International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 4 Issue 07, July-2015

A Review Report on Searching Techniques on Encrypted Data

Neha
PURCITM,
Department of Computer Science
Punjabi University
Mohali, India

Amninder Gill
PURCITM,
Department of Computer Science
Punjabi University
Mohali, India

***Abstract*— Cloud computing has come as a storm in the streets of Information technology industry. Cloud computing provides a platform to store data on the virtual network. There are a number of cloud storage providers that are providing storage as a service to their users. But the privacy of data on this platform is the main hindrance in adopting cloud technology. To overcome this problem of data privacy, many encryption algorithms have been proposed to be implemented on the data before uploading it on the cloud. But if we talk about large database, then the problem of searching for a chunk of data from this large encrypted database is still there. A number of searching algorithm is there that are suitable for cloud platform, but they may fail in the case of progressive elliptic curve encryption. In this paper, we are comparing various searching techniques on encrypted data suitable for cloud platform.**

Keywords*— *Cloud Computing, Multi-Keyword, Ranking, Encrypted Phrase, Synonym Query, Fuzzy, Privacy-Preserving, Order-Preserving, Semantic Search.

I. INTRODUCTION

Cloud computing is the most innovative technology in the history of internet technology. Cloud computing gives you the power to keep your data on the cloud storage provider. But the security of your sensitive data i.e. the trustworthiness of cloud storage providers is the main hindrance in adopting this technology. Cryptography is probably the best solution to overcome this problem. There are many encryption techniques such as RSA, Elliptic curve cryptography and many more to encrypt your confidential/sensitive data before uploading this to cloud. But another problem of searching from this encrypted data is still there. Many Scientist had discovered various alternatives to search from this encrypted database. Every algorithm has its own specialty some are faster while others are powerful. To have an efficient algorithm it terms of system usability, performance and speed, it is important to go through number of searching.

II. RELATED WORK DONE

Searchable encryption is not a new construct. However, all current strategies have not been so successful in varied aspects that keep them from changing into common or thought. Even hierarchal word proximity searches, a quest that ranks results supported. However shut the question keywords are along, has not been totally enforced by previous analysis.

Song et al. [1] projected a searchable encoding theme supported an interchangeable key. The theme concerned demonstrably secure, question isolation for searches, controlled looking, and support hidden question. Drawbacks: Case inability, regular expression and sub matches don't seem to be supported. Speed of looking and total area needed is big.

Eu-Jin Goh [2] projected AN index looking algorithmic rule supported a bloom filter. Their theme reduced the machine overhead of loading AN index and sorting out files. Drawbacks: Bloom filters lead to false positives; change procedure lacks security analysis, Security model not satisfactory for mathematician searches, unclear experimental evaluation.

PEKS: Public encoding Keyword Search [3] makes use of public key cipher technique. This system focuses on refreshing keywords, removing secure channel and process multiple keywords. Drawbacks: List of keyword should be determined fastidiously so as to stay length of message down. Public key algorithms need giant prime numbers to be calculated so as to come up with usable keys, therefore this method is probably terribly time intense.

PKIS: sensible Keyword Index Search on cloud data centre [4] focuses on cluster search over encrypted info. It involves 2 schemes: PKIS-1 and PKIS-2. PKIS give sensible, realistic, and secure solutions over the encrypted dB. Drawbacks: The common keywords in numerous documents surely cluster have identical index values that ends up in Brute Force attacks.

APKS: licensed personal keyword search over encrypted knowledge in cloud computing [5] deals with multi- keyword search. Multi-dimensional question are born-again to its CNF (Conjunctive traditional Form) formula and are organized in an exceedingly ranked approach. Drawbacks: APKS doesn't forestall keyword attack.

Multi-keyword hierarchal Search over Encrypted Cloud knowledge (MRSE) [6] uses "co-ordinate matching" principle i.e. as several matches as potential, it's AN economical principle among multi-keyword linguistics to refine the result connection. Drawbacks: although keywords square measure protected by trapdoors server will do some applied mathematics analysis over search result. Server will generate trapdoor for set of any multi keyword trapdoor request.

This paper presents the novel technique that gives sub-word matching, exact-matches, regular expressions, language searches, frequency ranking, and proximity-based queries i.e. all forms of looking out that trendy search engines use and users expect to possess.

III. METHOD TO USE SEARCHING OVER ENCRYPTED DATA

3.1 Encrypted Phrase looking within the Cloud [7]

This technique permits each encrypted phrase searches and proximity hierarchal multi-keyword searches to encrypted datasets on untrusted cloud.

A. *Overview*

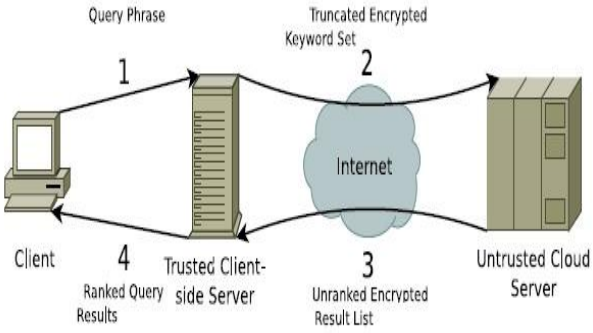

Fig. 1: flow diagram of the planned encrypted phrase looking procedure.

