A Privacy-Preserved & Encrypted Multi-Keyword Ranked Search in Cloud Storage

Swati S. Deshmukh[1], Prof. Shrikant V. Sonekar[2]

Department of Computer Science & Engineering
[1]M.Tech Student,J. D. College Of Engineering & Management
[2]Assistant Professor,J. D. College Of Engineering & Management

Abstract-*As of late, development of private and semi-private data has grown up quickly on data arrange; instruments to pursuit such data have bombed in security safeguarding. The security saving seeking is assuming imperative part in the field of data systems to perform different information mining operations on encoded information put away in different stockpiling frameworks. It is additionally vital and testing undertaking to secure the secrecy of private information shared among specialist co-ops and information proprietors. Existing framework gives one conceivable arrangement that is protection safeguarding ordering (PPI). In this framework, archives are put away in plain content shape on private server that is security is traded off. So to improve this framework to make it more secure and effective, first we store the records on server in scrambled shape and after that utilization Key Distribution Center (KDC) for permitting decoding of information gotten from private server, at customer side. We likewise actualize TF-IDF, which gives the effective positioning of results, to enhance the client look involvement. At last we direct the broad tests on dataset, to assess the execution of our proposed framework. Exploratory outcomes will demonstrate that the proposed framework is superior to anything existing one, as far as, protection safeguarding, proficient and secure inquiry on scrambled appropriated archives.*

Keywords-Cloud computing, Encryption, Inner product similarity, Single Keyword Search, Multi-keyword search, ranking.

I. INTRODUCTION

Consistently new data is outsourced as a result of advancement away notwithstanding requirements of customers, then essentially semi-place stock in servers. Cloud enlisting is a Web-based model, where cloud clients can supply their data into the cloud [1]. By stacking data into the cloud, the data proprietors stay unbound after the breaking point of limit. Along these lines, to ensure sensitive data genuineness is a principal errand. To shield data security in the cloud, the data proprietor must be outsourced in the encoded structure to individuals when all is said in done cloud and the data operation is built up on plaintext keyword look. We select the capable measure of "orchestrate organizing". Sort out organizing is used to gage the parallel whole. Encourage organizing gets the centrality of data records to the request address keywords. The request office and security guarded over mixed cloud data are crucial. If we focus huge measure of data reports and data customers in the cloud, it is hard for the necessities of execution, comfort, notwithstanding adaptability. Stressed to encounter the bona fide data recovery, the enormous measure of data files in the cloud server fulfills to come about imperative rank instead of returning undistinguishable outcomes. Situating arrangement minds numerous keyword interest to recover the request rightness. Today's Google organize look for devices, data customers offer game plan of keywords instead of noteworthy keyword look centrality to recuperate the most outrageous basic data. Compose planning is a synchronize coordinating of question keywords which are essentialness to that answer to the request. As a result of inherence prosperity and security, it remains the captivating work for how to relate the mixed cloud looks for. The troublesome of multi-keyword situated look for over encoded cloud data is settled by using stringent security necessities then different multi-keyword semantics. Among different multi-keyword situated semantics, we pick encourage planning. Our responsibilities are dense as takes after, 1) For the primary event when, we explore the issue of multi keyword situated investigate mixed cloud data, and develop a game plan of strict security essentials for such a sheltered cloud data utilize system. 2) We propose two MRSE arranges in perspective of the similarity measure of "organize planning" while at the same time meeting various assurance essentials in two particular hazard models. 3) Thorough examination investigating security and profitability confirmations of the proposed arrangements is given; an investigation on this present reality dataset also exhibit the proposed plots in actuality introduce low overhead on count and correspondence.

