International Journal of Computer Science
Oddity...Probe...Reviste...

http://www.ijcsjournal.com Volume 2, Issue 1, No 1, 2014. ISSN: 2348-6600
Reference ID: IJCS-032 PAGE NO: 172-175

AN EFFECTIVE SECURE KEYWORD SEARCH OVER ENCRYPTED OUTSOURCED CLOUD STORAGE

Gopinath Ganapathy[1], A.Kavitha[2]

[1]Professor & Head, [2]Student of M.Tech

School of Computer Science, Engineering and Applications,
Bharathidasan University,
Trichy-23, Tamilnadu, India.
gopinath.g@csbdu.in, *bdukavi.cs@gmail.com.*

Abstract **- As Cloud Computing becomes widespread, more and more sensitive data are being centralized into the cloud. Although normal searchable encryption schemes permit a user to securely search over encrypted data using keywords and retrieve files of its interest, these techniques support only perfect match keyword search. This work for the first time to formalizes and solves the problem of "Effective Keyword Search over Encrypted Cloud data". It maintains privacy and security of keyword more than Effective Search. The keyword search greatly increase system usability by returning list of matching files when user's searching inputs exactly match the predefined keywords or the closest possible matching files based on keyword similarity. In the proposed solution, the ranked search mechanism is constructing keyword sets based on two advanced techniques, which gives solution for optimized storage and representation overheads. Further a relevance score is built to form a searchable index and transformed from the resulted keyword sets. The data owner save their information with security and share the files into authenticate person with different level access permission.**

Index Terms: **Ranked search, searchable encryption, Cloud computing.**

I INTRODUCTION

Cloud computing is computing which delivers computing resources over the internet based on pay-for-use. It could run a program or application on many connected computers at the same time. This service may be appear in real server hardware, and are in fact served up by virtual hardware. This technology can be simulated by software running on single or multiple real machines. Cloud computing achieves economies of scale, similar to a utility over a network, converged infrastructure and shared services and focus on maximize effectiveness of the shared resources. The Cloud resources are dynamically re-allocated per demand. In Ranked search increases system reliability by returning the matching files into ranked order regarding to certain relevant progress, thus making forward closer to privacy-preserving data hosting services the context of Cloud computing.

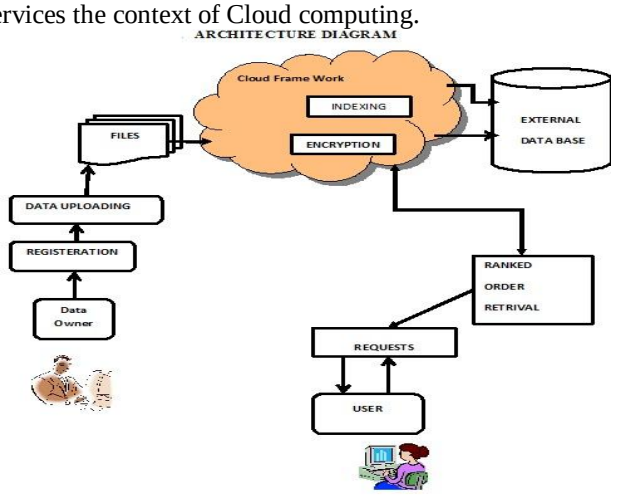

Fig 1:Architecture for keyword search over encrypted out sourced cloud storage.

II RANKED SEARCHABLE SYMMETRIC ENCRYPTION SCHEME

Published by SK Research Group of Companies (SKRGC).

International Journal of Computer Science
Oddity...Probe...Reviste...

http://www.ijcsjournal.com Volume 2, Issue 1, No 1, 2014. ISSN: 2348-6600
Reference ID: IJCS-032 PAGE NO: 172-175

A ranked searchable encryption scheme consists of four algorithms (KeyGen, BuildIndex, TrapdoorGen,SearchIndex). Our 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 Build Index 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.

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.

We now introduce some important information retrieval background for our proposed scheme:
Inverted Index
It stores a list of mappings from keywords to the corresponding set of files that contains this keyword, allowing full text search.
Ranking Function
A ranking function is used to calculate relevance scores of matching files to a given search request.
Ranking Function= (TF * IDF)
TF - Term Frequency
IDF - Inverse Document Frequency
TF - Number of time occurs in particular file
IDF - Whole number of files in particular collection is divided by number of files containing the term.

