‌کند، آن‌ها را با عناصر اولین ستون جدول فراداده رمز شده مقایسه می‌کند. یعنی تک‌تک عناصر دریچه $T_{w'}$ با عناصر $ES_{C_{i,t}}$ مقایسه می‌شوند و در صورت برابر بودن هر یک از عناصر، شاخص‌های متناظر یعنی $Ev(w')$ به‌عنوان نتیجه، به کاربر ارسال کرده و جستجو را خاتمه می‌دهد. علت ادامه ندادن جستجو، تجمیع کلمات کلیدی مشابه در یک خوشه است که عدم وجود دریچه تکراری را تضمین می‌کند.

### ۴-۲-۳- رمزگشایی

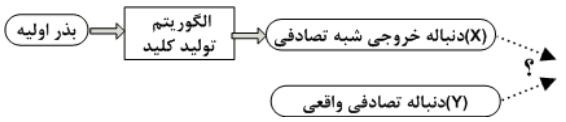
در این مرحله، کاربر شاخص‌های دریافتی از سرور ابری یعنی $v(w')$ را با محاسبه $v(w') = f_{k_1}(Ev(w'))$ رمزگشایی می‌کند. اگر به ازای $1 \le j \le N$، هر یک از $v(w')[j] = 1$ باشد کاربر $R_j$ را جزء رکوردهای حاوی کلمه کلیدی $w'$ در نظر می‌گیرد. سپس کاربر، شماره رکوردهای استخراج شده را به سرور ابری ارسال کرده و سرور ابری، محتوای رمز شده رکوردها را از داخل پایگاه داده رمز شده به کاربر برمی‌گرداند. درنهایت کاربر محتوای رکوردها را به‌صورت رابطه (۹) رمزگشایی می‌کند و به نتایج نهایی جستجو دست می‌یابد.

$$C_0 = IV = H(j), P_i = E_{sk}(C_{i-1} \oplus C_i), 1 \le i \le l$$
$$D_{R_i} = f_{sk}(ER_j) = P_1 || \dots || P_i \ , \ (1 \le i \le l) \qquad (۹)$$

## ۴-۳- ملاحظات امنیتی طرح پیشنهادی

### ۴-۳-۱- تولید کلیدهای رمزنگاری

مجموعه کلیدهای رمزنگاری $K = (sk, k_0, k_1)$ توسط الگوریتم تولید کلید رمزنگاری به شکل و شرایط زیر تولید و از طریق کانال امن در اختیار کاربران مجاز قرار می‌گیرد (شکل ۲).




**شکل (۲):** تولید شبه تصادفی کلیدهای رمزنگاری.

**تعریف ۱)** دو دنباله $X \stackrel{def}{=} \{X_n\}_{n\in\mathbb{N}}$ و $Y \stackrel{def}{=} \{XY_n\}_{n\in N}$ از نظر زمان چند جمله‌ای غیر قابل تمایز هستند، اگر برای هر الگوریتم زمان چند جمله‌ای احتمالی $D$، چند جمله‌ای مثبت $p(.)$، برای هر $n$ به‌اندازه کافی بزرگ رابطه (۶) برقرار باشد [۳۱]:

$$|P_r[D(X_w, w) = 1] - P_r[D(Y_w, w) = 1]| < \frac{1}{p(n)} \qquad (۶)$$

### ۴-۳-۲- الگوریتم‌های رمزنگاری

الگوریتم رمز بلوکی متقارن E برای رمزگذاری، D برای رمزگشایی و کلید $G(1^n)$ است و از نظر امنیت معنایی[1] مطابق تعریف (۲) می‌باشد.

**تعریف ۲)** طرح رمزنگاری $(G, E, D)$ با عملیات رابطه (۷)

$$P_r[D_d(E_e(\alpha))\ \alpha] = 1\ , \quad \alpha \in \{0,1\}^{l(n)} \qquad (۷)$$

و با طول بلوک $l$ دارای امنیت معنایی است $(d = e)$، اگر برای هر الگوریتم زمان چند جمله‌ای احتمالی A وجود داشته باشد الگوریتم زمان چند جمله‌ای احتمالی $A'$، به‌گونه‌ای که برای هر گروه داده $\{X_n\}_{n\in\mathbb{N}}$ با $|X_n| \le \text{poly}(n)$

برای هر دو تابع چند جمله‌ای محدود $f, h: \{0,1\}^* \to \{0,1\}^*$، برای هر چند جمله‌ای مثبت $P$ و $n$ به‌اندازه کافی بزرگ $n$ شرط رابطه (۸) برقرار باشد [۳۲].

$$P_r\left[A\left(1^n, E_{G_{1(1^n)}}(X_n), 1^{|X_n|}, h(1^n, X_n)\right) = f(1^n, X_n)\right] < P_r\left[A'\left(1^n, 1^{|X_n|}, h(1^n, X_n)\right) = f(1^n, X_n)\right] + \frac{1}{p(n)} \qquad (۸)$$

احتمال روی $X_n$، از پرتاب سکه و یا الگوریتم‌های G, E و یا الگوریتم A' به‌دست می‌آید.

