EFFICIENT DATA RETRIEVAL FROM CLOUD STORAGE USING DATA MINING TECHNIQUE

D.Pratiba[1], Dr.G.Shobha[2] and Vijaya Lakshmi.P.S[3]

Asst. Prof., Dept. of CSE, RVCE, Bangalore, India
HOD, Dept. of CSE, RCVE, Bangalore, India
M.Tech student, Dept. of CSE, RCVE, Bangalore, India

ABSTRACT

Cloud computing is an emanating technology allowing users to perform data processing, use as storage and data admission services from around the world through internet. The Cloud service providers charge depending on the user's usage. Imposing confidentiality and scalability on cloud data increases the complexity of cloud computing. As sensitive information is centralized into the cloud, this information must be encrypted and uploaded to cloud for the data privacy and efficient data utilization. As the data becomes complex and number of users are increasing searching of the files must be allowed through multiple keyword of the end users interest. The traditional searchable encryption schemes allows users to search in the encrypted cloud data through keywords, which support only Boolean search, i.e., whether a keyword exists in a file or not, without any relevance of data files and the queried keyword. Searching of data in the cloud using Single keyword ranked search results too coarse output and the data privacy is opposed using server side ranking based on order-preserving encryption (OPE).

The proposed scheme, guarantees top-n multi keyword retrieval over encrypted cloud data with high privacy and practical efficiency using vector space model and TRSE ,where in the majority of computing work is done on the server while the user takes part in ranking.

KEYWORD

Cloud Computing, RSA, Vector Space Model, two round searchable encryption.

1. INTRODUCTION

Cloud computing is a rapidly growing technology where resources such as storage devices, platform and applications are shared over the internet and is widely used by multiple users in small and medium business. Cloud services can be provided and delivered remotely by vendors such as Amazon or Microsoft as "public clouds", or the resources are designed, installed, monitored and controlled internally as "private clouds". Cloud data retrieval is an important service to be considered as certain specific data files the users are interested during a given session must be retrieved in an efficient way and quickly.

Services provided by the clouds are broadly divided into three categories: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS)

Infrastructure as a service (IaaS): In the IaaS model computers are offered as physical or as virtual machines, and other resources.

DOI: 10.5121/ijci.2015.4226

Platform as a service (PaaS): In the PaaS model, cloud providers offers a computing platform including operating system, programming language execution environment, database, and web server. Without the cost and complexity of buying and managing the respective hardware and software layers, application developers can develop and run their software solutions on a cloud platform.

Software as a service (SaaS): In the SaaS model, cloud providers install and operate application software in the cloud and cloud users access the software from cloud clients. The cloud users do not manage the cloud infrastructure and platform on which the application is running. This eliminates the need to install and run the application on the cloud user's own computers simplifying maintenance and support.

2. RELATED WORKS

To manage data the cloud systems rely on Distributed File Systems (DFS). Examples include Google's GFS [1] and Hadoop's HDFS [2]. There are several methods for the cloud data retrieval. Given a query, the corresponding data are retrieved from the DFS and sent to a set of processing nodes for parallel scanning. This employs simple query processing strategy and is suitable for a specific purpose of a single organization. In contrast, in an open service Cloud system, such as Amazon's EC2, different clients deploy their own software products in the same Cloud system. Processing nodes are shared among the clients and hence managing data becomes more complicated. Therefore, a more efficient data access service was required instead of scanning.

An indexing framework for the cloud system was designed. In this framework, processing nodes are organized in a structured overlay network, and each processing node builds its local index to speed up data access. A global index is built by selecting and publishing a portion of the local index in the overlay network and it is distributed over the network. Each node maintains a subset of the global index. This retrieves all the files which is impractical in cloud computing scenarios [3].

Considering, cloud computing system hosts data services as in figure 1, consists of three entities. They are cloud server, data owner and data user. The cloud server hosts data storage and retrieves services. Since data stored in cloud may contain sensitive information, the outsourced files must be encrypted.

