A Privacy And Dynamic Multi-Keywords Ranked Search Scheme Over Cloud Data Encrypted

G. KAVITHA
PG Scholar, Dept of CSE, Indira Institute of Technology & Science,Markapur, Prakasam Dist, AP, India.

K SURENDRA REDDY
Associate Professor, Dept of CSE , Indira Institute of Technology & Science,Markapur, Prakasam Dist, AP, India.

Abstract: **Cloud computing is emergent discipline as a new computing model in a number of trade domains. Enormous numbers of colossal-scale businesses are starting to shift the data on to the cloud environment. The proposed multi key word search based on rating over encrypted cloud data uses feature of similarity and inner product similarity matching. The vector space model helps to provide enough search accuracy and homomorphic encryption permits clients to contain in ranking whereas majority of computing work is done on server part by using operations only on cipher textual content. As a consequence in this system for top-k retrieval user gets an interested/used link in top.**

Keywords: **Searchable Encryption; Multi-Keyword Ranked Search; Inner Product;**

I. INTRODUCTION

Cloud computing is the wide imagined dream of computing as a service, the place data owners of cloud can remotely store their sensitive and personal data like emails, health files and personal photos onto the cloud. With the exceptional functions, we will get services on-demand from a shared environment with computing resources from anyplace on this cloud. It is a good tractability and cost saving to motivating both industry and individual to outsource their regional confidential elaborate data management method onto the cloud. The brand new cloud-computing paradigm has generated a vast curiosity in both enterprise and academia, resulting in a number of noted marketable and individual cloud computing services supplied through several cloud service providers, examples from Amazon internet services, Google App engine, Microsoft Azure, Aneka, Yahoo, and earnings drive. Top database vendors similar to Oracle and IBM with IBM BlueMix are including cloud sustenance to their databases.

To protect data privateness, confidentiality, and data safety, sensitive data like personal health records, e-mails, tax documents, picture albums, economic transactions, etc, need to be encrypted by way of data owners earlier than outsourcing to the general public cloud [1]. However, the average plaintext keyword search data utilization carrier is out of date. Downloading all of the data and decrypting at the data user part is trivially impractical duo to huge quantity of bandwidth cost is required in cloud scale techniques. The data can be readily searched and utilized or else no cause of storing data in the cloud. Therefore, exploring robust and privacy preserving search over encrypted cloud data is of most prominence. This is an extraordinarily difficult drawback; it degrades efficiency of system usability and scalability. It is very complicated to fulfil the standards of approach usability, performance, and scalability, through due to the fact the huge number of on-demand data customers and huge quantity of outsourced data records in the cloud. To meet powerful data retrieval, the massive amount of records demands the cloud server to participate in relevance rating consequently, as a substitute returning all outcome files. Such ranking procedure enables data users to seek out probably the most significant data quickly, instead than burdensome sorting by means of each suit in the content collection [2]. Ranked search eliminates the unnecessary network visitors by means of sending simplest most central files, which may be very so much fascinating in “pay-as-you-use” cloud model. Such ranked programs will have to no longer leak any keyword related understanding to the cloud server for privateness safety. Such rating programs should support fuzzy key word and multi-key phrase search, as a single keyword search often offers a long way too severe outcome.

II. RELATED WORK

Dynamic Multi-Keyword Ranked Searchable Security Algorithm Using CRSA and B-Tree Prasanna B T#1, C B Akki*2 #Department of ISE, EPCET Associate Professor, Bengaluru, INDIA-560049 *Department of ISE,SJBIT Professor, Bengaluru, INDIA-560060

With the abilities of storage as a provider many organizations are relocating their useful data to the cloud, in view that it costs less, quite simply scalable and can be accessed from wherever any time. The reliance between cloud user and provider is vital. We use security as a parameter to set up trust. Cryptography is one way of establishing trust. Searchable encryption is a cryptographic approach to furnish security. In literature many researchers have been engaged on establishing effective searchable encryption schemes. In this paper we explore probably the most mighty cryptographic methods based on data constructions like CRSA and B-Tree to enhance the level of security, thus trust. We tried to put in force the search on encrypted data making use of Azure cloud platform.

