Scientific Journal of Impact Factor(SJIF): 4.14
e-ISSN(O): 2348-4470
p-ISSN(P): 2348-6406

International Journal of Advance Engineering and Research Development

Special Issue for ICPCDECT 2016,Volume 3 Issue 1

MULTI KEYWORD RANKED SEARCH OVER ENCRYPTED CLOUD DATA

Nikhil jagtap[1], Tushar Kalbhor[2], Ravindra Rokade[3] , Devashish Shah[4] , Seema Vanjire[5]

[1-5]*Computer Department, Sinhgad Academy of Engineering,*

Abstract — *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 have 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. We define and solve the challenging problem of privacy-preserving multi-keyword ranked search over encrypted data in cloud computing* ***(****MRSE****).***

*Keywords-**component;*** *Cloud Computing*
Fault Tolerance Manager
Coordinate Matching
Multi-Keyword Ranked Search

I. INTRODUCTION

Privacy-preserving multi-keyword ranked search over encrypted cloud data (MRSE), and establish a set of strict privacy requirements for such a secure cloud data utilization system to become a reality. Among various multi-keyword semantics, we choose the efficient principle of coordinate matching, i.e., as many matches as possible, to capture the similarity between search query and data documents, and further use inner product similarity to quantitatively formalize such principle for similarity measurement. FTM is an innovative perspective on creating and managing fault tolerance that shades the implementation details of the reliability techniques from the users by means of a dedicated service layer. This allows users to specify and apply the desired level of fault tolerance without requiring any knowledge about its implementation.

Multi-keyword ranked search over encrypted cloud data (MRSE) while preserving strict system wise privacy in cloud computing paradigm. Among various multikeyword semantics, we choose the efficient principle of coordinate matching, i.e., as many matches as possible, to capture the similarity between search Query and data documents. Specifically, we use inner product similarity, i.e., the number of query keywords appearing in a document, to quantitatively evaluate the similarity of that document to the search query in coordinate matching principle. During index construction, each document is associated with a binary vector as a sub-index where each bit represents whether corresponding keyword is contained in the document. The search query is also described as a binary vector where each bit means whether corresponding keyword appears in this search request, so the similarity could be exactly measured by inner product of query vector with data vector. However, directly outsourcing data vector or query vector will violate index privacy or search privacy.

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 [1]. Its great flexibility and economic savings are motivating both individuals and enterprises to outsource their local complex data management system into the cloud. To protect data privacy and combat unsolicited accesses in the cloud and beyond, sensitive data, e.g., emails, personal health records, photo albums, tax documents, financial transactions, etc., may have to be encrypted by data owners before outsourcing to the commercial public cloud [2]; this, however, obsoletes the traditional data utilization service based on plaintext keyword search. The trivial solution of downloading all the data and decrypting locally is clearly impractical, due to the huge amount of bandwidth cost in cloud scale systems. Moreover, aside from eliminating the local storage management, storing data into the cloud serves no purpose unless they can be easily searched and utilized. Thus, exploring privacy-preserving and effective search service over encrypted cloud data is of paramount importance. Considering the potentially large number of on-demand data users and huge amount of outsourced data documents in the cloud, this problem is particularly challenging as it is extremely difficult to meet also the requirements of performance, system usability and scalability.

On the one hand, to meet the effective data retrieval need, the large amount of documents demand the cloud server to perform result relevance ranking, instead of returning undifferentiated results. Such ranked search system enables data users to find the most relevant information quickly, rather than burdensomely sorting through every match in the content collection [3]. Ranked search can also elegantly eliminate unnecessary network traffic by sending back only the most relevant data, which is highly desirable in the "pay-as-you use" cloud paradigm. For privacy protection, such ranking operation, however, should not leak any keyword related information. On the other hand, to improve the search result accuracy as well as to enhance the user searching experience, it is also necessary for such ranking system to support multiple keywords search, as single keyword search often yields far too coarse results. As a common practice indicated by today's web search engines (e.g., Google search), data users may tend to provide a set of keywords instead of only one as the indicator of their search interest to retrieve the most relevant data. And each keyword in the search request is able to help narrow down the search result further. "Coordinate matching" [4], i.e., as many matches as possible, is an efficient similarity measure among such multi-keyword semantics to refine the result relevance, and has been widely used in the plaintext information retrieval (IR) community. However, how to apply it in the encrypted cloud data search system remains a very challenging task because of inherent security and privacy obstacles, including various strict requirements like the data privacy, the index privacy, the keyword privacy, and many others .

