Multiuser Fuzzy Keyword Search over Encrypted Cloud Data

Ms N.A.Wasankar Dr.P.P.Karade Dr.V.M.Thakare Dr. R.V Dharaskar

Abstract **:- Cloud computing is a technology that uses the internet and central remote servers to maintain data and applications. Cloud computing allows consumers and businesses to use applications without installation and access their personal files at any computer with internet access. This technology allows for much more efficient computing by centralizing storage, memory, processing and bandwidth. Perhaps the biggest concerns about cloud computing are security and privacy. There are many searchable encryption technique which were implemented in the cloud, these technique supports only exact keyword search not similarity. This paper reviews efficient** Searchable symmetric encryption, **Wildcard –Based Technique, Gram – Based Technique, Symbol – Based Trie – traverse Search Scheme.After analyzing these techniques, has arisen some drawbacks, so we have proposed multiuser fuzzy keyword search over cloud computing approach. This proposed method, concentrates on solving the problems of the user who search the data with the help of fuzzy keyword on cloud and support multiuser environment.**

Keywords **: Wildcard, Gram – Based technique, Symbol – Based Trie – traverse Search Scheme, Conjunctive keyword.**

I. INTRODUCTION

Cloud storage is the next stage in the internet’s evolution providing the means through which everything-from computing power to computing infrastructure, applications, business processes to personal collaboration-can be delivered to users as a service wherever and whenever users need. Moreover, sensitive information are being centralized into the cloud, such as emails, personal health records, government documents, etc. By storing their data into the cloud, the data owners can be relieved from the burden of data storage and maintenance so as to enjoy the on-demand high quality data storage service. However, the threats to this technology are now more tangible than ever. The fact that data owners and cloud server are not in the same trusted domain may put the outsourced data at risk, as the cloud server may no longer be fully trusted. From privacy point of view, these data are encrypted before outsourcing which complicates file searching task. In cloud computing, many users are there to whom data owners may share their outsourced data. Often, users that perform searches have typos and format inconsistencies in their input string.Fuzzy search displayed the exact keywords along with similarity keywords when exact match fails, which solve the problems faced by the cloud users.Architecture forthe fuzzy keyword search shown in fig1 consists of three entities: data owner, user, and cloud server. Data owner may be either an individual or enterprise customer, who depend on cloud server for remote datastorage and maintenance.

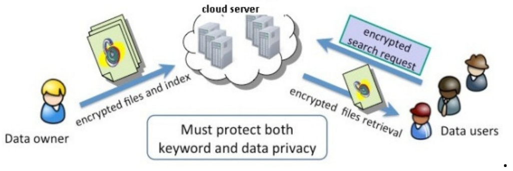

cloud server
encrypted search request
encrypted files and index
encrypted files retrieval
Must protect both keyword and data privacy
Data owner
Data users

Fig 1: Architecture of fuzzy keyword search

II. BACKGROUND

Fuzzy search, searches the underlying data as the user types in query keywords. It extends autocomplete interfaces by allowing keywords to appear in multiple attributes of the underlying data; and finding relevant records that have keywords matching query keywords approximately [1][5]. In [4] propose the B^{ed}-tree, based index structure for evaluating all types of similarity queries on edit distance and normalized edit distance. Privacy Preserving Keyword Searches on Remote Encrypted Data are efficient in the sense that no public-key cryptosystem is involved in [6]. Several algorithms that can greatly improve the performance of existing algorithms are developed in [7]. In [8] the problem of secure and efficient similarity search over outsourced cloud data is investigated. . In [10] general framework for constructing and analyzing public-key systems supporting queries on encrypted data has provided.

III. PREVIOUS WORK DONE

How to efficiently find in a collection of strings those similar to a given query string various similarity functions can be used, such as edit distance, Jaccard similarity, and cosine similarity. Several algorithms have been proposed based on the idea of merging inverted lists of grams generated from the strings. Many algorithms use gram-based inverted-list indexing structures to answer approximate string queries. These indexing structures are large compared to the size of their original string collection. To reduce the size of such an indexing structure to a given amount of space, while retaining efficient query processing. Two novel approaches for achieving the goal are: one is based on discarding gram lists, and one is based on combining correlated lists [1].

