Efficient Secured Search over Outsourced Encrypted Mobile Device Data in Cloud using the Proposed RFMKS Method

7.1 Introduction

To ensure the security of the outsourced mobile device data over cloud storage, a hybrid cryptographic MSAES algorithm is explained in chapter 6. In recent years cloud computing is gaining much popularity. Cloud has virtually unlimited data storage capabilities with elastic resource provisioning. Both individuals and enterprises are motivated to outsource their data to the cloud storage server to reduce cost of management. To prevent unauthorized access in the cloud, sensitive data should be secured by the data owners before outsourcing to the commercial public cloud. Encryption is the most effective way to achieve data security. The data which is to be outsourced over cloud storage is encrypted using the MSAES algorithm for data confidentiality and privacy protection. Another important issue in cloud is efficient data retrieval.

Sensitive data have to be encrypted before outsourcing in spite of the fact that, retrieval of encrypted data becomes an intriguing task. Retrieval of all the data and decrypting locally is not practical, as it results in huge amount of bandwidth cost in cloud scale systems. Retrieval of data demands the protection of keyword privacy since keywords usually contain important information related to the data files. Thus, exploring privacy preserving and effective search scheme over encrypted cloud data is essential.

In this chapter a novel approach namely, Random Fuzzy Multi keyword search scheme (RFMKS) which is a combination of Jaro Wrinkler algorithm, Sort Sorted Index Generation and Fast Ranking are explained**.** It is used to retrieve the encrypted data over cloud. Experimental analysis on dataset shows that the novel scheme can achieve security, efficiency, accuracy, and keyword fuzzy searching. The proposed RFMKS method enhances the user searching experience by returning the matching files when user's input query exactly matches the predefined keyword dictionary or closer possible keywords in the dictionary. Information discovery has been made efficient by searching with multiple keywords and also ranking eliminates false positives matches.

7.2 Steps of Proposed Contribution Four - RFMKS Method

The objective of the contribution four is to enhance user searching experience and fuzzy rank based retrieval of data over the cloud storage.

The proposed contribution RFMKS method consists of the following two steps are discussed below.

- Pre-Framework Phase
- Retrieval Phase

To improve the data discovery and the user's search experience on outsourced encrypted mobile device data, a new approach called RFMKS Method is proposed. Figure 7.1 shows the steps of proposed contribution RFMKS Method.

