Volume 3, Issue 4, April 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Efficient Interactive Fuzzy Keyword Search Over Encrypted Data in Cloud Computing

P. Kalidas
PG Student
Department of CSE
Bharath University
Chennai, India

R. Chandrasekaran
Assistant Professor
Department of CSE
Bharath University
Chennai, India

Dusmant Kumar Sahu
PG Student
Department of CSE
Bharath University
Chennai, India

G.Michale
Assistant Professor
Department of CSE
Bharath University
Chennai, India

Abstract— Cloud computing is an online computing within which info is keep and accessed employing a remote third party server known as cloud, comparatively than being keep regionally on our mechanism and also the resources, software's, and knowledge are provided to users on demand. In cloud computing background knowledge protection is on-going difficult task, and then the sensitive knowledge has got to be encrypted previous to outsourcing. Though typical searchable coding schemes permit a user to firmly search quite encrypted knowledge from facet to facet keywords and by selection retrieve files of interest, these techniques support solely actual keyword search. In this paper for the primary time we have a tendency to formalize and solve the problem of realistic fuzzy keyword search over encrypted cloud knowledge where as maintaining keyword privacy. Fuzzy keyword search greatly enhances system usability by returning the matching files once users' looking inputs precisely match the predefined keywords or the nearest doable matching files supported keyword similarity linguistics, once actual match fails. In our resolution, we have a tendency to exploit edit distance to quantify keywords similarity and extremely developed techniques on constructing fuzzy keyword sets, which understand optimized storage and demonstration overheads. We have a propensity to boot propose a novel symbol-based trie-traverse looking scheme, where a multi-way tree structure is constructed positive persecution symbols remodeled from the resulted fuzzy keyword sets. Through exact security analysis, we have a leaning to show that our designed resolution is secure and privacy-preserving, where as properly realizing the goal of fuzzy keyword search. Exhaustive experimental outcome expose the energy of the planned resolution.

Keywords— Effective Fuzzy Keyword Search, Searchable coding, Edit Distance theme, symbol based technique. Straight forward approach

1. Introduction

As Cloud Computing becomes current, more and more sensitive information are being centralized into the cloud, like emails, personal health records, company documents, etc. By storing their knowledge into the cloud, the information data owners will be eased from the burden of knowledge storage and maintenance therefore so as to enjoy the demand top quality knowledge storage service. However, the actual fact that knowledge house owners and cloud server don't seem to be within the same trust valuable domain might place the sourced knowledge in danger, because the cloud server might now not be absolutely trust worthy. It follows that sensitive knowledge typically ought to be encrypted before outsourcing for knowledge privacy and combating uninvited accesses. However, encryption makes effective knowledge utilization a really difficult task providing there may well be an outsized quantity of outsourced knowledge files. Moreover, in Cloud Computing, knowledge house owners might share their outsourced knowledge with an outsized variety of users.

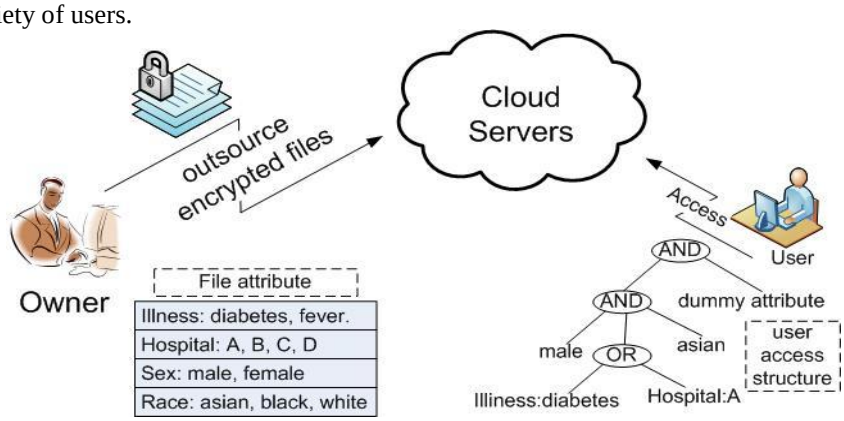

Fig 1. Architecture of Fuzzy Keyword Search

The individual users may wish to solely retrieve sure specific knowledge files they're fascinated by throughout a given session. One among the foremost common ways in that is to by selection retrieve files through keyword-based search rather than retrieving all the encrypted files back which is totally impractical in cloud computing situations. Such keyword-based search technique permits users to by selection retrieve files of interest and has been wide applied in plaintext search situations, like Google search. One among the foremost common ways in which or techniques is to by selection retrieve files through keyword primarily based search rather than retrieving all the encrypted files back. The data coding additionally demands the protection of keyword privacy since keywords typically contain vital information associated with the information files.

