Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 5.258

IJCSMC, Vol. 5, Issue. 7, July 2016, pg.171 – 176

A Secure Multi-Keyword Search Scheme for Encrypted Cloud Data using Cosine Similarity

Chitra R K[1], Sreeji K S[2], Deepa S[3]

[1]Department of Computer Science and Engineering, Malabar CET, India

[2]Department of Computer Science and Engineering, Malabar CET, India

[3]Department of Computer Science and Engineering, Malabar CET, India

[1] chitra.rk.das@gmail.com; [2] sreejiks71@gmail.com; [3] dpsnkr@gmail.com

Abstract*— Cloud computing enables data owner to store their data remotely in cloud and to enjoy the on-demand access and share the services from a pool with configurable computing resources. This paper solves the problem for searching data from cloud and implement the framework for supporting efficient ranked keyword search for utilize the data in encrypted cloud resources. Secure Multi-keyword Similarity Search framework is proposed using Cipher Text policy encryption algorithm and K-Nearest Neighbour classification technique. Using Cipher Text policy-Attribute Based Encryption (CP-ABE) algorithm to encrypt the cloud data and calculate the similarity computation to construct the index table and ranked based term frequency. Cosine measure along with the TF-IDF scheme can be used to find the most relevant documents to the query. The user accesses the documents through the access control mechanism which provides restricted permission to authorized users and overcome the user revocation problem.

Keywords: Multi Keyword search, Ranking, Indexing, Access control, Cosine Similarity

I. INTRODUCTION

With the great advancement of cloud computing, more and more users came forward to store their huge amount of data and access them in a convenient manner in shared pool of resources. This environment provides accessibility and also flexibility to the computing resources at lower cost. The outsourced data may contain sensitive information such as personal information files, health records and financial documents etc. To provide data confidentiality, the data are to be encrypted before outsourcing so that they cannot be accessed by the cloud provider. Although, encryption is a good solution for privacy requirements, the problem occurs when the user wants to perform the search over the encrypted data. Traditional plain text retrieval methods cannot be directly applied for searching over encrypted data because of the limited operations in cipher text. Also downloading mass amount of data and decrypting them locally is not a practical solution due to the increasing bandwidth cost in cloud systems. Searchable Encryption (SE) allows the user to perform secure search over the encrypted data. In order to reduce the network traffic, only the top k-relevant documents should be returned to the user, not all of them.

Thus, ensuring privacy and effective searching in encrypted cloud data is main problem in cloud storage systems. And considering the large number of data users in and huge amount of outsourced data storage in the cloud, this problem is particularly challenging as it is very difficult to conduct performance analysis.

II. BACKGROUND

A. Cloud computing

The cloud computing technology deals with manipulating, configuring, and accessing the hardware and software resources remotely. It is beneficial to the consumers by providing online data storage, infrastructure, and application. Since management of data and infrastructure in cloud is handled by third-party, it is danger to provide the sensitive information to cloud service providers. There is a number of technologies making cloud computing flexible, reliable, and usable. Some of them include virtualization, service-oriented architecture, grid computation etc. Encryption helps to protect data from being compromised. Although encryption provides security to the data from any unlawful access, it does not prevent data loss.

B. Searchable encryption scheme

The searchable encryption scheme falls to two types: Symmetric searchable encryption and asymmetric Searchable encryption. In private-key searchable encryption, the user encrypts the data with the private key and can use some data structures to make accessing relevant data efficiently. The data and the data structures are encrypted and stored on the server so that only user with the private key can access it. Here the initial work for preprocessing the data is very large, but later work for accessing the data is very small.

Public-key searchable encryption uses both public key and private key. The owner encrypts the data with public key and outsources it to the server. Both public key and private key was generated by the owner. The users having public key can add words to index but only the user who knows the private key can generate trapdoors and performs search.

III. RELATED WORK

The searchable encryption schemes facilitate the clients to store the encrypted data to the cloud and execute secure keyword search over cipher text domain. The single keyword searchable encryption problem is studied by Song et al. [1] for email systems in symmetric setting. This scheme has no index, thus, the search operation performed for the entire file and no ranking operation upon the result document. A secure index using the Bloom filter in [11] is proposed by Goh et al. Curtmola et al. gave the formal definition of the searchable encryption and proposed an index scheme based on the inverted list in [10]. In [9], Wang et al. solved the ranking problem with the keyword frequency and order-preserving encryption. The first searchable encryption scheme proposed by Boneh et al. [3] uses asymmetric encryption. In this work, the searching time is linear to the data collection and no ranking methods implemented. Naveed et al. [4] construct a blind storage system which achieves searchable encryption by concealing access pattern of search user. Data files are divided into blocks and blocks are indexed. All of these works are based on the single keyword search over the encrypted data.

