Improving the Efficiency of Data Retrieval in Secure Cloud by Introducing Conjunction of Keywords

Nisha T. M.
PG Student,
CSE Department
MES College of Engineering
Kuttippuram, Kerala

Lijo V. P.
CSE Department
MES College of Engineering
Kuttippuram, Kerala

ABSTRACT
Cloud computing uses internet and central remote servers to maintain data and applications. This allows much more efficient computing by centralizing storage, memory, procession and bandwidth. The data is stored in off-premises and accessing this data through keyword search. Traditional keyword search was based on plaintext keyword search. But for protecting data privacy the sensitive data should be encrypted before outsourcing. So there comes the importance of encrypted cloud data search. One of the most popular ways is selectively retrieve files through keyword-based search instead of retrieving all the encrypted files back. The data encryption also demands the preservation of keyword privacy since keywords usually contain important information related to the data files. So in order to improve adaptation of cloud computing, first ensure its security. Present methods are focusing on the fuzzy keyword search and which efficiently search and retrieve the data in most secure and privacy preserved manner. The existing system uses single fuzzy keyword searching mechanism. A conjunctive/sequence of keyword search mechanism will retrieve most efficient and relevant data files. The conjunctive/sequence of keyword search automatically generates ranked results so that the searching flexibility and efficiency will be improved.

Keywords
Fuzzy keyword, conjunction keyword, sequence keyword, edit distance, wildcard method.

1. INTRODUCTION
Cloud computing is so named because the information being accessed from a centralized storage, and does not need any user to be in a specific place to access it. This is a method in which information is delivered and resources are retrieved by web-based tools and its applications, rather than a direct connection to a server. Data and software packages are stored in servers, and which provides an environment for the employees to work remotely. Users may remotely store and access personal files such as music, pictures, videos, and bookmarks. They can as well play games, and can do word processing on a remote server. Data is centrally stored, so the user does not need to carry a storage medium such as a DVD or thumb drive. Internet-host email providers can consider cloud applications which include web-based Gmail, Hot-mail, or Yahoo! email services. The main services are Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a service (IaaS). There are mainly three types of clouds. Public cloud, Private cloud and Hybrid cloud. The main advantages and challenges [Sameer Rajan, 2011] of cloud computing is given below.

The main advantages of cloud computing are:

- Cloud computing provides faster, simpler and cheaper services.
- It is highly elastic.
- Everything is provided as service, less power consumed on hardware and software.
- High availability and scalability and no data loss.

Some challenges in cloud computing are:

- Security and trust[Lori M 2009].
- Interoperability and availability.
- Service level agreement.

There are so many issues in storing the data securely in the cloud because most of the sensitive information is centralized in to clouds. But the most important aspect arises in data retrieval part. The data owner stores their data in the cloud and any authorized person can access those files. The cloud server provides the authorization, otherwise on retrieval they can do modification, insertion and deletion in the original files and can store back in clouds. So the original data can be mishandled, which may cause security problems. So here encryption plays an important role. That is these sensitive data are encrypted before outsourcing.

One of the most popular ways or techniques is to selectively retrieve files through keyword based search instead of retrieving all the encrypted files back. The data encryption also demands the protection of keyword privacy since keywords usually contain important information related to the data files. The existing searchable encryption techniques do not suit for cloud computing scenario because they support only exact keyword search. This significant drawback of existing schemes signifies the important need for new methods that support searching flexibility, tolerating both minor types and format inconsistencies. A secure fuzzy search [S. Ji, 2009] capability is demanded for achieving enhanced system usability in Cloud Computing. The main problem is how efficiently searching the data and retrieves the results in most secure and privacy preserving manner. For retrieving the data in a most secure and privacy preserving manner the keyword searching technique is used and to search the data in more efficient manner, the fuzzy keyword search is introduced. So efficiency of fuzzy keyword search is the main aspect in the security of data retrieval. When the files are retrieved in an efficient manner, most relevant data can be retrieved. The existing system is mainly focusing on the 'fuzzy keyword search' method. The data that is outsourced is encrypted, constructs fuzzy sets based on both wild card technique and gram based technique, and also introduced a symbol-based trie-traverse search scheme [Xin Zhou 2006, J. Li 2009], where a multi-way tree was constructed for storing the fuzzy keyword set and finally retrieving the data.

