Improving Search Efficiency of Encrypted Cloud Data

Deepa P L, S Vinoth Kumar, Dr S Karthik

Abstract— CLOUD computing is a new technique to provide and consume the services based on the Internet, by providing for virtualized resources as a service. This is provided over the network. To protect the privacy of the sensitive data, the data stored in the cloud have to be encrypted before outsourced to the commercial public cloud. In such case, the effective data utilization service is a very challenging task. The searchable encryption techniques already developed allow users to securely search over encrypted data through keywords, but the problem is that they support only Boolean search and are not yet sufficient to meet the effective data utilization. In existing system they have used a ranked search technique. The crypto primitive OPSE is developed and an efficient one-to-many order preserving mapping function is derived. It allows the effective RSSE to be designed. To improve the search accuracy of the ranked search, in proposed work we implement the nearest neighbour search technique in which we find Euclidean distance between two vectors.

Index Terms— Cloud computing, Nearest neighbour search, One to many order preserving mapping, Ranked search.

1 INTRODUCTION

Cloud computing provides storage as well as service. It is the delivery of computing as a service, in which shared resources, software, and information are provided to computers and other devices as a metered service over the Internet.

The cloud computing technique provides computation, software, data access, and storage resources. The cloud users need not know the location and other details regarding the computing infrastructure. End users access cloud based applications through a web browser or a light weight desktop or mobile app while the business software and data are stored on servers at a remote location. Cloud application providers strive to give the same or better service and performance as if the software programs were installed locally on end-user computers.

The main objective is to solve the problem of supporting efficient ranked keyword search for achieving effective utilization of remotely stored encrypted data in Cloud. A basic scheme is given and shows that by following the same existing searchable encryption framework, it is very inefficient to achieve ranked keyword search. The security guarantee is weakened, resort to the newly developed crypto primitive OPSE, and an efficient one-to-many order preserving mapping function is derived. It allows the effective RSSE to be designed.

In Cloud Computing, the data owners may share their outsourced data with a large number of users, who might want to only retrieve certain specific data files they are interested in during a given session. The most popular ways to do so is through keyword-based search. Such type of keyword search technique allows users to selectively retrieve files of interest and has been widely applied in plaintext search scenarios. If data encryption is done, it restricts the user to perform keyword search and also demands the protection of keyword privacy, makes the traditional plaintext search methods fail for encrypted cloud data.

Therefore, to enable a searchable encryption system with support of secure ranked search is the problem solved in this paper. Ranked search greatly enhances system usability by returning the matching files in a ranked order regarding to certain relevance criteria (e.g., keyword frequency). The main task is to enable ranked searchable symmetric encryption for effective utilization of outsourced and encrypted cloud data. In the proposed system, the nearest neighbour search is performed in the results obtained from the ranked search.

2 RELATED WORKS

In the fuzzy keyword search, the problem of effective fuzzy keyword search over the encrypted data is solved. Fuzzy keyword search enhances the system usability by returning exact matching results. If the exact match fails, the closest match is returned as the result. Edit distance operation is used to quantify the keyword similarity [14].

The different modules in Fuzzy keyword search are

1. Wildcard-based technique
   A wildcard is used to edit the operations at the same position. The edit distance can be calculated using substitution, deletion and insertion [16].
2. Gram-based technique
   Here the fuzzy set is constructed based on grams. The gram of a string is a substring and can be used for

- *Deepa P L is currently pursuing masters degree program in Software engineering inSNS College of Technology, AnnaUniversity, India, PH-09495220779. E-mail: deepadnair1988@gmail.com*
- *S. Vinoth Kumar is currently working as Assistant Professor in Computer science and Engineering in SNS College of Technology, AnnaUniversity , India, E-mail: vinothmepco@gmail.com*
- *Dr S. Karthik is currently working as Dean in Computer science and Engineering, SNS College of Technology, Anna University, India*

effective approximate search. The order of the characters after the primitive operation is always kept the same before the operations.

3. Symbol-based trie-traversed scheme
In this technique, a multi-way tree is constructed for storing the fuzzy keyword set over a finite symbol set

All the trapdoors sharing a common prefix have common nodes. The fuzzy keyword in the trie can be found by depth first search approach [14].

