SEMANTIC MULTI KEYWORD SEARCH USING SNIPPETS

moorthy.S
Computer Science and Engineering dept.,
EBET Group of Institutions,
Kangayam, Tiruppur, India
sathiyborntowin@gmail.com

T.kumar,
Computer Science and Engineering dept.,
EBET Group of Institutions,
Kangayam, Tiruppur, India
tr.cse@ebet.edu.in

Abstract **- The advent of cloud computing, data owners are motivated to outsource their complex data management systems from local sites to commercial public cloud for great flexibility and economic savings. But for protecting data privacy, sensitive data has to be encrypted before outsourcing, which obsoletes traditional data utilization based on plaintext keyword search. Thus, enabling an encrypted cloud data search service is of paramount importance. Considering the large number of data users and documents in cloud, it is crucial for the search service to allow multi-keyword query and provide result similarity ranking to meet the effective data retrieval need. Related works on searchable encryption focus on single keyword search or Boolean keyword search, and rarely differentiate the search results. I propose a basic MRSE scheme using secure inner product computation, and then significantly improve it to meet different privacy requirements in two levels of threat models.**

1. Introduction

Semantic search is a data searching technique in which a search query aims not only to find keywords, but also to determine the intent and contextual meaning of the words that a person is using for searching. Semantic search provides more meaningful search results by evaluating and understanding the search phrase and finding the most relevant results in a website, database or any other data repository. Semantic search works on the principle of language semantics. Unlike typical search algorithms, semantic search is based on the context, substance, intent and concept of the searched phrase. Semantic search also incorporates location, synonyms of a term, current trends, word variations and other natural language elements as part of the search. Semantic search concepts are derived from various search algorithms and methodologies, including keyword-to-concept mapping, graph patterns and fuzzy logic.

In the literature, searchable encryption techniques are able to provide secure search over encrypted data for users. They build a searchable inverted index that stores a list of mapping from keywords to the corresponding set of files which contain this keyword. When data users input a keyword, a trapdoor is generated for this keyword and then submitted to the cloud server. Upon receiving the trapdoor, the cloud server executes comparison between the trapdoor and index, and finally returns the data users all files that contain this keyword. But, these methods only allow exact single keyword search.

Some researchers study the problem on secure and ranked search over outsourced cloud data. Wang *et al.*, propose a secure ranked keyword search scheme. Their solution combines inverted index with order-preserving symmetric encryption (OPSE). In terms of ranked search, the order of retrieved files is determined by numerical relevance scores, which can be calculated by $TF \times IDF$. The relevance score is encrypted by OPSE to ensure security. It enhances system usability and saves communication overhead. This solution only supports single keyword ranked search. Cao *et al.*, propose a method that adopts similarity measure of "coordinate matching" to capture the relevance of files to the query. They use "inner product similarity" to measure the score of each file. This solution supports exact multi-keyword ranked search. It is practical, and the search is flexible. Sun *et al.*, proposed a MDB-tree based scheme which supports ranked multi-keyword search. This scheme is very efficient, but the higher efficiency will lead to lower precision of the search results in this scheme.

In addition, fuzzy keyword search have been developed. These methods employ a spell-check mechanism, such as, search for "wireless" instead of "wireiess", or the data format may not be the same *e.g.*, "data-mining" versus "datamining. Chuah *et al.*, propose a privacy-aware bed-tree method to support fuzzy multi-keyword search. This approach uses edit distance to build fuzzy keyword sets. Bloom filters are constructed for every keyword. Then, it constructs the index tree for all files where each leaf node a hash value of a keyword. Li *et al.*, exploit edit distance to quantify keywords similarity and construct storage-efficient fuzzy keyword sets. Specially, the wildcard-based fuzzy set construction approach is designed to save storage overhead. Wang *et al.*, employ wildcard-based fuzzy set to build a private trie-traverse searching index. In the searching phase, if the edit distance between retrieval keywords and ones from the fuzzy sets is less than a predetermined set value, it is considered similar and returns the corresponding files. These fuzzy search methods support tolerance of minor typos and format

inconsistencies, but do not support semantic fuzzy search. Considering the existence of polysemy and synonymy, the model that supports multi-keyword ranked search and semantic search is more reasonable.

In this paper, we will solve the problem of multi-keyword latent semantic ranked search over encrypted cloud data and retrieve the most relevant files.

2. Problem Formulation

A. System Model

The system model can be considered as three entities, as depicted in Figure 1: the data owner, the data user and the cloud server.