The existing searchable coding techniques don't suit for cloud computing scenario as a result of they support solely actual keyword search. This important disadvantage of existing schemes signifies the essential would like for brand new strategies that support looking flexibility, tolerating each minor varieties and format inconsistencies. A secure fuzzy search capability is demanded for achieving increased system usability in Cloud Computing. The most disadvantages are however with efficiency looking the information and retrieve the leads to most secure and privacy protective manner. For retrieving the information during a most secure and privacy protective manner the keyword looking technique is employed and to look the information in additional economical manner, the fuzzy keyword search is introduced. There fore potency of fuzzy keyword search is that the main side within the security of knowledge retrieval. Once the files are retrieved in reasonable manner, most relevant knowledge will be retrieved. The present system is especially that specialize in the "fuzzy keyword search" technique. The information that's outsourced is encrypted, constructs fuzzy sets supported each wild card technique and gram based technique, and additionally introduced a symbol-based trie-traverse search scheme, wherever a multi-way tree was created for storing the fuzzy keyword set and at last retrieving the information. In this paper, we have a tendency to specialize in facultative effective however privacy protective fuzzy keyword search in cloud computing .to the simplest of our information we have a tendency to formalize for the primary time the matter of effective fuzzy keyword search over encrypted cloud knowledge, also maintaining keyword privacy. Fuzzy keyword search greatly enhances system usability by returning the matching files once user's looking inputs accurately match. Pre outlined keyword or the nearest possible matching files supported keyword similarity linguistics, once actual match fails. We have lot of specifically, have a tendency to use edit distance to quantify keywords similarity and develop a completely unique technique, i.e., a wildcard-based technique, for the development of fuzzy keyword sets. Supported the created fuzzy keyword sets, we have a tendency to propose practical fuzzy keyword search scheme. Through rigorous security analysis, we have a tendency to show that the planned resolution is secure and privacy-preserving, whereas properly realizing the goal of fuzzy keyword search.

2. Related Work

2.1 Plaintext fuzzy keyword search:

Recently, the importance of fuzzy search has received attention within the context of plain text looking in info retrieval community .They addressed this downside within the ancient info access paradigm by permitting user to look while not victimization try-and-see approach for locating relevant info supported approximate string matching. At the primary look, it appears doable for one to directly apply these string matching algorithms to the context of searchable coding by computing the trapdoors on a personality base inside action alphabet. However, this trivial construction suffers from the lexicon and statistics attacks and fails to realize the search privacy.

2.2 Searchable coding:

Established searchable encryption has been wide studied within the context of cryptography. Among those works, most are specialize in effectiveness enhancements and security which means formalizations. To realize a more efficient search, each planned similar "index" approaches, wherever one encrypted hash table index is constructed for the whole owner source files source encrypted files entice doors of search request file retrieval fuzzy keyword set Index users cloud server file group. With in the index table, every entry consists of the door of a keyword action an encrypted set of file identifiers whose corresponding knowledge files contain the keyword. This existing scheme support only exact keyword search, and so don't seem to be appropriate for cloud computing others. Personal matching , as another connected notion has been studied principally within the context of secure united computation to permit totally different parties compare some function of their own knowledge collaboratively while not informative their knowledge to the others. These functions may well be connection or approximate personal matching of 2 sets, etc. The personal information retrieval is an often-used technique to retrieve the matching things on the Q.T. that has been wide applied in info retrieval from information and frequently incurs unexpectedly computation quality.

2.3 Complete Search:

The planned techniques to support "Complete Search," within which a user selection in keywords letter by letter, and also the system finds records that embody these keywords (possibly at regardful places) [4, 5, 2,3].Our work differs from theirs as follows.

(1) Complete Search primarily targeted on compression of index structures, particularly in disk-based settings. Our work focuses on efficient query process wrongness in-memory indexes, so as to realize a high interactive speed.

(2) Our work permits fuzzy search, creating the computation more difficult.

(3) Used for a question with multiple keywords, Complete Search chiefly caches the results of the question excluding the last keyword, which can need computing and caching an outsized quantity of intermediate results.

3. Problem Formulation

A. System Model:

In this paper, we have a tendency to think about a cloud system consisting of knowledge owner, knowledge user and cloud server. Given a group Of n encrypted knowledge files *C* = (F1, F2, . . . , FN) keep within the cloud server, a predefined set of distinct keywords *W = {w*1, *w*2, ...,*wp}*, the cloud server provides the search service for the approved users over the encrypted knowledge C. we have a tendency to assume the authorization between the information owner and users is befittingly done. Licensed user varieties during a request to by selection retrieve knowledge files of his/her interest. The cloud server is accountable for mapping the looking request to line of knowledge files, wherever every file is indexed by a file ID and joined to a collection of keywords. The fuzzy keyword search theme returns the search results per the subsequent rules:

1) If the user's looking input precisely matches the pre-set keyword, the server is anticipated to come the files containing the keyword1

2) If there exist typos and/or format inconsistencies within the looking input, the server can come the nearest doable results supported pre-specified similarity linguistics (to be formally outlined in section III-D). Design of fuzzy keyword search is shown within the Fig. 1.

B. Threat Model

We think about a semi-trusted server. Even supposing knowledge files are encrypted, the cloud server might attempt to derive alternative sensitive info from users' search requests whereas playing keyword-based search over C. Thus, the search ought to be conducted during a secure manner that permits knowledge files to be firmly retrieved whereas revealing as very little info as doable to the cloud server. During this paper, once coming up with fuzzy keyword search theme, we are going to follow the safety definition deployed within the ancient searchable coding .More specifically, it's needed that nothing ought to be leaked from the remotely keep files and index on the far side the end result and also the pattern of search queries.

