www.ijecs.in
International Journal Of Engineering And Computer Science ISSN: 2319-7242
Volume 5 Issues 7 July 2016, Page No. 17273-17277

Securing Privacy over Encrypted Cloud Data Using Multiple Keywords Rank Based Search

Parasa Srinivas Manikanta[1], V. Baby[2]

1. PG Student, Department of Computer Science and Engineering, VNR Vignan Jyothi Institute of Engineering and Technology, Bachupally, Hyderabad- 500 090
2. Associate Professor, Department of Computer Science and Engineering, VNR Vignan Jyothi Institute of Engineering and Technology, Bachupally, Hyderabad- 500 090

Abstract: Due to the existence of cloud computing , owners of that particular data had determined to push their complex data systems from local database systems to the public cloud which is commercial for better flexibility and for economical preservation. In order to protect privacy of data, the data that is sensitive need to be encrypted before deploying the data onto the cloud, however, it is very different from the plaintext keyword match which is a old-fashioned technique. Therefore, the searching of data on encrypted data is the prime importance. There are many data users who wish to search a file, based on their desired keywords, the file must be extracted matching the relevance of that user entered keywords. In this paper, we are designing and solving the provocation problem of securing Privacy over Encrypted cloud-data using Multiple keywords Rank based Search (PEMRS). We initiate a set of strong privacy essentials for a secure cloud-data using system.

Keywords: Cloud computing, Multiple keywords, Rank, File.

1. INTRODUCTION:

Cloud computing is the contemplate vision of computing as a practicality, where cloud consumers can remotely store data over cloud so as to rejoice the on-demand high quality services and applications from a distributed pool of configurable computing services. It's better flexibility and for economical preservation, are encouraging both individuals and firms to push their complex data systems from local database systems to the public cloud. To preserve data privacy and to tackle unbidden accesses in the cloud, the data which is sensitive, ex: E-mails, financial transactions, tax documents, etc., needed to be encrypted by the owners of that data, before pushing the data onto the commercial public cloud. However, it is very different from the plaintext keyword match which is a old-fashioned technique. Downloading the entire data and locally decrypting it is unsuitable. Additionally, removal of local data storage systems and storing the data over cloud serves no purpose if the data retrieval and searching is not efficient. Therefore, the searching of data on encrypted data is the prime importance. Taking into consideration many data users who wish to search a file, based on their desired keywords, the file must be extracted matching the relevance of that user entered keywords. It is also very much difficult to encounter the requirements of scalability, system usability and performance. On the one hand, to encounter the efficient and effective data retrieval, the documents enforce the cloud server to implement result relevance ranking, rather returning undifferentiated outputs. Using ranked based searching systems enables the users to search the most appropriate data quickly and easily. Ranked based searching, can distinguishably abolish un-wanted network traffic by acknowledging only the most appropriate data, which is sensible in the PAY AS YOU USE paradigm. For privacy preserving, the rank based search, nevertheless should not dribble keyword relevant information. On the other hand, to enhance the search output accuracy, as well as to ameliorate the searching experience for the user, it is essential for such a rank based searching system to reinforce multiple keywords searching. As a frequent practice, data users may tend to give a set of keywords instead of only one keyword to retrieve the most appropriate data and every keyword in the search query helps to access the suitable data. "Co-ordinate matching" does as many resemblances as possible, which is an efficient similarity measure. Furthermore use "Inner Product Similarity" to significantly measure such a similarity measure. Nevertheless, to apply in encrypted cloud-data search persist a very invocation task because of intrinsic security, privacy obstacles, and of rigid keyword privacy, index privacy and data privacy.

In the literature, inquest encryption [7], [8], [9], [10],[11], [12], [13], [14], [15] is a helpful method which treats data which is encrypted as documents and permits a user to invulnerably search by a single keyword and extract documents of wish. Although, direct implementation of these techniques to preserve large scale cloud data utilization system won't be appropriately suitable, since they are expanded as Crypto primitives and cannot aid such top service-level requirements like user searching experience, system usability and simple information finding. The way to design a systematic encrypted data search mechanism which reinforce multiple keywords semantics without privacy contravention still remains a provocation trouble.

