International Journal of Pure and Applied Mathematics
Volume 116 No. 8 2017, 335-340
ISSN: 1311-8080 (printed version); ISSN: 1314-3395 (on-line version)
url: http://www.ijpam.eu
Special Issue

ijpam.eu

EFFICIENT DATA ACCESS THROUGH SECURE SEARCHING TECHNIQUES IN CLOUD STORAGE

[1]C. Nalini, [2]S. Magesh Kumar, [3]S. Sivasubramanian
CSE Department BIST, BIHER, Bharath university, Tamil Nadu, India
[1]drnalinichidambaram@gmail.com, [2]mageshkumars@yahoo.com
[3]drsivamdu2011@gmail.com

Abstract: Multi-keyword fuzzy search over encrypted data remains yet a challenging problem. The efforts on search over encrypted data involve not only data retrieval techniques such as advanced data structures used to represent the searchable index, and efficient search algorithms that run over the corresponding data structure, but also the proper pattern of cryptographic protocols to ensure the security and privacy of the overall system. Although multi-keyword search and fuzzy search have been implemented separately, a combination of the two does not lead to a secure and efficient multi-keyword fuzzy search scheme. In this paper, a brand new idea for achieving multi-keyword (conjunctive keywords) fuzzy search is proposed, different from existing multi-keyword search schemes; the proposed scheme eliminates the requirement of a predefined keyword dictionary. The fuzziness of the keyword is captured by an innovative data structure and algorithmic pattern without expanding the keyword index and hence exhibits a high efficiency regarding computation and storage. By constructing index indexes using LSH in the Bloom filter, the proposed scheme finds documents with matching keywords efficiently.

Keywords: Locality Sensitive Hashing, Fuzzy Multi Keyword Search

1. Introduction

As the data volume of the Internet is as yet developing, keyword search has turned out to be a standout amongst the most central yet vital capacities. The keyword look issue has been all around contemplated in plaintext situation for a considerable length of time[1-2]. The normal approach is to assemble a keyword index from an accumulation of reports, and the question is performed against the record. As indicated by the quantity of the keywords, inquiries can be classified as single keyword questions and multi keyword inquiries. The later has turned into the prevailing inquiry sort in the present Internet web index. Multi-keyword questions can be additionally arranged into conjunctive and disjunctive inquiries. While the output must contain all the question keyword s for a conjunctive inquiry, a disjunctive inquiry requires no less than one of the question keywords exists in the query item[5-6]. Other essential inquiry capacities incorporate output positioning and rough keyword search otherwise called the Fuzzy keyword search, which quantify the separation between the question keywords and the keywords in the report given likeness estimation[3-4]. On the off chance that the separation is inside an edge, the record is incorporated into the query item. Despite the fact that keyword look through that backings all the previously mentioned capacities have been very much concentrated in plaintext situation, searching over scrambled data is a testing issue. The test originates from two angles, i.e., the security perspective and the usefulness viewpoint.

1.1 Keyword Search

Estimated keyword search also called Fuzzy search has turned out to be one of the basic capacities in the present search engine considering the measure of incorrectly spelled keywords individuals make each day. The search engine should return indexed lists that compare to a legitimate keyword when a search keyword is incorrectly spelled. The procedure is finished by computing the distance between the field keyword s and the search word progressively in plaintext space. Be that as it may, it turns into a specialized test when the data is scrambled on the grounds that the distance between the encrypted keyword is not saved. Research endeavors have been made to address the protection safeguarding Fuzzy search issue over encrypted data. The current arrangements maintain a strategic distance from the distance computation issue by listing all conceivable incorrectly spells as keywords.

1.2 Secure Inverted Index

The altered document is an index data structure that stores the keyword document mapping data. Not the same as the forward index which stores list of words per archive, the modified index comprises of document documents for every keyword in the lexicon. Each archive index contains the IDs of the considerable number of documents that have the keyword. Contrasted and successive Bing emphasis through each document for every keyword utilizing the forward index, the reversed index restrains the pursuit inside a subset of the archives[11-12]. The rearranged index is considered as the focal segment of a commonplace web

search tool ordering computation. Along these lines, the upset index is an essential index structure to control up the present Internet.

