Indian J. Res. Found., (2017) 6, 24-29. ISSN 2454-6577

INDIAN JOURNAL OF RESEARCH FOUNDATION
www.indianjournal.net

RESEARCH ARTICLE

RANKED KEY SEARCH AND EFFICIENT RETRIEVAL OF GRAND DATA ON CLOUD COMPUTING

R. S. Aashmi*[a], J. Jeeva Santhana Selvi[b]

[a] *Department of Computer Science And Engineering, Rajas International Institute Of Technology For Womens, Nagercoil -629 001, Tamilnadu, India*

[B] *Department of Computer Science And Engineering, Rajas International Institute Of Technology For Womens, Nagercoil -629 001, Tamilnadu, India*

Received 10 March, 2017; Accepted 15 April, 2017
Available online 22 April, 2017

Abstract

Cloud computing cautiously facilitate the standard of data service outsourcing. However, to protect data solitude, responsive cloud data has to be encrypted before subcontract to the commercial public cloud. The conventional methods explore the encryption techniques which allow the users to shelter the search over encrypted data through keywords, they hold only Boolean search and are not yet adequate to meet the client's prerequisite. To prevail over that, this project introduced secure ranked keyword search over encrypted cloud data is used. Ranked search enhances the system usability. It results to retrieve the file accuracy instead of sending undifferentiated results. Specifically, explore the statistical measure method is used to fetch the relevant data. One to many order preserving mapping procedure is used to shield the receptive data in the cloud. One to many order preserving mapping is an server side ranking. Technique protects from without losing keyword privacy. De-dupliction is done in result set to increase system usability.

Keywords

Cloud Computing
Encryption
Statistical measure
Mapping
De-dupliction

Introduction

As cloud computing becomes ubiquitous more and more receptive information are being federal into the cloud such as e-mails, personal health records, company economics data, and authority documents. The fact that data owners and cloud server are no extended in the similar trusted domain may spot the outsourced unencrypted data at risk the cloud server may leak data information to unauthorized entities or even be hacked.

Data encryption makes efficient data consumption a very difficult task given that there could be a huge total of outsourced data folder. As well in cloud computing data owners may contribute to their outsourced data with a hefty number of users might want to only recuperate certain specific data files they are concerned in during a given session. One of the most prevalent ways to do so is using keyword-based search. Such keyword search method allows users to exclusively recuperate files of interest and has been widely applied in plaintext search circumstances.

*Corresponding author Tel. +91 7358829897
E-mail : Ashashmi12@gmail.com

Published by GM SOFTWARE

Earlier searchable encryption techniques allow a user to securely search over encrypted data using keywords with no initial decrypting method support only conservative Boolean keyword search devoid of capturing any relevance of the files in the search result. When directly applied in large mutual data outsourcing cloud surrounding that may undergo from the following two main shortcoming.

For the first time it describe the problem of secure ranked keyword search over encrypted cloud data and provide such an effective protocol which fulfills the secure ranked search purpose with little significance score information leakage against keyword privacy. Absolute security analysis shows that the ranked searchable symmetric encryption technique certainly enjoys as-strong-as-possible security guarantee evaluate to previous searchable symmetric.

On exploring the practical deliberation and augmentation of our ranked search mechanism includes the efficient support of significance score dynamics which validate ranked search results and the invalidate of our anticipated one-to-many order-preserving mapping technique. To enable ranked searchable symmetric encryption for effective utilization of outsourced and encrypted cloud data under the aforementioned model this system design should achieve the following security and performance guarantee. De-duplication is a specialized data compression technique for eliminating duplicate copies of repeating data. By delete the redundant copy duplication can be avoided.

II PROBLEM STATEMENT

2.1 The system and thread model

we consider an encrypted cloud data hosting service involving three different entities, as illustrated in Fig. 1: data owner, data user, and cloud server. Data owner has a collection of n data files C = (F_1; F_2; : : : ; F_n) that he wants to outsource on the cloud server in encrypted form while still keeping the capability to search through them for effective data utilization reasons. To do so, before outsourcing, data owner will first build a secure searchable index I from a set of m distinct keywords W =(w1;w2;:::::wm) extracted from the file collection C, and store both index I and the encrypted file collection C on the cloud server.