2. RELATED WORKS IN KEYWORD SEARCH MECHANISMS

The main challenges are security in data storage, searching, data retrieval etc. Here it is mainly focusing on the data searching and retrieval part. The security in the searching keyword is important because, each keyword should contain the meaning of the underlying information. That is the main reason for concentrating in different searching mechanisms. Traditional encryption techniques support only exact keyword search. This technique is insufficient, because it will retrieve the data, only if the given keyword matches for the predefined keyword set. So for increasing the flexibility in searching so many new searching techniques were introduced. To overcome the exact match searching method another method was proposed, called "fuzzy keyword search".

S. Ji, proposed a new computing paradigm, called *interactive, fuzzy search* [S. Ji, 2009]. It has two unique features such as Interactive [S. Ji, 2009] and Fuzzy [S. Ji, 2009]. This uses 'Straight forward method' for keyword construction. It gives the idea about the queries with a single keyword, and presents an incremental algorithm for computing keyword prefixes [Lori M. Kaufman 2009]. It also gives an idea about various techniques for computing the intersection of the inverted lists of query keywords. Here the keyword set construction needs more space for storing the keywords.

So in order to reduce the space complexity [J. Li 2009, J. Li 2010] another fuzzy keyword search method was proposed which includes 'Wild-card' [Xin Zhou 2006, J. Li 2009, J. Li 2010] based method and 'Gram based' [Xin Zhou 2006, J. Li 2009, J. Li 2010] method for constructing fuzzy keyword sets, a 'symbol-based trie-traverse search scheme' Xin Zhou 2006, J. Li 2010] where a multi-way tree was constructed for storing the fuzzy keyword set and finally retrieving the data. Here the storage space reduces drastically. Gram based method uses less space for keyword construction than wild card method. But in security aspect wild card method is the better.

A Secure Conjunctive Keyword Search [Philippe Golle 2004], was proposed which gives a clear idea about the conjunction of the keyword, which is trying to implement in the proposed system. For the retrieval of the stored data by certain single key word search criterion with exact match keywords are generally used. Here conjunctive keyword search is under practiced. We are applying this in our fuzzy keyword search where exact match fails. By introducing the conjunction of keywords the relevancy will be increase. That is the efficiency will be increasing and generate ranking automatically. In the case of conjunctive keyword search, there are two ways to do it. In the first case user must both give the server capabilities for each of the keyword individually and then do an intersection calculation (by the server or the user) to find the result. In the second case the user may store additional information on the server to make it easier for the searches. In the proposed method we are trying to implement the second method.

A Wild card search method for Digital dictionary based on mobile platform [Xin Zhou 2006] was proposed, it gives a good idea about the Wild card method, and trie tree approach which reduces the search range largely. It also compares the normal trie tree with the advanced trie tree method [Xin Zhou 2006], which includes the fuzzy pointer field, and also gives the idea for the process of inserting a word in the tree. A normal trie-tree data structure [Xin Zhou 2006] is given in the Figure 1. Every node of the tree represents a symbol of the alphabet. Here the trie-tree for "BIG, BIGGER, BILL, GOOD and GOSH" has been constructed.