As the most prominent search sort, conjunctive multi-keyword query support is basic in functional accessible encryption plans. Current plans don't support index refreshes when changes are made to the fundamental document set, i.e., archive expansion and erasure. Finally, as the cloud may cheat the clients to evade costly computation, the clients ought to have the capacity to check the accuracy of the query item. In this part, dynamic accessible encryption is proposed to conspire in light of the transformed document. Our plan highlights a probabilistic verification process which enables clients to confirm the rightness of the returned result. Contrasted and the current plan, our plan progresses in both protection shrewd and usefulness insightful. Security insightful, our plan ensures the pursuit pattern and in addition the get to pattern through a progression of novel outlines in view of a private set convergence convention. Usefulness insightful[7-8], our plan supports multi-keyword queries and enables the data proprietor to refresh the safe index when the fundamental archive set has been changed. The refresh convention supports archive expansion and erasure while safeguarding the index forward and in reverse protection.

The rest of the chapter as follows, Section 2 condenses the related work for Secure Data Accessing Techniques. Section 3 portrayed problem statements. In section 4[9-10], EDAC (Enhanced Data Access Framework for Cloud) is proposed for efficient data access through secure searching. Section 5 demonstrates result and discussion. Conclusion is presented with Section 6.

2. Related work

In [3] discuss the issue under the symmetric key setting for email. Their plan did not contain a index. In this manner, the search operation experienced the whole document. In [13-14] gave the formal meaning of the accessible encryption and proposed a index structure in light of the upset rundown in [5]. In [1], tackled the outcome positioning issue using the keyword recurrence and request saving encryption

In [6] proposed the main searchable encryption method utilizing the asymmetric encryption conspire. These works just supported the single keyword search over the encrypted data. Different keyword s searchable encryption to advance the search usefulness, the plans supporting conjunctive keywords search have been proposed [15-16]. Many works which supported the conjunctive keyword search, subset search, go queries were utilizing the asymmetric encryption [8] utilized the predicate encryption to accomplish the conjunctive keyword s search over encrypted data.

In [9], a logarithmic-time search method was introduced to help the range queries. In [10] proposed a security protecting multi-keyword positioned search plot utilizing symmetric encryption. In [11] proposed an effective protection safeguarding multi-keyword supporting cosine closeness estimation. Be that as it may, none of the plans can support Fuzzy keyword search.

In [12] proposed a trump card based Fuzzy search method over encrypted data. At that point [13] enhanced the plan by lessening the index estimate. In [14], the LSH capacities are utilized to produce document index. In any case, it took two rounds of correspondence to accomplish comes about positioning and just supported the single keyword search. All the previously mentioned conspires just help the single keyword search, the Fuzzy match OR the correct match.

In [8], presenting a tree structure document and enhanced the search functionality by treating the pre-characterized phrases, for instance, "cloud computing", as a solitary keyword The creators additionally recommend utilizing LSH works as tag-encoding capacities. Despite the fact that this is a simultaneous work, our plan has two noteworthy commitments contrasted with theirs. As a matter of first importance, our approach which uses Bloom channel is not the same as their tag-based plan. Accordingly, we kill there characterized word reference which is vital for their plan. Furthermore, our plan underpins conjunctive keyword search.

The searchable encryption issue is first concentrated in [17-18]. The proposed conspire supports single keyword search without a document which implies the server must sweep the entire archive to play out the keyword search. Subsequent meet-ups on searchable encryption more often than not assemble a protected searchable index to such an extent that specific trapdoors produced by means of mystery keys could coordinate with the document to get the query item while the substance of the index is escaped the cloud.

3. Problem statement

The data utilized as a part of cloud applications is specifically presented to the cloud service provider and could likewise be learned by adversaries as a result of the potential trade off of the cloud. A direct arrangement is to scramble the data before outsourcing it to the cloud. Be that as it may, the cloud is not ready to play out the first computation task over the data that is encrypted utilizing the customary data encryption plans, for example, DES [1] or AES [19-20]. The security challenge originates from the urgent call of secure data encryption conspires that empower different data usage works over the scrambled data. Despite the fact that this thesis is not ready to address

the data security in all cloud computing applications, we concentrate our consideration on a major yet vital data use work, i.e., data Search Function. Keyword search is a standout amongst the most usually utilized data usage works in our everyday life. It has wide applications in data recovery, data mining, and machine learning.

4. Enhanced data access farmework for cloud (edac)

4.1 Fuzzy based Multi-Keyword Search

To provide mean a protected and well running search method over encrypted data, one needs to settle on three vital plan decisions that are nearly between related and to a great extent decide the execution of the subsequent search methods. They are 1) Data structure 2) Effective Search computation 3) Integrated Security and Privacy Mechanisms.

