A Secure Search Scheme Of Encrypted Data on Mobile cloud
Prashanth Jayaraman¹, Praveen Jayasankar², Rachel Hannah³
¹, ², ³ Department Computer Science and Engineering
¹, ² Meenakshi Sundararajan Engineering College, Chennai
³ St. Joseph's College of Engineering, Chennai-119

Abstract:
As mobile cloud computing become more flexible & effective in terms of economy, data owners are motivated to outsource their complex data systems from local sites to commercial public mobile cloud. But for security of data, sensitive data has to be encrypted before outsourcing, which overcomes method of traditional data utilization based on plaintext keyword search. Due to the increasing popularity of mobile cloud computing, more and more data owners are motivated to outsource their datable mobile cloud servers for great convenience and reduced cost in data management. However, sensitive data should be encrypted before outsourcing for privacy requirements, which obsoletes data utilization like keyword-based document retrieval. In this paper, we present a secure multi-keyword ranked search scheme over encrypted mobile cloud data, which simultaneously supports dynamic update operations like deletion and insertion of documents. Considering the large number of data users and documents in mobile cloud, it is necessary for the search service to allow multi-keyword query and provide result similarity ranking to meet the effective data retrieval need. Retrieving of all the files having queried keyword will not be affordable in pay per use mobile cloud paradigm. In this paper, we propose the problem of Efficient Mobile Multikeyword search (EMMS) over encrypted mobile cloud data (ECD), and construct a group of privacy policies for such a secure mobile cloud data utilization system. From number of multi-keyword semantics, we select the highly efficient rule of coordinate matching, i.e., as many matches as possible, to identify the similarity between search query and data, and for further matching we use inner data correspondence to quantitatively formalize such principle for similarity measurement. Searchable encryption allows one to upload encrypted documents on a remote honest-but-curious server and query that data at the server itself without requiring the documents to be decrypted prior to searching. In this work, we first propose a basic Secured multi keyword ranked search scheme using secure inner product computation, and then improve it to meet different privacy requirements. The Ranked result provides top retrieval results. Due to the use of our special structure, the proposed scheme can achieve sub-linear search time and deal with the deletion and insertion of documents flexibly. Extensive experiments are conducted to demonstrate the efficiency of the proposed scheme. Also we propose an alert system which will generate alerts when unauthorized user tries to access the data from mobile cloud, the alert will generate in the form of mail and message.

Keywords — Multikeyword, secure search, mobile cloud, encrypted data, encrypted mobile cloud, mobile cloud privacy, data retrieval, outsourcing, server, mobile cloud connection.

I. INTRODUCTION

Mobile cloud computing enables new types of services where the computational and network resources are available online through the Internet. One of the most popular services of mobile cloud computing is data outsourcing. For reasons of cost and convenience, public as well as private organizations can now outsource their large amounts of data to the mobile cloud and enjoy the benefits of remote storage. At the same time, confidentiality of remotely stored data on untrusted mobile cloud server is a big concern. In order to reduce these concerns, sensitive data, such as,
personal health records, emails, income tax and financial reports, etc. are usually outsourced in encrypted form using well-known cryptographic techniques. Although encrypted data storage protects remote data from unauthorized access, it complicates some basic, yet essential data utilization services such as plaintext keyword search. A simple solution of downloading the data, decrypting and searching locally is clearly inefficient since storing data in the mobile cloud is meaningless unless it can be easily searched and utilized. Thus, mobile cloud services should enable efficient search on encrypted data to provide the benefits of a first-class mobile cloud computing environment.

