iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** VII **Month of publication:** July 2023

DOI: https://doi.org/10.22214/ijraset.2023.54920

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue VII Jul 2023- Available at www.ijraset.com

Feature Matching Search Mechanism Over Encrypted Cloud Data

Mr. Rakshith P Athreya[1], Dr. Kanta D Devangavi[2]
[1]*PG Scholar, VTU, CPGSB, Muddenahalli, Chikkaballapur-562101*
[2]*Asst. Professor, Dept. Of CSE (MCA), VTU, CPGSB, Muddenahalli, Chikkaballapur-562101*

Abstract: ***In recent years, there has been a lot of research done on encrypted search technologies. Creating indexes with separate keywords has led to high storage costs and poor search accuracy, which has made the issue urgently need to be solved as more and more data is being stored in the cloud. So, for encrypted cloud data, we provide a novel feature matching ranked search method (FMRSM) in this study. This approach builds indexes using the feature score algorithm (FSA), which enables multiple keywords to be mapped to an index dimension after being taken from a page as a feature. As a result, index storage costs may be decreased and encryption effectiveness can be increased. Furthermore, the FMRSM generates trapdoor processes using a matching score algorithm (MSA). The matching score algorithm is able to produce results with improved ranking accuracy because to FSA, which allows it to rank the search results according to the type of match and the amount of matching terms. A thorough investigation shows that our mechanism is more practical and efficient.***
Keywords: rating accuracy, matching score, feature score, storage cost, and encrypted search.

I. INTRODUCTION

Cloud computing is viewed as an alternative big business IT base model that can organize enormous processing, storage, and application assets while enabling clients to benefit from ubiquitous, advantageous, and on-interest system access to a common pool of configurable registering assets with exceptional efficiency and minimal financial overhead. Finding a solution to an accurate user question is particularly challenging in encrypted search algorithms.

Despite the fact that many searchable techniques enable multiple keywords, they do not take the relationships between the extracted keywords into account. Additionally, the current approaches of determining a keyword's significance for a text are insufficient, and when a significant number of keywords are extracted from the content, it will definitely result in a high storage cost.

As we all know, phrase search is a popular and effective approach for plain document queries.

Recently, many academics have started to employ the encrypted search system's phrase or combined keyword search strategy. The phrase search strategy can achieve greater query precision when compared to single-term or multi-keyword searches. However, if more phrases are taken from texts, the phrase search will become much more computationally expensive, and this cost tends to rise as more phrases are extracted. Therefore, it is crucial to create a workable and efficient search strategy that not only lightens the computing load but also produces correct results.

II. PREVIOUS WORK

1) With searchable encryption, users have access to keyword searches on encrypted content. A variety of encryption techniques, including symmetric and asymmetric encryption, homomorphic encryption, and secure index structures, are used to enable search capabilities while preserving the secrecy of the data. Examples of searchable encryption methods include Hierarchical Predicate Encryption (HPE), Conjunctive Keyword Search (CKS), and Privacy-Preserving Keyword Search (PPKS). Multiple parties can work together on the calculation of a function over their independently encrypted data using MPC protocols without sharing the underlying data with one another. MPC can be used to conduct secure search operations while retaining the material's encryption state while browsing through encrypted data that has been saved in the cloud. The Secure Multi-Party SQL (SMPSQL) system is an illustration of a system that allows searching and querying across encrypted databases while protecting users' privacy. Using PIR protocols, users may retrieve specific data from a database without having to reveal which precise data item is being accessed at any one moment. By hiding the user's search queries while the searches are being carried out, PIR may be used to conduct secure searches on encrypted cloud data. For keeping users' privacy during search operations, PIR systems including Single-Server PIR and Multi-Server PIR have been proposed as viable solutions.

2) Alice wants to preserve her email with storage-provider Bob using an email account from Yahoo or Hotmail. While giving Alice the capacity to store, retrieve, search, and delete emails, this storage provider shouldn't be able to deduce the contents of the messages that senders transmitted to Alice (via Bob as an intermediary) or the search criteria that Alice applied. A simple solution is for communications to be delivered to Bob in encrypted form, and Alice to request a copy of the complete database of encrypted emails anytime she needs to search for a specific message from Bob. However, this is incredibly ineffective. Effective communication while also respecting Alice's privacy. The authors of this work demonstrate how to build an Alice public-key encryption method that enables PIR searching over encrypted documents. This method is the first in a public-key situation with nontrivially low communication complexity to divulge no partial information about the user's search (including the access pattern). In their paper "Public-key Encryption with Keyword Search," Boneh, DiCrescenzo, Ostrovsky, and Persiano raised a challenge that is theoretically addressed here. We regard Single-Database PIR writing with sub-linear communication complexity, which is enabled by the core approach of our solution, to be of independent interest.