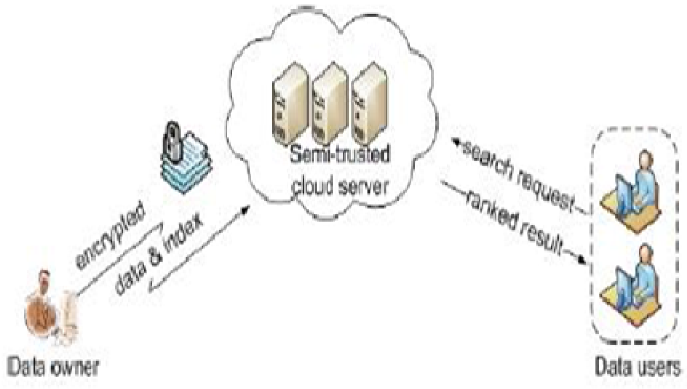

Figure.1 Encrypted cloud data retrieval

The data owner outsources the encrypted form of the collection of n files C = {f1,f2,…fn} onto the cloud server and expects the cloud server to provide keyword retrieval service to data owner or other authorized users. The keyword retrieval is possible by building a searchable index I from a collection of l keywords W={w1,w2,…wl} extracted out of C, and then outsources both the encrypted index I and encrypted files onto the cloud server. The data user is authorized to retrieve their interested files over the outsourced encrypted data through the multi keyword search. The data user sends the query to the cloud server that returns the relevant files to the data user. Next, the data user decrypts the files and makes use of the files.

Searchable encryption focus on the different methods to selectively retrieve the users interested files through keyword-based search technique which is widely applied in plaintext search, such as Google [4]. In Boolean searchable symmetric encryption it does not consider the difference of relevance with the queried keyword and of the files while searching whether a keyword consists in a file or not. It focuses on simple keyword matching and has long computation phase for each document [5]. Hence the traditional plaintext searching methods are unsuitable for cloud computing and restricts users ability to perform search over encrypted cloud data [6]-[8].

Searchable encryption techniques have been developed in recent years to securely search over encrypted data [9]–[14]. These Searchable encryption schemes build an index for each keyword of interest and associate the index with the files that contain the keyword. The file content and keyword privacy are well-secured and provides effective keyword search. The existing searchable encryption techniques do not suit for cloud computing scenario since they support only exact keyword search.

In secured ranking search, to retrieve the top matching files from the encrypted cloud an framework is built for privacy-preserving top-k retrieval, including secure indexing and ranking with OPE [15].

An ranking model was proposed, which allows privacy-preserving top-k retrieval from an outsourced inverted index. They proposed a transformation function for the relevance score to make relevance scores of different terms indistinguishable and hence improve the security of the indexed data [16].

On the basis of SSE, the one-to-many OPM was proposed to further improve the efficiency. However, these schemes support only single keyword retrieval and not suitable for the large number of data users and documents in the cloud. Hence it is necessary to allow multi keyword search request and return the most relevant documents in the order of their relevance with these keywords [17].

To retrieve the files quickly with multi keyword search over encrypted cloud files several schemes supporting Boolean multi keyword retrieval was proposed. [18]–[19]. This system checks for the files matching the multi keyword given during the search.

Fuzzy keyword search greatly enhances system usability by returning the matching files when users' searching inputs exactly match the predefined keywords or the closest possible matching files based on keyword similarity semantics, when exact match fails. Fuzzy keyword search uses wild card-based technique, Gram-based technique and symbol-based tie-traverse search system to return the files exactly matching the users searching keyword and the already defined keywords or based on keyword similarity semantics. [20]. It supports only Boolean keyword search and not support ranked search.

Cao et al. first attempted to define and solve the problem of top-k multi keyword retrieval over encrypted cloud data. The relevance scoring evaluation and the similarity was computed through the coordinate matching and inner product [21].

Hu et al. employed homomorphism to preserve the data privacy. The data privacy of the owner and the query privacy of the client is protected by devising a secure protocol for processing k-nearest-neighbor (kNN) index query. These schemes employed Boolean representation in their searchable index, i.e., 1 denotes the corresponding term exists in the file and 0 otherwise. Thus, files that share queried keywords have the same score, thus weakens the effectiveness of data utilization. The security is opposed since all these server-side schemes employ server-side ranking based on OPE. We, therefore, focus on the security, an issue the above schemes fail to address [22].

