International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 7, Issue 6, June 2017)

Secure and Dynamic Multi-keyword Ranked Search Scheme and Fuzzy Search over Encrypted Cloud Data

Ithape Pramod[1], Harsh Mathur[2]
[1]M.Tech Pursuing Student, [2]Associate Professor, Department of CSE

***Abstract*-- With the increased rate of growth and adaptation of cloud computing, daily, more and more sensitive information is being centralized onto the cloud. For the protection of valuable proprietary information, the data must be encrypted before outsourcing. There is no tolerance for typos and format inconsistencies which are normal user behavior. This makes effective data storage and utilization a very challenging task, rendering user searching very frustrating and inefficient. In this paper, we focus on secure storage using Advanced Encryption Standard (AES) and information retrieval by performing fuzzy keyword search on this encrypted data. We are proposing the implementation of an advanced fuzzy keyword search mechanism based technique which returns the matching files when users' searching inputs exactly match the predefined keywords or the closest possible matching files based on similarity keyword semantics, when exact match fails. In the proposed solution, we exploit edit distance to quantify keywords similarity and develop an efficient technique for constructing fuzzy keyword sets, which focus on reducing the storage and representation overheads.**

***Keywords*-- Fuzzy search, encryption on cloud, cloud computing, AES**

I. INTRODUCTION

Due to the flexibility and economic savings offered by the cloud server, the users have been motivated to outsource the management of their data to the cloud. However, because of privacy concerns, data owners encrypt sensitive data prior to outsourcing, which in turn makes data utilization a challenging problem. Thus, development of an efficient Secure and Dynamic Multi-keyword Ranked Search Scheme and Fuzzy Search over encrypted cloud data is of great importance. The most common search methods retrieve files using keywords instead of retrieving all the encrypted files back. To securely searching over encrypted data, the data owner usually builds an encrypted index structure using the extracted keywords from the data files and a corresponding index-based keyword matching algorithm and subsequently outsources both the encrypted data and this constructed index structure to the cloud.

When searching the files, the cloud server integrates the trapdoors of the keywords with the index information and then returns the corresponding files to the data users. Moreover, the data owner can share their data with a large number of users which requires the cloud server to have the ability to meet a large amount of requests with effective data retrieval services. One effective method for solving this problem we proposed Advanced Encryption Standard (AES) and information retrieval by performing fuzzy keyword search.

This paper proposes a secure tree-based search scheme over the encrypted cloud data, which supports multikeyword ranked search and dynamic operation on the document collection. Specifically, the vector space model and the widely-used "term frequency (TF) × inverse document frequency (IDF)" model are combined in the index construction and query generation to provide multikeyword ranked search. In order to obtain high search efficiency, we construct a tree-based index structure and propose a "Greedy Depth-first Search" algorithm based on this index tree. Due to the special structure of our tree-based index, the proposed search scheme can flexibly achieve sub-linear search time and deal with the deletion and insertion of documents. The secure kNN algorithm is utilized to encrypt the index and query vectors, and meanwhile ensure accurate relevance score calculation between encrypted index and query vectors.

Thus, we focus on enabling effective efficient Secure and Dynamic Multi-keyword Ranked Search for information stored in cloud environments. Fuzzy keyword augments system usability by returning the matching files when user searching inputs exactly the predefined keywords or the closest possible matching files based on keywords similarity semantics, when exact match fails. Edit distance is used to quantify keyword similarity and for the development of a novel technique i.e. a wildcard based technique, for constructing fuzzy keyword sets. This technique eliminates the need for counting all the fuzzy keywords and the total size of the fuzzy keywords sets is significantly decreases.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 7, Issue 6, June 2017)

II. Related Work

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[2] 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.* [3] 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

Cao *et al.*[4] 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. Kamara *et al.* [5] 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 singlekeyword Boolean search. In [6], 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.*[7] 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 [8], 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.

