ISSN No: 2348-4845
International Conference on Advanced Computer Science & Software Engineering (ICACSSE)
November 12, 2017 - Hyderabad, India
Paper Published in IJMETMR, A Peer Reviewed Open Access

A Proficient and Effective Multi Keyword Positioned Search System on Cipher Cloud Data

Kommana Nagendra Kumar
Department of Computer Science,
Technical lead, Trans IT mpower labs pvt.ltd,
Bangalore, Karnataka-560029, India.

Abstract:
Cloud is an Online Data Storage Unit. Where cloud computing offers close unbounded measure of assets limit (e.g., CPU, memory, System I/O, circle, and so on.) at a focused rate and enables clients to get assets on-request with pay-as-you-go estimating model. Rather than acquiring high forthright expenses in obtaining Data Innovation (IT) foundation and managing the upkeep and updates of both programming and equipment, associations can outsource their computational needs to the cloud. With the advancement in Cloud computing an ever increasing number of touchy information is being fused into the cloud. To guarantee security and protection this information are first encoded before being transferred onto the cloud server's subsequently making search a muddled assignment. Despite the fact that in conventional cloud computing encryption searching plans enables client to search scrambled information through keywords safely.

These procedures utilized correct keyword search and will fall flat if there are any morphological variations or spelling mistakes. This prompts low in proficiency and furthermore influences framework ease of use severely. Fuzzy keyword search expands the framework ease of use by permitting coordinating the correct or storeroom coordinate content to the put away keywords and recovering the estimated nearest comes about. Fuzzy search enhances client search understanding by recommending the client conceivable search keyword and rectifying the typographical mix-ups. A primary test is the means by which to decrease the search time. We have utilized the enhanced nearness positioning to discover the score of significant answers, which thusly finds most important answers in less time.

In this paper, we have reviewed how to coordinate closeness positioning data into fuzzy search to get the significant search result in least time. Moment search is a developing data recovery worldview in which a framework discovers answers to an inquiry in a split second while a client sorts in keywords character-by-character. Utilizing fuzzy search the correct keywords are shown alongside likeness keywords, which take care of the issues looked by the cloud clients. We demonstrate that our proposed arrangement is secure and protection safeguarding, while effectively understanding the objective of fuzzy keyword search.

Index Terms:
Fuzzy keyword search, Cloud computing.

1. INTRODUCTION:

Cloud computing allows users and enterprises to store and process data either in a privately-owned cloud or on a third-party server located in cloud environment. More number of sensitive data is brought together into the cloud such as messages, individual health records, government archives etc. In cloud computing data owners distribute their outsourced data with multiple numbers of users. Sensitive information is encrypted before outsourcing in order to protect its confidentiality. Individual users can get specific data files they are interested in by using keyword search technique instead of retrieving all the encrypted files which is completely impossible in cloud computing scenarios.

Cite this article as: Kommana Nagendra Kumar, "A Proficient and Effective Multi Keyword Positioned Search System on Cipher Cloud Data", International Journal & Magazine of Engineering, Technology, Management and Research, Volume 4 Issue 11, 2017, Page 145-150.

ISSN No: 2348-4845
International Conference on Advanced Computer Science & Software Engineering (ICACSSE)
November 12, 2017 - Hyderabad, India
Paper Published in IJMETMR, A Peer Reviewed Open Access

Such keyword search based techniques can be applied on plaintext search scenarios such as Google search but not on encrypted cloud data [1]. In order to address the above problem a searchable encryption (SE) scheme is introduced for providing efficiency, functionality and security. It develops a list for every keyword and connects the list with the documents having the keyword. By coordinating the keywords inside the data list, feasible keyword search can be acknowledged while both document substance and keyword security are safeguarded. In spite of the fact that considering performing searches safely and feasibly, the current searchable encryption procedure sometimes falls short as they support just correct keyword search. So, there is no chance of minor grammatical errors and configuration irregularities [2].

It is very basic that the clients' searching information may not coordinate with pre-set keywords due to grammatical errors like Illinois and Illinois' and representation irregularities like PO BOX and P.O. Box. In this paper, we considered fuzzy keyword based search which greatly enhances system usability by getting the matching files when users searching queries exactly match with its predefined keywords. In case if exact match fails it matches keyword with closest possible file. By combining fuzzy search with instant search we can improve search experience especially for mobile phone users. Instant search is a visible search system which provides the results immediately based on partial query typed by the user in the search interface[3].

