Efficient Similarity Search over Encrypted Data

Mohit Kulkarni[1], Nikhil Kumar[2], Santosh Vaidande[3], B.S.Satpute[4]

[1,2,3] *Student, Dept. of Computer Engineering, Dr. D. Y. Patil Institute of Technology, Maharashtra, India.*
[4] *Professor, Dept. of Computer Engineering, Dr. D. Y. Patil Institute of Technology, Maharashtra, India.*

Abstract – *In this present time, due to engaging components of appropriated processing, the colossal measure of data has been secured in the cloud. In spite of the way that cloud-based organizations offer many favorable circumstances yet assurance and security of the delicate data is a noteworthy issue. These issues are settled by securing sensitive data fit as a fiddle. Encoded limit secures the data against unapproved get to, yet it cripples some basic and basic handiness like interest operation on the data, i.e. looking the required data by the customer on the mixed data obliges data to be unscrambled first and a short time later look for, so this at last, backs off the route toward looking for. To achieve this various encryption arranges have been proposed, regardless, most of the arrangements handle amend Query organizing however not Similarity planning. While customer exchanges the record, components are expelled from each report. Right when the customer fires a request, trapdoor of that question is created and interest is performed by finding the relationship among files set away on cloud and request watchword, using Locality Sensitive Hashing.*

***Key Words* -** Locality sensitive hashing, Encrypted data, Similarity search, Cloud computing, AES Encryption, Trapdoor Construction

1. INTRODUCTION

In present information far reaching condition, cloud computing is normal since it expels the weight of gigantic information administration in a savvy way. The touchy information send to non-trusted cloud servers will prompt protection issues. To reduce the stresses, delicate information must be sent in the encoded shape which maintains a strategic distance from unlawful get to. There are numerous calculations which bolster the operation which is called Searchable Encryption Scheme.

Ordinarily, all plans are intended for correct question coordinating. Inquiry coordinating is the way toward finding the client's coveted information from the cloud as indicated by the expressed element. In any case, it is more sensible to perform recovery as per the closeness with the expressed element rather than the presence of it. Comparability hunt is an issue that upgrades and finds the point in a given set that is nearest to a given point. It is predefined that efficient techniques are required to do closeness seek over the gigantic measure of encoded information. The fundamental calculation of our venture is the rough close neighbor seeks calculation called Locality Sensitive Hashing (LSH). LSH is comprehensively utilized for quick likeness look on information in data recovery. Our venture has two sections: User 'Transferring the information' and 'Seeking the question'.

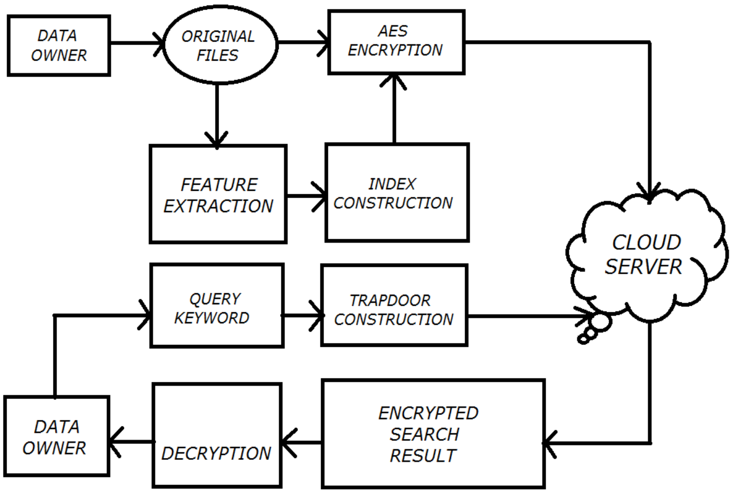

Fig 1: Complete Process

Right off the bat, the client transfers the information. At that point record components are extricated by preprocessing it. Stop words expulsion and Stemming id is the following procedure. The words are part at the specific character to shape pails and proceed till the aggregate length of the word after which they are put away in a document. After that Encryption of both Plaintext and the Indexed document is done and put away in the cloud.

