An Efficient and Secure BST Based Indexing of Encrypted Trigrams During Fuzzy Keyword Searching on Cloud

Josemary.A[1] and Shailesh.S[2]
[1] Department of Computer Engineering,Carmel Polytchnic College,Punnapra, Alappuzha, Kerala, India
[2] Department of Computer Science,Sacred Heart College, Thevara, Kochi, Kerala, India
Email: josemaryjos@hotmail.com[1], shaileshsivan@gmail.com[2]

***Abstract*—Cloud computing provides on-demand centralized information and services to the users from anywhere at any time. The users outsourced their valuable data to the cloud to reduce the storage requirement problems. Due to the increasing cloud usage, they must need some security features to maintain the confidentiality and integrity of outsourced data. Traditional encryption techniques are most frequently used to encrypt the data before storing in the cloud. The problem here is the searching of these data. Some keyword searching techniques are provided to find the relevant documents from a cloud for maintaining keyword and file privacy. In this paper propose a fuzzy keyword trigram based on searching with efficient tree based indexing. The fuzzy keyword n-grams improves its usability, and a hash table chaining tree-based indexing increases the performance of knowledge sharing. Here the preprocessing of a document is performed by natural language processing, which provides relevant keywords from it. Trigrams from the keywords where build with a hash table and chaining is constructed by a trigram tree which reduces the complexities in linear searching of files.**

***Index Terms*— Cloud,storage,searching,fuzzy keyword,trigram,encryption,hash function,indexing,chaining.**

I. INTRODUCTION

Cloud storage [1],now become the one of the most popular applications of cloud computing. So the companies and individuals use these service of cloud computing to outsource their private data to a large storage space provided by the cloud service providers. These data are controlled and managed by these providers. There are some issues and challenges in the storage of large documents sets into the cloud[2] to avoid the information leakage, the cloud data need to be encrypt before storing. There are some difficulty in searching this information. For accessing of encrypted data some keyword searching techniques are proposed. The plain text fuzzy keyword search techniques[3] which present a variety of indexing algorithms, such as su□x trees, metric trees, and q-gram methods. On the face of it, the direct use of these string matching algorithms to the context of searchable encryption by computing the trapdoors on a character base within an alphabet users from the dictionary and statistics attacks and fails to achieve the search privacy. The earliest methods [4] [5] also have some drawbacks. The first drawback is it allows only exact keyword

Grenze ID: 02.NCETICS.2019.12

searching and does not allow any misspelling of keywords. A wild card based fuzzy keyword set construction[6] which solves this problem. The post preprocessing cost and traffic problems during the searching process can be reduced by the storage of relevant search results. Single, as well as multykeyword ranked searching [7] is used for searching on cloud data files.To perform fuzzy keyword searching, we need to provide efficient indexing of items without any security loss.

This paper propose a binary search tree based indexing of encrypted data stored in the cloud.Encryption of searching trigrams and hash indexing with BST chaining provides more security. Here it constructs a trigram-based fuzzy keyword set, and the string similarity is quantified using Jaccard coefficient [5]. Here the performance can be increased by the elimination of fuzzy keyword with minimum threshold value. The advantages of using Tree based indexing and chaining fuzzy keyword set is that it reduces the searching time as compared to a linear searching method. The file preprocessing enumerate all relevant keywords of the document with its TF-IDF score value

II. Related Works

A.Keyword Indexed Searchable Encryption

The searchable encryption algorithms create an index for each keyword of the document, and the search operation was based on that keyword index. The first searchable encryption system was introduced by Boneh et al. [8]. It is public key encryption (PKE) scheme. The encrypted file with owner's public key can be searched and decrypt only by the authorized data users who know the owners private key. But the problem with this approach is that it takes too much time for public key calculation. Some research works [9]-[11] had developed to improve the security and efficiency of a single keyword searching. Song et al. [12] also suggest searchable encryption. Here the encrypted documents are scanned word by word. It is a sequential scanning without any specific indexing. Again a secure indexed architecture is proposed by Goh et al. [13], which hold a trapdoor corresponding to each keyword. Chang et al.[14] and Curtmola et al. [15] both propose a method for an efficient search by constructing a single encrypted hash table index, which contains trapdoor of a keyword and set of identifiers to the file corresponding to the keywords.

