International Journal of Scientific Engineering and Applied Science (IJSEAS) - Volume-1, Issue-8,November 2015
ISSN: 2395-3470
www.ijseas.com

RANKED SEARCH ENABLED FUZZY AND SYNONYM QUERY OVER ENCRYPTED DOCUMENT IN CLOUD

N.Jayashri [1], T.Chakravarthy [2]

[1] *Research Scholar. Dept. of Computer Science. AVVM Sri Pushpam College. Thanjavur, India.*

[2] *Asso. Professor. Dept. of Computer Science. AVVM Sri Pushpam College. Thanjavur, India.*

Abstract

Cloud is nothing but an extension of current Internet. Now-a-days there is no difference between organizations and individuals, both of them produce more and more documents because of smart devices. Outsourcing is the easiest way to maintain all those documents. To preserve document security, important data should be encrypted by the data owner before place it on an external server, which makes the conventional and effective plaintext keyword search methods useless. Earlier searchable encryption techniques encourage exact or fuzzy keyword search, not support semantic search. But in reality, it is pretty common that cloud users' searching input might be the synonyms of the predefined keywords and it also contain minor typos. In this paper, we recommend an efficient method to incorporate fuzzy keyword search with synonym-based ranked query on encrypted cloud documents. Fuzzy keyword search significantly improves system usability by retrieving the matching files when users' query keyword exactly match the existing keywords or the nearby feasible matching files on the basis of keyword similarity semantics, when *exact* match fails. Synonym-based search admitting synonym query and ranked search for achieving more accurate search result. Extensive experimental results shows that our proposed solution is capable to handle users both minor typos and synonym queries in an effective manner.

Keywords— *Cloud, Document Outsourcing, Fuzzy Keyword, Searchable Encryption, Ranked Search, Synonym Extension.*

1.Introduction

As cloud computing grows to be popular, common people and organizations are interested on it [1]. More and more important details are being outsourced into the cloud, such as personal details, personal health records, government documents, and emails, etc. By outsourcing their documents into the cloud, the owners can be relaxed from the trouble of document storage and maintenance so as to savour the on-demand quality data storage service. Though, the truth that owners and cloud server are not in the same reliable domain may put the outsourced document at risk, as the cloud server could no longer be completely trusted [2]. It results that important document generally have to be encrypted before outsourcing for document privacy and avoiding unwanted accesses. Yet, document encryption makes efficient utilization of documents a very difficult task in particular that there could be a large amount of outsourced documents. In addition, in Cloud Computing, owners may share their outsourced document with a huge number of users [3]. The individual users might want to access some particular documents they are concerned in during an allotted session. One of the most standard methods is to exclusively access documents by means of keyword-based search instead of getting back all the encrypted documents which is absolutely unfeasible in cloud environment. Such keyword-based search method permits users to exclusively access documents of interest and has been usually implemented in plaintext search environment, for an example Google search [4].

Unpredictably, encryption control user's capacity to execute keyword search and thus creates the conventional plaintext search techniques inappropriate for Cloud environment. Moreover, encryption schemes also call for the security of keyword confidentiality since keywords typically include essential details relevant to the document. Even though encryption of keywords can secure keyword confidentiality, it additionally provides the conventional plaintext search methods unsuitable in this condition. To secretly search on encrypted document, searchable encryption methodologies have been introduced in recent years [5]–[8]. Searchable encryption techniques generally create an index for each and every keyword of significance and link the index with the documents that contain the keyword. By incorporating the trapdoors of keywords inside the index information, efficient keyword search can be recognized while both document substance and keyword confidentiality are well-secured. Even though authorizing for executing searches confidentially and efficiently, the earlier searchable encryption methods do not match for cloud computing environment while they perform only exact keyword match.

