Multi-keyword Ranked Search Over Encrypted Cloud Data Supporting Synonym Query

Siddheshwar S. Metkari[1], Dr. S. B. Sonkamble[2]

[1, 2]Department of Computer Engineering, JSPM's Rajarshi Shahu School of Engineering & Research, Narhe, Pune, Maharashtra, India

Abstract: ***Cloud storage becomes more and more popular in recent years due to its benefits such as scalability, availability, low cost service over traditional storage solutions. Organizations are motivated to migrate their data from local site to central commercial public cloud server. By outsourcing data on cloud users gets relief from storage maintenance. Although there are many benefits to migrate data on cloud storage it brings many security problems. Therefore the data owners hesitate to migrate the sensitive data. In this case the control of data is going towards cloud service provider. This security problem induces data owners to encrypt data at client side and outsource the data. By encrypting data improves the data security but the data efficiency is decreased because searching on encrypted data is difficult. The search techniques which are used on plain text cannot be used over encrypted data. The existing solutions supports only identical keyword search, semantic search is not supported. In the paper we proposed semantic multi-keyword ranked search system. To improve search efficiency this system includes semantic search by using WordNet library. Vector space model and TF-IDF model is used for index construction and query generation.***

Keywords: Cloud data, TF-IDF, Searchable encryption, Multi-keyword Search

1. Introduction

In today's data intensive world, cloud computing is new type of computing paradigm which enables sharing of computing resources over the internet. The cloud characteristics are on-demand self-service, location independent network access, ubiquitous network access and usage based pay. Due to this charming features private and public organization are outsourcing their large amount of data on cloud storage. Organization can purchase only needed amount of storage from CSP to fulfil their data storage need instead of maintaining their own data storage. The data owner is relieved from purchasing hardware and software to manage data themselves. Instead of these tremendous advantages, the privacy and security concerns are preventing organizations to utilize these advantages. The data owner and the cloud server are not in same trusted domain. Security of remotely stored data is a big concern because user lost his physical control on data. The control is in the service provider's hand. The data is not secured as well as there are many attacks by internal external attackers. The valuable data such as social security number, email, Personal health records and organizations financial information must be stored securely. Solution is encryption of data at client side before outsourcing. But if you encrypt data the searching over chipper text is challenging. The existing search techniques are only applied on plain text data. The trivial solution of downloading all the data and decrypting locally is clearly impractical due huge amount of bandwidth cost in cloud scale system. Searchable encryption allows storing data in encrypted format and you can apply keyword search over chipper text data.

A semantic secure multi-keyword search scheme over encrypted cloud data is proposed in this paper. The semantic search is not only support exact keyword matched or structure matched but also supports the real intent of user search. The relevance score between documents and query keywords is calculated and files are returned in ranked order. Vector space model (VSM) and TF*IDF model is used for the index tree generation. The Greedy Depth First Search algorithm is used on tree based index structure for efficient search.

This paper is organized as follows. In section II we discussed about related work. Details about system are discussed in Section IV. Section V discusses preliminaries and section VI summary and conclusion.

2. Related Work

Searchable encryption has been an import research area and many researchers and organizations have investigated search techniques to search on chipper text data. Searchable encryption allows storing data in encrypted format and you can apply keyword search over chipper text data. These search techniques builds searchable index tree such that its contents are hidden from server however it still allows performing document searching.

These solutions differ from each other mostly in terms of whether they allow single keyword, multi-keyword, similarity search, ranked search. By using multi-keyword ranked search user can query with multiple keywords and retrieve accurate search result. But all these search schemes does not allow synonym based queries.