Despite of the various advantages of mobile cloud services, outsourcing sensitive information (such as e-mails, personal health records, company finance data, government documents, files, etc.) to remote servers brings privacy concerns. The mobile cloud service providers that keep the data for users may access users sensitive information without authorization. In the literature, searchable encryption techniques [2-4] are able to provide secure search over encrypted data for users. They build a searchable inverted index that stores a list of mappings from keywords to the corresponding set of files which contain this keyword. When data users input a keyword, a trapdoor is generated for this keyword and then submitted to the mobile cloud server. Upon receiving the trapdoor, the mobile cloud server executes on this keyword. But, these methods only allow exact single keyword search. Some researchers study the problem on secure and ranked search over outsourced mobile cloud data. Wang et al., [5] propose a secure ranked keyword search scheme. Their solution combines inverted index with order-preserving symmetric encryption (OPSE). Interns of ranked search, the order of retrieved files is determined by numerical relevance scores, which can be calculated by TF×IDF. The relevance score is encrypted by OPSE to ensure security. It enhances system usability and saves communication overhead. This solution only supports single keyword ranked search. Cao et al., [6] propose a method that adopts similarity measure of “coordinate matching” to capture the relevance of files to the query. They use “inner product similarity” to measure the score of each file. This solution supports exact multi-keyword ranked search. It is practical, and the search is flexible. Sun et al., [7] proposed a MDB-tree based scheme which supports ranked multi-keyword search.

This scheme is very efficient, but the higher efficiency will lead to lower precision of the search results in this scheme. In addition, fuzzy keyword search [8-10] have been developed. These methods employ a spell-check mechanism, such as, search for “wireless” instead of “wirelecss”, or the data format may not be the same e.g., “data-mining” versus “datamining. Chuah et al., [8] propose a privacy-aware bed-tree method to support fuzzy multi-keyword search. This approach uses edit distance to build fuzzy keyword sets. Bloom filters are constructed for every keyword. Then, it constructs the index tree for all files where each leaf node a hash value of a keyword. Li et al., [9] exploit edit distance to quantify keywords similarity and construct storage-efficient fuzzy keyword sets. Specially, the wildcard-based fuzzy set construction approach is designed to save storage overhead. Wang et al., [10] employ wildcard-based fuzzy set to build a private tree-traverse searching index. In the searching phase, if the edit distance between retrieval keywords and ones from the fuzzy sets is less than a predetermined set value, it is considered similar and returns the corresponding files. These fuzzy search methods support tolerance of minor typos and format inconsistencies, but do not support semantic fuzzy search. Considering the existence of polysemy and synonymy [11], the model that supports multi-keyword ranked search and semantic search is more reasonable. Comparison between the trapdoor and index, and finally returns the data users all files. A general approach to protect the data confidentiality is to encrypt the data before outsourcing.

However, this will cause a huge cost in terms of data usability. For example, 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 mobile cloud and decrypt locally is obviously impractical. In order to address the above problem, researchers have designed some general-purpose solutions with fully-homomorphic
encryption or oblivious RAMs. However, these methods are not practical due to their high computational overhead for both the mobile cloud server and user. On the contrary, more practical special purpose solutions, such as searchable encryption (SE) schemes have made specific contributions in terms of efficiency, functionality and security. However, we note that “fine-grained search authorization” is an indispensable component for a secure data outsourcing system. Although the accesses to actual documents can be controlled by separate cryptographic enforced access control techniques such as attribute-based encryption [9], [39], [26], “0-1” search authorization may still lead to leakage of data owners’ sensitive information. For example, if Alice is the only patient with a rare disease in a PHR database, by designing the query in a clever way (e.g., submitting two queries with/without the name of that disease and with Alice’s demographic info), from the results a user Bob will be certain that Alice has that disease. Thus, we argue that a user should only be allowed to search for some specific sets of keywords; in particular, the authorization shall be based on a user’s attributes. For instance, in a patient matching application in health social networks [28], [23], a patient should only be matched to patients having similar symptoms as her, while shall not learn any information about those who do not. Furthermore, system scalability is an important concern for SE. For symmetric-key based SE schemes, the encryption and search capabilities are not separable, so a multi-owner system would require every owner to act as a capability distribution center, which is not scalable. PKC-based schemes do not have this problem, but if every user obtains restricted search capabilities from a central trusted authority (TA) who assumes the responsibility of authorization at the same time, it shall be always online, dealing with large workload, and facing the threat of single-point-of-failure. In addition, since the global TA does not directly possess the necessary information to check the attributes of users from different local domains, additional infrastructure needs to be employed (such as using a credential chain [27]). It is therefore desirable for the user to be authorized locally.