2. LITERATURE SURVEY

Cong Wang et al. [1] presents secured rank watch word scan for information put away in the cloud. Right off the bat, they positioned the watch words utilizing compelling positioning compositions while keeping up security over the cloud and effectiveness of information looked. The positioning is done in view of file development and utilizations positioning capacity to give a score to watchwords. The significant issue in this is computational overhead to rank catchphrases.

Cheng and Mitzenmacher [2] offer to seek on encoded information put away remotely utilizing catchphrases that protect the security of information. Here they portray recovering scrambled information utilizing the watchword record (i.e. lexicon) which is made by the client itself. The word reference is available on the remote server including the record, utilizing which the client can recover the scrambled information while looking after security. It requires additional capacity to store watchword record.

Jan Li et al. [3] created plans to recover the scrambled information on the cloud utilizing fluffy catchphrase look while protecting the security and productivity. They have created two plans (trump card based method and gram-based strategy) to develop fluffy catchphrase set which produces coordinating records or shut coordinating documents. The proprietor stores fluffy catchphrases set alongside the record which is changed over into file frame. The proficiency of proposed framework can even now be additionally enhanced for achieving better conceivable outcomes.

Qin Lv [4] expounds on multi test LHS composition for ordering which helps effective closeness coordinating for looked inquiry and delivers most ideal outcomes. It utilizes KNN-calculation for coordinating records from numerous pails for a given information question by the client. In this way it requires few hash tables because of examining different basins and spares storage room required to store countless tables. They even looked at entropy based LSH and multi-test LSH demonstrating the upsides of multi-test LSH over entropy based LSH.

Dan Boneh and Brent Waters [5] offers people in general key framework that backings correlation questions on encoded information and additionally more broad inquiries, for example, subset inquiries. These frameworks bolster conjunctive questions which are discretionary in nature without spilling data on individual conjuncts. Also, a general structure for developing and breaking down open key frameworks supporting queries on scrambled information. Openly key frameworks, a mystery key can create tokens for testing any upheld inquiry predicate. The token gives anybody a chance to test the predicate on a given figure content without adapting some other data about the plaintext. It speaks to the general structure to dissect the security of looking on scrambled information frameworks.

Mehmet Kuzu [6] proposes an approach which utilizes Locality Sensitive Hashing (LSH) which is the closest neighbor calculation for the file creation. Comparative elements are put into a similar basin with a high likelihood because of the property of LSH while the elements which are not comparative are kept in the distinctive pails. On the off chance that the informational index is little, then the correspondence cost and pursuit time required by this plan is better. Yet, in the event that the database measure expands, then the time required for correspondence and for looking additionally increments quickly. They utilize sprout channel for interpretation of strings. However, the disservice of this structure is that it is a probabilistic information structure.

Wenjun Lu et al. [7] offer's a method to retrieve encrypted multimedia file without decrypting it from cloud while maintaining the confidentiality and privacy of data. They have used different encryption algorithms, secured inverted index and secure min-hash algorithm to achieve the goals. Firstly, the features of image are extracted and index is build and then the image is encrypted and stored in remote server. The user can retrieve the image by using different queries. The efficiency and security of given system can be further improved.

Bing Wang et al. [8] introduces about privacy preserving multi-keyword fuzzy search over encrypted data which is stored on the cloud. Here the author has used different encryption and hashing techniques to maintain privacy of data over cloud. The given method eliminates the need of dictionary.

3. PROPOSED METHODOLOGY

Step 1: Uploading of Data

Initially data in text file format is been uploaded by data owner. Subsequently three sub process occurs. Initially original data is been encrypted using AES and pattern buckets are stored in Encrypted format in cloud. Secondly data is been preprocessed, matrix Translation is been done creating buckets, data is been stored in as feature data in private cloud. An original copy of data is been stored as complete data in other cloud.

Step 2: Input Query preprocessing

Input Query is been submitted by Data user to System. System Accepts query performing preprocessing on query. Future query is been Trapdoor to create Query words pattern in encrypted format. Every word is been sent matrix translation creating buckets, this buckets are future encrypted using AES Algorithm. Here in This process pattern are been generated using bucket generation. A bucket is set of word list for every word starting from trigram to N gram Approach.