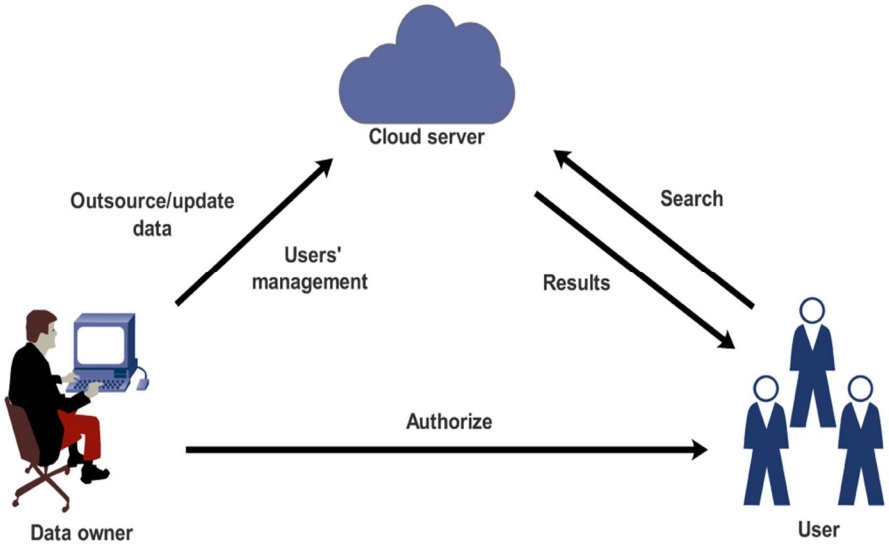
3) Some academics have suggested phrase search strategies for encrypted data in recent years. Contrary to methods that offer multikeyword search or conjunctive keywords, phrase search necessitates careful analysis of each information on the keywords, such as their placement, order, and meanings. Conjunctive keyword search and phrase search are both supported by the encrypted search technique created by Poon et al. To cut down on storage costs, the approach took into account all of natural language's statistical characteristics. Their subsequent research developed a bloom filter-based phrase search method that, for the first time, offers phrase-independent querying. A secure and effective phrase search over encrypted data may be performed thanks to the definition of a symmetric searchable encryption model by Tang et al. in their article

The following issues still need to be resolved even if the work mentioned above has enhanced searchable encryption technology in several ways: The search results of existing schemes depend on similarity scores or associated keyword relationships.

The amount of encryption calculation increases with the size of the dictionary, and when a large number of keywords are extracted, it will inevitably lead to huge computational cost. Low query accuracy will be the result since they do not fully take into account the quantity of matching keywords and the overall matching relationship..

III. SYSTEM DESIGN

There are three components to this system: a cloud server, a data consumer, and a data owner. All types of data are initially treated as documents by the data owner, who then uses the keywords from each document to generate an encrypted index before uploading both the encrypted documents and the encrypted indexes to the cloud server. The data user creates a trapdoor when querying and sends it to the cloud server for retrieval after first creating it through search control operations. Following receipt of the trapdoor, the cloud server determines the degree to which each document index and the trapdoor are identical, and then returns copies of the encrypted documents to the data user in accordance with the result. The data user ultimately obtains the findings that have been encrypted and decrypts them using access control.



Our technique aims to accomplish the following in order to lower storage costs and increase search accuracy.

1) *Feature Index:* Using a feature score technique, we aim to translate numerous keywords retrieved from a page to one dimension of the index. In other words, each index dimension corresponds to a feature.
2) *Precision in Matching Search:* To create a query vector, we compute matching scores based on the correspondence between the keyword set for the search and the feature dictionaries. The results of the query are delivered to the authorized user in order of the inner product of the index and the query vector.

IV. SYSTEM ANALYSIS

The following four crucial methods should be clarified in our design.