However, these earlier search techniques cannot perform synonym-based keyword ranked search. In the

actual search environment, it is somewhat regular that cloud users' searching input could be the synonyms of the predefined keywords, or fuzzy matching keywords not the exact one due to the possible synonym substitution, such as commodity and goods, and/or with possible typos, such as Tamilnadu and Thamilnadu, and/or her lack of exact knowledge about the data. The earlier searchable encryption techniques allow only exact or fuzzy keyword search. That is, there is no acceptance of synonym replacement, syntactic deviation which, but regular user searching activities and occurs very often. Hence, apart from the exact match similarity-based ranked search on encrypted cloud document remains a very difficult issue. To encounter the challenge of similarity-based search, in this paper, we suggest a reasonably effective and flexible searchable encrypted technique which performs both synonym and fuzzy keyword ranked search, to handle both minor typos and synonym keywords.

To improve result accuracy, documents and related keywords are ordered based on the relevance score, which is computed by the ranking scheme. Here we use modified page rank algorithms as document ranking algorithm combined with TFxIDF algorithm for ranking.

Our contribution are summarized as follows:

i. We combine the mechanism of fuzzy and synonym keyword search, to address both minor typos and synonym replacement of existing keywords.
ii. Our proposed method also incorporate new document ranking algorithm with existing TF_IDF algorithm to get accuracy in relevance score calculation.

The rest of the paper is ordered as follows: Section 2 List, synonym search in plaintext and some of the Searchable Encryption techniques. Problem statement explained in Section 3. In Section 4, proposed work is explained in detail. Security analysis is discussed in Section 5. Section 6 presents a performance analysis of our proposed work. Finally Section 7 gives the conclusion of the whole work done in this paper.

2. Related Work

2.1 Plaintext synonym keyword search

Semantic similarity is a main area which discovers immense value in numerous discipline such as natural language processing (NLP), cognitive science and psychology, both in the research area as well as in business. Meticulous significance of semantic similarity between two words is important for various assignments such as, document clustering [10], information retrieval, and synonym extraction [11], etc. The distance based approach is a more common and direct approach for measuring semantic similarity between words using taxonomy.[12]. Rada et al. [13] implemented the distance method to a medicinal field, and discover that the distance function replicated fine in human evaluations of theoretical distance. Though, Richardson and Smeaton [14] had worries that the calculation was less precise than assumed result when implemented to a moderately large area. Resnik [15] spotted the node based method to establish the theoretical similarity is known as information content based technique. Lin [11] measures semantic similarity via an equation obtained from information theory. Jiang and Conrath [16] represented a method for calculating semantic similarity between words. Sahami and Heilman [17] determine semantic similarity between two queries by means of snippets retrieved for those queries from a search engine. Cilibrasi and Vitanyi [16] initiated a distance measure between words using only page counts returned by a web search engine. Bollegala, Matsuo and Ishizuka [19] produced an automated approach to calculate the semantic similarity between words or entities with the help of web search engines.

2.2 Searchable Encryption in cloud

To implement the searchable encryption on cloud computing environment, several researchers have been examining more on how to search over encrypted cloud documents effectively. Li et al. [20] initially presented a fuzzy keyword search method on encrypted cloud document, to address issues of minor typos and format inconsistency. Wang et al. [21] delivered a secure ranked search method, in which the cloud server able to rank relevant document with no understanding of particular keyword weigh. But this method supports only single keyword search. Then Cao et al. [22] explained a privacy preserving ranked method allowing multi-keyword, which uses vector space model. Chai et al. [23] presented a provable symmetric search encryption method, which can verify the accuracy and comprehensiveness of result. Sun et al. [24] also presented a secure multi keyword ranked search method based on vector space model (VSM). Private matching [25], as another associated perception, has been examined vastly in the perspective of secure shared calculation to allow various users calculate some function of their own document jointly without exposing their contents to the others. These functions might be intersection or rough private matching of two sets, etc. The private information retrieval [26] is frequently used method to access the matching entries in secret, which has been broadly implemented in information retrieval from

database and generally suffers unexpected computation difficulty.

