RESEARCH ARTICLE OPEN ACCESS

Multiuser Multi-Keyword Ranked Search over Encrypted Cloud Using MHR and KP-ABE

Sonam.K.Darda [1], Prof. (Mrs). Manasi.K.Kulkarni [2]
Department of Computer Science and Engineering
P. E. S. Modern College of Engineering,
Pune - India

ABSTRACT
The benefit of storage as a service several enterprises area unit moving their valuable information to the cloud, since it prices less, simply scalable and can be accessed from anyplace any time. However, delicate information like email, personal health record, Gov. record ought to be encoded before outsourcing for security prerequisites, which obsoletes information utilization like keyword based document retrieval. During this paper, we present Multiuser Multi-keyword Ranked Search scheme over Encrypted Cloud using MHR Tree and KP-ABE. The MHR tree (Markle hash tree) algorithm can achieve logarithmic search time and deal with the deletion and insertion of documents flexibly. Keyword Policy-Attribute based encryption (KP-ABE) is secure encryption technique, it provide the fine grained access control and high security on document collection.
Keywords:-Cloud Computing, Searchable encryption, MHR tree, Attribute-based Encryption, Multi-keyword ranked search.

I. INTRODUCTION

Cloud computing is the long imagined vision of computing as a utility wherever cloud customers will remotely store their information into the cloud so as to enjoy the on-demand prime quality application and services from shared pool of configurable computing resources [1]. To ensure data security and combat unsought access within the cloud and beyond, sensitive information, e.g., emails, personal health records, photograph albums, tax documents, budgetary exchange, etc., might have to be encoded by data owners before outsourcing to business public cloud. Despite of the benefits, the cloud computing infrastructure faces very challenging tasks, particularly on data privacy, security and reliability issues. Therefore, providing high security and privacy protections on delicate information is extremely necessary. To prevent data revealing, mainstream solution is to encipher private data before uploading it onto the cloud server. But, there are severe processing limitations on encrypted data. As an sample, standard plain text based searching algorithms are not applicable. To resolve this issue, current solutions use the following strategy to provide keyword based searching capabilities on encrypted data. Fig.1 demonstrates the stream of generalized architecture of multi-keyword search [1]. The framework model involves 3 different entities: data owner, data user and cloud server. Data owner has a collection of documents D = (d1, d2, …., dn) that he/she wants to outsource to the cloud server. First, preprocessing (Tokenization, Stop word removal, Stemming) is done on document and keywords are extracted, after that hash code is calculated for every keyword and then Merkle hash tree is generated. Hash map maintains the information of which keywords this files contains. On other hand data owner gives some attributes, defines the policy on that attribute for each and every document and that document get encrypted on those policies by using KP-ABE algorithm. Second both the encrypted document and Merkle tree uploaded onto the data center servers in the cloud. Now, the data is ready to simply accept queries from the user.

Fig. 1. Basic Architecture

The cloud server can then support queries as follows, A data consumer submits a Multi-keyword query and attributes to the cloud server. The cloud server conduct search using Merkle tree search and returns only list of files name in which keyword is present. Once receiving encrypted files, the user

decrypts the files if the set of attributes of the user matches the policy of data owner. This approach guarantee the information security and preserve the data privacy. Throughout the complete process, no plain text data or keywords are visible to the cloud servers. Our contribution are summarized as follows:

1. We design a secure encryption scheme that supports accurate multi-keyword search using Merkle hash tree and provide the security by using KP-ABE encryption algorithm.
2. Due to special structure of our tree based the search complexity of implemented scheme is logarithmic.
3. Implementation scheme achieves the scalability, if number of user increases then also system is not affected.

II. LITERATURE SURVEY

A Encryption is an easiest way to secure the privacy of data. It is difficult to search documents over an encrypted data, so that to improve the efficiency of searching many techniques are available.

W. Sun, B. Wang [3], presents MTS scheme with similarity-based ranking. Author build the search index based on term frequency and the vector space model with cosine similarity measure to attain higher search result accuracy. To enhance the search efficiency, a tree-based index structure for multi-dimensional (MD) algorithm are proposed so that the search efficiency is much better than that of linear search, also two secure index schemes to satisfy the stringent privacy requirements under strong threat models, i.e., known cipher-text model and known background model. It provide the better efficiency but result in precision loss.