In the literature, searchable encryption [5]–[13] is a helpful technique that treats encrypted data as documents and allows a user to securely search through a single keyword and retrieve documents of interest. However, direct application of these approaches to the secure large scale cloud data utilization system would not be necessarily suitable, as they are developed as crypto primitives and cannot accommodate such high service-level requirements like system usability, user searching experience, and easy information discovery. Although some recent designs have been proposed to support Boolean keyword search [14]–[15] as an attempt to enrich the search flexibility, they are still not adequate to provide users with acceptable result ranking functionality. Our early work [15] has been aware of this problem, and provided a solution to the secure ranked search over encrypted data problem but only for queries consisting of a single keyword. How to design an efficient encrypted data search mechanism that supports multi-keyword semantics without privacy breaches still remains a challenging open problem.

In this paper, for the first time, we define and solve the problem of multi-keyword ranked search over encrypted cloud data (MRSE) while preserving strict system-wise privacy in the cloud computing paradigm. 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. Specifically, we use "inner product similarity" [4], i.e., the number of query keywords appearing in a document, to quantitatively evaluate such similarity measure of that document to the search query. During the index construction, each document is associated with a binary vector as a sub index where each bit represents whether corresponding keyword is contained in the document. The search query is also described as a binary vector where each bit means whether corresponding keyword appears in this search request, so the similarity could be exactly measured by the inner product of the query vector with the data vector. However, directly outsourcing the data vector or the query vector will violate the index privacy or the search privacy. To meet the challenge of supporting such multi-keyword semantic without privacy breaches, we propose a basic idea for the MRSE using secure inner product computation, which is adapted from a secure k-nearest neighbor (*kNN*) technique [4], and then give two significantly improved MRSE schemes in a step-by step manner to achieve various stringent privacy requirements in two threat models with increased attack capabilities.

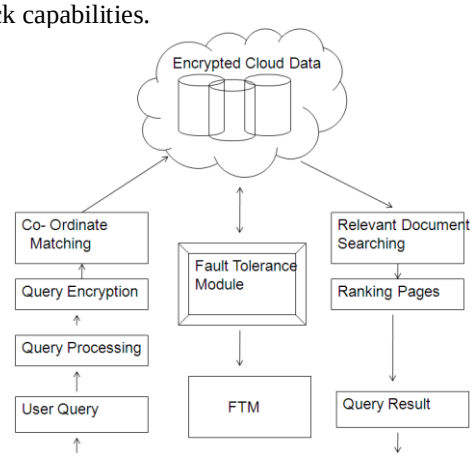

Figure 1. System Architecture

II. Design and Implementation Constraints proposed System

RSA algorithm- RSA is an algorithm for public-key cryptography that is based on the presumed difficulty of factoring large integers, the factoring problem. RSA stands for Ron Rivest, Adi Shamir and Leonard Adleman, who first publicly described it in 1977. Clifford Cocks, an English mathematician, had developed an equivalent system in 1973, but it was classified until 1997. A user of RSA creates and then publishes the product of two large prime numbers, along with an auxiliary value, as their public key. The prime factors must be kept secret. Anyone can use the public key to encrypt a message, but with currently published methods, if the public key is large enough, only someone with knowledge of the prime factors can feasibly decode the message. Whether breaking RSA encryption is as hard as factoring is an open question known as the RSA problem. The RSA algorithm involves three steps: key generation, encryption and decryption.

Key generation RSA involves a public key and a private key. The public key can be known to everyone and is used for encrypting messages. Messages encrypted with the public key can only be decrypted using the private key. The keys for the RSA algorithm are generated the following way:

I. Choose two distinct prime numbers p and q.

For security purposes, the integer's p and q should be chosen at random, and should be of similar bit-length. Prime integers can be efficiently found using a primarily test.

