ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 2, April 2013

A Scalable Multi Keyword Search and Relevance Oriented Ranking for Searchable Network Encrypted data in Cloud Storage Systems

Srinivasulu Chinna[1], A. Santosh Kumar[2], D. Priyanka[3], B. Sunil Kumar[4]

Pursing M.Tech(CSE), VignanaBharathi Institute of Technology, affiliated to JNTU-H, India[1]

Pursing M.Tech(CSE), VignanaBharathi Institute of Technology, affiliated to JNTU-H, India[2]

Asst. Professor, Dept of CSE, Vignana Bharathi Institute of Technology, Andhra Pradesh, India[3]

Asst. Professor, Dept of CSE, Jawaharlal Nehru Institute of Technology, Andhra Pradesh, India[4]

Abstract: Secured Cloud Storage service allows the users to store and access their data remotely from various locations by using various secured data access protocols. With the advent of cloud storage services data owners started to outsource their complex data management systems from local sites to the commercial public cloud for great flexibility and economic savings. Due to the high security and reliability of cloud storage services most of the data owners are persisting their personal and sensitive information like emails, financial details, photo albums and some other important documents in cloud. To prevent the data stored in the cloud from unauthorized access and manipulations, data is encrypted before persist in cloud. Now a days the data owners have the interest to share their private outsourced data with a large number of authorized users, who might want to only retrieve certain specific data files by using the most popular keyword search on cloud data.Keyword search on encrypted huge data would become more time consuming task and less results relevance. In this paper we are introducing an scalablemulti-keyword search against the searchable encrypted data to allow multiple keywords in the search query and return the documents in order of their relevance to these keywords.A scalable Relevance Oriented Ranking-ROR is used to display the retrieved results in an efficient order for a given keyword search query. Experiments on sample data set in cloud shows that our approach is retrieving and displaying the data from cloud effectively for a given query.

Keywords: Cloud Computing, Data Storage, Multi-Keyword Search, Relevance Oriented Ranking.

I. INTRODUCTION

Cloud computing is the use of computing resources that are delivered as a service over a network typically the Internet. Storage as a service (STaaS) is an architecture model in which a provider provides digital storage on their own infrastructure, where cloud customers can remotely store their data into the cloud so as to enjoy the on-demand high quality applications and services from a shared pool of configurable computing resources. STaaS can be implemented as a business model in which a large service provider rents space in their storage infrastructure on a subscription basis. There are many advantages to the users with this service like relief of the burden for storage management, universal data access with independent geographical locations, and avoidance of capital expenditure on hardware, software, and personnel maintenances, etc. The pioneer of Cloud Computing vendors, Amazon Simple Storage Service (S3) and Amazon Elastic Compute Cloud (EC2) are both well-known examples. While these internet-based online services do provide huge amounts of storage space and customizable computing resources, this computing platform shift, however, is eliminating the responsibility of local machines for data maintenance at the same time. As a result, users are at the mercy of their cloud service providers for the availability and integrity of their data. As Cloud Computing becomes prevalent, more and more sensitive information are being centralized intothe cloud, such as emails, personal health records, company finance data, and government documents, etc. Thefact that data owners and cloud server are no longerin the same trusted domain may put the outsourcedunencrypted data at risk, the cloud server may leakdata information to unauthorized entities or evenbe hacked.

To prevent the sensitive data to be stored in the cloud from un-authorized access, data files are encrypted before storing into cloud. Although encrypted data increases the data reliability, but it is a very challenging task given that there could be a large amount of outsourced data files.Moreover several data owners have the requirement to share their data among authorized users. These users can access the specific data stored in cloud by using the popular keyword search queries.These keyword search techniques allow the users to selectively retrieve files of interest and have been widely applied in plaintext search

ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 2, April 2013

scenarios. Traditional keyword search techniques downloading all the data and decrypting locally is clearly impractical, due to the huge amount of bandwidth cost in cloud scale systems. When STaaS using the traditional keyword search on large collaborative encrypted data, that returns all indexed files to users by decrypting them. In this approach, decrypting all of the retrieved huge files without finding their relevance with user interest will leads to time consuming and forwarding that huge files to user through network will consume high bandwidth, which is absolutely undesirable in today's pay-as-you-use cloud paradigm. Considering the potentially large number of on-demand data users and huge amount of outsourced data documents in the cloud, this problem is particularly challenging as it is extremely difficult to meet also the requirements of performance, system usability and scalability.

