ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 9, September 2015

Keyword Search over Outsourced Cloud Data with Improved Ranking Techniques

Deepa R, Reeja S L

M.Tech Student, Dept. of CSE, Marian Engineering College, Trivandrum, Kerala, India

Assistant Professor, Dept. of CSE, Marian Engineering College, Trivandrum, Kerala, India

ABSTRACT: Cloud Computing becomes prevalent, sensitive information are being increasingly centralized into the cloud. For the protection of data privacy, sensitive data has to be encrypted before outsourcing, which makes effective data utilization a very challenging task. Although traditional searchable encryption schemes allow users to securely search over encrypted data through keywords, without capturing any relevance of data files. Ranked search greatly enhances system usability by enabling search result relevance ranking instead of sending undifferentiated results, and further ensures the file retrieval accuracy. The statistical measure approach, i.e. relevance score, from information retrieval to build a secure searchable index. . In proposed system, stemming and Synonym clustering is applied on the score calculation of each keyword in the file collection, it increase the score value and it help to retrieve more relevant files.

KEYWORDS: Ranked search, searchable encryption, confidential data, cloud computing.

I. Introduction

Cloud computing provide reliable services delivered through data centres that are built on virtualized compute and storage technologies. Cloud Computing becomes more sensitive information are being centralized into the cloud such as e-mails, personal health records, company finance data, and government documents, etc. The fact that data owners and cloud server are no longer in the same trusted domain may put the outsourced unencrypted data at risk the cloud server may be leak data information to unauthorized entities are hacked[4].

Data encryption makes effective data utilization is a very challenging task. It follows that sensitive data has to be encrypted prior to outsourcing for data privacy and combating unsolicited accesses. Data encryption makes effective data utilization a very challenging task given that there could be a large amount of outsourced data files. Besides, in Cloud Computing, data owners may share their outsourced data with a large number of users, who might want to only retrieve certain specific data files they are interested in during a given session. One of the most popular ways to do so is through keyword-based search. Such keyword search technique allows users to selectively retrieve files of interest.

Although traditional searchable encryption schemes [7] allow a user to securely search over encrypted data through keywords without first decrypting it, these techniques support only conventional Boolean keyword search, without capturing any relevance of the files in the search result. When directly applied in large collaborative data outsourcing cloud environment, they may suffer from the following two main drawbacks. On the one hand, for each search request, users without pre-knowledge of the encrypted cloud data have to go through every retrieved file in order to find ones most matching their interest, which demands possibly large amount of post processing over-head; On the other hand, invariably sending back all files solely based on presence/absence of the keyword further incurs large unnecessary network traffic, which is absolutely undesirable in today’s pay-as-you-use cloud paradigm. In short, lacking of effective mechanisms to ensure the file retrieval accuracy is a significant drawback of searchable encryption schemes in the context of Cloud Computing. In information retrieval (IR) community has already been utilizing various scoring mechanisms [14] to quantify and rank-order the relevance of files in response to any given search query.

Ranked search greatly enhances system usability by returning the matching files in a ranked order regarding to certain relevance criteria thus making one step closer towards practical deployment of privacy-preserving data hosting services in the context of Cloud Computing. To achieve our design goals on both system security and usability, propose to bring together the advance of both crypto and IR community to design the ranked searchable symmetric encryption scheme, in the spirit of "as-strong as- possible" security guarantee [8]. Specifically, explore the statistical measure approach from IR and text-mining to embed weight information (i.e. relevance score) of each file during the establishment of searchable index before outsourcing the encrypted file collection.

II. RELATED WORK

Searchable encryption has been widely studied as a cryptographic primitive, with a focus on security definition formalizations and efficiency improvements. Songet al. [5] first introduced the notion of searchable encryption. They 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. Goh [6] 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. Chang et al. [10] also developed a similar per-file index scheme. To further enhance search efficiency, Curtmola et al. [7] proposed a per-keyword based approach, where a single encrypted hash table index is built for the entire file collection, with each entry consisting of the trapdoor of a keyword and an encrypted set of related file identifiers. Searchable encryption has also been considered in the public-key setting. Boneh et al. [9] presented the first public-key based searchable encryption scheme, with an analogous scenario to that of [9].

III. LITERATURE SURVEY

Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction[3].

A. *Essential Characteristics:*

Main five characteristics of cloud computing are

1. On-demand self-service: A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider.
2. Broad network access: Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, tablets, laptops, and workstations).
3. Resource pooling: The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or datacenter). Examples of resources include storage, processing, memory, and network bandwidth.
4. Rapid elasticity: Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time.
5. Measured service: Cloud systems automatically control and optimize resource use by leveraging a metering capability1 at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.