### ۴-۳-۳- پنهان کردن طول قالب‌بندی[2]

در طرح پیشنهادی، بعد از خوشه‌بندی، تولید مجموعه کلیدواژه فازی مرکز خوشه و ایجاد شاخص‌های مربوط به رکوردها، قالب به‌صورت لیست دو گره‌ای درمی‌آید. به دلیل متفاوت بودن طول کلمات، تعداد مجموعه کلیدواژه فازی هر کلمه کلیدی متفاوت است؛ بنابراین دشمن و سرور ابری می‌توانند طول کلمه کلیدی را با توجه به تعداد مجموعه کلیدواژه فازی یاد بگیرد. برای حل این مشکل، ابتدا طولانی‌ترین گره‌ها را پیدا می‌شود (تعداد مجموعه کلیدواژه فازی و تعداد رکوردهای مرتبط) و سپس بقیه گره‌ها را به اندازه آن‌ها لایه گذاری[3] می‌شود. برای ارتقای کارایی جستجو، ترتیب خوشه‌ها به‌صورت نزولی بر اساس تعداد شاخص‌ها (رکوردهای مرتبط) مرتب می‌شود. به این ترتیب، طول واقعی مراکز خوشه و تعداد رکوردهای مرتبط از سرور ابری پنهان می‌باشد.

## ۵- بررسی و تحلیل امنیت

در این بخش، امنیت طرح پیشنهادی را از نظر قابلیت اطمینان و حفظ حریم خصوصی (حفظ محرمانگی) بررسی و تحلیل می‌شود. در طرح پیشنهادی اطلاعات برون‌سپاری شده شامل موارد زیر می‌باشد:

۱. رکوردهای پایگاه داده به‌صورت رمز شده با طول یکسان.
۲. فراداده رتبه‌بندی[4] شده (دریچه‌های جستجو و رکوردهای مربوطه) رمز شده با طول یکسان.
۳. کلیدهای رمزنگاری به‌صورت شبه تصادفی (غیر قابل تشخیص با تصادفی) تولید شده و از طریق کانال امن توزیع می‌شوند.

با توجه به موارد فوق در تبادل بین کاربران مجاز و سرور، سرو ابتدا دریچه‌های رمز شده دریافت و شاخص‌های رمز شده را برمی‌گرداند. در ادامه شماره رکوردها را به‌صورت آشکار دریافت و محتوی رکوردها را به‌صورت رمز شده برمی‌گرداند. در اینجا فرض بر اینکه به سامانه رایانش ابری اطمینان است ولی اعتماد نیست و نگرانی محرمانگی اطلاعات هست؛ بنابراین در این محیط، یک حمله کننده (سروری ابری یا هر دشمن دیگر) ممکن است داده‌های تبادل شده را از هر یک از نهادهای درگیر شنود کند. اطلاعات به سرقت رفته شامل دریچه‌های جستجوی رمز شده، شاخص رمزنگاری شده، شماره رکوردها و محتوی رکوردهای پایگاه داده می‌باشد.

**قضیه ۱.** تقاطع مجموعه‌های فازی $S_{w,d}$ و $S_{w_i,d}$ برای کلمه کلیدی $w_i$ و $w$ تهی نیست، اگر و فقط اگر $Ed(w, w_i) \le d$ باشد.

ابتدا ثابت می‌شود که مجموعه $S_{w,d} \cap S_{w_i,d}$ تهی نیست اگر $Ed(w, w_i) \le d$ باشد. برای اثبات این موضوع کافی است که یک عضو در مجموعه $S_{w,d} \cap S_{w_i,d}$ پیدا شود. با توجه به تعریف فاصله ویرایش، می‌توان بعد از $Ed(w, w_i)$ عملیات ویرایش، $w$ را به $w_i$ تبدیل کرد. با نشان کردن موقعیت‌های$Ed(w, w_i)$ در $w$ و انجام

[1] Semantic Security
[2] Framing
[3] Padding
[4] Ranked

ویرایش با عنصر★، $w^\star$ به‌دست می‌آید. می‌توان $Ed(w,w_i)$ ویرایش را در موقعیت‌هایی که حاوی ★ در $w^\star$ است انجام داد و $w^\star$ را به $w_i$ تبدیل کرد. بنابراین با توجه به $Ed(w,w_i) \le d$، عنصر $w^\star$ در هر دو مجموعه $S_{w,d}$ و $S_{w_i,d}$ وجود دارد.

در مرحله بعد، ثابت می‌شود که $Ed(w,w_i) \le d$ اگر $S_{w,d} \cap S_{w_i,d}$ تهی نباشد. از $w^\star$ به‌عنوان عنصر مشترک $S_{w,d} \cap S_{w_i,d}$ استفاده می‌شود. فرض می‌شود تعداد ★ در $w^\star$، k تا باشد، دو حالت در نظر گرفته می‌شود: اگر k = 0 باشد، یعنی برای تبدیل $w$ به $w_i$ نیاز به هیچ ویرایشی نیست. به عبارتی $w = w_i = w^\star$ می‌باشد. پس واضح است که $Ed(w,w_i) \le d$ است. اگر $k > 0$ باشد، برای هر ★ در $w^\star$، می‌توان ویرایشی در موقعیت ★ انجام داد و به کاراکتر متناظر آن در $w$ و $w_i$ تبدیل کرد. نوع نتایج را به‌ترتیب با $w_1^*$ و $w_{i_1}^*$ نشان داده می‌شود. با توجه به دو نوع که فقط در یک موقعیت تفاوت دارند، می‌توان با یک عملیات ویرایشی $w_1^*$ را به $w_{i_1}^*$ تبدیل کرد. این یعنی $Ed(w_1^*, w_{i_1 i}^*) \le 1$. بعد از اینکه همه $k$ ★ اعمال شد، $w$ و $w_i$ حاصل می‌شود. با توجه اینکه داریم $w^\star \in S_{w,d} \cap S_{w_i,d}$ و تعداد ★ در $w^\star$ بیشتر از d نیست. پس دارید $Ed(w,w_i) = k \le d$.