Searchable encryption schemes enable the client to store the encrypted data to the mobile cloud and execute keyword search over ciphertext domain. So far, abundant works have been proposed under different threat models to achieve various search functionality, such as single keyword search, similarity search, multi-keyword boolean search, ranked search, multi-keyword ranked search, etc. Among them, multi-keyword ranked search achieves more and more attention for its practical applicability. Recently, some dynamic schemes have been proposed to support inserting and deleting operations on document collection. These are significant works as it is highly possible that the data owners need to update their data on the mobile cloud server. But few of the dynamic schemes support efficient multikeyword ranked search.

In order to obtain high search efficiency with the special structure of our data index, the proposed search scheme can flexibly achieve sub-linear search time and deal with the deletion and insertion of documents. We construct a tree-based index structure and propose a “Greedy Depth-first Search” algorithm based on this index tree. The secure kNN algorithm is utilized to encrypt the index and query vectors, and meanwhile ensure accurate relevance score calculation between encrypted index and query vectors. To resist different attacks in different threat models, we construct two secure search schemes: the basic dynamic multi-keyword ranked search scheme in the known cipher text model, and the enhanced dynamic multi-keyword ranked search scheme in the background model. Our contributions are summarized as follows:

a) Due to the special structure of our data index, the search complexity of the proposed scheme is fundamentally kept to logarithmic. And in practice, the proposed scheme can achieve higher search efficiency by executing our “Greedy Depth-first Search” algorithm. Moreover, parallel search can be flexibly performed to further reduce the time cost of search process.

b) We design a searchable encryption scheme that supports both the accurate multi-keyword ranked search and flexible dynamic operation on document collection.
II. RELATED WORK

Organizations, companies store more and more valuable information is on cloud to protect their data from virus, hacking. The benefits of the new computing model include but are not limited to: relief of the trouble for storage administration, data access, and avoidance of high expenditure on hardware mechanism, software, etc. Ranked search improves system usability by normal matching files in a ranked order regarding to certain relevance criteria (e.g., keyword frequency). As directly outsourcing relevance scores will drip a lot of sensitive information against the keyword privacy. We proposed asymmetric encryption with ranking result of queried data which will give only expected data.

Searchable encryption has been an active research area and many quality works have been published [1–6, 9–13, 16, 20–23, 25, 26]. Traditional searchable encryption schemes usually build an encrypted searchable index such that its content is hidden to the server, however it still allows performing document searching with given search query. Song et al. [23] were the first to investigate the techniques for keyword search over encrypted and outsourced data. The authors begin with idea to store a set of plaintext documents on data storage server such as mail servers and file servers in encrypted form to reduce security and privacy risks. The work presents a cryptographic scheme that enables indexed search on encrypted data without leaking any sensitive information to the untrusted remote server. Goh [16] developed a per-file Bloom filter-based secure index, which reduce the searching cost proportional to the number of files in collection. Recent work by Moataz et al. [21] proposed boolean symmetric searchable encryption scheme. Here, the scheme is based on the orthogonalization of the keywords according to the Gram-Schmidt process. Orencik’s solution [22] proposed privacy-preserving multi-keyword search method that utilizes minhash functions. Boneh et al. [6] developed the first searchable encryption using the asymmetric settings, where anyone with the public key can write to the data stored remotely, but the users with private key execute search queries. The other asymmetric solution was provided by Di Crescenzo et al. in [11], where the authors propose a public-key encryption scheme with keyword search based on a variant of the quadratic residuosity problem.

All secure index based schemes presented so far, are limited in their usage since they support only exact matching in the context of keyword search. Wang et al. [25] studied the problem of secure ranked keyword search over encrypted cloud data.

The authors explored the statistical measure approach that embeds the relevance score of each document during the establishment of searchable index before outsourcing the encrypted document collection. The authors propose a single keyword searchable encryption scheme using ranking criteria based on keyword frequency that retrieves the best matching documents. Cao et al. [9] presented a multi-keyword ranked search scheme, where they used the principle of "coordinate matching" that captures the similarity between a multi-keyword search query and data documents. However, their index structure is uses a binary representation of document terms and thus the ranked search does not differentiate documents with higher number of repeated terms than documents with lower number of repeated terms.