C. Design Goals

The Design goal of this project is to focus on enabling effective yet privacy-preserving fuzzy keyword search in Cloud Computing. To the most excellent of our understanding, we formalize for the first time the problem of effective fuzzy keyword search over encrypted cloud data while maintaining keyword privacy. In this paper, we try to solve the problem of effective fuzzy keyword search over encrypted cloud data while maintaining keyword privacy.

So our goals are:

1) To exploit edit distance to enumerate keywords comparison.

2) To design efficient and efficient fuzzy search scheme.

3) To authorize the safety of planned solution**.**

D. Existing System

Using this technique of secure trapdoor the existing system allows user to perform fuzzy keyword search over encrypted data. It also achieves search privacy method. But existing system have many disadvantages:

1) Approach used by existing system have efficiency problem.

2) Fuzzy keyword set requires large storage capacity.

For example, after substitution operation on the first character of keyword, results we get are given below. CASTLE: {AASTLE, BASTLE, DASTLE, YASTLE, ZASTLE}.

E. Preliminaries

Edit Distance

There are many strategies to quantitatively measure the string similarity. During this paper, we have a tendency to resort to the well-studied edit distance for our purpose. The distance ed (w1, w2) between 2 words w1 and w2 is that the variety of operations needed to rework one among them into the opposite. The 3 primitive operations are:

1) Substitution: dynamical one character to a different during a word;

2) Deletion: delete one character beginning a statement;

3) Insertion: inserting one character into a word. Given a keyword w, we let Sw,d denote the set of words we have a tendency to satisfying $d(w,w\_) \leq d$ for a precise number .

F.Fuzzy Keyword Search

The fuzzy keyword set can be defined by using edit distance as follows: Given a collection of n encrypted data files C = (F1, F2. . . FN) stored in the cloud server, a set of distinct keywords W = {w1, w2, …,wp} with predefined edit distance d, and a searching input (w, k) with edit distance k ($k \leq d$),the execution of fuzzy keyword search returns a set of file IDs whose corresponding data files possibly contain the word w, denoted as FIDw: if $w = wi \in W$ belongs to W, then return FIDwi ; otherwise, if $w\_ \in W$ does not belong to W,then return {FIDwi}, where $ed(w,wi) \leq k$. Note that the above definition is based on the assumption that $k \leq d$. Infect, d can be different for distinct keywords and the

system will return {FIDwi} satisfying ed(w,wi) ≤ min{k, d} if correct match fails. For model, the next is the register variants after a replacement operation on the first character of keyword CASTLE: {AASTLE, BASTLE, DASTLE, · · · YASTLE, ZASTLE}.

4. Constructions Of Effective Fuzzy Keyword Search In Cloud Technology

The key graph behind our secure fuzzy keyword search is two-fold: 1) construct up fuzzy keyword sets that include not only the exact keywords however additionally those differing slightly because of minor typos, format inconsistencies, etc.;

2) Upcoming with connect in nurture reasonable and secure quick look approach for file retrieval supported the resulted fuzzy keyword sets. During this section, we are going to specialize in the primary half, i.e., building storage-efficient fuzzy keyword sets to facilitate the pointed method.

4.1 The straightforward Approach:

Before introducing our constructions of fuzzy keyword sets, we have a tendency to 1st propose a simple approach that achieves all the functions of fuzzy keyword search, that aims at providing a summary of however fuzzy search theme works.

Assume Π =(Setup(1λ), Enc(sk, •), Dec(sk, •)) could be a even coding theme, wherever sk could be a secret key, Setup(1λ) is that the setup formula with security parameter λ, Enc(sk, •) and Dec(sk, •) ar the coding and decoding algorithms, severally.

1) The theme goes as follows: we are able to begin by constructing the fuzzy keyword set Swi, d for every keyword Wi ∈ W (1 ≤ i ≤ p) with edit distance d. The intuitive thanks to construct the fuzzy keyword set of WI is to enumerate all doable values.

2) To look with w, the permitted user computes the trapdoor Tw of w and sends it to the server;

3) Important getting the search request Tw , the server compares it with the index table and returns all the doable encrypted file identifiers per the fuzzy keyword definition in section III-D. The user decrypts the came back results and retrieves relevant files of interest. This simple approach a p p a r e n t l y provides fuzzy keyword search over the encrypted files whereas achieving search privacy persecution the technique of secure trapdoors. However, this approach has serious potency disadvantages. The straightforward enumeration technique in constructing fuzzy key- word sets would introduce massive storage complexities that greatly have an effect on the usability. Recall that within the definition of edit distance, substitution, and deletion and insertion are 3 types of operations in computation of edit distance. The numbers of all similar words of WI satisfying ed(wi , w′) ≤ d for d = 1, 2 and 3 are around 2k × 26, 2k2 × 262 , and 4 k3 × 263, severally. as an example, assume there are 104 keywords within the file assortment with average keyword length 10 and d = 2. The output length of hash operate is a hundred and sixty bits. The resulted storage price for the index is 30GB. Therefore, it brings forth the demand for fuzzy keyword sets with smaller size.

4.2 Advanced Techniques for Constructing Fuzzy Keyword Sets

To provide a lot of sensible and effective fuzzy keyword search constructions with respect to each storage and search potency, we have a tendency to currently propose 2 advanced techniques to boost the simple approach for constructing the fuzzy keyword set. While not loss of generality, we are going to specialize in the case of edit distance d = one to elaborate the planned advanced techniques. For larger values of d, the reasoning is analogous. Note that each techniques are rigorously designed in such the simplest way that whereas suppressing the fuzzy keyword set, they're going to not have an effect on the search correctness, as are represented in section five.

