Multi Keyword Searching Techniques over Encrypted Cloud Data

P. Shanmuga Priya[1], R. Sugumar[2]

[1]Research Scholar, Computer Science and Engineering, St. Peter's University, Chennai, India

[2]Associate Professor, Computer Science and Engineering, Veltech Multitech Engineering College, Chennai, India

Abstract: ***Cloud computing is the long dreamed vision of computing as a utility, where cloud customers can remotely store their data into the cloud so as to enjoy the on-demand high quality applications and services from a shared pool of configurable computing resources. However, to protect data privacy, sensitive cloud data has to be encrypted before outsourced to the commercial public cloud, which makes effective data utilization service a very challenging task. In this paper, we focus on addressing data privacy issues using SSE. It observes that server-side ranking based on order-preserving encryption (OPE) inevitably leaks data privacy. To avoid leakage, TRSE scheme that supports top-k multi-keyword retrieval. In TRSE, vector space model and fully homomorphic encryption (FHE) are employed. The vector space model helps to provide sufficient search accuracy, and the Fully Homomorphic encryption is a special type of encryption system that permits arbitrarily complex computation on encrypted data. It enables users to involve in the ranking while the majority of computing work is done on the server side by operations only on cipher text. It provides high security and efficiency.***

Keywords: Cloud, data privacy, ranking, similarity relevance, fully homomorphic encryption, vector space model

1. Introduction

Computers are needed everywhere, work, research, games. As the uses of computers are increasing day by day, the resources are also increasing at an alarming rate. Cloud computing can be stated as shift of work and resources from personal computers or individual enterprise applications to set a cloud of computers. Cloud Computing offers solutions to many problems like hardware, machine failures etc. The great advantage of cloud computing is ―elasticity‖: the ability to add capacity or applications almost at a moment's notice. The pay-as-you-go approach appeals to small- and medium-sized enterprises; since the vendor has many customers, it can lower the per-unit cost to each customer. Larger companies find it easier to manage collaborations in the cloud. Though cloud computing is highly flexible and cost effective, it introduces and lot of security issues. Some other advantages of cloud can be stated as ubiquitous network access, rapid resource elasticity, usage based pricing and transference of risk.

As cloud computing is becoming more accrual, more information is being centralized into the cloud. Data owners are relieved from the burden of data storage and maintenance, to enjoy the on demand high quality data service. The customers of the cloud, now has to be secure against the cloud service providers itself, because they can leak information to prohibited entities or get hacked.

Public cloud computing represents a significant paradigm shift from the conventional norms of an organizational data center to a deperimeterized transportation open to use by potential adversaries. As with any emerging information technology area, cloud computing should be approached watchfully with due kindness to the sensitivity of data. To avoid these kinds of attack, the sensitive data should be encrypted, before it is outsourced which obsoletes traditional data utilization based on plaintext keyword search. Thus, enabling an encrypted cloud data search service is of paramount importance. Considering the large number of data users and documents in cloud, it is crucial for the search service to allow keyword query and provide result similarity ranking to meet the effective data recovery needed.

2. Problem Statement

2.1 The System and Threat 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.

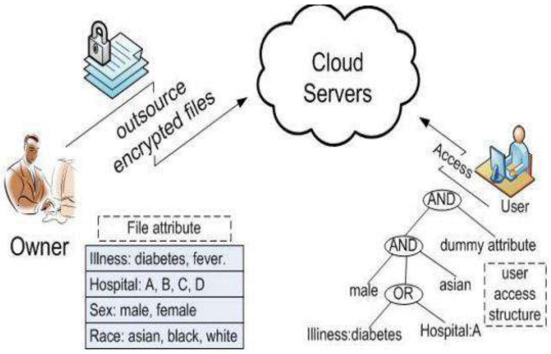

Figure 2.1: Architecture for retrieval of encrypted cloud data.

Data owner has a collection of n data files C = (F1; F2; : : : ; Fn) 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) extracted2 from the file collection C, and store both the index I and the encrypted file collection C on the cloud server. We assume the authorization between the data owner and users is appropriately done. To search the file collection for a given keyword w, an authorized user generates and submits a search request in a secret form— a trapdoor Tw of the keyword w—to the cloud server. Upon receiving the search request Tw, the cloud server is responsible to search the index I and return the corresponding set of files to the user.