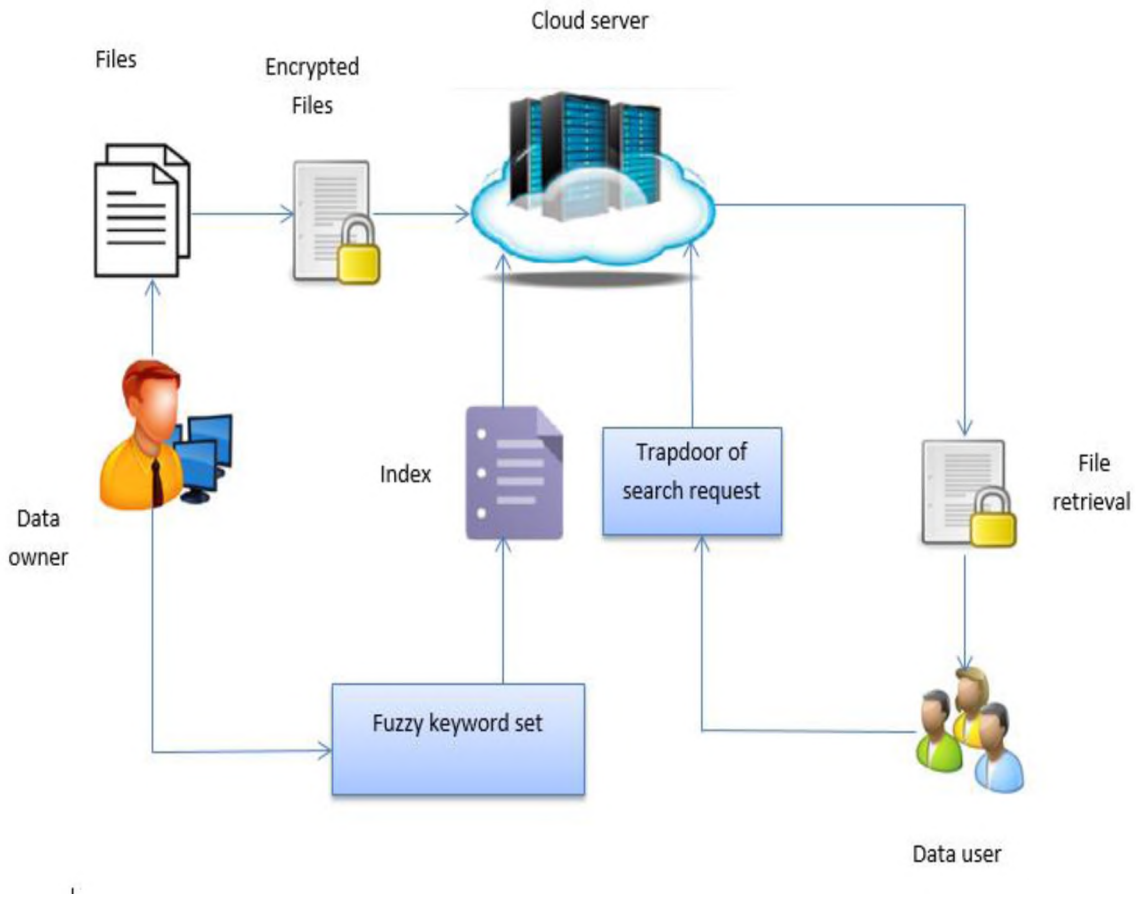

Figure 7.1 Block Diagram of Proposed Contribution Four -RFMKS Method

7.2.1 Pre-Framework Phase

Considering the large number of data users and documents in the cloud storage, it is necessary to allow multiple keywords in the search request and return documents in the order of their relevance to these keywords. To address the problems such as complexity, efficiency overhead, and searching time for multi-keyword ranked queries and index generation time over an encrypted cloud storage, the RFMKS method is proposed.

7.2.1.1 System Model

The cloud server is achieved by considering a cloud data hosting service involving three different entities such as the Mobile device data owner, the Mobile device data user, and the cloud server. The data contains sensitive information which are essential. As the cloud servers cannot be completely trusted to protect data, the files must be encrypted before outsourcing. The encryption of mobile device data over cloud storage is discussed in detail in Chapter 6.

Mobile Device Data owner

The data owner has a collection of data documents F= { f_1, f_2, f_3….f_m} to be outsourced to the cloud server in the encrypted form 'C'. A set of distinct keywords W = {w_1, w_2 , ..., w_n } is extracted from the data collection 'D' . To enhance the searching capability over 'C' for effective data utilization, the data owner, will build an encrypted searchable index 'I' from 'F', and then outsource both the index 'I' and the encrypted document collection 'C' to the cloud server. The document collection 'C' is referred to both labelled and unlabeled documents. Each file in the collection of the dataset is encrypted with contribution three MSAES Method.

Mobile Device Data user

To search the document collection for't' given keywords, an authenticated user requires a corresponding trapdoor 'T' through search control mechanisms. Authorization between the data owner and the data user is done. Then the server searches the index, and returns the matching files to the user in order.

Cloud Server

The Cloud Service Provider (CSP), has significant storage space and computation resources to maintain the clients' data. In the cloud paradigm, by transferring the large data files on the remote servers, the clients can be relieved of the burden of storage and computation. Upon receiving 'T' from a data user, the cloud server is responsible to search the index 'I' and return the corresponding set of encrypted documents. Ranking is done to improve the document retrieval accuracy, the search result should be ranked by the cloud server according to some ranking criteria. The fuzzy keyword search, reduces the communication cost, the data user may send an optional number 'k' along with the

trapdoor 'T' so that the cloud server only sends back top-k documents that are most relevant to the search query.

