An Effective Multikeyword Ranked Search Technique using greedy algorithm over Encrypted Cloud data

Miss. T.A.Rahane[1], Prof. S.K.Sonkar[2]
[1, 2] Department of Computer Engineering
[1, 2] AVCOE,sangamner ,Maharashtra.

Abstract- *Now a days there will be growing popularity of cloud computing, large number of users and data owners are motivated to outsource their data to cloud servers for large convenience and reduced cost required for data management. However, important data should be encrypted before outsourcing for privacy requirements, which uses data utilization technique like keyword based document recovery. A secure multi-keyword ranked search scheme over encrypted cloud data, which concurrently supports dynamic update operations like deletion and insertion of documents. mostly, the vector space model and the widely used TF-IDF model are combined in the index construction and query generation. Creating a special tree-based index structure with the help of Greedy Depth-first Search algorithm which gives well organized multi-keyword ranked search. The secure kNN algorithm is utilized to encrypt the index and query vectors, and for the time being ensures accurate relevance score calculation between encrypted index and query vectors. Dummy terms are added to the index vector for blinding search results, in order to resist statistical attacks. Due to the use of special tree-based index structure, the proposed concept can achieve sub-linear search time and deal with the deletion and insertion of documents flexibly. Extensive experiments are conducted to express the efficiency of the proposed scheme.*

Keywords- Searchable encryption, multi-keyword ranked search, dynamic update, cloud computing

I. INTRODUCTION

The large amount of documents demand the cloud server to perform result relevance ranking, instead of returning undifferentiated results, that required for efficient data retrieval. This type ranked search system helps enables data users to find the most relevant information speedily, rather than burdensomely 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, which is highly desirable in the pay-as-you-use cloud pattern.

For privacy protection, such as ranking operation, they should not to drop out any keyword related information. At the same time, this can improve the search result accuracy 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.

This system 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 multi-keyword semantics, to 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, To use inner product similarity, i.e., the number of query keywords appearing in that document, to quantitatively evaluate such similarity measure of that document to the search query. During construction of index, each document is associated with a binary vector as a sub-index where each bit represents whether corresponding keyword is appearing in the document. The search query is also narrate 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 point out the data vector or the query vector will violate the index privacy or the search privacy. To come across the challenge of supporting such multi keyword semantic without privacy risk, To put a basic idea for the MRSE using secure inner product computation, which is adapted from a secure k-nearest neighbor (KNN) technique, and then give two significantly improved MRSE schemes in a step-by-step manner to achieve various stringent privacy requirements.

II. RELATED WORK

Kui Ren, Cong Wang, and Qian Wang[1], Cloud computing represents todays most exciting computing pattern shift in information technology. However, security and privacy are perceived as primary obstacles to its large adoption. Here, outline several critical security challenges and motivate further investigation of security solutions for a trustworthy public cloud environment. cloud computing is the latest concept for the long-dreamed vision of computing as a usefulness. The cloud provides convenient, on-demand network access to a centralized pool of configurable computing resources that can be rapidly deployed with great efficiency and minimal management overhead. With its priority advantages, cloud computing enables a fundamental example shift in how to arrange and provide computing services. It makes possible computing outsourcing such that both individuals and enterprises can avoid committing large capital outlays when purchasing and managing software and hardware, as Toll as dealing with the operational overhead. S. Kamara and K. Lauter[2], To consider the problem of creating a secure cloud storage space service on top of a public cloud infrastructure where the service provider is not totally trusted by the customer. To explain, at a high level, several architectures that combine recent and non-standard cryptographic primitives in order to achieve aim. To study the benefits such an architecture would provide to both customers and service providers and give an outline of latest advances in cryptography forced specifically by cloud storage.

C. Gentry[3], propose the flrst completely homomorphic encryption system, evaluate a central open problem in cryptography. Such a scheme allows one to compute random functions over encrypted data without the decryption key i.e.,given encryptions E(m1),.,E(mt) of m1,, mt, one can efficiently compute a compact ciphertext that encrypts f(m1,,mt) for any efficiently computable function f. This problem was asked by Rivest et al. in 1978. Fully homomorphic encryption has several applications. For example, it enables private queries to a search engine the user gives an encrypted query and the search engine computes a to the point encrypted answer without ever looking at the query in the clear. It also enables searching on encrypted data a user stores encrypted files on a remote file server and can later have the server retrieve only files that (when decrypted) assure some Boolean constraint, even though the server cannot decrypt the files on its own. Fully homomorphic encryption improves the efficiency of secure multiparty computation. While creation begins with a somewhat homomorphic boostrappable" encryption scheme that works when the function f is the schemes own decryption function. To then show how, through recursive self-embedding, bootstrappable encryption gives fully homomorphic encryption. The construction makes use of hard problems on ideal lattices.

O. Goldreich and R. Ostrovsky[4], To present a theoretical handling of software security. To refine and formulate the key problem of learning about a program from its execution, and shrink this problem to the problem of on-line simulation of an arbitrary program on an oblivious RAM. It present main result an capable simulation of an arbitrary (RAM) program on a probabilistic oblivious RAM. Assuming that one-way functions exists, They show ones make software protection plan robust against a polynomial-time adversary who is allowed to alter memory contents during implementation in a dynamic style.

D. Boneh[5], G. Di Crescenzo, R. Ostrovsky, and G. Persiano , the problem of searching on data that is encrypted using a public key system. Assume user Bob who sends email to user Alice encrypted under Alices public key. An email gateway needs to test whether the email contains the keyword " so that it could direction the email accordingly. Alice, on the other hand does not wish to provide the gateway the ability to decrypt all her messages. To define and construct a mechanism that enables Alice to provide a key to the gateway that enables the gateway to test whether the word " is a keyword in the email without learning anything else about the email. By using this method as Public Key Encryption with keyword Search. Lets take different example, consider a mail server that stores various messages publicly encrypted for Alice by others. Using our method 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. To define the idea of public key encryption with keyword search and give several creations.

D. Boneh [6], E. Kushilevitz, R. Ostrovsky, and W. E. Skeith III, They consider the problem: Alice wishes to maintain her email using a storage-provider Bob (such as a Yahoo! or hotmail email account). That storage provider should gives for Alice the ability to gather, retrieve, search and delete emails but, simultaneously, should learn neither the content of messages sent from the senders to Alice (with Bob as an intermediary), nor the search criteria used by Alice.

A trivial solution is that messages will be sent to Bob in encrypted form and Alice, whenever she wants to search for some message, will ask Bob to send her a copy of the entire database of encrypted emails. This is highly ineffective. To will be interested in solutions that are communication efficient and, at the same time, respect the privacy of Alice. They gives us to create a public-key encryption scheme for Alice that allows PIR searching over encrypted documents. Our solution

is the first to reveal no partial information regarding the user's search (including the access pattern) in the public-key setting and with nontrivially small communication complexity.This provides a theoretical solution to a problem posed by Boneh, DiCrescenzo, Ostrovsky and Persiano on "Public-key Encryption with Keyword Search." The core method of their solution also allows for Single-Database PIR writing with sub linear communication complexity.

D. X. Song, D. Wagner, and A. Perrig [7], It is desirable to store data on data storage servers such as mail servers and file servers in encrypted form to decrease security and privacy risks. But this typically implies that one has to give up functionality for security. For example, if a client needs to retrieve only documents containing certain words, it was not before known how to let the data storage server execute the search and answer the query without loss of data privacy. To explain our cryptographic schemes for the problem of searching on encrypted data and give proofs of security for the resulting crypto systems. This techniques have a number of important advantages. They are provably secure: they give provable privacy for encryption, in the sense that the untrusted server cannot learn whatever thing about the plaintext when only given the ciphertext; they give query separation for searches, meaning that the untrusted servercannot learn anything more about the plaintext than the search result; they provide controlled searching, so that the untrusted server cannot search for an random word without the users authorization; they also carry 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 to 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 now a days

III. PROBLEM DEFINITION

i. The Problem is to determine how to securely search any document from cloud in form of encrypteddata.
ii. Rank Based Search data.
iii. How to Store data in Secure form on cloud.
iv. Despite of the various advantages of cloud services, outsourcing sensitive information such as e-mails, personal health records, company finance data, government documents, etc.

IV. SYSTEM ARCHITECTURE

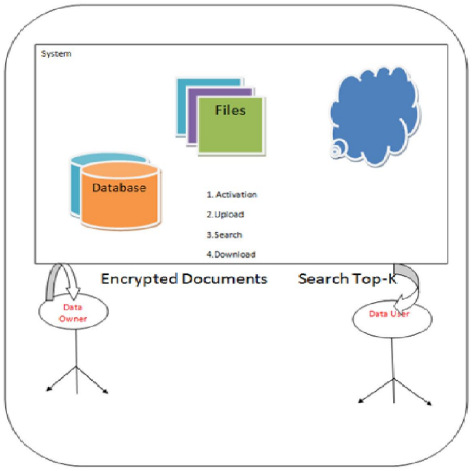

Figure 1. System Architecture

Data Owner : Register with cloud server and login(username must be unique). Send request to Private key generator (PKG) to generate IBE Key on the user name. Browse file and request Private key to encrypt the data, Upload data to cloud service provider. Verify the data from the cloud.

Public Key Generator : Receive request from the users to generate the key, Store all keys based on the user names. Check the username and provide the private key. Revoke the end user (File Receiver if they try to hack file in the cloud server and un revoke the user after updating the private key for the corresponding file based on the user)

Key Update : Receive all files from the data owner and store all files. Check the data integrity in the cloud and inform to end user about the data integrity. Send request to PKG to Update the private key of the user based on the date parameter (Give some date to update Private Key). List all files, List all updated Private Key details based on the date and users, List all File attackers and File Receive Attackers.

Search Top-K : In this module receiver first has to Register and login, Request secret key, Request Top-K files in the cloud and Receive files.

V. MATHEMATICAL MODEL

Input :

1. U(Z) = fu1; u2; u3:::::ung
2. K(Z) = fk1; k2; k3::::::kng
3. D(Z) = fd1; d2; d3::::::dng
4. S(Z) = fs1; s2; s3::::::::sng

P(Z) = F(un; sn; kn; dn):::::::ui > 0

Where,
U:::::::::::::::User
K:::::::::::::::Keyword
D:::::::::::::::Download
S:::::::::::::::Search

Output :

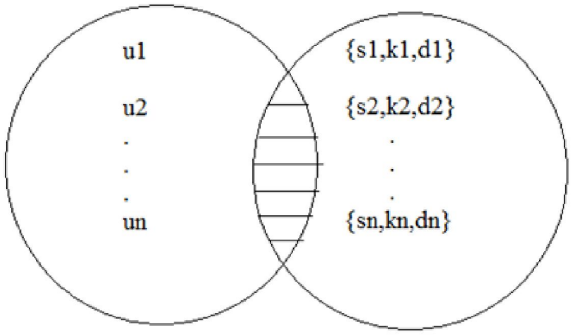

Figure 2. U(Z) U S(Z) U K(Z) U D(Z)

Above diagram shows how much search user as per keyword.

1. Keyword search data multiple order.
2. Every keyword check data as per TF.

Success Condition :
- TF is Properly done.

Failure Condition :
- Some docs not maintain TF.

VI. ALGORITHMS

1. Algorithm to provide efficient multi-keyword ranked search.
2. The secure kNN algorithm is utilized to encrypt the index and query vectors.
3. Propose a Greedy Depth-first Search algorithm based on this index tree.
4. Algorithm achieves better-than-linear search efficiency but results in precision loss.
5. The LSH algorithm is suitable for similar search but cannot provide exact ranking.
6. fIs; cig GenUpdateInfo (SK; Ts; i; up type) This algorithm generates the update information fIs; cigwhich will be sent to the cloud server.

VII. EXPERIMENTAL SETUP

In this the Structure consist of technologies like JAVA , HTML , CSS , Java script. For back end MySql is used. Hence before investigational set up Software like Eclipse, Tomcat is predictable to be installed on server. User must have basic windows Family, good browser to view the results. Supervised Dataset or Un-Supervised dataset is used for testing in MySQL is tested.

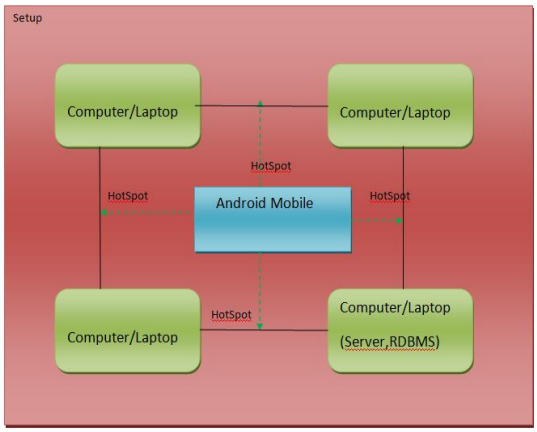

Figure 3. Experimental setup

VIII. RESULTS AND DISCUSSIONS

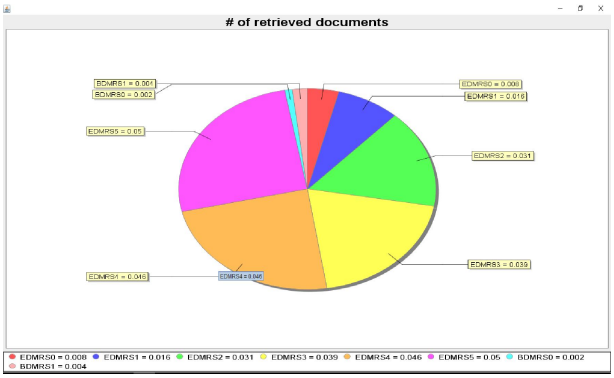

Figure 4. Given pie chart shows the documents retrieved from multiple users.

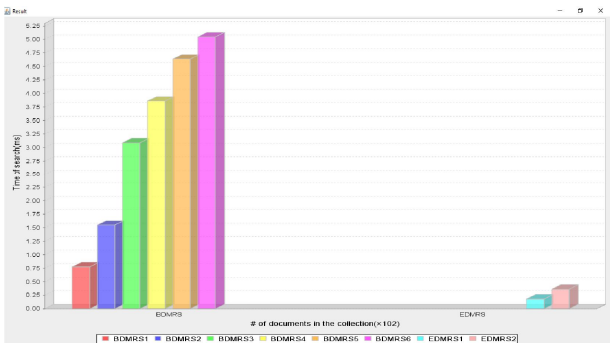

Figure 5. Time required to search BDMRS & EDMRS

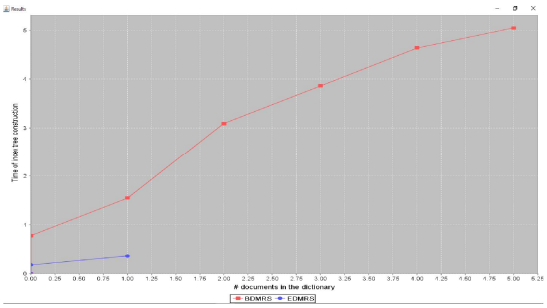

Figure 6. Time of index tree construction

IX. CONCLUSION

The Existing techniques on keyword-based information retrieval, which are widely used on the plaintext data, cannot be directly applied on the encrypted data. Downloading all the data from the cloud and decrypt locally is obviously impractical. All these multi keyword search schemes retrieve search results based on the existence of keywords, which cannot provide acceptable result ranking functionality.

However, sensitive data should be encrypted before outsourcing for privacy requirements, which obsoletes data utilization like keyword-based document retrieval.

X. ACKNOWLEDGMENT

It gives me an immense pleasure to express my sincere and heartiest gratitude towards my guide Prof. S.K.Sonkar for his guidance, encouragement, moral support and affection during the course of my work. I am especially appreciative of his willingness to listen and guide me to find the best solution, regardless of the challenge. This work is also the outcome of the blessing guidance and support of my parents and family members and friends. I am also thankful to all who have contributed indirectly and materially in words and deeds for completion of this work.

REFERENCES

[1] K. Ren, C. Wang, Q. Wang et al., Security challenges for the public cloud, IEEE Internet Computing, vol. 16, no. 1, pp. 6973, 2012.

[2] S. Kamara and K. Lauter, Cryptographic cloud storage, in Financial Cryptography and Data Security. Springer, 2010, pp. 136149.

[3] C. Gentry, A fully homomorphic encryption scheme, Ph.D. dissertation, Stanford University, 2009.

[4] O. Goldreich and R. Ostrovsky, Software protection and simulation on oblivious rams, Journal of the ACM (JACM), vol. 43, no. 3,pp. 431473, 1996.

[5] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, Public key encryption with keyword search, in Advances in CryptologyEurocrypt 2004. Springer, 2004, pp.506522.

[6] D. Boneh, E. Kushilevitz, R. Ostrovsky, and W. E. Skeith III,Public key encryption that allows pir queries, in Advances in Cryptology-CRYPTO 2007. Springer, 2007, pp. 5067.

[7] D. X. Song, D. Wagner, and A. Perrig, Practical techniques for searches on encrypted data, in Security and Privacy, 2000. SP2000.Proceedings.2000 IEEE Symposium on.IEEE, 2000, pp. 44 55.

[8] Y.-C. Chang and M. Mitzenmacher, Privacy preserving keyword searches on remote encrypted data, in Proceedings of the Third international conference on Applied Cryptography and Network Security. Springer-Verlag, 2005, pp. 442455.

[9] E.-J. Goh et al., Secure indexes. IACR Cryptology ePrint Archive, vol. 2003, p. 216, 2003.

[10] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, Searchable symmetric encryption: improved definitions and efficient constructions, in Proceedings of the 13th ACM conference on Computer and communications security. ACM, 2006, pp. 7988