Bucket generation Process is as below:

“Computing” generates bucket as shown below-

“Computing”➔{ com,comp,compu,comput,computi,computi,computin}

Step 3: Bloom Filter Application

Data in cloud Stored as feature data is been dynamically copied in feature vector consisting of File name and File Feature. This complete Data is termed as Bloom Filter Data.

Step 4: Search correlation

Search correlation Pattern matching is been done with Input Query and Trapdoor query. Both vectors are been compared and Correlation vector for input query is been generated. This vector is been sent for future correlation evaluation.

Step 5: Ginix index

Search results are been displayed in all files that have been matched for given query input

4. ALGORITHM

ALGORITHM 1: BUILD MATRIX

```
Require: D: data item collection,
Ψ : security parameter,
MAX: maximum possible number of features
$K_{id} \leftarrow$ Keygen(Ψ), $K_{payload} \leftarrow$ Keygen(Ψ)
for all Di € D do
Fi ← extract features of Di
for all fij € Fi do
fij ← apply metric space translation on fij
for all gk € g do
if gk(fij ) € bucket identifier list then
add gk(fij ) to the bucket identifier list
Initialize Vgk(fij) as a zero vector of size |D|
Increment recordCount
end if
$V_{gk}(f_{ij})[id(D_i)] \leftarrow 1$
end for
end for
end for
for all $B_k$ € bucket identifier list do
$V_{Bk} \leftarrow$ retrieve payload of $B_k$
$\pi_{Bk} \leftarrow Enc_{Kid}(B_k)$, $\sigma_{VBk} \leftarrow Enc_{Kpayload}(V_{Bk})$
add ($\pi_{Bk}$ , $\sigma V_{Bk}$) to I
end for
return I
```

ALGORITHM 2: MATRIX SPACE TRANSLATION

```
//Input : Data collection Set D ={ $D_i$ }
//Output: Matrix space Set MS
Step 0:Start
Step 1: Get the Set D
Step 2: FOR i=0 to Size of D
Step 3: get $S_i$ of Di
Step 4: FOR j=0 to length of si
Step 5: $sb_i$=substring($s_i$,2→j)
Step 6: Add $sb_i$ to MS
Step 7: END FOR
Step 8: END FOR
Step 9: return MS
Step 10: Stop
```

5. RESULTS

Some trial assessments are performed to demonstrate the adequacy of the framework. What's more, these analyses are led on windows based java machine with generally utilized IDE Net beans. Likewise the quantities of recovered records are utilized to set benchmark for execution assessment. Quantities of applicable recovered archives from the cloud for the arrangement of catchphrases are utilized to demonstrate the adequacy of the framework. The following are the meaning of the utilized measuring methods i.e. exactness and review.

Exactness: it is a proportion of quantities of appropriate archives recovered to the total of aggregate quantities of important and immaterial records recovered. Relative viability of the framework is very much communicated by utilizing accuracy parameters.

Review: it is a proportion of aggregate quantities of applicable archives recovered to the aggregate quantities of pertinent records not recovered. Supreme precision of the framework is all around described by utilizing review parameter.

Quantities of situations presents where one measuring parameter rules the other. By thinking about such parameters we utilized two measuring parameters, for example, accuracy and review.

For Detailed Examination

- A = No. of relevant docs retrieved
- B =No of relevant documents not retrieved
- C = No of irrelevant documents are retrieved.

So, Precision = (A/ (A+ C))*100

And Recall = (A/ (A+ B))*100

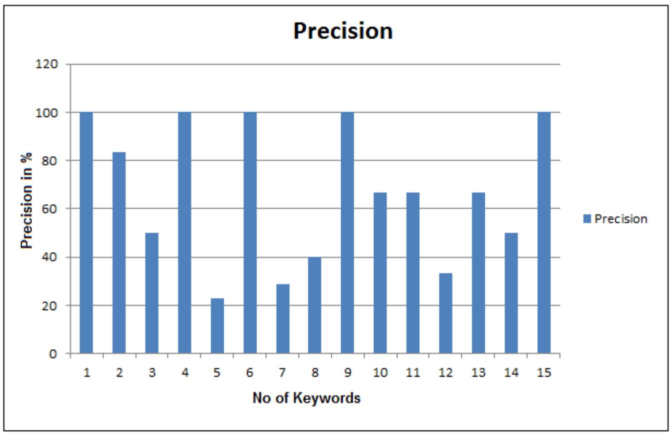

Fig.3.Average precision of the similarity search method

In Fig. 3, by observing figure 3 it is clear that average precision obtained by using similarity search method is approximately 65%.