2.1 Notation and Preliminaries

- **C** – the file collection to be outsourced, denoted as a set of n data files C= (F_1,F_2…….F_n).
- **W** – the distinct keywords extracted from file collection C, denoted as a set of m words W =(w_1,w_2…..w_m).
- **id(F_j)** – the identifier of file F_j that can help uniquely locate the actual file.
- **I** – the index built from the file collection, including a set of posting lists { $I(w_i)$}, as introduced below.
- **T_{wi}** – the trapdoor generated by a user as a search request of keyword w_i.
- **$F_{(wi)}$** – the set of identifiers of files in C that contain keyword w_i.
- **N_i** – the number of files containing the keyword w_i and N_i =| $F(w_i)$|.
- **R** – Bucket Range.

2.2 Algorithm

1. Build Index
   R=(C_1,C_2….C_n)
   C=(F_1,F_2……..F_n)
2. Searchable Index(I,T_{wi})
   I=(id(F_j),ScoreValue,Keyword)
   Score Value=($F_{(wi)}$ * (R/N_i))
   N_i=| $F_{(wi)}$|
3. Key Generate K
   K= (id(F_j),Score Value)
4. Trapdoor Generator
   T_{wi} =(id(F_j),Intrested Keyword)
5. File Retrieve
   Rt = $T_{wi} \leq I \leq R_i \leq C_i$, Rt=$F_i$

III MODULES

CLOUD FRAMEWORK
Here the cloud servers and cloud service provider are setup for use. Each server may have single or multiple service providers. Customers can be unsophisticated and terribly unreliable when it comes to preserving data. For example, they often accidentally or unknowingly lose data, e.g. by forgetting to back it up or by mistakenly overwriting or erasing files. Another common premise is that customers can maintain secret keys for the long term. Thus, we assume that we cannot rely on customers to maintain any long-term state, such as secret keys or the hashes needed for auditing or extraction. Although customers might not keep long-term secrets associated with the stored data, they still can use other

Published by SK Research Group of Companies (SKRGC).

International Journal of Computer Science
Oddity...Probe...Reviste...
http://www.ijcsjournal.com Volume 2, Issue 1, No 1, 2014. ISSN: 2348-6600
Reference ID: IJCS-032 PAGE NO: 172-175

methods to identify themselves to storage providers. Customers or the cloud platforms require users to authenticate themselves over the phone, mail, or through email to renew "short-term" passwords or identity tokens. The following are two additional important assumptions about customers that shape our protocols.

DATA UPLOADING

In this module the data owner uploads files to the cloud after registration. It is these files which are encrypted and stored in the outsourced cloud server. First, customers have an incentive to claim data loss fraudulently, for instance, to receive payment for losing data as specified by the retention contract. Second, customers want to keep their data private from third parties. So the customers or cloud users are clearly provided access resources and storage options here with the data they own and who or how they can see it. This causes the server to index, encrypt and store the content for the different data owners.

RANKED SEARCH ENCRYPTION SCHEME

The user gets the ranked results without giving permission to cloud server cannot learn any additional information more than the access pattern. In previously every user should wait for two round-trip times for each search request, it makes time loss or he may even lose the performance top-k retrieval, resulting unwanted communication overhead. These demerits will direct to our main result. Searchable symmetric encryption satisfies the non-adaptive security definition. The above scheme clearly satisfies the security guarantee of SSE, i.e., only the access pattern and search pattern are leaked. However, the ranking is done on the user side, which may bring in huge post processing overhead. Moreover, sending back all the files consumes large undesirable bandwidth. One possible way to reduce the communication overhead is that server first sends back all the valid entries. As the size of valid entries is far less than the corresponding files, significant amount of bandwidth is expected to be saved, as long as user does not retrieve all the matching files. Also note that in this way, server still learns nothing about the value of relevance scores, but it knows the requested files are more relevant than the unrequested ones, which inevitably leaks more information than the access pattern and search pattern. There are several methods available to measure the characters, string, words similarity. In this paper, we take distance between two words. The distance function ed(w1,w2) is the number of operations needed to transform one form into the other form. The three major operations are 1) Substitution: substituting one character to another in a word; 2) Deletion: removing one character from a word; 3) Insertion: adding a single character into a word. To build an index for, the data owner computes trapdoors with a secret key. The data owner also encrypts in the index table and encrypted data files are stored to the outsourced cloud server. The search with w, the authorized user manipulate the trapdoor Tw of w and sends it to the server, then server receiving the search request Tw, the server match it with the table of index and returns all the possible encrypted data identifiers according to the keyword definition. The user decrypts the returned data and retrieves related data of interest. We now given an enhanced technique to increase the straightforward approach for building the index set.