III RELATED WORK

3.1 Searchable Encryption

Focus on security definition formalizations and efficiency improvements. Fir introduced the notion of searchable encryption. Proposed a scheme in the symmetric key setting, where each word in the file is encrypted independently under a special two-layered encryption construction. Thus, a searching overhead is linear to the whole file collection length. developed a Bloom filter based per-file index, reducing the work load for each search request proportional to the number of files in the collection.

3.2 Secure top-k retrieval from data base community

The idea of uniformly distributing posting elements using an order-preserving cryptographic function was first discussed. How-ever, the order- preserving mapping function proposed. which does not support score dynamics, i.e., any insertion and updates of the scores in the index will result in the posting list completely rebuilt. uses a different order-preserving mapping based on pre-sampling and training of the relevance scores to be outsourced, which is not as efficient as our proposed schemes.

IV PROPOSED WORK

Ranked keyword search is used to fetch the relevant file. This ranked keyword support only statical measure approach. By calculating term frequency and inverted table exacted datas are fetched out. And one to many order preserving mapping technique help to protect the sensitive information. To avoid duplication in result set de-duplication techniques are used.

4.1 RANKED KEYWORD SERACH

Development of a private cloud is very expensive. Storage of sensitive data in public cloud is very risky. To make it possible, unauthorized access is avoided by storing the data in encrypted format. This paper tackles the problems of enabling searchable encryption system with support of secure ranked search in order to implement the top k retrieval. In this paper, statistical measure approach from IR and text mining to embed weight information of each file during establishment of searchable index before outsourcing the encrypted file collection is explored. Team frequency: Number of times a particular keyword appears within the file. Inverse document frequency (IDF): It is calculated as the total number of files by the number of files in particular keyword. Ranking function: It is calculated by using TF*IDF rule.

4.2 DE-DUPLICATION

De-duplication is a technique used to reduce the

amount of storage needed by the service providers. Client side de-duplication saves both network band width and the storage cost. Bring secure issue due to multi users. Several attackers target either the band width consumption or the confidentiality and privacy of legitimate cloud user.

4.3 One-to-many Order-preserving Mapping

In order to reduce the amount of information leakage from the deterministic property, an one-to-many OPSE scheme is thus desired, which can flatten or ob-furcated the original relevance score distribution, increase its randomness, and still preserve the plaintext order. encryption process of original deterministic OPSE, where a plaintext m in domain D is always mapped to the same random-sized non-overlapping interval bucket in range R, determined by a keyed binary search over the range R and the result of a random HGD sampling function.

A. SYSTEM ARCHITECTURE

Large number of datas are stored in cloud. To access the data following steps to be done

i) Login
ii) Privilege Access
iii) Ranked Keyword Search
iv)De-duplication
v) Retrieval Of Data

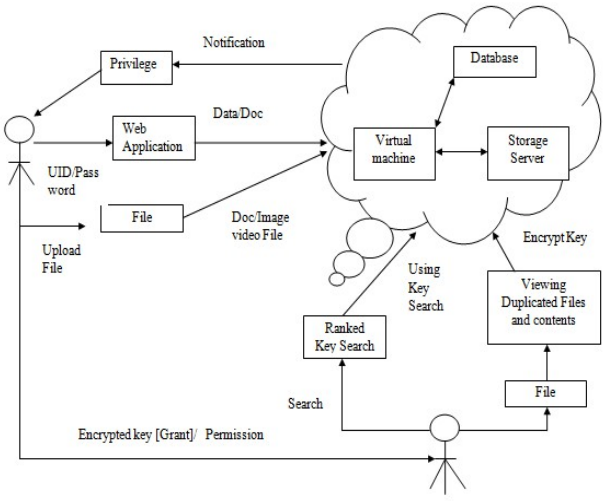

Fig 1.1: System Architecture

i) LOGIN

Clients can be sure that data returned from the public cloud server originated from the data owner and has not been tampered with. Clients are not allowed to access data outside of what they are querying. To access cloud data, a user has to be granted privilege by the owner with respect to the private keys related to the subject of his search. After authentication client can down load data from the cloud and decrypt the data locally.

ii) PRIVILEGE ACCESS

Authentication Module describes the interface between the user and system and the admin provided the type of authentication. The user is allowed to create his testimonial to login into the system. An admin needs to approve the users created and login approval the user will be allowed to access the application. Authentication is provided by encrypting the user name and password, protecting sensitive information from users. In this module, the users are allowed to upload their documents. While uploading, the user is allowed to provide the access nature on the document. The documents will be archived with the help of rich streaming API.