1) The consumer sends a plaintext search question to a sure client-side server.

2) The Client-side server encrypts all keywords within the search question singly exploitation symmetric-key encryption; it then truncates the encrypted keywords to a set range of bits to improve security by permitting for collisions, and queries the untrusted cloud server for the documents containing the set of truncated encrypted keywords.

3) The cloud server will an info question of its encrypted index and returns to the client-side server encrypted knowledge that corresponds to document methods, truncated encrypted keyword index offset, and encrypted keyword locations.

4) The client-side server decrypts this knowledge 1st. From the fresh decrypted keyword index offset it will then verify that came results are literally for the keywords searched and that are merely collisions.

It discards those collisions and filters and/or ranks the pertinent came documents supported relevant keyword locations and frequency. Finally, it sends this hierarchal listing to the first consumer.

B. Indexing

Prior to sorting out a document δ, Associate in nursing encrypted index of the corpus should be generated by the sure client-side server. The index is then encrypted and sent to the untrusted cloud server. Every row within the encrypted index table corresponds to at least one document $\delta \in$ corpus. every row contains 2 columns: Associate in Nursing at random allotted distinctive document id (ID) and a specialized organization that contains truncated symmetric-key encrypted keywords related to encrypted versions of the keyword‘s location in δ (Word Vectors). Additionally, this knowledge string contains Associate in nursing offset that's accustomed map the truncated encrypted keyword with its full version (stored on the trust client-side server). The scientific discipline keys used for each encryptions are a similar key $\mathcal{K}$ which $\mathcal{K}$ is used for decryption similarly. Solely the sure client-side server has access to the price of $\mathcal{K}$. Once the encrypted index is transferred to the untrusted cloud server, Associate in nursing inverted index, based on the encrypted index, is generated by the cloud server to facilitate the looking speed of the index.

C. *Keyword Truncation*

To improve the protection every encrypted keyword is truncated to a predefined range of bits β. The sure client-side server creates a distinctive keyword truncation index price for every encrypted keyword. A table that's keep on the client-side server maps the truncated index price with the totally encrypted keyword. Every index is keep in conjunction with the keyword locations within the encrypted index. This string is encrypted exploitation AES and an irregular salt. As several multiple keywords currently map to a similar bits it makes any applied math frequency analysis attack way less helpful.

D. Searching

When the search begins, the consumer sends the question phrase with multiple keywords, k1 . . . , kn, to a client-side server, that concatenates the keywords to a listing, K. The client-side server then encrypts every $k \in K$ exploitation $\mathcal{K}$ in which the order of keywords is irregular. Every keyword in this list is truncated to β bits to form the encrypted keyword list, K′. The client-side server then transfers this encrypted question, K′, to the untrusted cloud server.

The untrusted cloud server parses K′ into individual encrypted keywords k′ and, exploitation the inverted index, determines the documents, δ, that contain a k′. The selected IDs, the keyword locations, and therefore the truncated keyword index, are sent back to the trusty client- facet server.

The client-side server parses the results that the cloud server returns, that embody the document‘s IDs, paths, and associated encrypted keywords, index, and encrypted locations l′. Every li′ is decrypted to the truncation index τ and li. A proximity ranking perform R, hosted by the client- facet server and is employed to meaningfully rank the results.

Disadvantages:

1. Since it is not possible, previous to decoding, to see that keywords within the collision set were really being probe for, they need to all be came back, decrypted, and so filtered.

2. Doesn't support fuzzy keyword search.

3. Doesn't support word question.

4. Not appropriate for multi-user setting.

3.2. Programmable Order-Preserving Secure Index for Encrypted info question [8]

This technique proposes Associate in Nursing order-preserving theme for categorisation encrypted knowledge, that facilitates the vary queries over encrypted databases. The theme is secure since it randomizes every index with noises, specified the initial knowledge can't be recovered from indexes. Moreover, theme permits the programmability of basic categorisation expressions and therefore the distribution of the initial knowledge will be hidden from the indexes.

In this theme, the i_{th} worth within the plaintext domain is mapped to the i_{th} worth within the cipher text domain, specified the order between plaintexts is preserved between cipher texts. The theme is made over the easy linear expressions of the form a * $\mathcal{x}$ + b. the shape of the expressions is public, however the coefficients a & b are unbroken secret (not acknowledged by attackers). Supported the linear expressions, the categorisation scheme maps Associate in Nursing input worth $\mathcal{v}$ to a * $\mathcal{v}$ + b + noise, where noise may be a random worth. The noise is fastidiously chosen, such that the order of input values is preserved. Categorisation theme permits the programmability of basic categorisation expressions (i.e., the linear expressions). Users will build Associate in Nursing categorisation program that deals with totally different| completely different} input values with different categorisation expressions. On the one hand, the programmability improves the hardiness of the theme against brute-force attacks since there are additional categorisation expressions to attack. On the opposite hand, the programmability will facilitate decouple the distributions of input values and indexes. Once one linear expression is employed to index all input values, the distribution of indexes is similar to the distribution of input values. This downside are often addressed by planning acceptable assortment programs.