7.2.1.2 Threat Model

Addressing the issues related to vulnerabilities of the system model is discussed. In this particular scenario, the major threat is to ensure the trustworthiness of the cloud service provider. A semi-trusted server is implemented for the trustworthiness issue. Even though data files are encrypted, the cloud server may try to derive other sensitive information from users' search requests while performing keyword-based search over 'C'. Thus, the search should be conducted in a secure manner that allows data files to be securely retrieved while revealing as little information as possible to the cloud server. Assume that cloud server C correctly follows the designated protocol but is curious to derive sensitive information from user's search requests, while performing keyword based search over C. So, retrieval of data in a secure manner is a challenging task while revealing a minimum information to the cloud server C.

7.2.1.3 Design Goals

To enable effective and secured ranked search over outsourced encrypted mobile device data over cloud storage, the proposed mechanism should achieve the following design goals.

- **Authorized Keyword Search**

  The secure search scheme should enable data- owner enforced search authorization. Besides achieving fine-grained authorization, another challenge is to make the scheme scalable for dynamic cloud environment.

- **Multiple Keyword Search**

  It supports both multi-keyword query and support result ranking. To design search schemes which allow multi-keyword query and provide result similarity ranking for effective data retrieval, instead of returning undifferentiated results.

- **Privacy-Preserving**

  Privacy concerns arise whenever sensitive data is outsourced to the cloud. By using encryption, the cloud server is prevented from learning content in the outsourced databases.

- **Efficiency**

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

7.2.2 Retrieval Phase

In retrieval phase, the RFMKS method is proposed to construct efficient and secured ranking based fuzzy multi keyword search over cloud storage for the improvement of user searching experience and information discovery. The detailed description of the proposed method is explained below.

7.2.2.1 Build Index

The set of trapdoors for each keyword is generated and stored in an index file. The sort-sorted index generation is implemented in proposed approach. The index file is a n×m matrix where 'n' is the number of keywords in index file and 'm' is the maximum number of trapdoors of a keyword depending on occurrence of a particular keyword in file set F= { f_1, f_2, f_3….f_m}. Finally, both set of encrypted files E= (E_1, E_2, E_3... E_m) and encrypted index are uploaded on to the cloud server 'C'.

7.2.2.2 Fuzzy Multi-Keyword Search

The fuzzy keyword search is implemented in order to overcome the shortcomings of traditional searching schemes. Although traditional searchable encryption schemes allow a user to securely search over encrypted data through keywords and selectively retrieve files of interest. These techniques support only exact keyword search. The fuzzy keyword search takes place in accordance with the constructed fuzzy keyword set. The Jaro Wrinkler method is implemented for fuzzy multi keyword search. It can be defined as shown in equation 7.1 for the two strings 'a_1' and 'a_2', d_j is the jaro distance for the strings.

$$d_j = \begin{cases} 0 & \text{if } m = 0 \\ \frac{1}{3}\left(\frac{m}{|s1|} + \frac{m}{|s2|} + \frac{m-t}{m}\right) & \text{otherwise} \end{cases} \qquad (7.1)$$

where 'm' is matching characters, 't' is transpositions. Jaro–Winkler distance uses a prefix scale 'p' which gives more favourable ratings to strings that match from the beginning for a set prefix length l, two strings 's1' and 's2', their Jaro–Winkler distance 'd_w' given in equation 7.2 as,

$$d_w = d_j + \left(l_p(1 - d_j)\right) \quad (7.2)$$

where 'd_j' is jaro distance for strings 's1' and 's2', 'l' is the length of common prefix and 'p' is the scaling factor. The steps followed for jaro wrinkler fuzzy multiple keyword search as follows,

Step1: The data files to be outsourced over the cloud server is encrypted using the hybrid MSAES encryption algorithm.

Step2: After encryption of data set, data owner generates the fuzzy keyword set using jaro wrinkler fuzzy methods. Fuzzy keyword set generates the index for encrypted files which are outsourced over cloud server.

