# OPTIMAL MULTIKEYWORD RANKED SEARCH MATCHING IN PUBLIC CLOUD WITH ENHANCED SECURE SEARCH SCHEME OVER ENCRYPTED DATA

**B. Madhava rao**
Assistant Professor, DEPT of CSE
St.Martin's Engineering College, Dulapally, Secunderabad, Telangana 500014

**Abstract**

As cloud computing becomes more popular, more users are moving their data to the cloud. Datasets are frequently encrypted before re-appropriation to preserve privacy. However, the usual encryption approach makes data use difficult. Searching for keywords in encrypted datasets is difficult. Many proposals exist to make encrypted data keyword searchable. Keyword based plans ignore the semantic representation information of users retrieval, and hence cannot fully satisfy users search goal. Making semantic search more effective and setting aware is therefore a difficult problem. These results show that the concept hierarchy and semantic link between ideas in encrypted datasets may be used to enhance secure search. 2 Cloud Servers - Enhanced Secure Search Scheme One stores reevaluated datasets and provides ranked results to data consumers. The other one processes the similarity scores between records and queries and sends them to the main server. To boost search efficiency, we structure the record list vectors into trees. We offer two secure strategies based on multi-keyword ranked search over encrypted cloud data. The strategy outperforms previous strategies in real-world tests. Our designs are secure under the known ciphertext and realized backdrop models.

KEYWORDS:
Multi-keyword Ranked Search, Public Cloud, Enhanced Secure Search Scheme, Encrypted Data.

## I. INTRODUCTION

Cloud computing is another nevertheless slow maturity paradigm of huge corporate IT infrastructure that delivers high quality applications and administrations [1]. The cloud customers might transfer their local sophisticated data system onto the cloud to eliminate the overhead of administration and local storage. Be that as it may, the security of rethought data cannot be guaranteed, since the Cloud Service Provider (CSP) has total control of the data. Thus, it is vital to scramble data before to putting them into cloud to secure the privacy of sensitive data [13]. Li et al. [18] presented a secure privacy guarding re-appropriated categorization in cloud computing. Notwithstanding, encryption for reevaluated data may guarantee privacy against illegal behaviors, it also makes powerful data utilization, such search over encrypted data, a very hard challenge. As of late, numerous academics have suggested an arrangement of productive search plots over encrypted cloud data. The basic interaction of search plan may be broken into five steps: extracting report features, producing a searchable file, generating search trapdoor, searching the

record based on the trapdoor and returning the search results. These search plans supply various inquiry capabilities, counting solitary watchword search [2, 3, 4, 5, 6], multi catchphrase search [7, 8, 9, 10], fluffy catchphrase search [9, 11] similarity search [12], and so on. Nonetheless, all the present searchable encryption schemes, which think about keywords as the record feature, don't take the semantic relations between words into mind, both in the ways of extracting record characteristics and creating search trapdoor. As we all know, there are a variety of semantic relations between words [14], including synonymy and domain correlation. Thinking about the possibly massive quantity of reevaluated data archives in the cloud, the search accuracy and search proficiency are impacted badly if the semantic relations between words are not handled correctly. We are now providing a full breakdown of the current issues with the searchable plans. Right off the start, at the stage of extracting record features, the data owner records the heaviness of each word in a record and then picks t words with top-t loads as the feature of the record. In the interaction presented above, each two words with varied spelling are deemed uncorrelated, which is illogical. For example, two words "trousers", "pants" are remarkable in the point of view of spelling, however they are semantically equivalent. When semantic relations between words are ignored, the accuracy of the archive characteristics suffers, which in turn increases the heaviness of the words. It is also imperative that the search trapdoor is generated only from the keywords provided by the data user, since expanding the search keywords is impossible if the data user is unable to articulate his search expectation clearly. In this situation, the data user may get an unnecessary archive or may not receive the items that are really needed. On light of the possibly enormous size of the archive set reevaluated in the cloud server, it's critical to know the genuine search aim of the data user in order to prevent returning extraneous records and therefore increase search efficiency. Thirdly, a search demand typically revolves on a topic, and certain search words may be seen as the attribute of the topic, for example, birthday is an attribute of an individual. In present search strategies, an attribute value is frequently handled as a catchphrase that disregards the connection with the topic and resulting in bigger catchphrase dictionary, which then adversely influences the search accuracy and proficiency. Along these lines, it is an important and demanding job to conduct semantic search over encrypted data. In this work, we propose a competent searchable encrypted conspiracy based on idea hierarchy allowing semantic search with two cloud servers. Domain-related information from the rethought dataset is used to build an idea hierarchy tree. We broaden the idea hierarchy to encompass additional semantic relations between ideas. With the assistance of extended idea hierarchy, archive characteristics are extracted all the more definitely and search words are very much increased based on the semantic relations between ideas. In order to determine if the value of an attribute is met by the search demand, two file vectors are generated for each archive. One is used to match search demand ideas with the archive contents.