3.2.1. randomised Order-Preserving assortment Over Integers:

Suppose v^1 and v^2 square measure 2 integers and v^1 > v^2. Then, the gap between them is a minimum of one, that's $v^1 - v^2 \geq$ one. We will use sensitivity to mean the smallest amount gap. To see how much noise are often else into indexes, specified the indexes keep the order, we want to grasp the smallest amount gap; hence noise is chosen within the vary [0, a ∗ 1). Indexes square measure generated as: a ∗ v^1 + b + noise1 = i1, a ∗v^2 + b + noise2 = i2 so on.

3.2.2. Programmability of Indexes:

This section describes the way to compose basic assortment expressions (skindex or rindex) into assortment programs.

Suppose $\mathcal{v}$ is associate input price. Then, I(v) means that the appliance of I to v , generating v‘s index. If I is $rindex^{sens}$ [a,b], then I(v) = $rindex^{sens}$ [a,b](v). If I is S; $rindex^{sens}$ [a,b], then I(v) = $rindex^{sens}$[a,b](i), wherever i = S(v). The linguistics of assortment steps S is outlined inductively. If S is skindexsens [a,b], then S(v) = $skindex^{sens}$[a,b](v). If S is that the conditional assortment step, then S (v) = S1 (v) if $\mathcal{v}$ makes the condition C true; otherwise, S(v) = S2(v). The condition C is gt(c) or ge (c). The condition gt(c) is true if $\mathcal{v}$ > c, and ge(c) is true if $\mathcal{v} \geq$ c. If S may be a successive composition of steps, then S(v) = S2(i), where i = S1(v) An assortment program is claimed grammatical if it's order preserving. Since in associate assortment program the basic assortment expressions skindex and rindex square measure already order protective, it's order-preserving if all conditional assortment expressions also are order-preserving indexes generated by S1 and S2. In associate assortment program that consists of a sequence of expressions, all intermediate indexes square measure calculated by skindex, that doesn't amendment the sensitivity of input values. Hence, programmers will use the sensitivity of input values in the whole program, easing the burden of programming.

3.2.3. Question of Encrypted Databases:

3.2.3.1 Creation of Encrypted Databases and Tables

To create an information and a table, the information application will issue the subsequent 2 statements.

Create information dbname

Create table tblname (colnm sort,...)

In the statement on top of, kind is that the information kind for the column. The statements area unit translated into the subsequent statements by the proxy. Additionally, the proxy records the schema of the created table in its information. Create information Hash (k, dbname) produce table Hash (k, tblname) (Hash (k, colnm + "EqIdx") String, Hash (k, colnm + "RngIdx") Num, Hash (k, colnm + "Enc") String,..)

That is, 3 columns area unit created for the column colnm. The column colnm + ―EqIdx‖ have the kind String, since its values area unit continuously positional notation strings generated by secure hash functions. The values of column colnm + ―RngIdx‖ area unit generated by our classification mechanism and have the numerical kind. The column colnm + ―Enc‖ for cipher text even have the kind String.

3.2.3.2. Insertion of Values into Tables:

After a table is made, the information application will place a new record into the table by exploitation the following statement.

Insert into tblname (colnm,...) values (v,...)

In the new statement, the worth v is hashed, indexed and encrypted for storing into completely different columns.

Insert into Hash (k, tblname) (Hash (k, colnm + "EqIdx"), Hash (k, colnm + "RngIdx"), Hash (k, colnm + "Enc"),...) values (Hash (k, v), Index(v, sens), Enc(k, v),...)

3.2.3.3. Queries:

A query from the information application will take the subsequent basic type.

Select colnm,... from tblname wherever cond

The condition colnm < c is translated into Hash (k, colnm+ ―RngIdx‖) < Index(c, 0).

Recall that Index(c, 0) is that the minimum index of c. The condition colnm = c is just translated into Hash (k, colnm +“EqIdx”) = Hash (k, c). Assume the sensitivity of values within the colnm column is sens. Then, c + sens is that the next worth of c, and colnm > c is equivalent to the new condition colnm ≥ c + sens, which is translated into Hash (k, colnm + ―RngIdx‖) ≥ Index(c+sens,0).

Note that Index(c+sens,0) is that the minimum index of c + sens. The keywords order by colnm and cluster by colnm area unit of employed in queries. They area unit translated into order by Hash (k, colnm +―RngIdx‖) and cluster by Hash (k, colnm +“EqIdx”), severally.

3.3 Privacy- conserving Keyword-based linguistics Search over Encrypted Cloud information [9]

The linguistics search technique reinforces the system usability by returning the specifically matched files and therefore the files together with the terms semantically just like the question keyword. The co-occurrence of terms is employed because the metric to measure the linguistics distance between terms in linguistics relationship library (SRL). The SRL is built as a weighted graph structure.

A. Overview

1. Information owner includes an assortment of text files the owner then constructs an information for every file and outsources the encrypted information set to the non-public cloud server. The text files area unit encrypted by exploitation ancient even cryptography algorithmic program and uploaded to the general public cloud server.