**قضیه ۲:** طرح پیشنهادی محرمانگی را حفظ می‌کند. یا به عبارتی طرح رمزنگاری پیشنهادی $(G,E,D)$ دارای امنیت معنایی است اگر طرح غیر قابل تمایز[1] باشد.

در این طرح سرور یا هر دشمن دیگر به داده‌های رمز شده دریچه‌های جستجو و رکوردهای پایگاه داده، طول داده‌ها (بعد از لایه‌گذاری)، شماره رکوردهای درخواستی دسترسی دارد. همچنین الگوریتم‌های رمزنگاری و رمزگشایی معلوم و کلیدهای رمزنگاری مخفی می‌باشند.

بنابراین باید ثابت شود اگر مطابق تعریف (۲) برای طرح پیشنهادی رابطه (۸) برقرار باشد آنگاه طرح دارای امنیت معنایی است.

که $E_{R_j} \leftarrow f_{sk}(R_j)$ یا $ES_{C_{i,t}} \leftarrow f_{k_0}\left(S_{C_{i,t}}\right)$ یا $E_{G_{1(1^n)}}$ $Ev(C_i) \leftarrow f_{k_1}(v(C_i))$ و یا $T_{w'} = f_{k_0}(S_{w'})$ باشد.

$A, A'$ الگوریتم زمان‌چند جمله‌ای احتمالی و $h(1^n, X_n)$ اطلاعات جزئی و کمکی[2] مانند طول بلوک‌های داده می‌باشد.

**اثبات:**

فرض می‌شود دارید $A' = M^A$، که $M$ یک ماشین اوراکل منفرد است، یعنی، برای هر $A$ یک $M$ وجود دارد و رابطه $A' = M^A$ برقرار است.

۱. $A'$ الگوریتم تولید کلید $G(1^n)$ را برای تولید مجموعه کلید $K = (sk, k_0, k_1)$ فراخوانی می‌کند.

۲. $A'$ الگوریتم رمز را با ورودی ساختگی[3] متن کشف $1^{|\propto|}$ برای به‌دست آوردن متن رمز $\beta = E_k(1^{|\propto|})$ فراخوانی می‌کند. $E_k$ یکی از توابع $f_{sk}$، $f_{k0}$ و $f_{k1}$ می‌باشد.

۳. $A'$ با ورودی $\left(1^n, \beta, 1^{|\propto|}, h(1^n, \propto)\right)$ الگوریتم $A$ فراخوانی می‌کند و خروجی را به‌دست می‌آورد.

در اینجا $A'$ به‌صورت یک ماشین اوراکل توصیف می‌شود که از یک اوراکل منفرد (با ورودی هر نوع داده) $A$ ساخته شده است، علاوه بر این الگوریتم‌های ثابت $G$ و $E$ را هم فراخوانی می‌کند. همچنین در ساختن $A'$ نه‌تنها به $h$ و $f$ وابسته نیست بلکه به توزیع داده آشکار برای رمز کردن هم وابسته نیست. بنابراین $A'$ زمان چند جمله‌ای احتمالی است هر گاه $A'$ زمان چند جمله‌ای احتمالی باشد (صرف نظر از توابع $f, h$ و ورودی).

از عدم تمایز طرح رمزنگاری استفاده می‌شود که الگوریتم $A'$ به خوبی الگوریتم $A$ کار می‌کند. به‌طور خاص، اثبات از یک استدلال کاهش پذیری[4] استفاده کرده است.

## ۶- مقایسه نتایج و بررسی عملکرد

### ۶-۱- مقایسه نتایج

روش‌های مختلفی برای ارتقای کارایی جستجوی فازی بر روی داده‌های رمز شده ارائه شده است. در جدول (۵) روش طرح پیشنهادی و طرح‌های [۷، ۱۳، ۱۵، ۱۶، ۱۸ و ۲۱] برای انجام جستجوی مؤثر آمده است.

**جدول (۵):** روش‌های بهبود جستجو

| روش \ طرح | [7] | [13] | [15] | [16] | [18] | [19] | [20] | [21] | پیشنهادی |
|---|---|---|---|---|---|---|---|---|---|
| Wildcard | √ | | √ | | √ | | | √ | √ |
| Clustering | | | | | | | √ | | √ |
| Stemming | | | | | | √ | | | |
| LSH | | | | √ | | | | | |
| Dictionary | | √ | | | | | | | |
| BedTree | | | √ | | | | | | |
| Trie-traverse | | | | | √ | | | | |

طرح [۷] روش نشانه عام را معرفی کرده است. طرح [۱۳] بر مبنای لغت‌نامه‌ای با کم کردن اندازه شاخص، سربار ذخیره و ارتباط را بهبود دادند.