Which integrates the SQL Server Database Engine with a file system by storing binary (max) binary large object data as files on the file system. In addition, the user will be given the option of revoking the access at any time. Based on the provided accessibility, the documents will be accessed by the other users. Internally, the access details will be logged in the Access log file. In this module the uploaded document is parsed by using the document parser interface. A mechanism with the top-down keyword parsing technique reads the complete document with specified keywords. In addition it matches the version of the content type definition that is used by a list or document library.

iii) RANKEDKEY WORD SEARCH

In this module the secondary users can search for the documents and access the documents based on the user provided access. The search is made very accurate using a keyword. An automatic pure logging of access details which includes time, access personnel details, document details were loaded.

iv) De-DUPLICATION

De-duplication is a technique used to reduce the amount of storage needed by the service providers. Client side de-duplication saves both network band width and the storage cost. Bring secure issue due to multi users. Several attackers target either the band width consumption or the confidentiality and privacy of legitimate cloud user. If any file or content found to be duplicated then the duplicated files undergoes delete operation.

v) RETRIEVAL OF DATA

The major advantage of this is related to maintaining the history of the documents. All the files that are uploaded are stored in the library. Various versions of the documents were maintained by the cloud database. The data owner will be provided an option of taking back the previous version documents. This is one of the major jargons specified in our proposed system**.**

V IMPLEMENTATION

5.1 ALGORITHM

A. KEYGEN

Setup *a*lgorithm to generate a public key *PK* and a secret key *MSK*. Next uses *PK* to encrypt (with algorithm Encrypt) her PII and gets cipher text *CT*. Then, she can store *CT* (the encrypted *PII*) on an un trusted host (e.g., in a cloud). may also publish *PK*, so that it can be used to encrypt data that she can access. User has the function *p* representing a predicate that she wishes to evaluate for her encrypted PII. She uses the Key Gen algorithm, *PK*, MSK and *p* to output the token TKp (encoding *p*). Then, she gives TKp to the host that evaluates the token (with *p* included in the token) for *CT* (the encrypted PII), and returns the result p(PII) to that KeyGen uses the secret key *MSK* as input. KeyGen to generate TKp for *p.*

1. Setup *PK*, MSK
2. Encrypt(*PK*, PII) CT
3. KeyGen(*PK*, *MSK*, *p*) TKp

B. BUILDINDEX

For an unprotected term frequency table, both the search term and its term frequency information are in plaintext. To protect the confidentiality of the search, we encrypt each of them in an appropriate way.

A word *w* in a document first undergoes stemming to retain the word stem and to remove the word ending. The stemmed word *wS* is then encrypted using an encryptio function *E* and the *word-key KwS* - to obtain the encrypted word $w(e)$ $S = E(KwS, wS)$.

The word-key is unique to each stemmed word and is obtained with a key derivation function. $w(e)$ S is further mapped to a particular row i in the term frequency table where the index i is established via a hashing function such that $i = H(w(e)S)$. The term frequency information is collected by counting the number of occurrences of the stemmed word in the *jth* document, and stored in the table entry $\{TF(i, j)\}$. This process is repeated to obtain the term frequencies for all terms and documents, and the TF values are then further encrypted.

C. TRAPDOOR GEN

The notion of a security parameter k which will be provided as input to all algorithms. For technical reasons, the security parameter is given in unary and is thus represented as 1k. Larger value of the security parameter scheme. (Hopefully, the concrete example that follows will give some more motivation for the purpose of the security parameters.) A trapdoor permutation family is a tuple of ppt algorithms (Gen, Sample, Eval, Invert)

1. Gen(1k) is a probabilistic algorithm which outputs a pair (i; td). (One can think of i as indexing a particular permutation over some domain Di, while td represents some trapdoor" information that allows inversion of fi.)

2. Sample(1k; i) is a probabilistic algorithm which outputs an element x 2 Di (assuming i was output by Gen). Furthermore, x is uniformly distributed in Di. (More formally the distribution f Sample(1k; i)g is equal to the uniform distribution over Di.)

3. Eval(1k; i; x) is a deterministic algorithm which outputs an element y 2 Di (assuming iwas output by Gen and x 2 Di). Furthermore, for all i output by Gen, the function Eval(1k; i; _) : Di ! Di is a permutation. (Thus, one can view Eval(1k; i; _) as corresponding to a permutation fi mentioned above.)

