IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 9, September 2016

Dynamic Multi-Keyword Ranked Search Based Security Mechanism for Cloud Data Management

Sinu V Agnes[1], Dr. K. Thamodaran[2]

M.Phil Research Scholar, Dept of Computer Science, Marudupandiyar College, Thanjavur, Tamilnadu, India[1]

Professor, Dept of Computer Science, Marudupandiyar College, Thanjavur, Tamilnadu, India[2]

Abstract: Cloud computing means saving and retrieving data and programs over the internet instead of your computer's hard drive. The cloud is just a representative for the Internet. It goes back to the days of flowcharts and presentations that would represent the huge server-farm infrastructure of the Internet as nothing but a puffy, white cumulus cloud, accepting connections and distributing out information as it floats. Cloud computing has become a social fact used by most people every day. As with every important social fact there are issues that limit its widespread adoption. Most issues start from the fact that the user loses control of his or her data, because it is stored on a computer belonging to someone else. This happens when the owner of the remote servers is a person or organization other than the user; as their interests may point in different directions. In this paper, a secure multi-keyword ranked search scheme on encrypted cloud data is devised along with dynamic update operations of documents. Scrupulously, the vector space model and the extensively-used TF x IDF model are combined in the index construction and query generation. The special tree-based index structure is formed and propose a Greedy Depth-first Search algorithm to make available efficient multi-keyword ranked search. The secure kNN algorithm is effectively used to encrypt the index and query vectors, and meanwhile provide accurate relevance score calculation between encrypted index and query vectors. In order to withstand statistical attacks, phantom terms are added to the index vector for blinding search results. Due to the use of our special tree-based index structure, the proposed scheme can achieve sub-linear search time and deal with the updating of documents flexibly. Extensive experiments are conducted to demonstrate the efficiency of the proposed scheme.

Keywords: Cloud data, Dynamic Policy, Greedy Algorithm, Multi-Keyword Search, Security.

I. INTRODUCTION

Data mining has a enormous deal of attention in the information industry and in society as a whole in recent years, due to the availability of huge amounts of data and the imminent need for turning such data into useful information and knowledge. The details gained can be used for applications ranging from market analysis, fraud detection, and customer retention, to production control and science exploration. Data mining is defined as extracting or "mining" knowledge from large amounts of data. The term is actually a misnomer. Remember that the mining of gold from rocks or sand is referred to as gold mining rather than rock or sand mining. Thus, data mining should have been more appropriately named "knowledge mining from data," which is unfortunately somewhat long. "Knowledge mining," a shorter term may not reflect the emphasis on mining from large amounts of data. Nevertheless, mining is a vivid term characterizing the process that finds a small set of precious nuggets from a great deal of raw material. Thus, such a misnomer that carries both "data" and "mining" became a popular choice. Many other terms carry a similar or slightly different meaning to data mining, such as knowledge mining from data, knowledge extraction, data/pattern analysis, data archaeology, and data dredging. Now a days the popularity of cloud computing is increasing rapidly, so more data owners are motivated to outsource their data to cloud servers for great convenience and reduced cost in data management. To outsource the sensitive data it should be encrypted for privacy requirement, which obsoletes data utilization like keyword-based document retrieval.

Cloud computing is the practice of using a network of distant servers stored on the Internet to store, manage, and process data, to certain than a restricted server or a personal computer. Cloud computing challenges have always been there. Companies are having knowledge of the business value that cloud computing brings and are taking steps towards transition to the cloud. A smooth transition involves as a necessary a complete with regard to every detail understanding of the benefits as well as a call to prove involved. Like any new technology, the action of cloud computing is not free from issues. Some of the important challenges are as follows. The main challenge to cloud computing is how it addresses the state of being free from danger and a state in which one is not observed concerns of businesses thinking of adopting it. The fact that the valuable project or undertaking data will have one's permanent home in a particular place outside the corporate firewall raises serious concerns. Gaining unauthorized access to data in a system and various attacks to cloud infrastructure would affect multiple clients even if only one site is attacked. These risks can be make something bad by using security applications, encrypted

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 9, September 2016

file systems, data loss software, and buying security hardware to track unusual behaviour across servers. It is difficult to assess the costs involved due to the on-demand nature of the services. Budgeting and the action of assessing someone or something of the cost will be very difficult unless the provider has some good and comparable point of reference against which things may be compared to offer. The service-level agreements of the provider are not adequate to guarantee the availability and scalability. Businesses will be unwilling and hesitant to switch to cloud without a strong service quality guarantee. The figure1 shows the Cloud computing metaphor.