Correspondingly, the search trapdoor for a search demand likewise comprises two vectors. The reason why we chose two cloud servers is because two servers may save a lot of time in search. One is utilized to figure the similarity between the reports vector and the trapdoors vector. Another one is utilized to rate results and conveys them back to consumers. The following is a summary of our promises:

1) We examine the issue of the semantic search based on the idea hierarchy by employing two cloud servers. The idea hierarchy is stretched out to hold multiple semantic relations among ideas and utilized to extend the search keywords. To increase the efficiency and security of the search, the retrieval interaction is separated into two free approaches.

2) We propose a strategy to build the record file and search trapdoor based on the idea hierarchy to aid semantic search, which channels reports by evaluating the attribute value and classifies related reports based on the number of matched search words.

3) The security analysis confirms that our strategy is secure under the threat scenarios. A tree-based searchable record is constructed to increase search performance. According to tests on real-world datasets, our strategy is effective.

## II. RELATED WORK

Compared with the previous version [6], we have an innovation on this adaption is that we use two cloud servers to search, thus we produce another system model. There are four components in our environment as indicated in Fig. 1: the data owner, the data consumer, the cloud server an and the cloud server B. Data owner: The data owner encrypts the data kept locally and uploads it to the cloud server. In this article, an idea hierarchy is built based on the domain ideas related information on the dataset and two record vectors for each archive of the dataset are generated based on the critical ideas of the record and the idea hierarchy. Then, at that time, the searchable list which is constructed with all the list vectors is exported off the cloud A. Data users: The authorized data user submits a search demand. Then, at that time, the trapdoors are linked to the keywords are generated. At finally, the data user delivers the trapdoors to the cloud B. Cloud Server A: The cloud server A has two capabilities. One is putting away the re-appropriated dataset. The other one rates the results from the cloud B and delivers the particular encrypted archives that fulfill the search basis to data consumers. Cloud Server B: The cloud server B is utilized to process the similarity scores between archives vector and trapdoors vector when it obtains the trapdoor. The cloud B then delivers its findings to the cloud A when it has computed them.

Model of Threat

The previous paper [6] is fundamentally provided the threat model. Our proposal largely references to the double workers system [7] and the MRSE structure [7]. In this form, we believe the cloud worker to be semi-legitimate, which is embraced by most past works [7],[8],[9], in other words, who sincerely executes the convention as it is characterized and accurately returns the query items, however who may likewise attempt to surmise private data by dissecting the reevaluated dataset, accessible index and question assessment. What's more, we assume that there is no plot between two cloud employees.

Because of what data the cloud workers learns, we explore two threat models [15] as follows. Realized Ciphertext Model The recognized ciphertext model indicates that the cloud employees may obtain to the scrambled data which includes the records and indexes reevaluated by data owners, but the workers cannot apprehend the plaintext data in the bottom layer of the ciphertext. Known Background Model In this more amazing approach, the cloud employees have to have much more open data contrasted to known ciphertext model. These data contain the encoded data and the relationship between provided pursues needs (trapdoors) and the data set concerning the factual data. As the example which may be assailed in the current scenario, the cloud workers can gather/perceive some recovered catchphrases by using the known trapdoor data and the recurrence of records/watchwords.

**Design Goal**

We increase the piece of design aims to make this article more clear than prior paper [61]. To ensure that our replies can be implemented accurately and efficiently under the previously specified danger

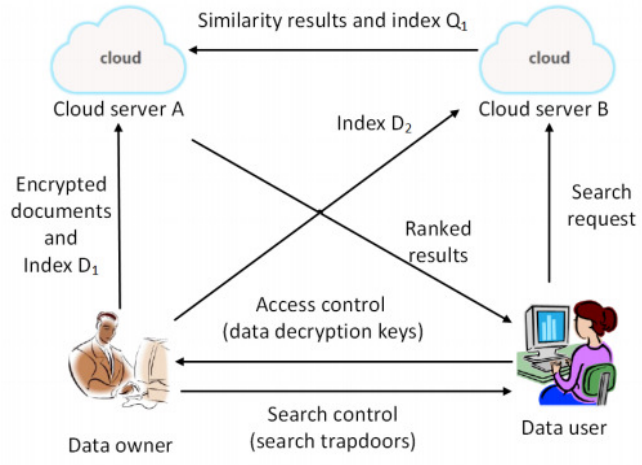


Fig. 1. System model

Models, our plans need fulfill two prerequisites: semantic recovery reliant on concept hierarchy and privacy preserving. The semantic recovery dependant on idea hierarchy means that our plan can figure the similarity scores between the data and the hunt solicitation and deliver the positioned results which satisfied the inquiry solicitations of customers. In this part, we portray privacy guarding in depth. In the investigation measures under the haze employees, our plans should fulfill the preceding privacy assurance:

1) Data privacy. At the point when we supply data records to customers, we furthermore need to ensure the private of the archive security, which is data privacy. To overcome this problem, the usual symmetric cryptography has been offered. The benefit of this encryption is that we can apply a symmetric key jumbled the data reports prior to outsoaring.