Step3: User search with the keyword using the trapdoors of search request. Trapdoor forwards search request keyword to fuzzy keyword search techniques.

Step4: After receiving the request, the fuzzy keyword search technique generates the query for searching the keyword with the index. The appropriate data files from data sets are obtained.

7.2.2.3 Rank Retrieval

A ranking function is a key method to measure relevance scores of retrieval results based on keyword weight in different files according to the keyword search request. The fast ranking method is implemented in proposed RFMKS. Equation 7.3 shows the calculation of the score for ranking as,

$$\text{Score}(w, F_i) = \frac{1}{F_i} \cdot (1 + \ln f_{i,w}) \cdot \ln\left(1 + \frac{N}{N_w}\right) \quad (7.3)$$

The ranking algorithm calculates the frequency of each keyword in each file from file set E= (E_1, E_2, E_3... E_m). It also calculates the count of matching keywords from query with each file.

7.3 Flow diagram of the Proposed RFMKS method

The proposed method RFMKS for efficient user search experience with rank retrieval of data is discussed. The flow diagram of the proposed RFMKS algorithm is shown in figure.7.2.

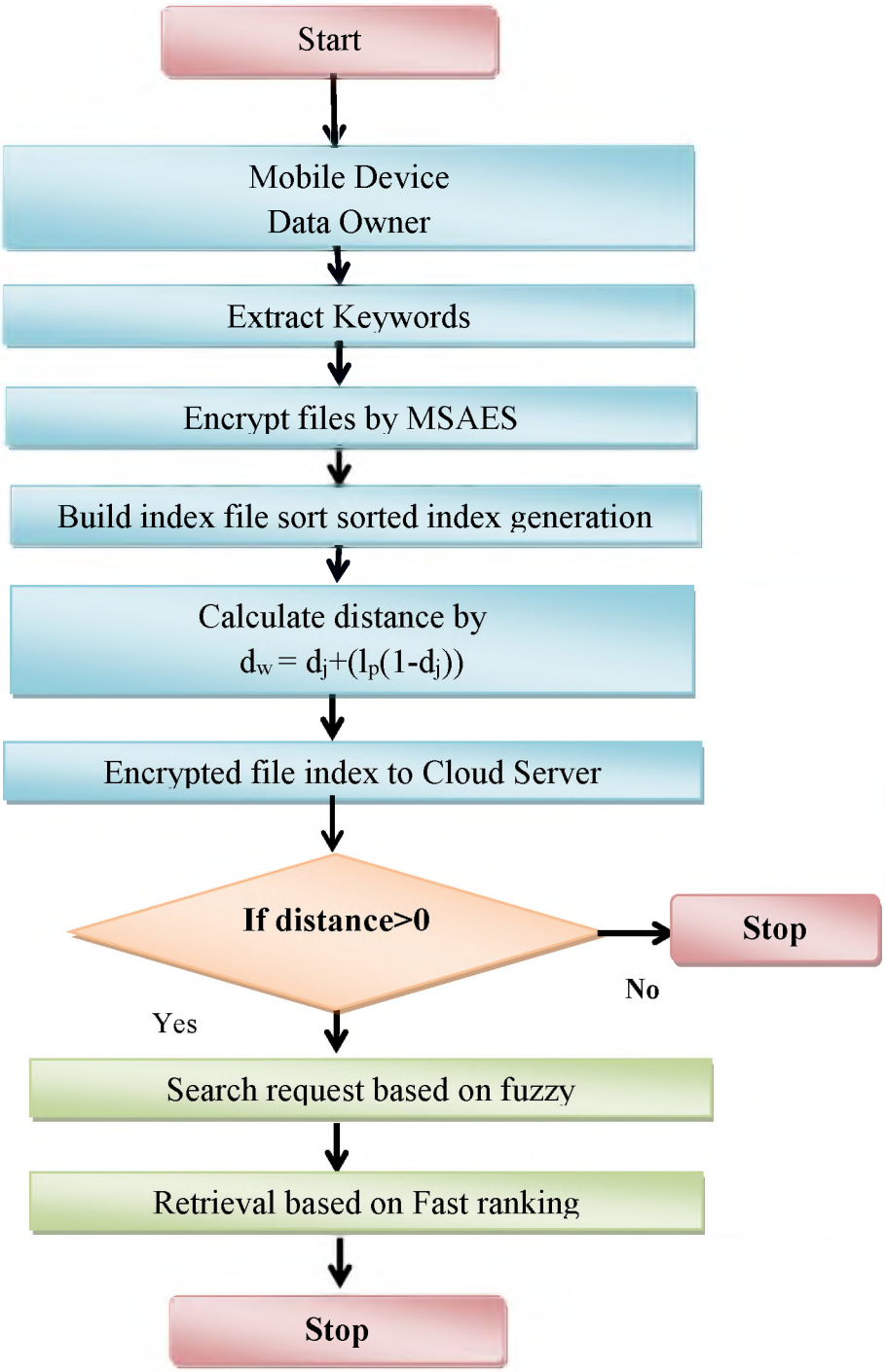

Figure 7.2 Flow Chart of Proposed RFMKS Method

7.4 Steps involved in the Proposed RFMKS Method

The steps used for the efficient retrieval of mobile device data over cloud storage is discussed below:

Step 1: *Building an Index file*

Step 2: *Initialization by selecting the keywords and building a keyword dictionary for all extracted keywords*

Step 3: *Select file to be encrypted using MSAES cryptographic algorithm and enter the key for encryption*

Step 4: *Enter the list of keywords that were selected from the file*

Step 5: *The data user sends the multi-keyword query* $w_q = (w_{q1}, w_{q2}, w_{qk})$ *to cloud server with the index file*

Step 6: *The jaro wrinkler distance function calculates the edit distance 'd' of each keyword in query 'wq' with all the 'n' keywords in index file*

Step 7: *If distance calculated returns zero, no changes takes place else 'wq' is replaced as* $w_d = (w_{d1}, w_{d2}, w_{dk})$.

Step 8: *The retrieval files are ranked based on fast ranking method which calculates the count 'R' of matching keywords from query '*w_d*' with each file.*