Reusability of Search Index over Encrypted Cloud Data on Dynamic update Kavitha R1, R J Poovaraghan2 Student, M.Tech, SRM University, Chennai, India1 Assistant Professor (OG), Department of Computer Science, SRM University, Chennai, India

Cloud computing is generating lot of interest to provide answer for data outsourcing and high great data offerings. Increasingly institutions, organizations and firms are exploring the likelihood of having their applications, information and their IT property in cloud. As the data and there through the cloud's dimension increases looking of the significant data is expected to be a mission. To overcome this task, search index is created to aid in turbo search. However, search Index creation and computation has been difficult and time ingesting, main to cloud-down time there by way of encumbering the quickness in reacting to data request for mission valuable requisites. Focal point of this paper is to explain how within the proposed method, reusability of search index helps to reduce the complexity of search index computation. Search index is proposed to be created utilising parameters like similarity relevance, user rating and scheme robustness. User rating helps to guarantee why a phrase or a sentence or a key phrase is used most commonly within the uploaded data. The proposed approach ensures that the reusability of search index proposal, incredibly reduces cloud down time while keeping the safety utilizing searchable symmetric encryption (SSE).The user requested file is retrieved from the cloud, making use of Two-round searchable encryption (TRSE) scheme that supports top-k-multi-key phrase retrieval.

Dynamic Multi-keyword Top-k Ranked Search over Encrypted Cloud Data Xingming Sun, Xinhui Wang, Zhihua Xia, Zhangjie Fu and Tao Li Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science & Technology, Nanjing, 210044, China sunnudt@163.com, wxh_nuist@163.com, xia_zhihua@163.com, wwwfzj@126.com.

These days, increasingly humans are influenced to outsource their neighbourhood data to public cloud servers for great comfort and decreased expenditures in data management. But in consideration of privacy issues, sensitive data will have to be encrypted earlier than outsourcing, which obsoletes normal data utilization like keyword-based file retrieval. In this paper, they propose a comfortable and efficient multi-keyword ranked search scheme over encrypted data, which additionally supports dynamic update operations like deletion and insertion of files. Principally, we construct an index tree originated on vector space model to provide multi-keyword search, which meanwhile helps flexible replace operations. Apart from, cosine similarity measure is utilized to support accurate ranking for search influence. To reinforce search effectivity, we further propose a search algorithm based on "greedy Depth-first Traverse process". Furthermore, to shield the search privateness, we endorse a relaxed scheme to meet various privacy standards in the known ciphertext danger model. Experiments on the actual-keyword dataset show the effectiveness and efficiency of proposed scheme.

III. PROPOSED METHODOLOGY

The proposed comfortable multikeyword top-k retrieval over encrypted cloud data make use of the methods of expertise retrieval group i.e. Vector space model and cryptographic group i.e. Homomorphic encryption. In proposed approach to make certain protection and efficiency lots of the work is finished by means of cloud, but ranking is left to the cloud user. As most users need the files which can be most important to them, in top-k-multi keyword retrieval the cloud calculate the ranking for each and every file as per the keyword phrases within the query.Term frequency-inverse record frequency weighing scheme is used to assign rating for every file and cosine similarity to search out the similarity. Cloud server then returns the calculated ratings to the data user and data user send again the top-k ratings to the cloud server. Consequently there's a two round conversation between cloud server and data user, as a consequence referred to as Two round Searchable Symmetric Encryption Scheme [TRSE].

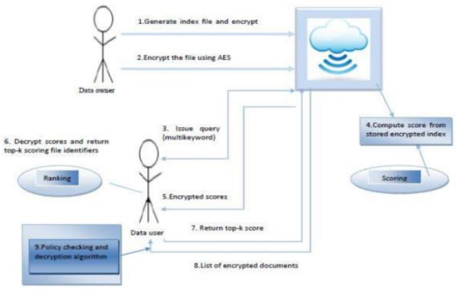

Fig.1. Architecture of the search over encrypted cloud data

Preprocessing Module: Three distinct entities are concerned in cloud computing environment: Cloud server, data owner and data user. Data owner at first has to register with the cloud. After that the user has to wait except the administrator approves him. The data owner can add the documents. Third party data storage and retrieval services are hosted by way of the cloud server. Because the uploaded data could contain sensitive, data is required to be encrypted before outsourcing. The admin also assigns a group to the data users.

Searchable Index Module: To with no trouble search documents, data owner has to construct a searchable relaxed index for the entire documents being uploaded into the cloud. The entire terms important to the document is extracted, restricted and calculated. The document together with the index is then uploaded into the cloud.

Encrypt Module: The documents earlier than uploading into the global space are encrypted using any encryption scheme, whereas for the comfortable index homomorphic encryption is applied. Homomorphic encryption maps the operations in cipher textual content and simple text domain. The additive and multiplicative property of homomorphic encryption is utilized right here. For that reason Pailier homomorphic encryption is utilized to the index construct. Homomorphic keys must be handled properly.

Processing Module: Processing module deals with how to get correct search outcome headquartered on the more than one key phrase queries. The customers can enter the multiple words query in the encrypted type; the server stems that query into single words. Then find the term frequency (tf) and inverse document frequency (idf) for each term.