3. Problem Statement

3.1 System Model

In this paper, we considered a system model contain three entities: data owner, cloud server and data user. Given a collection of *m* encrypted documents $C = (D_1, D_2, \ldots, D_M)$ stored in the cloud server, a set of keywords $K = \{k_1, k_2, \ldots, k_p\}$, the cloud server offers the searching facility for the approved users over the encrypted document *C*. The encrypted document collection *C* and searchable index *I* will be placed to the cloud by the owner of the document. We imagine the authorization between the owner and users is achieved properly. An approved user gives a request to retrieve particular documents of his/her interest. The cloud server is take charge to map the search request to a set of documents, where each document is indexed by a document ID and correlated to a set of keywords. In the search phase, the scheme will produce an encrypted search trapdoor depending on the keywords or the synonyms of the predefined keywords entered by the user with or without typos. The fuzzy and synonym keyword search technique retrieve the result documents according to the following rules: i) if the user's keyword of interest is perfectly matches the keyword in index I, the server is normally return the documents related to the keyword; ii) if there is any typos and/or format contradiction in the searching keyword, the server will send the nearby documents depending on pre-specified similarity semantics. iii) if the users searching keyword semantically related to the keyword in index I, but not exactly same, in that case server send documents which are having the keywords semantically equivalent to the given synonym. The search result is a set of encrypted documents and they are well ranked by our similarity measures.

3.2 Threat Model

We think about a semi-trusted cloud server. Although documents are encrypted, the cloud server may attempt to obtain some other important particulars from users' requests log. Thus, the search must be accomplished in a secure mode that permits documents to be retrieved while exposing as modest information as possible to the cloud server. In this paper, when proposing fuzzy and synonym keyword search method, we will keep the security description managed in the traditional searchable encryption [31]. More importantly, it is mandatory that nothing have to be disclosed from the outsourced documents and index apart from the results and the pattern of search requests.

3.3 Design Goals

i) To create fuzzy keyword set to tolerate minor typos and format inconsistency in user input.
ii) To create keyword set enhanced by synonym to perform synonym query.
iii) The search products can be accomplished when allowed users input the synonyms of the existing keywords or with minor typos not exact one.

3.4 Notation

$D_1, D_2, \ldots, D_M$ - Document set
C - $(D_1, D_2, \ldots, D_M)$ - Encrypted Documents
$K = \{k_1, k_2, \ldots, kn\}$ – Keyword set
d - distance
DID - Document ID
DID_k - word *k* in Document
Q - the searched keyword
df,t - the TF of term t in Document D_f
d_t - the number of files that contain term t
N - the total number of documents in the collection
|D*f*| - the length of document D*f*,
Sk_i,d - fuzzy keyword set
ks - secret key
y - encryption key

3.4 Preliminaries

Edit Distance: There are numerous techniques to quantitatively determine the string similarity. In this paper, we choose the well-studied edit distance [27] for our work. The edit distance **ed**(*k*1, *k*2) between two words *k*1 and *k*2 is the number of procedures involved to convert one of them into the another. The three primary procedures are i) Substitution: altering one character to another in a word; ii) Deletion: removing any one character from a word; ii) Insertion: adding a single character into a word. Given a keyword *k*, we let *Sk,d* denote the set of words *k'* fulfilling **ed**(*k, k'*) *d* for a some integer *d*.

Fuzzy Keyword Search: With the help of edit distance, the characterization of fuzzy keyword search can be designed as follows: Given a collection of *m* encrypted documents $C = (D_1, D_2, \ldots, D_M)$ placed in the cloud server, a group of distinct keywords $K = \{k_1, k_2, \ldots, kn\}$ with fixed edit distance *d*, and a searching input (*k*, *b*) with edit distance *b* ($b \leq d$), the implementation of fuzzy keyword search produces a group of document IDs whose corresponding documents possibly include the word *k*, denoted as DID_k: if $k = k_i \in K$, then return DID_{ki} ; otherwise, if $k' \in K$, then return {DIDki }, where $ed(k,k_i) \leq b$. Note that the above explanation is based on the supposition that $b \leq d$. In fact, *d* can be vary for different keywords and the scheme will return

$\{DID_{ki}\}$ fulfilling $ed(k,k_i) \leq \min\{b, d\}$ if exact match fails.