7.5 Pseudo code of the Proposed RFMKS Method

The algorithmic procedures applied for retrieval of encrypted mobile device data from the cloud server is shown in table 7.1

Table 7.1 Pseudo code of the Proposed RFMKS method

```
Initialize minimumTherosold,
        commonNum=0;
Calculate keyword length length1 and source length length2
if(length1 ==0) returns length2=0;
  Calculate Search range
     searchrange, max(0,max(length1, length2));
end if
Initialize the matched array elements
for (int i=0;i< length1;i++)
   {
        wordStart=Max(0,i- searchrange);
        wordEnd=min(i+ searchrange+1, length2;
         }
end for
if(keyword!=source) continue;
 else match=true;
    Calculate weight of word
Weight=(commonwords/lemgth1+commonwords/length2 + (commonword-
transporsedValue)/ commonword)/3.0;
      initialize the positionValue=0;
      calculate the position of words
end if
while (positionValue <maxLength&&keyword[pos] == source [pos])
         {
            Position++;
          }
       Calculate the proximity value
          Proximity=weight+0.1*position*(1.0-weigth);
          Disatance=1- Proximity;
End
Query received
     The DU sends the multi-keyword query
         wq = (wq 1, wq2,...wqk) to Cloud Server C.
Call to function
     The function calculates the edit distance d of each keyword in query wq with all the
n keywords in index file i-e., ed(wq,W)
if d returns zero
for any keyword then no replacement is made.
else
the keyword wqi is replaced with a keyword from index file where minimum edit
distance d is calculated.
end if
end for
    Searching is done on new replaced query wd = (wd1,wd2,….wdk).
Returns wd
```

7.6 Experimental Setup and Results

In the experimentation, an efficient data retrieval and enhanced user search experience using RFMKS algorithm is discussed. The input data files are search keywords related to mobile device data. For experimentation, the data files like text and image files are downloaded from the mobile device and encrypted are stored. The programs that are used for experimentation use two different file formats namely, text files and image files. A sample dataset is shown in Annexure IV. The experimentation methodology is shown in figure 7.3.