In this paper, we are designing and solving the provocation problem of securing Privacy over Encrypted

cloud-data using multiple keywords Rank based Search (PEMRS). We initiate a set of rigid privacy essentials for a secure cloud-data usage system. Among varied multiple keywords semantics, we select the effective similarity measure of "Co-ordinate matching" does as many resemblances as possible, which is the efficient similarity measure. Furthermore use "Inner Product Similarity" to notably calculate such a similarity measure. Throughout the index construction, every document is corresponded with a binary vector as a sub-index where every bit constitute whether associated keyword is included in the document. Even the search query is reported as a binary vector, in which every bit means whether associated keyword arrives in the search request, hence the similarity could be accurately measured by the inner product of the query vector with data vector. Nevertheless, instantly pushing the data vector or the query vector will contravene the search privacy or the index privacy. To meet the provocation of aiding such multiple keywords semantic without privacy contravention, we propose a fundamental idea for the PEMRS using reliable inner product reckoning, which is adapted from a reliable k-nearest neighbour (KNN) methodology [16], and then give two significantly improved PEMRS schemes in a step-by-step procedure to attain various rigid privacy requirements. Our contributions are summarized as follows:

1. For the first time, we explore the problem of multiple keyword rank based search over encrypted cloud data, and establish a set of rigid privacy requirements for such a secure cloud-data utilization system.
2. We suggest two PEMRS schemes based on the resemblance measure of "coordinate matching" by meeting different privacy.
3. We explore some further modification of our ranked search mechanism to support more search semantics and dynamic data operations.
4. Comprehensive analysis and explore efficiency and privacy assurance of the proposed schemes is given, and experiments on the real-world data set indeed introduce reduced overhead of computation.

.

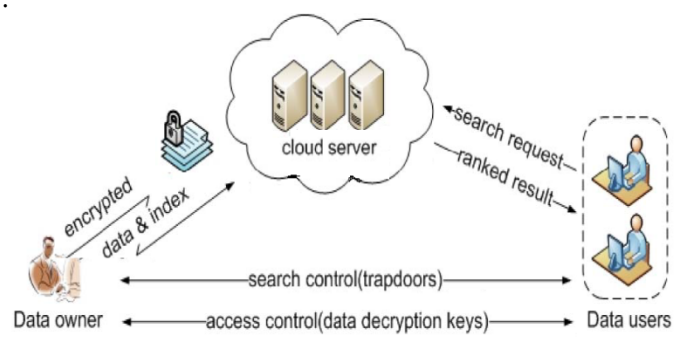

Fig1: Architecture of the searching over encrypted cloud-data

This paper is organized as follows: In Section 2, we instigate the system model, the threat model, and design goals. Section 3, outlines the PEMRS framework and privacy requirements, which is followed by Section 4, which describes the proposed schemes. Section 5, presents clone results, which is followed by section 6, which presents the results and in section 7, conclude the paper.

2. PROBLEM FORMULATION:

2.1 System Model:

Consider a cloud-data hosting service including three different entities, as illustrated in Fig. 1: data owner, data user, and the cloud server. The data owner has a gathering of data documents F to be pushed over the cloud server in the encrypted form C. To enable the searching ability over C for efficient data usage, the data owner, before pushing, must primarily build an encrypted searchable index I from F, and then only push both the index I and the encrypted document collection C onto the cloud server. To search a document collection for t given keywords, an authorized user obtains a corresponding trapdoor T through search control procedures, for example, broadcast encryption [10]. On receiving T from a data user, the cloud server is accountable to search the index I and return the associated set of encrypted documents. To enhance the document retrieval correctness, the search result must be ranked by the cloud server according to some ranking criteria (e.g., coordinate matching). Likewise, to decrease the communication cost, the data user may send an optional number k with the trapdoor T so that the cloud server only sends back top-k documents that are most appropriate to the search query.

2.2 Threat Model:

The cloud server is contemplated as "honest-but-curious" in our model, which is steady. Clearly, the cloud server acts in an "honest" fashion and exactly follows the designated protocol specification. However, it is "curious" to deduce and inspect data (including index) in its storage received during the protocol so as to learn more information. Based on what information the cloud server knows, we review two threat models with different attack abilities as follows. Known as ciphertext model. In this model, the cloud server is expected to only know encrypted data set C and searchable index I, both of which are pushed from the data owner. Known background model. In this stronger model, the cloud server is expected to have more knowledge than what can be acquired in the known ciphertext model. Such information may involve the correlation relationship of given search queries (trapdoors), as well as the data set related statistical information.