B. Fuzzy Keyword Indexed Searchable Encryption

Fuzzy keyword searchable encryption problem is first solved by Jin et al. [6]. Here the searching scheme deploy a Wildcard based Fuzzy Set Construction (WFSC) for each file. The fuzzy keyword set for each keyword can be reduced by a Dictionary-Based Fuzzy Set Construction (DFSC) which is mentioned in [16]. Later an efficient and privacy-preserving ranked fuzzy keyword searching is proposed in[17]. Here K-grams and Jaccard coefficient is calculated to construct a fuzzy keyword set and one to many order preserving mapping is build an inverted index. New index structure and the functionality of both fuzzy keyword searching and ranked keyword searching can be implemented with DFSC and OPM algorithm. Chauah et al [18] exploit a different approach, considering multiple keywords as a single keyword and build an index based on tree structure.

III. Problem Formulation

A. System model

The fuzzy keyword searching scheme consists of different components which perform the operations to make it fruitful to search the data from the cloud.The elements shown in figure 1

This model consist of various parts. The data owner uploads his file f into the cloud system. Thus the owner has a set of files $F=f_1,f_2,f_3\ldots$. A set of keywords $K=k_1,k_2,k_3\ldots$ were generated from that file f_i. Finally create a set of trigrams for each keyword k_i, i.e, $w_1w_2,w_3\ldots$. For efficient searching, it needs to create a tree based indices. Each trigrams are encrypted using an hash function method. For each trigram wi, generate a hash value $h(w_i)$, which is difficult to break. SHA algorithm can be used for this trapdoor encryption. Form the encrypted trigram generate a binary search tree. It consists an hash table with chaining of a binary tree associated with each item in the table. Here 512 bit hash block for tree generation. The primary index is represented using first 8 bits of the hash value.

In this paper it is assumes that authentication of the data user is done by the data owner and it is appropriately established. If a data user sends a search request for the specified file,the cloud system transmit this request to the data owner for decrypt the file. The data owner authenticate the user and sent its private key for the

ecryption of requested file. Fuzzy trigram based searching is used to identify the file. It provides security to the stored information.