Question ranking Module: Score for each and every document is calculated based on the tf-idf worth. The number of files to be retrieved is targeted with the aid of the user. The query issued with the aid of the user as good as all of the documents are represented in the form of vector. Each dimension of the vector represents the presence of the specified key word. If a distinct term is reward then its value is 1, or else the worth of that vector is ready to zero. To rank records cosine similarity between person question and all documents is discovered. After ranking prime-k rankings are again to the user.

Log File generation: For each user action a log file is generated on the cloud server. Log data is utilized by cloud admin to get all data related to file importing and downloading. Data owners on the other hand use this log file to get data about file‟s download.

k-nearest neighbor

k-nearest neighbor search identifies the top k nearest neighbors to the query. This technique is often utilized in predictive analytics to estimate or classify a factor customary on the agreement of its neighbors. k-nearest neighbor graphs are graphs in which each point is hooked up to its k nearest neighbors. The elemental proposal of our new algorithm: the value of dmax is lowered keeping step with the continued detailed evaluation of the thing similarity distance for the candidates. On the finish of the step-by-step refinement, d_{max} reaches the most reliable query range E_d and prevents the procedure from producing extra candidates than fundamental thus gratifying the r optimality criterion.

Nearest Neighbor Search (q, k) // optimal algorithm

1. Initialize ranking = index.increm-ranking (F(q),df)

2. Initialize result = new sorted-list (key, object)

3. Initialize dmax = w

4. While o = ranking.getnext and d,(o, q) I d,,, do

5. If do@, s> s dmax then result.insert (d,(o, q) , o)

6. If result.length 2 k then d_{max} = result[k].key

7. Remove all entries from result where key > d_{max}

8. End while

Report all entries from result where key I d_{max}

Homomorphic Encryption Scheme

Encryption scheme that presents security at both user side and on server side is required. Moreover best addition and multiplication operations over integers are wanted therefore of utilising the vector space model and relevance scoring, as a result homomorphic encryption is employed here. Homomorphic encryption maps certain operations in cipher text to that to straightforward textual content. Pailier cryptosystem shows additive homomorphic property. A Homomorphic encryption is additive,

Enc $(x \oplus y) = Enc(x) \otimes Enc(y)$

Enc $(\Sigma\ mi) = \Pi\ Enc\ (mi)$

i=1 i=1

Suppose C1 and C2 be the two cipher text such that:

C1 = gm1. r1n mod n2

C2 = gm2. r2n mod n2

Then,

C1.C 2 = gm1. r1n gm2. r2n mod n2 = gm1+ m2 (r1 r2)n mod n2

An example for additive homomorphic property is shown below:

Let (n, g) = (2501, 92)

m1 = 34 and m2 = 16

C1 = 9234 5 2501 mod n2 = 1129735 (r1 =5)

C2 = 9216 7 2501 mod n2 = 5140305 (r2 =7)

C1 . C2 = 2010769

m1 + m2 =50

C 1+2 =9250 . 352501 mod n2 = 2010769

Thus Paillier cryptosystem shows the property of additive Homomorphic encryption. Electronic voting is an example of additive Homomorphic encryption.

Scoring Scheme

To weight the relevance of a file with admire to a key phrase the easiest manner is scoring. For scoring exceptional rating models exists; of these most customary is TF-IDF weighing scheme. Many variations of the tf-idf exist. Term frequency and Inverse file frequency are the 2 attributes of this scoring scheme. How mostly a targeted key phrase occurs in a report are outlined by using term Frequency (tf) whereas in how many record a detailed key phrase exists is defined by record frequency (df). The inverse report frequency is calculated as follows:

$Id_f = \log [N/ d_f]$, where N is the number of files.

Then, by using tf-idf weighting scheme a term t in file f is given a score as:

$tf - idf_{t,f} = tf_{t,f} X\ idf_t$

Vector Space Model

To find the similarity of queried keyword with the existing documents, vector space model is used. In

vector space model both documents and query is represented in the form of vector. For example consider the

following two texts:

Text 1: Ram loves me more than Krishna loves me

Text 2: Lakshmana likes me more than Ram loves me

The keywords that occur in this are: me Ram loves Krishna than more likes Lakshmana

Then count the number of times each keyword exists in the text as:

me 2 2

Ram 1 1

likes 0 1

loves 2 1

Lakshmana 0 1

Krishna 1 0

than 1 1

more 1 1

The two vectors that correspond to the text are:

a: [2, 1, 0, 2, 0, 1, 1, 1]

b: [2, 1, 1, 1, 1, 0, 1, 1]

Here an 8 dimensional vector is formed. Cosine similarity is further used to find the similarity.