B. *Searchable symmetric encryption*

Searchable symmetric encryption (SSE) allows a party to outsource the storage of its data to another party (a server) in a private manner, while maintaining the ability to selectively search over it. Private-key storage outsourcing allows clients with either limited resources or limited expertise to store and distribute large amounts of symmetrically encrypted data at low cost. Since regular private-key encryption prevents one from searching over

encrypted data, clients also lose the ability to selectively retrieve segments of their data. To address this, several techniques have been proposed for provisioning symmetric encryption with search capabilities the resulting construct is typically called searchable encryption [7]. This leads to following drawbacks,
1) Non relevant data search result
2) Large unnecessary network traffic, which is absolutely undesirable in today's pay-as-you-use cloud paradigm.
3) Decrease the efficiency and File retrieval accuracy.

C. *Ranked Searchable Symmetric Encryption(RSSE)*

Cloud Computing ranked search greatly enhances system usability by returning the matching files in a ranked order regarding to certain relevance criteria (e.g., keyword frequency), thus making one step closer toward practical deployment of privacy-preserving data hosting services in the context of Cloud Computing. To achieve our design goals on both system security and usability, we propose to bring together the advance of both crypto and IR community to design the ranked searchable symmetric encryption (RSSE) scheme, in the spirit of "as-strong-as possible" security guarantee. Specifically, we explore the statistical measure approach from IR and text mining to embed weight information (i.e., relevance score) of each file during the establishment of searchable index before outsourcing the encrypted file collection[1].

A ranked searchable encryption scheme consists of four algorithms (KeyGen, BuildIndex, TrapdoorGen,SearchIndex). Ranked searchable encryption system can be constructed from these four algorithms in two phases Setup and Retrieval:

• Setup: The data owner initializes the public and secret parameters of the system by executing KeyGen, and pre-processes the data file collection C by using BuildIndex to generate the searchable index from the unique words extracted from C. The owner then encrypts the data file collection C, and publishes the index including the keyword frequency based relevance scores in some encrypted form, together with the encrypted collection C to the Cloud. As part of Setup phase, the data owner also needs to distribute the necessary secret parameters to a group of authorized users.

• Retrieval: The user uses TrapdoorGen to generate a secure trapdoor corresponding to his interested keyword, and submits it to the cloud server. Upon receiving the trapdoor, the cloud server will derive a list of matched file IDs and their corresponding encrypted relevance scores by searching the index via SearchIndex. The matched files should be sent back in a ranked sequence based on the relevance scores.

In information retrieval, inverted index is a widely-used indexing structure that stores a list of mappings from keywords to the corresponding set of files that contain this keyword, allowing full text search[11]. For ranked search purposes, the task of determining which files are most relevant is typically done by assigning a numerical score, which can be precomputed. In information retrieval, a ranking function is used to calculate relevance scores of matching files to a given search request. The most widely used statistical measurement for evaluating relevance score in the information retrieval community uses the TF× IDF rule, where TF (term frequency) is simply the number of times a given term or keyword (we will use them interchangeably hereafter) appears within a file (to measure the importance of the term within the particular file), and IDF (inverse document frequency) is obtained by dividing the number of files in the whole collection by the number of files containing the term[2].

$$\text{Score}(t, Fd)=(1/|Fd|)*\text{TF of term } t. \qquad \text{equ}(1)$$

|Fd| - Length of the file Fd.

To enable ranked searchable symmetric encryption for effective utilization of outsourced and encrypted cloud data under the aforementioned model, our system design should achieve the following security and performance guarantee.

D. *Stemming algorithm*

A stemming algorithm is a process of linguistic normalisation, in which the variant forms of a word are reduced to a common form, for example,

connection
connections
connective ---> connect
connected
connecting

It is important to appreciate that use stemming with the intention of improving the performance of IR systems. For stemming, porter stemming algorithm is a process of removing suffixes from words in English. Removing suffixes is automatically is an operation which is especially useful in the field of information retrieval[16].

E. Stopwords

It has been traditional in setting up IR systems to discard the very commonest words of a language - the stopwords - during indexing. A more modern approach is to index everything, which greatly assists searching for phrases for example getting a list of stopwords can be done by sorting a vocabulary of a text corpus for a language by frequency, and going down the list picking off words to be discarded. Alternatively, stopwords could be removed before the stemming algorithm is applied, or after the stemming algorithm is applied.