2. RELATED WORK:

Advances in cloud computing and Internet technologies made data owners to outsource their data to remote cloud servers with huge data management services with efficient cost. However, despite its technical advances, cloud computing introduces many new security challenges that need to be addressed well. Data owners maintain sensitive data confidentiality by outsourcing their encrypted data format to un-trusted cloud servers. Several approaches have been provided to enable searching in encrypted data.

However, majority of these approaches are limited to handle single keyword search but not multi keyword ranked search [5]. For providing data privacy to sensitive data it has to be encrypted before outsourcing which makes effective data usage. Although traditional searchable encryption scheme allows users to securely search over encrypted data through keywords and selectively retrieve files of interest such techniques support only exact keyword search. Drawback is there is no tolerance in minor typos and format inconsistencies [1] makes existing techniques unsuitable in cloud computing. A Public-key Encryption with Ranked Multi-Keyword Search (PERMKS) is proposed in which the receiver can query any subset of keywords and the number of queried keywords appearing in the data are provided which evaluate the similarity ranking of data to the search query. Then, a construction of public-key encryption with ranked multi keyword search scheme with sub-linear cipher texts based on Anonymous Hierarchical Identity based Encryption (AHIBE) is put forward and its security is analyzed[6]. In this paper, we formalize and solve the problem of effective fuzzy keyword search over encrypted cloud data while maintaining keyword privacy.

3. EXISTING SYSTEM:

In the existing system, much of the importance is given to fuzzy search on plain text with the help of fuzzy keyword search by many communities. At the starting stage it seems possible for fuzzy keyword search on encrypted cloud data but it fails in data privacy. Even though data files are encrypted, cloud server may try to obtain other sensitive information from users search requests while performing keyword-based search. Thus, the search should be conducted in a secure manner that allows data files to be securely retrieved while revealing little information to the cloud server.

Design Goals:

In this paper, we address the problem of supporting efficient and privacy-preserving fuzzy keyword search services over encrypted cloud data. Specifically, we have the following goals:

ISSN No: 2348-4845
International Conference on Advanced Computer Science & Software Engineering (ICACSSE)
November 12, 2017 - Hyderabad, India
Paper Published in IJMETMR, A Peer Reviewed Open Access

i) To explore new mechanism for constructing storage efficient fuzzy keyword sets;
ii) To design efficient and effective fuzzy search scheme based on the constructed Fuzzy Keyword sets;
iii) To validate the security of the proposed scheme.

4. PROPOSED SYSTEM:

In proposed system we considered a large dataset that cannot be indexed by a single machine is assumed to be partitioned into multiple parts for ensuring its scalability. Each server builds an index structure on its own data parts and is responsible for finding results to a query in its part. The third party on the web server receives a query for each keystroke of a user. Here, the third party is responsible for sending requests to multiple search servers for retrieving and combining results and returns query results to the user. When a search server receives a request, it first identifies all the phrases in the query that are in dictionary D, and intersects their inverted lists. For this we have a module called phrase validator which identifies the phrases called "valid phrases" in the query q that is similar to a term in the dictionary D. For example we considered a hospital database containing patients records in which query q = <cancer, kidney>.

Here "cancer" is a valid phrase. Since, dictionary contains "kidney", "cancer kidney" are also valid phrases. To identify all the valid phrases in a query, the phrase validator uses a tree-based algorithm [7], which can compute all the similar terms efficiently. If a query keyword appears in multiple valid phrases, the query can be segmented into phrases in different ways. Let "|" denote a place between two adjacent valid phrases. For instance, "cancer | kidney" and "cancer, kidney" are two different segmentations for q. The query segmentations that consist of only valid phrases are referred. After identifying the valid phrases, the query plan builder generates a query plan 'Q' which contains all the possible valid segmentations in a specific order[4]. The ranking of Q determines the order in which the segmentations will be executed. After Q is generated, the segmentations are passed into the Index Searcher one by one until the results are computed. The index searcher uses fuzzy keyword search algorithm for computing results. Since the subsequent queries of the user typically share many keywords with previous queries due to incremental typing, it is very important to do computation incrementally and distribute the computational cost of a query between its preceding queries.

Functionality of Fuzzy Keyword Based Search:

In proposed approach as illustrated in Figure-1, V_1 is stored in the cache, vector V2 can be incrementally computed using V1 as follows. The first element of V1 is copied to V2, because Q1 and Q2 share the same first keyword (lines 4–5). Then, the second element of V2 is computed incrementally starting from the active-node sets $S_{2,2}$ and $S_{1,2}$ in the second element of V1 (lines 8–14). The incremental computation from V2 to V3 is an example case where there are additional keywords in the new query. In this case, it copies the first two elements of V2 to V3 since the queries share their first two keywords. We compute the third element of V3 based on the active node sets of the second element of V2 (lines 15–21). In particular, it traverses the tree starting from nodes n_5 and n_6 to see if it contains a term prefix similar to "kidney blood" or "cancer kidney blood", respectively.