Cosine Similarity

To rank the documents different ranking techniques are available. Here to retrieve the top-K documents matching the user‟s interest, cosine similarity is used. An example for calculating cosine similarity is given below:

Consider a small collection of documents, consisting the following three documents:

document1: “very good times”

document2: “very good post”

document3: “los angeles times”

Step1: Calculating Term frequency

For all the documents, then calculate the tf scores for all the terms in C. Assign the score 1 if the keyword appear in that particular document, otherwise assign 0:

Step2: Inverse Document Frequency

The total number of documents is N=3. Therefore, the idf values for the terms are:

angles log2(3/1)=1.584

los $\log_2(3/1)=1.584$

very $\log_2(3/2)=0.584$

post $\log_2(3/1)=1.584$

times $\log_2(3/2)=0.584$

good $\log_2(3/2)=0.584$

Step 3: TF x IDF

Then multiply the tf scores by the idf values of each term, obtaining the following matrix of documentsby-terms:

Step 4: Vector Space Model And Cosine Similarity

Let the query given by the user be: “very very times”, calculate the tf-idf vector for the query, and compute the score of each document in C relative to this query, using the cosine similarity. When computing the tf-idf values for the query terms, divide the frequency by the maximum frequency (2) and multiply with the idf values. Using the formula given below we can find out the similarity between any two documents. The query entered by the user can also be represented as a vector

q [0 0 (2/2)*0.584 0 (1/2)*0.584=0.292 0]

Calculate the length of each document and of the query:

Length of d1 = sqrt(0.584^2+0.584^2+0.584^2)=1.011

Length of d2 = sqrt(0.584^2+1.584^2+0.584^2)=1.786

Length of d3 = sqrt(1.584^2+1.584^2+0.584^2)=2.316

Length of q = sqrt(0.584^2+0.292^2)=0.652

Then the similarity values are:

cosSim(d1,q) = (0*0+0*0+0.584*0.584+0*0+0.584*0.292+0.584*0) / (1.011*0.652) = 0.776

cosSim(d2,q) = (0*0+0*0+0.584*0.584+1.584*0+0*0.292+0.584*0) / (1.786*0.652) = 0.292

cosSim(d3,q) = (1.584*0+1.584*0+0*0.584+0*0+0.584*0.292+0*0) / (2.316*0.652) = 0.112

According to the similarity values, the final order in which the documents are presented as result to the query will be: d1, d2, d3.

IV. CONCLUSIONS

Our Proposed a scheme [TRSE Scheme] which allows data owners to add encrypted data documents into cloud storage and permit a few authorized clients to perform operations corresponding to uploading, search, share and retrieval over them. As a substitute of the entire files, proposed approach enable users to receive the result with essentially the most important records that healthy customers' requirement. Records of easiest relevance are only sending back to the user. In TRSE, the concept of homomorphic encryption, relevance scoring, vector space model and key management are used. When we consider that the search operation is performed over encrypted data, different leakage can also be eliminated and data can be searched and retrieved efficaciously. TRSE scheme helps a couple of data owners and their related set of clients. Separate space for each data owner is maintained at storage portion. To facilitate search on encrypted index homomorphic encryption is used.

V. REFERENCES

[1] S. Kamara and K. Lauter, “Cryptographic Cloud Storage,” Proc. 14th Int’l Conf. Financial Cryptography and Data Security, Jan. 2010.

[2] Zhihua Xia, Xinhui Wang, Xingming Sun, and Qian Wang, “A Secure and Dynamic Multi-keyword Ranked Search Scheme over Encrypted Cloud Data,” IEEE Transactions on Parallel and Distributed Systems, 2015.

[3] R. Curtmola, "Searchable symmetric encryption: improved definitions and efficient constructions",Proceedings of the 13th ACM conference on Computer and communications security, ACM, (2006).

[4] D.X.Song, D. Wagner and A.Perrig,"Practical techniques for searches on encrypted data. in Security and Privacy", 2000. S&P 2000, Proceedings 2000 IEEE Symposium, IEEE, (2000).

[5] C. Wang, "Secure ranked keyword search over encrypted cloud data”, Distributed Computing Systems (ICDCS), 2010 IEEE 30th International Conference, IEEE, (2010).

[6] C. Wang, N. Cao, K. Ren, and W. Lou, “Enabling secure and efficient ranked keyword search over outsourced cloud data,” Parallel and Distributed Systems, IEEE Transactions on, vol. 23, no. 8, pp. 1467–1479, 2012.

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

[8] 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. ACM, 2013, pp. 71–82.

AUTHOR BIOGRAPHY

G.Kavitha is pursuing her M-Tech in Dept of CSE, Indira Institute of Technology & Science,Markapur, Prakasam Dist, Affiliated to JNTUK University.

K Surendra Reddy has 8 years of experience in Teaching. Currently working as Associate Professor in Dept of CSE, Indira Institute of Technology & Science,Markapur, Prakasam Dist, Affiliated to JNTUK University.