The search is improved by proposing schemes that supporting conjunctive keyword search [6-7]. The proposed privacy-preserving multi-keyword ranked search scheme by Cao et al. [4], has huge computation overhead as it requires calculation of relevance score for all documents. Sun et al. [2] proposed an efficient privacy-preserving multi-keyword search with cosine similarity measure.

Li et al. proposed a wildcard based fuzzy search over encrypted data in [12]. Then Liu et al. [13] improved the scheme by reducing the index size. In [14], Chuah et al. improved [12] by introducing a tree structure index and enriched the search functionality.

IV. PRELIMINARIES

This section gives are view on the secure primitives used in the secure multi-keyword search system.

A. Vector space model

The vector space model allows representation of document as a vector. If a term present in the file, its value in the vector is nonzero, otherwise is zero. A query is also represented as a vector and if the term is queried, the dimension assigned as 1, 0 if not queried. The (tf-idf) weighting involves two components: Term frequency and inverse document frequency. Term frequency denotes the number of occurrences of term t in file f. The inverse document frequency is idf = log (N/df) where N denotes the total number of files and df is the number of files that contains the term t.

B. Ciphertext Policy – Attribute Based Encryption

In CP-ABE, every user associated with a set of attributes and private key is generated based on these attributes. The User's secret key is attached with the attributes and access policy is attached to the cipher text. If the attributes associated private key satisfies the policy, then only authorized user can perform decryption. The access structure contains authorized set of attributes which represent the policy and can be specified by the user who perform encryption. Non genuine users are not able to decrypt the cipher text even if they collude and the access structure is sent in plaintext. A CP-ABE system consists of four algorithms:

- Setup: it's a randomized algorithm and that allows security parameter as input, and returns the public parameters PK and a master key MK as result. PK is used for encryption and MK is used to create user secret keys and is known only to the administrator.
- Encryption: it's a randomized algorithm and we are going to pass input as message M, an access formation T, and the public parameters PK. It produces cipher text CT as output.
- KenGen: it's a randomized algorithm. Here we are going to pass input as the set of a user (say X)'s attributes SX, the master key MK and it produces secret key SK as output and that identifies with SX.
- Decryption: Here we are going to pass input the cipher text CT, a secret key SK for an attribute set SX.

C. B- Tree

The tree contains index nodes and leaf nodes. All leaf nodes are at the same level (same depth). Each index nodes contain keywords and pointers. Each node except root node in a B-tree with order n must contain keys between n to 2n keys. Each node also contains (number of keys + 1) pointers to its child nodes. If the root node is an index node then it must have at least 2 children. The insertion, deletion, search operations takes only logarithmic time.

V. PROPOSED METHODOLOGY

The overview of the proposed secure search system model is given in figure 1.