II. PROBLEM STATEMENT

The expansive number of information clients and archives in cloud, it is essential for the hunt administration to permit multi-keyword question and give result likeness positioning to meet the compelling information recovery require. The searchable encryption concentrates on single keyword inquiry or Boolean keyword look, and once in a while separates the list items.

a) Single-keyword search without Ranking
b) Boolean-keyword look without Ranking
c) Single-keyword hunt with Ranking

We characterize and take care of the testing issue of protection saving multi-keyword positioned look over scrambled cloud information (MRSE), and set up an arrangement of strict security necessities for such a safe cloud information usage framework to wind up noticeably a reality. Among different multi-keyword semantics, we pick the proficient rule of "facilitate coordinating". Multi-keyword positioned seek over scrambled cloud information (MRSE).Coordinate coordinating" by inward item comparability.

III. LITERATURE SURVEY

Qin Liu et al. proposed Secure and confirmation saving keyword search for in [1]. It gives keyword protection, information affirmation and semantic secure by open key encryption. The rule issue of this pursuit is that the correspondence and computational cost of encryption and unscrambling is more.

Ming Li et al. proposed Authorized Private keyword Search (APKS) in [2]. It gives keyword security, Index and Query Privacy, Fine-grained Search Authorization and Revocation, Multi-dimensional Keyword Search, Scalability and Efficiency. This intrigue system makes the pursuit effectiveness utilizing quality chain of noteworthiness however after a short time each one of the characteristics are not distinctive leveled.

Cong Wang et al in [3] proposed Secure and Efficient Ranked Keyword Search which illuminates prepare overhead, information and keyword protection, least correspondence and figuring overhead. It is not valuable for different keyword missions, Also there is a humble bit of overhead in record building.

Kui Ren et al. [4] proposed Secured padded keyword search for with symmetric searchable encryption (SSE). It doesn't strengthen delicate enthusiasm with open key based searchable encryption, moreover it can't play out various keywords semantic pursue. The upgrades for padded searchable report are not competently performed.

Ming Li et al. [5] proposed Privacy guaranteed searchable disseminated stockpiling system. It is executed utilizing SSE, Scalar-Product-Preserving Encryption and Order-Preserving Symmetric Encryption. It fortifies the security and utilitarian fundamentals. This game plan does not strengthen open key based searchable encryption.

Wei Zhou et al. [6] proposed K-gram based fluffy keyword Ranked Search. In this proprietor make k-gram delicate keyword appeal to for records D and tuple <I, D> is traded to demand server (SS) which is embedded to develop channel for size controlling. The blended record D is traded to point of confinement server. Regardless, the issue is that, the measure of the k-gram creates padded keyword set depends in light of the jacquard coefficient respect.

J. Baek et al. in [7] proposed Secure Channel Free Public Key Encryption with Keyword Search (SCF-PEKS) framework. In these system package servers makes its own specific open and private key join however this methodology experiences outside aggressor by KGA.

H. S. Rhee et al. [8] proposed Trapdoor in recognisability Public-Key Encryption with Keyword Search (IND-PEKS). In this outsourcing is done as SCF-PEKS. It experiences outside aggressor utilizing KGA and isolating the rehash of event of keyword trapdoor.

Peng Xu et al. [9] proposed Public-Key Encryption PEFKS with Fuzzy Keyword Search, in this client makes padded keyword trapdoor Tw and right keyword trapdoor Kw for W. Client asks for Tw to CS. By then CS checks Tw with delicate keyword record and sends superset of sorting out figure messages by Fuzz Test estimation that is executed by CS. The client procedure Exact Test means checking figure works with Kw and recover the encoded records. The course toward making padded keyword archive and right keyword once-over is troublesome for gigantic size database.

Ning et al. [10] proposed Privacy Preserving Multi Keyword Ranked Search (MRSE). It is valuable for known figure content model and foundation show over blended information. It gives low calculation and correspondence overhead. The workplace arranging is chosen for multi-keyword searches for. The downside is that MRSE have negligible standard deviation which diminishes the keyword security.