[1] Indistinguishable
[2] Auxiliary
[3] Dummy
[4] Reducibility

طرح [۱۵] بر مبنای bedtree اندازه و زمان ساخت و زمان جستجو را نسبت به نشانه عام بهبود داده است. طرح [۱۶] بر اساس LSH کارایی و دقت را افزوده است. طرح [۱۸] با قابلیت درخت پیشوند و اضافه کردن قابلیت تصدیق طرح [۷] را بهبود داده است. طرح [۲۱] با استفاده از روش تخصیص یک بردار به کل مجموعه کلیدواژه فازی برای کاهش حجم محاسبه و فضای ذخیره‌سازی استفاده کرده است.

در جدول (۶) مقایسه کارایی طرح پیشنهادی با طرح‌های [۷، ۱۸ و ۲۱] از نظر هزینه شاخص گذاری، هزینه تولید دریچه، هزینه جستجو و هزینه حجم دریچه‌ها، انجام گرفته است.

در اینجا N به‌عنوان تعداد اسناد، n به‌عنوان تعداد کلماتی ثابت، $l$ به‌عنوان حداکثر طول کلمات کلیدی ثابت، m به‌عنوان تعداد کل کلمات کلیدی فازی، M به‌عنوان حداکثر تعداد کلمات کلیدی فازی در مجموعه کلید $S_{w_{i,d}}$ و $M'$ تعداد کلمات کلیدی فازی برای جستجوی کلمات کلیدی است. در طرح پیشنهادی N به‌عنوان تعداد رکوردهای پایگاه داده و C تعداد خوشه می‌باشد.

**جدول (۶):** مقایسه کارایی.

| طرح / عملیات | [7] | [18] | [21] | پیشنهادی |
|---|---|---|---|---|
| هزینه شاخص‌دهی | $O(nM)$ | $O(nM)$ | $O(n)$ | $O(n)$ |
| هزینه تولید دریچه | $O(M)$ | $O(M)$ | $O(1)$ | $O(1)$ |
| هزینه جستجو | $O(M'h)$ | $O(M'h)$ | $O(m)$ | $O(m/C)$ |
| حجم دریچه‌ها | $O(mlM)$ | $O(mlM)$ | $O(mlM)$ | $O(MlC)$ |
| دقت | ۱۰۰٪ | ۱۰۰٪ | ۱۰۰٪ | ۱۰۰٪ |

از تجزیه و تحلیل نتایج فوق می‌توان نتیجه گرفت که طرح پیشنهادی از طرح‌های موجود در مراحل جستجو و میزان حجم حافظه کارآمدتر است.

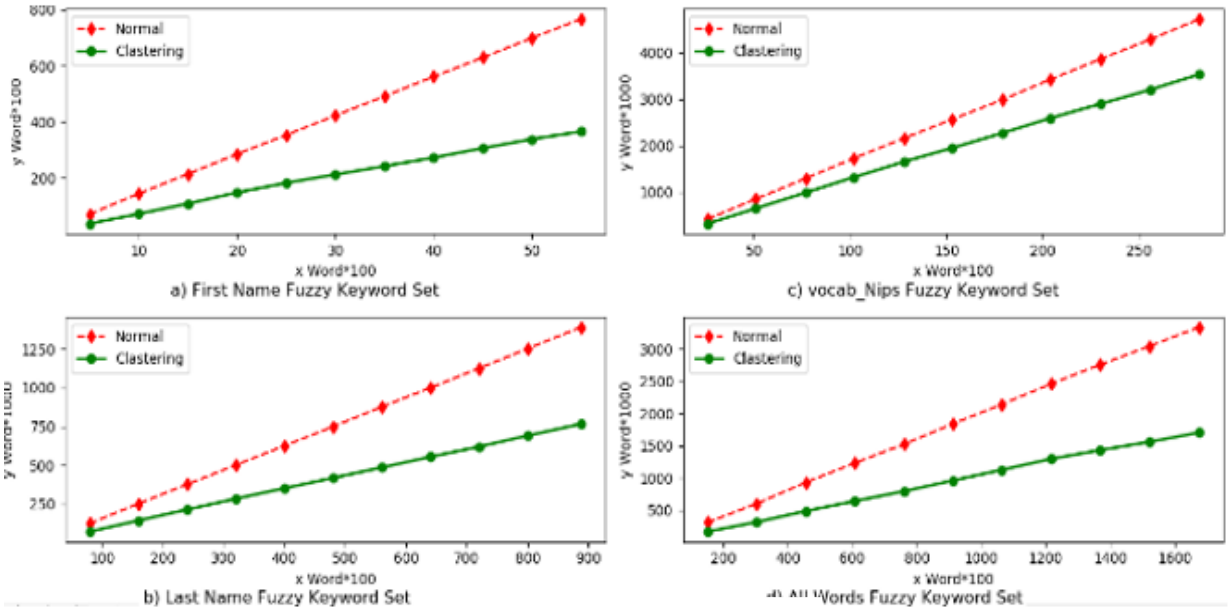
## ۶-۲- مقایسه عملکرد

در این بخش به ارزیابی کارایی طرح پیشنهادی بر روی مجموعه‌ای از داده‌ها که شامل انواع داده‌هایی مثل اسامی و لغات با حجم مختلف است، پرداخته می‌شود. فرض می‌شود مجموعه کلمات کلیدی را بعد از استخراج از پایگاه داده و حذف ایست واژه‌ها، در اختیار دارید.