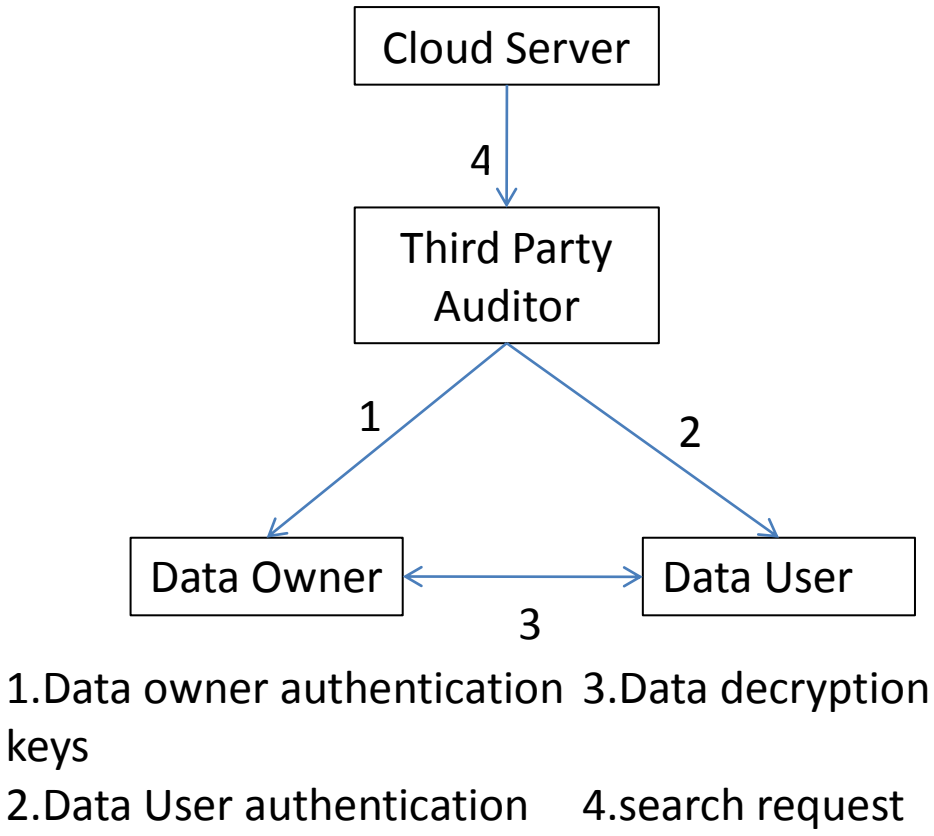

Fig 1: Secure search

In this diagram a third party auditor is shown that is remaining ideal in initial phases. First a user makes a request from the server to authenticate the user. The primary server initiate the third party auditor initiate a authentication process through a web service. When the user authenticated through the user name and password,

a onetime password is generated for secure session. This password is active for only a single session and then the user get access to the system. During this process the third party auditor send the user attributes to the primary server which is used to encrypt and decrypt data for storage.

The proposed system chooses the B-tree as indexing data structure to identify the match between search query and data documents. To calculate the similarity of documents to the search query, make use of inner data association, i.e., the number of keywords in query appearing in document. Each document is converted to a balanced B-tree according to the keywords and encrypted using CP-ABE. Whenever user wants to search, he/she creates a trapdoor for the keywords.

VI. SYSTEM COMPONENTS

Privacy preserving multi-keyword similarity search scheme has the following components.

A. Data Owner

The Data owner has a collection of documents D={d1,d2,…...,dn} which to be outsourced in encrypted form to the cloud server. In PMSS scheme, the data owner builds a secure searchable tree index I from document collection D, using some preprocessing IR operations and then creates an encrypted document collection. Then, the data owner outsources both the encrypted collection C and the secure index I to the cloud server and allowing secure search by the user.

B. data User

Data users are authorized ones to access the documents of data owner. While performing the search, the authorized user can generate a trapdoor TD correspond to 'n' keywords and send to the cloud server. The user can view the top k encrypted documents and if the user satisfying access policy and wish to download, can decrypt the documents with the shared secret key.

C. Cloud server

Cloud server stores the encrypted document collection C and the secure searchable tree index I for the data owner. Upon receiving the trapdoor TD from the data user, the cloud server performs search over the index tree I, and returns the top-k relevant encrypted documents.

VII. SYSTEM ARCHITECTURE

The architecture of privacy preserving multi-keyword similarity search consists of the components such as data owner, data users and the semi-trusted cloud server and the system is illustrated in figure 2. The system will also have a third party auditor to provide authentication and verification. The user is authenticated using attributes that are issued by data owner. The proposed scheme is flexible to replay attacks and using cipher text attribute based encryption (CP-ABE) for authentication purpose; ABE is the one of several cryptographic algorithms, and often used to verify file based on attributes. And also implement attribute based access control whereby access privileges are granted to users during the use of policies which merge attributes together. The policies can use any type of attributes such as user attributes, resource attributes, environment attribute etc.

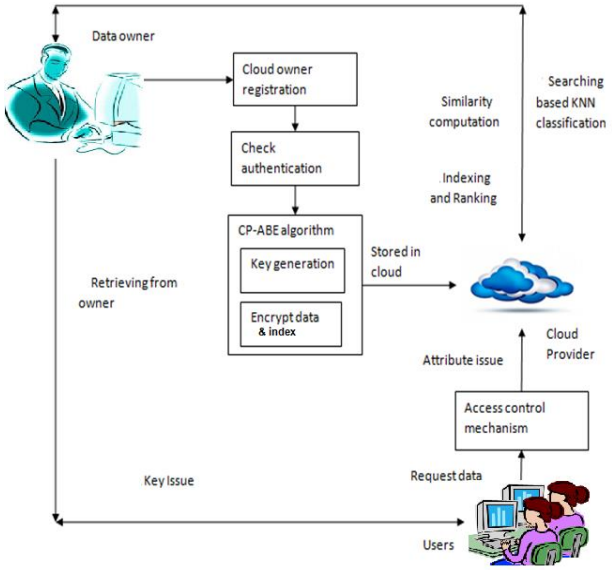

Fig 2: Architecture Diagram

The system consists of following four major modules i.e., File upload, Authentication, search and ranking.

A. File upload

The data owner creates the file and index. First he selects the file and performs some IR processing such as tokenization etc. Also calculate the TF and IDF values of each data item. After measuring the frequency of tokens, they are sorted in descending and most frequent tokens are chosen as keywords for search. These keywords are added to the index table in sorted order. In encryption module, setup() outputs the public key PK and private key SK. The file F and index I is also encrypted using PK and uploaded to the server. The encrypted keywords are inserted into the b-tree for efficient search.