IV. PROPOSED SOLUTION

We propose an effective system where any endorsed customer can do an interest on mixed data with different keywords, without revealing the keywords he searches for, nor the data of the records that match by the question. Affirmed customers can make look for structures by unmistakable keywords on the cloud to recoup the correlated reports. Our suggestion system empowers that a social affair of customers can request the database gave that they have implied trapdoors for the chase terms that favour the customers to consolidate them in their request. Our proposed system can play out numerous keyword chases in a single question and positions the results so the customer can recoup only the most vital matches requested. Likewise, we develop a plan of strict security essentials. Among different multi keyword semantics, we select the feasible control of "sort out planning".

V. SYSTEM OVERVIEW

The system architecture is stressed by making a direct assistant structure for a structure. It portrays the general edge of the wander which rapidly delineates the working of the structure and the inspiration driving the wander stage is to mastermind an answer of the issue recognized by the need archive. The underneath Figure 1 exhibit the system of the structure. We consider three segments in our structure designing: Data Owner, Data customer and Cloud Server.

- Data Owner is in charge of the making of the database.
- Data Users are the devotees in a gathering who can utilize the documents of the database.
- Cloud Server bargains information offices to confirmed clients. It is fundamental that server be torpid to substance of the database it keeps.

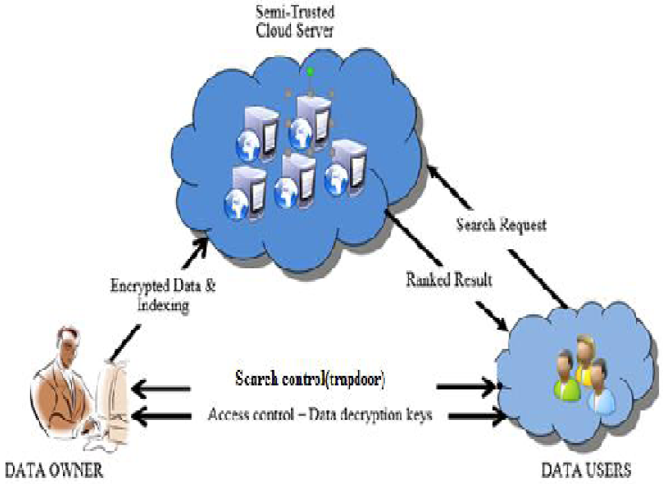

Fig 1: Search over Encrypted Cloud

Data proprietor has measure of data records that he wishes to outsource on cloud server in mixed edge. Before outsourcing, data proprietor will initially assemble a shielded searchable record from a course of action of varying keywords ousted from the report assembling and store both the rundown and the encoded archive on the cloud server. We endeavor the underwriting between the data proprietor and customers are done. To search the record gathering for a given keyword, guaranteed customer makes and displays a request in a secret casing a trapdoor of the keyword to the cloud server. In the wake of getting the chase request, the server is in charge to look for the record and give back the planning course of action of reports to the customer. We think the ensured situated keyword look risky as takes after: the question yield must be returned accommodating clear situated noteworthiness principles, to make record recuperation precision for customers. In any case, cloud server must audit dark or insignificant about the basic guidelines themselves as they reveal critical sensitive data against keyword insurance. To decay exchange speed, the customer may send possible regard k nearby the trapdoor and cloud server just sends back the top-k most appropriate archives to the customer's concerned keyword. Plot Goals: To allow situated output for specialist use of outsourced cloud data under the already specified show, our system setup should rapidly finish security and execution attestations as takes after.

Multi-keyword Ranked Search: To design look for arrangements which allow multi-keyword question and give result closeness situating to fruitful data recuperation, instead of returning undifferentiated results.

Insurance Preserving: To shield the cloud server from taking in additional data from the dataset and the record, and to meet security.

Viability: Above goals on helpfulness and security should be expert with low correspondence and estimation overhead. Mastermind Matching: "Compose planning" [2] is a widely appealing likeness measure which uses the amount of question keywords appearing in the answer to assess the significance of that chronicle to the request. Exactly when customers recognize the right subset of the dataset to be recovered, Boolean request finish well with the right chase require communicated by the customer. It is more adaptable for customers to perceive a summary of keywords exhibiting their stress and recoup the most imperative reports with a rank demand.

