ISSN: 2321-7782 (Online)
Impact Factor: 6.047

Volume 4, Issue 5, May 2016

International Journal of Advance Research in Computer Science and Management Studies

Research Article / Survey Paper / Case Study

Available online at: www.ijarcsms.com

Retrieving documents from encrypted cloud data in a secured way using cosine similarity search with multiple keyword search support

Shrilakshmi Prasad[1]
M.Tech. in Computer Science & Engineering
Department of Computer Science & Engineering
PES College of Engineering
Mandya, Karnataka – India

B. S. Mamatha[2]
Associate Pofessor
Department of Computer Science & Engineering
PES College of Engineering
Mandya, Karnataka – India

Abstract: ***Cloud computing and storage solutions provide users and enterprises with various capabilities to store and process their data in third party data centers. The enterprises are interested in storing their data in the public cloud. Before uploading the data on to the cloud, it needs to be encrypted to preserve privacy. In order to ease searching, the index file should be built for each document. The index file contains the keyword and its count in the particular document. The unencrypted index file leads to association attack since with the keywords and their count, the content of the document can be known. In this paper, we define and solve the problem of association attack by encrypting the index file using Paillier cryptographic algorithm. So cloud will have the challenge of searching the index file with the search query where both will be in an encrypted format. Hence privacy of the document will be preserved. Cosine similarity search is used to retrieve the top matching documents based on their relevance score. And the beauty of the proposed system is the user can give multiple keywords in their search query.***

Keywords: Cloud computing; index file; association attack; cosine similarity search; multiple keywords.

I. INTRODUCTION

Cloud computing is a technology that uses the internet and the 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 data storage, processing and bandwidth. A simple example of cloud computing is Yahoo email, Gmail or Hotmail etc. All you need is just an internet connection and you can start sending emails. The server and email management software is all on the cloud (internet) and is totally managed by the cloud service provider Yahoo, Google etc. The consumer gets to use the software alone and enjoy the benefits. The analogy is ‘ If you need milk would you buy a cow ?’ All the users or consumers need is to get the benefits of using the software or the hardware of the computer like sending emails etc. Just to get this benefit (milk) why should a consumer buy cow/ hardware/ software ? With the increasing growth of cloud computing , data owners are interested to outsource their complex data management systems from local sites to the commercial public cloud for great flexibility and economic savings. But for protecting data privacy, sensitive data has to be encrypted before uploading to the cloud which obsoletes traditional data utilization based on plaintext keyword search. Thus developing an encrypted cloud data search service is very important. Considering the large number of data users and documents in the cloud, it is necessary to allow multiple keywords in the search request and return documents in the order of their relevance to these keywords. Cosine similarity search is used to find the relevance of the documents.The idea is very beautiful. It is based on the fearsome sounding vector space model for documents. Although it sounds fearsome, the vector space model is actually very simple. The key idea is to transform search from a linguistic problem into a geometric problem. Instead of thinking of document and queries as strings of letters we adopt a point of view in which both documents and queries are represented as vectors in a vector space. In this point of view, the

problem of determining how relevant a document is to a query is just a question of determining the cosine of the angle between them.

Cosine similarity search

The cosine similarity of the two vectors d1 and d2 is defined as cos(d1, d2) = dot(d1, d2)/||d1|| ||d2||. Where dot(d1, d2) = d1[0]* d2[0] + d1[1] * d2[1] +..... and where ||d1|| = sqrt(d1[0]^2 + d1[1]^2.....).

Here the beauty is these two vectors are in an encrypted format using Paillier cryptographic algorithm. So that the cloud will not be aware of the content of the document as well as the search query.

The modules identified are as follows:

Building the index file : The index file is built for each document file by fetching the keywords and their count in the file.

Key generation: Symmetric keys are generated for document encryption using Rijndael algorithm. Asymmetric keys are generated for index encryption using Paillier cryptographic algorithm.

Uploading the documents: The documents and index files are uploaded to the cloud.