II. Compute n = pq.
n is used as the modulus for both the public and private keys

III. Compute f(n) = (p1)(q1), where is Euler's totient function.

IV. Choose an integer e such that 1 < e < (n) and greatest common divisor of
(e, f (n)) = 1; i.e. and f (n) are co-prime.

Encryption Alice transmits her public key to Bob and keeps the private key secret. Bob then wishes to send message M to Alice. He first turns M into an integer m, such that by using an agreed-upon reversible protocol known as a padding scheme. He then computes the cipher text corresponding to this can be done quickly using the method of exponentiation by squaring. Bob then transmits to Alice. Note that at least nine values of m could yield a cipher text equal to m, but this is very unlikely to occur in practice.

Decryption Alice can recover from by using her private key exponent via computing. Given, she can recover the original message M by reversing the padding scheme.

KNN-ALGORITHM- K-nearest neighbor search identifies the top k nearest neighbors to the query. This technique is commonly used in predictive analytics to estimate or classify a point based on the consensus of its neighbors. K-nearest neighbor graphs are graphs in which every point is connected to its k nearest neighbors. The basic idea of our new algorithm: The value of dmax is decreased keeping step with the ongoing exact evaluation of the object similarity distance for the candidates. At the end of the step by step refinement, dmax reaches the optimal query range Ed and prevents the method from producing more candidates than necessary thus fulfilling the optimality criterion. NearestNeighhorSearch (q, k) // optimal algorithm

I. Initialize ranking = index.increm-ranking (F(q);d f)

II. Initialize result = new sorted-list (key, object)

III. Initialize dmax = w

IV. While o = ranking.getnext and d; (o;q)Id; ; ;do

V. If do s > sdmax then result.insert (d; (o;q);o)

VI. If result.length 2 k then dmax = result[k].key

VII. Remove all entries from result where key > dmax

VIII. End while Report all entries from result where key I dmax

III. System Design

MRSC and FTM - We are developing the project which search on encrypted document and gives the encrypted result. And also fault tolerance manager is doing the fault detection and recovery.

Fault Tolerance Manager-This is the central computing component of FTM which manages all the reliability mechanisms present in the framework. It contemplates the user's requirements and accordingly selects the Web (reliability) services from other components. The chosen modules are then orchestrated to form an aggregate solution that is delivered to the user's application. If a failure is detected, the fault masking and recovery services are invoked.

Recovery Manager - This component includes all the mechanisms that resumes error-prone nodes to a normal operational mode. The impact of failure detection and masking techniques on the system is complementary to that of recovery mechanisms. By continuously checking for the occurrence of faults and invoking the recovery service when exceptions happen, our framework maximizes the systems lifetime and minimizes the downtime during failures.

Coordinate Matching - As a hybrid of conjunctive search and disjunctive search, coordinate matching is an intermediate approach which uses the number of query keywords appearing in the document to quantify the similarity of that document to the query. When users know the exact subset of the dataset to be retrieved, Boolean queries perform well with the precise search requirement specified by the user. In cloud computing, however, this is not the practical case, given the huge amount of outsourced data. Therefore, it is more flexible for users to specify a list of keywords*** indicating their interest and retrieve the most relevant documents with rank order.

IV. Related Work

Single keyword Searchable Encryption -Traditional single keyword searchable encryption schemes [5][13], 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) [2]. It is first studied by Song et al. [5] in the symmetric key setting, and improvements and advanced security definitions are given in Goh [6], Chang et al. [7] and Curtmola et al. [8]. 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. In the public key setting, Boneh et al. [9] present the first searchable encryption construction, 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. Furthermore, the keyword privacy could not be protected in the public key setting since server could encrypt any keyword with public key and then use the received trapdoor to evaluate this cipher text.