3. MOTIVATION

As Cloud Computing becomes prevalent, sensitive information are centralized into the cloud. The sensitive data must be encrypted before outsourcing them into the cloud for the protection of data privacy, which makes effective data utilization a very challenging task. Data users must be allowed to search and retrieve the files of theirs area of interest.

The traditional searchable encryption schemes allows users to search in the encrypted cloud data through keywords, which support only Boolean search, i.e., whether a keyword exists in a file or not, without any relevance of data files and the queried keyword.

In the ranked search the matching files in a ranked order are returned through single keyword search which yields too coarse result and server side ranking based on order-preserving encryption (OPE) inevitably violates data privacy.

The proposed scheme, guarantees top-n multi keyword retrieval over encrypted cloud data with high security and practical efficiency using TRSE scheme.In the proposed system score computation is performed at the server while the user takes part in ranking,

4. PROPOSED WORK

4.1. Aim of the Proposed System

- The proposed system ensures protection of sensitive information by encrypting cloud data at the administrator side and provides the data user with secure encrypted data.
- The system provides authentication to the multi users by registering themselves to the administrator.
- The users must retrieve top-n files matching the multi keywords submitted during searching of the files through the relevance of the files and the keyword.
- Preserves privacy of encrypted data using Two round searchable encryption method. The Multi-keyword ranked search using TRSE scheme retrieves data accurately when compared to single keyword search.

4.2. Techniques Used

4.2.1 Encryption Techniques

As sensitive data are outsourced to the cloud, these data must be encrypted before they are outsourced into the cloud for the privacy purpose.

In the proposed system the files are encrypted using RSA encryption function by the administrator before outsourcing them into the cloud.

- **RSA Encryption**

RSA algorithm is developed by Ron Rivest, Adi Shamir and Len Adleman and it is the first public-key encryption algorithm. RSA is a block cipher in which the plaintext and ciphertext are integers between 0 and n-1 for some n.

The following summarizes the RSA algorithm. Selecting two prime numbers, p and q calculate their product n, which is the modulus for encryption and decryption. Next, we need the quantity ø(N) ,referred to as the Euler totient of n, which is the number of positive integers less than n and relatively prime to n. Then select an integer e that is relatively prime to ø(N). Finally, calculate d as multiplicative inverse of e, mudulo ø(N). To encrypt a message M the sender obtains public key of recipient KU={e,N} and computes: $C=M^e \bmod N$, where $0 \le M < N$. To decrypt the ciphertext C the owner uses their private key KR={d,p,q}and computes: $M=C^d \bmod N$.

- **Homomorphic Encryption**

An index of keywords is constructed using document parser interface where the unnecessary words are ignored and only the keywords of the document is listed in the vector word index. This index in encrypted using homomorphic encryption and uploaded to the cloud.

To eliminate the burden on the user side computing work should be performed at the server side, hence we need an encryption scheme which guarantee the operability and security at the same time on server side. Homomorphic encryption allows specific types of computations to be performed on the corresponding ciphertext. The result is the ciphertext, the result of the same operations performed on the plaintext. That is, homomorphic encryption performs computation of ciphertext without knowing anything about the plaintext to get the correct encrypted output. The original fully homomorphic encryption scheme, is too complicated and inefficient for practical utilization as it employs ideal lattices over a polynomial ring. To compute the relevance scores from the encrypted searchable index for the top-n retrieval using vector space model we need only addition and multiplication operations over integers, hence the original homomorphism in a full form is reduced to a simplified form that only supports integer operations, which is more efficient than the full form.

Let p and c denote the plaintext and ciphertext of the integer, respectively. Our encryption scheme can be expressed as the following formulation: $c=sn+2r+p$ where s denotes the secret key, n denotes the multiple parameter, and r denotes the noise to achieve proximity against brute-force attacks. M is the plaintext .The public key is sn + r.

On the basis of homomorphism property, the encryption scheme can be described as four stages: KeyGen, Encrypt, Evaluate, and Decrypt.