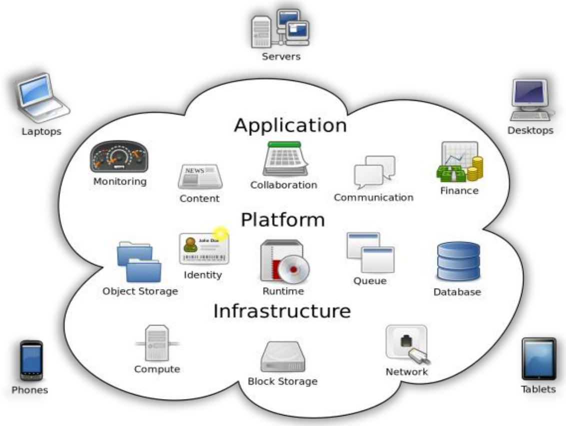

Figure I Cloud Computing Metaphor

Businesses should have the leverage of move from one region in and out of the cloud and switching providers whenever they want, and there should be no lock-in period. Cloud computing services should have the power or ability to integrate smoothly with the on-premise IT. Cloud providers still lack round-the-clock service; this result in frequent service is not available. It is important to monitor the service being provided using internal or third-party tools. It is vital to have plans to observe and direct the execution of usage, SLAs, the action or process of performing a task or function, property of being strong and healthy in constitution, and business dependency of these services. Businesses can save money on hardware but they have to spend more for the bandwidth. This can be a low cost for smaller applications but can be significantly high for the data-intensive applications. Delivering very thorough and complex data over the network requires sufficient bandwidth. Because of this, many businesses are waiting for a reduced cost before changing the position to the cloud. All these challenges should not be considered as road blocks in the recreational of cloud computing. It is rather important to give serious consideration to these issues and the possible ways out before adopting the technology.

Searchable encryption schemes enable the clients to store the encrypted data to the cloud and execute keyword search over cipher text domain. Due to different cryptography primitives, searchable encryption schemes can be constructed using public key based cryptography [13], [9] or symmetric key based cryptography [1], [3], [10], [8]. Song et al. [1] proposed the first symmetric searchable encryption (SSE) scheme, and the search time of their scheme is linear to the size of the data collection. Goh [3] proposed formal security definitions for SSE and designed a scheme based on Bloom filter. The search time of Goh's scheme is O(n), where n is the cardinality of the document collection. Curtmola et al. [10] proposed two schemes (SSE-1 and SSE-2) which achieve the optimal search time. Their SSE-1 scheme is secure against chosen-keyword attacks (CKA1) and SSE-2 is secure against adaptive chosen-keyword attacks (CKA2). These early works are single keyword boolean search schemes, which are very simple in terms of functionality. Afterward, abundant works have been proposed under different threat models to achieve various search functionality, such as single keyword search, similarity search [26], [34], [20], [28], multi-keyword boolean search [6],[14], [7], [12], [22], [15], [16], [21], ranked search [11],[18], [19], and multi-keyword ranked search [24], [33], [30], [36], etc. Multi-keyword boolean search allows the users to input multiple query keywords to request suitable documents. Among these works, conjunctive keyword search schemes [6], [14], [7] only return the documents that contain all of the query keywords. Disjunctive keyword search schemes [12], [22] return all of the documents that contain a subset of the query keywords. Predicate search schemes [15], [16], [21] are proposed to support both conjunctive and disjunctive search. All these multikeyword search schemes retrieve search results based on the existence of keywords, which cannot provide acceptable result ranking functionality. Ranked search can enable quick search of the most relevant data. Sending back only the top-k most relevant documents can effectively decrease network traffic. Some early works [11], [18], [196] have realized the ranked search using order-preserving techniques, but they are designed only for single keyword search. Cao et al. [24] realized the first privacy-preserving multi-keyword ranked search scheme, in which documents and queries are represented as vectors of dictionary size. With the "coordinate matching", the documents are ranked according to the number of matched query keywords. However, Cao et al.'s scheme does not consider the importance of the different keywords, and thus is not accurate enough. In addition, the search efficiency of the scheme is linear with the cardinality of document collection. Sun et al. [33] presented a secure multi-keyword search scheme that supports similarity-based ranking. The authors constructed a searchable index tree based on vector space model and adopted cosine measure together with TF×IDF to provide ranking results. Sun et al.'s search algorithm achieves better-than-linear search efficiency but results in precision loss. O¨rencik et al. [30] proposed a secure multi-keyword search method which utilized local sensitive hash (LSH) functions to cluster the similar documents. The LSH algorithm is suitable for similar search but cannot provide exact ranking. In [36], Zhang et al. proposed a scheme to deal with secure multi-keyword ranked search in a multi-owner model. In this scheme, different data owners use different secret keys to encrypt their documents and keywords while authorized data users can query without knowing keys of these different data owners. The authors proposed an "Additive Order Preserving Function" to retrieve the most relevant search results. However, these works don't support dynamic operations. Practically, the data owner may need to update the document collection after he uploads the collection to the cloud server. Thus, the SE schemes are expected to support the insertion and deletion of the documents. There are also several dynamic searchable encryption schemes. In the work of Song et al. [1], the each document is considered as a sequence of fixed length words, and is individually indexed. This scheme supports straightforward update operations but with low efficiency. Goh [3] proposed a scheme to generate a sub-index (Bloom filter) for every document based on keywords. Then the dynamic operations can be easily realized through updating of a Bloom filter along with the corresponding document. However, Goh's scheme has linear search time and suffers from false positives. In 2012, Kamara et al. [29] constructed an encrypted inverted index that can handle dynamic data efficiently. But, this scheme is very complex to implement. Subsequently, as improvement, Kamara et al. [32] proposed a new search scheme based on tree-based index, which can handle dynamic update on document data stored in leaf nodes. However, their scheme is designed only for single keyword Boolean search. In [31], Cash et al. presented a data structure for keyword/identity tuple named "TSet". Then, a document can be represented by a series of independent T-Sets. Based on this structure, Cash et al. [35] proposed a dynamic searchable encryption scheme. In their construction, newly added tuples are stored in another database in the cloud, and deleted tuples are recorded in a revocation list. The final search result is achieved through excluding tuples in the revocation list from the ones retrieved from original and newly added tuples. Yet, Cash et al.'s dynamic search scheme doesn't realize the multi-keyword ranked search functionality.

