Enabling Privacy: Multikeyword Ranked Search over The Encrypted Cloud Data

T.Vanitha
PG Student, University College Of Engineering, Nagercoil

M.Muthuselvi
Assistent Professor, University College Of Engineering, Nagercoil

Abstract – Cloud computing has emerging as a promising pattern for data outsourcing and high-quality data services. As Cloud Computing becomes prevalent, more and more sensitive information are being centralized into the cloud, such as emails, 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 leak data information to unauthorized entities or even be hacked. It follows that sensitive data has to be encrypted prior to outsourcing for data privacy and combating unsolicited accesses. Data encryption protects data security to some extent, but at the cost of compromised efficiency. Searchable symmetric encryption (SSE) allows retrieval of encrypted data over cloud. However, 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. To enable privacy, MRSE proposes multi keyword ranked search over the encrypted cloud data. In MRSE, vector space model and homomorphic encryption are employed. The vector space model helps to provide sufficient search accuracy, and the homomorphic encryption 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. As a result, information leakage can be eliminated and data security is ensured. Thorough security and performance analysis show that the proposed scheme guarantees high security and practical efficiency.

Index Terms – Cloud, data privacy, ranking, similarity relevance, keyword search, ranked Search.

1. INTRODUCTION

With the advent of cloud computing, data owners are motivated to outsource their complex data management systems from local sites to the commercial public cloud for great flexibility and economic savings. But for protecting data privacy, sensitive data has to be encrypted before outsourcing, 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 the cloud, it is necessary to allow multiple keywords in the search request and return documents in the order of their relevance to these keywords. Related works on searchable encryption focus on single keyword search or Boolean keyword search, and rarely sort the search results. In this paper, for the first time, we define and solve the challenging problem of privacy preserving multi-keyword ranked search over encrypted cloud data (MRSE).We establish a set of strict privacy requirements for such a secure cloud data utilization system. Among various multikeyword semantics, we choose the efficient similarity measure of "coordinate matching", i.e., as many matches as possible, to capture the relevance of data documents to the search query. We further use "inner product similarity" to quantitatively evaluate such similarity measure. We first propose a basic idea for the MRSE based on secure inner product computation, and then give two significantly improved MRSE schemes to achieve various stringent privacy requirements in two different threat models. Thorough analysis investigating privacy and efficiency guarantees of proposed schemes is given. Experiments on the real-world dataset further show proposed schemes indeed introduce low overhead on computation and communication [1].

2. RELATED WORKS

We study the problem of searching on data that is encrypted using a public key system. Consider user Bob who sends email to user Alice encrypted under Alice's public key. An email gateway wants to test whether the email contains the keyword \urgent" so that it could route the email accordingly. Alice, on the other hand does not wish to give the gateway the ability to decrypt all her messages. We define and construct a mechanism that enables Alice to provide a key to the gateway that enables the gateway to test whether the word \urgent" is a keyword in the email without learning anything else about the email. We refer to this mechanism as Public Key Encryption with keyword Search. As another example, consider a mail server that stores various messages publicly encrypted for Alice by others. Using our mechanism Alice can send the mail server a key that will enable the server to identify all messages containing some specific keyword, but learn nothing else. We