2. Non-Public cloud server constructs the inverted index and linguistics relationship library exploitation information set provided by information user. Then the Inverted index is outsourced to the general public cloud server for retrieval.

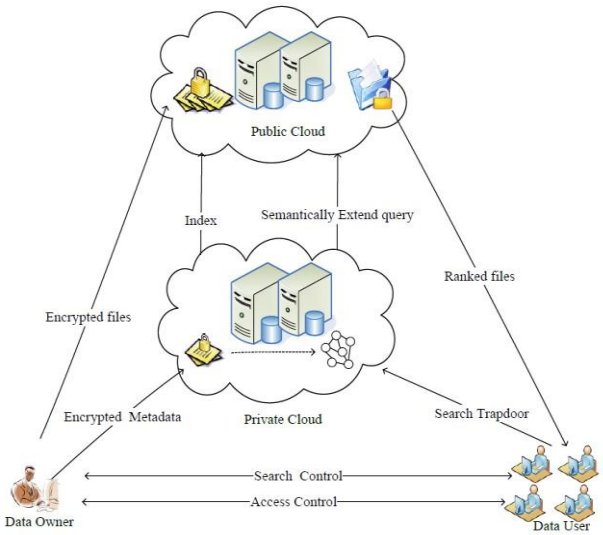

Fig. 2: design of the linguistics Search theme.

3. The licensed information users offer the search trapdoor to the non-public cloud server. Here, the authorization between the information owner and users is fitly done.

4. Upon receiving the request, the non-public cloud server extends the question keyword upon SRL and uploads the extended question keywords set to the general public cloud.

5. Upon receiving the search request, the general public cloud retrieves the index, and returns the matching files to the user so as.

6. Finally, the access management mechanism is utilized to manage the aptitude of the user to rewrite the received files.

B. Notations

- F - The plaintext file assortment denoted as a collection of n information files F = (F1, F2, …, Fn)
- M - The encrypted information set, denoted as, wherever Mi is constructed for Fi.
- I - The inverted index designed from the information set by the server, as well as a collection of posting lists.
- w - The distinct keywords set extracted from Fi.
- Tw - the trapdoor generated for a question keyword w by a user.

C. Algorithm

- KeyGen (, , ,): during this formula the information owner takes k, l, 1 , p as inputs and generates the random keys. It generates the random keys during this way: x ←k, y ← l and outputs Key =.
- BuildMD (key, F): the information owner builds the secure information for every go in file assortment F.
- BuildIndex (M): On receiving the secure information, the personal cloud builds the inverted index.
- BuildSRL (M): This formula is additionally travel by the personal cloud server to construct the linguistics relationship library. It takes the information set as inputs, and exploits data processing formula e.g., Apriori formula, to find the co-occurrence chances of keywords.
- TrapdoorGen (Key, w): In this method, for a search input, the user computes a trapdoor and sends it to the private cloud. The second step is for the personal cloud to extend the question trapdoor and acquire the denotative question keywords set.
- Search(Tw, I): upon receiving the request, the search will be divided into 2 steps : the primary step is for the personal cloud to increase the question trapdoor and sends the denotative question keywords set to the general public cloud. The second step is for the general public cloud to find the matching entries of the index that embody the file identifiers and the associated order-preserved encrypted scores. The general public cloud server then computes the overall connexion score of every file to the question. In the end, the general public server sends back the matched files in an exceedingly stratified sequence, or sends top-k most relevant files.

D. Security Analysis

For one – to – several Order conserving encoding (OPE)

The one-to-many order-preserving encoding introduce the file ID because the further seed within the cipher text chosen method, that the same plaintext won't be deterministically mapped to an equivalent cipher text , however a random worth within the allotted bucket type vary by sampling, that facilitate flatten the score distribution, and shield the keyword privacy type applied math attack.

For stratified linguistics keyword search

1. File Confidentiality: The file is encrypted with ancient regular encoding formula. The encoding cipher is preserved by the information owner solely, that the file confidentiality depends on the inherently security strength of the regular encoding theme. So the file content is protected.

2. Keyword Privacy: If the information owner properly enlarges the vary R, the connexion score can be indiscriminately mapped to solely a sequence of order-preserved numeric values with terribly low duplicates. The planar encrypted connexion score distribution makes it troublesome for the mortal to predict the plaintext score distribution, including predict the keywords.

Search Result analysis:

The overall recall rate is improved, and also the question results area unit a lot of in line with the user‘s actual intentions. For e.g., a user inputs a keyword ‗protocol‘, the files that contain connected words like ‗internet‘, ‗network ‘,‘ authentication ‘will be came back, additionally, the files that embody most of the words will be stratified forward.

Disadvantages:

1. Doesn't support multi-user atmosphere.

2. The scale of vary cannot be unboundedly giant. That the vary size | R | ought to be properly exchange between randomness and potency.