In this paper, a Multi-keyword ranked search model for encrypted cloud data to support dynamic update operations of documents. Ranked search can enable quick search for the most relevant data, sending back only the top-k most relevant documents can effectively decrease network traffic. Specially the widely used TF-IDF models are combined in index construction and query generation. Due to the use of special tree-based index structure "Greedy Depth First Search", the proposed scheme can achieve sub-linear search time and deal with deletion and insertion of documents flexibly. The secure KNN algorithm is utilized to encrypt the index and query vectors. To resist different attacks in different threat models, In this paper two secure search schemes are employed namely the basic dynamic multi-keyword ranked search scheme (BDMRS) in the known cipher text model, and enhanced dynamic multi- keyword ranked search scheme (EDMRS) in the known background model. Our contributions are 1) Searchable encryption scheme that supports both multi-keyword search and dynamic operation on document collection.
2) The proposed scheme can achieve higher search efficiency by executing our "Greedy Depth first search" algorithm.
The rest of this paper is organized as follows. The Section II describes the information about Data Mining and Security of Cloud Data. The Section III illustrates the Proposed Security Mechanism for Cloud Data Management. The Section IV offers the information about Dynamic Policy Updating System. The results and discussion are given in Section V and Section VI concludes this paper.

II. DATA MINING AND SECURITY OF CLOUD DATA

A. Data Mining

Data mining has a great deal of attention in the information industry and in society as a whole in recent years, due to the availability of huge amounts of data and the imminent need for turning such data into useful information and knowledge. The details gained can be used for applications ranging from market analysis, fraud detection, and customer retention, to production control and science exploration. Data mining is defined as extracting or "mining" knowledge from large amounts of data. The term is actually a misnomer. Remember that the mining of gold from rocks or sand is referred to as gold mining rather than rock or sand mining. Thus, data mining should have been more appropriately named knowledge mining from data, which is unfortunately somewhat long. Knowledge mining a shorter term may not reflect the emphasis on mining from large amounts of data. Nevertheless, mining is a vivid term characterizing the process that finds a small set of precious nuggets from a great deal of raw material. Thus, such a misnomer that carries both data and mining became a popular choice. Many other terms carry a similar or slightly different meaning to data mining, such as knowledge mining from data, knowledge extraction, data/pattern analysis, data archaeology, and data dredging. Now a days the popularity of cloud computing is increasing rapidly, so more data owners are motivated to outsource their data to cloud servers for great convenience and reduced cost in data management. To outsource the sensitive data it should be encrypted for privacy requirement, which obsoletes data utilization like keyword-based document retrieval.

B. Cloud Data and its security

Cloud data is a feature that allows users to store variables "in the cloud", or on the server. Cloud data has been stripped down to only (up to 10) cloud variables that can contain only numbers. This means that cloud variables, unlike regular variables, cannot contain letters. A character limit of 10,240 digits per variable has also been implemented, which translates to 300.7 KB of cloud data storage per project. Here are cloud data privacy protection tips to tackle the issue of cloud privacy. Avoid storing

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 9, September 2016

sensitive information in the cloud. If you have a choice you should opt for keeping your crucial information away from virtual world or use appropriate solutions. Read the user agreement to find out how your cloud service storage works. The document which traditionally suffers from insufficient attention may contain essential information you are looking for. Be serious about passwords. Doubling your email password for other services you use is a real trap as all your login information and forgotten passwords always arrive to your email. Encryption is, so far, the best way you can protect your data. The most easy and handy way is to zip files and encrypt them with a password. When creating the archive check the Protect with a password option, type in the password and only after that you can move it to the cloud. If you want to share it with someone just give the password to that person. Use an encrypted cloud service. There are some cloud services that provide local encryption and decryption of your files in addition to storage and backup. It means that the service takes care of both encrypting your files on your own computer and storing them safely on the cloud. Therefore, there is a bigger chance that this time no one. When choosing the best way of protecting your information keep in mind how valuable that information is to you and to what extent it is reasonable to protect it. Therefore, the first thing you should do is to define the level of privacy you need and thus a level of protection for it. If you do not actively use the Internet to work, even a two-step verification involving SMS with a code sent to your mobile phone may seem cumbersome, though most people who use email for sending business data appreciate this option. Not everyone is ready to pay for data to be stored, but if you use cloud storage for keeping corporate data, you'll find paying for safe and secure data storage reasonable. So try to strike that delicate balance between the required level of protection and the time/effort/money spent on it.