The B^{ed}-tree, a B^{+}-tree based index structure is used for evaluating all types of similarity queries on edit distance and normalized edit distance. Identify the necessary properties of a mapping from the string space to the integer space for supporting searching and pruning for these queries is identified. Three transformations are proposed that capture different aspects of information inherent in strings, enabling efficient pruning during the search process on the tree.

When user wants to store his files in an encrypted form on a remote file server. Later if the user wants to efficiently retrieve some of the encrypted files containing (or indexed by) specific keywords, keeping the keywords themselves secret and not jeopardizing the security of the remotely stored files. Solutions was proposed which is independent of the encryption method chosen for the remote files yet preserves the privacy[3].

This paper is organizes as follows. **Section I** Introduction. **Section II** discusses Background. **Section III** discusses Previous Work Done. **Section IV** discusses various methodologies. **Section** Discusses attributes and parameters and how these are affected on search. **Section VI** proposed Method and possible outcome result. Finally **section VII** Conclude this paper.

IV. Existing Methodology

Fuzzy keyword search enhances system usability by returning the matching files when users' searching inputs exactly match the predefined keywords or the closest possible matching files based on keyword similarity semantics, when exact match fails. Fuzzy keyword search begins with constructing the fuzzy keyword set. Methods for constructing fuzzy set are Editdistance, Wildcard- based technique, Gram – Based Technique. For example, the following is the listing variants after a substitution operation on the first character of keyword CASTLE: {AASTLE, BASTLE, DASTLE, YASTLE, ZASTLE} [1][2].

Wildcard is used to denote edit operations at the same position. The wildcard-based fuzzy set of w_i with edit distance d is denoted as $Sw_i,d=\{S'w_{i,0},S'w_{i,1}, \cdots , S'w_{i,d}\}$, where $S'w_i,\tau$denotes the set of words w'_i with τ wildcards.For example, for the keyword CASTLE with the pre-set edit distance 1, its wildcard based fuzzy keyword set can be constructed as SCASTLE, 1 = {CASTLE, CASTLE,*ASTLE, C*ASTLE, C*STLE, CASTL*E, CASTL*, CASTLE*}[1].Another efficient technique for constructing fuzzy set is based on grams. The gram of a string is a substring that can be used as a signature for efficient approximate search. In this any primitive edit operation will affect at most one specific character of the keyword, leaving all the remaining characters untouched. Alternatively it means, the relative order of the remaining characters after the primitive operations is always kept the same as it is before the operations. For example, the gram-based fuzzy set SCASTLE, 1 for keyword CASTLE can be constructed as{CASTLE, CSTLE, CATLE, CASLE, CASTE, CASTL, ASTLE} [2].Efficient fuzzy keyword search approach as compared to straight forward approach given in [1]. Consist of 1) Data owner first constructs a fuzzy keyword set Sw_i,dusing the wildcard based technique to build the index. Then trapdoor set $\{Tw_{i}'\}$ for each $w_i'\in Sw_i$,d with a secret key sk sharedbetween data owner and authorized users is computed by dataowner. He encrypts $FIDw_i$ as Enc(sk, $FIDw_i \| w_i$). The index table$\{(\{Tw_i'\}w_i'\in Sw_i$,d ,Enc(sk, $FIDw_i \| w_i))\}w_i\in W$ and encrypteddata files are outsourced to the cloud server for storage;2) When user want to search word w with edit distance k, the authorized user computes the trapdoor set $\{Tw'\}w'\in S_{w,k}$, where $S_{w,k}$is also derived from the wildcard-based fuzzy set construction. He thensends $\{Tw'\}w'\in S_{w,k}$ to the server;3) After receiving the search request $\{Tw'\}w'\in S_{w,k}$, theserver compares them with the index table and returns allthe possible encrypted file identifiers {Enc(sk, $FIDw_i \| w_i$)}.The user decrypts the returned results and retrieves relevantfiles of interest [1].

For improving the search efficiency, one way is to use a symbol-based trie-traverse searching mechanism.In symbol-based trie-traverse searching mechanism a multi-way tree can be constructed. In this each trapdoor is divided as a collection of concatenated symbols, which are all represented as ɵ-bit binary vector. In this multi-way tree, all trapdoors sharing a common prefix have a common symbol node. The root is associated with an empty set,and the symbols in a trapdoor can be recovered in a searchfrom the root to the leaf that ends a trapdoor. [2].