Zhihua Xia, Xinhui Wang [1], present a secure multi-keyword ranked search scheme over encrypted cloud data to overcome the drawback of [2][3], which simultaneously supports dynamic update operations of documents. Specifically, the vector space model and the widely-used TF*IDF model are combined within 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. Due to the special structure of our tree-based index, the proposed scheme can flexibly achieve sub linear search time and cope with the deletion and insertion of documents.

Bin Yao, Feifei Li, [8] 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 are going to be search using hash keys which are identical to related set and element. Pruning the tree according to signature and query string. During this tree information is arranged in tree form before giving any query and after we give any query then we get hash code associate to keyword. So, it is more efficient, low time cost required compare to alternative.

To overcome disadvantages of sequential[10] and binary search [6] Pokorn'y J [9] proposed multi-dimension B-tree for large data. The multidimensional binary search tree is a data structure for storage of data to be retrieved by associative searches. A major advantage of this structure is that a single data structure can handle
many varieties of queries very efficiently.

To provide a security to data there is need of encryption of data. Liang Wang [11] proposed a 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. Anyone will use the public key to encrypt a message, but only someone with knowledge of the nonpublic (private) key can hope to decrypt the message in a reasonable amount of time. The full decryption of an RSA cipher text is unfeasible because no efficient rule currently exists for factoring large numbers. But in other hand Joan Daemen [9] declared AES encryption technique for encryption. AES is based combination of both substitution and permutation, and is fast in both software and hardware. AES could be a variant of Rijndael that has a fixed block size of 128 bits, and a key size of 128, 192, or 256 bits. AES provide the very excellent security over RSA.

Later around 2004 the concept of KP-ABE was introduced. John Bethencourt[17] explicit attribute base encryption. Attribute-based encryption could be a sort 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 type of subscription he has). In such a system, the decryption of a cipher text is feasible only if the set of attributes of the user key matches the attributes of the cipher text.

III. SYSTEM ARCHITECTURE

Steps for system modules:

1. User Side:

(a) It sends the multi-keyword query request and user attributes to cloud server and fetch encrypted document from cloud server. Then the data user will decrypt the documents with the shared secret key if and only if user attribute satisfy the cipher text policy.

2. Data Owner Side:

(a) It builds a secure searchable tree using Merkle hash tree, first preprocessing (Tokenization, Stop word removal,

Stemming) is conducted over the set of document. Hash code will be generated for each keyword and Merkle tree is constructed which are identical to related set and element.

(b) Subsequently data owner defines the attribute and policy for every document and then encrypt document on that policy using KP-ABE algorithm. Then the data owner outsources the encrypted documents and secure index to the cloud server.

3. Cloud Server:

(a) It stores the encrypted document collection and the encrypted searchable tree for data owner.

(b) The cloud server executes search over the hash tree using MHR tree search technique. It first tokenize the multi-keyword query, calculate hash code for each keyword and then searches query keyword hash code into Merkle tree and finally returns the corresponding collection

of top k encrypted document if and only if user attributes matches with the data owners access tree structure.

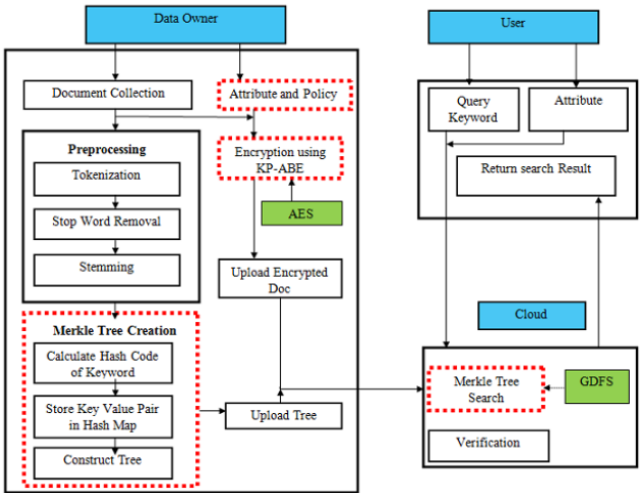

Fig.2 System Architecture

A. *Design Goals*

To enable secure, efficient, accurate and dynamic multi-keyword search over encrypted cloud data, the system has the following design goals:

- Multi-keyword Search Efficiency: To study search techniques over encrypted cloud data, which improve efficiency by using Merkle hash tree algorithm.
- Privacy Preserving: To prevent documents from varied attacks and preserve in its original kind over the cloud using (KP-ABE).
- Scalability: If number of user increases then also systems performance is not affected.