Retrieving the documents: The cloud receives the search query and perform cosine similarity search and based on their scores top most matching documents will be retrieved.

Algorithm used in the proposed system is Paillier cryptography

The Paillier cryptosystem, named after and invented by Pascal Paillier in 1999, is a probabilistic asymmetric algorithm for public key cryptography. The problem of computing n-th residue classes is believed to be computationally difficult. The decisional composite residuosity assumption is the intractability hypothesis upon which this cryptosystem is based. The scheme is an additive homomorphic cryptosystem; this means that, given only the public-key and the encryption of m1 and m2, one can find the encryption of m1+m2 .

Algorithm

The schemes work as follows

1. Choose two large prime numbers p and q randomly and independently of each other such that gcd (pq, (p-1)(q-1)) = 1.
2. Compute n = pq and lamda = lcm (p-1, q-1).
3. Select random integer g where g is a randomly generated integer.
4. Ensure n divides the order of g by checking the existence of the following modular multiplicative inverse

   mu = (L(g^ lamda mod n^2))^-1 mod n.

   Where function L is defined as L(u) = u-1/n
5. The public (encryption key) are n and g
6. The private (decryption key) are mu and lamda

Encryption

Let m be a message to be encrypted where m is an integer.

Select random r

Compute cipher text as c = g^ m . r^ n mod n^2

Decryption

Let c be the cipher text to decrypt where c is an integer

Compute the plain text message as m = L(c^ lamda mod n^2) mu. Mod n

II. LITERATURE SURVEY

The keyword searchable encryption scheme allows a user with a trapdoor for a keyword to efficiently retrieve some of encrypted data containing the specific keyword over a remote server. The Conjunctive keyword scheme is proven secure against adaptive chosen keyword attacks in the random oracle model under the external co-diffie Hellman assumption. In this paper, conjunctive keyword search scheme find encrypted data containing each of several keywords just by asking one search query.

The main assumptions made in conjunctive keyword search are:

- There never exists the same keyword in two different keyword fields.
- Every keyword field is pre-defined for every document.

Unfortunately, the drawback of this scheme is that the size of trapdoors is linear and this makes the scheme impartial, this search returns “all-or-nothing”.

In this paper, they have proposed a practical multi-user searchable encryption scheme instead of old approach that is single-user searchable encryption. There are more factors to be considered in the multi-user setting than in the single user setting, ex user accountability, user dynamics. Here they have formulated a model for multi-user searchable encryption and set out security requirements. Under the model they have proposed an efficient scheme, which not only achieves the conventional query privacy, but also posses distinct query keys, efficient yet complete user revocation and query un forget ability features. In order to extend this scheme, need to achieve different capabilities such as fuzzy keyword search and conjunctive keyword search .

Public key encryption with keyword search[PKES] enable senders to send encrypted data to a receiver like traditional public key encryption[PKE]. The main difference between PKES and PKE is that receiver in PKES can search on the encrypted data which is stored on the third party server. And moreover, PKES scheme are based on bilinear map which are costlier in computation. So PKES scheme based on factoring is computationally efficient and secure.

The data sources are encrypted by the standard RSA encryption scheme there is no feasible means decrypt without private key. The main disadvantage of this factoring method, it cannot hide the access pattern which allows the server to infer some information about queries. So the malicious server could store the trapdoors which have been used to search the encrypted data .

To achieve this similarity search, utilize a state-of-the-art algorithm for fast near neighbour search in high dimensional spaces called locality sensitive hashing(LSH). A similarity search problem consists of a collection of data items that are characterised by some feature and a similarity metric to measure the relevance between the query and data items. The goal is to retrieve the items whose similarity against the specified query is greater than a predetermined threshold under the utilized metric. Although exact matching based searchable encryption methods are not suitable to achieve this goal.

Searchable encryption schemes leak some information such as the identifiers of encrypted items corresponding to the trapdoor of a search query. Such leakage may lead to statistical attacks. The LSH is an approximation algorithm for near neighbour search in high dimensional spaces. The basic idea of LSH is to use a set of hash functions to map objects into several buckets such that similar objects share a bucket with high probability, while dissimilar ones do not. LSH uses locality sensitive function families to achieve this goal.