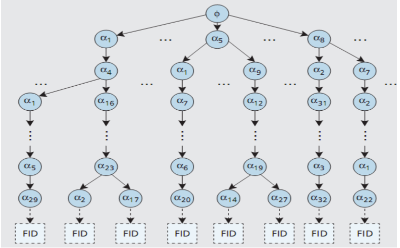
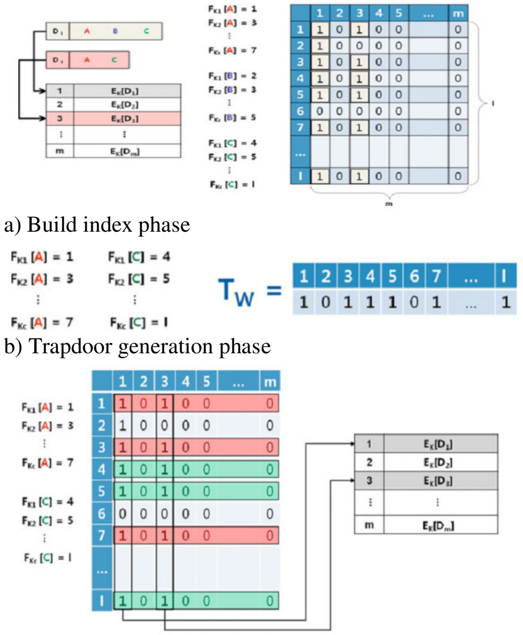


Fig2: An example of integrated symbol-based index for all words in thefuzzy keyword set.

Efficientsearchable encryption system that support exact keyword search allows userstosearch data by multiple keywords. It consist steps for building index, trapdoor generation and data search shown in fig.3 [3].



	1	2	3	4	5	...	m
1	1	0	1	0	0		0
2	1	0	0	0	0		0
3	1	0	1	0	0		0
4	1	0	1	0	0		0
5	1	0	1	0	0		0
6	0	0	0	0	0		0
7	1	0	1	0	0		0
...							
l	1	0	1	0	0		0

a) Build index phase

1	2	3	4	5	6	7	...	l
1	0	1	1	1	0	1	...	1

b) Trapdoor generation phase

	1	2	3	4	5	...	m
1	1	0	1	0	0		0
2	1	0	0	0	0		0
3	1	0	1	0	0		0
4	1	0	1	0	0		0
5	1	0	1	0	0		0
6	0	0	0	0	0		0
7	1	0	1	0	0		0
...							
l	1	0	1	0	0		0

c) Data search phase

Fig3: Efficient searchable encryption system

V. Analysis and Discussion

Characteristics	symbol-based trie-traverse Search Scheme	Efficient searchable encryption system
Fuzzy keyword approach	Yes	No
Multi-keyword support	No	Yes

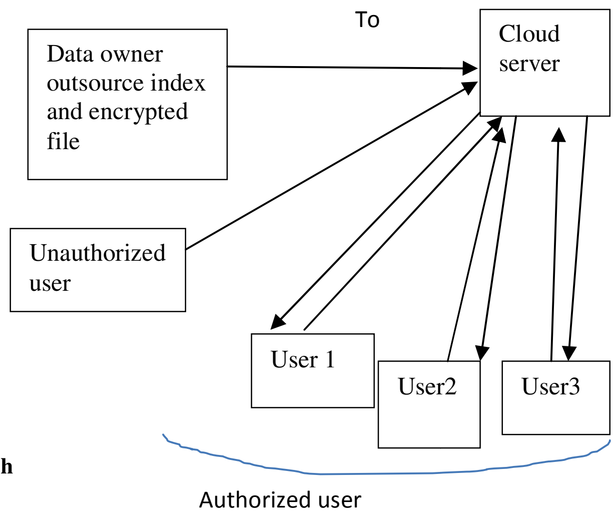
Table 1: Comparison between Symbol-based trie traverse search schemes, Efficient searchable encryption system