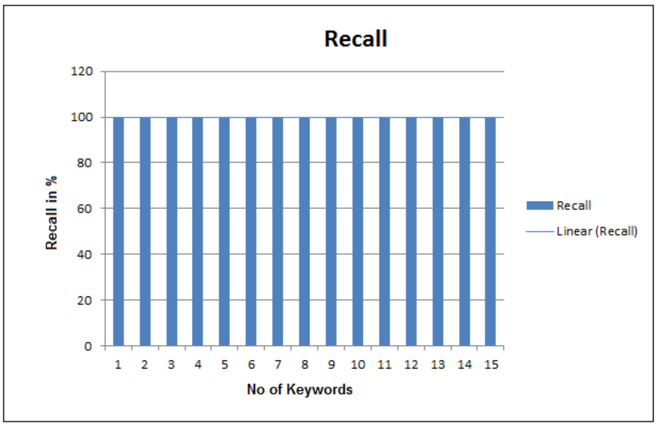

Fig.4. Average Recall of the similarity search method

In Fig. 4, figure shows that the system gives 95% recall for the similarity search method. By comparing these two graphs we can conclude that the similarity search method gives high recall value compare to the precision value.

6. CONCLUSION

Search over encrypted System assist in retrieving file that contains required information without decryption process. This would enhance search process in cloud where files are b been encrypted and stored. In above project work search time has been found to be low. Effective retrieval of documents measuring precision and recall.

REFERENCES

[1] Cong Wang, Ning Cao, Jin Li, Kui Ren and Wenjing Lou, “Secure Ranked Keyword Search over Encrypted Cloud Data”, IEEE 30th International Conference on Distributed Computing Systems, 2010, pp. 253-262, doi:10.1109/ICDCS.2010.34.

[2] Yan-Cheng Chang and Michael Mitzenmacher, “Privacy Preserving Keyword Searches on Remote Encrypted Data”, in Proc. of ACNS’05, 2005, pp. 442-455, Springer Berlin Heidelberg.

[3] Jan Li, Q. Wang, C. Wang, N. Cao, K. Ren, W. Lou, “Enabling Efficient Fuzzy Keyword Search over Encrypted Data in Cloud Computing”, Proc. of IEEE INFOCOM’10 Mini-Conference, March 2010, pp. 1-5, IEEE.

[4] Qin Lv, William Josephson, Zhe Wang, Moses Charikar, Kai Li, “Multi-Probe LSH: Efficient Indexing for High Dimensional Similarity Search”, in Proceedings of the 33rd international conference on very large databases, September 2007, pp. 950-961, VLDB Endowment.

[5] Dan Boneh and Brent Waters, “Conjunctive, Subset and Range Queries on Encrypted Data”, in Theory of Cryptography Conference, February 2007, pp. 535-554, Springer Berlin Heidelberg.

[6] Kuzu Mehmet, Mohammad Saiful Islam, Murat Kantarcioglu, “Efficient Similarity Search Over Encrypted Data”, in Data Engineering (ICDE), 2012 IEEE28th International Conference, IEEE, 2012.

[7] Wenjun Lu, Ashwin Swami Nathan, Avinash L. Varna, and Min Wu, “Enabling search over encrypted multimedia databases” InIS&T/SPIE Electronic Imaging, Feb 2009 , pp. 725418-725418, International Society for Optics and Photonics.

[8] Bing Wang, Shucheng Yu, Wenjing Lou, Y. Thomas Hou, “Privacy-preserving multi-keyword fuzzy search over encrypted data in the cloud” InINFOCOM, 2014 Proceedings IEEE, April 2014, pp. 2112-2120, IEEE.