پارامترهای که برای ارزیابی و مقایسه مدنظر قرار گرفته شامل تعداد کلمات اصلی، تعداد کلیدواژه فازی، حجم حافظه مورد نیاز برای ذخیره دریچه‌های جستجو، سرعت جستجو و قابلیت اطمینان جستجو است که با طرح‌های مشابه مانند طرح‌های [۷ و ۲۱] مقایسه شده است. با توجه به اینکه اصلی‌ترین منابع برون‌سپاری در رایانش ابری، حجم حافظه مورد نیاز و زمان جستجو است، بنابراین بهبود اصلی بر روی این دو پارامتر انجام گرفته است.

برای ارزیابی از یک سامانه رایانه‌ای با پردازنده اصلی اینتل Core i3-2350M CPU 4.0GB RAM با 2.30 GHz, استفاده شده است. کد (بر اساس الگوریتم) به زبان ویژوال $C++$ در یک سامانه عامل ۶۴ بیتی ویندوز ۸، نوشته شده است. تمامی الگوریتم‌های طرح پیشنهادی و طرح‌های مشابه در شرایط یکسان پیاده‌سازی و با داده‌های آزمایشی یکسان، مورد ارزیابی قرار گرفته‌اند. روال اجرای طرح پیشنهادی با بقیه طرح‌ها به جز الگوریتم‌های خوشه‌بندی یکسان است. الگوریتم رمز ۲۵۶ بیتی AES با مد عملکردی CBC برای ذخیره و جستجوی امن دریچه‌ها استفاده شده است. حداکثر طول کلمات کلیدی ۳۰ کاراکتر و فاصله ویرایش (آستانه تحمل خطا) عدد ۱ فرض شده است. جهت ارتقای کارایی، کلیدواژه‌های که کمتر از ۱۵ کاراکتر باشند با ۱۶ کاراکتر و اگر بیشتر باشند با ۳۲ کاراکتر لایه گذاری شده‌اند.

قابل توجه است که آزمایش طرح پیشنهادی و طرح‌های قبلی بر روی بیش از ۱،۰۰۰،۰۰۰ کلمه در قالب ۸ مجموعه داده اجرا گردیده است که در اینجا تنها به گزارش ۴ نمونه اکتفا می‌کنیم.




**شکل(۳)**: تعداد مجموعه کلیدواژه فازی.

جزئیات این چهار مجموعه داده کلمات متعلق به دانشگاه کالیفرنیا به شرح ذیل است [۳۳]:

a) ۵۵۰۰ نام ۳ تا ۱۱ کاراکتری به زبان لاتین

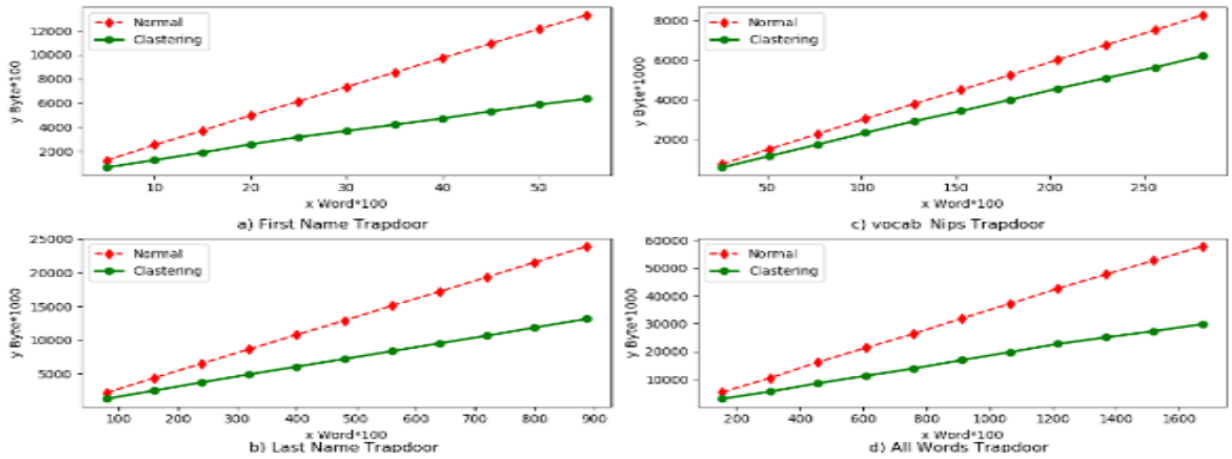
b) ۸۸۷۰ نام خانوادگی ۵ تا ۱۷ کاراکتری به زبان لاتین

c) ۲۸۱۰۰ لغات ۱ تا ۲۰ کاراکتری به‌کار برده شده در مجموعه مقالات کنفرانس نیپس[۱] به زبان لاتین

d) ۱۶۷۳۵۰ کلمه ۱ تا ۲۶ کاراکتری موجود در لغت‌نامه جیبی تزاروس[۲]

شکل (۳) تعداد مجموعه کلیدواژه تولید شده، شکل (۴) حجم دریچه‌های جستجو و شکل (۵) زمان جستجوی کل کلمات کلیدی را بر روی هر چهار مجموعه داده برای طرح پیشنهادی و دیگر طرح‌ها را نشان می‌دهد. مقایسه در نمودارها به ترتیب مجموعه داده‌های فوق می‌باشد.