Boolean Keyword Searchable Encryption - To enrich search functionalities, conjunctive keyword search [14][15] over encrypted data have been proposed. These schemes incur large overhead caused by their fundamental primitives, such as computation cost by bilinear map, e.g. [15], or communication cost by secret sharing, e.g. [15]. As a more general search approach, predicate encryption schemes [1][2] are recently proposed to support both conjunctive and disjunctive search. Conjunctive keyword search returns all-or-nothing, which means it only returns those documents in which all the keywords specified by the search query appear; disjunctive keyword search returns undifferentiated results, which means it returns every document that contains a subset of the specific keywords, even only one keyword of interest. In short, none of existing Boolean keyword searchable encryption schemes support multiple keywords ranked search over encrypted cloud data while preserving privacy as we propose to explore in this paper. Note that, inner product queries in predicate encryption only predicates whether two vectors are orthogonal or not, i.e., the inner product value is concealed except when it equals zero. Without providing the capability to compare concealed inner products, predicate encryption is not qualified for performing ranked search. Furthermore, most of these schemes are built upon the expensive evaluation of pairing operations on elliptic curves. Such inefficiency disadvantage also limits their practical performance when deployed in the cloud. On a different front, the research on top retrieval [15] in database community is also loosely connected to our problem.

A. Design Goals

Multi-keyword Ranked Search: To design search schemes which allow multikeyword query and provide result similarity ranking for effective data retrieval, instead of returning undifferentiated results.

Privacy-Preserving: To prevent cloud server from learning additional information from dataset and index, and to meet privacy requirements specified.

Fault Tolerance: If a failure is detected, the fault masking and recovery services are invoked.

Efficiency: Above goals on functionality and privacy should be achieved with low communication and computation overhead.

Conclusion

The System Multi-keyword Ranked Search with Fault Tolerance Manager which work on encrypted cloud data, which provide users many specialized services according to the users specific needs so that the system can satisfied all special requirements of users. After searching the data the result has been return in the decrypted document.

In Fault Tolerance, Log file describes the logged in details of user or any other person who is trying to access the account. System which gives alert sending message (SMS) from Mobile Server on users mobile. It describes which file property has been updated or deleted by hacker. But the best part is only data which is visible to hacker on cloud is updated, original data is not updated. So when user will get alert user could again upload data on cloud and the original data is secured and protected from hacker.

Thorough analysis investigating privacy and efficiency guarantees of proposed schemes is given, and experiments on the real-world dataset show our proposed schemes introduce low overhead on both computation and communication

REFERENCES

[1] Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters. In Proc. Of ICDCS10, 2010.
[2] Wang, N. Cao, J. Li, K. Ren, and W. Lou. "Secure ranked keyword search over encrypted cloud data" in Proc. of ICDCS10, 2010.
[3] Shen, E. Shi, and B. "Waters privacy in encryption systems" in Proc. of TCC, 2009.
[4] Zhao, P. M. Melliar-Smith, and L. E. Moser "Fault Tolerance Middleware for Cloud Computing" IEEE Computer Society, 2010, pp. 6774
[5] M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindne, "A break in the clouds: towards a cloud denition" ACM SIGCOMM Compute. Commun. Rev. Vol. 39, no. 1, pp. 5055, 2009
[6] Kamara and K." Lauter Cryptographic cloud storage" in RLCPS, January 2010, LNCS. Springer, Heidelberg.
[7] "Singhal Modern information retrieval: A brief overview IEEE Data Engineering Bulletin", vol. 24, no. 4, pp. 3543, 2001.
[8] H. Witten, A. Moffat, and T. C. Bell, "Managing gigabytes: Compressing and indexing documents and images Morgan Kaufmann Publishing, San Francisco", May 1999.
[9] Song, D.Wagner, and A. Perrig, "Practical techniques for searches on encrypted data" in Proc. of S&P, 2000.
[10] J. Goh, Secure indexes Cryptology ePrint Archive, 2003, http://eprint.iacr.org/2003/216.
[11] C. Chang and M. Mitzenmacher Privacy preserving keyword searches on remote encrypted data in Proc. of ACNS, 2005.
[12] Curtmola, J. A. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved denitions and efficient constructions", in Proc. of ACM CCS,2006.
[13] Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search", in Proc. of EUROCRYPT, 2004.
[14] Bellare, A. Boldyreva, and A. ONeill, "Deterministic and efficiently searchable encryption" in Proc. of CRYPTO, 2007
[15] Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, "Searchable encryption revisited: Consistency properties", relation to anonymous ibe, and extensions vol. 21, no. 3, pp. 350391, 2008