III. Problem Formulation

In this paper, we consider a cloud data system consisting of cloud server, data owner and data user. With the prevalence of cloud services, more and more sensitive information are being centralized into the cloud servers, such as emails, personal health records, private videos and photos, company finance data, government documents, etc. To protect data privacy and combat unsolicited accesses, sensitive data has to be encrypted before outsourcing so as to provide end-to-end data confidentiality assurance in the cloud and beyond. However, data encryption makes effective data utilization a very challenging task given that there could be a large amount of outsourced data files. Besides, in Cloud Computing, 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. One of the most popular ways to do so is through keyword-based search. Such keyword search technique allows users to selectively retrieve files of interest and has been widely applied in plaintext search scenarios. Unfortunately, data encryption, which restricts user's ability to perform keyword search and further demands the protection of keyword privacy, makes the traditional plaintext search methods fail for encrypted cloud data.

Disadvantage:

1. For each search request, users without pre-knowledge of the encrypted cloud data have to go through every retrieved file in order to find ones most matching their interest, which demands possibly large amount of post processing overhead.
2. Invariably sending back all files solely based on presence/absence of the keyword further incurs large unnecessary network traffic, which is absolutely undesirable in today's pay-as-you-use cloud paradigm.

IV. Proposed Work

This paper proposes a secure tree-based search scheme over the encrypted cloud data, which supports multi keyword ranked search and dynamic operation on the document collection. Specifically, the vector space model and the widely-used "term frequency (TF) × inverse document frequency (IDF)" model are combined in the index construction and query generation to provide multi keyword ranked search. In order to obtain high search efficiency, we construct a tree-based index structure and propose a "Greedy Depth-first Search" algorithm based on this index tree.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 7, Issue 6, June 2017)

And to secure encrypted & decrypted data provide an AES algorithm. Due to the special structure of our tree-based index, the proposed search scheme can flexibly achieve sub-linear search time and deal with the deletion and insertion of documents. The secure kNN algorithm is utilized to encrypt the index and query vectors, and meanwhile ensure accurate relevance score calculation between encrypted index and query vectors.

kNN Algorithm

The training examples are vectors in a multidimensional feature space, each with a class label. The training phase of the algorithm consists only of storing the feature vectors and class labels of the training samples.

In the classification phase, *k* is a user-defined constant, and an unlabeled vector (a query or test point) is classified by assigning the label which is most frequent among the *k* training samples nearest to that query point.

A commonly used distance metric for continuous variables is Euclidean distance. For discrete variables, such as for text classification, another metric can be used, such as the **overlap metric** (or Hamming distance). In the context of gene expression microarray data, for example, *k*-NN has also been employed with correlation coefficients such as Pearson and Spearman Often, the classification accuracy of *k*-NN can be improved significantly if the distance metric is learned with specialized algorithms such as Large Margin Nearest Neighbor or Neighborhood components analysis.

AES Encryption:

AES comprises three block ciphers: AES-128, AES-192 and AES-256. Each cipher encrypts and decrypts data in blocks of 128 bits using cryptographic keys of 128-, 192- and 256-bits, respectively.

Symmetric (also known as secret-key) ciphers use the same key for encrypting and decrypting, so the sender and the receiver must both know -- and use -- the same secret key. All key lengths are deemed sufficient to protect classified information up to the "Secret" level with "Top Secret" information requiring either 192- or 256-bit key lengths. There are 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys -- a round consists of several processing steps that include substitution, transposition and mixing of the input plaintext and transform it into the final output of cipher text.

The AES encryption algorithm defines a number of transformations that are to be performed on data stored in an array.

The first step of the cipher is to put the data into an array; after which the cipher transformations are repeated over a number of encryption rounds. The number of rounds is determined by the key length, with 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys.

The first transformation in the AES encryption cipher is substitution of data using a substitution table; the second transformation shifts data rows, the third mixes columns. The last transformation is a simple exclusive or (XOR) operation performed on each column using a different part of the encryption key -- longer keys need more rounds to complete.

Architecture of Secure Search Scheme

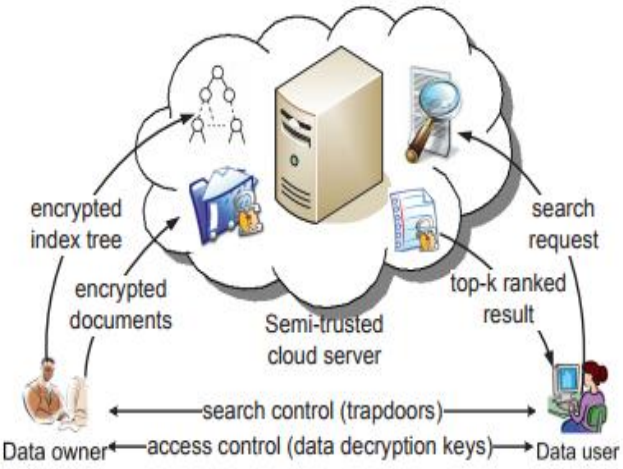


Fig. 1. The architecture of ranked search over encrypted cloud data

Modules

1. Data Owner module
2. Data User module
3. Semi-Trusted Cloud Server module

Module Description:

Data Owner:

The data owner is responsible for the update operation of his documents stored in the cloud server. While updating, the data owner generates the update information locally and sends it to the server.