The public key encryption with conjunctive keyword search(PECK) scheme enables one to search a document included multiple encrypted keywords without compromising any original data information. These schemes does not relay on secure channel for sending encryption between client and server, which can reduce communication overhead. But the drawback arises by making some following assumption.

We assume that the same keyword never appear in the two different field. Then assumption is made that every keyword field is defined for every document.

With the popularity of cloud services, some applications require searching in multilingual environment. This paper proposes an algorithm using bloom filters to perform efficient multilingual search. When user sends in a keyword to be searched its language is first determined and its corresponding language with respect to bloom filters are checked for presence of the keyword. To make the algorithm more efficient and accurate, they have created two categories of bloom filter namely, primary bloom filter: this bloom filter is created by using most frequent words used in that language. Secondary bloom filter: this bloom filter consists of total words in the language minus words already used to create primary bloom filter.

Bloom filter is a data structure which can store elements of a set in a space efficient manner. Different variants of bloom filters can be used with this algorithm to suit different requirements of application like deleting, editing etc..

Traditional encryption techniques support only Boolean search. Here we explore statistical measure approach i.e., relevance score, from information retrieval to build a secure searchable index and develop a one-to-many order preserving mapping technique to properly protect those sensitive score information. The two methods used to achieve system security and usability are: Crypto and Information Retrieval(IR) community to design the ranked searchable symmetric encryption(RSSE) scheme.

Then integrate a recent crypto primitive order-preserving symmetric encryption(OPSE) and properly modify it to develop a one-to-many order preserving mapping technique for the purpose to protect sensitive information. However, in symmetric key-based searchable encryption(SSE) schemes, the support of disjunctive Boolean operation or on multi keywords searches still remains an open problem.

Fuzzy keyword search is hard to do over outsource cipher texts. In INFOCOM Li et al proposed a fuzzy keyword search scheme over encrypted data based on edit distance, but this scheme is insecure and the index generation of this scheme cannot make indexes of different keywords independently. So its provable security reduction was incorrect. In this paper, a clever adversary is constructed to break its provable security reduction, such that the adversary always successfully guesses the random coin generated by any challenger .

The fuzzy keyword search on encrypted data allows minor typos and format inconsistency, secure ranked keyword search captures the relevance of data files and returns the results that are wanted by users. These techniques function unilaterally, which reduces the system usability and efficiency. Ranked fuzzy keyword search enhances system usability and efficiency when exact match fails. It returns matching files in ranked order with respect to keyword frequency.

Here it uses the edit distance technique. The edit distance between two words is the number of operations required to transform one of them into the other. The three operations are: Insertion, Deletion and Substitution. The solution exploits the edit distance to quantify keyword similarity and dictionary-based fuzzy set construction to construct fuzzy keyword set. The fuzzy keyword is a simple way spell check mechanisms. However this approach doesn't completely solve the problem and sometimes even can be ineffective.

By combining cloud computing and peer-to-peer computing form a P2P storage cloud which offers highly available strong service with low economic cost. But cloud servers are assumed to be trust worthy but not in reality. P2P brings new challenges for data security. To overcome these issues, "cipher text-policy attribute based encryption[ABE]" schemes are used. But the main disadvantage is it is hard to execute user revocation efficiencies in ABE schemes since each attribute is usually shared by multiple users.

In this paper, Access Control mechanisms for P2P storage Cloud(ACPC) is associated with an expensive access tree which is defined over attributes and associated each with users secrete key with a set of attributes. This scheme aims to provide secure, efficient and fine-grained data access control in P2P storage cloud, but which is not supported by current works .

Here the method called Min hashing is used. Min hash based privacy-preserving multi-keyword search method provide high precision rates. In Min hashing each document is represented by a small set called signature. Important property of signature is that, it should be possible to compare two signatures and estimate distance between the underlying sets without any other information. Although the exact similarity cannot be declared from signatures, they still provide a good approximation. The data owner with only encrypted document, a searchable index are formed and outsourced into semi-honest cloud. Here the index file has not been encrypted. As the result association attack can occur .