Searchable encryption schemes enable the clients to store the encrypted data to the cloud and execute keyword search over ciphertext domain. Due to different cryptography primitives, searchable encryption schemes can be constructed using public key based cryptography [5], [6] or symmetric key based cryptography [7], [8], [9], [10]. Song et al. [7] proposed the first symmetric searchable encryption (SSE) scheme, and the search time of their scheme is linear to the size of the data collection. Goh [8] proposed formal security definitions for SSE and designed a scheme based on Bloom filter. The search time of Goh’s scheme is O (n), where n is the cardinality of the document collection. Curtmola et al. [10] proposed two schemes (SSE-1 and SSE-2) which achieve the optimal search time. Their SSE-1 scheme is secure against chosen-keyword attacks (CKA1) and SSE-2 is secure against adaptive chosen-keyword attacks (CKA2). These early works are single keyword boolean search schemes, which are very simple in terms of functionality. Afterward, abundant works have been proposed under different threat models to achieve various search functionality, such as single keyword search,
similarity search[11], [12], [13], [14], multi-keyword boolean search [15],[16], [17], [18], [19], [20], [21], [22], ranked search [23],[24], [25], and multi-keyword ranked search [26], [27],[28], [29], etc.Multi-keyword boolean search allows the users to input multiple query keywords to request suitable documents. Among these works, conjunctive keyword searcheschemes [15], [16], [17] only return the documents that contain all of the query keywords. Disjunctive keywordsearch schemes [18], [19] return all of the documents that contain a subset of the query keywords. Predicatesearch schemes [20], [21], [22] are proposed to support both conjunctive and disjunctive search. All these multi-keyword search schemes retrieve search results based on the existence of keywords, which cannot provide acceptable result ranking functionality. Ranked search can enable quick search of the most relevant data. Sending back only the top-
k most relevant documents can effectively decrease network traffic. Someearly works [23], [24], [25] have realized the ranked search using order-preserving techniques, but they are designed only for single keyword search. Cao et al.[26] realized the first privacy-preserving multi-keyword ranked search scheme, in which documents and queries are represented as vectors of dictionary size. With the “coordinate matching”, the documents are ranked according to the number of matched query keywords. However, Cao et al.’s scheme does not consider the importance of the different keywords, and thus is not accurate enough. In addition, the search efficiency of the scheme is linear with the cardinality of document collection. Sun et al. [27] presented a secure multi-keywordsearch scheme that supports similarity-based ranking. The authors constructed a searchable index tree based on vector space model and adopted cosine measure together with TF×IDF to provide ranking results. Sun et al.’s search algorithm achieves better-than-linear search efficiency but results in precision loss."

Orencik et al. [28] proposed a secure multi-keyword search method which utilized local sensitive hash (LSH) functions to cluster the similar documents. The LSH algorithm is suitable for similar search but cannot provide exact ranking. In[29], Zhang et al. proposed a scheme to deal with secure multi-keyword ranked search in a multi-owner model. In this scheme, different data owners use different secret keys to encrypt their documents and keywords while authorized data users can query without knowing keys of these different data owners. The authors proposed an “Additive Order Preserving Function” to retrieve the most relevant search results. However, these works don’t support dynamic operations.

Practically, the data owner may need to update the document collection after he upload the collection to the cloud server. Thus, the SE schemes are expected to support the insertion and deletion of the documents. There are also several dynamic searchable encryption schemes. In the work of Song et al. [7], the each document is considered as a sequence of fixed length words, and is individually indexed. This scheme supports straightforward update operations but with low efficiency. Goh[8] proposed a scheme to generate a sub-index (Bloom filter) for every document based on keywords. Then the dynamic operations can be easily realized through updating of a Bloom filter along with the corresponding document. However, Goh’s scheme has linear search time and suffers from false positives. In 2012, Kamara et al. [30] constructed an encrypted inverted index that can handle dynamic data efficiently. But, this scheme is very complex to implement. Subsequently, as an improvement, Kamara et al. [31] proposed a new search scheme based on tree-based index, which can handle dynamic update on document data stored in leaf nodes. However, their scheme is designed only for single keyword Boolean search. In [32], Cash et al. presented a data structure for keyword/identity tuple named “TSet”. Then, a document can be represented by a series of independent T-Sets. Based on this structure, Cash et al.[33] proposed a dynamic searchable encryption scheme. In their construction, newly added tuples are stored in another database in the cloud, and deleted tuples are recorded in a revocation list. The final search result is achieved through excluding tuples in the revocation list from the ones retrieved from original and newly added tuples. Yet, Cash et al.’s dynamic search scheme doesn’t realize the multi-keyword ranked search functionality.