Wildcard-based Fuzzy Set Construction within the higher than simple approach, all the variants of the keywords got to be listed not with standing operation is performed at a similar position. Supported the higher than observation, we have a tendency to plan to use a wildcard to denote edit operations at a similar position. The Wildcard-based fuzzy set of wi with edit distance d is denoted as S,• • • , S }, wherever S' denotes the set of words w,ti with t wildcards. Note every wildcard represents edit operation on wiw. The procedure for wildcard-based fuzzy set construction is shown in formula one. as an example, for the keyword CASTLE with the predetermined edit distance one, its wildcard-based fuzzy keyword set will be created as S = the entire variety of variants on CASTLE created during this means is merely thirteen + one, rather than thirteen × twenty six + one as within the higher than thorough enumeration approach CASTLE,1 once the edit distance is ready to be one. Generally, for a given keyword w with length l, the dimensions of Swi,1i are only 2l + 1 + 1, as compared to (2l + 1) × 26 + 1 obtained within the simple Approach. The larger the pre-set edit distance, the a lot of storage overhead will be reduced: with a similar setting of the instance within the simple approach, the planned technique will facilitate scale back the storage of the index from 30GB to around 40MB.

4.3 Gram-Based Fuzzy Search

There are recent studies to support e client fuzzy string search discrimination grams [7, 1,8, 10, 15,16, 13, 12, 11, 20, 6]. A gram of a string could be a substring that may be used as a signature for economical search. These algorithms answer a fuzzy question on group of strings discrimination the subsequent observation: if a string are with in the collection is parallel to the question string, then must to share a precise variety of common grams with the question string. This "count filter" will be wont to construct gram inverted lists for string ids to support economical search. We have a tendency to evaluate a number of the representative algorithms. The results showed that, not astonishingly,

they're not as economical as trie-based incremental-search algorithms, chiefly as a result of it's dangerous to try to to progressive computation on gram lists, particularly once a user varieties during a comparatively short pre fix, and count filtering doesn't offer enough pruning power to eliminate false positives.

5. Efficient Fuzzy Keyword Schemes

As shown in section four, the dimensions of fuzzy keyword set is greatly reduced persecution the planned advanced techniques. However, the higher than constructions introduce another challenge: the way to generate the search request and the way to perform fuzzy keyword search? within the simple approach, as a result of the index is formed by enumerating all of fuzzy words for every keyword, there continuously exists matching words for the search request as long because the edit distance between them is equal or but d. to style fuzzy search schemes supported the fuzzy keyword sets created from wildcard-based or gram-based technique, we have a tendency to reckon the looking request relating to (w, k) as w′ ∈ Sw,k , where Sw,k = is generated within the same means as within the fuzzy keyword set construction. during this section, we are going to show the way to come through fuzzy keyword search supported the fuzzy sets created from the planned advanced techniques. For simplicity, we are going to solely think about the mounted d in our theme styles. during this section, we have a tendency to begin with some intuitive solutions, the analysis of which is able to encourage U.S.A. to develop a lot of economical ones.

5.1 The instinctive Solutions

Based on the storage-efficient fuzzy keyword set created as higher than, efficient thanks to understand fuzzy keyword search is to use the standard listing approach. Specifically, the theme goes as