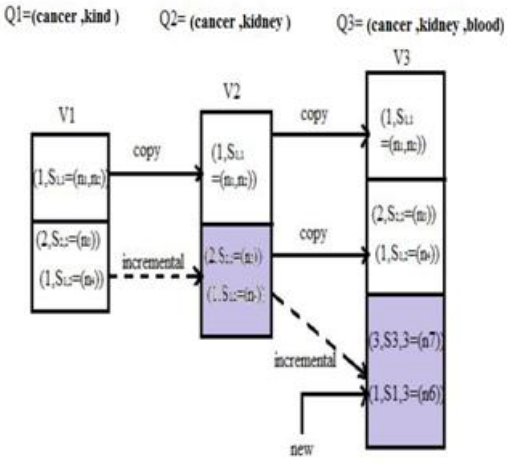

Figure-1: Incremental computation of valid phrase using cache valid phrase

The traversal results in no active node for n_5 and the active node n_8 for n_6. Thus it add the pair (1, $S_{1,3}$ = {n_8}) to the third element of V3, indicating that there is

ISSN No: 2348-4845
International Conference on Advanced Computer Science & Software Engineering (ICACSSE)
November 12, 2017 - Hyderabad, India
Paper Published in IJMETMR, A Peer Reviewed Open Access

a valid phrase starting from the 1st keyword and ending at the 3rd keyword. We also add an element ($3,S_{3,3} = \{n_7\}$) for the 3rd keyword "blood " since it is also a valid phrase with an active node n_7 (lines 22–29).

Algorithm 1: CompValPhrase(q, C)

Input: keywords q=(w1,w2,….wt) where wi is a keyword, & C is Cache module
Output: a valid-Phrase vector **V**;
1. (qc, Vc)<-- FindLongCachePrefix(q, C)
2. M<--no. of keywords in qc
3. **if** m>0 **then** // cache hit
4. **for** i<--1 to m-1 **do** //copy valid phrase vector
5. V[i]<--Vc[i]
6. **if** Wm==qc[m] **then** // qc[m] last word is complete keyword
7. V[m]<--Vc[m]
8. else // computation for last keyword retrieved from C
9. V[m]<--Null
10. **foreach** (start, S) in Vc[m] **do**
11. newS<--find active nodes for wm from S
12. **if** newS== Null **then**
13. V[m]<--V[m]U(start,newS)
14. **foreach**(start, S) in V[m] **do** // compute for partially cached phrase
15. **for** j<--m+1 to l **do**
16. newS<--compute active nodes from S by appending wj
17. **if** newS==Null **then** break
18. V[j]<--V[j]U(start, **newS**)
19. S<-- **newS**
20. **for** i<--m+1 **to** l **do** // computes uncached phrase
21. S<--compute active nodes for Wi
22. V[i]<--V[i]U(i,S)
23. **for** j<--i+1 **to** l **do**
24. **newS**<--compute active nodes from S by appending wj
25. **if newS**==Null **then** break
26. V[j]<--V[j]U(i,**newS**)
27. S<--**newS**
28. Cache(q,V) in C
29. **Return** V

After receiving a list of valid phrases, the query plan builder computes the valid segmentations. For example, for the query q = <cancer, kidney, blood >, "cancer | kidney | blood " is the basic segmentation. In the running example, "cancer kidney" is a valid phrase, and "cancer kidney |blood " is a segmentation. A divide-and-conquer algorithm is used for generating all the segmentations from the valid phrase vector V. Each phrase has a start position and an end position in the query. The start position is stored in V[end] along with its computed active-node set. If there is a segmentation for the query $<w_1,\ldots,w_{start-1}>$, we can append the phrase [start, end] to it to obtain a segmentation for the query $<w_1,\ldots,w_{end}>$. Therefore, to compute all the segmentations for the first j keywords, we can compute all the segmentations for the first i - 1 keywords, where (i, $S_{i,j}$) <--V [j], and append the phrase [i,j] to each of these segmentations to form new segmentations.