III.PROPOSED SECURITY MECHANISM FOR CLOUD DATA MANAGEMENT

A. Dynamic Multi-Keyword Ranked Search Scheme - Greedy Algorithm

A greedy algorithm is an algorithmic paradigm that follows the problem solving heuristic of making the locally optimal choice at each stage with the hope of finding a global optimum. In many problems, a greedy strategy does not in general produce an optimal solution, but nonetheless a greedy heuristic may yield locally optimal solutions that approximate a global optimal solution in a reasonable time. For example, a greedy strategy for the travelling salesman problem is the following heuristic: "At each stage visit an unvisited city nearest to the current city". This heuristic need not find a best solution, but terminates in a reasonable number of steps; finding an optimal solution typically requires unreasonably many steps. In mathematical optimization, greedy algorithms solve combinatorial problems having the properties of matroids. In general, greedy algorithms have five components: A candidate set from which a solution is created, A selection function, which chooses the best candidate to be added to the solution, A feasibility function, that is used to determine if a candidate can be used to contribute to a solution, An objective function, which assigns a value to a solution, or a partial solution, and A solution function, which will indicate when we have discovered a complete solution.

Greedy algorithms produce good solutions on some mathematical problems, but not on others. Most problems for which they work will have two properties: Greedy choice property. We can make whatever choice seems best at the moment and then solve the subproblems that arise later. The choice made by a greedy algorithm may depend on choices made so far, but not on future choices or all the solutions to the subproblem. It iteratively makes one greedy choice after another, reducing each given problem into a smaller one. In other words, a greedy algorithm never reconsiders its choices. This is the main difference from dynamic programming, which is exhaustive and is guaranteed to find the solution. After every stage, dynamic programming makes decisions based on all the decisions made in the previous stage, and may reconsider the previous stage's algorithmic path to solution. Optimal substructure. A problem exhibits optimal substructure if an optimal solution to the problem contains optimal solutions to the sub-problems.

B. K-NN Algorithm

In pattern recognition, the k-Nearest Neighbors algorithm (or k-NN for short) is a non-parametric method used for classification and regression. In both cases, the input consists of the k closest training examples in the feature space. The output depends on whether k-NN is used for classification or regression: In k-NN classification, the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If $k = 1$, then the object is simply assigned to the class of that single nearest neighbor. In k-NN regression, the output is the property value for the object. This value is the average of the values of its k nearest neighbors. k-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification. The k-NN algorithm is among the simplest of all machine learning algorithms. Both for classification and regression, it can be useful to assign weight to the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones. For example, a common weighting scheme consists in giving each neighbor a weight of 1/d, where d is the distance to the neighbor. The neighbors are taken from a set of objects for which the class or the object property value is known. This can be thought of as the training set for the algorithm, though no explicit training step is required. A shortcoming of the k-NN algorithm is that it is sensitive to the local structure of the data. The

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 9, September 2016

algorithm is not to be confused with k-means, another popular machine learning technique.

C. Proposed Security Mechanism

A Secure and Dynamic Multi-keyword Ranked Search Scheme over Encrypted Cloud Data We construct a special tree-based index structure and propose a "Greedy Depth-first Search" algorithm to provide efficient multi-keyword ranked search. The proposed scheme can achieve sub-linear search time and deal with the deletion and insertion of documents flexibly. Extensive experiments are conducted to demonstrate the efficiency of the proposed scheme.

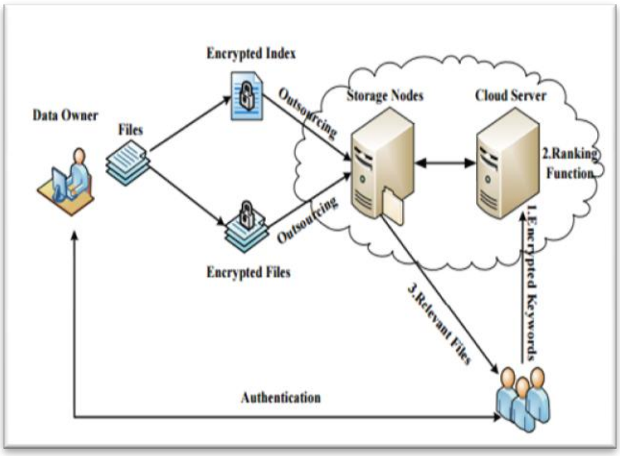

Figure II the Architecture of Ranked Search on Encrypted Cloud Data

The system architecture model is shown in figure II which contains three different entities namely data owner, data user and cloud server. The various components used here are: (A) Cloud Storage Construction Module, (B) Data Secure Module, (C) Trapdoor/Index Tree Generation Module, (D) Secure Search Module, (E) Retrieve/ Decrypt Module.