3. Doesn't support equivalent word question.

3.4 Verifiable Attribute-based Keyword Search with Fine-grained Owner-enforced Search Authorization within the Cloud [10]

The outsourced dataset will be contributed from multiple house owners and area unit searchable by multiple users, i.e. multi-user multi-contributor case. The attribute-based keyword search theme with economical user revocation (ABKS-UR) permits climbable fine-grained (i.e. file-level) search authorization (fine-grained owner-enforced search authorization) victimization attribute primarily based (CP-ABE) technique. In the CP-ABE technique specifically, for every file, the information owner generates associate access-policy-protected secure index, wherever the access structure is expressed as a series of AND gates. Solely approved users with attributes satisfying the access policies will get matching result. Users will generate their own search capabilities while not hoping on associate continually on-line trusty authority. The theme permits believability examine the same search results. The index is encrypted with associate access structure instead of public or secret keys supported the attributes (properties of users) of approved users that makes the planned theme a lot of scalable and appropriate for the big scale file sharing system. In this key policy attribute-based cryptography (KP-ABE) theme, cipher text may be decrypted as long as the attributes that are used for cryptography satisfy the access structure on the user personal key.

A. Overview

1. The information owner generates the secure indexes with attribute-based access policies before outsourcing them together with the encrypted information into the cesium (cloud server).

2. To look the information sets contributed from numerous data homeowners, an information user generates a trapdoor of keyword of interest exploitation his personal key and submits it to the cesium. Thus on accelerate the whole search method, enforce the coarse-grained dataset search authorization with the per-dataset user list specified search doesn't have to be compelled to visit a selected dataset if the user isn't on the corresponding user list.

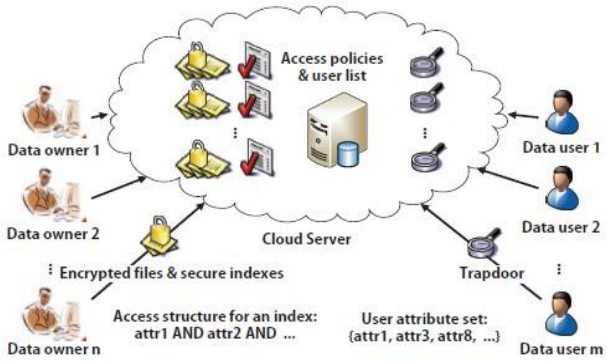

Fig. 3: Framework of approved keyword search over encrypted cloud information.

3. The fine grained file-level search authorization is applied on the approved dataset within the sense that solely users, UN agency are granted to access a selected file, will search this file for the meant keyword. The information owner defines associate access policy for every uploaded file.

4. The cesium can search the corresponding datasets and come back the valid search result to the user if and as long as the attributes of the user on the trapdoor satisfy the access policies of the secure indexes of the same files, and also the meant keyword is found in these files.

B. Search section

In the search section, the cesium returns the search result together with the auxiliary info for result believability check later by the information user. The auxiliary info includes all the user list bloom filters BFUL of the datasets keep on the server , the keyword bloom filters BFw of the datasets that the user is permitted to access, the file list L′w for the meant keyword w. If the search result contains files from this dataset, the tuple in every connected UL (user list) and all the corresponding signatures. If the search result doesn't contain files from this dataset, it's not necessary to come back the corresponding file list. Otherwise, the cesium generates L′w as follows. For the file li in atomic number 103 however not within the search result, the cesium simply computes its hash price h(li) and puts the tuple < Di, w, h(li) > in L′w.

C. Result Authentication

The result could contain errors which will come back from the attainable storage corruption, code malfunction, and intention to avoid wasting machine resources by the server, etc. assure information user of the believability of the came search result by checking its correctness (the same search result so exist within the dataset and stay intact), completeness (no qualified files are omitted from the search result), and freshness (the same result's obtained from the newest version of the dataset).

The user 1st computes tuple hash values hl1, hl2 and hl3 severally. User then generates the hash chain to get the file list hash price hLw, and verifies σ (hLw). Next, user will search this list along with his trapdoor and corresponding Df from the cesium to see if all the matching files are came. Thus, the information user will make sure the believability of the came search result.

D. User Revocation

The aim is to expeditiously revoke users from the present system whereas minimizing the impact on the remaining legitimate users. To revoke a user from current system, we tend to re-encrypt the secure indexes keep on the server and update the remaining legitimate user's secret keys. These tasks may be delegated to the cesium exploitation proxy re-encryption technique in order that user revocation is extremely economical. Specifically, the Ta (Trusted authority) adopts the re- cryptography key generation algorithmic rule to generate the re- cryptography key set. Disadvantages:

1. Doesn't support word question.

2. Doesn't support linguistics based mostly keyword search

3.5 Privacy-Preserving Multi-Keyword Fuzzy Search over

Encrypted knowledge within the Cloud [11]

The multi-keyword fuzzy search theme exploits the locality-sensitive hashing technique. This theme eliminates the need of a predefined keyword lexicon.