VI. METHODOLOGY

A. Stemming:

In phonetic morphology and data recovery, stemming is the way toward decreasing bent (or once in a while determined) words to their pledge stem, base or root shape—for the most part a composed word frame. The stem require not be indistinguishable to the morphological foundation of the word; it is generally adequate that related words guide to a similar stem, regardless of the possibility that this stem is not in itself a substantial root. Calculations for stemming have been considered in software engineering since the 1960s. Many web indexes treat words with an indistinguishable originate from equivalent words as a sort of question extension, a procedure called conflation. Stemming projects are regularly alluded to as stemming calculations or stemmers.

A stemmer for English, for example, should identify the string "cats" (and possibly "catlike", "catty" etc.) as based on the root "cat", and "stems", "stemmer", "stemming", "stemmed" as based on "stem". A stemming algorithm reduces the words "fishing", "fished", and "fisher" to the root word, "fish". On the other hand, "argue", "argued", "argues", "arguing", and "argus" reduce to the stem "argu" (illustrating the case where the stem is not itself a word or root) but "argument" and "arguments" reduce to the stem "argument".

B. Suffix-stripping algorithms:

Suffix-stripping algorithms don't depend on a query table that comprises of curved structures and root frame relations. Rather, a commonly littler rundown of "tenets" is put away which gives a way to the calculation, given an information word shape, to discover its root frame. A few cases of the principles include:

- if the word ends in 'ed', remove the 'ed'
- if the word ends in 'ing', remove the 'ing'
- if the word ends in 'ly', remove the 'ly'

Addition stripping approaches appreciate the advantage of being considerably easier to keep up than savage constrain calculations, accepting the maintainer is adequately educated in the difficulties of etymology and morphology and encoding postfix stripping rules. Addition stripping calculations are here and there viewed as unrefined given the poor execution when managing remarkable relations (like "ran" and 'run'). The arrangements delivered by postfix stripping calculations are restricted to those lexical classes which have surely understood additions with couple of special cases. This, notwithstanding, is an issue, as not all parts of discourse have such an all-around planned arrangement of standards. Lemmatization endeavors to enhance this test.

C. Stop-Words:

In registering, stop words will be words which are sifted through before or subsequent to handling of normal dialect information (text).Though stop words more often than not allude to the most widely recognized words in a dialect, there is no single all inclusive rundown of stop words utilized by all common dialect preparing apparatuses, and in fact not all devices even utilize such a rundown. A few apparatuses particularly abstain from evacuating these stop words to bolster state seek.

Any gathering of words can be picked as the stop words for a given reason. For some web crawlers, these are the absolute most normal, short capacity words, for example, the, is, at, which, and on. For this situation, stop words can bring about issues when scanning for expressions that incorporate them, especially in names, for example, "The Who", "The", or "Take That". Other web crawlers expel the absolute most normal words—including lexical words, for example, "need"— from an inquiry with a specific end goal to enhance execution.

Hans Peter Luhn, one of the pioneers in data recovery, is credited with begetting the saying and utilizing the idea. The expression "stop word", which is not in Luhn's 1959 introduction, and the related terms "stop rundown" and "stoplist" show up in the writing in the blink of an eye a short time later.

A forerunner idea was utilized as a part of making a few concordances. For instance, the principal Hebrew concordance, Meir local, contained a one-page rundown of unindexed words, with no substantive relational words and conjunctions which are like present day stop words.

D. TF-IDF

TF-IDF remains for term recurrence opposite archive recurrence, and the TF-IDF weight is a weight regularly utilized as a part of data recovery and content mining. This weight is a factual measure used to assess how critical a word is to a record in an accumulation or corpus. The significance builds relatively to the quantity of times a word shows up in the archive yet is balanced by the recurrence of the word in the corpus. Varieties of the TF-IDF weighting plan are regularly utilized via web search tools as a focal apparatus in scoring and positioning an archive's importance given a client inquiry.