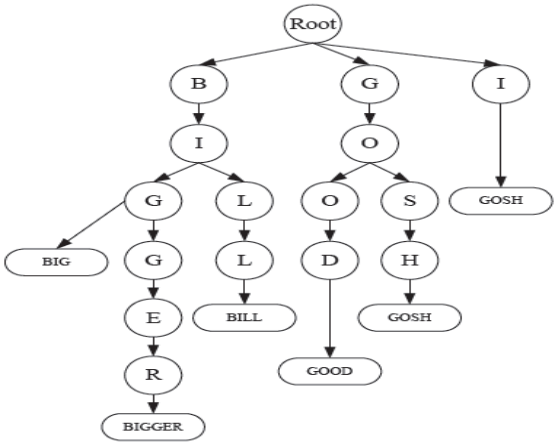

Figure 1. A Normal Trie-tree Data Structure [Xin Zhou 2006 **]**

For example, assume there are 104 keywords in the file collection with average keyword length 10 and d = 2. The output length of hash function is 160 bits. In the present system comparing the storage of keyword set generated for straight forward method, wild card method and gram-based method [J. Li 2009]. This is shown in Table 1. It can be concluded that gram -based method uses less space compared to the other two. But in security aspect wild card method is better.

Table 1: Keyword Set Storage Cost

Keyword set construction Methods	Straight forward	Wild-card	Gram- based
Storage space	30 GB	40 MB	10 MB

3. CONJUNCTION OF FUZZY KEYWORD SEARCH

The working of fuzzy keyword search is described here. Initially given *'n'* encrypted data, which is stored in the cloud server along with that an encrypted predefined set of distinct keywords are also stored in the cloud server. Cloud server provides the authorization to the users who want to access the encrypted files. Only the authorized person can access the stored data. The authorized user types the request that they want to search. The cloud server maps the request to the data files, which is indexed by a file ID and is linked to a set of keywords. Fuzzy keyword search will return the results by keeping the following two rules.

1. If the user's searching input exactly matches the predefined keyword, then the server is expected to return the files containing that keyword.

2. If there is no exact match or some inconstancies in the searching input, the server will return the closest possible results based on pre-defined similarity semantics.

Here it uses 'Wild-card' based method and 'Gram based' method for constructing fuzzy keyword sets, a 'symbol-based trie-traverse search scheme' where a multi-way tree was constructed for storing the fuzzy keyword set and finally retrieving the data. This greatly reduces the storage and representation overheads. It also exploits 'Edit distance' to quantify keywords similarity, to build storage- efficient fuzzy keyword sets to facilitate the searching process.

In this method first we implemented the single fuzzy keyword search mechanism. Wild card method and Gram based method are used for fuzzy keyword set construction, Edit distance for similarity measure and normal trie-tree for data retrieval. Then introduced conjunction of keyword [D. Boneh 2007] or sequence of keywords (AND, OR, NOT, BOTH) in the existing method, so that we can get an idea of the difference between single fuzzy keyword search and sequence of fuzzy keyword search. Here the normal trie-tree is replaced by advanced trie-tree, which contains two fields called exact index and fuzzy pointers. The exact index's value contains the offset of the word's entry in the dictionary, which contain all the information about the word. The fuzzy pointer is a pointer to record a list of keyword's offsets in the dictionary. The root node contain the sequence of words AND, OR, NOT, BOTH, which is represented in the Figure 2.

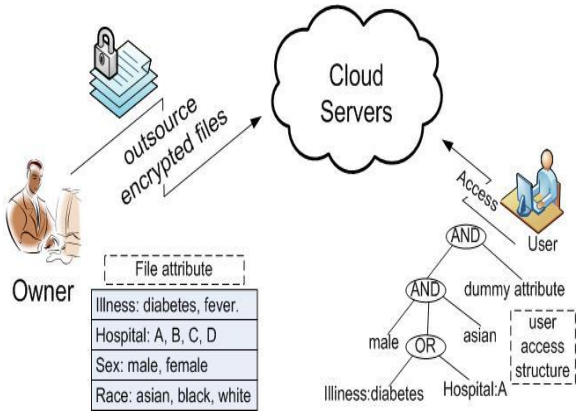

Figure 2. Data Retrieval in from the Cloud Server

Working model: Here it contains a cloud database, which consists of almost one thousand of data files, and has an Administrator (owner) and user. The administrator can add, view and delete the data. When a data or file is added it should be encrypted and stored in the database that is the data entered will be stored as encrypted files. AES encryption algorithm is used for encryption. The administrator can also view the encrypted data and delete the data. This encryption provides the security of the stored data. When a user wants to access the data by keyword search mechanism, first he wants to get the authorization. That is the authorized person can only retrieve the data. The authorization is provided by a key which is randomly generated. That key is unique for each user. The user should remember this key throughout the searching process. The user can enter the key words which is the conjunction of single keywords. That is AND, OR, BOTH and he get a search result which is in a ranked order.