5. RESULTS AND DESCRIPTION

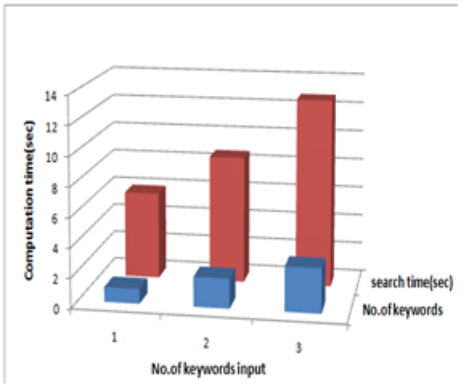

Figure-2: General keyword search on encoded cloud data

As illustrated in figure-2, when the number of keywords increases the computation time also increases because when the query has more keywords the number of phrases to be validated also increases.

ISSN No: 2348-4845
International Conference on Advanced Computer Science & Software Engineering (ICACSSE)
November 12, 2017 - Hyderabad, India
Paper Published in IJMETMR, A Peer Reviewed Open Access

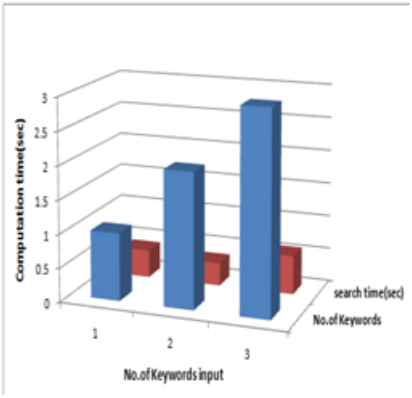

Figure-3: Fuzzy keyword based search on encoded cloud data

The above figure-3 shows that by using fuzzy keyword based algorithm the search time is reduced when number of keywords increases.

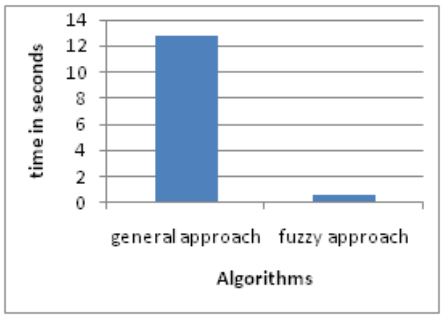

Figure-4: Performance Comparison between general approach and fuzzy approach

As illustrated in figure-4, fuzzy keyword based algorithm improved the efficiency tremendously. For instance, we considered 3-keyword queries resulted in reduction of computation time from 12.77 to 0.57 seconds.

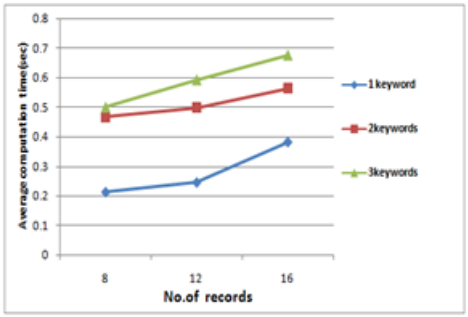

Figure-5: Query execution time for different number of keywords.

The query time & scalability for a data set related to patient health records using fuzzy keyword based search is shown in figure-5. It has been observed that the average computational time is increased linearly when the number of keywords increased.

6. CONCLUSION:

In this paper, we improved security and privacy of encrypted data by using fuzzy keyword based search algorithm to construct storage-efficient fuzzy keyword sets. Based on the constructed fuzzy keyword sets, we further propose an efficient fuzzy keyword search scheme. Our experiment results on real data showed the efficiency of the proposed technique for multi-keyword queries that are common in search applications. It has been concluded that computing all results of multi keyword queries would give the best performance and satisfies the high- efficiency requirement for fuzzy based instant search. Through this particular security analysis, we show that our proposed solution is secure and privacy-preserving while correctly realizing the goal of fuzzy keyword search.

7. REFERENCES:

[1] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, “Fuzzy Keyword Search Over Encrypted Data in Cloud Computing,” Mar. 2010.

[2] B. Wang, S. Yu, W. Lou, and Y. T. Hou, “Privacy-preserving multikeyword fuzzy search over encrypted data in the cloud,” 2014.

[3] R. Curtmola, J. A. Garay, S. Kamara, and R. Ostrovsky, “Searchable symmetric encryption: improved definitions and efficient constructions,” 2006.

[4] Zhonghua Sheng; Zhiqiang Ma; Lin Gu; Ang Li,"A privacy-protecting file system on public cloud storage",2011.

[5] Ibrahim, A.; Hai Jin; Yassin, A.A.; Deqing Zou ,"Secure Rank-Ordered Search of Multi-keyword Trapdoor over Encrypted Cloud Data",2012.

[6] Chengyu Hu; Pengtao Liu ,"Public Key Encryption with Ranked Multi-keyword Search", 2013.

[7] S. Ji, G. Li, C. Li, and J. Feng, "Efficient interactive fuzzy keyword search," 2009.