ONE-TO-MANY ORDER-PRESERVING MAPPING

One-to-many OPSE scheme increases relevance score randomness, and reduce the amount of information leakage, still maintain the plaintext order. The encryption process of OPSE has a domain *D* and plaintext *m*. Domain always mapped to the same random-sized bucket in range R, where buckets should possess non-overlapping interval. A ciphertext *c* is then selected within the bucket range by using random selection function. The proposed one-to-many order-preserving mapping employs the random plaintext-to-bucket mapping of OPSE, but incorporates the unique file IDs together with the plaintext m as the random seed in the final ciphertext chosen process. Due to the use of unique file ID as part of random selection seed, the same plaintext m will no longer be deterministically assigned to the same ciphertext *c*, but instead a random value within the randomly assigned bucket in range R.

KEYWORD SEARCH

The above straightforward approach demonstrates the core problem that causes the inefficiency of ranked searchable encryption. That is how to let server quickly perform the ranking without actually knowing the relevance scores. To effectively support ranked search over encrypted file collection, we now resort to the newly developed cryptographic primitive–order preserving symmetric encryption (OPSE) to achieve more practical performance. Note that by resorting to OPSE, the security guarantee of RSSE is inherently weakened compared to SSE, as we now let server know the relevance order. However, this is the information we want to tradeoff for efficient RSSE. Then it is shown that it can be adapted to suit the purpose for ranked searchable encryption with an "as-strong-as-possible" security guarantee. Thus the keyword search supports score dynamics

Published by SK Research Group of Companies (SKRGC).

International Journal of Computer Science
Oddity...Probe...Reviste...
http://www.ijcsjournal.com Volume 2, Issue 1, No 1, 2014. ISSN: 2348-6600
Reference ID: IJCS-032 PAGE NO: 172-175

is also the reason why they do not use the naive approach for RSSE, where data owner arranges file IDs in the posting list according to relevance score before outsourcing. As whenever the file collection changes, the whole process, including the score calculation, would need to be repeated, rendering it impractical in case of 93 frequent file collection updates. In fact, supporting score dynamics will save quite a lot of computation overhead during the index update, and can be considered as a significant advantage compared.

IMPLEMENTATION

Ranked searching process greatly increases system usability by returning exact matching files in a ranked order based on certain relevance keyword frequency; it is given one step

forward in privacy-preserving data hosting services in Cloud Computing. Our design goals are to achieve system security and usability. Specifically, we inquire the statistical measurement from IR and text mining to embed with relevance score. When keyword privacy will leaks relevance score is directly outsourced, so we protect those sensitive weight information to integrate order-preserving symmetric encryption(OPSE) and one-to-many order-preserving mapping technique, for efficient ranked search functionalities.

IV CONCLUSION

Our ranked search mechanism, highly support for relevance score dynamics, authentication of particular user for ranked search results. From the security analysis, the proposed solution is more secure and privacy, the data owner save, search their information with security and share the files into authenticate person with different level access permission. Now we achieve the goal of ranked keyword search.

V REFERENCES

[1]. Cong Wang, Ning Cao,Kui Ren, Weijing Lou " Enabling Secure and Efficient Ranked Keyword Search over Outsourced Cloud Data" - *2012.*

[2]. 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.*

[3]. J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy keyword search over encrypted data in cloud computing " inProc.of IEEE INFOCOM'10 Mini-Conference, 2010.

[4]. N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-preserving multi-keyword ranked search over encrypted cloud data," in Proc.of INFOCOM'11, 2011.

Published by SK Research Group of Companies (SKRGC).