ISSN:0975-9646

Study of Multi-keyword Ranked Searching and Encryption Technique over Cloud

Mrs. **SonamDarda**
Computer Department
P.E.S Modern College of Engineering Pune-05

Prof. **Manasi. K. Kulkarni**
Computer Department
P.E.S Modern College of Engineering Pune-05

Abstract-The advantage of storage as a service many enterprises are moving their valuable data to the cloud, since it costs less, easily scalable and can be accessed from anywhere any time. The trust between cloud user and provider is paramount. They use security as a parameter to establish trust. Cryptography is one way of establishing trust. Searchable encryption is a cryptographic method to provide security. In literature many researchers have been working on developing efficient searchable encryption schemes. To protect data privacy, the sensitive data should be encrypted by the data owner before outsourcing, which makes the traditional and efficient plaintext keyword search technique useless. Hence, it is an important to explore secure encrypted cloud data search service. Considering the huge number of outsourced data, there are three problems we are focused on to enable efficient search service: multi-keyword search, result relevance ranking and dynamic update. In this study we present the comparative analysis of searching methods for keyword over cloud as well as various encryption techniques which is used for data security.

Keywords-Cloud Computing, Attribute-based Encryption, public keys, private keys, cipher text, Searchable encryption, multi-keyword ranked search, dynamic update

1. INTRODUCTION

In outsourcing sensitive information to cloud there is always a risk of unauthorized access. To protect data privacy, the sensitive data should be encrypted by the data owner before outsourcing, which makes the traditional and efficient plaint text keyword search technique useless. Hence, it is an important to explore secure encrypted cloud data search service. Considering the huge number of outsourced data, there arethree problems mainly focused i.e. multi-keyword search, result relevance ranking and dynamic update.

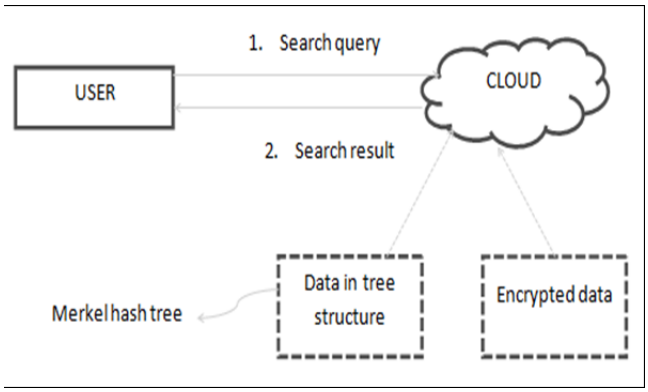

Fig.1 Cloud search system from user side

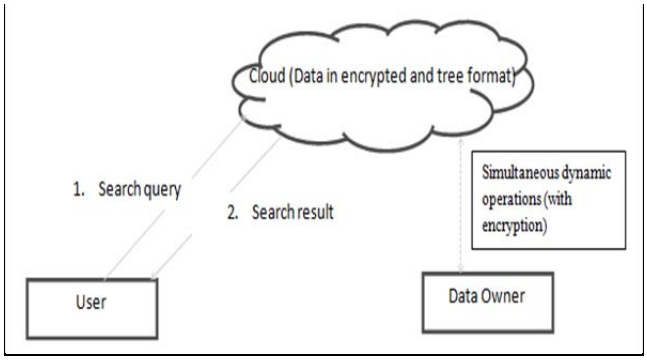

Fig.2 Cloud search system from data owner side

2. RELATED WORK

Lu Zhou, Zhangjie Fu [2] proposed privacy preserving in multi keyword rank search. This schema shows practically efficient and flexible searchable encrypted scheme which supports both multi-keyword ranked search and dynamic update. To support multi-keyword search and result relevance ranking, Vector Space Model (VSM) is used it will help to build the searchable index to achieve accurate search result. To improve search efficiency, a tree-based index structure which supports insertion and deletion update well without privacy leakage was introduced.