The basic idea of similarity search scheme is the state-of-art approximation near neighbour search algorithm in high dimension spaces called locality sensitive hashing(LSH). This LSH is widely used for fast similarity search on plain data. Here each document is transformed into a fingerprint with Sim hash and Hamming distance is used as the similarity score between documents. The Trie-based index is adopted to address the top-k problem and search efficiency is improved greatly .

It consists of a scalable framework where user can use his attribute value and a search query to locally derive a search capability, and a file can be retrieved only when its keywords match the query and the users attribute value. Using this framework a scheme called KSAC was proposed. KSAC utilizes a recent cryptographic primitive called HPE(Hierarchical Predicate Encryption) to enforce fine-grained access control. Access control over encrypted cloud data can be categorized into two group i.e., Key-based access control(KBAC) and Attribute based access control(ABAC).

The search operations over encrypted data is performed at the cloud servers and access control for the in-cloud data is usually enforced by users. Separation of two types of operations can lead to reduced efficiency and compromised privacy for users with a given set of access privileges to search over encrypted cloud data. KBAC assigns each files decryption key directly to authorized users. When a user receives increasing number of such keys accumulated, its load on the management of keys can be too high. To reduce the load, ABAC attaches a set of attribute values to a user (or a file) and designs access policy for a file (or a user, respectively) .

In this paper, it defines and solves the challenging problem of privacy-preserving multi-keyword ranked search over encrypted cloud data(MRSE). Efficient similarity measure of “coordinate matching” is used and further “inner product similarity” to quantitatively evaluate such similarity measure. Relevance ranking is used for effective data retrieval. This can eliminate unnecessary network traffic by sending back only the most relevant data.

Storage services play an important role in a public cloud. To protect the privacy, documents must be encrypted before outsourcing, while many mechanisms have been proposed to support secure search over the encrypted documents, but most of these mechanisms require secure channels to transmit secret information such as the secret keys and trapdoors.

But in this paper Non-Interactive Key Exchanged based on in distinguishability obfuscation(i.e., IO-based NIKE) method is used to share secret information via insecure transmission channels, this new search protocol are designed to deny “unauthorized request of keyword search”. Security analysis shows that the mechanism can guard against an eavesdropper who intends to search over encrypted documents without the permission of the data owner.

III. Related Work

Single Keyword Searchable Encryption

Traditional single keyword searchable encryption schemes usually build an encrypted searchable index such that its content is hidden to the server unless it is given appropriate trapdoors generated via secret key(s). It is first studied by Song et al. in the symmetric key setting, and improvements and advanced security definitions are given in Gob [6], Chang et al. [7] and Curtmola et al. [8]. Our early work solves secure ranked keyword search which utilizes keyword frequency to rank results instead of returning undifferentiated results. However, it only supports single keyword search. In the public key setting, Boneh et al. [9] present the first searchable encryption construction, where anyone with public key can write to the data stored on

server but only authorized users with private key can search. Public key solutions are usually very computationally expensive however. Furthermore, the keyword privacy could not be protected in the public key setting since server could encrypt any keyword with public key and then use the received trapdoor to evaluate this cipherText.

Boolean Keyword Searchable Encryption

To enrich search functionalities, conjunctive keyword search over encrypted data have been proposed These schemes incur large overhead caused by their fundamental primitives, such as computation cost by bilinear map or communication cost by secret sharing . As a more general search approach, predicate encryption schemes are recently proposed to support both conjunctive and disjunctive search. Conjunctive keyword search returns "all-or-nothing", which means it only returns those documents in which all the keywords specified by the search query appear; disjunctive keyword search returns undifferentiated results, which means it returns every document that contains a subset of the specific keywords, even only one key- word of interest. In short, none of existing Boolean keyword searchable encryption schemes support multiple keywords ranked search over encrypted cloud data while preserving privacy as we propose to explore in this paper. Note that, inner product queries in predicate encryption only predicates whether two vectors are orthogonal or not, i.e., the inner product value is concealed except when it equals zero. Without providing the capability to compare concealed inner products, predicate encryption is not qualified for performing ranked search. Furthermore, most of these schemes are built upon the expensive evaluation of pairing operations on elliptic curves. Such inefficiency disadvantage also limits their practical performance when deployed in the cloud. On a different front, the research on top-*k* retrieval in database community is also loosely connected to our problem