B. *GDFS Algorithm*

In "Greedy Depth first Search (GDFS) algorithm"[1], search process is a recursive procedure upon the binary tree. In which TF*TDF and vector space model is used for creation of index and generation of query. Here, author construct a result list denoted as RList, whose element is defined as (Rscore, FileID). The RScore is the relevance score of the document fFileID to the query. Largest relevance scores of k accessed documents is stored in RList. According to the Rscore the keyword in the list are ranked in descending order, and will be updated timely during the search process.

C. *MHR Tree Algorithm*

Markle tree is tree [8] in which every non-leaf node is labeled With the hash of the labels of its children nodes. Its hash value is calculating by concatenation of its left and right child hash value and all leaf nodes stored the hash of data blocks, in that files/tokens hash value is stored. It uses the perfect binary tree of left and right chilled. A perfect binary tree of depth h contains 2^h leaves. In other terms if range contains n tokens then the Merkle tree representing log(n) levels. Hash map stored the (key,value) pair for all tokens. Tree traversing is done using hash key of each node. Each node has a hash key and keyword will be search using hash key which is identical to related set and element. It pruning the tree according to the query string. Tree hash function randomly select the range and level and according to that it arrange the node in tree, all documents/tokens present in leaf nodes.

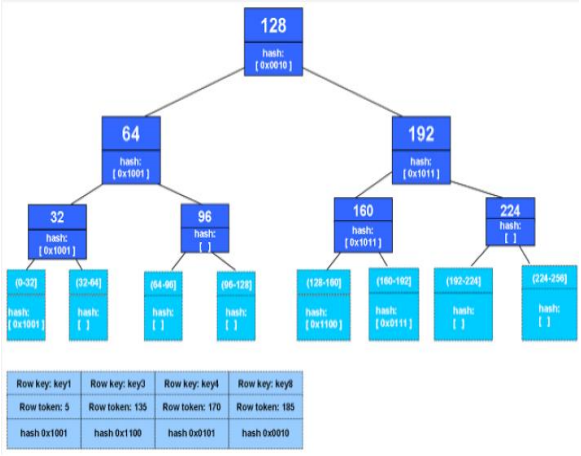

Fig. 3. MHR Tree Algorithm

For example, the token range(0,256), it contains 256 sub ranges (0,1),(1,2),…., (255,256) each containing single token. A perfect binary tree of depth 8 is required to store all 256 sub range hashes at leaves. Consider depth 3 for the same range contains only 8 leaves representing hashes of sub range (0,32),(32,64),..,(224,56) each containing 32 tokens. In all 32 tokens separate hash value is calculated for each token.

D. KP-ABE Algorithm

In this plan data encryption and decoding is depend on the attributes of end client where the mystery key produced for the attributes of the data. It utilizes attributes for describing encoded data and it will form an access tree structure for those attributes. This policy will generate user keys for the data that was encrypted and certain access control tree policies are embedded into the private key of the users. The encryptor is just permitted to know the decryptors public attributes. If the legitimate user needs to ingress the data from storage cloud, then end user attribute sets should match with the access control policy in order to get access for effective data retrieval. This plan is essentially intended for a single person to multiple group communications. This algorithm manages the four stage process as demonstrated as follows:

Steps:

Step1: Setup(p, U). Input limit are the security parameter p and the universe of attribute U. It create a master key MK along with the public key PK by using the bilinear group pairing(Type A symmetric pairing) and SHA-1 algorithm. It defines a bilinear group G1 of prime order p with generator q, a bilinear map is symmetric if G1 * G1 → G2

$$PK : \{Y,T_1,T_2,\ldots,T_N\} \quad \ldots\ldots(1)$$

$$MK : \{y,t_1,t_2,\ldots,t_N\} \quad \ldots\ldots(2)$$

Where Ti ε G1and ti ε Zp are attributes i, $1 \leq i \leq N$, Y ε G2, is another public key component. $Y = e(g; g)^y$, y ε Zp