2.3 Design Goals:

To authorize ranked search for effective usage of pushed cloud-data under the before mentioned model, our system design should simultaneously attain performance and security guarantees as follows.

Multiple keyword rank based search. To design search strategy which allow multi-keyword query and provide result similarity ranking for efficient data retrieval, instead of returning unvaried results.

Privacy-preserving. To prevent the cloud server from knowing additional information from the data set and the index, and to encounter privacy requirements.
Efficiency. Above goals on functionality and privacy must be attained with decreased communication and computation overhead.

3. FRAMEWORK AND PRIVACY REQUIREMENTS FOR PEMRS

In this section, we define the framework of Privacy over Encrypted cloud-data using Multiple keywords Rank based Search (PEMRS) and establish various rigid system wise privacy requirements for a secure cloud data usage system.

3.1 PEMRS Framework

For easy demonstration, operations on the data documents are not displayed in the framework since the data owner can certainly employ the traditional symmetric key cryptography scheme to encrypt and then push data. With focus on the index and query, system consists of four algorithms as follows:

Setup (1^l). Taking a security parameter l as input, the data owner outputs a symmetric key as SK.

BuildIndex (F, SK). Based on the data set F, the data owner builds a searchable index I which is encrypted by the symmetric key SK and then only it is pushed onto the cloud server. After the index creation, the document collection can be independently encrypted and outsourced

Trapdoor (W). With t keywords of interest in W as input, this algorithm generates a corresponding trapdoor T_w

Query (T_w, k, I). When the cloud server receives a query request as (T_w, k), it performs the rank based search on the index I using the help of trapdoor T_w, and finally returns F_w, the ranked id list of top-k documents sorted by their similarity with W.

3.2 Privacy Requirements for PEMRS

The privacy is ensured in the literature, through searchable encryption, represents that the server should grasp nothing but the search results. With this well-established privacy description, we investigate and launch a new set of rigid privacy requirements specifically for the PEMRS framework.

As per the data privacy, the owner of the data can resort to the conventional symmetric key cryptography technique, to encrypt the data before pushing onto the cloud server, and should successfully block the cloud server from prying into the pushed data. According to the index privacy, if the cloud server infer any corresponding keywords and encrypted documents from index, it may grasp the major idea of a document, even if the content is short in a document. Accordingly, the searchable index must be developed to block the cloud server from doing such type of association attack. While data and index privacy ensures the demanded task, many search privacy requirements employed in the query method are more difficult and complex to handle as follows.

Keyword privacy. Users generally prefer to keep their search from being revealed to others like the cloud server, the most principal concern is to obscure what they are searching, i.e., the keywords designated by the associated trapdoor. However, the trapdoor can be generated in a cryptographic way to secure the query keywords, the cloud server could perform some statistical analysis on the search result to make an analysis. When the cloud server knows some background information of the data, the keyword specific information may be used to reverse engineer the keyword.

Trapdoor unlinking. The trapdoor generating function should be randomized instead of being deterministic. Particularly, the cloud server should not be able to infer the relationship of any given trapdoors, For example, to confirm whether the two trapdoors are organised by the similar search request. In other case, the deterministic trapdoor generation would give the cloud server advantage to gather frequencies of different search requests about different keyword(s), which may further contravene the before mentioned keyword privacy requirement. Hence, the fundamental protection for trapdoor unlinking is to introduce enough non-determinacy of the trapdoor generation policy.

Access pattern. Within the rank based search, the access pattern is the succession of search results where each search result is a set of documents with rank order.

4. PRIVACY PRESERVING AND EFFICIENT PEMRS

To effectively attain multiple keyword rank based search, we suggest to consider “inner product similarity” [6] which quantitatively assess the effective similarity measure “coordinate matching.”