Synonym extension: Synonyms are words with the similar or same sense. In order to increase the precision of search results, the keywords obtained from outsourced documents require to be enhanced by regular synonyms, as cloud users' searching keyword might be the synonyms of the existing keywords, not the exact one because of the probable synonym substitution and/or her inefficiency of exact knowledge about the data. The synonyms of extracted keywords vary greatly from actual keywords in spelling. For example, the synonym of the keyword "technology" is "machinery" or "knowledge", these keywords are completely different in spelling. So we construct a general synonym thesaurus on the basis of the New American Roget's College Thesaurus (NARCT) [28]. Then the keyword index is enhanced by using our synonym thesaurus.

Rank function: In information retrieval, a ranking function is generally used to calculate relevant scores of matching documents to a request. Among lots of ranking functions, the "*TF×IDF*" algorithm [9] is broadly used, where *TF* (term frequency) measures the number of times a term exists in the document, and *IDF* (inverse document frequency) is normally measured by dividing the overall documents count by the count of documents related to the given keyword. From numerous verities of the TF IDF algorithm, not even a single permutation of them outperforms any of the existing schemes generally [29]. In this work, we prefer an example formula that is widely used and commonly seen in the literature (see [9, Ch. 4]) for the relevance score computation in the following arrangement. Its description is as follows:

$$Score\ (Q,F_d)=\sum_{t\in Q}\frac{1}{|D_f|}\cdot(1+\ln d_{f,t})\cdot ln\left(1+\frac{N}{d_t}\right) - \quad (1)$$

Here, Q denotes the searched keywords, *df,t* denotes the TF of term t in Document D_f; d_t denotes the number of files that contain term t; N denotes the total number of documents in the collection; and |D*f*| is the length of document D*f*, obtained by counting the number of indexed terms, functioning as the normalization factor.

Inverted Index: In IR community, inverted index is a extensively employed indexing form that contains a list of mappings from keywords to the consequent set of document that include this keyword, permitting full text search [29]. For ranked search scenario, the task of concluding which documents are most significant is usually done by assigning a numerical score, which can be computed advance, to each document based on ranking function. An inverted index is shown in Figure 1. We will use this inverted index arrangement to give our ranked fuzzy and synonym query supporting construction.

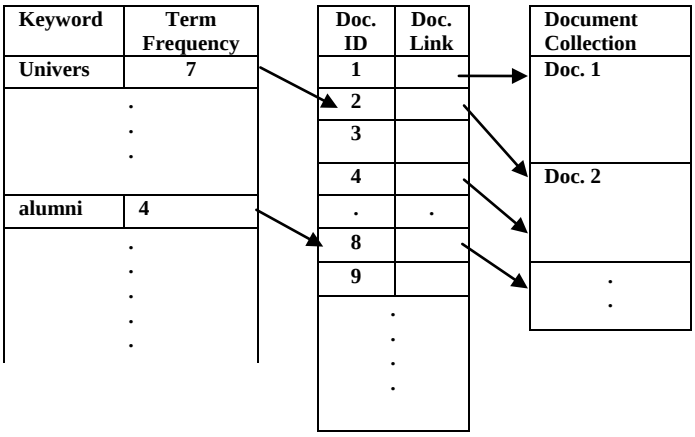

Fig. 1. Inverted Index

4. Proposed Work

4.1 Construction of Fuzzy Keyword Set:

We present an advanced method to constructing the fuzzy keyword set. Without loss of simplification, we will concentrate on the case of edit distance $d = 1$ to enhanced the proposed technique. Note that the method is sensitively constructed in such a manner that while restrain the fuzzy keyword set, it will not change the search correctness