The conjunctive keyword search mechanism can be used to improve the efficiency of search. This technique will retrieve most efficient and relevant data files. The conjunctive keyword search automatically generates ranked results so that the searching flexibility and efficiency will be improved [15].

This technique uses the wildcard based method and gram based method for constructing fuzzy keyword sets and symbol based trie- traverse scheme for generating a multi way tree to store the fuzzy keyword sets generated. This reduces the storage overhead. It also uses the Edit distance concept to quantify the keyword similarity.

Using the verifiable fuzzy keyword search, the user generates a symbol based index tree with encrypted documents and outsources it in the cloud server. When the search request is received by the server, the server maps the searching request to a set of documents. Each document is assigned an identifier and a set of keywords [6].

After searching, the server retrieves the search request and the proof for the result to the user. Using the proof, the user can verify the correctness and completeness of the result.

The searching is done based upon following rules.

(a) The input given to search exactly matches the preset keyword.
(b) If format inconsistencies in the searching input exist means, it will return the closest possible matches available.

Search privacy as well as the document privacy is ensured. The document privacy is ensured by the encryption algorithm [12].

The edit distance can be embedded to enable private keyword search. Here the private identification scheme is combined with embedding of edit distance into the hamming distance. This is done to obtain a fuzzy keyword search for the edit distance. This method does not need to a priori define the set of words which are considered as acceptable for the search. It increases security in this model [7].

Managing the nearest neighbour search in encrypted domain is the principle of a private identification scheme. This technique associates a message into a set of keywords and to consider each keyword as a virtual address. Receiver can recover link toward the associated messages. Information retrieval enables to retrieve a block from the database without knowledge about the query and answer.

A searchable re-encryption scheme is introduced in the index management scheme. Using this technique, user can share the data with others safely by generating searchable encryption index and re-encrypting it. The security requirements are set up and it uses two techniques-Proxy re-encryption function and searchable encryption function. These methods provide efficiency. The search method uses multiple keywords and thus the flexibility is provided [13].

3 PROBLEM STATEMENT

Ranked search can also elegantly eliminate unnecessary network traffic by sending back only the most relevant data, which is highly desirable in the “pay-as-you use” cloud paradigm. For privacy protection, such as ranking operation, however, it should not leak any keyword related information. To improve search result accuracy as well as to enhance user searching experience, it is crucial for such ranking system to support multiple keywords search.. As a common practice indicated by today’s web search engines, data users may tend to provide a set of keywords instead of only one as the indicator of their search interest to retrieve the most relevant data.

Our early work has been aware of this problem, and solves the secure ranked search over encrypted data with support of only single keyword query. But how to design an efficient encrypted data search mechanism that supports multi-keyword semantics without privacy breaches still remains a challenging open problem. In this work, we define and solve the problem of secure ranked keyword search over encrypted cloud data. Ranked search enhances system usability by enabling search result relevance ranking and thus ensures the file retrieval accuracy.

We explore the statistical measure approach. In that, the relevance score is calculated, from information retrieval to build a secure searchable index. And thus develop a one-to-many order-preserving mapping technique to properly protect those sensitive score information. The resulting design is able to facilitate efficient ranking without losing keyword privacy. The analysis shows that the proposed solution enjoys “as strong-as-possible” security guarantee compared to previous searchable encryption schemes, while correctly realizing the goal of ranked search.

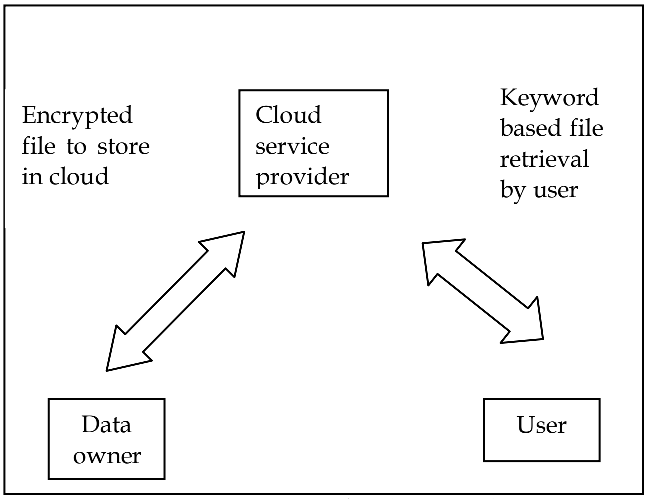
Encrypted file to store in cloud
Cloud service provider
Keyword based file retrieval by user
Data owner
User