Step2:Encryption (PK,M,AP). The public key PK, the message M and the access policy AP specified as a Boolean formula whose operands are a subset of the universe of attributes are taken as the input for this algorithm. Then, this algorithm encrypts M as the cipher text CT in such a way that only those user who has the set of attributes required to satisfy the access policy AP, will be able to decrypt it. It is assumed that the cipher text and the access policy AP must be transmitted together as a pair. For encryption it uses the symmetric algorithm(AES), it first encrypt policy(CPH) by using data owners public key and then this policy is embed into encrypted document and this whole document upload onto the cloud server.

$$CT = \{AP, CT = MY^s , \{CT = T^s_i\} \; i \varepsilon AP \} \quad \ldots\ldots\ldots(3)$$

Where s ε Zp, M ε G2, i ε {U = 1, 2, …,n}, s is randomly chosen from Zp

Step3: Key generation (PK,MK,S). The master key MK along with a set of attributes S is taken as the input in this algorithm. Then, the private key PR for user is generated with the prescribed set of attributes. If this key and data owners policy(CPH) is matched then user gets documents for decryption.

$$PR = \{ski\}i\varepsilon L \quad \ldots\ldots\ldots\ldots(4)$$

Where L= set of attributes attach to the leaf nodes of T

Step4:Decryption(PK,CT,PR). This primitive will be able to recover the message M from the cipher text CT, in the case that the set of attributes of user satisfies the policy AP. It takes public key for data owner, private key for user and cipher text as a input. It first computes $e(CT, sk_i) = e(g, g)^{pi(0)s}$ For leaf nodes. Then it aggregates this pairing results in bottom up manner using polynomial interpolation technique. Finally it recover blind factor $Y^s = e(g; g)^y$ and gives the output as a message M.

$$M = \{ e(CT, sk_i) = e(g, g)^{pi(0)s}$$
$$Y^s = e(g; g)^y \; \}$$

$$\ldots\ldots\ldots\ldots\ldots(5)$$

IV. EXPERIMENTAL SETUP AND PERFORMANCE EVALUATION

We implement system using Java (Net Beans 8.0) as front end and SQL as back end in Windows 7 operating system and test its efficiency on real world document collection. The Request for Comments (RFC), in that files in .txt format and check efficiency on 500 documents[13].

Test includes:

1. Precision
2.Search time for searching using Merkle tree search algorithm.

Most of the results are obtained with Intel Core (TM) Duo processor. These are the results of Merkle tree search. It gives the logarithmic time complexity and it compare with GDFS [1] search algorithm. Greedy DFS gives sub linear time complexity.

Experimental Results :

PRECISION :Precision (also called positive predictive value) is the fraction of retrieved instances that are relevant.

$$\text{Precision} = \frac{\{\text{RelevantDocument}\} \cap \{\text{RetrivedDocument}\}}{\{\text{RetrivedDocument}\}}$$

A text search on a set of documents precision is the number of correct results divided by the number of all returned results. By comparing index ranked search(GDFS) [1] and Merkle tree search, Merkle tree search gives high precision.