III. PROBLEM BASE
A. System Model

The system model can be considered as three entities, the data owner, the data user and the cloud
server. Before you begin to format your paper, first write and save the content as a separate text file. Keep your text and graphic files separate until after the text has been formatted and styled. Do not use hard tabs, and limit use of hard returns to only one return at the end of a paragraph. Do not add any kind of pagination anywhere in the paper. Do not number text heads—the template will do that for you.

Finally, complete content and organizational editing before formatting. Please take note of the following items when proofreading spelling and grammar:

- **Data Owner**
  
  Data owner has a collection of data documents A set of distinct keywords is extracted from the data collection. The data owner will firstly construct an encrypted searchable index I from the data collection D. All files in D are encrypted and form a new file collection, C. Then, the data owner upload both the encrypted index I and the encrypted data collection C to the cloud server.

- **Data User**
  
  **Data user** provides t keywords for the cloud server. A corresponding trapdoor through search control mechanisms is generated. In this paper, we assume that the authorization between the data owner and the data user is approximately done.

- **Cloud Server**
  
  **Cloud server** received from the authorized user. Then, the cloud server calculates and returns to the corresponding set of encrypted documents. Moreover, to reduce the communication cost, the data user may send an optional number along with the trapdoor T so that the cloud server only sends back top-l files that are most relevant to the search query.

D. Threat models and Design Goals

The cloud server is considered as “honest-but-curious” in our model. Particularly, the cloud server both follows the designated protocol specification but at the same time analyzes data in its storage and message flows received during the protocol so as to learn additional information [12]. In this paper, we purpose to achieve security and ranked search under the above model. The designed goals of our system are following:

- **Latent Semantic Search**: We aim to discover the latent semantic relationship between terms and documents. We use statistical techniques to estimate the latent semantic structure, and get rid of the obscuring “noise” [11]. The proposed scheme tries to put similar items near each other in some space in order that it could return the data user the files contain the terms latent semantically associated with the query keyword.

- **Multi-keyword Ranked Search**: It supports both multi-keyword query and support result ranking.

- **Privacy-Preserving**: Our scheme is designed to meet the privacy requirement and prevent the cloud server from learning additional information from index and trapdoor.
  
  1) **Index Confidentiality**. The TF values of keywords are stored in the index. Thus, the index stored in the cloud server needs to be encrypted;
  
  2) **Trapdoor Unlinkability**. The cloud server could do some statistical analysis over the search result. Meanwhile, the same query should generate different trapdoors when searched twice. The cloud server should not be able to deduce relationship between trapdoors.
  
  3) **Keyword Privacy**. The cloud server could not discern the keyword in query, index by analyzing the statistical information like term frequency.

B. Threat models and Design Goals

The cloud server is considered as “honest-but-curious” in our model. Particularly, the cloud server both follows the designated protocol specification but at the same time analyzes data in its storage and message flows received during the protocol so as to learn additional information [12]. In this paper, we purpose to achieve security and ranked search under the above model. The designed goals of our system are following:
**Latent Semantic Search:** We aim to discover the latent semantic relationship between terms and documents. We use statistical techniques to estimate the latent semantic structure, and get rid of the obscuring “noise” [11]. The proposed scheme tries to put similar items near each other in some space in order that it could return the data user the files contain the terms latent semantically associated with the query keyword.

**Multi-keyword Ranked Search:** It supports both multi-keyword query and support result ranking.

**Privacy-Preserving:** Our scheme is designed to meet the privacy requirement and prevent the cloud server from learning additional information from index and trapdoor.

- **1) Index Confidentiality.** The TF values of keywords are stored in the index. Thus, the index stored in the cloud server needs to be encrypted;
- **2) Trapdoor Unlink ability.** The cloud server could do some statistical analysis over the search result. Meanwhile, the same query should generate different trapdoors when searched twice. The cloud server should not be able to deduce relationship between trapdoors.
- **3) Keyword Privacy.** The cloud server could not discern the keyword in query, index by analyzing the statistical information like term frequency.