Data owner has a collection of data documents $D = \{ d1 , d2 , ..., dm \}$ .A set of distinct keywords $W == \{ w1 , w2 , ..., wn \}$ is extracted from the data collection D . The data owner will

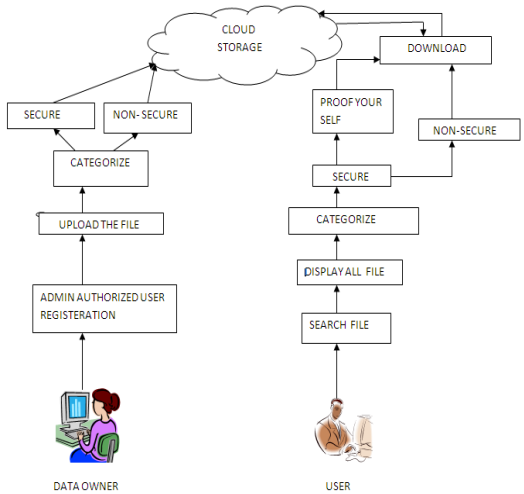
firstly construct an encrypted searchable index I from the data collection D . All files in D are encrypted and form a new file collection, C .Then, the data owner upload both the encrypted index I and the encrypted data collection C to the cloud server.

Data user provides t keywords for the cloud server. A corresponding

trapdoor T_w through search control mechanisms is generated. In this paper, we assume that the authorization between the data owner and the data user is approximately done.

Cloud server received T_w from the authorized user. Then, the cloud server calculates

and returns to the corresponding set of encrypted documents. Moreover, to reduce the communication cost, the data user may send an optional number l along with the trapdoor T so that the cloud server only sends back top- l files that are most relevant to the search query.



Architecture

B. Threat models and Design Goals

The cloud server is considered as "honest-but-curious" in our model. Particularly, the cloud server both follows the designated protocol specification but at the same time analyzes data in its storage and message flows received during the protocol so as to learn additional information.

In this paper, we purpose to achieve security and ranked search under the above model. The designed goals of our system are following:

Latent Semantic Search: We aim to discover the latent semantic relationship between terms and documents. We use statistical techniques to estimate the latent semantic structure, and get rid of the obscuring "noise".The proposed scheme tries to put similar items near each other in some space in order that it could return the data user the files contain the terms latent semantically associated with the query keyword.

Multi-keyword Ranked Search: It supports both multi-keyword query and support result ranking.

Privacy-Preserving: Our scheme is designed to meet the privacy requirement and prevent the cloud server from learning additional information from index and trapdoor.

1) *Index Confidentiality*. The *TF* values of keywords are stored in the index. Thus, the index stored in the cloud server needs to be encrypted;

2) *Trapdoor Unlinkability*. The cloud server could do some statistical analysis over the search result. Meanwhile, the same query should generate different trapdoors when searched twice. The cloud server should not be able to deduce relationship between trapdoors.

3) *Keyword Privacy*. The cloud server could not discern the keyword in query, index by analyzing the statistical information like term frequency.

3. PROPOSED SCHEME

Considering the large number of data users and documents in cloud, it is crucial for the search service to have an efficient search algorithm. Hence, we propose an efficient multi keyword semantic search over the encrypted cloud data. For the efficient arrangement and retrieval of data stored in the cloud, we make use of Utility Pattern (UP) and Frequency Pattern (FP) algorithm. In addition to this, we impose security features for sensitive and secure data storage and access in the cloud environment.

A. FREQUENT PATTERN

The FP-Growth Algorithm is an alternative way to find frequent item sets without using candidate

generations, thus improving performance. For so much it uses a divide-and-conquer strategy. The core of this method is the usage of a special data structure named frequent-pattern tree (FP-tree), which retains the item set association information.

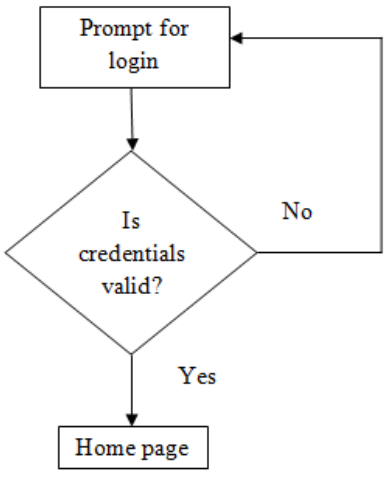
B. UTILITY PATTERN

UP-Growth (Utility Pattern Growth), for mining high utility item sets with a set of techniques for pruning candidate item sets. The information of high utility item sets is maintained in a special data structure named UP-Tree (Utility Pattern Tree) such that the candidate item sets can be generated efficiently with only two scans of the database. The performance of UP-Growth was evaluated in comparison with the state-of-the-art algorithms on different types of datasets.

PROPOSED SYSTEM ADVANTAGES

- **Reliability**
  The proposed system is more secure and reliable. It is established with the help of authorization.
- **Quick access**
  The system can be accessed quickly as searching can be made through the encrypted files.
- **Semantic search**
  Along with the syntactic search, semantic search can also be possible. Semantic search allows searching the content with the meaning of the search query.
  The explanation of the modules is given below:

I. LOGIN MODULE



Login Module

This module is meant for creating login id for the administrators or data owners who may be login into cloud database to register the secured users. Only the secured users are allowed to access the secured files from the cloud database. This module creates the login id and also performs authentication of the data owners who logs into the cloud environment for registration purpose.