4. Invert(1k; td; y) is a deterministic algorithm which outputs an element x 2 Di, where (i; td) is a possible output of Gen.

D. SEARCH INDEX

When the search begins, the client sends the query phrase with multiple keywords, *k1, . . . , kn*, to a client-side server, which concatenates the keywords to a list, *K*. The client-side server then encrypts each $k \in K$ using Z' in which the order of keywords is randomized . Each keyword in this list is truncated to β bits to create the encrypted keyword list, *K′*. The client-side server then transfers this encrypted query, *K′*, to the un trusted cloud server. The cloud server parses *K′* into individual encrypted keywords *k′* and, using the inverted index, determines the documents, δ, that contain a *k′*

5.2 TECHNIQUE USED

A. FILE LEVEL DE-DUPLICATION

The files are converted into small segments. These segments which is of the size of 8kb to 64 kb. Then the segments are checked for redundancy. Duplication is identified if the segment has the same hash value. The hash value are generated using message digest. If duplicated data find out it undergoes delete operation

B.BLOCK LEVE DE-DUPLICATION

Block level de-duplication performed over blocks. The files are divided into blocks and each block is checked for its redundancy. Block level de-duplication detect the small redundant chunk of data using hash algorithm. The result of hash algorithm is is unique file Id the block cloud be in the size of 4 kb. Duplicated datas are deleted

VI EXPERIMENT

6.1 Performance evaluation

The efficiency of our proposed one-to-many order-preserving mapping is determined by both the size of score domain M and the range R M affects how many rounds (O(log M)) the procedure BinarySearch( ) or HGD( ) should be called. Meanwhile, M together with R both impact the time consumption for individual HGD( ) cost. That's why the time cost of single one-to-many mapping order-preserving operation goes up faster than logarithmic, as M increases. The result represents the mean of 100 trials. Note that even for large range R, the time cost of one successful mapping is still finished in 200 milliseconds,when M is set to be our choice 128.

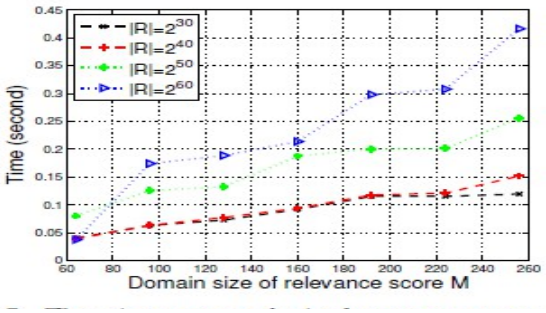

Fig1.2: time cost of single one to many order preserving

VII CONCLUSION

An initial attempt to motivate and solve the problem of supporting efficient ranked search for achieving effective utilization of remotely stored encrypted data in cloud computing

VII FUTURE ENHANCEMENTS

The proposed system also investigate some further enhancements of ranked search mechanism, including the efficient support of relevance score dynamics, the authentication of ranked search results, and the reversibility of proposed one-to-many order-preserving mapping technique.

. REFERENCES

[1] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in Proc. of EUROCRYP'04, volume 3027 of LNCS. Springer, 2004.

[2] Y.-C. Chang and M. Mitzenmacher, "Privacy preserving keyword searches on remote encrypted data," in Proc. of ACNS'05, 2005.

[3] 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.

[4] E. Shi, J. Bethencourt, H. Chan, D. Song, and A. Perrig, "Multi-dimensional range query over encrypted data," in Proc. of IEEE Symposium on Security and Privacy'07, 2007..

[5] A. Swaminathan, Y. Mao, G.-M. Su, H. Gou, A. L. Varna, S. Wu, and D. W. Oard, "Confidentiality-preserving rank-ordered search," in Proc. of the Workshop on Storage Security and Survivabil-ity, 2007.

[6] C. Nang, S. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Secure Cloud Storage," IEEE Transactions on Computers (TC), to appear.

[7] C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, "Secure ranked keyword search over encrypted cloud data," in Proc. of ICDCS'10, 2010.

[8] C. Wang, Q. Wang, K. Ren, and W. Lou, "Towards Secure and Dependable Storage Services in Cloud Computing," IEEE Transactions on Service Computing (TSC), to appear.