IV. PROBLEM FORMULATION

***System model*:** The proposed system has three actors as shown in the fig 1.They are Data owner, Data user and the Cloud. The data owner would be provided with an interface to browse the documents from the system machine. For each browsed document file, index file should be built by fetching the keywords of the document and their count. This index files are encrypted to prevent association attacks. Before uploading the documents and their index files into the cloud ,keys are generated in order to encrypt them. Symmetric and asymmetric keys are generated using Rijndael and Paillier cryptographic algorithm respectively. Once keys are generated they are used to encrypt the document as well as the index files and uploaded into the cloud. In point of data user, the data user will send the search query to the data owner. Data owner in response will encrypt the search query and send back the encrypted form of the search query (trapdoor) and the corresponding decryption keys to the user machine. Data user will send that trapdoor to the cloud. On receiving the encrypted form of search query (trapdoor) from the data user the cloud will perform the cosine similarity search with the index files and calculate the relevance score which is between 0 to 1. Based on these scores the topmost matching documents are retrieved on the user machine. These documents are decrypted on the user machine using the decryption keys and are downloaded.

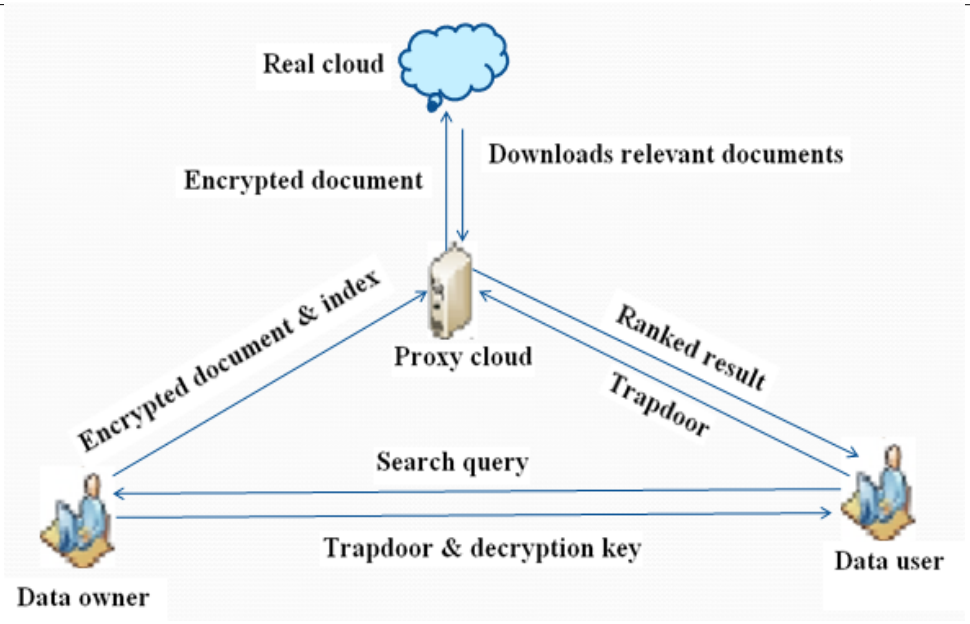

Fig 1: Architecture of the proposed system

***Threat model*:** In this model, firstly the index files are not encrypted so that by knowing the keywords of the file most of the content can be known**.** This leads to association attack. Single keyword search and Boolean keyword search methods are used to search documents. In single keyword search, only one keyword can be given in the search query. In Boolean keyword search, all the keywords that are present in the index file must be given. If any of the keywords in the index file is not given in the search query, then that document will not be given from the cloud.