In this subsection, we layout the key thoughts behind our plan for 1) and 2). We will display the key thought of the data structure and search algorithm in the plaintext arrange for simplicity of comprehension. More point by point scheme outline with incorporated security and protection systems, Our plan manufactures document on a for each document premise, in particular, ID for document D. The index ID is a m-bit channel which contains every one of the keyword s in D. To help Fuzzy and multiple keyword searches, we initially change over every keyword into a bigram vector.

Data structure: One key step to building index is keyword change on the grounds that the LSH work does not have any significant bearing to string. We utilize the accompanying strategy to change a keyword to its vector structure. A keyword is first changed to a bigram set. A bigram set of a keyword contains all the adjoining two letters showed up in the keyword. We utilize a 262, 26-bit long vector to speak to a bigram set. The request of the bigrams in the vector is settled for all keywords. Every component in the vector speaks to one of the 262 conceivable bigrams. The component is set to 1 if the relating bigram exists in the bigram set of a given keyword. This bigram vector based keyword portrayal is not delicate to the position of the incorrect spelling, nor is it touchy to which letter it was incorrectly spelled to. In this manner, the distance between the vectors depends on the quantity of the diverse bigrams. The incorrectly spelled keyword will all be mapped to a vector with two-component contrast from the bigram vector of the keyword "PC". By this portrayal, a keyword can be incorrectly spelled in various ways yet at the same time be spoken to in a vector that is near the right one. We measure closeness (remove) utilizing Euclidean distance, the notable distance metric for vector-sort data things. The bigram vector is strong, comprehensive, and the way to empowering the utilization of LSH functions [21].

Effective Search algorithm: Filter has been utilized to fabricate per document index for single keyword correct search situation some time recently. Notwithstanding, these plans utilize consistent hash works in channel so they can't support Fuzzy keyword search. In our plan, we embrace a unique class of hash capacities - territory delicate hash - to construct the index. Since LSH capacities hash contributions between which the comparability is inside a specific edge into a similar yield with high likelihood, our document supports Fuzzy keyword search to as vectors rather than words.

4.2 Integrated Security Mechanisms

The last step of our plan is to discover all the matches and sort them in light of the coordinating score. As appeared in Fig.1, the protected index of each document is a channel that contains every one of the keywords in the document. We create the query vector similarly as producing the document vector. As a matter of first importance, we speak to each of the query keyword s as a bigram vector. At that point we embed the bigram vectors to a channel which has an indistinguishable length from the index and uses the same LSH capacities. Since the query vector and the index channel have a similar length, we would computer be able to the internal result of them. Note that we utilize the same LSH hash works in both the index channel and the search vector. On the off chance that each position of 1s in the search vector likewise coordinates a 1 in the index channel, the query matches with the index. Accordingly, the internal item result is the biggest esteem that it can be.

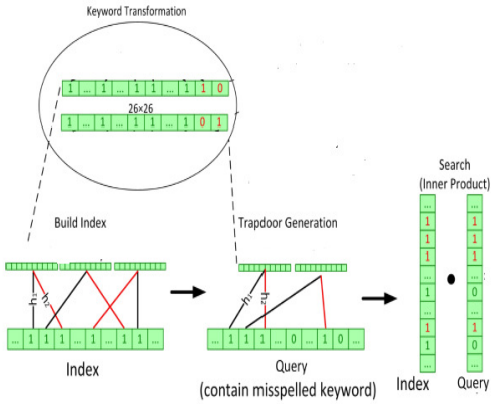

Figure 1. Query support system.

Since a block cipher, for example, AES [4] is utilized to ensure the document content; there are just two routes for the cloud to take in the protection of the indexes and the queries. The first is through the safe index and the trapdoors. The other path is through the

pursuit procedure and the get to pattern which is the output, i.e., the internal item comes about, for our situation. The get to pattern in our plan would release the accompanying data. To start with, it might release the quantity of the keyword s in an search. Since we accept the outline of our plan is openly known aside from the mystery key, the cloud could refers quantity of the keyword s in an search from the internal item comes about. Second, the cloud can interface trapdoors of a similar search by the query items. This is instinctive since a similar query will dependably prompt similar indexed indexes. At last, the indexed indexes likewise uncover the document set that offers similar query keywords.