Gram-Based Technique: Efficient technique for building fuzzy keyword set. It acts based on the grams. Gram of a string can be believed as a substring. Edit operation will change one character and left are untouched. In this, gram is applied for building inverted list for matching function. Here, edit operations will change only one character in the specified keyword and all other characters are same. For example, the gram based fuzzy set ASYSTEM, 1 for the keyword SYSTEM can be constructed a {SYSTEM, YSTEM, SSTEM, SYTEM, SETMS}. The total number of variants on SYSTEM constructed in this way is only 13 + 1, instead of 13 × 26 + 1 as in the above exhaustive enumeration approach when the edit distance is set to be 1. Generally, for a given keyword *ki* with length *l*, the size of Sk_i,1 will be only $2l + 1 + 1$, as compared to $(2l + 1) \times 26 + 1$ obtained in the straightforward approach. The larger the pre-set edit distance, the more storage overhead can be reduced: with the same setting of the example in the straightforward approach, the proposed technique can help reduce the storage of the index from 30GB to approximately 40MB. In case the edit distance is set to

be 2 and 3, the size of Sk_i,2 and Sk_i,3 will be $C_{l+1}^1 + C_l^1.C_l^1 + 2C_{l+2}^2$ and $C_l^1 + C_l^3 + 2C_l^2 + 2C_l^2.C_l^1$. In other words, the number is only $O(l^d)$ for the keyword with length l and edit distance d.

4.2 Construction of Keyword Set Extended by Synonym

In order to search the specific data instead of others efficiently, keywords want to be separated firstly from cloud document before place it on the cloud server. Here we expalin an enhanced text feature weighting scheme that include a new weighting factor to replicate the distinguishability of the keyterm on the base of the original TFIDF method [27]. Let N be the total number of documents in corpus, let n be the number of documents including the term i in corpus, let E_1 be the number of documents in the largest group including the term i, let E_2 be the number of documents in the second largest group including the term i. The new weighting factor C_d is appended to the formula of TFIDF, the enhanced formula is as follows:

$$W'_{ik} = TF \times IDF \times C_d = TF \times \frac{1}{DF} \times C_d$$
$$= f_{ik} \times \log \frac{N}{n_k} \times \frac{E_1 - E_2}{n} \quad \text{---------- (2)}$$

So the keywords are derived from every outsourced document by using our enhanced scheme. All keywords are derived from the same one text type one keyword subset, and all subsets form the keyword set finally. All the outsourced documents can be denoted as follows:

Doc 1: $k_{d_1}^1, k_{d_1}^2, \ldots\ldots, k_{d_1}^{n-1}, k_{d_1}^n$

Doc 2: $k_{d_2}^1, k_{d_2}^2, \ldots\ldots, k_{d_2}^{n-1}, k_{d_2}^n$

In order to accomplish a improved synonym-based search methodology for outsourced document, the keyword set want to be expanded by common synonym. Firstly, we create a general synonym thesaurus on the basis of the New American Roget's College Thesaurus (NARCT) [28]. NARCT is reduced in quantity by us because of the following two values: (i) choosing the common words; (2) choosing the words which can be semantically replaced completely. The built synonym set contains a total of 6953 synonym categories after the elimination. Secondly, the keyword set is enhanced by our constructed synonym thesaurus. The new keyword set including synonym is shown as follows:

Doc 1: $k_{d_1}^1 \; or \; s_1, k_{d_1}^2 \; or \; s_2, \ldots\ldots, k_{d_1}^{n-1} \; or \; s_{n-1}, k_{d_1}^n \; or \; s_{n_i}$

Doc 2: $k_{d_2}^1 \; or \; s_1, k_{d_2}^2 \; or \; s_2, \ldots\ldots, k_{d_2}^{n-1} \; or \; s_{n-1}, k_{d_2}^n \; or \; s_{n_i}$

Where s_1 depicts the synonym of k^1_{d1} . If a keyword has more than one synonyms, then all synonyms are appended into the keyword set. The duplicate keywords are removed to decrease the load of storage. Finally, a simplified keyword set and equivalent keyword scoring table are created.