F. Wordnet

WordNet is a lexical database for the English language. It groups English words into sets of synonyms called synsets, provides short definitions and usage examples, and records a number of relations among these synonym sets or their members. WordNet can thus be seen as a combination of dictionary and thesaurus. Words from the same lexical category that are roughly synonymous are grouped into synsets[12].

IV. PROPOSED ALGORITHM

Ranked search greatly enhances system usability by returning the matching files in a ranked order regarding to certain relevance criteria. The task of determining which files are most relevant is typically done by assigning a numerical score. Details of Ranked Searchable Symmetric Encryption mentioned in literature survey. To improve the score value, this system uses improved ranking technique.

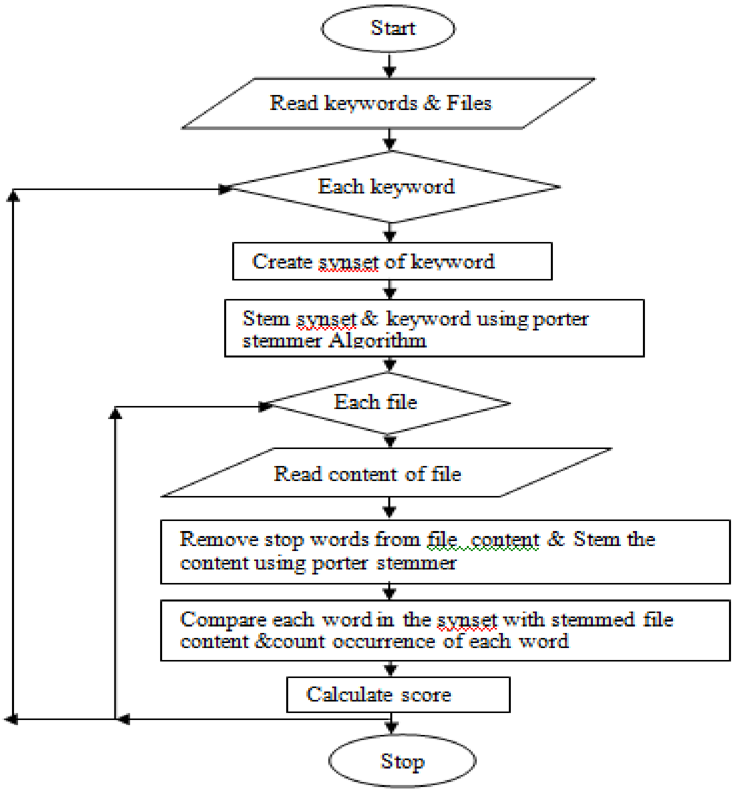
Start
Read keywords & Files
Each keyword
Create synset of keyword
Stem synset & keyword using porter stemmer Algorithm
Each file
Read content of file
Remove stop words from file content & Stem the content using porter stemmer
Compare each word in the synset with stemmed file content &count occurrence of each word
Calculate score
Stop

Figure 1: Proposed Algorithm

In stemming different forms of a word are reduced to a common form which improves the performance of information retrieval process.WordNet groups English words into sets of synonyms called synsets.For stemming process system uses porter stemmer algorithm. For to find synset of word Wordnet2.1 is used. This two technique improved the score value and helps to retrieve more relevant files.

Proposed algorithm is shown in Figure 1.First read the keywords and files. Consider each keyword, generate the synset of each keyword and stem the keyword using porter stemmer algorithm [15]. Consider each file, remove the stop word from each file and stem the content of the file using porter stemmer algorithm[13].Compare each word in the synset with stemmed file content and count the occurrence of each word. Calculate the keyword ranking score using count as term frequency.

In Ranked Searchable Symmetric Encryption returning the matching file in ranked order. Ranking is done with the help of numerical score using the formula mentioned in equ(1).Proposed algorithm improved the score value with the help of stemming algorithm and wordnet.It improves the keyword search over outsourced cloud data.

V. EXPERIMENTAL ANALYSIS

This section provides experimental analysis on the proposed system based on actual system implementation using J2EE and Android. Score of proposed is score2 and existing is score1 is given below. Here three files named Basic, Cloudstore and Ranked and three keyword named compute, encryption and secure are considered. Term Frequency(TF) without applying stemming algorithm and with applying stemming algorithm are taken. If proposed algorithm is applied in the score calculation the score is improved as shown below.