4.3 Indexed key Based Secure Search.

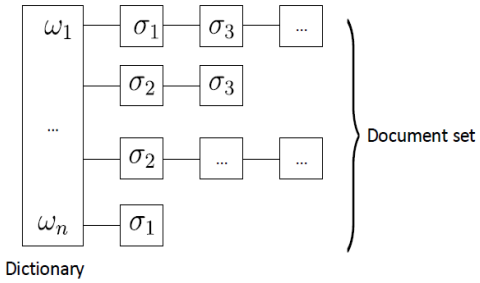

Figure 2. Inverted index illustration.

Index privacy: The index security is twofold. Right off the bat[20-21], the cloud server ought not take in the substance of the document since the substance of the index specifically mirrors the substance of the archives. Furthermore, the cloud server ought to conclude no data about the report through investigating the encrypted index. Such data incorporates 1) regardless of whether a report contains the specific keyword(s), and 2) whether diverse archives contain a typical keyword.

When playing out a pursuit over plaintext, the server coordinates the query keyword(s) to the word reference to find the objective report index(s) first. At that point the server gets the archive competitors by incorporating the document index(s) together. At last, the archive competitors are come back to the client as the item. While searching over encrypted transformed indexes, we need to unravel three testing errands. We initially require a protection safeguarding technique to decide the match between the search keyword(s) and the lexicon. To cover the search pattern, we have to choose the related rearranged documents without telling the cloud server which ones are recovered. At long last, the query item should not specifically uncover the get to pattern, i.e., the document IDs of the output. We address these difficulties through a progression of novel plans in light of the FNP PSI convention.

Document set update: We consider two operations of report refresh, i.e., the archive expansion and the archive erasure. The archive change can be accomplished through a two-platform operation by erasing the first document first and afterward including the altered report as new to the archive set.

Adding: When including an archive, we initially separate every one of the keywords in the report. At that point for every keyword we include into the relating document index, i.e., Recall that each of the report list of our safe index is spoken to as a encrypted polynomial.

Deleting: Deleting a report is to play out a polynomial division. We use the Newton's cycle trap to change over polynomial division to polynomial increase.

5. Results and Discussion

5.1 Accuracy

We embrace the meanings of the broadly utilized execution measurements, exactness and review to quantify the output precision. Indicate tp as genuine positive, fp as false positive and fn as false negative, at that point the exactness equivalents to tp/tp+fp while the review is tp/tp+fn. To produce the Fuzzy queries, we haphazardly pick two keyword s and alter them into the keywords. Figure 3. demonstrates the execution measurements of our plan concurring. Note that there is no review for the correct coordinating in light of the fact that the false negative doesn't exist. One perception is that exactness is low when k is little, i.e. 5% at l = 1. Since that various LSH capacities are utilized together to develop the hole amongst p1 and p2. So when the l is little,

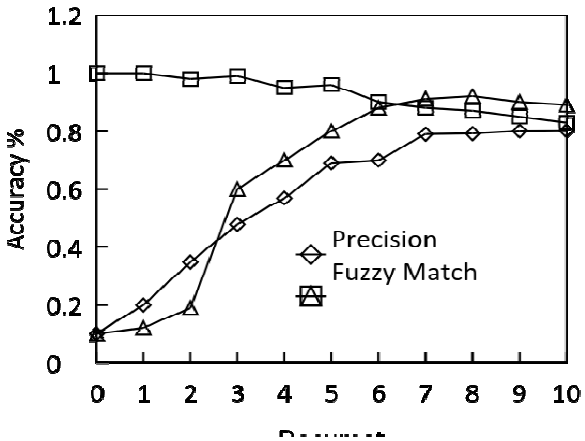

Figure 3. Document Match

the crevice is not sufficiently enormous to recognize the distinctive keyword s, and the vast majority of the files in the dataset have been returned, which prompts

high fp and low fn. The seize l = 5 is because of the hole amongst p1 and p2 increments exponentially regard to l. After a specific l, i.e., l = 8, the accuracy stays at an abnormal state, which is over 90% for the correct scan or more 80% for the Fuzzy search. Another perception is that the review drops while expanding the l. This is on account of that expanding the l will cause all the more false negatives. All in all, the false positive and the false negative can't be enhanced in the meantime. Another imperative parameter is the quantity of the keywords in the search.