A. Overview

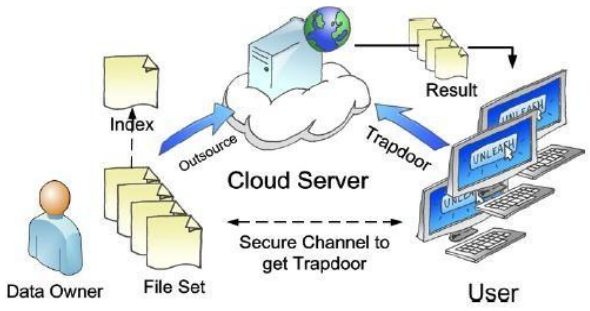

Fig. 4: design of the Multi-keyword fuzzy Search theme

1. To source a set of files to the cloud, the knowledge owner builds a secure searchable index for the file set and so uploads the encrypted files, alongside the secure index, to the cloud server.

2. To look over the encrypted files, a certified user 1st obtains the trapdoor, i.e., the ―encrypted‖ version of search keyword(s), from the information owner, and so submits the trapdoor to the cloud server.

3. Upon receiving the trapdoor, the cloud server executes the search algorithmic rule over the secure indexes and returns the matched files to the user because the search result.

B. Locality-Sensitive Hashing

A Locality-Sensitive Hashing (LSH) perform hashes shut things to identical hash price with higher likelihood than the things that area unit so much apart. To support fuzzy and multiple keyword search, we have a tendency to 1st convert every keyword into a written word vector and so use LSH functions rather than customary hash functions to insert the keywords into the Bloom filter ID.

C. *Main Idea*

1) Written word vector illustration of keyword: written word vector may be a 262-bit long vector that represents a written word set. Every part in the vector represents one of the 262 potential bigrams. For example, the written word set of keyword ―network‖ is. The part is about to one if the corresponding written word exists within the written word set of a given keyword. it's not sensitive to the position of spelling, nor is it sensitive to that letter it was misspelled to. ―nwtwork‖, ―nvtwork‖, or ―netwoyk‖ can all be mapped to a vector with two-element distinction from the initial vector.

2) Bloom filter illustration of index/query: Bloom filter has been wont to build per document index for single keyword precise search state of affairs. With those hash functions, 2 similar inputs, albeit they're solely off by one bit, are hashed to 2 wholly totally different random values. Therefore, they'll solely be used for precise keyword search. LSH functions can hash inputs with similarity inside sure threshold into the same output with high likelihood.

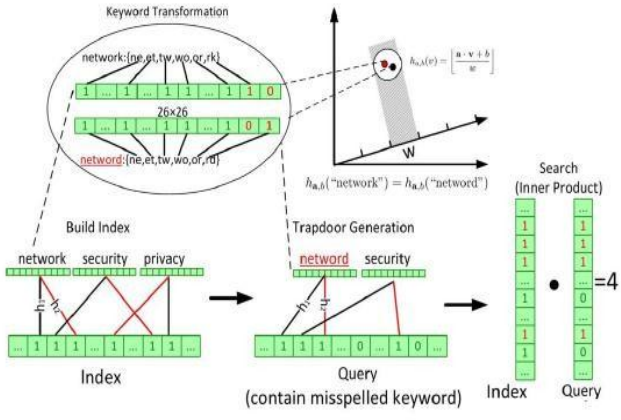

Fig. 5: i).Transform a keyword into a vector. ii).Use 2 LSH functions h1, h2 from identical hash family to get the index and also the question. The word "network" angular distances identical hash price with the misspelled work "netword" beneath LSH perform ha,b as a result of the geometrician distance between their vector representations is inside the pre-defined threshold. iii).The misspelled question matches specifically with the index contains the keywords "network" and "security".

Fig. five shows the plan that a misspelled keyword—network‖ within the user question is hashed into identical bucket because the properly spelled keyword —network‖ in order that a match may be found throughout the search method. The utilization of LSH functions in building the per-file Bloom filter based mostly index is that the key to implementing fuzzy search.

3) Real number based mostly matching algorithm: The question may be generated within the by inserting multiple keywords to be searched into a question Bloom filter. The search will then be done by qualifying the relevancy of the question to every file that is finished through an easy real number of the index vector and also the question vector.

If a document contains the keyword(s) within the question, the corresponding bits in each vectors are going to be one therefore the scalar product can come back a high worth. This straightforward scalar product result therefore may be a sensible live of the quantity of matching keywords.

• Known Cipher text Model: The cloud server will solely access the encrypted files, the secure indexes and also the submitted trapdoors. The cloud server can even understand and record the search results. The linguistics that means of this threat state of affairs is captured by the non-adaptive attack model.

• Known Background Model: The cloud server is aware of extra background data during this model. The background refers to the data which might be learned from a comparable dataset, for instance, the keywords and their applied math data, like the frequency.

This technique is supported below known background model, oppose probably will recover the encrypted indexes through linear analysis and any infer the keywords within the index. To secure the linkage between the keywords and also the Bloom filter, we tend to introduce an additional security layer, i.e., a pseudo-random operate f.

D. Algorithm

• KeyGen (m, s): This algorithmic program given a parameter m, generates the key key SK (M1, M2, S), where M1, M2 ∈ℝ m×m square measure invertible matrices whereas S ∈ m is a vector. Given another parameter s, generate the hash key pool (HK).