In the existing system we are giving a conjunctive keyword for search and retrieve the data. Here an advanced tire-tree is used for storing this conjunction of keywords and searching each separately. The AND, OR, BOTH are also defined. We are using 'gram based' method and 'wild card' method for 'fuzzy keyword construction'. In both these methods the conjunction of keyword is implemented, which will produce a highly relevant ranked result [C.Wang 2010, Ning Cao 2011]. The advanced trie-tree data structure is used for data retrieval. This will retrieve the data which is also highly efficient.

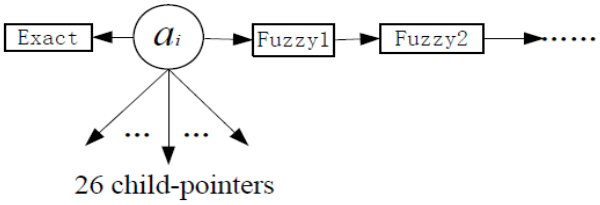

Figure3. A Node of Advanced Trie-Tree [Xin Zhou, 2006**]**

Figure3. shows the node of an advanced trie-tree . It consists of nodes, in which each node has 26 child pointers. In the proposed work the existing trie-tree method is replaced with the advanced trie-tree method.

Figure 4. Shows the process of inserting a word in to the tree. Here two fields namely exact index and fuzzy pointers are added in to the node. The exact index's value contains the offset of the word's entry in the dictionary, which contain all the information about the word. The fuzzy pointer is a pointer to record a list of keyword's offsets in the dictionary.

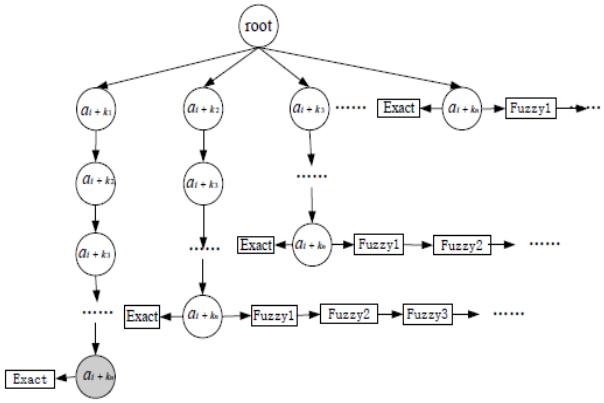

Figure 4. The Process of Inserting a Word in to the Tree
[Xin Zhou , 2006**]**

Algorithm 1 Searching the Tree

Procedure $ConjunctSearch(T_w, j)$
Set as root
for $i = 1 \rightarrow |\ T_w\ |$ **do**
  $l = Currentnode[(hash(\alpha_i) + 26) \times i]$
  if $l\ != NULL$ **then**
    Set $Currentnode$ *as* $Currentnode[l]$
  else
    break
  end if
  if $Currentnode.Exactindex = -1$ **then**
    continue
  else
    break;
  end if
end for
if $Currentnode != root$ **then**
  Appened $Current.FID$ *to* $Result_j.IDSet$;
  Appened $Currentnode.FuzzysetID$ *to* $Result_j.IDSet$;
end if
Return $Result.IDSet$
End procedure.

A searching algorithm is given in Algorithm 1.

3.1 Insertion Steps

1. Initially each character in the given keyword is added in to the node.
2. The exact index field is set to -1 until reaches at the end of the word.
3. When the word's end is reached, the exact index field stores the offset of the file where the word is stored or stores the file ID (FID).
4. The fuzzy pointer will stores the offsets of the fuzzy keywords set. Each fuzzy pointer
can store the fuzzy set with different edit distance.
5. Then the next word reads and the process goes on until all the keyword sets are inserted in the trie-tree.