Keyword	File	TF without apply porter stemmer (1)	TF with apply porter stemmer (2)	Score1	Score2
Compute	Basic	2	18	3.0	9.5
	Cloudstore	2	26	1.0	9.0
	Ranked	2	48	1.0	16.3
Encryption	Basic	0	0	0	0
	Cloudstore	18	32	6.3	11.0
	Ranked	58	67	19.6	22.6
Secure	Basic	0	8	0	4.5
	Cloudstore	4	25	1.6	8.6
	Ranked	17	100	6.0	33.66

Figure 2: Comparison of score values & Term frequency

DOI: 10.15680/IJIRCCE.2015. 0309103

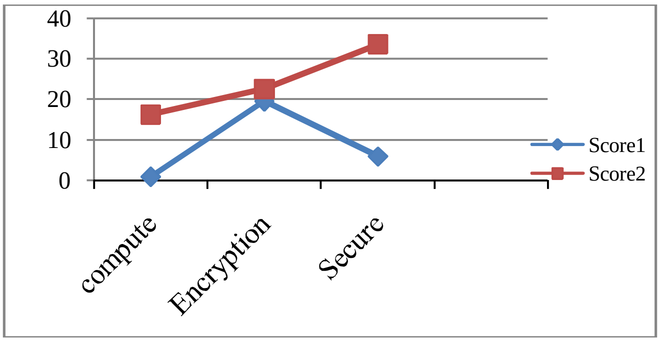

Figure 3: Scores of different keywords in Ranked file.

Figure 3 compare the score of proposed system and the existing system as score2 &score1.Graph is drawn based on the comparison table Figure 2.The score2 of each keyword is greater than score1.Hence score is improved which helped to retrieve more relevant files.

VI. Conclusion and Future Work

In Cloud Computing more sensitive information are being centralized into the cloud such as e-mails, personal health records, company finance data, and government documents, etc.So to protect cloud data it must be encrypted before outsourced to public cloud. Ranked search greatly enhances system usability by returning the matching files in a ranked order regarding to certain relevance criteria. Stemming & Synonym clustering improved the score value and help to retrieve more relevant files. In future using another improved ranking technique will improve the score value.

References

1. Cong Wang, Ning Cao, Kui Ren, Wenjing Lou, Senio(2012), "Enabling Secure and Efficient Ranked Keyword Search over Outsourced Cloud Data", *IEEE Transactions on Parallel and Distributed systems, VOL.23,NO.8.*

2. 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.

3. P. Mell and T. Grance, "Draft nist working definition of cloud computing," Referenced on Jan. 23rd, 2010 Online at http://csrc. nist.gov/groups/SNS/cloud-computing/index.html, 2010.

4. M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the clouds: A berkeley view of cloud computing," University of California, Berkeley, Tech. Rep. UCBEECS- 2009-28, Feb 2009.

5. 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.

6. E.-J. Goh, "Secure indexes," Cryptology ePrint Archive, Report 2003/216, 2003

7. R. Curtmola, J. A. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions," in *Proc. of ACM CCS'06*, 2006.

8. Cloud Security Alliance, "Security guidance for critical areas of focus in cloud computing," 2009, http://www.cloudsecurityalliance.org.

9. 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.

10. A. Singhal, "Modern information retrieval: A brief overview," *IEEE Data Engineering Bulletin*, vol. 24, no. 4, pp. 35–43, 2001.

11. S. Zerr, D. Olmedilla, W. Nejdl, and W. Siberski, "Zerber+r: Top-kretrieval from a confidential index," in Proc. of EDBT'09, 2009.

12. 12. Julian Sedding &Dimitar Kazakov "WordNet-based Text Document Clustering" Department of Computer Science University of York Heslington, York YO10 5DD, United Kingdom.

13. Daniel Waegel "The porter stemmer" CISC889/Fall 2011.

14. I. H. Witten, A. Moffat, and T. C. Bell, "Managing gigabytes: Compressing and indexing documents and images," Morgan Kaufmann Publishing, San Francisco, May 1999.

15. Http:/Tartarus.Org/artin /Porter stemmer

16. Donna Harman" Ranking Algorithms" national Institute Of Standards And Technology"

BIOGRAPHY

Deepa R is a M Tech computer science student in Marian Engineering college Kazhakootam Trivandrum Kerala. She completed her B Tech with first class in Computer Science from Cochin University.

Reeja S L is working as an Assistant professor at Computer Science and Engineering department of Marian Engineering College Kazhakootam Trivandrum Kerala.