A. *Anti-term Frequency and Normalized Term Frequency*

The correlation between the query's keywords and characteristics is represented by the query score. A document score is frequently determined using term and anti-term frequency. The frequency of a term indicates how significant it is for a document. In general, we use a term's frequency as the number of times it appears in the text. The combined score in this study is generated using the standardized word frequency range, and each feature keyword's normalized term frequency may be determined.

B. *"Top-k" Query*

When a user types in terms to search, it's not always required to return all of the results. We utilized the "top-k" approach of querying in this study. The user must additionally include the option k, which specifies the total number of outcomes to be sent back, when submitting a query request. The cloud-based server ranks the returned results as part of the search operation and gives the query provided by the user the "top-k" document.

C. *A Precise Match and a Partial Match*

We categorize the matching relationship into two different forms, which are stated as follows, depending on the association among document attribute g_i (i=1,2,...,m) or query keyword set e and attribute d_j (j=1,2,...,n).
True match: It implies that the keywords in g or e are equivalent to those in d.
Partial match: This indicates that the terms in g or e do not quite match those in d_j.

D. *Ranking Accuracy*

To quantify the correctness of the results returned, we define ranking accuracy δ in this work. In a partial match, we presume that the document F_i feature g_i matches n terms from the search term set e. We say that a document F_i is returned in order if F_i comes before every other document whose number of matching keywords is less than n. Additionally, we specify that it is returned in order if the precise matching document is returned before all partial matching documents. In this study, we utilize the formula $\delta = R/k$ to determine the ranking accuracy, where r is the quantity of documents obtained in order upon running the "top-k" search and k is the overall number of matches received.

V. RESULTS

In this section, we use the Java language to assess the effectiveness of our FMRSM system. We compare our technique to MRSE (multi-keyword ranked search over encrypted cloud data) and CKSER (central keyword semantic extension ranked scheme) using the Request for Comments database (RFC) as our dataset.
Here is how the experiment is conducted. In FMRSM, the document features are created by extracting the keywords from each document, and the feature dictionary is created by removing duplicate features. Then, in order to create the keyword dictionary for MRSE and CKSER, we extract the keywords from each document. Finally, we conduct five groups of tests, including index creation, trapdoor generation, search, and measuring storage correctness and cost.
The total number of features in FMRSM and the total number of keywords in MRSE and CKSER are all connected to the length of time it takes to create an index, generate a trapdoor, and generate a search, as well as the size of the index and trapdoor. Each dimension of the index or query request corresponds to a keyword in the keyword dictionary in the MRSE and CKSER schemes.
As a result, the dimensions of the index and the query request will be (N+u+1) when the total number of keywords is N.

In FMRSM, the feature score method maps all of the retrieved keywords from a document to a single dimension of the index. The greatest dimension of the index and query request when there are m documents is (m+u+1).

The time it takes to create an index, generate a trapdoor, and conduct a search for FMRSM is shorter than it would be for MRSE and CKSER since the number of keywords N retrieved from the document is significantly more than the number of documents m. It should be noted that CKSER will take less time than MRSE to create an index, generate a trapdoor, and conduct a search since it leverages sub-matrices technology. In addition, FMRSM saves on index and trapdoor storage costs when compared to MRSE and CKSER.

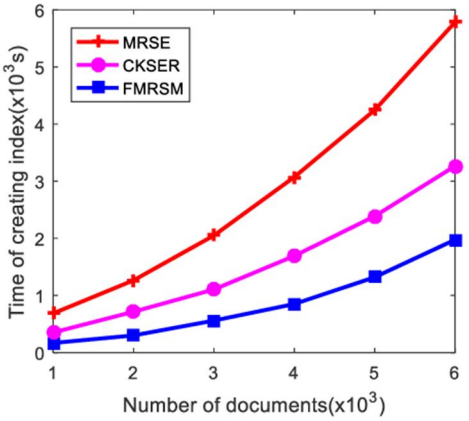


Fig. 1 Time of creating index.

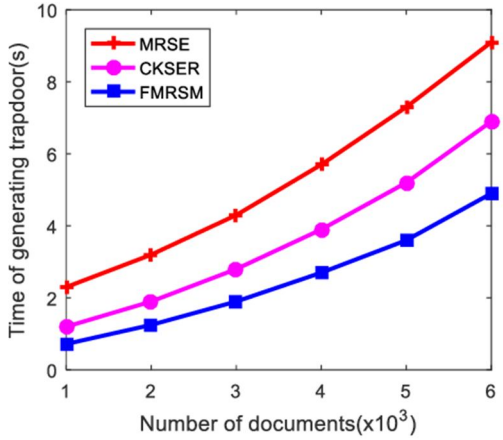


Fig. 2 Time of generating trapdoor.