Searching Techniques

a) Boolean Keyword Search over Encrypted data [5]
Ning Caoy et al. focused on Boolean Keyword Searchable Encryption. This technique has two drawbacks. In this scheme user have to processes each and every returned file to find the desired one, because user aware of a pre-knowledge of the encrypted cloud data. Second the search sends back all files which are only depend on presence or absent of query keywords. This increases unnecessary network traffic and consumes bandwidth.

b) Wildcard-based Fuzzy Set Construction (WFSC)[18]

Li, Wang, et al in proposed the "Wildcard-based Fuzzy Set Construction (WFSC)" scheme to enable fuzzy keyword search over encrypted cloud data. The key concept behind WFSC is maintaining an index that covers all possible variations of a keyword within a predefine edit distance. Instead of simply encrypting the keywords extracted from the data file, WFSC expands each extracted keyword into a set of modified keywords by inserting wildcard character into the keyword. The number of wildcard character used to modify the keyword is based on a predefine edit distance value. The drawback of this system is that with the increase in the edit distance value, the search quality is improved but there is huge increase in modified keyword set.

c) Dictionary-based Fuzzy Set (DFSC)[19]

Liu, Zhu, et al in [4] modified WFSC and proposed another scheme called "Dictionary-based Fuzzy Set Construction (DFSC)". Instead of injecting keyword with the wildcard character, DFSC uses a dictionary to pull in only the valid words that are within the range of the predefine edit distance to form the modified keyword set. The authors showed the size of the index file created by DFSC is much smaller than WFSC when the predefine edit distance increased. Even DFSC shows better storage usage than WFSC, DFSC inherited the search quality degradation problem because more unrelated keywords can still be pulled into the modified key set from the dictionary as edit distance value increases. Furthermore, DFSC does not support variations of newly invented words or keywords that contain multiple typos because they are words that cannot be found in the dictionary. This problem decreases the search coverage of DFSC and makes it less accurate than WFSC. The major weakness of previously investigated schemes is that they are not considering user real intent of search.

d) Ranked Keyword Search[4]

Cong Wang et al discuss the major disadvantages of traditional searchable encryption schemes and give the better technique of keyword search. In this technique the search results are well ranked. This technique gives user most relevant document in the relevance order with query. And user gets relief from sorting through every match in the content collection. This technique avoids unnecessary traffic. But they are only useful single keyword search

e) Multi-keyword ranked Search[6]

Cao et al is the first to define and solve privacy preserving problem of multi-keyword ranked search and establish a variety of privacy requirements.

In this scheme the dictionary words are extracted from documents. And the queries and documents are expressed as vectors. The elements in the vector are the normalized TF values. The documents in the result are ranked by matching the vector co-ordinates. The importance (weightage) of different keyword is not taken in account.

f) Secure multi-keyword search[7]

Sun et al describes secure multi-keyword search which support similarity search. In this scheme the searchable index tree is constructed by using vector space model. The cosine similarity with TF*IDF is used to find the relevant score between document and query vectors query vectors and results are returned in rank order. This algorithm search time is better than linear search but the precision is not good.

g) Secure kNN algorithm[8]

W. K. Wang focuses on query processing over encrypted cloud database. In this scheme the distance between the documents and query are computed to find the nearest neighbour to the query. In secure KNN technique first data owner encrypt each attribute of database is encrypted. And the encrypted database is stored on cloud. The authorized user who want access k closest documents to his query, encrypt the query keywords and the encrypted tokens are send to cloud server for searching. The database as well as query is encrypted therefore the query and database confidentiality is preserved.

3. Existing System

As shown in the fig system consist of following three entities

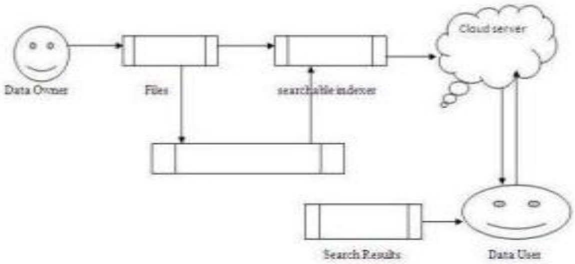

Figure 1: Search on encrypted data

- **Data Owner** has a collection of documents. He want to store these documents on cloud storage. Data owner perform preprocessing, index construction, encryption and upload of documents.
- **Data Users** wants to access these files. Using search keywords and secure key a trapdoor TD is generated. Trapdoor TD is used to search. After search processes, the cloud server sends most relevant k files to user. The retrieved documents are decrypted using secrete key.
- **Cloud Server:** Upon receiving request form user, the cloud server searches the index tree and sends back top k relevant files to user.

Disadvantages of existing system

1) Supports only identical keyword search, semantic search is not supported.
2) Data owner has to perform all pre-processing tasks
3) Index encryption is sequential.