To solve the above stated problems, In this paper we introduced ascalable multi-keyword search against the searchable encrypted data to allow multiple keywords in the search query and return the documents in order of their relevance to these keywords. Among various multi-keyword semantics, we choose the efficient similarity measure of "coordinate matching", i.e., as many matches as possible, to capture the relevance of data documents to the search query.A scalable Relevance Oriented Ranking-ROR is used to display the retrieved results in an efficient order for a given keyword search query.

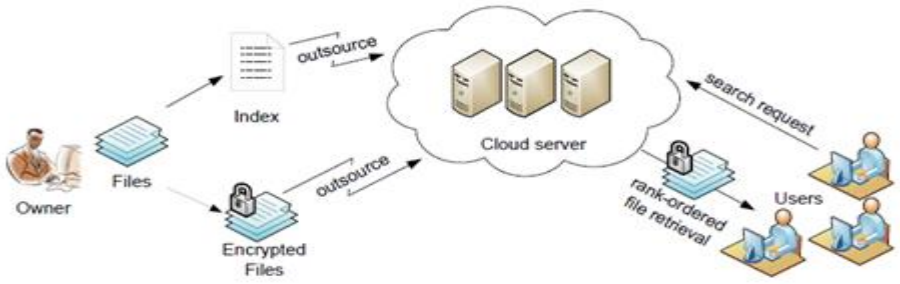

Fig.1. Architecture of the search over encrypted cloud data

II. RELATEDWORK

Recently, much of growing interest has been pursued in the context of remotely stored data search. Remote data storage in outsourced databases is of increasing interest [1]. Data will be stored in encrypted form. We are interested in a public key setting, where anyone can add to the database encrypted data which a distinguished "receiver" can retrieve and decrypt. The encryption scheme must permit search for data retrieval. Public-key encryption with keyword search (PEKS) [6, 1, 8] is a solution that provably provides strong privacy but search takes time linear in the size of the database. Given that databases can be terabytes in size, this is prohibitive. The practical community indicates that they want search on encrypted data to be as efficient as on unencrypted data, where a record containing a given field value can be retrieved in time logarithmic in the size of the database for example, via appropriate tree-based data structures. Deterministic encryption allows just this. The encrypted fields can be stored in the data structure, and one can find a target ciphertext in time logarithmic in the size of the database.

We consider an encrypted cloud data hosting service involving three different entities Data Owner, Data User, and Cloud Storage Server.

Data Owner an entity, which has large data files to be stored in the cloud and relies on the cloud for data maintenance and computation, can be either individual consumers or organizations. Data User is an entity, which has authorized by data owner with sufficient data access privileges and has the requirement to retrieve the data from cloud by using keyword search. Cloud Storage Server (CSS) is an entity, which is managed by Cloud Service Provider (CSP), has significant storage space and computation resource to maintain the clients' data.

III. RELEVANCE ORIENTED MULTIPLE KEYWORD SEARCH AND RANKING FOR SEARCHABLE ENCRYPTED CLOUD DATA

System Design: Data owner hasa collection of n data files C = (F1, F2,. . .Fn) that hewants to outsource on the cloud server in encryptedform while still keeping the capability to search throughthem for effective data utilization reasons. To do so,before outsourcing, data owner will first build a securesearchable index I from a set of m distinct keywordsIW = (w1,w2,. . . wm) extractedfrom the file collectionC, and store both the index I and the encrypted filecollection C on the cloud server.

ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 2, April 2013

To search the file collection for a given keyword query Q which has a set of keywords from k1. . .kn an authorized user generates and submits a search request to the cloud server.

Upon receiving Q from a data user, the cloud server is responsible to search the index I and return the corresponding set of encrypted documents. To improve the document retrieval accuracy, the search result should be ranked by the cloud server according to some ranking criteria. Moreover, to reduce the communication cost, the data user may send an optional number k along with the Q so that the cloud server only sends back top-k documents that are most relevant to the search query.

Efficient Multi-Keyword Search

To efficiently achieve multi-keyword search, we propose to employ "Keyword co-existence" [4] to quantitatively evaluate the efficient similarity measure "coordinate matching". This search first considers all the keywords of user given query by eliminating the supporting words using stemming techniques of porter's stemming algorithm[6]. For example, from the given query "DataMining by Charles " stemming algorithm will eliminate the word by from query and treats the datamining and charles as keywords. After stemming the query keywords are stored in a query keyword set QK as QK=(k1,k2 . . . kn). This Query keywordset is compared against the index keywords IW of encrypted file collection C on the cloud server. First it compares the query keywordsagainst file index keywords(IW) to calculate keyword occurrence ratio R . In this manner it compares the keyword set against all file collections indices and identifies the keyword occurrence ratio with every collection of files.