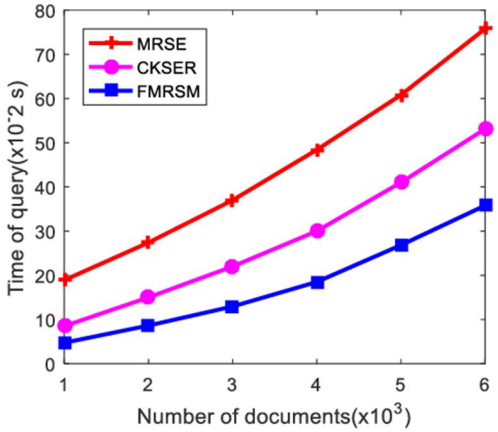


Fig. 3 Time of query.

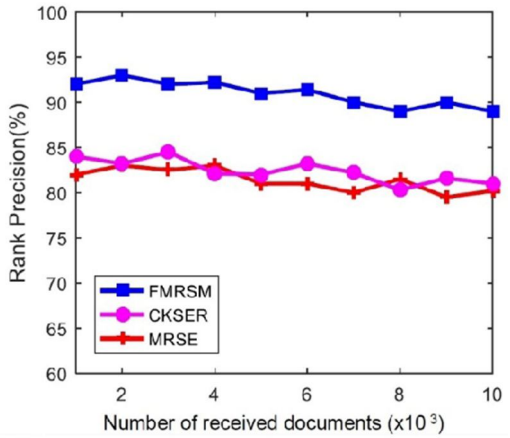


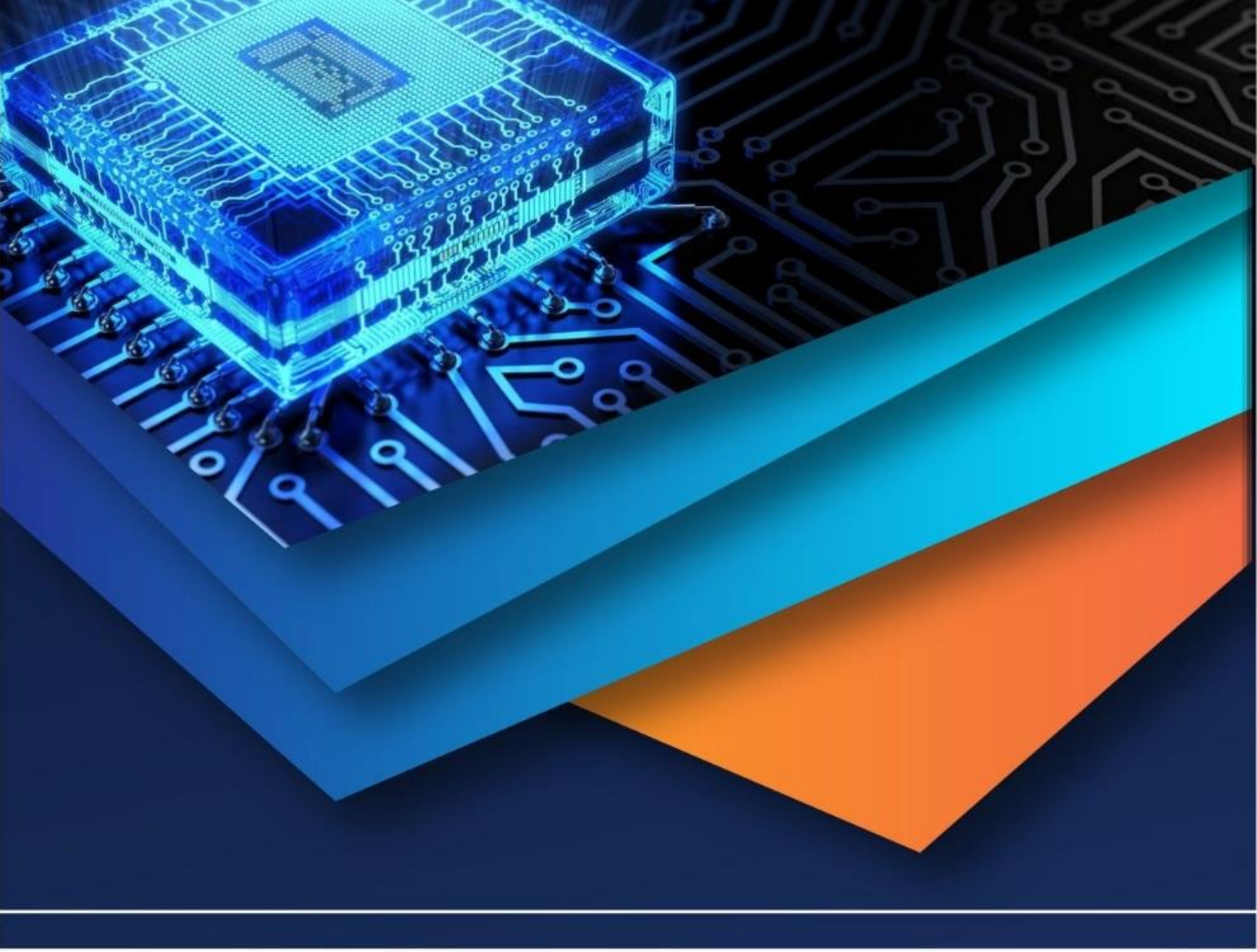
Fig. 4 Ranking accuracy.