4.1 PEMRS_I: Privacy-Preserving Strategy in Known Ciphertext Method

The inner product computation strategy is not suitably enough for our PEMRS design because randomness included is the scale factor r in the trapdoor generation, which does not impart enough non-determinacy in the overall strategy as essential by the trapdoor, keyword privacy requirement, the unlinking requirement. To provide a more enhanced design for the PEMRS, we now provide PEMRS_I strategy as follows.
As indicated in the keyword privacy requirement, randomness must also be clearly calibrated in the search result to obscure the document frequency and to decrease the chances for re-identification of keywords. Introducing such randomness in the similarity score is an efficient way that what we expect. More precisely, unlike the randomness employed in query vector, we insert a dummy keyword into each data vector and allocate a random value to it. Each individual vector Di is expanded to (n+2) dimension instead of (n+1), where a random variable "i representing the dummy keyword is saved in the expanded dimension.

4.2 Supporting Data Dynamics:

After the data set is pushed onto the cloud server, it may be improved in addition to being accessed. Along with the updating process on data documents, reinforcing the score dynamics in the searchable index is very important. When we consider three major data operations such as inserting new documents, updating existing documents, and

deleting existing documents, the associated operations on the searchable index involves generating new index, updating existing index, and deleting existing index.

For inserting new documents in the data set, there might be few new keywords in new documents which are essential to be inserted in the dictionary W. Remember that each subindex in our scheme has fixed dimension which is same as the number of keywords in the old dictionary, so the simple solution is to retrieve every subindexes from the cloud server, afterwards decrypt, rebuild, and encrypt them before pushing it to the cloud server. Still, this proposal introduces much cost on communication and computation for both sides which is inappropriate in the "pay-as-you-use" cloud paradigm. To decrease such high cost, we preserve some empty entries in the dictionary and set associated entries in the data vector as 0. If the dictionary needs to index new keywords, at that instance of inserting new documents, we just replace the empty entries in the dictionary by new keywords, and produce subindexes for new documents based on the updated dictionary. The remaining documents and their subindexes are stored on the cloud server and are not affected, therefore remain as same before

When existing documents are changed, associated subindexes are also retrieved from the cloud server and are updated in terms of term frequency before pushing. If new keywords are instantiated during the modification operation, we use the same method which is suggested in the previous insertion operation. As a special case of enhancement, the operation of deleting existing documents introduce less communication and computation cost because it only needs to update the document frequency of all the keywords contained by these documents.

5. PERFORMANCE ANALYSIS

In this section, we randomly choose varied number of e-mails to build data set. The performance of our technique is evaluated regarding the efficiency of proposed PEMRS scheme, as well as the tradeoff between search privacy and precision.

5.1 Privacy and Precision

As discussed in Section 4, dummy keywords are placed into each data vector and few of them are selected in every query. Hence, similarity scores of documents will not be exactly accurate. In other words, the cloud server recur top-k documents based on similarity scores of data vectors to query vector, some of the factual top-k relevant documents for the query may be eliminated. This is because either their authenticate similarity scores are reduced or the similarity scores of few documents out of the factual top-k are increased, both of which are due to the effect of dummy keywords inserted into data vectors. To assess the purity of the k documents retrieved by user, we estimate as precision $P_k=k^J/k$ where k^J is number of real top-k documents that are returned by the cloud server.

6. RESULT

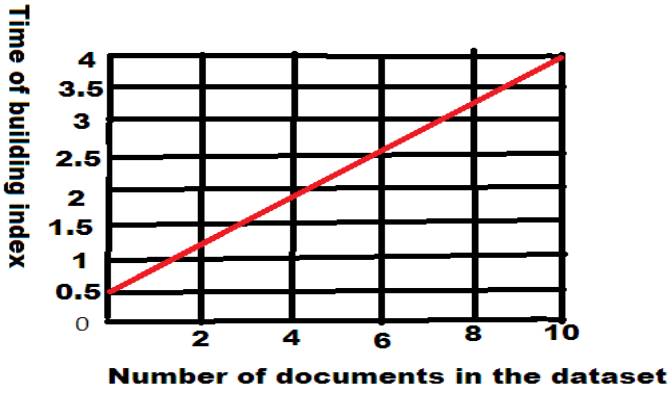

Fig.2. Time cost of query. For the same query keywords in different sizes of data set