4. Proposed System

As shown in fig the proposed system consist of four entities:- Data Owner, Web Application, Data User and Cloud Server. In this system the load data owner is reduced by adding one user application. Data pre-processing, encryption, decryption all task are done by user application

System module consists of three modules as data preprocessing, construction of searchable index tree and query processing module. Data preprocessing module performs various preprocessing functions on entire data set and generate list of terms. Construction of index tree module creates searchable index tree by using TF values of keyword from each documents. Query processing module process the query terms and search over index tree. The similarity between the terms and document is calculated and set ranked relevant documents are returned to the user.

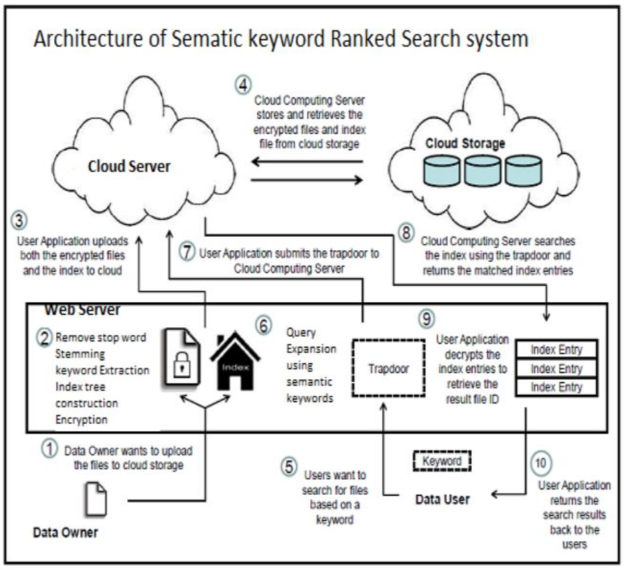
Architecture of Sematic keyword Ranked Search system
Cloud Server
4
Cloud Computing Server stores and retrieves the encrypted files and index file from cloud storage
Cloud Storage
3
User Application uploads both the encrypted files and the index to cloud
7
User Application submits the trapdoor to Cloud Computing Server
8
Cloud Computing Server searches the index using the trapdoor and returns the matched index entries
Web Server
2
Remove stop word
Stemming
keyword Extraction
Index tree
construction
Encryption
Index
6
Query Expansion using semantic keywords
Trapdoor
9
User Application decrypts the index entries to retrieve the result file ID
Index Entry
Index Entry
Index Entry
1
Data Owner wants to upload the files to cloud storage
Data Owner
5
Users want to search for files based on a keyword
Keyword
Data User
10
User Application returns the search results back to the users

Figure 2: Proposed Architectural Design

Mathematical Model

We consider the set theory to deal with the system S

System

S = {s, e, X, Y,F$_s$,SS,FS| Φ$_s$ }

Notation

s= Start state. Preprocessing of entire document collection**.**

DC- Set of n original text files, *DC*={d| $d_{1,}d_{2,}d_{3....}d_n$)

f_{dp}={f_l,f_s,f_{st}}

$f_{dp}$is a document collection preprocessing function which consist of various functions, lexical analysis (f $_l$), stopword removal (f_s) and stemming (f_{st})

f_{dp}={DC}→{T}

{T} is the list of terms or keywords generated after preprocessing. The weight of each term is calculated using following equation.

Wieght=*tf*idf*

Where tf is a term frequency of term and idf is the inverse document frequency of term

W- Set of m keyword dictionary, W= {w| $w_{1,}$ $w_{2,}$ $w_{3...,}w_m$} after preprocessing

I - Searchable Keyword balanced binary(KBB) Index tree built form all files DC.

I - Encrypted index tree generated from I.

C- Set of n chipper text files after encryption,

C={c| $c_{1,}$ $c_{2,}$ $_{...,}c_n$)

X = Input of the system

Here X is the search request or query entered by the user.

X=Q;

k- No. of expected relevant documents. For the query same document preprocessing functions are applied.

Y = Output of the system

Y ⊆ *DC*={ $d_{1,}$ $d_{2,...,}$ d_k}

SS= Success state

Set of relevant documents such that

$Relv_{d1,Q}$>= , $Relv_{d1,Q}$>= , >=, $Relv_{dk,Q}$

FS= Failure state

Failure state occurs when there are no documents presents on cloud storage for the search request or query.