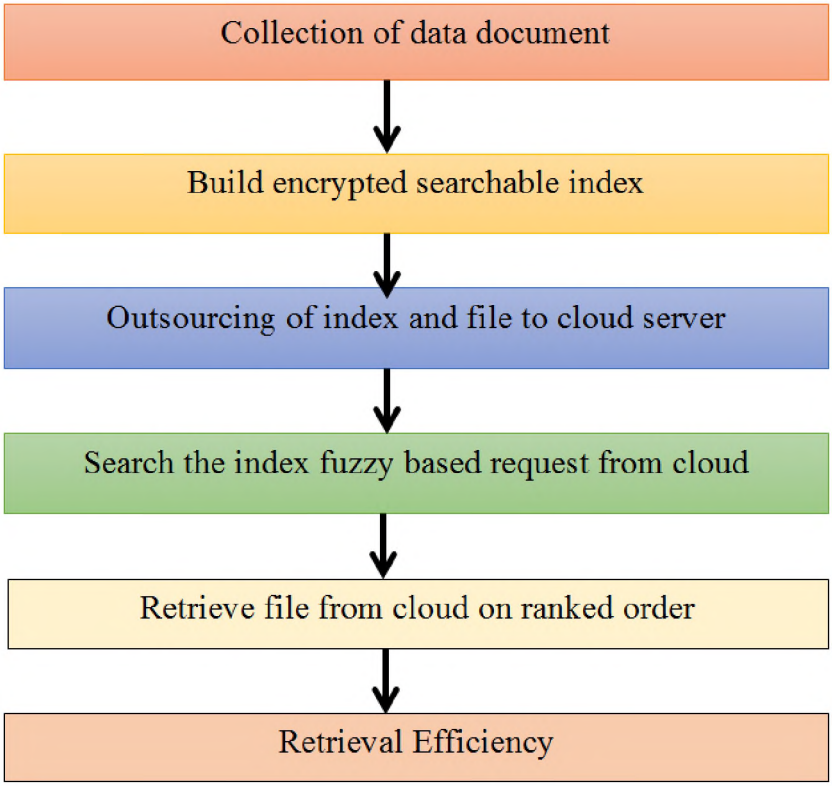

Figure 7.3 Experimentation Methodology for Contribution Four

To evaluate the performance of the proposed method, the following four parameters listed below are used and defined as,

i. User Search Time
ii. Index generation Time
iii. Encryption Time
iv. Decryption Time

i. User Search Time

Searching process includes processing the user query by applying fuzzy search. After correcting the query, each keyword is searched in the index file. The trapdoor set of each keyword is fetched and is matched with encrypted file set with the frequency of each keyword calculated. The time taken for searching based on request and response of the cloud server is defined as user search time as shown in equation 7.4,

User searching time = Request time – Server response time (7.4)

ii. Index generation time

It is defined as the difference between the optimized memory time and the data insertion time. It is shown in equation 7.5,

Index generation time = Optimized memory time – Insertion of data time (7.5)

iii. Encryption time

It is defined as the total time taken for encryption of file to upload. The equation 7.6 shows the time complexity of encryption,

Encryption Time = Time complexity (n^2) (7.6)

iv. *Decryption time*

It is defined as the time taken to perform decryption on encrypted data. The decryption time complexity is shown in equation 7.7,

Decryption time = Time complexity (n^3) (7.7)

Secure RFMKS search on cloud with the parameters obtained for the keyword searching are displayed in Table 7.2 and time taken for searching the keyword is displayed in Table 7.3. The keywords are listed when comparing with the existing algorithm, the RFMKS algorithm shows better efficiency in terms of index generation and searching time.