2) Index privacy. Index privacy is that the cloud workers cannot determine the connection between the watchwords and the encoded reports via the scrambled index.

3) Concept privacy. In this work, we acknowledge that the ideas and the watchwords are related to a certain degree. Thusly, we need to ensure that the security trapdoor we built doesn't disclose the watchwords and the inquiry data of customers.

4) Trapdoor unlinks ability. While the cloud workers restore archives, it may access to the generated trapdoors. Consequently, we need verify that the arbitrariness of trapdoor

age. At the comparable time, we need to ensure that similar enquiries link with numerous trapdoors. Along these lines, the cloud worker cannot obtain connections which exist in these trapdoors.

## III. PROPOSAL WORK

Creating a Document Index Vector for a More Secure Search Enhanced Search Scheme

Two index vectors should be generated for each document in the dataset as we introduce the "attribute-value" relation in the hierarchy; one vector is used to match concepts in the search request, while the other is used to determine whether the value of an attribute is satisfied by the search request. The process of generating these two n-dimension index vectors based on the expanded concept hierarchy is shown as follows. For a document F, we denote its two index vectors by D1 and D2. D1 [I] corresponds a node (which holds the concept chi) in the dimension for each of D1's dimensions. If F has the concept chi, then D1[I] = 1,otherwise D1[i] = 0. Similarity, each dimension of D2,denoted by D2[i], corresponds to a node (stores concept ci)in the hierarchy.

**Creating a Trapdoor Mechanism**

There are various concepts that make up a search request. Semantic similarity between a search concept and its procedure concepts in the extended concept hierarchy is calculated after a search request is received. For each search concept, the candidate concepts are its “brother”, “father”, and “direct child”nodes in the concept hierarchy. We let γ be a parameter todetermine if a candidate concept merit to be addedto search queries. To be particular, given a search concept c1 andits candidate concept c2, if sim(c1, c2) > γ, then we include c2to the search request. An expanded search request is then generated at the conclusion. Note that we do not attempt to deal with attributeconcepts in the concept expanding process above. For a search request comprising numerous concepts, twon-dimension vectors are also generated, one is used tostore the information about concepts in the search requestand another one is used to store the search limitation onattribute. For a search request Q, we denote its two searchvectors by Q1 and Q2. The process of generating the valueof each dimension of Q1 is same to that for D1, that is,if Q includes ci, then Q1[i] = 1, else Q1[i] = 0. If cj is an extended concept of ci and sim(ci, cj) =, then Q1[j] = for the extended search concepts. If the value of attribute concept ci is restricted and the constraint value is, then vector Q2[i] =, else vector Q2[i] = 0. We also consider T in Fig. 4as an example to show the process. Assume that a searchrequest Q comprises concepts e, b, f, j following concept extendingprocess, where j is an attribute concept whose value shouldsatisfy val(j) > 1990. The search trapdoor Q1 and Q2 for Q. Given the index vectors of a document F and the searchtrapdoor of a search request Q, the search procedure isconducted as follows. Firstly, the procedure evaluates whetherthe document fulfils search constraints specified in searchrequest utilising vectors D2 and Q2. A similarity score between a document and a search request is determined by multiplying the document's similarity score by the number of concepts F and Q that are

conceptually comparable. At end, all the linked documents aresorted based on their similarity scores and the top-k relateddocuments are returned to the user, where k is a parameterreceived from the user.

**Conclusion**

In this study, to solve the challenge of semantic retrieval, we propose effective strategies based on concept hierarchy. Our solutions utilise two cloud servers for encrypted retrieval and provide contributions both on search accuracy and efficiency. To boost accuracy, we extend the concept hierarchy to widen the search criteria. The concept hierarchy is used to establish a tree-based index structure to arrange all of the document index vectors, which is important for search performance. The security study reveals that the suggested system is secure under the threat models. Experiments on real world dataset indicate that our technique is efficient.