Fs= Functions of the system

={data_Preproecssing(),
generate_Searchable_Index_Tree(),encryption(),
upload_doc_indextree_to_cloudserver(),
genQuery(),search()}

5. Preliminaries

A. Data preprocessing method.

The document preprocessing performs all the text operations on the dataset and extract meaningful keywords. f_{dp}={f_l,f_s,f_{st}} is a data preprocessing function consist of various functions, lexical analysis (f $_l$), stop word removal (f_s) and stemming (f_{st}).

The term frequency and inverse document frequencies are calculated to compute the weight of each term. On the basis of the weight, the top terms are selected for creation the dictionary keyword set.

$$W_{ik} = TF * IDF = TF * \frac{1}{DF} = f_{ik} * \log\frac{n}{n_k} \qquad (1)$$

Here

f_{ik}=Frequency of term i in document.

n_k=no. of documents which contains term i.

keyword set is generated by extracting important keywords from all documents.

B. Index tree construction.

In the index construction process first the tree node is generated for each document. These nodes are the leaf nods of the tree. Form these leaf nodes the internal nodes are generated. The algorithm 1 gives a detail of index construction process.

Algorithm 1GenerateIndexTree I

Procedure GenerateIndex(F)

```
for each file $f_{FID}$ in $F$
    Generate leaf node $u$ for $f_{FID}$,
    $v.ID$ =GenerateID(), $v.Pl = v.Pr= null$, $v.FID= FID$,
    $D[j] = TF_{fFID},w_i$ for j = 1, ...,$m$;
    Add node v to $CurNodeSet$;
end for
while( |Cur$NodeSe\ t$|$>1$)
    for each node $CNode$ in CurNodeSet
        if C$Node$ is last node in $CurNodeSet$ then
            $LeftNode$=last node from TempNodeSet
            $RightNode$=CurrentNode;
            Remove last node from TempNodeSet
        else
            $LeftNode$=CurrentNode
            $RightNode$=Next node from CurrentNodeSet;
        end if
        Generate a parent node $u$ formpair of nodes,
        $u:ID$ = GenerateID(), $v.P_l = LeftNode$,$v.Pr=RightNode$,
        $v.FID$=0 and $D[j] = max\{LeftNode.D[j],RightNode.D[j]\}$,
        for each $k$ = 1,...,$m$;

        add node v to $TempNodeSet$;
```

```
end for
Replace CurNodeSetwith TempNodeSetand
make empty TempNodeSet;
end while
 return the one node left in CurNodeSet, which is a root of tree T;
```

C. Efficient Ranked Search Scheme

There are four algorithms which are described as follows

- Generate secrete key

SK=GenSK(S,M1,M2)

Where

Sk- Secrete key in the form of a 3-tuple.

S- Splitting indicator, Randomly generated m-bit vector.

M1,M2- (m*m) random invertible matrices. These are used for encryption.

- *I* = encryption(I, sk)

To generated Encrypted index tree from the Tree generated by algorithm 1. The index vector D_v of each node in the tree is splitted two random vectors $\{D_{v1},D_{v2}\}$ .The spiltting procedure is expressed as follows. If S[j]=0, $D_v[j]=D_{v1}[j]=D_{v2}[j]$.

if S[j] = 1, $D_{v1}[j]$ and $D_{v2}[j]$ are set randomly and $D_v[j]=D_{v1}[j]+ D_{v2}[j]$. Now each node in the encrypted index tree contains two vectors as follows,

$I_v=\{M_1^TD_{v1,}M_2^TD_{v2}\}.$

- TD<-Gen_Trapdoor(W_q, Sk)

From user entered query keywords trapdoor is generated. The query is also splitted in two random vector $Q_1$and Q_2as follows. If s[j]=0, $Q_1[j]+Q_2[j]= Q[j]$. and if s[j]=1, $Q_1[j]=Q_2[j]=Q[j]$. The trapdoor is