One of the least difficult positioning capacities is figured by summing the TF-IDF for each question term; numerous more complex positioning capacities are variations of this straightforward model.

TF-IDF can be effectively utilized for stop-words separating in different subject fields including content outline and characterization.

Commonly, the tf-idf weight is formed by two terms: the principal processes the standardized Term Frequency (TF), otherwise known as. The quantity of times a word shows up in a report, isolated by the aggregate number of words in that archive; the second term is the Inverse Document Frequency (IDF), processed as the logarithm of the quantity of the records in the corpus partitioned by the quantity of records where the particular term shows up.

TF: Term Frequency, which measures how much of the time a term, happens in a report. Since each record is distinctive long, it is conceivable that a term would seem significantly more circumstances in long reports than shorter ones. Along these lines, the term recurrence is regularly separated by the report length (otherwise known as. the aggregate number of terms in the record) as a method for standardization:

TF (t) = (Number of times term t appears in a document) / (Total number of terms in the document).

IDF: Inverse Document Frequency, which measures how important a term is. While computing TF, all terms are considered equally important. However it is known that certain terms, such as "is", "of", and "that", may appear a lot of times but have little importance. Thus we need to weigh down the frequent terms while scale up the rare ones, by computing the following:

IDF (t) = log_e (Total number of documents / Number of documents with term t in it).

E. Build Index Tree

Input: the document collection F = ff1, f2, ..., fn g with the identifiers FID = {FID—FID = 1, 2... n }.

Output: the index tree T

```
for each document fFID in F do
    Construct a leaf node u for fFID,
    Insert u to CurrentNodeSet;
end for
while the number of nodes in CurrentNodeSet is larger than 1 do
    if the number of nodes in CurrentNodeSet is even, i.e. 2h then
        for each pair of nodes u0 and u00 in CurrentNodeSet do
            Generate a parent node u for u0 and u00 ,
            Insert u to TempNodeSet;
        end for
    else
        for each pair of nodes u0 and u00 of the former (2h □ 2) nodes in CurrentNodeSet do
            Generate a parent node u for u0 and u00 ;
            Insert u to TempNodeSet;
        end for
        Create a parent node u1 for the (2h - 1)-th and 2h-th node, and then create a parent node u for u1 and the (2h + 1)-th node;
        Insert u to TempNodeSet;
    end if
    Replace CurrentNodeSet with TempNodeSet and then clear TempNodeSet;
end while
return the only node left in CurrentNodeSet, namely, the root of index tree T ;
```

F. BDMRS

SK ⟵ Setup () initially, the data owner generates the secret key set SK, including 1) A randomly generated m-bit vector S where m is equal to the cardinality of dictionary, and 2) two (m X m) invertible matrices M1 and M2. Namely, SK = {S, M1, M2}.

I ⟵ GenIndex (F, SK) First, the unencrypted index tree T is built on F by using

T⟵ BuildIndexTree (F) Secondly, the data owner generates two random vectors (D’u, D”u) for index vector Du in each node u, according to the secret vector S. Specifically, if S[i] = 0, D’u[i] and D”u[i] will be set equal to Du[i]; if S[i] = 1, D’u[i] and Du”u[i] will be set as two random values whose sum equals to Du[i]. Finally, the encrypted index tree I is built where $\{M_1^T D'_u, M_2^T D''_u\}$ the node u stores two encrypted index vectors Iu=

TD ⟵ GenTrapdoor (Wq, SK) with keyword set Wq, the unencrypted query vector

Q with length of m is generated. If wi “Wq, Q[i] stores the normalized IDF value of wi; else Q[i] is set to 0. Similarly, the query vector Q is split into two random vectors Q’ and Q”. The difference is that if S[i] = 0, Q? [i] and Q” [i] are set to two random values whose sum equals to Q[i]; else Q’ [i] and Q” [i] are set as the same as Q[i]. Finally, the algorithm returns the trapdoor TD = $\{M_1^{-1} D'_u, M_2^{-1} D''_u\}$

Relevance Score ⟵ SRScore (Iu, TD) With the trapdoor TD, the cloud server computes the relevance score of node u in the index tree I to the query.