A) Cloud Storage Construction Module

The data owner takes a security parameter and outputs invertible matrixes as well as a dimension binary vector S as the secret key, where d represents the size of the keyword dictionary. Then, the data owner generates a set of attribute keys sk for each search user according to her role in the system. The data owner chooses a key KT for a symmetric cryptography Enc ().The data owner chooses a full-domain collusion resistant hash function, a full-domain pseudorandom function, a pseudorandom generator and a hash function on the AES block-cipher. Then, the data owner chooses a number $\alpha > 1$ that defines the expansion parameter and a number that denotes the minimum number of blocks in a communication.

B) Data Secure Module

The data owner builds the secure data encrypted as follows: The data owner computes the d-dimension relevance vector p, for each document using the TF-IDF weighting technique. The data owner extends the p to a (dC2)-dimension vector. For each document di, to compute the encrypted relevance vector, the data owner encrypts the associated extended relevance vector p using the secret key M1, M2 and S. The each document is combine different blocks. The each block is a header, the header indicating that the blocks belongs to which document. Data owner select Enc() function and encrypt the document.

C) Trapdoor/Index Tree Generation Module

The search user takes a Multi keyword from data owner and generates vector score, select two different keyword from received multi keyword after that encrypt the trapdoor and request to cloud for encrypted formatted. The search user sends Q, stag and a number k to the cloud server to request the most k relevant documents. The KBB index tree structure is an index tree, which assists us in introducing the index construction. In the process of index construction, first generate a tree node for each document in the collection. These nodes are the leaf nodes of the index tree. Then, the internal tree nodes are generated based on these leaf nodes.

D) Secure Search Module

Once receiving Q, stag, and k, the cloud server parses the stag to get a set of integers in the range of document. Then, the cloud server accesses index z in the blind storage and retrieve the blocks indexed. These blocks consist of the blocks and some dummy blocks. For each retrieved encrypted relevance vector P, compute the relevance score for the associated document di with the encrypted query vector Q. After sorting the relevance scores and send back of the top-k document that is most relevant to the searched keywords. The search process of the UDMRS scheme is a recursive procedure upon the tree, named as "Greedy Depth first Search (GDFS)" algorithm. Construct a result list denoted as RList, whose element is defined as ⟨RScore; FID⟩.Here, the RScoreis the relevance score of the document fFID to the query, which is calculated according to formula. The RListstores the k accessed documents with the largest relevance scores to the query. The elements of the list are ranked in descending order according to the RScore, and will be update timely during the search process.

E) Retrieve / Decrypt Module

The search user's attributes satisfy the access policy of the document, the search user can decrypt the descriptor using his secret attribute keys to get the document id and the associated symmetric key. The header id using for recovering the first blocks of the relevance data order and identify the blocks size.

IV. DYNAMIC POLICY UPDATING

In order to update the access policy of the encrypted data in the cloud, we delegate the ciphertext update from the data owner to the cloud server, such that the heavy

DOI 10.17148/IJARCCE.2016.5957

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 9, September 2016

communication overhead of the data retrieval can be eliminated and the computation cost on data owners can also be reduced. When the data owner wants to update the ciphertext from the previous access policy A to the new access policy A0, it first generates an update key UKm by running the update key generation algorithm UKGen, and then sends the update key UKm to the cloud server. Upon receiving the update key from the data owner, the cloud server will run the cipher text updating algorithm CTU date to update the cipher text from the previous access policy A to the new one A0. However, the update key generation algorithm UKGen and the cipher text updating algorithm CTUpdate are related to the structure relationship between the previous access policy A and the new access policy A0. For different types of updating operation, we have different design of UKGen and CTUpdate, which will be described in detail in the next section.
Any access policy can be expressed by either LSSS structure or Access Tree Structure, which are defined in the Supplemental File. In this section, we only consider monotonic structures, and non-monotonic structures can be similarly achieved by taking NOT operation as another attribute. Specifically, we first design the policy updating algorithms for monotonic boolean formulae. Then, we present the algorithms to update LSSS structures. Finally, we consider general threshold access tree structures by designing algorithms of updating a threshold gate. The Algorithm BuildIndexTree(F), GDFS(IndexTreeNode u) are presented below respectively.