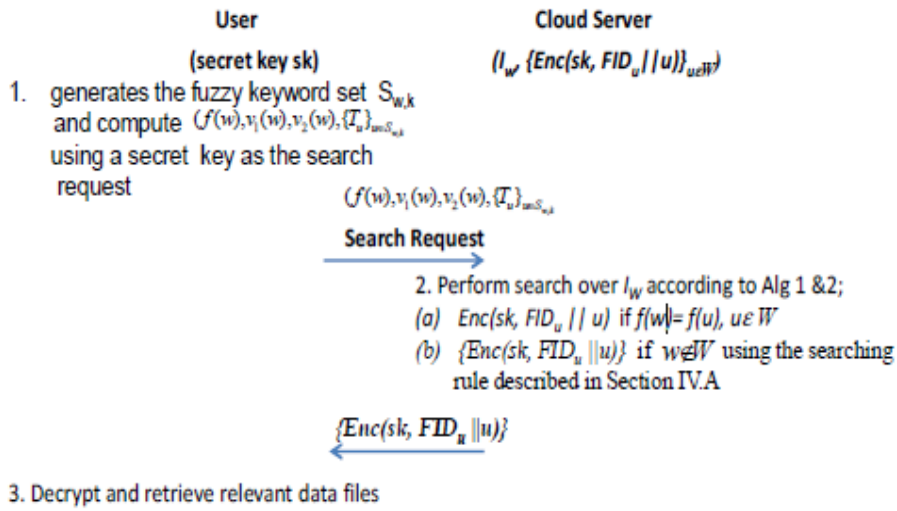

Fig 2: Protocol for the symbol – based trie – traverse fuzzy keyword search

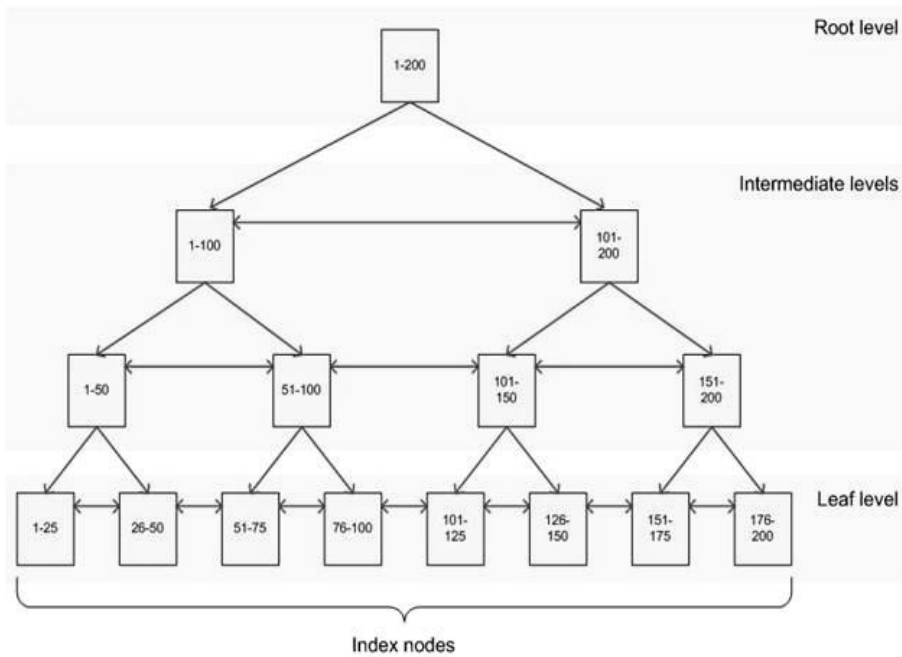

Fig 3: An Example of Integrated symbol – based index for all words in the fuzzy keyword set

5.2 The Symbol-based Trie-Traverse Search theme

To enhance the search potency, we have a tendency to currently propose a symbol-based trie-traverse search theme, wherever a multi-way tree is made for storing the fuzzy keyword set {S WI , d } i wi∈W over a finite image set. The key plan behind this construction is that every one trapdoor sharing a typical prefix might have common nodes. the basis is related to empty set and also the symbols during a trapdoor will be recovered during a search from the basis to the leaf that ends the trapdoor. All fuzzy words within the trie will be found by a depth-first search. Assume Δ = could be a predefined image set, wherever the amount of various symbols is | Δ |= 2n, that is, every image a Δ will be denoted by n bits. The scheme, as represented in Fig. 2, works as follows:

5.3 Supporting Multiple Users

In this section, we have a tendency to think about a natural extension from the previous single-user setting to multi-user setting, wherever an information owner stores a file assortment on the cloud server connect in handling permits an arbitrary cluster of users to look over his file assortment. Let BE = (Key GenBE , EncBE , DecBE) be a broadcast coding theme providing revocation-scheme security against a coalition of all revoked to boot, let π be a pseudo-random permutation. The index computation is nearly a similar because the single-user setting aside from every trapdoor Tw , a pseudo-random permutation π(ξ, •) is applied with a secret key ξ that is encrypted with the printed encryption theme and keep on the server. to look with (w, k), a licensed user computes trapdoors w′ Sw,k with a secret key ξ that is distributed by the information owner. Upon receiving the request, the server recovers the trapdoors by computing π−1 (ξ,π(ξ,Tw′)). As a result of the key ξ presently used is merely glorious by the server and also the set of presently approved users, the search request is valid on condition that the user isn't revoked. on every occasion a user is revoked, the information owner picks a brand new ξ and stores it on the server encrypted such solely non-revoked users will decode it. When the update, the server can use the new ξ to reckon π−1 (ξ, •) for following search requests. Moreover, the revoked users cannot recover this ξ and so, their requests won't yield valid trapdoors when the server applies π−1 (ξ, •).

6. Verifiable Fuzzy Keyword Search Scheme

In this section, we have a tendency to gift the planned theme in details and also the security analysis. Our theme consists of 5 algorithms (Key Gen; Build index; trapdoor; search; verify).

6.1 Construction of the VFKS theme

Key generation:

In this method, the user generates the index generation key and document coding key. The Key info could be a randomised key generation formula that generates the keys during this way: sk; sk0:
R f0;1g:

Build directory:

In this method, we have a tendency to incorporate the generation technique of the symbol-based index in and also the creation means of looking tree in to construct a brand new symbol-based tree GW. The key plan behind this construction is that every one trapdoor sharing a typical prefix might have common nodes. The basis is related to empty set and also the symbols during a trapdoor will be recovered during a search from the basis to the leaf that ends the trapdoor. All fuzzy keywords within the tire will be found by a depth-first search. Assume D = faig could be a pre outlined image set, wherever the amount of various symbols is jDj = 2n, that's every image ai two D will be denoted by n bits. The formula works as follows:

(1) Initialization: – The user scan the D and build W, the set of distinct keywords of D.– The user source the coding document assortment D to the server and receive the identifiers of every document(denote as IDfFig). For all document of containing the keyword WI, denote the symbol set as IDwi = IDfF1gkIDfF2g:::;kID

(2) Build fuzzy set trapdoor– for every keyword WI 2W, construct the fuzzy keyword set Swi;d with the wildcard-based technique g.