همان‌طوری که شکل (۳) نشان می‌دهد تعداد مجموعه کلیدواژه فازی تولید شده طرح پیشنهادی نسبت به سایر طرح‌ها به‌طور چشم‌گیری کاهش یافته و در تعداد بالا تقریباً به نصف می‌رسد. حجم کلیدواژه‌های تولید شده به ظاهر کلمات بستگی دارد و در واقع میزان شباهت کلمات هستند که شیب رشد تعداد کلیدواژه‌ها را تعیین می‌کنند؛ اما در تمامی آزمایش‌ها انجام گرفته، شیب رشد تعداد مجموعه کلیدواژه فازی طرح پیشنهادی نسبت به طرح‌های قبلی کمتر بوده است.




**شکل(۴)**: حجم دریچه‌های جستجو.

مستقیم با تعداد دریچه‌های دارد، مپرداخته می‌شود. همان‌طوری که در شکل نیز می‌بینید حجم و اندازه دریچه‌های جستجو طرح پیشنهادی نسبت به سایر طرح‌ها تقریباً به نصف رسیده است.

البته شیب اختلاف در مجموعه داده‌ها با توجه به نوع داده‌ها متفاوت است. قابل توجه است که حجم دریچه‌ها و تعداد مجموعه کلیدواژه‌ها تا حدی با یکدیگر رابطه مستقیم دارند، به‌طوری که اگر شیب در یکی افزایش یابد در دیگری نیز افزایش خواهد داشت؛ البته قابل توجه است که میزان افزایش متفاوت است.

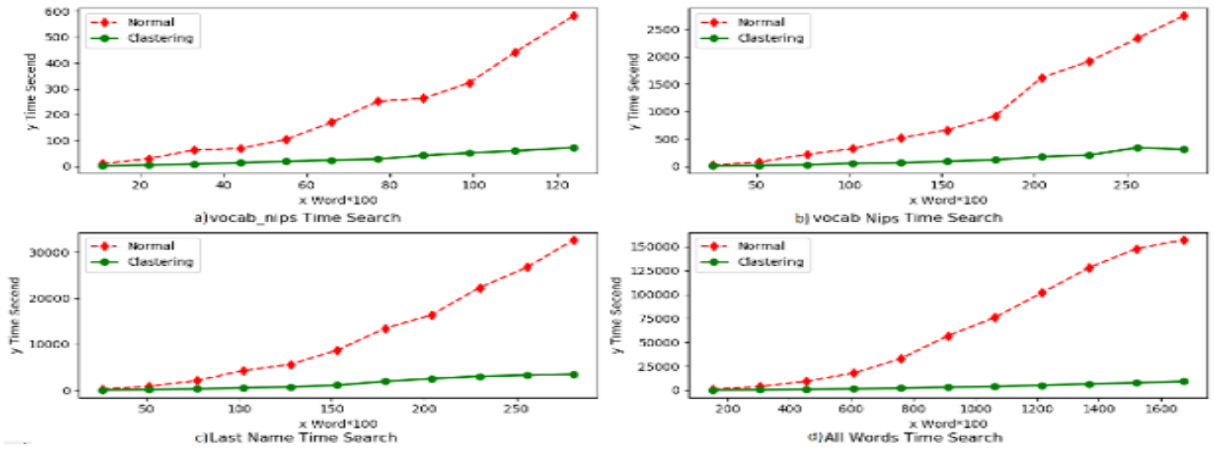
شکل (۵) نشان می‌دهد زمان جستجوی کلمات کلیدی، نسبت به تعداد کلمات در طرح پیشنهادی تقریباً خطی است ولی در سایر طرح‌ها به‌صورت نمایی افرایش پیدا می‌کند و اختلاف در تعداد کلمات بالا، قابل توجه و چشمگیر است.

عامل دیگر قابل بررسی میان طرح پیشنهادی و طرح‌های قبلی، قابلیت اطمینان جستجوی کلمات است و به معنای میزان خطای به‌دست آمده در جستجو و یا دقت جستجو است. این عامل در طرح پیشنهادی همانند طرح‌های قبلی، ۱۰۰٪ است و این یعنی به کمک خوشه‌بندی انجام گرفته هیچ داده‌ای از بین نرفته و تمامی کلمات به‌طور مجدد قابل بازیابی می‌باشند.