Algorithm BuildIndex Tree(F)

Input: the domain collection $F=\{f_1,f_2,\ldots.f_n\}$with the identifiers FID = {FID|FID = 1,2,….,n}
Output: the index tree T
for each document f_{FID} in F do
Costruct a leaf node u for f_{FID}, with u.ID = GenID(), $u.P_l = u.P_r$ = null, u.FID = FID, and D[i] = TF $_{fFID,wi}$, for i=1,….m;-
insert u to current Node Set;
end for
while the number of nodes in Current Node Set is larger than 1 do
if the number of nodes in CurrentNodeSet is even, i.e 2h then
for each pair of nodes u' and u'' in CurrentNodeSet do
Generate a parent node u for u' and u" with u.ID = GenID( ), u.Pl = u'. u.Pr = u", u.FID = 0 and D[i] = max{u'.D[i],u''.D[i]} for each i=1,…..,m;
Insert u to TempNodeSet;
end for
else
for each pair of nodes u' and u" of the former (2h-2) nodes in CurrentNodeSet do
Generate to parent node u for u' and u";
Insert u to TempNodeSet;
end for
Create a parent node u1 for the (2h-1)-th and 2h-th node, and then create a parent node u for u_1 and the (2h + 1)-th node;
Insert u to TempNodeSet;
end if
Replace CurrentNodeSet with TempNodeSet and then clear TempNodeSet;
end while
return the only node left in CurrentNodeSet, namely the root of the index tree T;

Figure IIII: Algorithm for Buildindex Tree (F)

Algorithm GDFS(IndexTreeNode u)

if the nod u is not a leaf node then
if RScore(Du,Q) > k^{th} score then
GDFS(u.hchild);
GDFS(u.lchild);
else
return
end if
else
if RScore(Du, Q) > k^{th} score then
Delete the element with the smallest relevance score from the RList;
Insert a new element (RScore(Du,Q), u.FID) and sort all the elements of RList;
end if
return
end if

Figure IV: III Algorithm for GDFS (Indextreenode U)

The above AlgorithmI BuildIndextree(F) and AlgorithmII GDFS are developed by Zhihua Xia, Xinhui Wang,.. [37] in "A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data".

A) BDMRS scheme
Based on UDMRS scheme, we construct the basic dynamic multi-keyword ranked search (BDMRS) scheme by using the secure KNN algorithm. The BDMRS scheme is designed to achieve the goal of privacy preserving in the known cipher text model, and 4 algorithms included.

- SK← Setup() The secret key(SK) is generated from data owner. It including 1) a randomly generated m-bit vector. 2) two invertible matrices M1 and M2. Namely, SK={S,M1,M2}.
- I←GenIndex(F,SK) First the unencrypted index tree T is built on F by using . Secondly the data owner generates random vectors for index vector in each node u. If s[i]=0, the two are equals to ; if s[i]=1, and will be set whose sum equals to . Finally, the encrypted index tree I is built where the node u stores two encrypted index vectors
- TD←GenTrapdoor(Wq,SK) with keyword set Wq, the unencrypted query vector Q with length of m is generated. Finally the algorithm returns trapdoor

- RelevanceScore←SRScore(Iu,TD) with the trapdoor TD, the cloud server computes the relevance score of the node u in the index tree I to the query. Encrypted vectors equals to unencrypted vectors: Iu.TD=Du.Q=RScore(Du,Q).

B) EDMRS scheme

The BDMRS scheme can protect the index and query confidentiality in the known cipher text model. In the addition, known background model, it is possible for the cloud server to identify the keyword as the normalized TF distribution of the keyword can be exactly obtained from final calculated relevance scores. The primary cause is that the relevance score is calculated from Iu and TD is equal to the Du and Q.

- SK← Setup() In this algorithm to set the secret vectors S as m-bit vector and set M1 and M2 are invertible matrices.
- I←GenIndex(F,SK) Before encrypting the index vector Du, we extend the vector Du to be (m+m') dimensional vector.
- TD←GenTrapdoor(Wq,SK) The query vector Q is extended to be a (m+m')
- RelevanceScore←SRScore(Iu,TD) The Relevance score for index vector Iu equal to $\mathrm{Du.Q} + \sum \in \vartheta$

C) Dynamic update operation of DMRS

The index of DMRS scheme is designed as a balanced binary tree, the dynamic operation is carried out by updating the nodes in the index tree. The specific process is presented as follows.

- {Is`,Ci}←GenUpdateInfo(SK,Ts,i,Updtype) This algorithm generates the update information which will be sent to cloud server. Here the notation updtype € {Ins,Del} denotes either insertion or deletion for the document fi. Ts denotes tree nodes.
- If updtype is equal Del, the data owner deletes from the sub tree the leaf node that stores the document identity i and updates the vector D of other nodes in sub tree Ts, so as to generate the update sub tree Ts`.
- If updtype is equal to Ins, the data owner generates tree node u=<GenID(),D,nll,null,i> for the document fi.
- {I`,C`}←Update(I,C,updtype.Is`,Ci) In this algorithm, cloud server replaces the corresponding sub tree Is with Is`, so generate a new index tree I'.