7. CONCLUSION

In this paper, we define and solve the problem of multiple keywords based rank search over encrypted cloud data, and construct a variety of privacy requirements. Among different multi-keyword semantics, we select the effective similarity measure of "coordinate matching," i.e., to check many matches as feasible, to efficiently represent the relevance of pushed documents to the queried keywords, and use "inner product similarity" to quantitatively cakulate such similarity measure. For meeting the provocation of supporting multiple keywords semantic without privacy contravention, we suggest a basic idea of PEMRS using secure inner product computation. Then, we give two improved PEMRS schemes to attain various rigid privacy requirements in two distinct threat models. We also investigate few further modifications of our ranked search mechanism, including more search semantics. Thorough study and investigating privacy, efficiency is ensured. Experiments on the real-world data set exhibit our recommended scheme introduce lower overhead on both communication and computation..

In our future work, we will explore validating the integrity of the rank order in the search output and by using higher level of encryption standards so as to preserve the data when it is been pushed onto the cloud.

REFERENCES:

[1] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-Preserving Multi-Keyword Ranked Search over Encrypted Cloud Data," Proc. IEEE INFOCOM, pp. 829-837, Apr, 2011.

[2] 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.

[3] N. Cao, S. Yu, Z. Yang, W. Lou, and Y. Hou, "LT Codes-Based Secure and Reliable Cloud Storage Service," Proc. IEEE INFOCOM, pp. 693-701, 2012.

[4] S. Kamara and K. Lauter, "Cryptographic Cloud Storage," Proc.14th Int'l Conf. Financial Cryptograpy and Data Security, Jan. 2010.

[5] A. Singhal, "Modern Information Retrieval: A Brief Overview," IEEE Data Eng. Bull., vol. 24, no. 4, pp. 35-43, Mar. 2001.

[6] I.H. Witten, A. Moffat, and T.C. Bell, Managing Gigabytes:Compressing and Indexing Documents and Images. Morgan Kaufmann Publishing, May 1999.

[7] D. Song, D. Wagner, and A. Perrig, "Practical Techniques for Searches on Encrypted Data," Proc. IEEE Symp. Security and Privacy, 2000.
[8] E.-J. Goh, "Secure Indexes," Cryptology ePrint Archive, http://eprint.iacr.org/2003/216. 2003.
[9] Y.-C. Chang and M. Mitzenmacher, "Privacy Preserving Keyword Searches on Remote Encrypted Data," Proc. Third Int'l Conf. Applied Cryptography and Network Security, 2005.
[10] R. Curtmola, J.A. Garay, S. Kamara, and R. Ostrovsky, "Searchable Symmetric Encryption: Improved Definitions and Efficient Constructions," Proc. 13th ACM Conf. Computer and Comm. Security (CCS '06), 2006.
[11] D. Boneh, G.D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public Key Encryption with Keyword Search," Proc. Int'l Conf. Theory and Applications of Cryptographic Techniques (EUROCRYPT), 2004.
[12] M. Bellare, A. Boldyreva, and A. ONeill, "Deterministic and Efficiently Searchable Encryption," Proc. 27th Ann. Int'l Cryptology Conf. Advances in Cryptology (CRYPTO '07), 2007.
[13] M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J.Malone-Lee, G.Neven, P. Paillier, and H. Shi, "Searchable Encryption Revisited: Consistency Properties, Relation to AnonymousIbe, and Extensions," J. Cryptology, vol. 21, no. 3, pp. 350-391,2008.
[14] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy Keyword Search Over Encrypted Data in Cloud Computing,"Proc. IEEE INFOCOM, Mar. 2010.
[15] D. Boneh, E. Kushilevitz, R. Ostrovsky, and W.E.S. III, "Public Key Encryption That Allows PIR Queries," Proc. 27th Ann. Int'lCryptology Conf. Advances in Cryptology (CRYPTO '07), 2007.
[16] W.K. Wong, D.W. Cheung, B. Kao, and N. Mamoulis, "Secure KNN Computation on Encrypted Databases," Proc. 35th ACM SIGMOD Int'l Conf. Management of Data (SIGMOD), pp. 139-152, 2009.