G. EDMRS Scheme

The enhanced EDMRS scheme is almost the same as BDMRS scheme except that:

SK⟵ Setup (): In this algorithm, we set the secret vector S as a m-bit vector, and set M1 and M2 are (m + m') (m + m') invertible matrices, where m' is the number of phantom terms.

I ⟵ GenIndex (F; SK): Before encrypting the index vector Du, we extend the vector Du to be a (m+m') - dimensional vector. Each extended element Du [m+ j], j = 1... m', is set as a random number"j .

TD ⟵ GenTrapdoor (Wq, SK) The query vector Q is extended to be a (m + m')- dimensional vector. Among the extended elements, a number of m" elements are randomly chosen to set as 1, and the rest are set as 0.

Relevance Score ⟵ SRScore(Iu, TD) After the execution of relevance evaluation by cloud server, the final relevance score for index vector Iu equals to Du ˆA $\Sigma\varepsilon v$, where $v \ \varepsilon \ \{j|Q[m + j] = 1\}$

VII. IMPLEMENTATION

A. Data User Module:

Information clients are clients on this framework, will's identity arranged to download documents from the cloud that are traded by the information proprietors. Since the documents set away on the cloud server could be in huge numbers, there is an intrigue office accommodated the client. The client ought to be able to do a multi-keyword look on the cloud server. Once, the outcome shows up for the particular intrigue, these clients ought to be able to send a demand to the individual information proprietors of the document through the framework (likewise called trap-section ask for) for downloading these records. The information clients will comparatively be given a demand bolster screen, where it will tell if the information proprietor has perceived or rejects the demand. On the off chance that the demand has been affirmed, the clients ought to be able to download the decoded record.

B. Information Owner Module:

In this module, the data proprietors should have the ability to exchange the records. The reports are encoded before the records are exchanged to the cloud. The data proprietors are given another option to enter the keywords for the record that are exchanged to the server. These keywords are used for the requesting reason which helps the interest return values quickly. These records when once available on the cloud, the data customers should be skilled interest using the keywords. The data proprietors will moreover be outfitted with a request underwriting screen so they can support or reject the request that is gotten by the data customers.

C. Document Upload and Encryption Module:

In this module, the data proprietors should have the ability to exchange the archives. The records are mixed before the reports are exchanged to the cloud. The data proprietors are given a contrasting option to enter the keywords for the record that are exchanged to the server. These keywords are used for the requesting reason which helps the chase return values quickly. These records when once open on the cloud, the data customers should have the ability to chase using keywords. The data proprietors will in like manner be outfitted with a request underwriting screen so they can support or reject the requests that are gotten by the data customers. The record before exchange ought to be encoded with a key so that the data customers can't just download it without this key. This key will be requested by the data customers through the trap-portal. The encryption of these records uses RSA figuring so that unapproved customers won't have the ability to download these archives.

D. Document Download and Decryption Module:

Information clients are clients on this framework, will's identity ready to download documents from the cloud that are transferred by the information proprietors. Since the records put away on the cloud server could be in immense numbers, there is a pursuit office gave to the client. The client ought to have the capacity to do a multi-keyword seek on the cloud server. Once, the outcome shows up for the particular pursuit, the clients ought to have the capacity to send a demand to the individual information proprietors of the document through the framework (additionally called trap-entryway ask for) for downloading these records. The information clients will likewise be given a demand endorsement screen, where it will tell if the information proprietor has acknowledged or dismisses the demand. On the off chance that the demand has been endorsed, the clients ought to have the capacity to download the unscrambled document. The record before download should be unscrambled with a key. This key will be asked for by the information clients through the trap-entryway ask. Once the key is given amid the download, the information clients will have the capacity to download the record and utilize them.