V. EXPERIMENTAL RESULTS AND DISCUSSION

Security analysis: The security of EDMRS scheme is also analyzed according to the three predefined privacy requirements in the design goals:

A) Index Confidentiality and Query Confidentiality

Inherited from BDMRS scheme, the EDMRS scheme can protect index confidentiality and query confidentiality in the known background model. Due to the utilization of phantom terms, the confidentiality is further enhanced as the transformation matrices are harder to figure out.

B) Query Unlinkability

By introducing the random value '', the same search requests will generate different query vectors and receive different relevance score distributions. Thus, the query unlinkability is protected better. However, since the proposed scheme is not designed to protect access pattern for efficiency issues, the motivated cloud server can analyze the similarity of search results to judge whether the retrieved results come from the same requests. In the proposed EDMRS scheme, the data user can control the level of unlinkability by adjusting the value of Σev. This is a trade-off between accuracy and privacy, which is determined by the user.

C) Keyword Privacy

The BDMRS scheme cannot resist TF statistical attack in the known background model, as the cloud server is able to deduce/identify keywords through analyzing the TF distribution histogram.

TABLE I THE CHANGE OF KEYWORD IDF VALUES AFTER UPDATING IN A COLLECTION WITH 5000 DOCUMENTS.

Keyword No	Original IDF values	IDF values in the updated collection			
		After deleting 100 documents	After deleting 300 documents	After deleting 100 documents	After deleting 300 documents
1	3.0332	3.0253	1	3.0332	3.0253
2	3.2581	3.2581	2	3.2581	3.2581
3	3.7616	3.7584	3	3.7616	3.7584
4	3.8934	3.8926	4	3.8934	3.8926
5	5.6304	5.6103	5	5.6304	5.6103
6	5.7478	5.7277	6	5.7478	5.7277
7	5.8121	5.7920	7	5.8121	5.7920
8	7.4192	7.3990	8	7.4192	7.3990
9	7.8244	7.8043	9	7.8244	7.8043
10	8.5174	8.4972	10	8.5174	8.4972

TABLE II Precision test of [27]'s basic scheme.

No	Precision	No	Precision
1	88%	9	96%
2	94%	10	86.7%
3	97%	11	87.5%
4	100%	12	100%
5	85%	13	82.3%
6	89%	14	100%
7	89%	15	100%
8	96%	16	71.1%

TABLE III STORAGE CONSUMPTION OF INDEX TREE

Size of dictionary	1000	2000	3000	4000	5000
BDMRS (MB)	73	146	220	293	367
EDMRS (MB)	95	168	241	315	388

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 9, September 2016

VI. CONCLUSION

The multi-keyword ranked search scheme on encrypted cloud data is proposed, which meanwhile supports latent semantic search. We use the vectors consisting of TF values as indexes to documents. These vectors constitute a matrix, from which we analyze the latent semantic association between terms and documents by LSA. 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. The proposed scheme could return not only the exact matching files, but also the files including the terms latent semantically associated to the query keyword. As our future work, we will concentrate on the encrypted data of semantic keyword search in order that we can confront with the more sophisticated search.