To assign a numeric score to a document for a query, the model measures the similarity between the query vector (since query is also just text and can be converted into a vector) and the document vector. The similarity between two vectors is once again not inherent in the model. Typically, the angle between two vectors is used as a measure of divergence between the vectors, and cosine of the angle is used as the numeric similarity. As an alternative, the inner-product (or dot-product) between two vectors is often used as a similarity measure. If all the vectors are forced to be unit length, then the cosine of the angle between two vectors is same as their dot-product.If D is the document vector and Q is the then the similarity of document Di to query Qi can be represented as

$\text{Sim}(D,Q) = \sum_{ti \in Q,D} \text{WtiQ}.\text{WtiD}$

Where WtiQ is the value of the i th component in the query vector Q, and WtiD is the i th component in the document vector D.

Term Weighting for Document Similarity Calculation

While identifying the similarity of between the query keywords and index contained keywords we have to consider the term weight to determine the relevant file from files collection C. we are using a method for weighting terms have been developed in the field. Weighting methods developed under the probabilistic models rely heavily upon better estimation of various probabilities. Methods developed under the vector space model are often based on researchers' experience with systems and large scale experimentation. In both models, three main factors come into play in the final term weight formulation. a)Term Frequency (or tf): Words that repeat multiple times in a document are considered salient. Term weights based on tf have been used in the vector space model b) Document Frequency: Words that appear in many documents are considered common and are not very indicative of document content. A weighting method based on this, called inverse document frequency (or idf) weighting. c) Document Length: When collections have documents of varying lengths, longer documents tend to score higher since they contain more words and word repetitions. This effect is usually compensated by normalizing for document lengths in the term weighting method. To calculate the weightage of a document based on score is :

$$\sum_{t \in Q,D} \frac{1 + ln\,(1 + ln(tf))}{(1-s) + s\frac{dl}{avdl}} \cdot qtf \cdot ln\frac{N+1}{df}$$

Where tf is the term's frequency in document, df is the document frequency in document, avdl is the average document length, dl is the document length (in bytes), and N is the total number of documents in the collection. By applying the above formula to each document we can calculate the similarity. Based on this similarity we can determine a threshold value (by default 0.80). The documents which can have the high similarity score than threshold value can be treated as relevant documents.

Relevance oriented ranking for documents

We propose a recursive Formula , which captures index hierarchical structure, to compute document TF*IDF similarity between an keyword query of the desired type to search for and a index keyword set. The input parameters keywords[m] is a keyword query containing m keywords. Based on the inverted lists built after pre-processing the document indices, we extract the corresponding documents d1,d2 . . .dn for each keyword in the query F is the frequency table. The ranking is estimated in three steps, First, it identifies the search intention of the user by using the term weighting, i.e. to identify the most desired keyword. In particular, it first collects all distinct documents, then for each document, we compute its

ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 2, April 2013

confidence (similarity) by using above formula and choose the one with the maximum confidence as the desired document. Second, for each document di , it computes the document TF*IDF similarity between d and the given keyword query Q. We maintain a rankedList to contain the similarity of each document. Third, it returns the ranked list of all documents by their similarity to the query. Function Similarity presents the procedure of computing document TF*IDF similarity between a document D and a given query Q of size n. This ranked list considers all the documents similarity value and sets the display priority based on their similarity against threshold value.

IV. EXPERIMENTS

We have performed comprehensive experiments to compare the effectiveness, efficiency and scalability of this keyword search on encrypted data with multi keyword search and Relevance Oriented Ranking approach. They are implemented in Java and run on a 3.6GHz Pentium 4 machine with 1GB RAM running Windows XP. We have tested both synthetic and real datasets. The synthetic dataset is generated using amazon cloud benchmark with size 115MB.the performance of our technique is evaluated regarding the efficiency of two proposed search techniques, as well as the tradeoff between search precision and privacy.

To build a searchable sub index Iist for each document Fi in the dataset F, the first step is to map the keyword set extracted from the document Fi to a data vector Di, followed by encrypting every data vector. The time cost of mapping or encrypting depends directly on the dimensionality of data vector which is determined by the size of the dictionary, i.e., the number of indexed keywords. And the time cost of building the whole index is also related to the number of subindex which is equal to the number of documents in the dataset. In our approach the number of keywords indexed in the dictionary determines the time cost of building a subindex. While the computation and communication cost in the query procedure is linear with the number of query keywords in other multiple-keyword search schemes [11], [12], our proposed schemes introduce nearly constant overhead while increasing the number of query keywords.