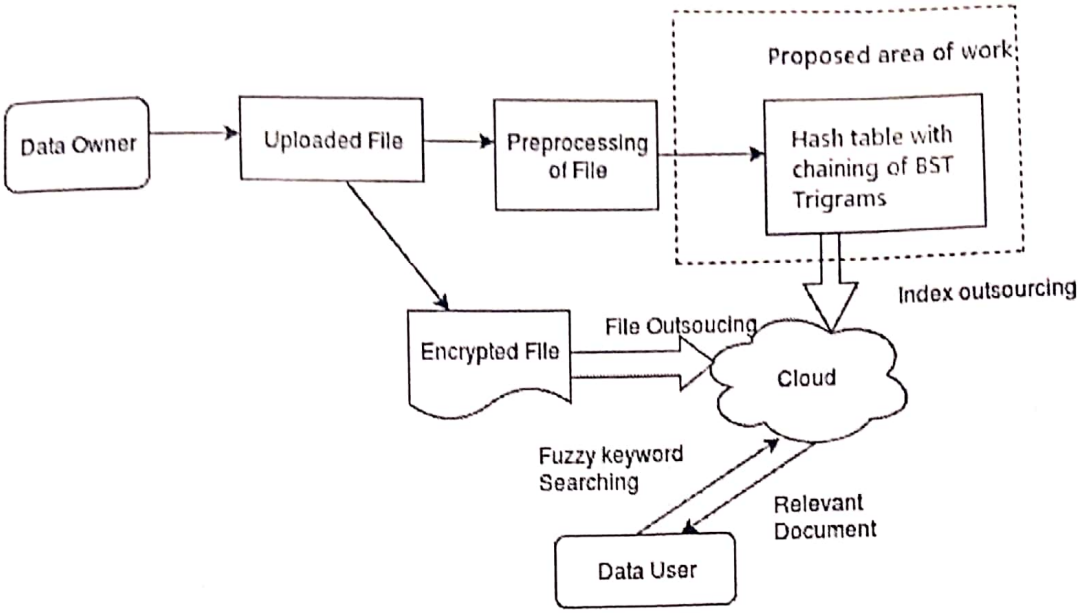

Figure. 1: A design framework for BST index based Fuzzy keyword searching

A. Threat model

The cloud server can't be trusted because from the search request using keyword the server can identify some user-related information's. In this paper, the trigrams are encrypted and will follow the security definition implemented in the traditional encryption [2]. We should prefer more security to the stored files and indices. Make sure that it doesn't leak any user information.

B.Design Goals

This paper defines the problem of efficient fuzzy keyword search on the cloud with different concerns including security, retrieval speed of files. In this definitions, we have the following goals.

- Derive trigram based fuzzy keyword searching over encrypted data in cloud computing
- Privacy preserving search means the encryption prevents the access of user information from the files, trapdoors, and indices by the cloud server.
- Efficient searching indices, a BST construction of encrypted trigram improves the retrieval speed of files

C.Preliminaries

*Fuzzy keyword search :*The fuzzy keyword searching in the existing system was used edit distance scheme which provides a probability of the occurrence of a keyword in the list of files. If there are n encrypted files F, $K=k_1,k_2,k_3\ldots$ are e_t of keyword the edit distance e is predefined. The searching input (K_i, e'), where K_i is the target keyword and threshold of fuzzy search is e'. Then the search output provides a set of file IDs that contain the keyword, K_i. It is represent as Fw_i.

Trigram of a keyword: Trigram is a substring of length 3. Trigram helps to find the highly cohesive features of a string. For example, clo, lou, oud are the trigrams of the keyword cloud. In this case, for a keyword K with length n we generate n-3+1 trigrams.

Binary Search Tree: The binary search tree or ordered tree is that,which stores items in each node. The tree allows fast accessing, addition and removal of items from the tree which can be implemented using lookup tables. The binary search operations are used to find a data from the tree and also satisfies the binary search tree property which is the value of left sub tree which is less than the value of right sub tree, i.e, For all nodes x and y, if y belongs to the left sub tree of x, then the at y is less than the key at x, and if y belongs to the right sub tree of x, then the key at y isgreater than the key at x. In addition to this each node, p points to the parent and, left and right pointers point to the left child and right child which is stored at the node respectively.

*Hash-table and Separate Chaining:*Hash-table is an efficient data structure for the storage and retrieval of data from the storage medium. Each object in a hash table is associated with a key. Due to the similarity in

the hash value indexing, there is a chance of collision, i.e, both keys are mapped to the same index. Hash-table with separate chaining should solve this problem. If it occurs hash table with chaining will insert a new node. The separate chaining implementation consists of each independent bucket, and has some nodes of entries with the same index.

IV. PROPOSED SYSTEM

The proposed system of new BST Index based Fuzzy keyword searching consists of different procedures.

A. File preprocessing system

Each user file is preprocessed before it stored into the cloud. Here the files uploaded to the cloud by the user is scanned and derive appropriate keywords associated with each one. Then calculate tf-idf score of each words. Across them choose keywords with highest tf-idf frequency as the appropriate keyword set.

B. Trigram generation and Trapdoor encryption

A trigram is a substring of a keyword with 3-grams. Taken each keyword from the keyword set of the given document and derive its trigram.

For improving the security of the trigram, it must be encrypted. Here the proposed method using hash function trapdoor reducing the chance of intrusion. The hash function converts the variable length output into a fixed length value. A more beneficial feature is that no two results has same hash value. There are some features associated with the hash functions.

Output hash value has fixed length: The hashing of data produces fixed length output, it also called a compression function in which the size of the outcome is less than the input data value. The n-bit hash function always produce an n-bit output.