W. Sun, B. Wang [1], presents a privacy-preserving multi-keyword text search (MTS) scheme with similarity-based ranking to address this problem. To support multi-keyword search and search result ranking, to build the search index based on term frequency and the vector space model with cosine similarity measure to achieve higher search result accuracy. To improve the search efficiency, a tree-based index structure and various adaption methods for multi-dimensional (MD) algorithm are proposed so that the practical search efficiency is much better than that of linear search, also two secure index schemes to meet the stringent privacy requirements under strong threat models, i.e., known cipher-text model and known background model.

Zhihua Xia, Xinhui Wang [3], present a secure multi-keyword ranked search scheme over encrypted cloud data, which simultaneously supports dynamic update operations of documents. Specifically, the vector space model and the widely-used TF_IDF model are combined in the index construction and query generation. For efficient multi-keyword rank search here proposed a special tree-based index structure and named it a “Greedy Depth-first Search” algorithm.

Bin Yao, Feifei Li, [5] present the approximate String Search in Spatial Databases using MHR tree concept. It is based on R-Tree augmented with min wise signature and linear hashing. Keyword will be search using hash keys which are identical to related set and element. pruning the tree according to signature and query string. Signature for an indexed node u keeps a very concise representation of node string under subtree.

For searching keyword various method are user like sequential search [6] and binary search [7][8] are introduce. The sequential search examines the first element in the list and then examines each "sequential" element in the list (in the order that they appear) until a match is found. Sequential search is easy to compute but fails when dataset large in number. On other hand binary search works different than sequential search. For binary search data should be sorted, this makes binary search more complicated as a result it also not useful to compute large dataset.

To overcome such disadvantages of sequential and binary search PokornyJ [4] proposed multi-dimension B-tree for large data. The multidimensional binary search tree is a data structure for storage of information to be retrieved by associative searches. The k-d tree is defined and examples are given. It is shown to be quite efficient in its storage requirements. A significant advantage of this structure is that a single data structure can handle many types of queries very efficiently.

To provide a security to data there is need of encryption of data. Liang Wang [11] proposed a personal information protection using RSA algorithm. The RSA algorithm involves three steps: key generation, encryption, and decryption. RSA involves two keys – a public key and a private key. As the names suggest, anyone can be given information about the public key, whereas the private key must be kept secret. Anyone can use the public key to encrypt a message, but only someone with knowledge of the private key can hope to decrypt the message in a reasonable amount of time. The power and security of the RSA cryptosystem is based on the fact that the factoring problem is "hard." That is, it is believed that the full decryption of an RSA cipher text is infeasible because no efficient algorithm currently exists for factoring large numbers. But other hand RSA fails for complete security so later Joan Daemen [9] stated AES encryption technique for encryption. AES is based on a design principle known as a substitution-permutation network, combination of both substitution and permutation, and is fast in both software and hardware. Unlike its predecessor DES, AES does not use a Feistel network. AES is a variant of Rijndael which has a fixed block size of 128 bits, and a key size of 128, 192, or 256 bits. AES provide the excellent security over RSA.

Later around 2004 the concept of KP-ABE was introduced. John Bethencourt [12] stated attribute base encryption. Attribute-based encryption is a type of public-key encryption in which the secret key of a user and the cipher text are dependent upon attributes (e.g. the country he lives, or the kind of subscription he has). In such a system, the decryption of a cipher text is possible only if the set of attributes of the user key matches the attributes of the cipher text. A crucial security aspect of Attribute-Based Encryption is collusion-resistance: An adversary that holds multiple keys should only be able to access data if at least one individual key grants access.

3. EXISTING SEARCHING AND ENCRYPTION TECHNIQUES

3.1 Greedy DFS

Greedy DFS need to construct a result list denoted as RList, whose element is defined as ⟨RScore; FID⟩. Here, the RScoreis the relevance score of the document fFIDto the query. 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 updated timely during the search process[3].

RScore (Du,Q) – The function to calculate the relevance score for query vector Q and index vector Du stored in node u.

k^{th}score– The smallest relevance score in current RList, which is initialized as 0.

hchild– The child node of a tree node with higher relevance score.

lchild– The child node of a tree node with lower relevance score.