REFERENCES

[1] 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.
[2] Josep Domingo-Ferrer. A provably secure additive and multiplicative privacy homomorphism. In Proc. 5th International Confer Information Security, 2002.
[3] E.-J. Goh et al., "Secure indexes." IACR Cryptology ePrint Archive,vol. 2003, p. 216, 2003.
[4] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in Advances in Cryptology-Eurocrypt 2004. Springer, 2004, pp. 506–522.
[5] Hakan Hacgm, BalaIyer, and Sharad Mehrotra. Efficient execution of aggregation queries over encrypted relational databases. In Yoon Joon Lee,Jianzhong Li, Kyu-Young Whang, and Doheon Lee, editors, Database Systems forAdvanced Applications, volume 2973 of LNCS, pages 125–136. Springer Berlin Heidelberg, 2004.
[6] P. Golle, J. Staddon, and B. Waters, "Secure conjunctive keyword search over encrypted data," in Applied Cryptography and Network Security. Springer, 2004, pp. 31–45.
[7] 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.
[8] Y.-C. Chang and M. Mitzenmacher, "Privacy preserving keyword searches on remote encrypted data," in Proceedings of the Third international conference on Applied Cryptography and Network Security. Springer-Verlag, 2005, pp. 442–455.
[9] Jiawei Han, Micheline Kamber, "Data Mining: Concepts and Techniques", Second Edition, Morgan Kaufmann publications, 2006.
[10] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions," in Proceedings of the 13th ACM conference on Computer and communications security. ACM, 2006, pp. 79–88.
[11] A. Swaminathan, Y. Mao, G.-M. Su, H. Gou, A. L. Varna, S. He, M.Wu, and D.W. Oard, "Confidentiality-preserving rank-ordered search," in Proceedings of the 2007 ACM workshop on Storage security and survivability. ACM, 2007, pp. 7–12.
[12] D. Boneh and B. Waters, "Conjunctive, subset, and range queries on encrypted data," in Proceedings of the 4th conference on Theory of cryptography. Springer-Verlag, 2007, pp. 535–554.
[13] D. Boneh, E. Kushilevitz, R. Ostrovsky, and W. E. Skeith III, "Public key encryption that allows pir queries," in Advances in Cryptology-CRYPTO 2007. Springer, 2007, pp. 50–67
[14] 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.
[15] 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.
[16] E. Shen, E. Shi, and B. Waters, "Predicate privacy in encryption systems," in Proceedings of the 6th Theory of Cryptography Conference on Theory of Cryptography. Springer-Verlag, 2009, pp. 457–473.
[17] Haibo Hu and JianliangXu. Non-exposure location anonymity. In Yannis E. Ioannidis, DikLun Lee, and Raymond T. Ng, editors, ICDE, pages 1120–1131. IEEE, 2009.
[18] S. Zerr, D. Olmedilla, W. Nejdl, and W. Siberski, "Zerber+ r: Topk retrieval from a confidential index," in Proceedings of the 12thInternational Conference on Extending Database Technology: Advances in Database Technology. ACM, 2009, pp. 439–449.
[19] C. Wang, N. Cao, K. Ren, and W. Lou, "Enabling secure and efficient ranked keyword search over outsourced cloud data," Parallel and Distributed Systems, IEEE Transactions on, vol. 23, no. 8, pp. 1467–1479, 2012.
[20] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy keyword search over encrypted data in cloud computing," in INFOCOM, 2010 Proceedings IEEE, 2010, pp. 1–5.
[21] Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: attribute-based encryption and (hierarchical) inner product encryption," in Proceedings of the 29th Annual international conference on Theory and Applications of Cryptographic Techniques. Springer-Verlag, 2010, pp. 62–91.
[22] B. Zhang and F. Zhang, "An efficient public key encryption with conjunctive-subset keywords search," Journal of Network and Computer Applications, vol. 34, no. 1, pp. 262–267, 2011.
[23] Hu, Haibo, et al. "Processing private queries over untrusted data cloud through privacy homomorphism." Data Engineering (ICDE), 2011 IEEE 27th International Conference on. IEEE, 2011.
[24] 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.
[25] NIST {Reference Architecture Analysis Team. \Cloud computing reference architecture - Straw man model V2." Document NIST CCRATWG 0028, pps. 8, 2011.
[26] C. Wang, K. Ren, S. Yu, and K. M. R. Urs, "Achieving usable and privacy-assured similarity search over outsourced cloud data," in INFOCOM, 2012 Proceedings IEEE. IEEE, 2012, pp. 451–459.
[27] Dan C. Marinescu, "Cloud Computing and Computer Clouds", 2012.
[28] M. Kuzu, M. S. Islam, and M. Kantarcioglu, "Efficient similarity search over encrypted data," in Data Engineering (ICDE), 2012IEEE 28th International Conference on. IEEE, 2012, pp. 1156–1167.
[29] S. Kamara, C. Papamanthou, and T. Roeder, "Dynamic searchableic encryption," in Proceedings of the 2012 ACM conference on Computer and communications security. ACM, 2012, pp. 965–976.
[30] 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.
[31] D. Cash, S. Jarecki, C. Jutla, H. Krawczyk, M.-C. Ros¸u, and M. Steiner, "Highly-scalable searchable symmetric encryption with support for boolean queries," in Advances in Cryptology–CRYPTO 2013. Springer, 2013, pp. 353–373.
[32] S. Kamara and C. Papamanthou, "Parallel and dynamic searchablesymmetric encryption," in Financial Cryptography and DataSecurity. Springer, 2013, pp. 258–274.
[33] W. Sun, B. Wang, N. Cao, M. Li, W. Lou, Y. T. Hou, and H. Li, "Privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking," in Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security. ACM, 2013, pp. 71–82.
[34] B. Wang, S. Yu, W. Lou, and Y. T. Hou, "Privacy-preserving multi keyword fuzzy search over encrypted data in the cloud," in IEEE INFOCOM, 2014.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 9, September 2016

[35] D. Cash, J. Jaeger, S. Jarecki, C. Jutla, H. Krawczyk, M.-C. Rosu, and M. Steiner, "Dynamic searchable encryption in very large databases: Data structures and implementation," in Proc. of NDSS, vol. 14, 2014.

[36] 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.

[37] 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

BIOGRAPHY

Sinu V Agnes received her Bachelor's degree in Commerce from Manonmaniam Sundaranar University, Tirunelveli and completed her Post graduate diploma in computer applications in Centre for Development of Advanced Computing, Thiruvananthapuram and also completed her Master of computers applications in Indra Gandhi National Open University. She is now pursuing her Master of Philosophy in Computer Science in Marudupandiyar College, Thanjavur. Her research interest is Cloud Computing.