II. REGISTRATION MODULE

This module comprises of registering users who are allowed to access or download the secured files from the cloud database. This module is needed to prevent the unauthorized access of secure or private files in the cloud environment. The actor involved in this module is data owner. The Data owner registers the user details, so that the users may able to view the secured details which were stored in the cloud database.

III. FILE UPLOAD MODULE

In the cloud environment, data owners are capable of uploading both secure and non-secure files to the database. The authentication mechanism differs slightly for these two types of files which are detailed as below.

1. Secure file upload

Data owners upload the secure files into the cloud database and also register the users who can view or download these secure files from the cloud environment.

2. Non-secure file upload

File upload is usually done by data owners. In the non-secure file upload, data owners upload the files which can be viewed or downloaded by any user who uses the cloud database.

IV. SEARCH

This module is used for searching the cloud database. The search option is supported with "multi keyword" and "semantic" searching. The multi keyword option is an added feature in the searching which forgives the users for the spelling mistakes. That is, if the user misspells a word, our system will take the actual word which the user intended to search. The semantic search is intended to provide a search method with which the users' search keyword has been taken with its various contexts and each context (meaning) of the searched query will be taken into account for searching.

V. DOWNLOAD

This module enables the cloud users to download the files that are being uploaded into the cloud database. As in the case of file upload module, the download module also consists of two categories – one for secure files and the other one for non-secure files.

1. Secure file download

The users need to prove their identity if they want to download the secure files from the cloud database. When the user clicks on the secure file, the system will prompt for login credentials from the user. Once user enters the login credentials, an email will be sent to the user's email id which they have been given during their registration process. The email would consist of a random number which user needs to enter it back in the login prompt to prove their identity. Once the system identifies the user as secured user, user would be able to download the secured file. Otherwise, the download would be denied.

2. Non-secure file download

Downloading of non-secured file would just be the normal case wherein user can download the files from the cloud database simply. This type of file download doesn't need any authentication techniques.

VI. SECURITY IMPLEMENTATION

High security is provided here by using the login based form. And it sends the random number to the mail through java mail API which is generated while we login in that form. We can download the secure file by the verification of random.

4. CONCLUSION

An efficient algorithm for securely searching over the data stored in cloud has been identified using semantic and multi keyword search techniques. A system capable of downloading the secure and non-secure files from the cloud storage has been designed. In this phase, I have devised the system which has the provision for registering new users, login for existing users, download and upload the files in the cloud database and also the system allows users to search through the files semantically.

References

[1] M.Armbrust, "A view of cloud computing", Communications of the ACM,vol.53, no. 4, (2010),pp. 50-58.
[2] D. Boneh, "Public keyencryption with keyword search", Advances in Cryptology-Eurocrypt 2004,Springer, (2004).
[3] R. Curtmola, "Searchable symmetric encryption: improved definitions and efficient constructions", Proceedings of the 13th ACM conference on Computer and communications security,ACM, (2006).
[4] D.X.Song,D. Wagner and A.Perrig,"Practical techniques for searches on encrypted data in Security and Privacy", 2000. S&P 2000, Proceedings 2000 IEEE Symposium,IEEE, (2000).
[5] C. Wang, "Secure ranked keyword search over encrypted cloud data", Distributed Computing Systems (ICDCS), 2010 IEEE 30th International Conference,IEEE, (2010).
[6] N. Cao,"Privacy-preserving multi-keyword ranked search over encrypted cloud data", INFOCOM, 2011 Proceedings IEEE,IEEE, (2011).
[7] Bellare.M, Boldyreva.A, and O'Neill.A (2007), 'Deterministic and efficiently searchable encryption', in Proceedings of Crypto.
[8] Cong Wang, Student Member, Sherman S.-M. Chow (2013), 'Privacy-Preserving Public Auditing for Secure Cloud Storage'.
[9] Curtmola.R, Garay.J.A Kamara.S, and Ostrovsky.R (2006), 'Searchable symmetric encryption: improved definitions and efficient constructions'.
[10] Hao Wu_, Guoliang Li, and Lizhu Zhou (2013), 'Ginix: Generalized Inverted Index for Keyword Search'.
Jin Li, Qian Wang, Cong Wang, Ning Cao, Kui Ren, and Wenjing Lou (2010), 'Fuzzy Keyword Search over Encrypted Data in Cloud Computing'.
[11] Li.C ,Lu.J, and Lu.Y(2008), 'Efficient merging and filtering algorithms for approximate string searches', in Proc. of ICDE'08.
[12] Liu.F,YuC.T ,Meng.W, and Chowdhury.A(2006), 'Effective keyword search in relational databases'.
Mohamed Nabeel, Elisa Bertino Fellow(2012), 'Privacy Preserving Delegated AccessControl in Public Clouds'.