Operational efficiency: Operation to compute an h(x) from an input trigram x is fast. It is also better than symmetric encryption methods.Properties of hash functions also improves the security features. If a hash function produces a hash value of trigram y, it is not easy to find out original trigram. This feature protects the trigram from an attacker. The Hash function is collision resistant, that is no two trigrams having the same hash value. The secure hash algorithm SHA-1 is used for the trapdoor encryption. In this algorithm after some initial processing of 512-bit blocks of the input text, which are further divided into 16 blocks. Each consists of 32-bit size. The output is 16-bit message digest of 5 32-bit blocks

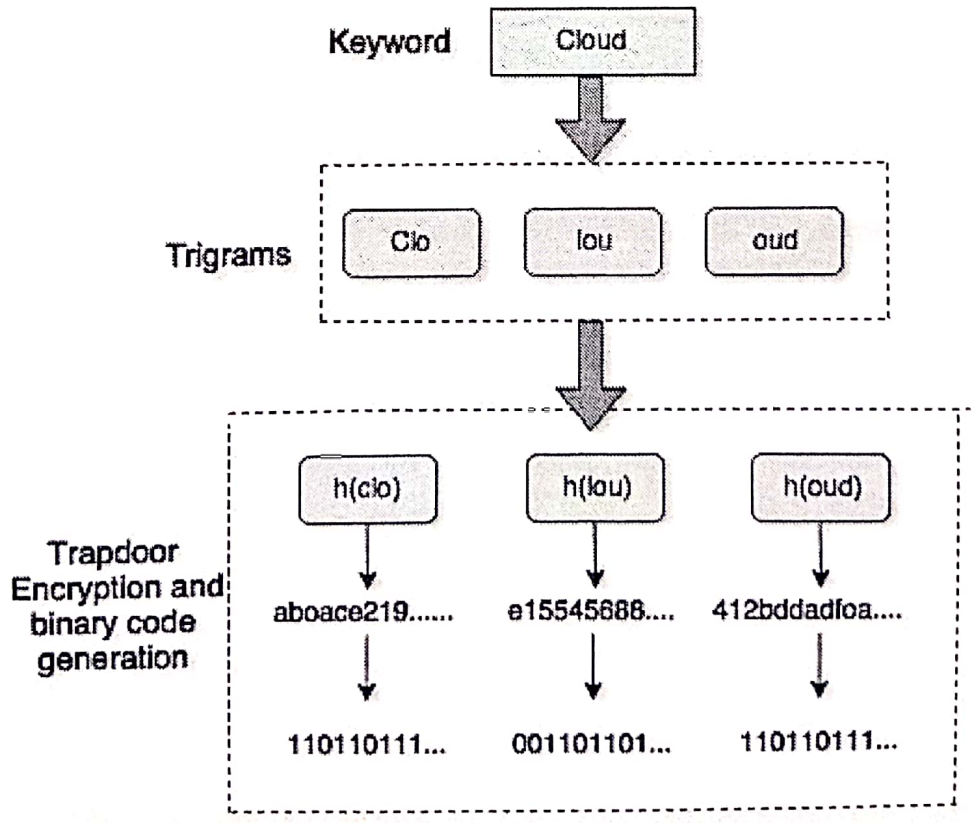

Figure. 2: SHA1 Hashing and Binary code generation

C. BST based indexing of trigrams

The user wants to access the encrypted data, and the searching request consists of a set of keywords to the specified document D_i. It is represented as $R=r_1,r_2,r_3....$For each keyword r_i, we calculate it's trigrams $T=t_1,t_2,t_3....$ Here we generate a fuzzy keyword set F_z. The indexing generates for the encrypted fuzzy keyword set. Trapdoor encryption of this set specifies in the previous session. Initially, create a linear list which stores the first 8-bits of the hash 512-bit hash value of trigrams. It starts from 00000000 to 11111111, which is a hash table having separate chaining with 2^8 head data cells. From these, each data cells follows chaining of nodes in the binary search tree. Following figure 2 shows the structure of the indexing tree. Each sub trees has nodes that stores the hash value starts from a root node. This tree also maintains the binary search tree property that is the value of left sub tree less than the value of right sub tree. This property is also maintains during addition and removal operations. The leaf node of each sub tree contains the document name in which the trigram includes. The proposed indexing structure helps fast accessing of documents that contain the searching request keywords.