4.3 Ranked Search Scheme

To formulate successful search methods based on fuzzy and synonym extension, we execute a four step process: **Setup, FSIndex**, **FSQuery**, **Search**. Based on the storage-efficient fuzzy and synonym keyword sets, we demonstrate how to build an effective fuzzy and synonym keyword search method. The method of the fuzzy keyword search goes as follows:

i) To build an index for *ki* with edit distance *d*,

a. the data owner first constructs a fuzzy keyword set Sk_i,d using the gram based technique.

b. Next fuzzy keywords are compared with dictionary and synonym set extended for meaningful word

Finally the trapdoor set $\{Tk_i'\}$ for each $ki \in Ski,d$ with a secret key *ks* shared between data owner and authorized users. The data owner encrypts DID*ki* as Enc(*ks*, DID*ki* $||k_i$). The index table $\{(\{Tk_i'\}k_{i'} \in Sk_i,d$, Enc(*ks*,DID$k_i||ki$))$\}ki \in K$ and encrypted data files are outsourced to the cloud server for storage;

ii) To search with (*k*, *y*), the authorized user computes the trapdoor set $\{Tk'\}k' \in S_{k,y}$, where *Sk*,*y* is also derived from the fuzzy and synonym set . Then sends $\{Tk'\}k' \in Sk,y$ to the server;

iii) Upon receiving the search request $\{Tk'\}k' \in Sk,y$, the server compares them with the index table and returns all the possible encrypted file identifiers {Enc(*ks*, DID*ki*||*ki*)} according to the fuzzy and synonym keyword definition.

The user decrypts the returned results and retrieves relevant files of interest. In this construction, the technique of constructing search request for *w* is the same as the construction of index for a keyword. As a result, the search request is a trapdoor set based on *Sw*,*k*, instead of a single trapdoor as in the straightforward approach. In this way, the searching result correctness can be ensured.

5. Security Analysis

We assess the protection of the scheme explained in the previous section by validating its assurance of the

security promise. That is, the cloud server must not discover the plaintext of whichever the documents or the searched inputs. We initiate from the security analysis of relevance score. Then, we examine the security intensity of the combination of fuzzy and synonym keyword set. Hence, the server can know the normalized Term Frequency distributions of a few important keywords, which are keyword precise correspondingly. With the value range and slope of these distributions, the server can distinguish the equivalent keywords. Suppose that the customer is only concerned in one keyword k , that is to say, only keyword k appears in the query. In this case the normalized TF distribution of the keyword is exposed directly. In the gram based scheme, the calculation of index and request of the same keyterm is equal. So, we simply require to confirm the index privacy by implementing reduction. Suppose the searchable encryption technique fails to accomplish the index privacy against the indistinguishability in the chosen keyword attack, which signifies there is an algorithm who can obtain the background information of keyterm from the index.

6. Performance Ananlysis

We evaluate the overall execution of our proposed methods by implementing the secure search system by C# language. The document set is constructed from the real data set: Reuters News stories [31]. This dataset is a collection of 18, 821 newsgroup documents including 11, 293 train documents and 7, 528 test documents. Using our improved E-TFIDF method presented in section 3.5, keywords are extracted from the Reuters News stories of 18, 821 newsgroup documents. Final number of individual keywords in keyword set is 46,153 with the normal word length 5.63 after eliminating the repeated keywords. Comparison of Index Construction time of existing and proposed work if shown in **Figure**. 2.

The search process, which is executed by the cloud server, is organized by calculating the similarity scores of related documents and result ranking depending on these scores.

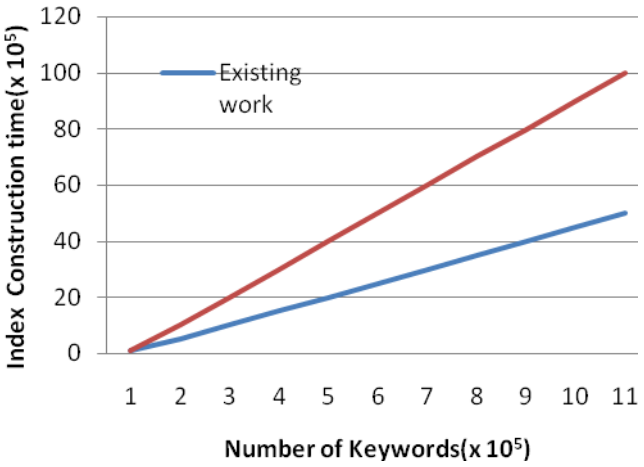

Fig. 2. Comparison of Index Construction Time

Fig. 2 shows the precision for the existing works and proposed one. From Fig. 3, we can recognize that the precision is primarily depends on the number of keywords in the index. Basic Ranked and Simple Fuzzy Keyword search results nearly the same count of documents whereas the fusion of fuzzy and synonym search increase the Precision rate.