To improve file retrieval accuracy for users without prior knowledge on the file collection C. However, cloud server should learn nothing or little about the relevance criteria as they exhibit significant sensitive information against keyword privacy. To reduce bandwidth, the user may send an optional value k along with the trapdoor Tw and cloud server only sends back the top-k most relevant files to the user's interested keyword w.

3. Related Work

Searching Techniques

a) Searchable Encryption

It allow users to securely search complete encrypted data through keywords, these method support only Boolean search, without capturing any relevant data. This approach suffers from two main drawbacks when directly applied in the context of Cloud Computing. First one, users, who do not necessarily have pre-knowledge of the encrypted cloud data, have to post process every got file in order to find ones most matching their interest; another drawback, invariably getting all files containing the queried keyword further incurs unnecessary network traffic, when retrieve more than one files.

b) Single Keyword Searchable Encryption

A single keyword searchable encryption schemes usually build an encrypted searchable index such that its content is hidden to the server unless it is given appropriate trapdoors generated via secret key(s). Our early work solves secure ranked keyword search which utilizes keyword frequency to rank results instead of returning undifferentiated results. However, it only supports single keyword search. Where anyone with public key can write to the data stored on server but only authorized users with private key can search. Public key solutions are usually very computationally expensive however.

c) Ranked Keyword Search

The major disadvantage of above mentioned techniques gets the better of in ranked keyword search. This system enables data users to find the most related information rapidly, rather than burdensome sorting through every match in the content collection. Ranked search can also elegantly eliminate unnecessary network traffic by sending back only the most relevant data. For privacy protection, such ranking function, however, should not leak any keyword relevant information. Another One, to improve search result accuracy as well as enhance user searching experience, it is also essential for such ranking system to support multiple keywords search.

d) Clustering Algorithm

Clustering is an important application area for many fields including data mining, statistical data analysis, compression, vector quantization, and other business applications. Clustering has been formulated in various ways in the machine learning, pattern recognition, optimization and statistics literature. The fundamental clustering problem is grouping together (clustering) similar data items. During the search process, the user has always desired to input multiple related keywords of his interest rather than a single keyword. Basically any document deal with single concept in brief and the interrelated sub-topics. Grouping the related topics together and forming cluster helps customers to get the desired document of their interest.

The most general approach is to view clustering as a density estimation problem. We assume that in addition to the observed variables for each data item, there is a hidden, unobserved variable indicating the "cluster membership". The data are assumed to arrive from a mixture model with hidden cluster identifiers. In general, a mixture model M having K clusters Ci, i=1,…,K, assigns a probability to a data point x: where Wi are the mixture weights. The problem is estimating the parameters of the individual Ci, assuming that the number of clusters K is known. The clustering optimization problem is that of finding parameters of the individual Ci which maximize the likelihood of the database given the mixture model. For general assumptions about the distributions for each of the K clusters, the EM algorithm is a popular technique for estimating the parameters.

4. Proposed Work

a) Searchable Symmetric Encryption

A Searchable symmetric encryption (SSE) allows a party to outsource the storage of its data to another party (a server). SSE schemes enable users to securely retrieve the cipher text, but these method support only Boolean keyword search, i.e., whether a keyword subsists in a file or not, without regarding the difference of relevance with the queried keyword of these files in the result. To improve security without sacrificing efficiency.

b) Order-Preserving Symmetric Encryption

Order-preserving symmetric encryption (OPE) is a deterministic encryption scheme whose encryption function preserves numerical ordering of the plaintexts. OPE has a long account in the form of one-part codes, which are lists of plaintexts and the matching cipher texts, both ordered in alphabetical or numerical order so only a single copy is required for efficient encryption and decryption.