define the concept of public key encryption with keyword search and give several constructions [2]. The proposed framework not only protects document/query Confidentiality against an outside intruder, but also prevents an untrusted data center from learning information about the query and the document collection. Experimental results on the W3C collection show that these techniques have comparable performance to conventional search systems designed for non-encrypted data in terms of search accuracy. the first steps to bring together advanced information retrieval and secure search capabilities for a wide range of applications including managing data in government and business operations, enabling scholarly study of sensitive data, and facilitating the document discovery process in litigation.[3] the user gives the server capability that allows the server to identify exactly those documents. Work in this area has largely focused on search criteria consisting of a single keyword. If the user is actually interested in documents containing each of several keywords (conjunctive keyword search) the user must either give the server capabilities for each of the keywords individually and rely on an intersection calculation (by either the server or the user) to determine the correct set of documents, or alternatively, the user may store additional information on the server to facilitate such searches the user allows for searches on every set of keyword [4]. We present a fully homomorphic encryption scheme which has both relatively small key and cipher text size. Our construction follows that of Gentry by producing a fully homomorphic scheme from a "somewhat" homomorphic scheme. For the somewhat homomorphic scheme the public and private keys consist of two large integers (one of which is shared by both the public and private key) and the cipher text consists of one large integer. As such our scheme has smaller message expansion and key size than Gentry's original scheme. In addition, our proposal allows efficient fully homomorphic encryption [5]. In this paper, for the first time we define and solve the problem of effective yet secure ranked keyword search over encrypted cloud data. 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 towards practical deployment of privacy-preserving data hosting services in Cloud Computing. We first give a straight forward yet ideal construction of ranked keyword search under the state-of-the-art searchable symmetric encryption (SSE) security definition, and demonstrate its inefficiency. To achieve more practical performance, we then propose a definition for ranked searchable symmetric encryption, and give an efficient design by properly utilizing the existing cryptographic primitive, order-preserving symmetric encryption (OPSE). Thorough analysis shows that our proposed solution enjoys "as-strong-as possible" security guarantee compared to previous SSE schemes, while correctly realizing the goal of ranked keyword search. Extensive experimental results demonstrate the efficiency of the proposed solution [6].

We describe cryptographic schemes for the problem of searching on encrypted data and provide proofs of security for the resulting crypto systems. Techniques have a number of crucial advantages. They provide provable secrecy for encryption, in the sense that the untrusted server cannot learn anything about the plaintext when only given the cipher text, they provide query isolation for searches, meaning that the untrusted server cannot learn anything more about the plaintext than the search result, they provide controlled searching, so that the untrusted server cannot search for an arbitrary word without the user's authorization they also support hidden queries, so that the user may ask the untrusted Server to search for a secret word without revealing the word to the server. The algorithms we present are simple, fast (for a document of length n, the encryption and Search algorithms only need O(n) stream cipher and block cipher operations), and introduce almost no space and communication overhead, and hence are practical to use today[7].

3. PROBLEM STATEMENT

3.1. Statistic Leakage

Although all data files, indices, and requests are in encrypted form before being outsourced onto cloud, the cloud server can still obtain additional information through statistical analysis. We denote the possible information leakage with statistic leakage. There are two possible statistic leakages, including term distribution and inter distribution. The term distribution of term t is t's frequency distribution of scores on each file i($i \in C$). The inter distribution of file f is file f's frequency distribution of scores of each term j($j \in f$). Term distribution and interdistribution are specific. They can be deduced either directly from ciphertext or indirectly via statistical analysis over access and search pattern. Here, access pattern refers to which keywords and the corresponding files have been retrieved during each search request, and search pattern refers to whether the keywords retrieved between two requests are the same. Based on our observation, distribution information implies a similarity relationship among terms or files. On the one hand, terms with similar term distribution always have simultaneous occurrence. For instance, obviously, the term "states" are very likely to co-occur with "united" in an official paperwork from the White House, and their term distribution, not surprisingly, are very same in a series of such a kind of paperwork. Given that this paperwork is encrypted but term distribution is not concealed, once an adversary somehow cracks out the plaintext of "united," he can reasonably guess the term that shares a similar term distribution with "united" may be "states."

3.2. K-Similarity Relevance

To avoid information leakage in server-side ranking schemes, a series of techniques have been employed to flatten or transfer the distribution of relevance scores. These approaches, however, only cover the distribution of individual term or file, ignoring the relevance between them and the violation of data privacy that arouses thereafter. To formulate this problem, we propose the concept of k-similarity relevance. Moreover, although the expected value of ratio can be reduced by properly choosing mapping function, the relatively order of them still remains as a result of the order-preserving property. Therefore, the fact that some terms are more relevant than other terms is still exposed after order preserving one-to-many mapping. However, most existing studies, including those on data outsourcing, address the data privacy and query privacy separately and cannot be applied to this problem. We propose a holistic and efficient solution that comprises a secure traversal framework and an encryption scheme based on privacy homomorphism. The framework is scalable to large datasets by leveraging an index-based approach. Based on this framework, they devise secure protocols for processing typical queries such as k-nearest-neighbor queries (kNN) on R-tree index.