Data User:

Data users are authorized ones to access the documents of data owner. He fetches encrypted documents from cloud server, and then he can decrypt the documents with the shared secret key.

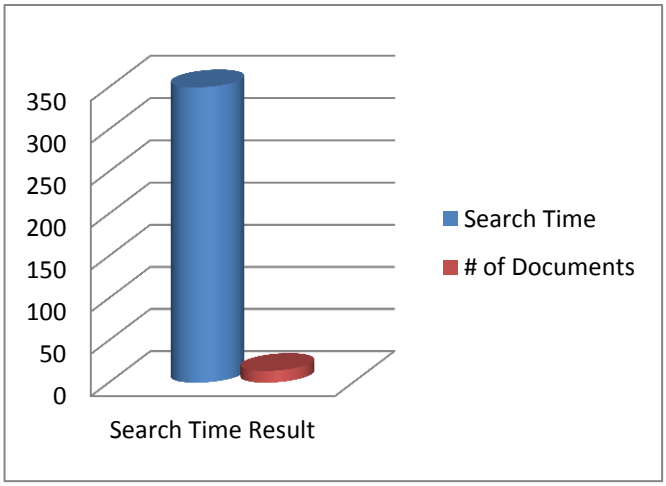
International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 7, Issue 6, June 2017)

Semi-Trusted Cloud Server:

Cloud server stores the encrypted document collection and the encrypted searchable tree index for data owner.

V. Performance Analysis

During the search process, if the relevance score at node u is larger than the minimum relevance score in result list RList, the cloud server examines the children of the node; else it returns. Thus, lots of nodes are not accessed during a real search. We denote the number of leaf nodes that contain one or more keywords in the query as generally is larger than the number of required documents k, but far less than the cardinality of the document collection n. As a balanced binary tree, the height of the index is maintained to be log n, and the complexity of relevance score calculation is O(m). Thus, the time complexity of search is O (_mlog n). Note that the real search time is less than _mlog n. It is because 1) many leaf nodes that contain the queried keywords are not visited according to our search algorithm, and 2) the accessing paths of some different leaf nodes share the mutual traversed parts. In addition, the parallel execution of search process can increase the efficiency a lot



We test the search efficiency of the proposed scheme on a server which supports 24 parallel threads. The search performance is tested respectively by starting 1,4,8 and 16 threads. We compare the search efficiency of our scheme with that of Sun et al. [9]. In the implementation of Sun's code, we divide 4000 keywords into 50 levels. Thus, each level contains 80 keywords. According to [9], the higher level the query keywords reside, the higher the search efficiency is.

In our experiment, we choose keywords from the user uploads the files among with the corresponding set of keywords that are used later for perform fuzzy keyword search, with the keyword score of kNN which is nearest element and then user has to list all the files that has found the keywords and can be download from the cloud.

VI. Conclusion

In this paper, a secure, efficient and dynamic search scheme is proposed, which supports not only the accurate multikeyword ranked search but also the dynamic deletion and insertion of documents. We design an advanced search mechanism for constructing storage efficient fuzzy keyword sets based on the similarity metric.

VII. Future Scope

In the future scope of this system we are willing to do fuzzy sets so as to increase the functionality of the search procedure. Encryption of more file formats can be done. Also decryption of image files and media files can be done.

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] E.-J. Goh et al., "Secure indexes." IACR Cryptology ePrint Archive,vol. 2003, p. 216, 2003.

[3] 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

[4] 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.

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

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] "Request for comments," http://www.rfc-editor.org/index.html.

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

[12] 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.

[13] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving Secure, Scalable,and Fine-Grained Data Access Control in Cloud Computing," Proc. IEEE INFOCOM, 2010.