**3. Proposed Scheme**

In this section, we give a detailed description of our scheme. We firstly propose to employ “Latent Semantic Analysis” to implement the latent semantic multi-keyword ranked search.

**A. Our Scheme**

Data owner wants to outsource $m$ data files that he prepares to outsource to the cloud server in encrypted form while still keeping the capability to search $m$ through them(Fig 3.1). To do so, data owner firstly builds a secure searchable index from a set of $n$ distinct key words extracted from the file collection $D$.

According to the above definition about $LSA$, the data owner builds a term-document matrix $A$. Matrix $A$ can be decomposed into the product of three other matrices. And then, we reduce the dimensions of the original matrix $A$ to get a new matrix $A'$ which is calculated the best “reduced-dimension” approximation to the original term-document matrix [16]. With $t$ keywords of interest in as input, one binary vector $Q$ is generated where each bit indicates whether $W$ is true or false. The similarity score is expressed as the inner product of data vector $jA$ and query vector $Q$. Specially, $[A/j]$ denotes the $j-$th column of the matrix $A$.

**Setup**

The data owner generates a $2n$-bit vector as $X$ and two $(2n)$ invertible matrices $M$, $M'$. The secret key $SK$ is the form of a 3-tuple as $M^\perp$.

**Build Index**

The data owner extracts a term-document matrix $A$ from $D$ and decomposes it into three other matrices $U$, $V$. According to our scheme, the data owner adopts statistical techniques to estimate the latent structure, and get rid of the obscuring “noise”. To reduce dimensions, we choose previous $tm\times k$ columns of $S$, and then deleting the corresponding columns of $U$ and $V$ respectively. Following, we multiply these three matrices.

**4. Performance and Security Analysis**

In this section, we show a thorough experimental evaluation of the proposed technique on a real dataset: the $MED$ dataset [17]. The whole experiment is implemented by C++ language on a computer with Core 2.83GHz Processor, on Windows 7 system. For the proposed scheme, we will reduce to separate dimensions. The performance of our method is compared with the original MRSE scheme.

**D. Effectiveness and Efficiency**

The proposed scheme is depicted in details in previous section, except the Key Gen algorithm. In
our scheme, we adopt Gauss-Jordan to compute the inverse matrix. The time of generating key is decided by the scale of the matrix. Besides, the proposed scheme that processed by SVD algorithm will consume time. Other algorithms, such as index construction, trapdoor generation, query, which is put forward by us, are consistent with the original MRSE in time-consuming.

1) Index and Trapdoor Generation: The index generation process is a one-time computation which contains two major steps: the Bloom filter generation and the encryption. During the Bloom filter generation, the computation mainly comes from the hash function calculation. Figure 3.2 shows the Bloom filter generation for the search index and the trapdoor Bloom filter. The generation time increases linearly respect to the number of the inserted keywords. The trapdoor generation time is very close to the index generation time due to the identical procedure. The encryption time which involves the matrix multiplications is showed in figure 3.(3). The time cost of encryption increases linearly respect to the number of the files in the dataset.

2) Search over Encrypted Index: The search operation executed at the cloud server side consists of computing the inner product calculation for all the files in the dataset. Figure 3.(c) shows the search time grows linearly with the size of the file set while the number of keywords in the query has little impact as showed in figure 3.(d). This is intuitive because the search process needs to go over all the files in the dataset before the cloud server can get the final result. The inner product computation is only related to the length of the index, so the computation time changes little in figure 3.(d).

E. F-measure of the scheme:
In this paper, we still use the measure of traditional information retrieval. Before the introduction of the F-measure’s concept, we will firstly give the brief of the precision and recall. Precision is the fraction of retrieved instances that are relevant, while recall is the fraction of relevant instances that are retrieved. Both precision and recall are therefore based on an understanding and measure of relevance[18]. F-measure that combines precision and recall is the harmonic mean of precision and recall[19]. Here, we adopt F-measure to weigh the result of our experiments.