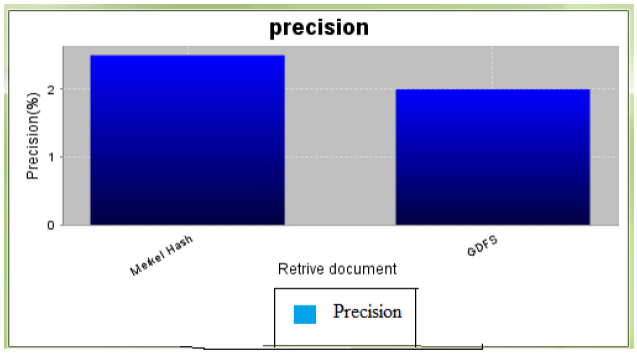

Fig.4. Precision of retrieved document

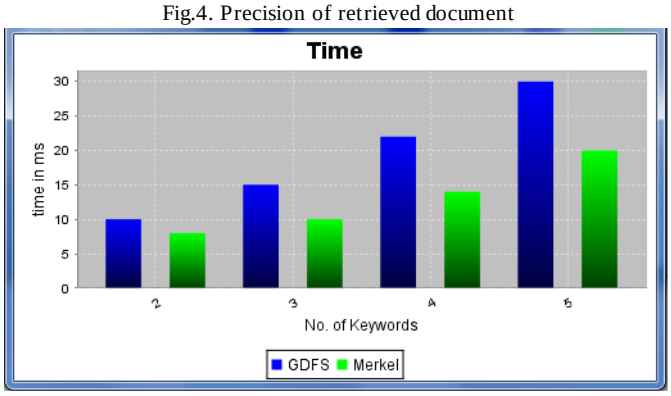

Fig.5. Searching time for keyword search

During the search process, Merkle Hash Tree calculate the hash code for the keyword and searches in the tree which we have build from hashing, so comparison will easy and it gives O(log n) time complexity. In other hand GDFS search algorithm [1] gives O(ωmlog n). Fig.5 shows the particular number of query how much time is required for searching. For example suppose 2 keyword query required 8ms.

V. CONCLUSIONS

In this paper we are implemented an improved technique for multi-keyword searching over secured cloud server. We make contribution mainly in two aspects: Merkle hash tree search algorithm for fast searching and key policy attribute based encryption for providing more security on document over cloud. In terms of efficiency, we implement the hash tree, in that all data present in at leaf node in the hash code format, so searching over the hash tree is easy, it gives conjunctive as well as disjunctive search result. This tree search only in leaf nodes hence its time complexity is O(log n). KP-ABE provide one extra level of security. It achieve fine grained access control, scalability and data confidentiality simultaneously.

ACKNOWLEDGMENT

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 apportunity 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] Zhihua.X, Xinhui.W, Xingming.S, Qian, "Secure and Dynamic Multi-keyword Ranked Search over Encrypted Cloud Data," IEEE Transaction on Parallel and Distributed Systems Vol:pp No:99 Year 2015.

[2] 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. 829837.

[3] W. Sun,B. Wang,N. Cao, M. Li, W. Lou, Y. 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. 7182.

[4] 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. 276286.

[5] Li.C, Xingming.S, Zhihua.X and Qi.L, "An Efficient and Privacy-Preserving Semantic Multi-Keyword Ranked Search over Encrypted Cloud Data," Vol.8,No.2 (2014), pp.323-332,2014.

[6] Prasanna.B , Akki.C, "Dynamic Multi-Keyword Ranked Searchable Security Algorithm Using CRSA and B-Tree," Vol. 6 (1) ,2015,826-83.

[7] Ruixuan.L,Zhiyong.X,Wanshang.K, Kin.Y, Cheng.X, "Efficient Multi-Keyword Ranked Query over Encrypted Data in Cloud Computing," Future Generation Computer System ELSEVIER 2014.

[8] Feifei.L, Bin.Y, Mingwang.T, and Marios.H, "Spatial Approximate String Search," IEEE Transaction on Knowledge and Data Engineering , VOL. 25, NO. 6, JUNE 2013.

[9] Ondreicka, M, Pokorny J, "Extending Fagins algorithm for more users based on multidimensional B-tree," In: Proc. of ADBIS 2008, LNCS 5207, 2008, pp. 199-214.

[10] Zhangjie.F, Xingming.S, Nigel.L and Lu.Z, "Achieving Effective Cloud Search Services: Multi-keyword Ranked Search over Encrypted Cloud Data Sup porting Synonym Query," IEEE Transactions on Consumer Electronics, Vol. 60, No.1, February 2014.

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

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

[13] Request for Comments, http://www.rfceditor.org/index.html (2011).

[14] Mayank.P and Tanushri.M,A, "Survey of Cryptographic based Security Algorithms for Cloud Computing," HCTL Open Int.J. of Technology Innovations and Research HCTL Open IJTIR, Volume 8, March 2014.

[15] Prachi.K and Natikar.B, "Keyword Query Routing Using MHR Tree," 3rd ed. International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online) Vol. 3, Issue 3, pp: (128-131), Month: July -September 2015.

[16] Dimpi.R and Rajiv.K , "Enhance data security of private cloud using encryption scheme with RBAC," International Journal of Advanced Research in Computer and Communication Engineering Vol. 3, Issue 6, June 2014.

[17] Kaitai.L and Willy.S, "Searchable Attribute Based Mechanism with Efficient Data Sharing for Secure Cloud Storage," IEEE Transaction on Information Forensics and Security. VOL.10,No.9 September 2015.