Fig 1: Cloud Network

4 Implementation of K-Nearest Neighbor Search

The network model for the cloud data storage consists of three different network entities such as:

Client (Data Owner): An entity, and has large data files to be stored in the cloud and relies on the cloud for data maintenance and computation. This can be either individual consumers or organizations.
Cloud Storage Server (CSS): An entity, which is managed by Cloud Service Provider (CSP), has significant storage space and computation resource to maintain the clients' data; in the cloud paradigm, by putting the large data files on the remote servers, the clients can be relieved of the burden of storage and computation.
Public User: The one who access the cloud data which is the private data of cloud data owners. The public data is stored in the cloud by data owners for business purposes it can be accessed by any user for their needs.

A framework for ranked searchable encryption scheme is developed. An Order Preserving Symmetric Encryption technique is use. It is a deterministic encryption scheme where the numerical ordering of the plaintexts gets preserved by the encryption function. Construction that is provably secure under the security framework of pseudorandom function or pseudorandom permutation.

A one-to-many order-preserving mapping employs the random plaintext-to-bucket mapping of OPSE, but incorporates the unique file IDs together with the plaintext m as the random seed in the final ciphertext chosen process.
Because of the use of unique file ID as part of random selection seed, the same plaintext will no longer be deterministically assigned to the same ciphertext c, but instead a random value within the randomly assigned bucket in range R.

At last we implement the nearest neighbor search technique. This technique depends on the distances between the query vector and the database vectors, or equivalently the squared distances. For nearest neighbors search, we do not compute the square roots in practice: the square root function is monotonically increasing and the squared distances produce the same vector ranking.

An exhaustive comparison of the query vector with all codes is prohibitive for very large datasets. Therefore, a modified inverted file structure is introduced to rapidly access the most relevant vectors. A coarse quantizer is used to implement this inverted file structure. In that, the vectors corresponding to a cluster (index) are stored in the associated list. The vectors in the associated list are represented by short codes, and can be computed by our product quantizer, which is used here to encode the residual vector with respect to the cluster center.

The Nearest neighbor search depends on the distances between the query vector and the database vectors, or equivalently the squared distances. For nearest neighbors search, we do not compute the square roots in practice: the square root function is monotonically increasing and the squared distances produce the same vector ranking. We propose two methods to compute an approximate Euclidean distance between these vectors, a symmetric and an asymmetric one. To avoid exhaustive search we combine an inverted file system with the asymmetric distance computation (IVFADC). It slightly improves the search accuracy, as encoding the residual is more precise than encoding the vector itself.

The system design is summarized as follows:

- Initially, we define the problem of secure ranked keyword search over encrypted cloud data, and provide such an effective protocol, which fulfils the secure ranked search functionality with little relevance score information leakage against keyword privacy.
- The security analysis shows that our ranked searchable symmetric encryption scheme indeed enjoys "as-strong-as-possible" security guarantee compared to previous searchable symmetric encryption (SSE) schemes.
- The practical considerations and enhancements of our ranked search mechanism are investigated. This includes the efficient support of relevance score dynamics, the authentication of ranked search results, and the reversibility of our proposed one to- much order-preserving mapping technique.
- Extensive experimental results demonstrate the effectiveness and efficiency of the proposed solution.

Searching the nearest neighbor(s) of a query x consists of:

1) Quantize x to its w nearest neighbors in the codebook qc; For the sake of presentation, in the two next steps we simply denote by r(x) the residuals associated with these w assignments. The two steps are applied to all w assignments.
2) Compute the squared distance d(uj(r(x)); $c_{j;i}$) for each subquantizer j and each of its centroids $c_{j,i}$;

3) Compute the squared distance between r(x) and all the indexed vectors of the inverted list. Using the subvector-to-centroid distances computed in the previous step, this consists in summing up m looked-up values;
4) Select the K nearest neighbors of x based on the estimated distances. This is implemented efficiently by maintaining a Maxheap structure of fixed capacity that stores the K smallest values seen so far. After each distance calculation, the point identifier is added to the structure only if its distance is below the largest distance in the Maxheap.