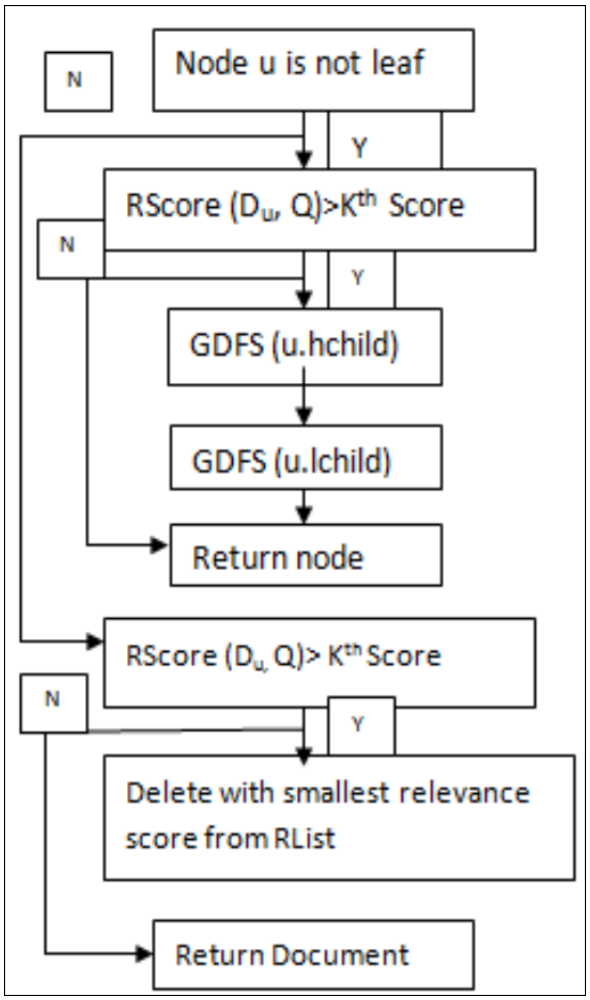

Fig.3 Greedy DFS algorithm

3.2 MHR Tree

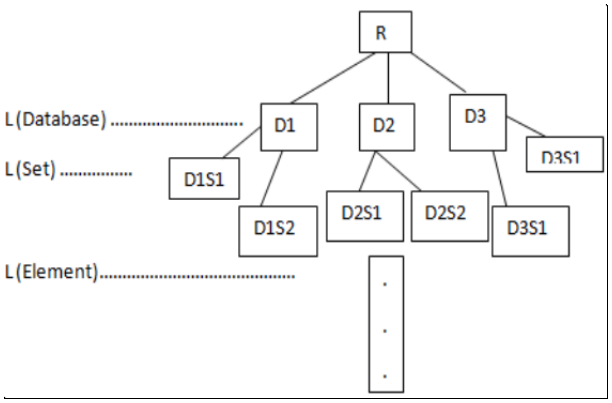

Fig.4 MHR Tree Construction.

It is based on R-Tree augmented with min-wise signature and linear hashing. Keyword will be searched using hash key which is identical to related set and element. It prunes the tree according to signature and query string. Signature for an indexed node 'u' keeps a very concise representation of node string under subtree [5].

3.3 MDB Tree

The multidimensional binary search tree (or k-d tree, where k is the dimensionality of the search space) as a data structure for storage of information to be retrieved by associative searches. The k-d tree is defined and examples are given. It is shown to be quite efficient in its storage requirements. A significant advantage of this structure is that a single data structure can handle many types of queries very efficiently [4]. Various utility algorithms are developed; their proven average running times in an n record file are: insertion, O(log n); deletion of the root, O(n(k-1)/k); deletion of a random node, O(log n); and optimization (guarantees logarithmic performance of searches), O(n log n). Search algorithms are given for partial match queries with t keys specified [proven maximum running time of O(n(k-t)/k)] and for nearest neighbor queries [empirically observed average running time of O(log n).] These performances far surpass the best currently known algorithms for these tasks. An algorithm is presented to handle any general intersection query.