**References**


1. L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, "A break in the clouds: towards a cloud definition," ACM SIGCOMM Comput. Commun. Rev., vol. 39, no. 1, pp. 50–55, 2009.
2. C. Wang, N. Cao, K. Ren, and W. Lou, "Enabling secure and efficient ranked keyword search over outsourced cloud data," IEEE TPDS, vol. 23, no. 8, pp. 1467–1479, 2012.
3. D. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in Proc. of S&P, 2000.
4. R. Curtmola, J. A. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions," In Proc. of ACM CCS, 2006, pp. 79–88.
5. A. Swaminathan, Y. Mao, G.-M. Su, H. Gou, A. L.Varna, S. He, M. Wu, and D. W. Oard, "Confidentiality-preserving rank-ordered search," in Proc. of the 2007 ACM Workshop on Storage Security and Survivability, 2007, pp. 7–12.
6. S. Zerr, D. Olmedilla, W. Nejdl, and W. Siberski, "Zerber+r: Topk retrieval from a confidential index," in Proc. of EDBT, 2009, pp. 439–449
7. N. Cao, C. Wang, and M. Li, "Privacy-preserving multi-keyword ranked search over encrypted cloud data," Parallel and Distributed Systems,IEEE Transactions on, vol.25,no.1, pp.222-233,2014.
8. W. Sun, B. Wang, N. Cao, M. Li, W. Lou, YT. Hou, and H.L, "Privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking," in Proc. of ACM SIGSAC symposium on Information, computer and communications security, 2013, pp. 71–82.
9. M. Chuah and W. Hu, "Privacy-aware bedtree based solution for fuzzy multi-keyword search over encrypted data," in Proc. of the 31st ICDCSW, 2011, pp. 273–281.
10. Ayad Ibrahim, Hai Jin, Ali A. Yassin, and Deqing Zou, "Secure Rank-ordered Search of Multi-

keyword Trapdoor over Encrypted Cloud Data," in Proc. of APSCC, 2012 IEEE Asia-Pacific, pp. 263–270.
11. 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 IEEE INFOCOM'10 Mini-Conference, San Diego, CA, USA, March 2010, pp. 1–5.
12. C. Wang, K. Ren, S. Yu, K. Mahendra, and R. Urs, "Achieving Usable and Privacy-Assured Similarity Search over Outsourced Cloud Data," in Proc. of IEEE INFOCOM, 2012.
13. S. Kamara and K. Lauter, "Cryptographic cloud storage," in RLCPS, January 2010, LNCS. Springer, Heidelberg.
14. G. A. Miller, "WordNet: a lexical database for English," Communications of the ACM, vol.38, issue 11, pp. 39–41, 1995.
15. E.-J. Goh, "Secure indexes," Cryptology ePrint Archive, 2003, http://eprint.iacr.org/2003/216.
16. P. Scheuermann and M. Ouksel, "Multidimensional B-trees for associative searching in database systems," Information systems, vol. 7, issue 2, pp. 123–137, 1982.
17. D. Boneh and B. Waters, "Conjunctive, subset, and range queries on encrypted data," in Proc. of TCC, 2007, pp. 535–554.
18. P. Golle, J. Staddon, and B. R. Waters, "Secure conjunctive keyword search over encrypted data," in Proc. of ACNS, 2004, pp. 31–45.
19. J. Katz, A. Sahai, and B. Waters, "Predicate encryption supporting disjunctions, polynomial equations, and inner products," in Proc. of EUROCRYPT, 2008.
20. T. Okamoto and K. Takashima, "Adaptively Attribute-Hiding (Hierarchical) Inner Product Encryption," in Proc. of EUROCRYPT, 2012, pp. 591–608.
21. E. Shen, E. Shi, and B. Waters, "Predicate privacy in encryption systems," in Proc. of the 6th TCC, 2009.
22. V. Y. Lum and K. Meyer-Wegener, "An architecture for a multimedia database management system supporting content search," in Proc. of International Conference on Computing and Information, 1990, pp.304–313.
23. N. Guarino, C. Masolo, and G. Verete, "OntoSeek: Content-Based Access to the Web," IEEE Intelligent Systems, vol. 14, no. 3, pp. 70–80, 1999.
24. G. Varelas, E. Voutsakis, and P. Raftopoulou, "Semantic Similarity Methods in WordNet and their Application to Information Retrieval on the Web," in Proc. of 7th ACM international workshop on Web information and data management, 2005, pp. 10–16.
25. P. Cimiano, A. Pivk, L. Schmidt-Thieme, and S. Staab, "Learning taxonomic relations from heterogeneous sources," in Proc. of

the ECAI 2004 Ontology Learning and Population Workshop, 2004.

26. W. Wang, W. Meng, and C. Yu, “Concept Hierarchy Based Text Database Categorization in a Metasearch Engine Environment,” in Proc. of the First International Conference on Web Information Systems Engineering, 2000.
27. N. Nanas, V. Uren, and A. D. Roeck, “Building and applying a concept hierarchy representation of a user profile,” in Proc. of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, 2003.