## ۷- نتیجه‌گیری

در این مقاله ابتدا لزوم برون‌سپاری در پایگاه داده و مزایای آن را بررسی شد و سپس به دغدغه حفظ حریم خصوصی (مؤلفه محرمانگی امنیت) اشاره شد. راهکار اصلی برای رفع این معضل، برون‌سپاری پایگاه داده، به‌صورت رمز شده می‌باشد. چالش‌های اصلی در این روش بازیابی اطلاعات و انجام پرس‌وجوی امن است که سرعت جستجو، حجم داده‌های کمکی و تحمل خطای از موضوعات اساسی است. راه‌حل به‌کار گرفته شده، روش‌های جستجوی فازی به‌صورت رمز شده است. حجم فراداده‌های ذخیره شده و سرعت جستجوی فازی از گلوگاه‌های این روش‌ها است. در این مقاله برای کاهش حجم فراداده و افزایش سرعت پرس‌وجو، از روش خوشه‌بندی با سنجه تشابه خاص استفاده شده است.

از طریق آزمایش‌ها صورت گرفته نشان داده شد که در طرح پیشنهادی، ضمن حفظ دقت جستجو، حجم فراداده کاهش و سرعت جستجو، افزایش پیدا کرده است. با تحلیل‌های انجام گرفته به این نتیجه رسیده می‌شود که طرح پیشنهادی، قابلیت مقیاس پذیری[1] خوبی دارد و هر چه تعداد کلمات بیشتر می‌شود، عملکرد طرح پیشنهادی از جنبه‌های مصرف حافظه و سرعت جستجو به نسبت روش‌های پیشین فاصله بیشتری گرفته و به نتایج بهتری دست می‌یابد.




**شکل(۵):** زمان جستجو.

## ۸- مراجع

[1] X. Song, D. Wagner, and A. Perrig, "Practical Techniques for Searches on Encrypted Data," In Proc. of the 2000 IEEE Symposium on Security and Privacy, IEEE, Berkeley, California, USA (2000), pp. 44–55.

[1] Scaling factor

over Encrypted Data in Cloud Computing," In Proc.IEEE INFOCOM, pp. 441–445,2010.

[18] J. Wang, H. Ma, Q. Tang, j. Li, H. Zhu, S. Ma, and X. Chen: "Efficient Verifiable Fuzzy Keyword over Encrypted Data in Cloud Computing," Com. SIS. Vol. 10, No, 2 Special Issue, pp. 667-684, 2013.

[19] Z. Fu, X. Wu, C. Guan, X. Sun, and K. Ren, "Toward Efficient Multi Keyword Fuzzy Search over Encrypted Outsourced Data with Accuracy Improvement," IEEE Trans. Inf. Forensics Security, pp. 2706 - 2716, 2016.

[20] N. Mahajan, V. Barkade, "Clustering Based Efficient Privacy Preserving Multi Keyword Search over Encrypted Data," IEEE Trans, pp. 1-6, 2018.

[21] X. Ge, j. Yu, H. Zhangr and R. Hao: "Enabling Efficient Verifiable Fuzzy Keyword Search over Encrypted Data in Cloud Computing", IEEE Access, August 17, pp. 1-15, 2018.

[22] L. Xie, Z. Wang, Y. Wang, H. Yang and J. Zhang, "New Multi-Keyword Ciphertext Search Method for Sensor Network Cloud Platforms", Sensors, vol. 18, no. 9, pp. 1-11, 2018.

[23] Z. Fang, J. Wang, B. Wang, J. Zhang, and Y. Shi: "Fuzzy Search for Multiple Chinese Keywords in Cloud Environment", tsp.techscience.com, 2019.

[24] J. Han, M. Kamber, "Data Mining Concepts and Techniques", Third Ed., 3rd, 2016.

[25] H.C. Chang, C.C. Chang, "Using Topic Keyword Clusters for Automatic Document Clustering", Proc. of the Third International Conf. on Information Technology and Applications IEEE, pp. 419-425, 2005.

[26] D. K. Jangir, S. Kurapati, and A. K. Gupta, "Comparison of Document(s) search by Keyword using Normal Search and Clustering based Search", Int. J. of Engineering Technology, vol. 5, no. 7, pp. 1-8, 2017.

[27] R. Handa, CR. Krishna, N. Aggarwa, "Document Clustering for Efficient and Secure Information Retrieval from Cloud", Wiley Online Library, 2019

[28] P. Samantaray, N. Randhawa, S. Pat, "An Efficient Multi-keyword Text Search over Outsourced Encrypted Cloud Data", Springer Nature Singapore Pte Ltd. 2019.

[29] A. Patidar, J. Agrawal, N. Mishra, "Analysis of Different Similarity Measure Functions and their Impacts on Shared Nearest Neighbor Clustering" Approach Int. J. of Computer Applications, vol. 40, no. 16, pp. 1-5, February 2012.

[30] J. Wang, J. Feng, G. Li, "TrieJoin: Efficient Triebased String Similarity Joins with Edit Distance Constraints", Proc. of the VLDB Endowment, pp. 1219--1230, 2010.

[31] O. Goldreich, "Foundations of Cryptography Basic Tools", Weizmann Institute of Science, 2004.

[32] O. Goldreich, "Foundations of Cryptography II Basic Applications", Weizmann Institute of Science, 2009.