The performance by search time it includes fetching the posting list in the index, decrypting, and rank ordering each entries. Our focus is on top-k retrieval. As the encrypted scores are order preserved, server can process the top-k retrieval almost as fast as in the plaintext domain.

Searching the nearest neighbor(s) of a query x consists of

1) Quantize x to its w nearest neighbors in the codebook qc; For the sake of presentation, in the two next steps we simply denote by r(x) the residuals associated with these w assignments. The two steps are applied to all w assignments.

2) Compute the squared distance d(uj(r(x)); $c_{j;i}$) for each subquantizer j and each of its centroids $c_{j,i}$;

3) 3) Compute the squared distance between r(x) and all the indexed vectors of the inverted list. Using the subvector-to-centroid distances computed in the previous step, this consists in summing up m looked-up values;

4) 4) Select the K nearest neighbors of x based on the estimated distances. This is implemented efficiently by maintaining a Maxheap structure of fixed capacity that stores the K smallest values seen so far. After each distance calculation, the point identifier is added to the structure only if its distance is below the largest distance in the Maxheap.

The performance by search time it includes fetching the posting list in the index, decrypting, and rank ordering each entries. Our focus is on top-k retrieval. As the encrypted scores are order preserved, server can process the top-k retrieval almost as fast as in the plaintext domain.

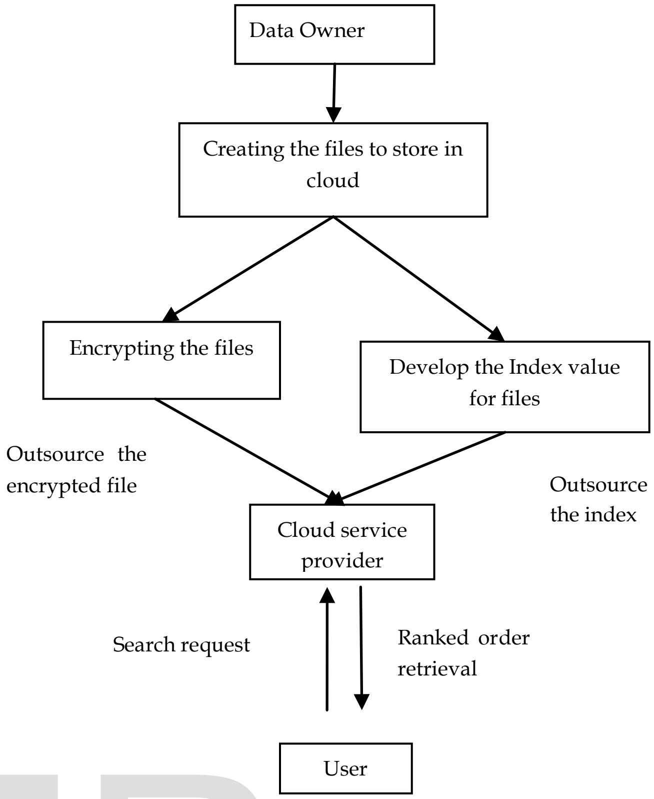

Fig 2: System flow diagram

5 RESULTS AND DISCUSSION

The performance of the existing and proposed system can be evaluated in terms search time, precision, recall and f-measure.

Search time here is defined to be the response time i.e.., time taken to complete the search of keyword in the cloud storage that is graphically shown below in Fig.1. It can be inferred from the graph that search of time of Existing system (RSA with ranking) is higher than that of Proposed (Ranking + KNN) which indicates that proposed system searches and provides faster results than existing.