• BuildIndex(D, SK, l) : select l freelance LSH functions from the p-stable LSH family H and one

Pseudorandom operate f: ∗ × s → ∗.For each file D,

1) Extract the keywords set W = from D.

2) Generate a m-bit Bloom filter ID. Insert W into ID using the hash functions gi = fki ◦ hi, hi ∈ H, 1 ≤ i

≤ l.

3) Inscribe the ID with SK and come back EncSK (ID) because the index.

• Trapdoor (Q, SK): Generate a m-bit long Bloom filter. Insert the letter of the alphabet exploitation an equivalent hash functions gi, i.e., gi = fki ◦ hi, hi ∈ H, one ≤ i ≤ l into the Bloom filter. Encrypt the letter of the alphabet with SK and come back the EncSK (Q) as the trapdoor.

• Search (EncSK(Q),EncSK(ID)): Output the scalar product < EncSK(Q), EncSK(ID) > because the search result for the query letter of the alphabet and also the document D.

Disadvantages:

1. The technique is vulnerable to prophetic attacks below known cipher text model.

2. Doesn't support word question.

3. Not appropriate for multi-user atmosphere.

4. Doesn't enable linguistics based mostly search.

3.6 Multi-keyword graded Search over Encrypted Cloud

Data supporting word question [12]

The technique proposes a semantics-based multi- keyword graded search theme over encrypted cloud knowledge that supports word question. The search results is achieved once approved cloud customers input the synonyms of the predefined keywords, not the precise or fuzzy matching keywords, because of the attainable word substitution and/or her lack of tangible information concerning the info.

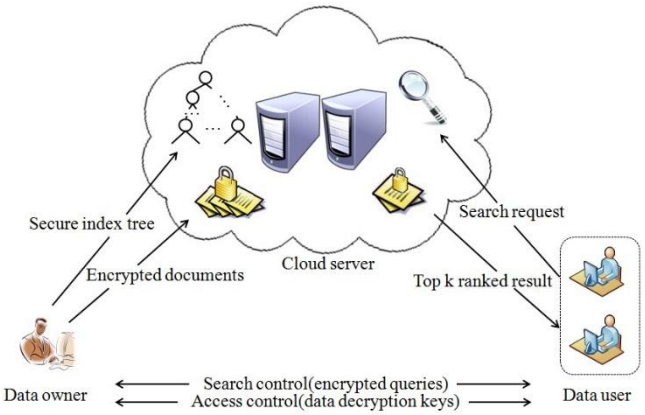

Fig. 6: design of the Multi-keyword graded search supporting word question

Notations

• DC – the plaintext document assortment, expressed as a collection of m documents DC = d1, d2, d3..., dm.

• C – the encrypted type of DC hold on within the cloud server, expressed as C = c1, c2...cm.

• W – the keyword wordbook, as well as n keywords, expressed as W = w1, w2...wn.

• I – the searchable index tree generated from the total document set DC. (Each leaf node within the index tree is related to a document in DC.)

• Doctor of Divinity – the index vector of document d for all the keywords in.

• Letter of the alphabet – the question vector for the keyword set W.

• d – The encrypted type of Doctor of Divinity. Rank function: In info retrieval, a ranking operate is typically wont to value relevant lots of matching files to asking. The ranking

operate used here is ―TF×IDF‖ wherever TF (term frequency) denotes the prevalence of the term showing in the document, and military force (inverse document frequency) is usually obtained by dividing the full variety of documents by the quantity of files containing the term. That means, TF represents the importance of the term in the document and military force indicates the importance or degree of distinction in the whole document assortment. Every document is reminiscent of Associate in nursing index vector DD that stores normalized TF weight, and also the question vector letter of the alphabet stores normalized military force weight. every dimension of DD or letter of the alphabet is said to a keyword in W, and also the order is same thereupon in W, that is, Dd[i] is reminiscent of keyword Wisconsin in W. The notations used in similarity analysis operate are showed as follows:

• fd,j, the TF of keyword wj inside the document d;

• fj, the quantity of documents containing the keyword wj ;

• M, the full variety of documents within the document collection;

• N, the total variety of keywords in the keyword dictionary;

• wd ,j , the TF weight computed from fd, j;

• wq,j, the military force weight computed from N and fj;

The definition of the similarity operate is as follows:

SC (Q ,Dd) = Where wq,j = 1+ ln fd,j . The normalized TF and military force weight are and severally and the vectors letter of the alphabet and DD ar unit vectors.Construction of keyword set extended by synonym:

Let N be the full variety of texts in corpus, let n be the quantity of texts containing the term i in corpus, let E1 be the quantity of texts within the largest class containing the term i, let E2 be the quantity of texts within the second largest class containing the term i. The new weight issue Cd is more to the formula of TFIDF, the improved formula is as follows:

Wik = TF*IDF*Cd = TF* *Cd = fik * *

So the keywords are extracted from every outsourced text document by mistreatment our improved methodology. All keywords extracted from identical one text type one keyword set, and every one subsets type the keyword set finally. All the outsourced text documents is expressed as follows:

We build a standard equivalent word synonym finder on the muse of the New Yankee Roget‘s school synonym finder (NARCT). NARCT is ablated in amount by US consistent with the subsequent 2 principles:

(1) Choosing the common words;

(2) Choosing the words that will be semantically substituted fully.