- KeyGen (K). The secret key p is an odd n-bit number randomly selected. The set of public keys PK =(k0,k1,k2…kr) is the subset of sn+r .where r is a small even number noise factor, n is the multiple of s say n.s Note that the secret key is used for encryption and the public keys are used for decryption.
- Encrypt (PK, p). ciphertext $c = n\cdot s+2\cdot r+p$.

- Evaluate (c1; c2; . . . ; ct). Integer x is returned by applying the binary addition and multiplication gates to ciphertext ci.
- Decrypt (s,x). c (mod s) = 2·r+b (mod s)

4.2.2 TRSE

A Two Round Searchable Encryption (TRSE) scheme supports top-n multi keyword retrieval. The TRSE framework consists of four algorithms: Set, IndexTable, TrapGen, ScoreCal and Ranking.

- Set(S): The secret key and public keys for the encryption scheme are generated by the data owner. The security parameter S is taken as the input, the output are a secret key SK and a public key PK.
- IndexTable(N ,PK): From the file set C the data owner builds the secure searchable index.
- TrapGen (REQ,PK): The data user generates secure trapdoor from his request.
- ScoreCal (T,I): The encrypted result vector V is returned to the data user by the server as it computes the scores of each files in I with trapdoor T.
- Ranking(V,SK): The data user decrypts the vector V with secret key SK.

4.2.3 Vector Space Model

Vector space model is used to order the cloud files either in ascending or descending order based on the relevance scoring. tf-idf weighting is the widely used model for relevance scoring, which involves two attributes-term frequency and inverse document frequency. Term frequency (tft, f) is the number of occurrences of term t in file f. Document frequency (dft) refers to the number of files that contains term t. The inverse document frequency (idft) is defined as: idft = log N/dft , where N denotes the total number of files. Then the tft-idft weighting scheme assigns to term t a weight in file f given by tft -idft,f = tft,f × idft . The weights of terms that occur very frequently in the collections are diminished and the weights of terms that occur rarely are increased by introducing IDF factor.

The vector space model is an algebraic model which represents each file and query as a vector. For a file, each dimension corresponds to a separate term, i.e., if a term occurs in the file, its value in the vector is non-zero integer, otherwise is zero and for a query each dimension is assigned with 0 or 1 according to whether the term is queried. It computes the degree of similarity between queries and files, and then ranks the files according to their relevance scores. It meets the needs of top-n retrieval. The score of file f on query q (sof, q) is deduced by the inner product of the two vectors: sof, q = vf .q.0 Once the scores are computed, files can be ranked in order and hence the most relevant files can be found.

Consider, a table below showing the vector space model computation for the query: “best colleges” in the document: “best colleges in Karnataka”.

Table 1. Similarity scoring between query and document

Query					**Document**			
Word	***tft***	***dft***	***idft***	***wtt***	***tft***	***wtt***	***cn’lized***	***pro***
Karnataka	0	250	0.30	0	1	1	0.57	0.17
Best	1	300	0.22	0.22	1	1	0.57	0.12
Colleges	1	400	0.09	0.09	1	1	0.57	0.05

Limit of the files collected is considered to be 500. “Frequent term, tft”, “Document frequency, dft”, “Inverse Document Frequency, idft”, “Final weight of the term in the query or document, wtt”, Document weights after cosine normalization, cn’lized”, “Product of final query weight and final document weight, pro”. Final similarity score between query and document: Pi wqi*wdi = 0.17 + 0.12 + 0.05 =0.34

5. Overview of The Proposed System

In the proposed scheme, the data owner encrypts the file and the searchable index to the amazon cloud. Files are encrypted using RSA encryption and apply Homomorphic encryption to encrypt the searchable index. The cloud server computes the scores from the encrypted index stored on cloud and returns the encrypted file with its scores to the data user as the server receives a query consisting of multi- keywords from the user. Once the scores are received the user decrypts the scores and selects the files and sends the respective file IDs to the server. Server sends the encrypted file to the user and then the user decrypts the file using the private key sent by the administrator.

It takes two-round communication between the cloud server and the data user to retrieve the top-n files and hence we name the scheme the Two round searchable encryption scheme, in which score calculation is done at the server side and ranking is performed at the user end.