$TD=\{M_1^{-1}Q_{1,}M_2^{-1}Q_2\}.$

- *RelevanceScore<-RelScore(Iu,TD)*

The cloud uses search algorithm 2. Find relevance score between each node and query. If tree node has relevance score less than k-th score in RList this sub tree is not visited. The Relevance Score between document and query is calculated by following formula

$$I_v \,.\, TD$$
$$= (M_1^TD_{v1}).\,(M_1^{-1}Q_1)+(M_2^TD_{v2}).(M_2^{-1}Q_2)$$
$$=(M_1^TD_{v1})^T.\,(M_1^{-1}Q_1)+(M_2^TD_{v2})^T.(M_2^{-1}Q_2)$$
$$=(M_1D_{v1}^T).\,(M_1^{-1}Q_1)+(M_2D_{v2}^T).(M_2^{-1}Q_2)$$
$$= D_{v1}^T M_1 M_1^{-1} Q_{1+} D_{v2}^T M_2 M_2^{-1}Q_2$$
$$= D_{v1}^T \,.\, Q_1+ D_{v2}^T \,.\, Q_2$$
$$=D_v \,.\, Q$$
$$=RelScore(D_v , Q) \quad (5)$$

D. Search Process

Semantic keyword expansion:- The query keywords are expanded by using WordNet Dictionary and library. The expanded query is encrypted and used send to cloud for searching.

Security Analysis Following are the privacy requirements of the system.

1) Index and Query confidentiality:
2) In this scheme Iu and TD and splitted and encrypted form of original vector Du and Q. The random invertible matrices are used for encryption. The attacker of the COA is not able to calculate the matrices from the chipper text.
3) Query Unlinkability: The trapdoor generated is also randomly splitted therefore the search request is generated different and therefore the Unlinkability is protected.
4) Keyword Privacy: The query keywords are encrypted by random splitting method and encrypted with random matrices. Therefore no statistical information is leaked about keywords.

6. Experimental Setup and Results

To carry out the experiment we have installed Ubuntu 14.04 and web server.

Index construction efficiency

This process has two steps 1. Creating a plain index tree form document collection 2. Encrypt the index tree by first splitting and two multiplications of (m*m) matrix.

Total nodes generated are O (n) splitting operation time is O(m) time two multiplication requires O(m2) time. Total time is O (nm2). Most of time required for multiplication So I parallelized this process using threads and the multiplication times drastically changed. Fig shows the index tree building cost compare the time cost of base paper and mine.

Fig Index tree construction cost for different sizes of document collection with the fixed dictionary m=4000

	Time of index tree construction(*10^2) s	
No of documents in the collection(*10^2)	Base Paper (Sequential)	Proposed System (Parallel)
5	5	1.59
10	10	2.15
15	15	2.72
20	20	3.23
25	25	3.78
30	30	4.34
35	35	4.91

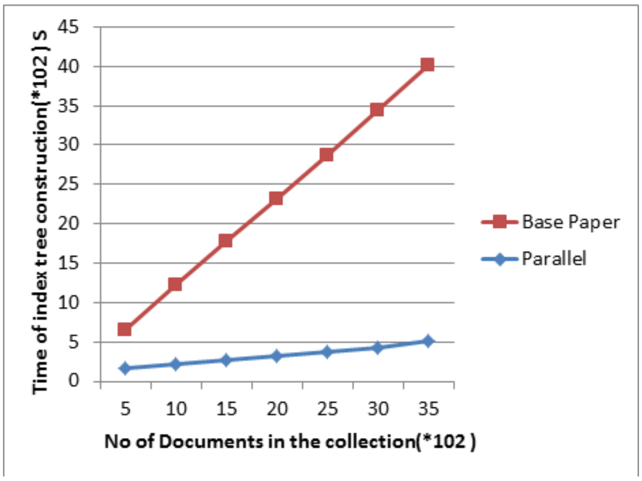

Figure 2: Index tree construction cost for different sizes of document collection with the fixed dictionary, m=4000

7. Conclusion and Future Work

In this scheme, a Semantic Multi-keyword Ranked Search scheme over encrypted cloud data with indexing is designed and implemented which meanwhile supports semantic search. The keyword extraction method is used to select meaningful dictionary keywords. The widely used TF*IDF and Vector space model is used to calculate relevance score between query and documents. Taking security and privacy into consideration, we employ a secure splitting k-NN technique to encrypt the index and the queried vector, so that we can obtain the accurate ranked results and protect the confidence of the data well. Our solution could return not only the exactly matched files, but also the files including the terms semantically related to the query keyword. For fast index tree generation we parallelized index construction using multi-threading. Experimental results demonstrate the efficiency of our proposed scheme.