Fuzzy keyword set Techniques	Characteristics
Edit Distance	Storage cost for index is very large. Efficiency in searching fuzzy keyword is low
Wildcard- based technique	Efficiency in searching fuzzy keyword is medium.
Gram-based technique	1) Efficiency in searching fuzzy keyword is high. 2) It is less secure than wildcard technique

VI. Proposed Methodology

The Edit distance, Wildcard-based, gram-based and symbol-based trei traverse technique support single fuzzy keyword search. Efficient searchable encryption system support exacts multiple keyword search but not fuzzy keyword search. Also above mentioned techniques support single user environment. So there is need to support multiuser environment. We propose multiuser fuzzy keyword search over cloud computing approach. The framework consist of

1) Building the index using the fuzzy keyword set Sw_i,d that constructed using wildcard technique.
2) $sk_0 \leftarrow Gen(1^{sk})$: used by owner. It takes as input a security parameter sk, and output an owner secret key sk_0.
3) $\{Tw'\} \leftarrow Trpdr(sk_u,w,k)$: computed by user to generate a trapdoor set for a keyword. It takes as input a user U’s secret key sk_u, a keyword w, and edit distance k ($k \leq d$) and outputs a trapdoor set {Tw´}.
4) x←search(sts,I,t):is used by the server S to perform a search. It takes as input a server state sts, an index I and trapdoor t.
5) $sku \leftarrow Add(sk_0,st_0,U)$: use by the owner to add a user. It takes as input the owner’s secret key sk_0 and state st_0 and a unique user id U and outputs U’s secret key sku.



↑ Upward arrow represent search request

↓ Downward arrow represents file retrieval

Outcome Result Possible

Suggested method provides security as it uses wildcard based technique. This is more secure than gram based technique. And allow fuzzy keyword search in multiuser environment.

VII. Conclusion

This paper exhausted different searching techniques used in remotely stored data. The wild card method and gram method to construct fuzzy keyword set and edit distance to quantify keywords similarity are explained in this paper , it allow the user to search data even if exact search fails. At the end, multiuser fuzzy keyword search over encrypted cloud is provided that support searching when there is typos, format inconsistencies and multiple user.

Future Scope

As an ongoing work, there is a need to provide mechanismthat support search ranking that sorts the searching results according to relevance criteria and multi-keyword search.

References

[1] Jin Li, QianWang , Cong Wang, Ning Cao, KuiRen and Wenjing Lou. “Fuzzy Keyword Search overEncryptedDatain Cloud Computing”,INFOCOM, 2010 proceedings IEEE, VOL. 10.1109/INFCOM.2010.5462196, PP.1-5, March2010.

[2] KuiRen, Cong Wang and Qian Wang. “Toward Secure and Effective Data Utilization in Public Cloud”, IEEE Network, VOL. 26, No. 6, PP 69-74, November/ December 2012.

[3] Sun-Ho Lee and Im-Yeong Lee. “Effective Searchable Symmetric Encryption System Using Conjunctive Keyword”, Information Technology Convergence, VOL. 253, PP. 477-483, 2013.

[4] Z. Zhang Bedtree: “an All-Purpose Index Structure for string Similarity Search based on Edit distance,”SIGMOD, PP.915-26, 2010.

[5]S.Ji, G.Li, C.Li, “Efficient interactive fuzzy keyword search,” in Proc.Of WWW’09, 2009.

[6]Chang YC, Mitzenmacher M, “Privacy Preserving Keyword Searches on Remote Encrypted Data,” Applied cryptography and network security conference-ACNS, LNCS, vol.3531, PP. 442–255.
[7] C. Li, J. Lu, and Y. Lu, “Efficient merging and filtering algorithms for approximate string searches,” ICDE’08, 2008.
[8] C. Wang et al., “Achieving usaBle and Privacy-Assured SimilaritySearch Over Outsourced Cloud Data,” INFOCOM, 2012, PP.451–59.
[9]R. Curtmola, J. A. Garay, S. Kamara, and R. Ostrovsky, “Searchable symmetric encryption: improved definitions and efficient constructions,” ACM CCS’06, 2006.
[10]Cong Wang, “Enabling Secure and Efficient Ranked KeywordSearch over Outsourced Cloud Data,” IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, vol.23,No.8, 2007, August 2012