c) Security Analysis for One-to-Many Mapping

Our one-to-many order-preserving mapping is adapted from the original OPSE, by introducing the file ID as the additional seed in the final ciphertext chosen process. Since such adaptation only functions in the final ciphertext selection process, it has nothing to do with the randomized plaintext-to-bucket mapping process in the original OPSE. In other words, the only effect of introducing file ID as the new seed is to make multiple plain text duplicates m's no longer deterministically mapped to the same ciphertext c, but instead mapped to multiple random values within the assigned bucket in range R. This helps flatten the ciphertext distribution to some extent after mapping. However, such a generic adaptation alone only works well when the numbers of plaintext duplicates are not large. In case there are many duplicates of plaintext m, its corresponding ciphertext distribution after mapping may still exhibit certain skewness or peaky feature of the plaintext distribution, due to the relative small size of assigned bucket selected from range R. This is why we propose to appropriately enlarge R. Note that in the original OPSE, size R is determined just to ensure the number of different combinations between D and R is larger than 280. But from a practical perspective, properly enlarging R in our one-to-many case further aims to ensure the low duplicates (with high probability) on the ciphertext range after mapping. This inherently increases the difficulty for adversary to tell precisely which points in the range R belong to the same score in the domain D, making the order-preserving mapping as strong as possible. Note that one disadvantage of our scheme, compared to the original OPSE, is that fixing the range size R requires preknowledge on the percentage of maximum duplicates among all the plaintexts (i.e., max=_ in (3)). However, such extra requirement can be easily met in our scenario when Building the searchable index.

d) Fully Homomorphic Encryption

To support the efficient evaluation of an arbitrary function f we may make use of a powerful class of homomorphic encryption schemes named "fully homomorphic encryption" (FHE) which support efficient homomorphic evaluation of any circuit1. Gentry proposed the first FHE scheme based on lattices that supports addition and multiplication circuits for any depth. Since addition and multiplication on any non-trivial ring constitute a Turing-complete set of gates, this scheme – if made efficient – allows one to employ *any* untrusted computing resources without risk of revealing sensitive data. This paper proposed a FHE scheme based on integers.

5. Conclusion

In this paper, we motivate and solve the problem of secure multikeyword top-k retrieval over encrypted cloud data. We define similarity relevance and scheme robustness. Based on OPE invisibly leaking sensitive information, we devise a server-side ranking SSE scheme. We then propose a TRSE scheme employing the fully homomorphic encryption, which fulfills the security requirements of multikeyword top-k retrieval over the encrypted cloud data. By security analysis, we show that the proposed scheme guarantees data privacy.

References

[1] Toward Secure Multikeyword Top-k Retrieval over Encrypted Cloud Data. Jiadi Yu, Member, IEEE, Peng Lu, Yanmin Zhu, Member, IEEE, Guangtao Xue, Member, IEEE Computer Society, and Minglu Li.

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

[3] C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, "Secure Ranked Keyword Search over Encrypted Cloud Data," Proc. IEEE 30th Int'l Conf. Distributed Computing Systems (ICDCS), 2010.

[4] M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, "Fully Homomorphic Encryption over the Integers," Proc. 29th Ann. Int'l Conf. Theory and Applications of Cryptographic Techniques, H. Gilbert, pp. 24-43, 2010.

[5] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-Preserving Multikeyword Ranked Search over Encrypted Cloud Data," Proc. IEEE INFOCOM, 2011.

[6] P. Golle, J. Staddon, and B. Waters, "Secure Conjunctive Keyword Search over Encrypted Data," Proc. Second Int'l Conf. Applied Cryptography and Network Security (ACNS), pp. 31-45, 2004.

[7] L. Ballard, S. Kamara, and F. Monrose, "Achieving Efficient Conjunctive Keyword Searches over Encrypted Data," Proc. Seventh Int'l Conf. Information and Communications Security (ICICS), 2005.

[8] J.-S. Coron, A. Mandal, D. Naccache, and M. Tibouchi, "Fully Homomorphic Encryption over the Integers with Shorter Public Keys," CRYPTO '11: Proc. 31st Ann. Conf. Advances in Cryptology, 2011.

[9] N. Smart and F. Vercauteren, "Fully Homomorphic Encryption with Relatively Small Key and Ciphertext Sizes," Proc. 13th Int'l Conf. Practice and Theory in Public key Cryptography (PKC), 2010.

[10] Sun-Ho Lee and Im-Yeong Lee," Secure Index Management Scheme on Cloud Storage Environment", International Journal of Security and Its Applications Vol. 6, No. 3, July, 2012.

[11] T.Balamuralikrishna,C.Anuradhaand N.Raghavendrasai, "Fuzzy keyword search over encrypted data over cloud computing", Asian Journal of Computer Science and Information Technology 2011.

[12] T. M Nisha and V. P Lijo, "Improving the Efficiency of Data Retrieval in Secure Cloud by Introducing Conjunction of Keywords", Proceedings published in International Journal of Computer Applications (IJCA) 25.