References

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

[2] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in Security and Privacy, 2000. S&P 2000. Proceedings. 2000 IEEE Symposium on. IEEE, 2000, pp. 44– 55.

[3] Ning Caoy, Cong Wangz, Ming Liy, Kui Renz, and Wenjing Louy" Privacy-Preserving Multi-keyword Ranked Search over Encrypted Cloud Data" INFOCOM, 2011 Proceedings IEEE.

[4] Cong Wang†, Ning Cao‡, Jin Li†, Kui Ren†, and Wenjing Lou"Secure Ranked Keyword Search over Encrypted Cloud Data" Distributed Computing System,2010 IEE 30th international conference.

[5] B. Wang, S. Yu, W. Lou, and Y. T. Hou, "Privacy-preserving multi-keyword fuzzy search over encrypted data in the cloud," in IEEEINFOCOM, 2014.

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

[7] W. Sun, B. Wang, N. Cao, M. Li, W. Lou, Y. T. Hou, and H. Li, "Privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking," in Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security. ACMs, 2013, pp. 71–82.

[8] W. K. Wong, D. W.-l. Cheung, B. Kao, and N. Mamoulis, "Secure knn computation on encrypted databases", in Proceedings of the 2009 ACM SIGMOD International Conference on Management of data. ACM, 2009, pp. 139152

[9] P. Golle, J. Staddon, and B. Waters, "Secure conjunctive keyword search over encrypted data," in Applied Cryptography and NetworkSecurity. Springer, 2004, pp. 31–45.

[10] Y. H. Hwang and P. J. Lee, "Public key encryption with conjunctive keyword search and its extension to a multi-user system," in Proceedings of the First international conference on Pairing-Based Cryptography. Springer-Verlag, 2007, pp. 2–22.

[11] L. Ballard, S. Kamara, and F. Monrose, "Achieving efficient conjunctive keyword searches over encrypted data," in Proceedings of the 7th international conference on Information and Communications Security. Springer-Verlag, 2005, pp. 414–426.

[12] C. Orencik, M. Kantarcioglu, and E. Savas, "A practical and secure multi-keyword search method over encrypted cloud data," in Cloud Computing (CLOUD), 2013 IEEE Sixth International Conference on. IEEE, 2013, pp. 390–397.

[13] W. Zhang, S. Xiao, Y. Lin, T. Zhou, and S. Zhou, "Secure ranked multi-keyword search for multiple data owners in cloud computing," in Dependable Systems and Networks (DSN), 2014 44th Annual IEEE/IFIP International Conference on. IEEE, 2014, pp. 276–286.

[14] S. Kamara, C. Papamanthou, and T. Roeder, "Dynamic searchable symmetric encryption," in Proceedings of the 2012 ACM conferenceon Computer and communications security. ACM, 2012, pp. 965–976.

[15] S. Kamara and C. Papamanthou, "Parallel and dynamic searchable symmetric encryption," in Financial Cryptography and DataSecurity. Springer, 2013, pp. 258–274.

[16] J. Katz, A. Sahai, and B. Waters, "Predicate encryption supporting disjunctions, polynomial equations, and inner products," in Advances in Cryptology–EUROCRYPT 2008. Springer, 2008, pp. 146–162.

[17] E. Shen, E. Shi, and B. Waters, "Predicate privacy in encryptionsystems," in Proceedings of the 6th Theory of Cryptography Conferenceon Theory of

[18] Li, J., Wang, Q., Wang, C., Cao, N., Ren, K., Lou, W."Fuzzy Keyword Search overEncrypted Data in Cloud Computing, "in INFOCOM, 2010 Proceedings IEEE, pp 1-5.

[19] Liu, C., Zhu, L., Li, L., Tan, Y. "Fuzzy keyword search on encrypted cloud storage data with small index, "Cloud Computing and Intelligence Systems (CCIS), 2011 IEEE International Conference on, 2011, pp 269-273