The made equivalent word set contains a complete of 6353 equivalent word teams when the reduction. The keyword set is extended by mistreatment our made equivalent word synonym finder. The new keyword set containing equivalent word is shown as follows:

Where s1 represents the equivalent word of kfi. If a keyword has 2 or additional synonyms, then all synonyms are more into the keyword set. The repetitive keywords are deleted to cut back the burden of storage. At last, a simplified keyword set and corresponding keyword evaluation table are made and used.

Disadvantage:

1. Doesn't support syntactical transformation, anaphora resolution and alternative linguistic communication process technology.

IV. CONCLUSION

In this paper varied techniques to search out documents with needed keywords are mentioned. to form the search of needed documents additional correct these techniques is integrated which will give the user with easy supporting multi keyword search, fuzzy keyword search, search supported equivalent word words within the question, linguistics primarily based search. The efficient keyword search would be provided using Ranked Search Algorithm based on proximity and similarity based ranking techniques. Also technique to deal with indexing and searching on encrypted databases and data shared between multi-users, multi-contributors scenarios is discussed in this paper. To provide confidentiality user authentication, user revocation and authentication of returned results is done using Fine- grained Owner-enforced Search Authorization. To provide better security at server technique that supports keyword truncation is used.

ACKNOWLEDGEMENTS

I would like to thank my guide Ms. Amninder Gill for helping me out in my research work.

REFERENCES

[1] Dawn Xiaodong, Song David Wagner and Adrian Perrig, “Practical Techniques for Searches on Encrypted Data‖” in Proc. of IEEE Symposium on Security and Privacy‘00, 2000.
[2] Eu–Jin Goh, “Secure indexes‖ in the Cryptology ePrint Archive”, Report 2003/216, March 2004.
[3] Joonsang Baek, Reihaneh Safiavi-Naini, Willy Susilo, “Public Key Encryption with Keyword Search Revisited‖”, in Cryptology ePrint Archive, Report 2005/191, 2005.
[4] Hyun-A Park, Jae Hyun Park and Dong Hoon Lee, “PKIS: practical keyword index search on cloud datacenter‖”, in EURASIP Journal on Wireless Communications and Networking 2011.
[5] Li, M., S. Yu, N. Cao and W. Lou, “Authorized private keyword search over encrypted data in cloud computing”.‖ in Proceedings of the 31st International Conference on Distributed Computing Systems, June 20-24, 2011, Minneapolis, MN, USA, pp: 383-392.
[6] Ning Cao, Cong Wang, Ming Li, Kui Ren, and Wenjing Lou, “Preserving Multi-keyword Ranked Search over Encrypted Cloud Data”‖.
[7] Steven Zittrower and Cliff C. Zou, “Encrypted Phrase Searching in the Cloud”‖ in Globecom - Communication and Information System Security Symposium, 2012.
[8] Dongxi Liu Shenlu Wang, “Programmable Order-Preserving Secure Index for Encrypted Database Query”‖, in IEEE Fifth International Conference on Cloud Computing, 2012.

[9] Xingming Sun, Yanling Zhu, Zhihua Xia and Liahong Chang, "Privacy- Preserving Keyword-based Semantic Search over Encrypted Cloud Data"‖, in International Journal of Security and Its Applications Vol.8, No.3, pp.9-20, 2014.
[10] Wenhai Sun, Ahucheng Yu, Wenjing Lou and Y. Thomas Hou, "Verifiable Attribute-based Keyword Search with Fine-grained Owner-enforced Search Authorization in the Cloud"‖, in IEEE Journal, 2013.
[11] Bing Wang, Shucheng Yu, Wenjing Lou, Y. Thomas Hou, "Privacy- Preserving Multi-Keyword Fuzzy Search over Encrypted Data in the Cloud"‖, in IEEE INFOCOM 2014 - IEEE Conference on Computer Communications.
[12] Zhangjie Fu, Xingming Sun, Nigel Linge and Lu Zhou, "Multi-keyword Ranked Search over Encrypted Cloud Data Supporting Synonym Query"‖, in IEEE Transactions on Consumer Electronics, Vol. 60, No. 1, February 2014.
[13] Ankit Doshi, Kajal Thakkar, Sahil Gupte and Anjali Yeole, "A Survey on Searching and Indexing on Encrypted Data"‖, in International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181, Vol. 2 Issue 10, October – 2013.
[14] Xingming Sun, Lu Zhou, Zhangjie Fu and Jin Wang, "Privacy-Preserving Multi-Keyword Ranked Search over Encrypted Data in the Cloud supporting Dynamic Update"‖, in International Journal of Security and its Application (IJSIA), Vol.8, No.6, pp.1-16, 2014.
[15] Zhihua Xia**,** Yanling Zhu**,** Xingming Sun, and Liahong Chen, "Secure Semantic expansion based search over encrypted cloud data supporting similarity ranking"‖, in Journal of Cloud Computing,2014.