6. Performance Analysis

In Secure searchable encryption technique single keywords are used to retrieve the interested files of the users which yields in coarse results and performs server-side ranking which burdens the server. The proposed system uses multi keyword in the query for searching the files using TRSE scheme.

The graph below indicates the time to upload set of 100 files into the amazon cloud after encrypting them using RSA algorithm by the administrator.

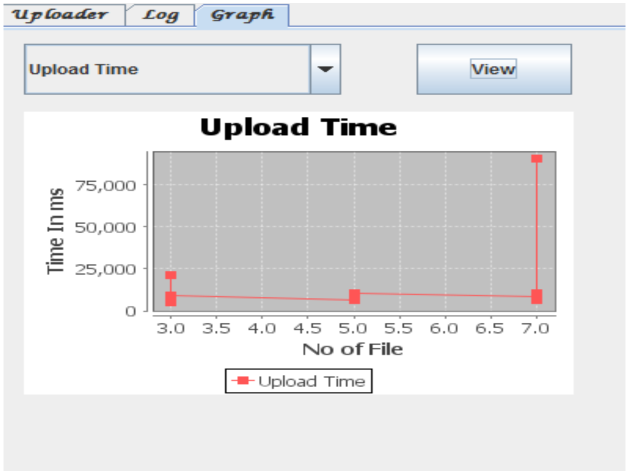

Figure 2. Graphic indicating time to upload the files into the cloud

The graph below indicates the time to calculate the index for the same set of 100 files using Homomorphic encryption by the administrator.

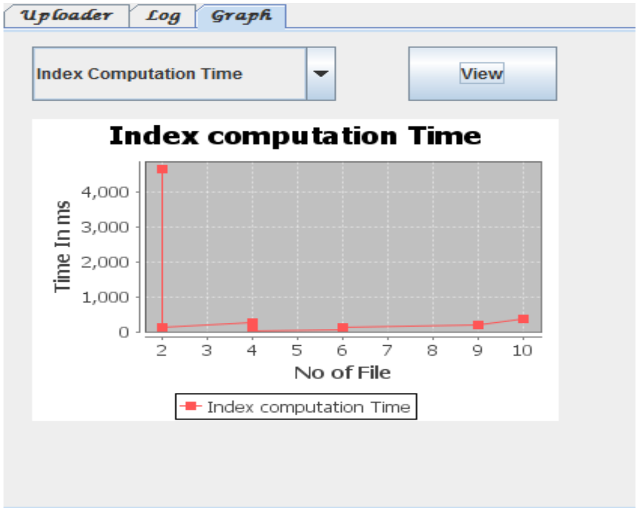

Figure 3. Graph indicating time to construct the index from the file set

In the following example, sample of 400 files are collected. Our proposed system is efficient with respect to time to retrieve the top-n files matching the keyword comparatively with SSE scheme. This is illustrated with a graph below.

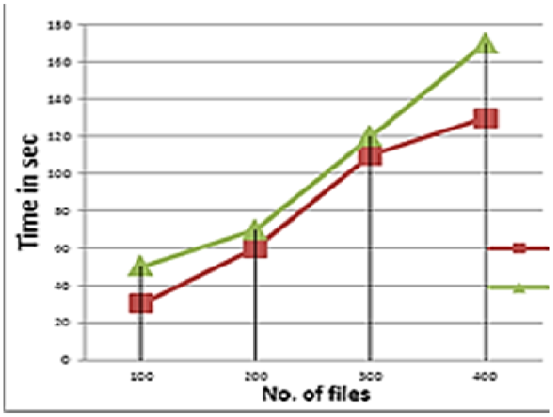

Figure 4. Graph indicating data retrieval using TRSE scheme and SSE scheme

7. Conclusion

The proposed system makes the system highly scalable and minimizes information leakage. Prevents overloads by ranking the files at the user side, reducing bandwidth and protects document frequency. The proposed solution is secure, scalable and accurate compared to the other ranked keyword search.

REFERENCES

[1] D.Nurmi, R.Wolski, C.Grzegorczyk, G.Obertelli,S.Soman, L.Youseff and D.Zagorodnov, "The eucalyptus open-source cloud- computing system," CCGRID 20009.9th IEEE/ACM International Symposium, 2009.