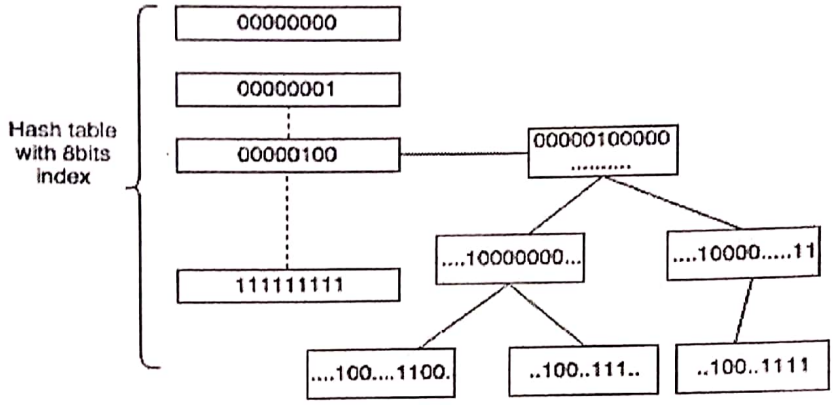

Figure. 3: Trapdoor based indexing tree structure design

*D.Searching of Documents*Here,it uses a key to search relevant document from the cloud storage. The key is a corresponding trigram of a file. The trapdoor hashing is performed on the given string and generate its binary code value. During the search process, it starts from the 8-bit hash table index. For example, first 8-bit of the encrypted search keyword is 00000011, then starts from this node and find the trees root node for tree traversal.The searching is over when we are at the key node. Otherwise use the BST property for searching, which can be implemented using the following algorithm.

Input : B, x, t_b

Output : D

a <- gets first 8 bits of t_b

If B[i]== a **then**

y -> B[i].next

While y!= nil

If key[y]== k then

return y

Else If key[y] < k then

y -> right[y]

Else

y -> left[y]

return("Not Found")

Algorithm 1 : BST TRigramSearch

V. Implementation And Analysis

The applications or research work under cloud computing requires large configurations and deployment requirements.The performance evaluation of such models are difficult to achieve.So in-order to overcome these problems ,we can use a clouSim simulator that enables modeling and simulation of cloud computing systems and applications.The cloud system stores no of files in the cloud.Before storing these files,the file preprocessing system extract keywords.Then generates the trigram and perform trapdoor based indexing. The proposed scheme required a linear searching and BST based searching. Here the linear searching time is comparatively smaller because of 8-bit key size. So the list only contains 2^8 items. The advantage of using binary searching is that it never meets collision. The binary search tree operations can be implemented in O(log(n)) time, which is faster than linear time. The space needed by tree is exactly same as its input data size. The other benefits of using BST over a hash table is that we do not need to know the input size in advance. So we can use any no of keywords which doesn't need a fixed set of keywords for storing the document in the cloud.

The data structure considered here is hashing with chaining to store data as a binary search tree b of lists. An integer n consists of total no of items in all BST lists. The binary hash value of the data item x can be represent as hash(x) in the range [0,...,(b.length-1)]. The average no of elements stored on the list is

$$\frac{n}{b.length} \leq 1$$

The searching over a hash table performs with linear searching. The simple uniform hashing take an assumption that each key is equally liked to be hashed to any slot of a table. The slot value can be represented using a variable m. Here,

$$loadfactor, \alpha = \frac{n}{m}$$

So the expected time for searching is $O(1+\alpha)$, which is same as the insertion and deletion. If

$\alpha = o(1)$,the time complexity becomes O(1). When the value of m increases and n is small ~2^8 the load factor approximately equal to n, so the time complexity will also reach O(n).

In the proposed approach,instead of a list for a chain,it is implemented using binary search trees.As the searching time for each element in a BST with almost n nodes is O(log(n)). So by using this indexing, searching time reduced to (1+log(n)).