3.4 Sequential Search

The sequential search examines the first element in the list and then examines each "sequential" element in the list (in the order that they appear) until a match is found. This match could be a desired word that you are searching for, or the minimum member in the list [6]. Variation on this include, searching a sorted list for all occurrences of a data value (or counting how many matches occur: inventory), or searching an unsorted list for first occurrence or every occurrence of a data value.

3.5 Binary Search

The binary search algorithm begins by comparing the target value to the value of the middle element of the sorted array. If the target value is equal to the middle element's value, then the position is returned and the search is finished. If the target value is less than the middle element's value, then the search continues on the lower half of the array; or if the target value is greater than the middle element's value, then the search continues on the upper half of the array. This process continues, eliminating half of the elements, and comparing the target value to the value of the middle element of the remaining elements - until the target value iseither found (and its associated element position is returned), or until the entire array has been searched (and "not found" is returned)[7][8].

3.6 AES Algorithm (Advanced Encryption Standard)

AES is based on a design principle known as a substitution-permutation network, combination ofboth substitution and permutation, and is fast in both software and hardware. Unlike its predecessor DES, AES does not use a Feistel network. AES is a variant of Rijndael which has a fixed block size of 128 bits, and a key size of 128, 192, or 256 bits[9].

AES operates on a 4×4 column-major order matrix of bytes, termed the *state*, although some versions of Rijndael have a larger block size and have additional columns in the state. Most AES calculations are done in a special finite field.

For instance, if you have 16 bytes, $b_0,b_1,...,b_{15}$, these bytes are represented as this matrix:

$$\begin{bmatrix} b_0 & b_4 & b_8 & b_{12} \\ b_1 & b_5 & b_9 & b_{13} \\ b_2 & b_6 & b_{10} & b_{14} \\ b_3 & b_7 & b_{11} & b_{15} \end{bmatrix}$$

The key size used for an AES cipher specifies the number of repetitions of transformation rounds that convert the input, called the plaintext, into the final output, called the cipher text [10].

3.7 RSA

The RSA algorithm involves three steps: key generation, encryption, and decryption. RSA involves two keys – a public key and a private key. As the names suggest, anyone can be given information about the public key, whereas the private key must be kept secret [11]. Anyone can use the public key to encrypt a message, but only someone with knowledge of the private key can hope to decrypt the message in a reasonable amount of time. The power and security of the RSA cryptosystem is based on the fact that the factoring problem is "hard." That is, it is believed that the full decryption of an RSA cipher text is infeasible because no efficient algorithm currently exists for factoring large numbers.

3.8 ABE (Attribute Based Encryption)

Attribute-based encryption is a type of public-key encryption in which the secret key of a user and the cipher text are dependent upon attributes (e.g. the country he lives, or the kind of subscription he has). In such a system, the decryption of a cipher text is possible only if the set of attributes of the user key matches the attributes of the cipher text . A crucial security aspect of Attribute-Based Encryption is collusion-resistance: An adversary that holds multiple keys should only be able to access data if at least one individual key grants access [12].

3.9 BLOWFISH Algorithm

Blowfish is a fast block cipher, except when changing keys. Each new key requires pre-processing equivalent to

encrypting about 4 kilobytes of text, which is very slow as compared to other block ciphers [13].

In one application Blowfish's slow key changing is actually a benefit: the password-hashing method used in Open BSD uses an algorithm derived from Blowfish that makes use of the slow key schedule, the idea is that the extra computational effort required gives protection against dictionary attacks [14].