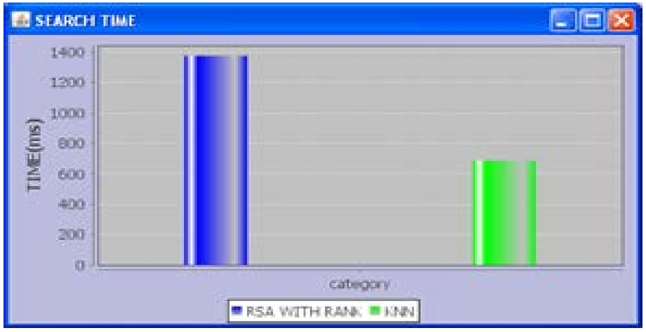

Fig 3: Search time graph

Precision (P) is the proportion of the predicted positive cases that were correct, as calculated using the equation:

P=d/(b+d)

It can be observed from the graph given below in Fig.2 that Precision of Proposed system is higher than the Existing System.

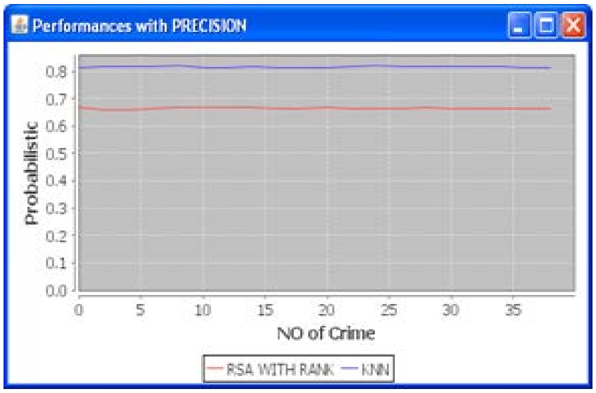

Fig 4: Precision graph

The recall or true positive rate (TP) is the proportion of positive cases that were correctly identified, as calculated using the equation:

TP=d/(c+d)

It can be inferred from the graph given below in Fig.3 that Recall of Proposed system is higher than the Existing System.

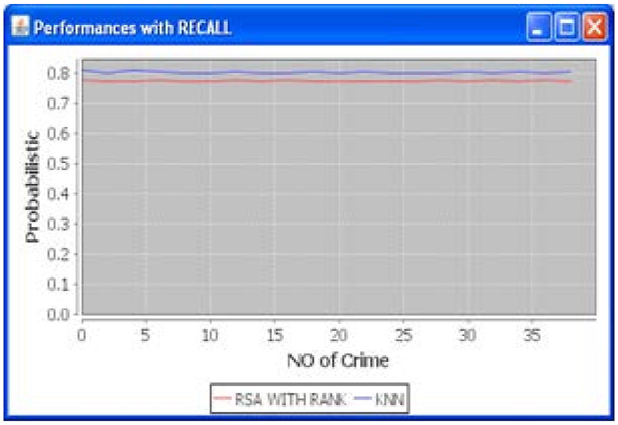

Fig 5: Recall graph

F-measure is a measure of a test's accuracy. It considers both the precision P and the recall R of the test to compute the score: P is the number of correct results divided by the number of all returned results and R is the number of correct results divided by the number of results that should have been returned. The score can be interpreted as a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and worst score at 0. It is given by,

F=2*((P*R)/(P+R))

It can be observed from the graph given below in Fig.2 that F-measure of Proposed system is higher than the Existing System.