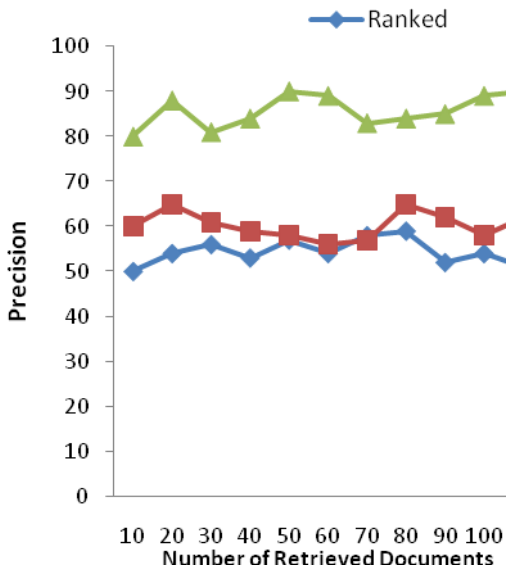

Fig 3. Comparison of Precision Values

7. CONCLUSION

In this paper, we propose and efficient methodology to solve the issue of both synonym and fuzzy based ranked search on encrypted cloud documents. We contribute mainly in two portions: synonym and fuzzy based search and similarity ranked search. We create an improved method, which is gram-based method to build the fuzzy keyword sets by exploiting a significant observation on the similarity metric of edit distance. And then Synonym set is also constructed

from thesaurus and document keywords. Depending on the constructed keyword sets, we can able to achieve search result when authorized users input may contain minor typos or it is synonym instead of exact one. Through rigorous security analysis, we show that our proposed solution is secure and privacy-preserving, while correctly realizing the goal of fuzzy keyword search. As our ongoing work, we will continue to research on security mechanisms that support: a) search semantics that takes into consideration conjunction of keywords, sequence of keywords, and even the complex natural language semantics to produce highly relevant search results; and a) search ranking that sorts the searching results according to the relevance criteria.