4. COMPARATIVE STUDY OF SEARCHING AND ENCRYPTION TECHNIQUES

TABLE I
COMPARATIVE STUDY OF ENCRYPTION ALGORITHMS

Parameter	AES	RSA	Blowfish	KP-ABE
Developed	2000	1978	1993	2004
Key size	128,192, 256 bit	>1024 bit	32-448 bit	NA
Block size	128 bit	Min 512 bit	64 bit	NA
Scalable	Not	Not	Not	Yes
Algorithm	Symmetric	Asymmetric	Symmetric	Symmetric
Encryption	Faster	Slower	Faster	Faster
Decryption	Faster	Slower	Faster	Faster
Security	Excellent	Less secure	Secure	Excellent

TABLE II
COMPARISON BETWEEN SEARCHING TECHNIQUES

Algorithm	Constraint	Speed	Time taken
Greedy DFS	NA	Fast	Less
MHR Tree	NA	Faster	Much Less
MDB Search	NA	Average	Average
Sequential search	Data should be limited	Comparative slow	Comparative more
Binary search	Sorted data needed	Comparative slow	Comparative more

5. CONCLUSION AND FUTURE WORK

After studying and analysing all methods we conclude it we can contribute mainly in two aspects: similarity ranked search for more accurate search result and MHR tree-based searchable index for more efficient searching and dynamic updating. Our proposed searching technique i.e. MHR is useful in large databases. Results show that proposed algorithm is better in terms of search complexity and time complexity. Finally, we analyze the performance of our scheme in detail by experimenting on real-world dataset. But, there still exist some problems, such as how to further reduce the time cost for index tree construction and so on. We will do more research in the future. Further, we intend to analyze the behavior of our proposed system(s) for multiuser environment.

ACKNOWLEDGMENTS

Every orientation work has an imprint of many people and it becomes the duty of author to express deep gratitude for the same. I take this opportunity to express my deep sense of gratitude towards my esteemed guide Prof. Manasi.K.Kulkarni for giving me this spleen did opportunity to present this paper.

REFERENCES

[1] 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.

[2] Xingming Sun, Lu Zhou, Zhangjie Fu and Jin Wang., “Privacy-preserving Multi-keyword Ranked Search over Encrypted Cloud Data Supporting Dynamic Update”, Vol.8, No.6 (2014), pp.1-16,2014.

[3] Zhihua Xia, Xinhui Wang, Xingming Sun, “A Secure and Dynamic Multi-keyword Ranked Search Scheme over Encrypted Cloud Data”, on parallel and distributed systems vol: pp no: 99 year 2015.

[4] Ondreicka, M, Pokorn´y J,”Extending Fagin’s algorithm for more users based on multidimensional B-tree”, In: Proc. of ADBIS 2008, LNCS 5207, 2008, pp. 199-214.

[5] Bin Yao, Feifei Li, “Approximate String Search in Spatial Databases”, Computer Science Department, Florida State University, Tallahassee, FL, USA.

[6] Heydari, J, “Quickest sequential search over correlated sequences”, ECSE Dept., Rensselaer Polytech. Inst., Troy, NY, USA

[7] Zhenzheng Ouyang, Quanyuan Wu, Tao Wang "An Efficient Decision Tree Classification Method Based on Extended Hash Table for Data Streams Mining", Fuzzy Systems and Knowledge Discovery, 2008. FSKD '08. Fifth International Conference on, On page(s): 313 - 317 Volume: 5, 18-20 Oct. 2008

[8] Tao Wang, “A new fuzzy decision tree classification method for mining high-speed data streams based on binary search trees” ,Nat. Univ. of Defense Technol., Changsha, Second international conference on System , 2007. ICONS ’07.

[9] Joan Daemen and Vincent Rijmen, "The Design of Rijndael: AES-The Advanced Encryption Standard," Springer-Verlag, 2002.

[10] National Inst. Of Standards and Technology, "Federal Information Processing Standard Publication 197, the Advanced Encryption Standard (AES)," Nov. 2001

[11] Liang Wang, Yonggui Zhang, 2011, “A New Personal Information Protection Approach Based on RSA Cryptography”, IEEE.

[12] John Bethencourt Carnegie Mellon University, Cipher text -Policy Attribute-Based Encryption.

[13] Strong Encryption,http://www.tropsoft.com/strongenc/blowfish.htm(dated 1/Feb/2004) .

[14] B. Schneier, Description of a New Variable-Length Key, 64-bit Block Cipher (Blowfish), Proceedings of Fast Software Encryption, Security Cambridge 1994, pp. 191-204