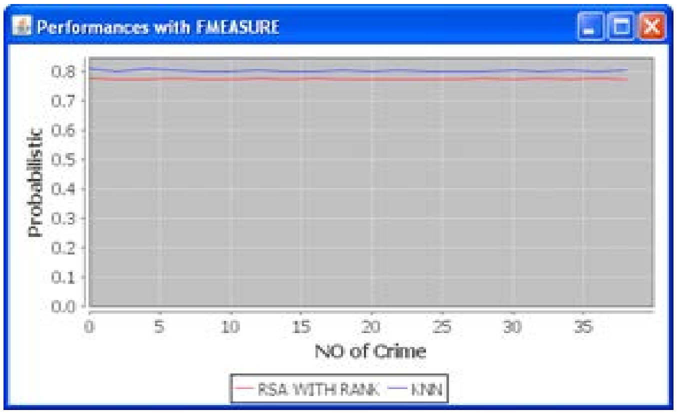

Fig 6: F-measure graph

6 CONCLUSION AND FUTURE WORK

The problem of supporting efficient ranked keyword search for achieving effective utilization of remotely stored encrypted data in Cloud Computing is solved. Initially a basic scheme is given and show that by following the same existing searchable encryption framework, it is very inefficient to achieve ranked search. Then the security guarantee is approximately weakened, resort to the newly developed crypto primitive OPSE, and derive an efficient one-to-many order preserving mapping function, and it allows the effective RSSE to be designed. The efficiency of the system can be further improved by our proposed KNN that develops high precision and recall rate leading to improved F-measure. We also liked to evince some further enhancements of our ranked search mechanism, including the efficient support of relevance score dynamics, the authentication of ranked search results, and the reversibility of our proposed one-to-many order-preserving mapping technique. Through thorough security analysis, we show that our proposed solution is secure and privacy preserving, while correctly realizing the goal of ranked keyword search. Extensive experimental results demonstrate the efficiency of our solution.

REFERENCES

[1] B. Zhang and F. Zhang, "An efficient public key encryption with conjunctive- subset keywords search", Journal of Network and Computer Application s, vol. 34, no. 1, (2011)

[2] Cong Wang,Ning Cao,Kui Ren and Wenjing Lou,"Enabling Secure and Efficient Ranked Keyword Search over Outsourced Cloud

Data",Proc.IEEE Transactions on parallel and distributed system,Aug2012

[3] C. Wang, K. Ren, S. Yu, K. Mahendra, and R. Urs, "Achieving Usable and Privacy-Assured Similarity Search over Outsourced Cloud Data,"Proc. IEEE INFOCOM,2012..

[4] D. Boneh, G.D.Crescenzo, R. Ostrovsky, and G.Persiano,"Public key encryption with keyword search," in Proc. of EUROCRYP'04, 2004. (2002) The IEEE website. [Online]. Available: http://www.ieee.org/

[5] E. Shi, J. Bethencourt, H. Chan, D. Song, and A. Perrig, "Multi-Dimensional Range Query over Encrypted Data,"Proc. IEEE Symp. Security and Privacy,2007.

[6] Jianfeng Wang,Xiaofeng Chen,Hua Ma,Qiang Tang and Jin Li, "A Verifiable Fuzzy Keyword Search Scheme Over Encrypted Data",Journal of Internet Services and Information Security (JISIS), volume: 2, number: 1/2, pp. 49-58

[7] Julien Bringer and Hervé Chabanne," Embedding edit distance to enable private keyword search"Springeropen journal 2012

[8] K. Ren, C. Wang, and Q. Wang, "Security Challenges for the Public Cloud,"IEEE Internet Computing, vol. 16, no. 1, pp. 69-73, 2012.

[9] M.Belare, A.Boldyreva, and A.O'Neil, "Deterministic and efficiently searchable encryption," in Proceedings of rypto 2007, volume 4622 of LNCS. Springer- Verlag, 2007.

[10] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-Preserving Multi-Keyword Ranked Search over Encrypted Cloud Data,"Proc. IEEE INFOCOM '11,2011

[11] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling PublicVerifiability and Data Dynamics for Storage Security in Cloud Computing,"IEEE Trans. Parallel and Distributed Systems, vol. 22,no. 5, pp. 847-859, May 2011

[12] S. Ji, G. Li, C. Li, and J. Feng. Efficient interactive fuzzy keyword search. In Proc. of 18th International World Wide Web Conference(WWW'09), Madrid, Spain. ACM, April 2009

[13] Sun-Ho Lee and Im-Yeong Lee,"Secure Index Management Scheme on Cloud Storage Environment" ,*International Journal of Security and Its Applications Vol. 6, No. 3, July, 2012*

[14] T.Balamuralikrishna,C.Anuradhaand N.Raghavendrasai, "Fuzzy keyword search over encrypted data over cloud computing",Asian Journal of Computer Science and Information Technology 2011

[15] T. M Nisha and V. P Lijo ,"Improving the Efficiency of Data Retrieval in Secure Cloud by Introducing Conjunction of Keywords",