References

[1] C.Horwath, W.Chan, E.Leung , H.Pili. "Enterprise Risk Management for Cloud Computing". Research Commissioned by COSO.(June 2012).
[2] "Security Guidance for Critical Areas of Focus in Cloud Computing," Cloud Security Alliance, Dec. 2009; https://cloudsecurityalliance.org/csaguide.pdf.
[3] "Addressing Data Security Challenges in the Cloud". The Need for Cloud Computing Security.
[4] Google, "Britney spears spelling correction," Referenced online at http: //www.google.com/jobs/britney.html, June 2009.
[5] M. Bellare, A. Boldyreva, and A. O'Neill, "Deterministic and efficiently searchable encryption," in Proceedings of Crypto 2007, volume 4622 of LNCS. Springer-Verlag, 2007.
[6] D. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in Proc. of IEEE Symposium on Security and Privacy'00, 2000.
[7] E.-J. Goh, "Secure indexes," Cryptology ePrint Archive, Report 2003/216, 2003, http://eprint.iacr.org/.
[8] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in Proc. of EUROCRYP'04, 2004.
[9] I. H. Witten, A. Moffat, and T. C. Bell. "Managing gigabytes: Compressing and indexing documents and images". Morgan Kaufmann Publishing, San Francisco, May 1999.
[10] R. K. Srihari, Z. F. Zhang & A. B. Rao, (2000) "Intelligent indexing and semantic retrieval of multimodal documents", Information Retrieval, Vol. 2, pp245-275.
[11] D. Lin, (1998) "An information-theoretic definition of similarity", Proceeding of International Conference on Machine Learning.
[12] A.Maind, A.Deorankar and P.Chatur. "Measurement Of Semantic Similarity Between Words: A Survey" IJCSEIT, Vol.2, No.6, December 2012.
[13] R. Rada, H. Mili, E. Bichnell & M. Blettner, (1989) "Development and application of a metric on semantic nets", IEEE Transaction on. Systems, Man and Cybernetics, Vol. 9, No. 1, pp17- 30.
[14] Richardson, R. & A.F. Smeaton, (1995), "Using wordnet in a knowledge-based approach to information retrieval", School of Computer Applications, Dublin City University, Ireland.
[15] P. Resnik, (1995) "Using information content to evaluate semantic similarity in a taxonomy", Proceeding of 14th International Conference on Artificial Intelligence.
[16] J. Jiang & D. Conrath, (1997) "Semantic similarity based on corpus statistics and lexical taxonomy", Proceeding of International Conference on Research in Computational Linguistics (ROCLING X).
[17] M. Sahami & T. Heilman, (2006) "A web-based kernel function for measuring the similarity of short text snippets", Proceeding of 15th International World Wide Web Conference.
[18] R. Cilibrasi & P. Vitanyi,(2007) "The google similarity distance", IEEE Transactions on Knowledge and Data Eng., Vol. 19, No. 3, pp370-383.
[19] D. Bollegala, Y. Matsuo &M. Ishizuka, (2011) "A web search engine-based approach to measure semantic similarity between words", IEEE Transactions on Knowledge and Data Eng., Vol. 23, No. 7, pp977-990.
[20] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou. "Fuzzy keyword search over encrypted data in cloud computing". in Proc. of IEEE INFOCOM'10 Mini-Conference, San Diego, CA, USA, pages 1-5, March 2010.
[21] C. Wang, N. Cao, J. Li, K. Ren, W. J. Lou. "Secure Ranked Keyword Search over Encrypted Cloud Data". Proceedings of IEEE 30th International Conference on Distributed Computing Systems (ICDCS), 2010, pp. 253-262.
[22] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou. "Privacy-preserving multikeyword ranked search over encrypted cloud data". In Proc. of IEEE INFOCOM, pages 829–837, 2011.
[23] Q.Chai and G.Gong. " Verifiable Symmetric Searchable Encryption for Semi-Honest-but -Curious Cloud Servers". Proceedings of IEEE International Conference on Communications (ICC'12), 2012, pp. 917- 922.
[24] W. Sun, B. Wang, N. Cao, M. Li, W. Lou, YT. Hou, H.L. "Privacypreserving multi-keyword text search in the cloud supporting similarity based ranking". in Proc, of ACM CCS, pages 71-82,2013.

[25] J. Feigenbaum, Y. Ishai, T. Malkin, K. Nissim, M. Strauss, and R. N. Wright, "Secure multiparty computation of approximations," in Proc. of ICALP'01.
[26] R. Ostrovsky, "Software protection and simulations on oblivious rams," Ph.D dissertation, Massachusetts Institute of Technology, 1992.
[27] V. Levenshtein, "Binary codes capable of correcting spurious insertions and deletions of ones," Problems of Information Transmission, vol. 1, no. 1, pp. 8–17, 1965.
[28] Philip D. Morehead. The New American Roget's College Thesaurus in Dictionary Form, Turtleback Books, 2002
[29] J. Zobel and A. Moffat, "Exploring the Similarity Space," SIGIR Forum, vol. 32, no. 1, pp. 18-34, 1998.
[30] A. Singhal, "Modern Information Retrieval: A Brief Overview," IEEE Data Eng. Bull., vol. 24, no. 4, pp. 35-43, 2001.
[31] R. Curtmola, J. A. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions," in Proc. of ACM CCS'06, 2006.

N.Jayashri received MSc and MPhil degrees in Computer Science in 2007 and 2010, respectively. She is currently working toward PhD degree in Computer Science Department at AVVM Sri Pushpam College. Thanjavur, India.(Affiliated by Bharathidasan University). Her Research interests are Cryptograpgy, cloud Computing, Information Retrieval.

T.Chakravarthy received the PhD degree from Bharathidasan University. He is currently as an associate Professor in the Computer Science department at AVVM Sri Pushpam College. Thanjavur, India (t(Affiliated by Bharathidasan University). His research expertise includes Human Computer Interaction, Neural Network, Data Mining.