E. Rank-Search Module:

This module enables the information clients to search for the reports with multi-keyword rank looking. This model uses the on occasion utilized rank pursuing figuring down present the yield for multi-keywords. "Energize Matching" administer will be gotten a handle on for the multi-keyword pursuing. This module in like way oversees making an archive for speedier pursue.

VIII. EXPERIMENTAL RESULT

Fig. 2 shows look time correlation diagram; in roar chart X-hub demonstrates the calculation by which records are sought while Y-pivot indicate time required for seeking question related in ms.

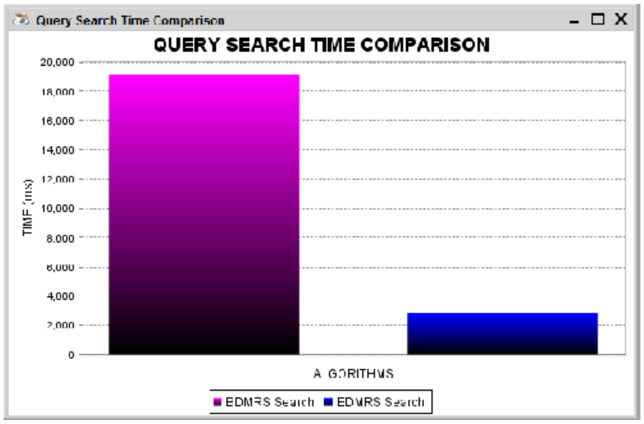

Fig 2: Query Search Time Comparison

Fig. 3 shows time diagram; in above chart X-pivot indicates number of records in gathering while Y-hub demonstrate time required for producing file tree in ms, with increment in number of archives the time required to create list tree is additionally increment.

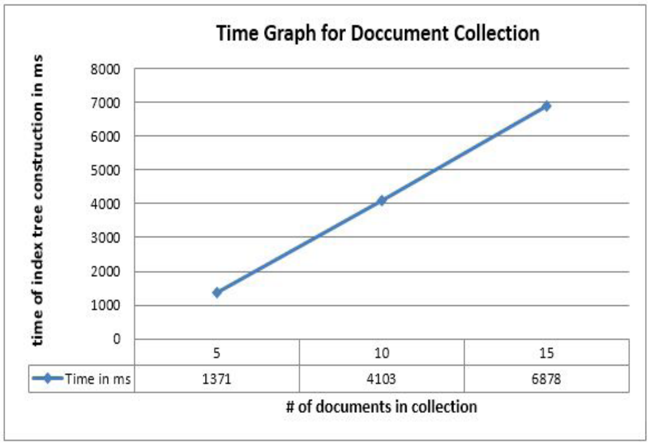

Figure 3: Time Graph for Document Collection

Fig. 4 shows time diagram; in above chart X-hub indicates number of keywords in word reference while Y-pivot demonstrate time required for producing file tree in ms, with increment in number of keywords the time required to create list tree is additionally increment.