3.3. Relevance Scoring

Some of the multikeyword SSE schemes support only Boolean queries, i.e., a file either matches or does not match a query. Considering the large number of data users and documents in the cloud, it is necessary to allow multikeyword in the search query and return documents in the order of their relevancy with the queried keywords. Scoring is a natural way to weight the relevance. Based on the relevance score, files can then be ranked in either ascending or descending. Several models have been proposed to score and rank files in IR community. Among these schemes, we adopt the most widely used one tf-idf weighting, which involves two attributes-term frequency and inverse document frequency. The tf-idf weighting involves two attributes, Term frequency and inverse document frequency.

3.4. Index organized by keywords

Such index data structure is usually called *inverted index*, or *inverted file* [31]. In this data structure, each keyword is followed by a file list which consists of all the files in the dataset containing this keyword. The advantage of this index structure is to allow significantly efficient text search instead of the full domain text search. But when a file is added to the dataset, it needs increased processing since all the indexes containing the keywords in this file have to be updated. It meets our needs of top-k retrieval.

4. SYSTEM DESIGN AND IMPLEMENTATION

Existing SSE schemes employ server-side ranking based on OPE to improve the efficiency of retrieval over encrypted cloud data. However, server-side ranking based on OPE violates the privacy of sensitive information, which is considered uncompressible in the security-oriented third party cloud computing scenario, i.e., security cannot be tradeoff for efficiency. To achieve data privacy, ranking has to be left to the user side. Traditional user-side schemes, however, load heavy computational burden and high communication overhead on the user side, due to the interaction between the server and the user including searchable index return and ranking score calculation. Thus, the user-side ranking schemes are challenged by practical use. A more server-siding scheme might be a better solution to privacy issues .We propose a new searchable encryption scheme, in which novel technologies in cryptography community and IR community are employed, including homomorphic encryption and the vector space model. In the proposed scheme, the data owner encrypts the searchable index with homomorphic encryption. When the cloud server receives a query consisting of multikeywords, it computes the scores from the encrypted index stored on cloud and then returns the encrypted scores of files to the data user. Next, the data user decrypts the scores and picks out the top-k highest scoring files' identifiers to request to the cloud server. The retrieval takes a two-round communication between the cloud server and the data user. We, thus, name the scheme the TRSE scheme, in which ranking is done at the user side while scoring calculation is done at the server side. Many addition or multiplication operations can be executed without decryption. In fact, the cipher text of an integer, which is another integer, can be applied to as many evaluations as needed.