V. CONCLUSION

In this paper, for the first time, we define and solve the challenging problem of retrieving the documents from the encrypted cloud data in a secured way with the multi keyword search support. Among various multi keyword semantics we choose the efficient similarity measure of 'cosine similarity search' to calculate the relevance score of the documents with the search query by treating the document and query as the vectors. This ranked search system enables data users to find the most relevant information quickly, rather than burdensomely sorting through every match in the content collection. This can also elegantly eliminate unnecessary network traffic by sending back only the most relevant data, which is highly desirable in the "pay-as-you-use" cloud paradigm. On the other hand, since the index files are also been encrypted , the privacy of the documents is preserved from the association attacks.

In the future work, we will explore supporting other multi keyword semantics (e.g., weighted query) over encrypted data and checking the integrity of the rank order in the search result.

References

1. L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, "A break in the clouds: towards a cloud definition," ACM SIGCOMM Comput. Commun Rev vol. 39, no. 1, pp. 50-55, 2009.
2. S. Kamara and K. Lauter, "Cryptographic cloud storage," in RLCPS, January 2010, LNCS. Springer, Heidelberg.
3. A. Singhal, "Modern information retrieval: A brief overview," IEEE Data Engineering Bulletin, vol. 24, no. 4, pp. 35-43, 2001.
4. I. H. Witten, A. Moffat, and T. C. Bell, "Managing gigabytes: Compressing and indexing documents and images," Morgan Kaufmann Publishing, San Francisco, May 1999.
5. D. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in Proc. of S&P, 2000.
6. E.-J. Goh, "Secure indexes," Cryptology ePrint Archive, 2003, http:// eprint.iacr.org/2003/216.
7. Y.-C. Chang and M. Mitzenmacher, "Privacy preserving keyword searches on remote encrypted data," in Proc. of ACNS, 2005.
8. R. Curtmola, J. A. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions," in Proc. Of ACM CCS, 2006.
9. D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in Proc. of EUROCRYPT, 2004.

10. M. Bellare, A. Boldyreva, and A. ONeill, "Deterministic and efficiently searchable encryption," in Proc. of CRYPTO, 2007.
11. M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, and H. Shi, "Searchable encryption revisite Consistency properties, relation to anonymous ibe, and extensions," J. Cryptol., vol. 21, no. 3, pp. 350-391, 2008.
12. 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, March 2010.
13. D. Boneh, E. Kushilevitz, R. Ostrovsky, and W. E. S. III, "Public key encryption that allows pir queries," in Proc. of CRYPTO, 2007.
14. P. Golle, J. Staddon, and B. Waters, "Secure conjunctive keyword search over encrypted data," in Proc. of ACNS, 2004, pp. 31-45.
15. L. Ballard, S. Kamara, and F. Monrose, "Achieving efficient conjunctive keyword searches over encrypted data," in Proc. of ICICS, 2005.
16. D. Boneh and B. Waters, "Conjunctive, subset, and range queries on encrypted data," in Proc. of TCC, 2007, pp. 535-554.
17. R. Brinkman, "Searching in encrypted data," in University of Twente, PhD thesis, 2007.
18. Y. Hwang and P. Lee, "Public key encryption with conjunctive keyword search and its extension to a multi-user system," in Pairing, 2007.
19. J. Katz, A. Sahai, and B. Waters, "Predicate encryption supporting disjunctions, polynomial equations, and inner products," in Proc. of EUROCRYPT, 2008.
20. A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption," in Proc. of EUROCRYPT, 2010.
21. E. Shen, E. Shi, and B. Waters, "Predicate privacy in encryption systems," in Proc. of TCC, 2009.
22. "Privacy preserving multikeyword ranked search over encrypted cloud data" by Ning cao, Cong wang, Ming li and Wenjing Lou in IEEE transactions on parallel and distributed systems vol: 25 no: 1 year : 2014.