B. Result Accuracy
We adopt the definitions of the widely used performance metrics, precision and recall to measure the search result accuracy. Denote tp as true positive, fp as false positive and fn as false negative, then the precision equals to while the recall is tp / tp+fn. To generate the fuzzy queries, we randomly pick two keywords and modify them into the fuzzy keywords. Figure 4.(a) shows the performance metrics of our scheme according to k. Note that there is no recall for the exact matching because the false negative doesn’t exist. One observation is that precision is very low when k is small, i.e. 5% at k = 1. Because that multiple LSH functions are used together to enlarge the gap between p1 and p. So when the k is small, the gap is not big enough to distinguish the different keywords, and most of the files in the dataset have been returned, which leads to high fp and low fn. The jump at k = 5 is due to the gap between p1 and p2 increases exponentially respect to k. After a certain k, i.e., k = 8, the precision is remained at a high level, which is above 90% for the exact search.
and above 80% for the fuzzy search. Another observation is that the recall drops when increasing the k. This is because that increasing the k will cause more false negatives. In general, the false positive and the false negative cannot be improved at the same time. Another important parameter is the number of the keywords in the query. Figure 4.(b) shows the precision of the exact match decreasing slightly, from 100% to 96% while the number of the keywords in the query increases from 1 to 10. This is reasonable because the false positive generated by each keyword accumulates. But the precision for the fuzzy search doesn’t show the same pattern. It is slightly increased from 70% to 81% when the number of the keywords in the query increases from 1 to 10. The reason is that the false positive caused by the LSH functions contributes much more.

I. ALGORITHMS USED

A. RSA Algorithm

This algorithm is used to encrypt and decrypt file contents. It is an asymmetric algorithm. 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:

1. Choose two distinct prime numbers a and b.
2. Compute n = ab. n is used as the modulus for both the public and private keys.
3. Compute f(n) = (a − 1)(b − 1), where f is Euler's totient function.
4. Choose an integer e such that 1 < e < f(n) and greater common divisor of (e, f(n)) = 1; i.e., e and f(n) are co prime. e is released as the public key exponent. Having a short bit-length.

Encryption
Person A transmits her public key to Person B and keeps the private key secret. Person B then wishes to send message M to Person A. 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. Person B then transmits to Person A. Note that at least nine values of m could yield a cipher text c equal to m, but this is very unlikely to occur in practice.

Decryption
Person A can recover from by using her private key exponent via computing . She can recover the original message M by reversing the padding scheme. (In practice, there are more efficient methods of calculating using the pre computed values below.)

C. Predicate Privacy in Searchable Encryption
Shen et al [32] proposed a SKC-based predicate encryption scheme with predicate privacy. However, in a SKC-based scheme with an encryption key one can generate any search capabilities, which is undesirable for search authorization. Under public key setting, [4] proposed a key refreshing solution to randomize the original keywords used in generating the trapdoors in PEKS, but that requires a user to share a secret with all potential encryptors. In contrast, in our enhanced solution some secrets are kept by proxies, thus it does not incur any interaction between owners and users.

VI. CONCLUSION AND FUTURE SCOPE

Thus we proposed the problem of multiple-keyword ranked search over encrypted cloud data, and construct a variety of security requirements. From various multi keyword concepts, we choose the efficient principle of coordinate matching. We first...
propose secure inner data computation. Also we achieve effective ranking result using our technique. This system is currently work on single user mobile cloud, In future is will extended up to big computing & Provide better security for multi-user systems.

REFERENCES
[1] Ning Caoy, Cong Wangz, Ming Liy, Kui Renz, and Wenjing LouyPrivacy-Preserving Multi-keyword Ranked Search over Encrypted Cloud Data
[2] Weifeng Su, Jiying Wang, and Frederick H. Lochovsky, Member, IEEE Computer Society Record Matching over Query Results from Multiple Web Databases
[4] Cong Wang†, Ning Cao‡, Jin Li†, Kui Ren†, and Wenjing Lou‡†Department of ECE, Illinois Institute of Technology, Chicago, IL 60616
†Department of ECE, Worcester Polytechnic Institute, Worcester, MA 01609 Secure Ranked Keyword Search over Encrypted Cloud Data