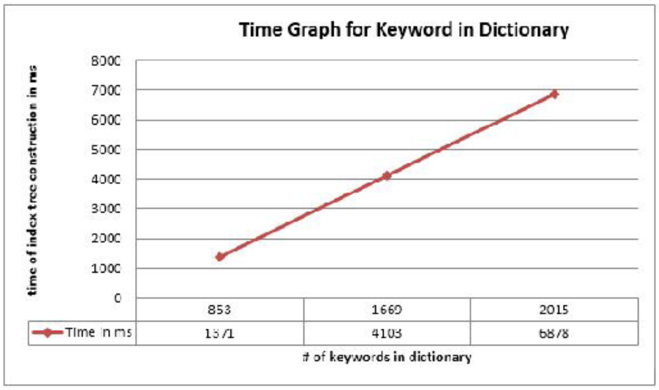

Figure 4: Time Graph for Keyword in Dictionary

XI. CONCLUSION

In this work, firstly we portray and resolve the troublesome of multi-keyword positioned look over scrambled cloud information, and make an assortment of protection necessities. Between various multi-keyword semantics, we select the compelling likeness measure of "facilitate coordinating", i.e., as different matches as likely, to adequately catch the importance of outsourced archives to the question correspondence .In our future work, we will seek supporting other multi keyword semantics over encoded information and checking the honesty of the rank request in the item keywords. For tradition the test of steady multi-keyword semantic without security breaks, we propose an essential thought of MRSE. At that point we give two better MRSE diagrams to acknowledge numerous stringent security necessities in two divergent risk models. Nitty gritty examination contemplating security and effectiveness assurances of proposed plans is given, and trials on this present reality information set demonstrate our future frameworks present low overhead on both calculation and correspondence.

REFERENCES

[1] Qin Liuy, Guojun Wangyz, and Jie Wuz,"Secure and privacy preserving keyword searching for cloud storage services", ELSEVIER Journal of Network and computer Applications, March 2011

[2] Ming Li et al.," Authorized Private Keyword Search over Encrypted Data in Cloud Computing,IEEE proc. International conference on distributed computing systems, June 2011,pages 383-392

[3] Cong Wang et al.,”Enabling Secure and Efficient Ranked Keyword Search over Outsourced Cloud Data”, IEEE Transactions on parallel and distributed systems, vol. 23, no. 8, August 2012

[4] Kui Ren et al., “Towards Secure and Effective Data utilization in Public Cloud”, IEEE Transactions on Network, volume 26, Issue 6, November / December 2012

[5] Ming Li et al.,”Toward Privacy-Assured and Searchable Cloud Data Storage Services”, IEEE Transactions on Network, volume 27, Issue 4, July/August 2013

[6] Wei Zhou et al., “K-Gram Based Fuzzy Keyword Search over Encrypted Cloud Computing “Journal of Software Engineering and Applications, Scientific Research , Issue 6, Volume 29-32,January2013

[7] J. Baek et al., “Public key encryption with keyword search revisited", in ICCSA 2008, vol. 5072 of Lecture Notes in Computer Science, pp. 1249 - 1259, Perugia, Italy, 2008. Springer Berlin/Heidelberg.

[8] H. S. Rhee et al., “Trapdoor security in a searchable public-key encryption scheme with a designated tester," The Journal of Systems and Software, vol. 83, no. 5, pp. 763-771, 2010.

[9] Peng Xu et al., Public-Key Encryption with Fuzzy Keyword Search: A Provably Secure Scheme under Keyword Guessing Attack”,IEEE Transactions on computers, vol. 62, no. 11, November 2013

[10] Ning Cao et al.,” Privacy-Preserving Multi- Keyword Ranked Search over Encrypted Cloud Data”, IEEE Transactions on parallel and distributed systems, vol. 25, no. 1, jan 2014

[11] D. X. D. Song, D. Wagner, and A. Perrig, ”Practical techniques for searches on encrypted data,” in Proc. S & P, BERKELEY, CA, 2000, pp. 44.

[12] C. Wang, N. Cao, K. Ren, and W. J. Lou, Enabling Secure and Efficient Ranked Keyword Search over Outsourced Cloud Data, IEEE Trans. Parallel Distrib. Syst., vol. 23, no. 8, pp. 1467-1479, Aug. 2012.

[13] 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 Proc. ASIACCS, Hangzhou, China, 2013, pp. 71-82.

[14] R. X. Li, Z. Y. Xu, W. S. Kang, K. C. Yow, and C. Z. Xu, Efficient multi-keyword ranked query over encrypted data in cloud computing, Futur. Gener. Comp. Syst., vol. 30, pp. 179-190, Jan. 2014.

[15] Gurdeep Kaur, Poonam Nandal, “Ranking Algorithm of Web Documents using Ontology”, IOSR Journal of Computer Engineering (IOSR-JCE) eISSN: 2278-0661, p-ISSN: 2278-8727Volume 16, Issue 3, Ver. VIII (May-Jun. 2014), PP 52-55