[33] https://archive.ics.uci.edu/ml/machine-learning-databases

[2] J. Domingo-Ferrer, "A New Privacy Homomorphism and Applications," Information Processing Letters. Vol. 60, no. 5, pp. 277–82, Dec 1996.

[3] R. Agrawal, J. Kieman, R. Srikant, and Y. Xu; "Order-Preserving Encryption for Numeric Data," In Proc. of the ACM SIGMOD 2004 Conf. Paris, France, pp. 563-574, June 2004.

[4] R. Brinkman, J. M. Doumen, P. H. Hartel, and W. Jonker, "Using Secret Sharing for Searching in Encrypted Data," In Secure Data Management VLDB 2004 Workshop, Volume LNCS 3178, Toronto, Canada, August 2004. Springer-Verlag, Berlin, pp. 18–27.

[5] H. Hacıgumus, R. Iyer, and S. Mehrotra: "Executing SQL over Encrypted Data in the Database Service Provider Model," In SIGMOD Conf., pp. 677-688, 2002.

[6] H. Hacıg, B. Iyer, S. Mehrotra, "Efficient Execution of Aggregation Queries over Encrypted Relational Databases," In ACM SIGMOD, 2002 June 46, Madison, Wisconsin, USA Copyright 2002 ACM 1581134975/02/06.

[7] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren and W. Lou, "Fuzzy Keyword Search over Encrypted Data in Cloud Computing," In Proc. of the 29th IEEE Int. Conf. on Computer Communications, pp. 1-5, 2010.

[8] E. J. Goh, "Secure Indexes," In Cryptology ePrint Archive, Report 2003/216, 2003.

[9] R. Curtmola, J. Gary, S. Kamara, and R. Ostrovsky, "Searchable Symmetric Encryption: Improved Dentitions and Efficient Constructions," In Proc. ACM Conf. Comput. Commun. Secur, pp. 79-88, 2006.

[10] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-Preserving Multi-keyword Ranked Search over Encrypted Cloud Data," In Proc.IEEE INFOCOM, pp. 829-837, 2011.

[11] Z. Xia, X. Wang, X. Sun, and Q. Wang, "A Secure and Dynamic Multikeyword Ranked Search Scheme over Encrypted Cloud Data," IEEE Trans. Parallel Distrib, Syst., vol.27,no. 2, pp. 340-352, 2016.

[12] Z. Fu, K. Ren, J. Shu, X. Sun, and F. Huang, "Enabling Personalized Search over Encrypted Outsourced Data with Efficiency Improvement," IEEE Trans. Parallel Distrib. Syst., vol. 27, no. 9, pp. 2546 - 2559, 2016.

[13] C. Liu, L. Zhu, L. Li, and Y. Tan, "Fuzzy Keyword Search on Encrypted Cloud Storage Data with Small Index," ICCCIS 2011, pp. 269–273, 2011.

[14] M. Chuah and W. Hu, "Privacy-aware Bedtree Based Solution for Fuzzy Multi-Keyword Search over Encrypted Data," ICDCSW 2011, pp. 273–281, 2011.

[15] M. Kuzu, M. S. Islam, and M. Kantarcioglu, "Efficient Similarity Search over Encrypted Data," 28th Int. Conf. on Data Engineering, pp. 1156–1167, 2012.

[16] B. Wang, S. Yu, W. Lou, and Y. T. Hou, "Privacy-preserving Multi Keyword Fuzzy Search over Encrypted Data in the Cloud", In Proc. IEEE INFOCOM, pp. 2112–2120, 2014.

[17] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Enabling Efficient Fuzzy Keyword Search

**Journal of Electronical & Cyber Defence**
**Vol. 8, No. 3, 2020, Serial No. 31**



# Fuzzy Keyword Search Scheme on an Encrypted Database in Cloud Computing Using Word Clustering

Y. Dehghanian, M. Ghayouri Sales*, A. Rahimi

*Imam Hosein University

(Received: 17/11/2019, Accepted: 01/02/2020)

**ABSTRACT**

*Database outsourcing in cloud computing is one of the main solutions to maintain and access a database easily without the need for large infrastructure investment. Although data owners usually trust service providers and infrastructure providers in terms of maintainability and accessibility, but they are concerned about the privacy and confidentiality of information, and therefore prefer to keep data encrypted on cloud servers. Encrypted data is not searchable, and a solution needs to be provided by the server to search for that data. One solution is to use keyword indexing as metadata alongside the encrypted database. There are several key challenges to using these solutions: high volume of indexes, user error exposure and search speed. In this study, fuzzy keyword sets are used instead of fixed words when searching for users, and in order to reduce storage space using the keyword clustering method, appropriate fuzzy keyword sets are selected and metadata with less production volume is used and encrypted. Using hierarchical clustering methods with specific metrics, the same keywords are placed in a cluster, and to find the desired keyword, it is not necessary to search all metadata and thus the search time is reduced. Practical results and evaluations show that the proposed method is practical, safe and efficient.*

**Keywords**: Fuzzy Search, Cloud Computing, Database Outsourcing, Searchable Encryption, Clustering

* Corresponding Author Email: Ghayoori@ihu.ac.ir