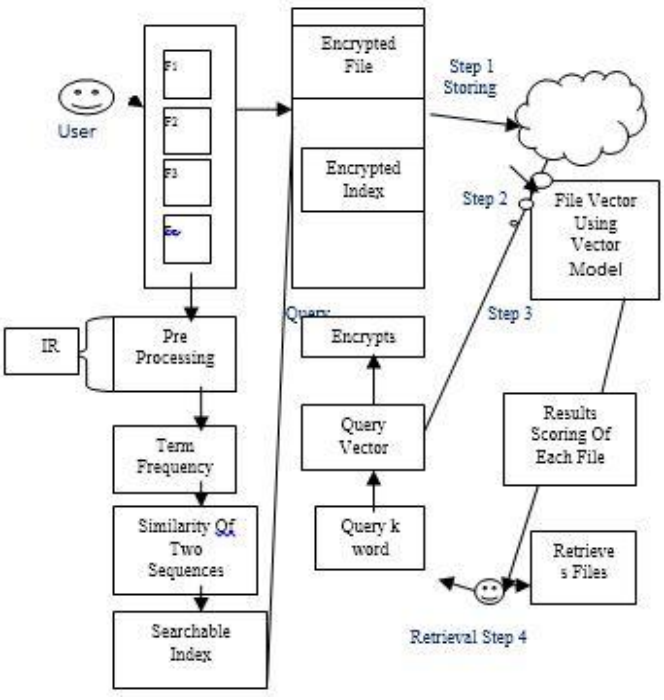

Figure 1. System Architecture

4.1. Framework of TRSE

The framework of TRSE includes four algorithms: Setup, IndexBuild, TrapdoorGen; ScoreCalculate, and Rank.
.Setup (_). The data owner generates the secret key and public keys for the homomorphic encryption scheme. The security parameter _ is taken as the input, the output is a secret key SK, and a public key set PK.
.IndexBuild(C,PK). The data owner builds the secure searchable index from the file collection C. Technologies from IR community like stemming are employed to build searchable index I from C, and then I is encrypted into I0 with PK, output the secure searchable index I0.
.TrapdoorGen(REQ; PK). The data user generates secure trapdoor from his request REQ. Vector T! is built from user's multi keyword request REQ and then encrypted into secure trapdoor T$ with public key from PK, output the secure trapdoor T$.
.ScoreCalculate(T$; I0). When receives secure trapdoor T$, the cloud server computes the scores of each files in I0 with T$ and returns the encrypted result vector @ back to the data user.
. Rank(@,SK,k). The data user decrypts the vector @ with secret key SK and then requests and gets the files with top-k scores.

4.2. Homomorphic Encryption

Homomorphic encryption is a form of encryption which allows specific types of computations to be carried out on cipher text and obtain an encrypted result which decrypted matches the result of operations performed on the plaintext. For instance, one person could add two encrypted numbers and then another person could decrypt the result, without either of them being able to find the value of the individual numbers. This is a desirable feature in modern system architectures. Homomorphic encryption would allow the chaining together of different services without exposing the data to each of those services, for example a chain of different services from different companies could 1) calculate the tax 2) the currency exchange rate 3) shipping, on a transaction without exposing the unencrypted data to each of those services. Homomorphism encryption schemes are malleable by design. The homomorphic property of various cryptosystems can be used to create secure voting systems, collision-resistant hash functions, and private information retrieval schemes and enable widespread use of cloud computing by ensuring the confidentiality of processed data. There are several efficient, partially homomorphic cryptosystems, and a number of fully homomorphic, but less efficient cryptosystems. Although a cryptosystem which is unintentionally homomorphic can be subject to attacks on this basis, if treated carefully homomorphism can also be used to perform computations securely. Homomorphic scheme merely uses addition and multiplication over the integers rather than working with ideal lattices over a polynomial ring. The main appeal of their approach is the conceptual simplicity. They reduce the security of our somewhat homomorphic scheme to finding an approximate Integer gcd – i.e., given a list of integers that are near-multiples of a hidden integer, output that hidden integer. They investigate the hardness of this task, building on earlier work of How grave-Graham [8].

5. CONCLUSION

We motivate and solve the problem of secure multikeyword top-k retrieval over encrypted cloud data. On the basis of SSE, they proposed the one-to-many OPM to further improve the efficiency while security guarantee and retrieval accuracy are slightly weakened. However, these schemes support only single keyword retrieval. We then propose a TRSE scheme employing the fully homomorphic encryption, which fulfils 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. According to the efficiency evaluation of the proposed scheme over a real data set, extensive experimental results demonstrate that our scheme ensures practical efficiency.

REFERENCES

[1] D. Song, D. Wagner, and A. Perrig, "Practical Techniques for Searches on Encrypted Data," Proc. IEEE Symp. Security and Privacy, 2000.
[2] D. Boneh, G. Crescenzo, R. Ostrovsky, and G. Persiano, "Public-Key Encryption with Keyword Search," Proc. Int'l Conf. Theory and Applications of Cryptographic Techniques (Eurocrypt), 2004.
[3] A. Swaminathan, Y. Mao, G.-M. Su, H. Gou, A.L. Varna, S. He, M.Wu, and D.W. Oard, Confidentiality-Preserving Rank-Ordered Search," Proc. Workshop Storage Security and Survivability, 2007.
[4] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-Preserving Multikeyword Ranked Search over Encrypted Cloud Data," Proc.IEEE INFOCOM, 2011.
[5] 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.
[6] N. Smart and F. Vercauteren, "Fully Homomorphic Encryptionwith Relatively Small Key and Ciphertext Sizes," Proc. 13th Int'l Conf. Practice and Theory in Public Key Cryptography (PKC), 2010.
[7] 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.
[8] 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.