(3) Build symbol-based index tree: – produce a depth of l=n full 2n-binary tree GW, wherever every node contains 2 attributes (r0;r1) = (null; null) well and l is that the out length of hash operate f(x). --For every fuzzy keyword w0i two southwest i, d, divided T into l=n elements, every n-bits hash price represents an emblem in four. swing all the sequence of image filing into the G Wand appending the corresponding symbol IDwikgk (IDwi) to the leaf node. Search

– Upon receiving the search request, the server divides every Tw0 into a sequence of symbols;
– Performs the search over GW discrimination formula represented in Fig two and returns the ID WI and proof to the user.
– per the symbol, the user will get the interest documents. Verify
– check whether or not the amount of received proof is equals to the amount of sent trapdoor.
– input the Tw and also the corresponding proof, per the formula take a look at whether or not the server is honest.

6.2 Security analysis

Data Privacy:

During this work, we have a tendency to think about solely search privacy; as a result of privacy of the documents will be ensured by the coding formula. That is, we have a tendency to specialized in the confidentiality of the search request and also the index T. discrimination the trapdoor technology, the aggressor on to get the plaintext is

not possible from the cipher text. There fore we have a tendency to principally concern the confidentiality of the index T.

Verifiable Searchability:

We have a tendency to assume k steps ar performed by the server. If "No" came back, we'd understand that the primary k one characters ar matched whereas Tw [k] is mismatched, that may well be represented by a k bit binary sequence b = (1;:::;1;0); if "Yes" is came back, b = (1;:::;1;1). In our theme, firstly, we have a tendency to check the amount of proof whether or not is up to the causing trapdoors. If not, we are able to say the server isn't creating a full search. If pass, we have a tendency to exploit a sampling technique to see. for every of Proof to be tested, similarly, ranging from the last (or k-th) step, if "Yes", verify checks the integrity of the concatenation of the document identifiers by computing a keyed hash of it and scrutiny with the received one. In fact, the completeness of the search outcome is examine adhere. If the server returns a fraction of the search outcome, the user will notice the server isn't honest. Then we have a tendency to take a look at that whether or not the trapdoor equals the received image string of the proof. After that, j weakened by one. If "No", the higher than step is skipped. Next, verify validates the correctness of the search outcome by decrypting the primary a part of Tj qj[r1] = Enc(Tj;qj[r0]; parent (Tj;qj)[r1]) to urge (x; y) and testing whether: (1) Tw[j] equals x (2) Tw[j 1] equals y. To cheat the search results, the server ought to forge the proof. There ar 2 doable case: (1) the server honestly seek for a fraction of trapdoors and forge the proof or alternative trapdoors, at worst, the server don't any search; (2) the server forge the proof per the received trapdoor. For the case (1), the server should will generate a valid r1 with a special alphabetic characterˆ =6 mem, think about the opponent don't understand the sk, it will be seen a random oracle. just in case (2), the opponent might use the r1 of another node, note that every node includes a international distinctive r1, which is able to be rejected by verify. Additionally, the argument higher than will be applied recursively to the (j 1)th step in verify and then on.

7. Performance Analysis

We conducted an intensive experimental analysis of the planned techniques on actual knowledge set: the fashionable 10 years 'IEEEINFOCOM publications. the information set includes concerning two,600 publications. we have a tendency to put off the words within the paper titles to construct the core keyword set in our experiment. the entire variety of keywords is3, 262 and their average word length is7.44. Our experiment is conducted on a Linux machine with Intel Core 2processor running at one.86GHz and 2G DDR2-800 memory. The performance of our theme is evaluated relating to the time price of fuzzy set construction, the time and storage price of index construction, the search time of the listing approach and also the symbol-based trie-traverse approach.

Fig 4 shows the index tree construction time (measured in terms of ms)

Fig 5 shows the index storage cost (measured in terms of Kbytes) when we use different numbers of Infocom publication titles (and hence different number of distinct Keywords).

Table Fuzzy Keyword Search Using Symbol – based Trie - traversed Approach (Searching time, query word and returned word status) Our search time is larger (but still small) with single fuzzy keyword search but the search time for our approach tends to be better for fuzzy keyword queries where many encrypted file identifiers for each word are returned using the keyword approach