VI. CONCLUSION

In this paper, we provide a feature matching method that prioritizes searches for encrypted cloud data. This system uses a feature score technique to build indexes such that several keywords retrieved from a page are only mapped to one dimension of the index. Using this method can result in the index's dimension being less than it would be if it were created using independent keywords. A matching score algorithm is also developed as part of the FMRSM trapdoor production process. Based on the kind of match and the number of keyword matches, this approach can accurately score the query request. Consequently, the search outcomes will be closer to what the users actually searched for. The results of the experiment show that our method has the ability to speed up index creation, trapdoor generation, and the search process itself. The cost of storage may be decreased while also improving ranking accuracy using our method, which is very useful.

REFERENCES

[1] Q. Chai and G. Gong, "Verifiable symmetric searchable encryption for semi-honest-but-curious cloud servers," published in the proceedings of the IEEE International Conference on Communications (ICC), held in Ottawa, Ontario, Canada, in June 2012, pages 917–922.
[2] X. Zhang, C. Xu, R. Xie, and C. Jin, "Designated cloud server public key encryption with keyword search from lattice in the standard model," [2] X. Zhang, C. Xu, R. Xie, and C. Jin, "Designated cloud server public key encryption with keyword search from lattice in The Chinese Journal of Electron, volume 27, issue 2, pages 304–309, March 2018.
[3] Y. Chang and M. Mitzenmacher, "Privacy preserving keyword searches on remote encrypted data," in Proceedings of the Fifth International Conference on Applied Cryptography and Network Security, volume 5, published by Springer in Berlin, Germany, in 2005, pages 442–455.
[4] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in Proceedings of the IEEE Symposium on Security and Privacy (S&P), held in Berkeley, California, United States of America, in May 2000, pages 44–55.
[5] A. J. Aviv, M. E. Locasto, S. Potter, and A. D. Keromytis, "SSARES: Secure searchable automated remote email storage," in Proc. 23rd Annu. Comput. Secur. Appl. Conf. (ACSAC), Miami Beach, Florida, United States of America, December 2007, pages 129–139.
[6] M. Raykova, B. Vo, S. M. Bellovin, and T. Malkin, "Secure anonymous database search," in Proceedings of the ACM Workshop on Cloud Computing Security (CCSW), New York, New York, United States of America, 2009, pages 115–126.
[7] S. M. Bellovin and W. R. Cheswick, "Privacy-enhanced searches using encrypted Bloom filters," IACR Cryptol. ePrint Arch., Tech. Rep., 2004, volume 22.
[8] Z. Fu, K. Ren, J. Shu, X. Sun, and F. Huang, "Enabling personalised search over encrypted outsourced data with efficiency improvement," published in IEEE Transactions on Parallel and Distributed Systems, volume 27, number 9, pages 2546–2559, September 2016.
[9] Z. Fu, X. Sun, Z. Xia, L. Zhou, and J. Shu, "Multi-keyword ranked search enabling synonym query over encrypted data in cloud computing," in Proc. IEEE 32nd Int. Perform. Comput. Commun. Conf. (IPCCC), San Diego, California, United States of America, December 2013, pages 1–8.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY