Enabling Protection and Well-Organized MRSE over Encrypted Cloud Data Using CP-ABE

Revathy B.D[#1], Tejaswini .B[#2] ,

Abstract— By means of the new arrival of cloud computing, data proprietors are provoked to outsource their multifarious data management systems from local sites to the profitable public cloud for great elasticity and financial investments. Protecting data privacy, valuable data has to be encrypted before outsourcing, which drawback of conventional data base management system. Providing encrypted cloud data search service is of supreme significance. Taking into consideration the large number of clients and credentials in the cloud, it is essential to allow multiple keywords in the search appeal and return credentials in the order of their significance to these keywords. To solve the challenging problem providing security for the cloud credentials and multi-keyword ranked search over encrypted cloud data (MRSE), we set up a set of stringent privacy necessities for such a secure cloud data utilization system. We suggest a basic idea for the MRSE based on Automatic Annotation, and then give two appreciably enhanced MRSE schemes to accomplish various stringent privacy requirements in two special threat models. Methodical analysis investigating privacy and effectiveness guarantees of projected schemes is given.

***Index Terms*—Encryption, Multi keyword, Ranked Search, Searchable.**

I. INTRODUCTION

Cloud computing is new technology for storage and it is famous for storage, scalability and remote access. It allows the data manager to store and retrieve the data remotely, once the data is stored on the cloud, the complete control of the data comes under cloud server. Cloud servers are the semi trusted storage device, to protect valuable information (password, Account number); the sensitive data's must be encrypted before storing it on the cloud. Exploring data Security and effective search service over encrypted cloud data is important Cloud storage services allow the users to outsource their data in the cloud storage servers and retrieve them whenever and wherever required, To meet the successful data retrieval, the large amount of data demand the cloud server to perform result significance ranking, as an alternative of returning undifferentiated results. Such ranked seek out system enables data proprietor to find the most Related information rapidly, Ranked search can also gracefully removes unnecessary network traffic by sending back only the most related data, which is extremely fashionable in the "pay-as-you-use" cloud model. For privacy Safety, such ranking procedure should not disclose any keyword related information. To improve the search result correctness as well as to improve the user searching knowledge, it is also essential for such ranking system to support multiple keywords search. "Automatic Annotation", I.e., as many matches as achievable, is a well-organized similarity Measure among various multi-keyword semantics to process the result significance, and has been generally used in the plaintext information retrieval (IR) community. To apply it in the encrypted cloud data search system is a very difficult job because of intrinsic security and isolation obstacles, including various strict requirements like the data authentication, the index authentication, and the keyword authentication.

***Revathy B.D**, ME, Computer Science and Engineering, Mahendra Institute of Technology(MIT), Mallasamuram, Namakkal-637503.*

Tejaswini .B, *Assistant Professor, Computer Science and Engineering, Coorg Institute of Technology(CIT), Ponnampet-571216.*

II. RELATED WORK

In this segment we are going to discuss about the presented techniques and proposed techniques.

A. Existing System

A searchable encryption [5][9] is a supportive procedure that treats encrypted data as credentials and allows a client to firmly search through a single keyword and retrieve documents of interest. on the other hand, direct function of these approaches to the protected large scale cloud data utilization system would not be essentially suitable, as they are developed as crypto primitives and cannot contain such high service level necessities like system usability, customer searching knowledge, and effortless information detection. Even though some current designs have been projected to carry out Boolean keyword search [4][2] as an effort to improve the search elasticity, they are still not adequate to provide users with satisfactory result ranking functionality. This method has some pitfall, they are

1. It cannot provide high level system requirement like usability.
2. It is not adequate to provide search result based on ranking.
3. Sharing of data is not secured under this technique
4. It supports only single and Boolean keyword searching so it is not flexible and efficient.
5. Single keyword search often yields far too coarse result

B. Proposed System

Here we classify and resolve the difficulty of multi-keyword ranked seek over encrypted cloud data (MRSE) while authentication strict system-wise privacy in the cloud computing standard. Among various multi keyword semantics, we choose the well-organized match measure of "Automatic Annotation". For the period of the index creation, every document is connected with a binary vector as a sub index wherever each bit represents whether matching

ISSN: 2278 – 7798

keyword is enclosed in the document. The search inquiry is also described as a binary vector where each bit means whether matching keyword appears in this search request. To meet the challenge of supporting such multi-keyword semantic without privacy breaches, we suggest a basic idea for the MRSE using protected inner product calculation, which is modified from a secure k-nearest neighbor (kNN) method [5] .Objective of this technique,

1. We discover the difficulty of multi keyword ranked search over encrypted cloud data, and create a set of strict privacy necessities for such a protected cloud data consumption system.
2. We suggest two MRSE techniques based on the correspondence measure of "Annotation based search" while gathering unusual privacy requirements in two unusual threat models.
3. Systematic analysis investigating privacy and efficiency guarantees of the proposed method is given, and also experiments on the real-world dataset further show the proposed schemes indeed introduce low overhead on computation and communication

III. PROBLEM FORMULATION

A. System Model

Taking into account a cloud data hosting service concerning three different entities, as illustrated in Fig. 1 Data Proprietor Customer, Cloud server, the data proprietor has a group of data credentials F to be outsourced to the cloud server in the encrypted form C. To facilitate the searching ability over C for successful data utilization, the data proprietor, prior to Outsourcing, will initial build an encrypted searchable index I from F, and then farm out both the index I and the encrypted credentials group C to the cloud server. To search the credentials collection for t given keywords, an approved user acquire a corresponding trapdoor T through search organized mechanism, e.g., broadcast encryption [8]. Upon getting T from a data user, the cloud server is answerable to search the index I and return the equivalent set of encrypted credentials. To progress the document recovery correctness, the seek result should be ranked by the cloud server according to some grading criterion. furthermore, to reduce the communication cost, the data consumer may send an optional number k along with the trapdoor T so that the cloud server only sends back top-k credentials that are most appropriate to the search inquiry.

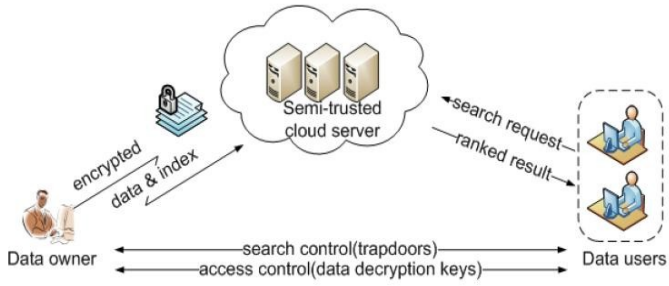

Fig. 1: Architecture of the search over encrypted cloud data

B. Threat Model

The cloud server is considered as "honest-but-curious", specifically; the cloud server acts as an "honest" fashion and correctly follows the designated protocol specification. But still it is "curious" to gather and analyze data (including index) in its storage and message flows received during the protocol so as to learn supplementary information and based on recognized information the cloud we consider two threat models with unusual attack capabilities as follows.

Identified Cipher text Representation: In this representation, the cloud server is supposed to only identify encrypted data set C and searchable index I, and both of which are outsourced from the data proprietor

Identified Background Representation: This is stronger representation, the cloud server is supposed to hold more information than what can be accessed in the identified cipher text representation. Such information may take in the association relationship of given search requirements (trapdoors), seeing that data set related statistical information.

C. Design Goals

To facilitate ranked search for successful exploitation of outsourced cloud records under the abovementioned model, our scheme should concurrently accomplish security and performance guarantees as follows.

Multi-keyword Grad Exploration: To aim search schemes which permit multi-keyword query and afford result similarity ranking for successful data recovery, as an alternative of returning undifferentiated outcome.

Privacy-Preserving: To avoid the cloud server from knowledge additional information from the data set and the index.

Competence: The exceeding goals on functionality and privacy should be achieved with low communication and computation overhead.

D. Notations

• F – The plaintext file set, represented as a set of m data documents $F = (F_1, F_2, \ldots, F_m)$.

• Cont – The encrypted record group stored in the cloud server represented as $Cont = (Cont_1, Cont_2, \ldots, Cont_m)$.

• W – The vocabulary, i.e., the keyword set consisting of n keyword, represented as $W = (word_{a1}, word_{a2}, \ldots, word_{an})$.

• I – The searchable index related with C, represented as $(I_1, I_2, \ldots, I_m)$ where every sub index I_i is built for F_i.

• f_W – The division of WW, on behalf of the keywords in a search appeal, represented as $f_W = (word_{j1}, word_{j2}, \ldots, word_{jt})$.

• Tf_W – The trapdoor used for the search appeal f_W.

• FFf_W – The ranked id index of all credentials according to their significance to f_W.

IV. STRUCTURE AND PRIVACY NECESSITIES FOR MRSE

In this segment, we describe the structure of multi-keyword ranked search over encrypted cloud data (MRSE) and establish various strict system-wise privacy requirements for such a secure cloud data utilization system.

A. MRSE STRUCTURE

For the simple appearance, operations on the data credentials are not shown in the structure since the data proprietor could simply make use of the traditional symmetric key cryptography to encrypt and then outsource data. By means of center of attention on the index and query, the MRSE structure consists of four algorithms as follows.

ISSN: 2278 – 7798

- Setup (1ℓ) Captivating a security parameter ℓ as input, the data proprietor outputs a symmetric key as SK.
- Build Index (F, SK) it's purely based on the dataset F, the data proprietor builds a searchable index I which is encrypted by the symmetric key SK and afterwards outsourced to the cloud server.
- Trapdoor (fW) by t keywords of attention in fW as input, this algorithm produces a equivalent trapdoor TfW.
- Query (TfW, k, I) while the cloud server receives a query request as (TfW, k), it executes the ranked search on the index I with the help of trapdoor TfW, and lastly returns FfW, the graded id list of top-k credentials sorted by their similarity with fW.

B. Privacy Requirements for MRSE

With the general privacy explanation, we discover and establish a set of stringent privacy necessities specifically for the MRSE framework.

Data privacy: The data proprietor can resort to the conventional symmetric key cryptography to encrypt the data before outsourcing, and successfully avoid the cloud server from snooping into the outsourced data

Index privacy: If the cloud server deduces any relationship between keywords and encrypted credentials from index, it may find out the most important subject of a manuscript, even the content of a short manuscript [26].

Keyword Privacy: since users frequently prefer to keep their search from being exposed to others like the cloud server, the majority vital concern is to hide what they are searching, i.e., the keywords indicated by the equivalent trapdoor. **Manuscript frequency** (i.e., the number of credentials containing the keyword) is adequate to recognize the keyword with high possibility.

Trapdoor Unlink ability: The trapdoor creation function should be a randomized one instead of being deterministic. In exacting, the cloud server should not be able to infer the relationship of any given trapdoors, e.g., to conclude whether the two trapdoors are produced by the same search request.

Access Guide: contained by the ranked search, the access pattern is the cycle of search results where every search result is a set of credentials with grade.

V. ALGORITHM USED

Attribute-Based Encryption (ABE) Systems

Recently, the investigation community has proposed Attribute-based Encryption (ABE) systems where encryption and decryption are strong-minded by the attributes of the data and the recipient. An ABE cryptosystem is intended to facilitate fine-grained access control of the encrypted data. It allows the encryptor to fix attributes or policies to a message being encrypted so that only the receiver(s) who is (are) assigned well-matched policies or attributes can decrypt it. Officially, the attributes can be measured as Boolean variables with random tags, and the policies are expressed as conjunctions and disjunctions of attribute variables. The ABE systems can be viewed as a simplification of Identity Based Encryption (IBE) systems. In IBE systems, only one attribute is used which is the uniqueness of the receiver, whereas ABE systems facilitate the use of numerous attributes at the same time.

In reality, present ABE schemes are built by skillfully combining the basic techniques of IBE with a linear secret distribution idea. We have two alternatives in enforcing the access policy. The access policy can be entrenched in the private key of a user, which fallout in a cryptosystem called Key Policy ABE (KP-ABE. on the other hand, the entrée policy can be fixed in the cipher text, which yields the Cipher text Policy ABE (CP-ABE) system. Mutually KP-ABE and CP-ABE systems make sure that a group of users cannot access any unauthorized data by colluding with each other.

A. Cipher text-policy ABE (CP-ABE)

The CP-ABE scheme describes, every consumer is connected with a set of attributes and her private key is generated based on these attributes. After encrypting a message M, the encryptor specifies an access configuration which is expressed in terms of a set of chosen attributes for M. The message is then encrypted based on the entrée configuration such that only those whose attributes convince this access arrangement can decrypt the message. Unlawful users are not able to decrypt the cipher text even if they collude and the access structure is sent in plaintext. A CP-ABE system consists of four algorithms:

- Setup: it's a randomized algorithm and that allows security parameter as input, and returns the public parameters *PK* and a master key *MK as result*. *PK* is used for encryption and *MK* is used to generate user secret keys and is known only to the central authority.
- Encryption: it's a randomized algorithm and we are going to pass input as message *M*, an access formation *T*, and the public parameters *PK*. It produces cipher text *CT as output*.
- KenGen: it's a randomized algorithm. Here we are going to pass input as the set of a user (say *X*)'s attributes *SX*, the master key *MK* and it produces secret key *SK as output and* that identifies with *SX*.
- Decryption: Here we are going to pass input the cipher text *CT*, a secret key *SK* for an attribute set *SX*. If SX verifies the access structure embedded in *CT*, it returns the original message *M as output*.

B. Automatic Annotation:

Automatic Annotation technique primarily line up all the data units on a result page then line up the outcome into different groups (i.e according to same data semantics). Allow $data_j^k$ represents the data part belonging to the $_j$th SRR of concept k(fig 2a). The Search Result Records (SRR) on a outcome page can be denoted in a table layout with each row representing an SRR. Automatic annotation search techniques are alienated into 3 different phases.

Placement: Recognize all data units in the SRR and then position them into different groups with each group equivalent to a dissimilar concept (e.g., all titles are grouped together). Alignment of same data units of the same semantic help to identify the frequent patterns and features among these data units (fig 2b) and it is input for annotators.

Explanation: both fundamental annotators are used to create a label for the units within their group holistically, and a

ISSN: 2278 – 7798

prospect representation is adopted to recognize the most suitable marker for each group (fig 2c and fig 3).

Annotation Binding Production: All the recognized perception, produce an annotation regulation R that describes how to take out the data units from the notion present in the outcome page and what is the suitable semantic marker should be. The regulations for all associated groups, together, form the annotation covering for the equivalent Web Database (WDB), In a straight line which can be used for annotate the data retrieved from the same WDB in reply to new queries without the need to perform the alignment and annotation phases again (fig 2d).

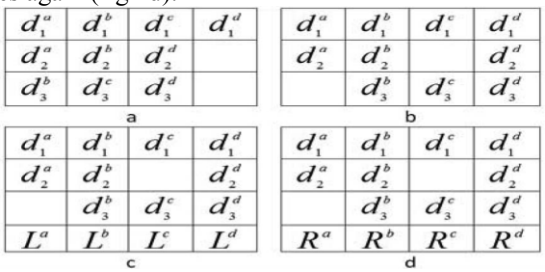

Fig (2): Phases of automatic annotation

```
ALIGN(SRRs)
1.  j ← 1;
2.  while true
      //create alignment groups
3.    for i ← 1 to number of SRRs
4.      Gj ← SRR[i][j];    //j^th element in SRR[i]
5.    if Gj is empty
6.      exit; //break the loop
7.    V ← CLUSTERING(G);
8.    if |V| > 1
        //collect all data units in groups following j
9.      S ← Ø;
10.     for x ← 1 to number of SRRs
11.       for y ← j+1 to SRR[i].length
12.         S ← SRR[x][y];
        //find cluster c least similar to following groups
13.     V[c] = min_{k=1to|V|} (sim(V[k], S));
        //shifting
14.     for k ← 1 to |V| and k ≠ c
15.       foreach SRR[x][j] in V[k]
16.         insert NIL at position j in SRR[x];
17.   j ←j+1;        //move to next group

CLUSTERING(G)
1.  V ←all data units in G;
2.  while |V| > 1
3.    best ← 0;
4.    L ←NIL; R ←NIL;
5.    foreach A in V
6.      foreach B in V
7.        if ((A != B) and (sim(A, B) > best))
8.          best ← sim(A,B);
9.          L ←A;
10.         R ←B;
11.   If best > T
12.     remove L from V;
13.     remove R from V;
14.     add L ∪ R to V;
15.   else break loop;
16. return V;
```

Fig3. Explanation algorithm

C. K-Nearest Neighbour

kNN[3] inquiry is an most significant analysis operation applied for database and it is used as a standalone query or core module for data mining. A kNN query searches is applied for K points in the database and that are the nearest to a given query point Q. Distance Preserving Transformation (DPT) supports kNN technique. *K-nearest* neighbor search classifies the top *k* adjacent neighbors to the request. This method is commonly used in extrapolative analytics to estimate or classify a point based on the consensus of its neighbors. The basic idea of our new algorithm: The value of $data_{max}$ is decreased keeping stage with the ongoing careful estimation of the object correspondence distance for the applicants. Step by step after reaching end of the, $data_{max}$ extents the best query range End_d and avoids the method from creating more applicants than essential thus fulfilling the K-optimality principle.

Nearest Neighbor Search (query, k_n) // optimal algorithm

1. Set ranking = index.increm-ranking (F(query), $data_f$)
2. Set result = new sorted-list (key, object)
3. Set $data_{max}$ = W
4. While O = ranking.getnext and data,(O, query) Item data,,, do
5. If do@, s> s $data_{max}$ then result.insert (data,(O, query) , O)
6. If result.length 2 k_n then $data_{max}$ = result[k_n].key
7. Eliminate all entries from result where key > $data_{max}$
8. End while

Shot all records from result where key Item $data_{max.}$

VI. CONCLUSION AND FUTURE SCOPE

In this manuscript, on behalf of we identify and resolve the problem of multi-keyword ranked search over encrypted cloud data, and set up a range of privacy requirements. Along with different multi-keyword semantics, we prefer the well-organized correspondence measure of "Automatic Annotation", i.e., at the same time as many matches as achievable, to successfully capture the significance of outsourced documents to the query keywords. Used for gathering the challenge of supporting multi-keyword semantic without privacy breaches, we suggest a basic idea of MRSE using secure inner product computation. Here we suggest two enhanced MRSE schemes to complete a variety of strict privacy necessities in two different threat models. Methodical study investigating privacy and efficiency guarantees of proposed method is given, and experiments on the real-world dataset show our new schemes introduce low overhead on both computation and communication.

In future, we will explore supporting other multi keyword semantics over encrypted data and inspection the integrity of the rank order in the search result. This system is currently work on single cloud, In future is will extend up to sky computing & Provide better security in multi-user.

REFERENCES

[1] L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, "A break in the clouds: towards a cloud definition," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 1, pp. 50–55, 2009.

[2] S. Kamara and K. Lauter, "Cryptographic cloud storage," in *RLCPS,January 2010, LNCS. Springer, Heidelberg*.

ISSN: 2278 – 7798

[3] Y.-C. Chang and M. Mitzenmacher, "Privacy preserving keyword searches on remote encrypted data," in *Proc. of ACNS*, 2005.
[4] R. Curtmola, J. A. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions,"
in *Proc. of ACM CCS*, 2006.
[5] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Proc. of EUROCRYPT*, 2004.
[6] D. Boneh, E. Kushilevitz, R. Ostrovsky, and W. E. S. III, "Public key encryption that allows pir queries," in *Proc. of CRYPTO*, 2007.
[7] R. Brinkman, "Searching in encrypted data," in *University of Twente, PhD thesis*, 2007.
[8] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption," in *Proc. of EUROCRYPT*, 2010.
[9] C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, "Secure ranked keyword search over encrypted cloud data," in *Proc. of ICDCS'10*, 2010.
[10] W. K. Wong, D. W. Cheung, B. Kao, and N. Mamoulis, "Secure knn computation on encrypted databases," in Proceedings of the 35th SIGMOD international conference on Management of data, 2009, pp. 139–152.

ISSN: 2278 – 7798