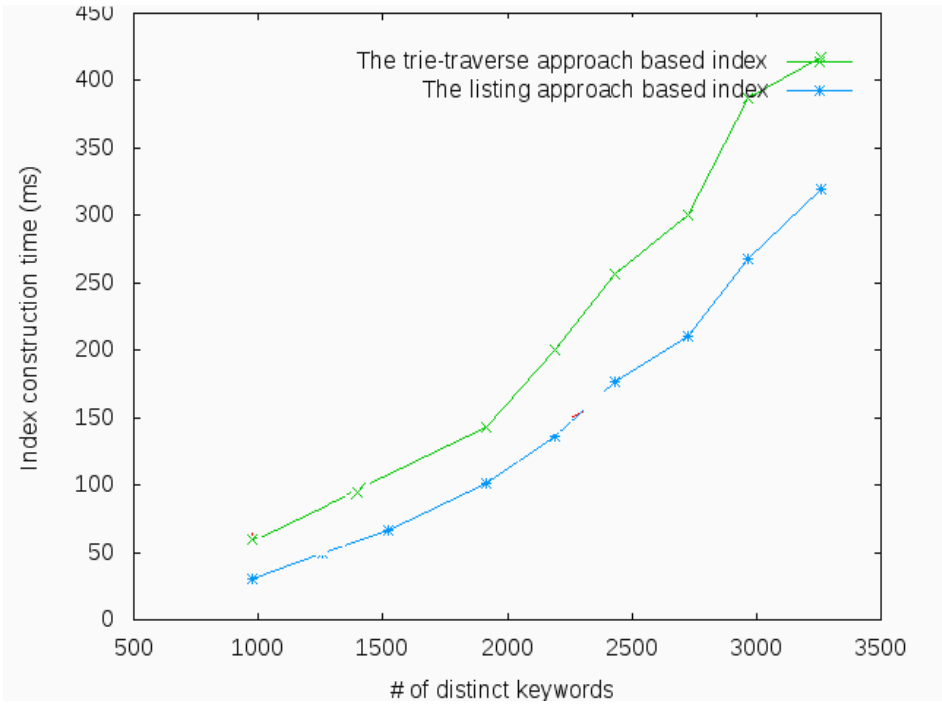

Fig 4: Index Construction Time for our Wildcard Technique approaches with edit distance d=1

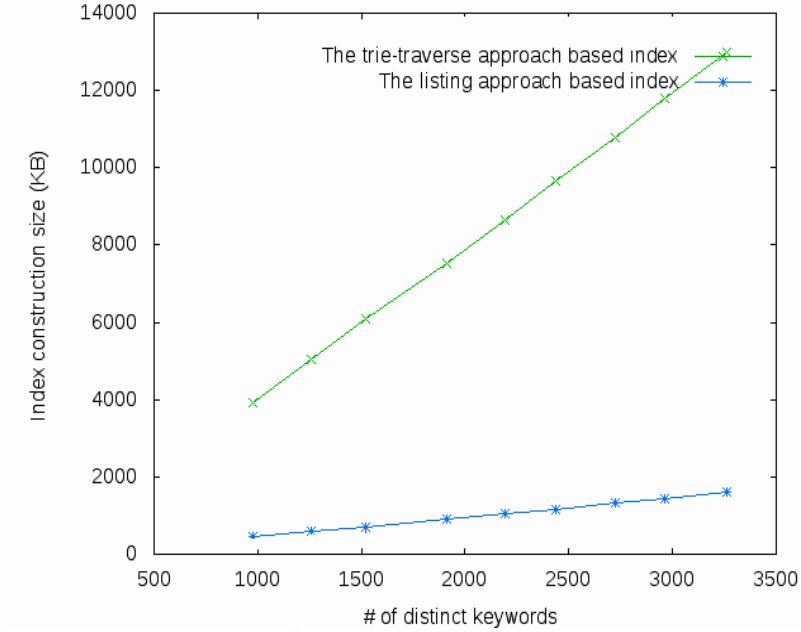

Fig 5: Index Storage Size (Kbytes) for our Wildcard Technique approaches with edit distance d=1.

Query Word	***Searching Time***	***Returned Words***
Programming	79.11 usec	Programming
Internet	514.7 usec	Internet
Wireless	2312.4 usec	Rulsets,wireless
Delay	552.3 usec	Delav,delay
Power	1300.2 usec	power
Data mining	635.4 usec	Data mining
Proxy set	215.32 usec	proxy set

Table : Fuzzy Keyword Search Using Symbol – based Trie - traversed Approach.

7.1 Performance of Fuzzy Keyword set Construction:

In section four, we have a tendency to propose 2 advanced techniques for the development of fuzzy keyword sets, that each will be utilized in our planned fuzzy search schemes. In our experiment, we have a tendency to solely specialize in the wildcard-based fuzzy set construction as a result of it provides the sound results compared to the gram-based fuzzy set construction as mentioned insection6.Fig.4showsthe fuzzy set construction time by discrimination the wildcard-based approach with edit distanced=1and2.We can see that in each cases, the wildcard-based approach is extremely economical and also the construction time will increase linearly with the amount of keywords. the price of constructing fuzzy keyword set below d=1 is far but the case of d=2 because of the smaller set of doable wildcard-based words.

7.2 Performance of Fuzzy Keyword Search

Efficiency of Index Construction Given the fuzzy keyword set created persecution wildcard- primarily based technique, we have a tendency to live the time price of index construction for the listing approach and symbol-based trie-based approach. In our experiment, we have a tendency to chosen d=4 and use SHA-1as our hash operate with output length of l=160bits.The resulted height of the looking tree is l/n=40. The index construction time for edit distance d=1 and d=2. the same as the fuzzy keyword set construction, the index construction time additionally will increase linearly with the amount of distinct keywords. Compared to the listing approach, the index construction of the trie-traverse approach includes the method of building the looking tree to boot, so its time price is larger than that of listing approach. However, the entire index construction method is conducted off-line, so it'll not have an effect on the looking potency. Table 1shows the index storage price of the 2 approaches. The symbol-based trie traverse approach consumes a lot of cupboard space than the listing approach because of its multi-way tree structure. This extra storage price, however, isn't a main issue in our setting, in and of itself index info solely take up low quantity of cupboard space on the cloud server.