VI. Conclusion

This paper proposes a design of efficient indexing to the remotely stored encrypted cloud data, which makes the searching faster. For security consideration and improvements in the retrieval speed this paper uses the advantages of both trigrams based fuzzy keyword searching and binary search tree based indexing. Trigram based fuzzy keyword searching allows flexibility in searching for the users. Jaccard coefficient is used to find the similarity of the keywords to identify the relevant documents that the user required. In the future work, we can improve the security features and access efficiency by using semantic-based searching with BST indexing. Some relevant criteria were implemented for ranking of relevant documents in the cloud

References

[1] Kamara S, Lauter K. Cryptographic cloud storage[C]//International Conference on Financial Cryptography and Data Security. Springer Berlin Heidelberg, 2010: 136-149.

[2] Singh A, Chatterjee K. Cloud security issues and challenges: A survey[J]. Journal of Network and Computer Applications, 2017, 79: 88-115.

[3] C. Li, J. Lu, and Y. Lu, Ecient merging and ltering algorithms for approximate string searches, in Proc. of ICDE08, 2008.

[4] Yan-Cheng Chang and Michael Mitzenmacher. Privacy preserving keyword searches on remote encrypted data. In Applied Cryptography and Network Security, volume 3531, pages 391421. Springer Berlin / Heidelberg, 2005
[5] Reza Curtmola, Juan Garay, Seny Kamara, and Rafail Ostrovsky. Searchable symmetric encryption: improved definitions and efficient constructions. In Proceedings of the 13th ACM conference on Computer and communications security, CCS 06, pages 7988, 2006.
[6] Jin Li, Qian Wang, Cong Wang, Ning Cao, Kui Ren, and Wenjing Lou. Fuzzy keyword search over encrypted data in cloud computing. In INFOCOM, 2010 Proceedings IEEE, pages 15, march 2010.
[7] Cong Wang, Ning Cao, Jin Li, Kui Ren, andWenjing Lou. Secure ranked keyword search over encrypted cloud data. In Distributed Computing Systems (ICDCS), 2010 IEEE 30th International Conference on, pages 253 262, june 2010.
[8] D. Boneh, G. D. Crescenzo, R. Ostrovsky and G. Per-siano, Public KeyEncryption with Keyword Search, Proceedings of EUROCRYPT, Interlaken, 2-6 May 2004.
[9] E.-J. Goh, Secure Indexes, Cryptology ePrint Archive, 2003. http://eprint.iacr.org/2003/216
[10] Y.-C. Chang and M. Mitzenmacher, Privacy Preserving Keyword Searches on Remote Encrypted Data, Lecture Notes in Computer Science, Applied Cryptography and Network Security, Vol. 3531, 2005, pp. 391-421.
[11] R. Curtmola, J. A. Garay, S. Kamara and R. Ostrovsky, Searchable Symmetric Encryption: Improved Definitions and Efficient Constructions, Proceedings of ACM CCS, Alexandria, 30 October-3 November 2006
[12] D. Song, D. Wagner, and A. Perrig, Practical techniques for searches on encrypted data, in Proc. of IEEE Symposium on Security and Privacy00, 2000.
[13] E.-J. Goh, Secure indexes, Cryptology ePrint Archive, Report 2003/216, 2003, http://eprint.iacr.org/.
[14] Y.-C. Chang and M. Mitzenmacher, Privacy preserving keyword searches on remote encrypted data, in Proc. of ACNS05, 2005.
[15] R. Curtmola, J. A. Garay, S. Kamara, and R. Ostrovsky, Search- able symmetric encryption: improved definitions and efficient constructions, in Proc. of ACM CCS06, 2006.
[16] Chang Liu, Liehuang Zhu, Longyijia Li, and Yuran Tan. Fuzzy keyword search on encrypted cloud storage data with small index. In Cloud Computing and Intelligence Systems (CCIS), 2011 IEEE International Conference on, pages 269 273, sept. 2011.
[17] Qunqun Xu, Hong Shen, Yingpeng Sang, and Hui Tian. Privacypreserving ranked fuzzy keyword search over encrypted cloud data. december 2013
[18] M. Chuah and W. Hu, Privacy-aware bedtree based solution for fuzzy multi-keyword search over encrypted data, in Proceedings of the 2011 31st International Conference on Distributed Computing Systems Workshops. IEEE, 2011, pp. 273281.