3.2 Performance

For a normal trie-tree method, for each single keyword request, the search cost only O(m) at the server side, where 'm' is the length of hash value. That is for each character a hash value is calculated. By using this hash value it is directly find the next character. In order to pointing in to the child node each hash value is added with key value, which is specifically defined for each levels in the advanced trie-tree. Space complexity is of O(m * p), p is the total number of words that want to insert. So by introducing conjunction of keyword in normal trie-tree data structure, for 'n' conjunction of keywords having average keyword length 'l' keeping the edit distance 1 is of O(l) * n * O(m). That is O(l * m * n). Space complexity is O(l * m * n * p). Here each fuzzy set is storing separately in the normal trie-tree. So conjunction of keyword will produce the combination of different small normal trie-tree joint together by the root nodes containing AND, OR, NOT etc. But our proposed work the storage complexity and searching complexity is reduced. Which is of the order of O(m * n * p) and O(m) * O(n) respectively.

4. CONCLUSION

Searching the encrypted data from an encrypted keyword, and then retrieving the data is one of the areas where the security issues occur in the cloud computing scenario. Privacy-preserving fuzzy search for achieving effective utilization of remotely stored encrypted data in Cloud Computing is the recent search technique. The wild card method and gram method to construct fuzzy keyword set and edit distance to quantify keywords similarity are used in this system. Also uses an advanced trie-traverse search scheme, where a multi-way tree is constructed for storing the fuzzy keyword set and finally for retrieving the data. In this system we implemented conjunction/sequence of key words for searching. Conjunction/sequence of keywords automatically generates a ranking mechanism according to the keyword sequence which retrieves a highly relevant search result.

As our ongoing work, we will continue to research on security mechanism that supports for complex natural language semantics to produce highly relevant search result. And multiple semantics like weighted query over encrypted data and checking the integrity of the rank order in the search result.

5. REFERENCES

[1] Philippe Golle, Jessica Staddon, Brent Waters. 2004. Secure Conjunctive Keyword Search Over Encrypted Data. Applied Cryptography and Network, Springer.

[2] Xin Zhou, Yunlong Xu, Gongming Chen, Zhigeng Pan. 2006. A New Wild- card Search Method for Digital Dictionary Based on Mobile Platform. Proceedings of the 16th International Conference on Artificial Reality and Telexistence Workshops (ICAT'06), IEEE.

[3] S. Ji, G. Li, C. Li, and J. Feng. 2009. Efficient interactive fuzzy keyword Search, in Proc. of WWW09.

[4] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, March 2009. Enabling Efficient Fuzzy keyword search over encrypted data in cloud computing, in Proc. of IEEE INFOCOM10 Mini-conference, San Diego, CA, USA.

[5] Lori M. Kaufman. July/Aug.2009, Data Security in the World of Cloud Computing, IEEE Security and Privacy, vol. 7, no. 4, pp. 61-64.

[6] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou. March 2010. Fuzzy keyword search over encrypted data in cloud computing, in Proc. of IEEE INFOCOM10 Mini-Conference, San Diego, CA, USA.

[7] C. Wang, N. Cao, J. Li, K. Ren, and W. Lou2010. Secure ranked key word search over encrypted cloud data," in Proc. of ICDCS10.

[8] Ning Cao, Cong Wang, Ming Li, Kui Ren, Wenjing Lou. April 2011. Privacy Preserving Multi-keyword Ranked Search over Encrypted Cloud Data, in Proc. of IEEE INFOCOM11.

[9] D. Boneh and B. Waters. 2007. Conjunctive, subset, and range queries on encrypted data, in Proc. of TCC, pp. 535554.

[10] Sameer Rajan , Apurva Jairath.Cloud Computing. 2011. The Fifth generation of Computing, International Conference on Communication Systems and Network Technologies.