8. Conclusion & Future Work

In this paper, for the primary time we have a tendency to formalize and solve the theme of supporting efficient however privacy protective fuzzy seek for achieving effective utilization of remotely keep encrypted knowledge in

Cloud Computing. We have a tendency to style 2 advanced techniques (i.e., wildcard-based and gram- primarily based techniques) to construct the storage-efficient fuzzy keyword sets by exploiting 2 important observations on the similarity metric of edit distance. supported the created fuzzy keyword sets, we have a tendency to any propose a novel image primarily based trie-traverse looking theme, wherever a multi-way tree structure is constructed up persecution symbols remodeled from the resulted fuzzy keyword sets. Through rigorous security analysis, we have a tendency to show that our planned resolution is secure and privacy- protective, whereas properly realizing the goal of fuzzy keyword search. Intensive experimental results demonstrate the potency of our resolution.

As our ongoing work, we will continue to research on security mechanisms that support Search semantics that takes into consideration conjunction of keywords, sequence of keywords, and even the complex natural language semantics to produce highly relevant search results. Search ranking that sorts the searching results according to the relevance criteria.

References

[1]Google, "Britney spears spelling correction", www.google.com/jobs/britney.html, June 2009

[2] M. Bellare, A. Boldyreva, and A. O. Neil,"Determinstic and efficiently searchable encryption", Proceedings of Crypto 2007, LCNS, Vol 4622, 2007.

[3] D. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data", Proceedings of IEEE Security and Privacy, 2000.

[4] E. J. Goh, "Secure indexes", Cryptology ePrint Archive, Report 2003/216, 2003.

[5] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search", Proceedings of EUROCRYP,04, 2004

[6] B. Waters, D. Balfanz, G. Durfee, and D. Smetters, "Building an encrypted and searchable audit log", Proceeding of 11th Annual network and Distributed System, 2004

[7] R. Cutmola, J. A. Garay, S. Kamara, R. Ostrovsky, "Searchable symmetric encryption: improved definition and efficient constrcutions", Proceedings of ACM CCS, 2006

[8] B. Boneh, B. Waters, "Conjunctive, subset and range queries on encrypted data", Proceedings of TCC, 2007 pp 535-554

[9] J. Li et al, "Fuzzy Keyword Search over Encrypted Data in Cloud Computing", Proceedings of IEEE Infocom, April, 2010

[10] N. Cao et al, "Privacy-Preserving Multi-keyword Ranked Search over Encrypted Cloud Data", to appear IEEE Infocom, 2011.

[11] S. Zerr et al, "Zerber: r-confidential indexing for distributed documents", Proceedings of EDBT, 2008 pp 287-298.

[12] Z. Zhang et al, "Bed-Tree: An All Purpose Index Structure for String Similiarity Search Based on Edit Distance", Proceedings of ACM SIGMOD 2010.

[13] Microsoft Academic Search http:// academic.research.microsoft.com, 2009.

[14] J. Diederich, W.T. Balke, "The Semantic GrowBag Algorithm: Automatically Deriving Categorization Systems", Proceedings of ECDL, 2007.

[15] C. Li, J. Lu, Y. Lu, "Efficient merging and filtering algorithms for approximate string searches", Proceedings of ICDE, 2008

[16] A. Behm, S. Ji, C. Li, J. Lu, "Space-constrained gram-based indexing for efficient approximate string search", Proceedings of ICDE, 2009

[17] S. Ji, G. Li, C. Li, J. Feng, "Efficient interactive fuzzy keyword search",Proceedings of ACM WWW, 2009.

[18] Y. C. Chang, M. Mitzenmacher, "Privacy preserving keyword searches on remote encypted data", Proceedings of ACNS, 2005.

[19] E. Shi, J. Bethencourt, T. H. H. Chan, D. Song, A. Perrig, "Multidimensional range query over encrypted data", IEEE Symposium on Security and Privacy, 2007.

[20] Y. Liu, A. Niculescu-Mizil, W. Gryc, "Topic-Link LDA: Joint Models of Topic and Author Community", Proceedings of International Conference on Machine Learning, June, 2009.

[21] Ning Cao, Cong Wang, Ming Li, KuiRen, Wenjing Lou. April 2011. Privacy Preserving Multi-keyword Ranked Search over Encrypted Cloud Data, in Proc. of IEEE INFOCOM11.

[22] Sameer Rajan, ApurvaJairath. Cloud Computing. 2011. The Fifth generation of Computing, International Conference on Communication Systems and Network Technologies.

[23] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou. Fuzzy keyword search over encrypted data in cloud computing. In Proc. of the 29th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'10), San Diego, CA, USA, pages 1–5. IEEE, March 2010.

[24] F. Bao, R. Deng, X. Ding, and Y. Yang, "Private query on encrypted data in multi-user settings," in Proc. Of ISPEC'08, 2008.

[25] J. Feigenbaum, Y. Ishai, T. Malkin, K. Nissim, M. Strauss, and R. N.Wright "Secure multiparty computation of approximations," in Proc. of ICALP'01.

[26] R. Ostrovsky, "Software protection and simulations on oblivious rams," Ph.D dissertation, Massachusetts Institute of Technology, 1992.

[27] V. Levenshtein, "Binary codes capable of correcting spurious insertions and deletions of ones," Problems of Information Transmission, vol. 1,no.1, pp. 8–17, 1965.