[2] S.S and A. Basu, "Performance of eucalyptus and open stack clouds on future grid,"International Journal of Computer Applications, vol. 80,no.13,pp.31-37, 2013.

[3] Z.Pantić and M. A.Babar, "Guidelines for Building a Private Cloud Infrastructure,“ IT University of Copenhagen, Denmark, Copenhagen, Denmark,2012.

[4] B. Beal, "Public vs. private cloud applications: two critical differences,"23May2012. [Online]. Available:http://searchcloudapplications.techtarget.com/feature/Public-vs-private-cloud-applications-Two-critical-differences.

[5] Tarik Moataz, Abdullatif Shikfa, "Boolean symmetric searchable encryption," ASIA CCS '13 Proc. of the 8th ACM SIGSAC symposium on Information computer and communications security, .pp. 265-276, NY, USA , 2013.

[6] R. Curtmola, J.A. Garay, S. Kamara, and R. Ostrovsky, "SearchableSymmetric Encryption: Improved Definitions and Efficient Constructions,"Proc. ACM 13th Conf. Computer and Comm. Security (CCS), 2006

[7] D. Song, D. Wagner, and A. Perrig, "Practical Techniques for Searches on Encrypted Data," Proc. IEEE Symp. Security and Privacy, 2000.

[8] 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.

[9] S.Adhikari,G.Bunce,W.Chan,A.Chandramouly,D.Kamhout, B.McGeough,J.JonSlusser,C.Spence and B. Sunderland, "Best practices for building and enterprise private cloud," Intel IT Centre,2011.

[10] B.Adler,"Designing Private and hybrid clouds: architectural best practices," RightScaleInc.,2012.

[11] "Planning Guide: Virtualisation and cloud computing," Intel IT Centre,2013.

[12] Y.Wadia,"TheEucalyptusOpenSourcePrivateCloud,"[Online]Available:http://www.cloudbook.net /resources/stories/the-eucalyptus-open-source-private-cloud.

[13] G.VonLaszewski,J.Diaz, F.WangandG.Fox, "Comparison of multiple cloud frameworks," IEEE on Cloud computing(CLOUD),vol.734,no.741,pp.24-29,2012,5th International Conference

[14] F. Bao, R. Deng, X. Ding, and Y. Yang, "Private query on encrypted data in multi-user settings," in Proc. of ISPEC 2008.

[15] A. Swaminathan, Y. Mao, G.-M. Su, H. Gou, A.L. Varna, S. He, M. 5u, and D.W. Oard, "Confidentiality-Preserving Rank-Ordered Search," Proc. Workshop Storage Security and Survivability, 2007.

[16] Cong Wang, Ning Cao, Jin Li, Kui Ren, Wenjing Lou, "Secure ranked keyword search over encrypted cloud data," IEEE 2010 30th International Conference 2010.

[17] S. Zerr, D. Olmedilla, W. Nejdl, and W. Siberski, "Zerber+r: Top-k Retrieval from a Confidential Index," Proc. 12th Int'l Conf.Extending Database Technology: Advances in Database Technology (EDBT), 2009.

[18] P. Golle, J. Staddon, and B. Waters, "Secure Conjunctive Keyword Search over Encrypted Data,"

[19] L. Ballard, S. Kamara, and F. Monrose, "Achieving Efficient Conjunctive Keyword Searches over Encrypted Data,

[20] Jin Li, Qian Wang, Cong Wang, Ning Cao, Kui Ren† , and Wenjing Lou‡"Fuzzy Keyword Search over Encrypted Data in Cloud Computing".

[21] Ning Cao, Cong Wang , Li, Ming , Kui Ren, Wenjing Lou, "Privacy preserving multi-keyword ranked search over encrypted cloud data," INFOCOM, 2011 Proceedings IEEE April 2011.

[22] H. Hu, J. Xu, C. Ren, and B. Choi, "Processing Private Queries over Untrusted Data Cloud through Privacy Homomorphism,"Proc. IEEE 27th Int'l Conf. Data Eng. (ICDE), 2011.