V. CONCLUSION

In this paper, for the first time we define and solve the problem of multi-keyword ranked search over encrypted cloud data, by using an effective document similarity calculations and relevance oriented ranking. Our construction is deliberately designed to meet these two important goals while efficiency being kept closely in mind. First one is finding the keyword priority to retrieve the relevant documents as results by using the term weighting, which is an efficient mechanism to find user intention and dramatically increases the speed of search by eliminating the supporting words in the given query. Second one is providing the ranking to retrieved documents in order to display the results based on query relevance. Our scheme is also very flexible, and it can easily be extended to support more advanced search queries. We conclude that this provides a powerful new building block for the keyword search in cloud storage environment.

REFERENCES

1. S. Kamara and K. Lauter, “Cryptographic cloud storage,” in RLCPS, January 2010, vol-3,p-7,LNCS. Springer, Heidelberg.
2. Singhal, “Modern information retrieval: A brief overview,” IEEE Data Engineering Bulletin, vol. 24, no. 4, pp. 35–43, 2001.
3. D. Song, D. Wagner, and A. Perrig, “Practical techniques for searches on encrypted data,” in Proc. of S&P, 2000.
4. Y.-C. Chang and M. Mitzenmacher, “Privacy preserving keyword searches on remote encrypted data,” vol 5,no-7,in Proc. of ACNS, 2005.
5. C. Cocks, An identity based encryption scheme based on quadratic residues, Eighth IMA International Conference on Cryptography and Coding, Dec. 2001,vol-12,pages 308-314, Royal Agricultural College,Cirencester, UK.
6. Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, “Enabling public verifiability and data dynamics for storage security in cloud computing,” in Proc. of ESORICS’09. Saint Malo, France: Springer-Verlag, 2009.
7. M. A. Shah, R. Swaminathan, and M. Baker, “Privacy-preserving audit and extraction of digital contents,” Cryptology.
8. C. Wang, Q. Wang, K. Ren, and W. Lou, “Ensuring data storage security in cloud computing,” in Proc. of IWQoS’09, Charleston, South Carolina,vol-15, USA, 2009.
9. R. Brinkman, “Searching in encrypted data,” in University of Twente,PhDthesis,Book, 2007.
10. S. Zerr, D. **OLMEDILLA**, W. Nejdl, and W. Siberski, “Zerber+r: Top-k retrieval from a confidential index,” in Proc. of EDBT, 2009, pp. 439–449.

BIOGRAPHY

Mr. B. Sunil Kumar Pursing Mtech(CSE) from VignanBharathi Institute of Technology affiliated to JNTU-H and completed MCA from LokamanyaTilak PG College Affiliated to Osmania University. I am Presently working as Asst Professor in Department of Computer Science & Engineering in Jawaharlal Nehru Institute of Technology, I am having 3years of Teaching Experience. My interested subjects are Web Technologies, Web services, Mobile computing, cloud computing, Computer Networks, Operating System, Computer Organisation, Java, C, and C++

ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 2, April 2013

Mrs.D. Priyanka B.Tech from Sri Sarathi Institute of Engineering and Technology College Affiliated to JNTU. I am Presently working as Assistant Professor in Department of Computer Science & Engineering in VignanaBharathi Institute of Technology, I am having 5+years of Teaching Experience. My interested subjects are Formal Languages and Automata Theory, Data Base Management Systems, Computer Networks, Mobile computing, Digital Logic Design and Computer Organization,

Mr. C. Srinivasulu pursing M.Tech(cse) from VignanaBharathi Institute of Technology affiliated to JNTU-H. Completed B.Tech(cse) from Visvodaya Institute of Technology & Science, Kavali (formerly affiliated to JNTU-H). I am currently working as a Team Lead with NTT DATA. I am having 1.7 years of experience in Teaching and 8 years of experience as a Java Developer. My interested subjects are Cloud computing, Java, C++, Web Technologies, Web Services, Computer Networks.

Mr. AkulaSantosh Kumar pursing M.Tech(cse) from VignanaBharathi Institute of Technology affiliated to JNTU-H. Completed Msc(computers) from A. V. College Of Arts, Science & Commerce, Affiliated to Osmania University. I am currently working as a Software developer. I am having 1.5experience as Software developer in Dot Net . My interested subjects are Cloud computing, Computer Networks, Network Security, dot Net,Computer organization and Operating Systems.