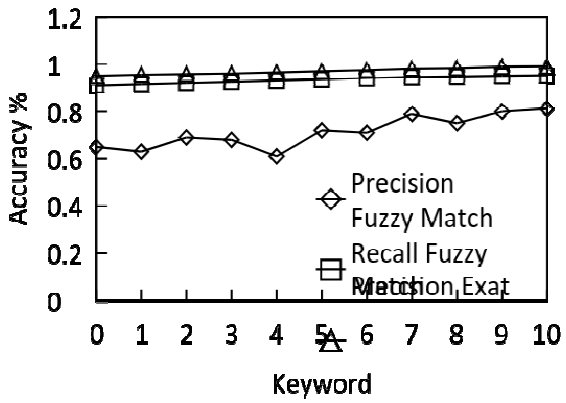

Figure 4. Keyword Match

Figure 4 demonstrates the exactness of the correct match diminishing marginally, from 100% to 96% while the quantity of the keywords in the search increments from 1 to 10. This is sensible in light of the fact that the false positive produced by every keyword collects. In any case, the accuracy for the Fuzzy search doesn't demonstrate a similar example. It is marginally expanded from 70% to 81% when the quantity of the keywords in the search increments from 1 to 10. The reason is that the false positive caused by the LSH capacities contributes considerably more than the false positive presented by the fillter. As the part of the Fuzzy keyword s diminishes, the effect of the false positive caused by the Fuzzy keyword is lessened since the Fuzzy keyword s contribute less in the query output.

6. Conclusions

In this part, we handled the testing multi-keyword Fuzzy search issue over the encrypted data. We proposed and coordinated a few creative plans to explain the multiple keywords search and the Fuzzy pursuit issues all the while with high proficiency. Our approach of utilizing LSH works in the channel to develop the document index is novel and gives a proficient answer for the safe Fuzzy search of various keyword s. Additionally, the Euclidean distance is received to catch the comparability between the keywords, and the protected internal item computation is utilized to ascertain the closeness score in order to empower result positioning. We proposed a fundamental plan and an enhanced plan to meet distinctive security prerequisites. We Compared with the current works, our plan accomplishes more grounded security ensure by lifting the one-time-just search impediment, securing the pursuit pattern and the get to pattern, giving both forward protection and in reverse security for secure index refresh. The proposed method likewise progresses in functionality.

References

[1] W. Sun, B. Wang, N. Cao, M. Li, W. Lou, Y. T. Hou, and H. Li, "Privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking," in Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security, ser. ASIA CCS '13. New York, NY, USA: ACM, 2013, pp. 71-82.

[2] N. Cao, C.Wang, M. Li, K. Ren, and W. Lou, "Privacy-preserving multi-keyword ranked search over encrypted cloud data," in INFOCOM, 2011 Proceedings IEEE, April 2011, pp. 829-837.

[3] M. Naveed, M. Prabhakaran, and C. A. Gunter, "Dynamic searchable encryption via blind storage.", IACR Cryptology ePrint Archive, vol. 2014, p. 219, 2014.

[4] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in Security and Privacy, Proceedings. 2000 IEEE Symposium on, 2000, pp. 44-55.

[5] N. Attrapadung and B. Libert, "Functional encryption for inner product: Achieving constant-size ciphertexts with adaptive security or support for negation," PKC 2010, vol. 6056, pp. 384-402, 2010.

[6]Udayakumar R., Kaliyamurthie K.P., Khanaa, Thooyamani K.P., Data mining a boon: Predictive system for university topper women in academia, World Applied Sciences Journal, v-29, i-14, pp-86-90, 2014.

[7]Kaliyamurthie K.P., Parameswari D., Udayakumar R., QOS aware privacy preserving location monitoring in wireless sensor network, Indian Journal of Science and Technology, v-6, i-SUPPL5, pp-4648-4652, 2013.

[8]Brintha Rajakumari S., Nalini C., An efficient cost model for data storage with horizontal layout in the cloud, Indian Journal of Science and Technology, v-7, i-, pp-45-46, 2014.

[9]BrinthaRajakumari S., Nalini C., An efficient data mining dataset preparation using aggregation in relational database, Indian Journal of Science and Technology, v-7, i-, pp-44-46, 2014.

[10]Khanna V., Mohanta K., Saravanan T., Recovery of link quality degradation in wireless mesh networks, Indian Journal of Science and Technology, v-6, i-SUPPL.6, pp-4837-4843, 2013.