B. Authentication

Authentication procedure is run by the admin. The user sends the authentication parameters (such as a user name and password) to the TPA. The TPA verifies the details and identify whether the user is authorized or not. The user is registering by providing various attributes such as name, password, email etc. and these attributes are used define access policies for the user. Before uploading, the owner may verify the content of their file by sending a request to admin. After processing the request, the admin can accept or reject the file. This ensures that same file is not uploaded many times. Admin and user share a secret key, say ki. Then the user encrypts his personally identifiable information di using ki. In next step, user sends the encrypted data to the admin then, admin decrypts the received data with ki and authenticates user.

C. Search

The data user has many functions such as user registration, keyword search, trapdoor generation and file download. The user can register by giving his/her credentials such as name, password, mail id, phone number etc. In keyword searching function, the user can specify multiple keywords, also the number of relevant documents he need. User creates trapdoor by encrypting the multiple keywords for which search to be performed. The data user generates secure trapdoor such as TrapdGen (Qr, PK) where Qr is the user query and PK is the Primary key which outputs the secure trapdoor Tr. After generating the trapdoors, user send it to the server.

D. Ranking

Document ranking is done by measuring the cosine similarity between the document and query vector. The cosine measure uses TF×IDF rule, where TF denotes the occurrence count of a term within a document (high TF means the term highly correlated to a particular document), and IDF is obtained by dividing the total number of documents in the collection by the number of documents containing the term. This paper adopt the similarity evaluation function for cosine measure from [2],

- fd,t, the TF of the keyword t within the document d;
- ft, the number of documents containing the keyword t;
- N, the total number of documents in the document set;
- wd,t, the TF weight for fd,t;
- wq,t, the IDF weight (query weight);
- Wd, the Euclidean length of wd,t;
- Wq, the Euclidean length of wq,t.

$$Cos(D_d, Q) = \frac{1}{W_d W_q} \sum_{t \in Q \cap D_d} w_{d,t} \cdot w_{q,t}$$

VIII. FUTURE WORKS

Although the scheme provides integrity verification for different data storage systems, does not concerned about the data dynamics. The work needs a predefined dictionary, which makes the dynamic operations complicated. And the system considers only single-owner model. As a future work, the system can be improved by creating dynamic index and extending to support multi-owner model.

IX. CONCLUSIONS

This paper proposes an efficient scheme that supports multi-keyword ordered search and uses CP-ABE algorithm to encrypt the data and index structure. Then implement similarity computation approach for retrieving data with reduced response time in blind cloud storage system. The access control mechanism helps to secure the data from unauthorized access and tries to simplify the file access control to the privilege control, by which all operations upon the cloud data can be handled in a fine-grained manner.

REFERENCES

[1] Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," S&P 2000, vol. 8, pp. 44–55, 2000.

[2] W. Sun, B. Wang, N. Cao, M. Li, W. Lou, T. Hou, and H. Li, "Privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking," in ASIACCS 2013, May 2013.

[3] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," EUROCRYPTO 2004, pp. 506–522, 2004.

[4] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-preserving multi-keyword ranked search over encrypted cloud data," INFOCOM 2011, pp. 829–837, 2011.

[5] M. Naveed, M. Prabhakaran, and C. A. Gunter, "Dynamic searchable encryption via blind storage," in Proc of IEEE Symp. Secur, Privacy, May 2014, pp 639-654

[6] Y. Hwang and P. Lee, "Public key encryption with conjunctive keyword search and its extension to a multi-user system," Pairing 2007, pp. 2–22, 2007.

[7] D. Boneh and B. Waters, "Conjunctive, subset, and range queries on encrypted data," Theory of Cryptography, vol. 4392, pp. 535–554, 2007.

[8] P. Golle, J. Staddon, and B. Waters, "Secure conjunctive keyword search over encrypted data," ACNS 2004, vol. 3089, pp. 31–45, 2004.

[9] C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, "Secure ranked keyword search over encrypted cloud data," ICDCS 2010, pp. 253–262, 2010.

[10] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions," CCS 2006, vol. 19, pp. 79–88, 2006.

[11] E.-J. Goh, "Secure indexes," Cryptology ePrint Archive on October 7th, pp. 1–18, 2003.

[12] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy keyword search over encrypted data in cloud computing," in IEEE INFOCOM 2010, mini-conference, San Diego, CA, USA, March 2010.

[13] Encrypted cloud storage data with small index," ICCCIS 2011, pp. 269–273, 2011.

[14] M. Chuah and W. Hu, "Privacy-aware bedtree based solution for fuzzy multi-keyword search over encrypted data," ICDCSW 2011, pp. 273–281, 2011.

[15] P. Indyk and R. Motwani, "Approximate nearest neighbors: towards removing the curse of dimensionality," Proceedings of the 30th ACM symposium on Theory of computing, vol. 126, pp. 604–613, 1998.