Table 7.2: Performance Comparison Results of Contribution Four - RFMKS Method

Keywords	Existing search time (s)	RFMKS search time (s)	Encryption time (s)	Decryption time (s)	Efficiency (+)
Hello	0.014566	0.006612	0.00285	0.003545	0.007954
chat Message	0.015645	0.00766	0.00568	0.005566	0.015479
Fuzzy search	0.07858	0.03222	0.04256	0.042865	0.077358
Facebook	0.020566	0.002578	0.02365	0.02334	0.017988

Table 7.3 Comparison Results of Search Time for RFMKS Method

Keyword Names	10 files, 10 keywords	25 files, 25 keywords	50 files,50 keywords	75 files, 75 keywords	100 files, 100 keywords
Camera	0.054566	0.0244663	0.124588	0.19345	0.190544
Contacts	0.066676	0.147992	0.182445	0.190334	0.285754
Location	0.062336	0.17439	0.195434	0.193554	0.023233
Calendar	0.0675645	0.142129	0.176343	0.012123	0.194554
Facebook	0.066523	0.156120	0.150334	0.113443	0.211833

The index generation time for the existing method and proposed RFMKS method is shown in figure 7.4.

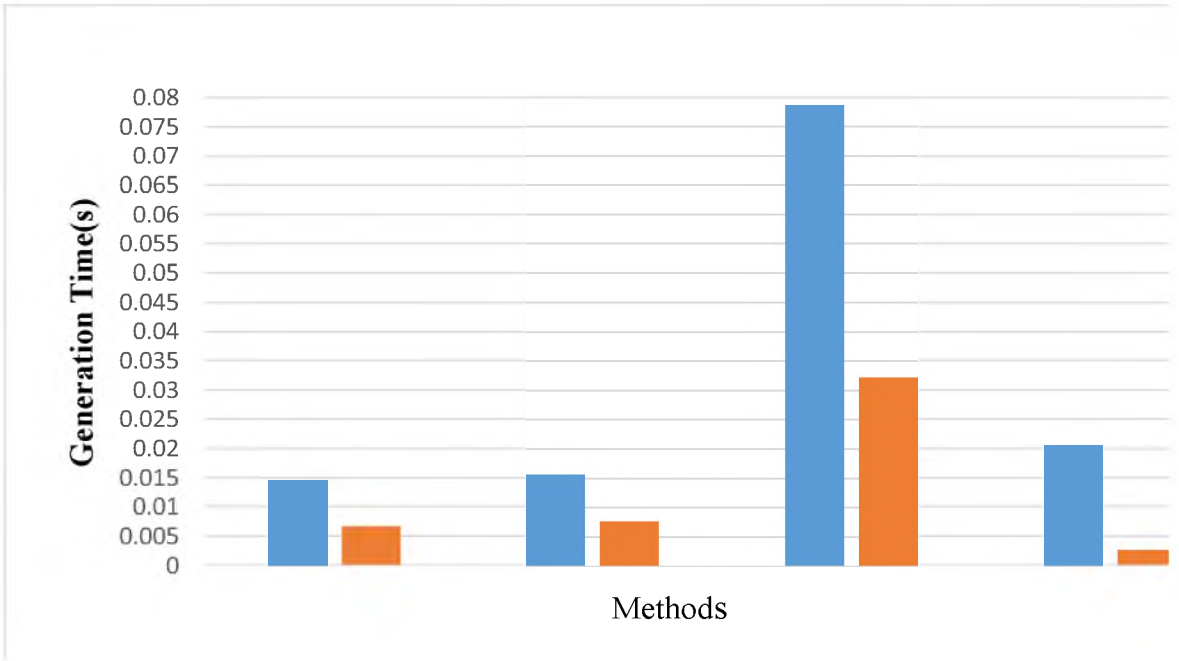

Figure 7.4 Comparison of Index generation time for Contribution Four

The encryption and decryption time for the keywords to search over cloud is shown in figure 7.5.

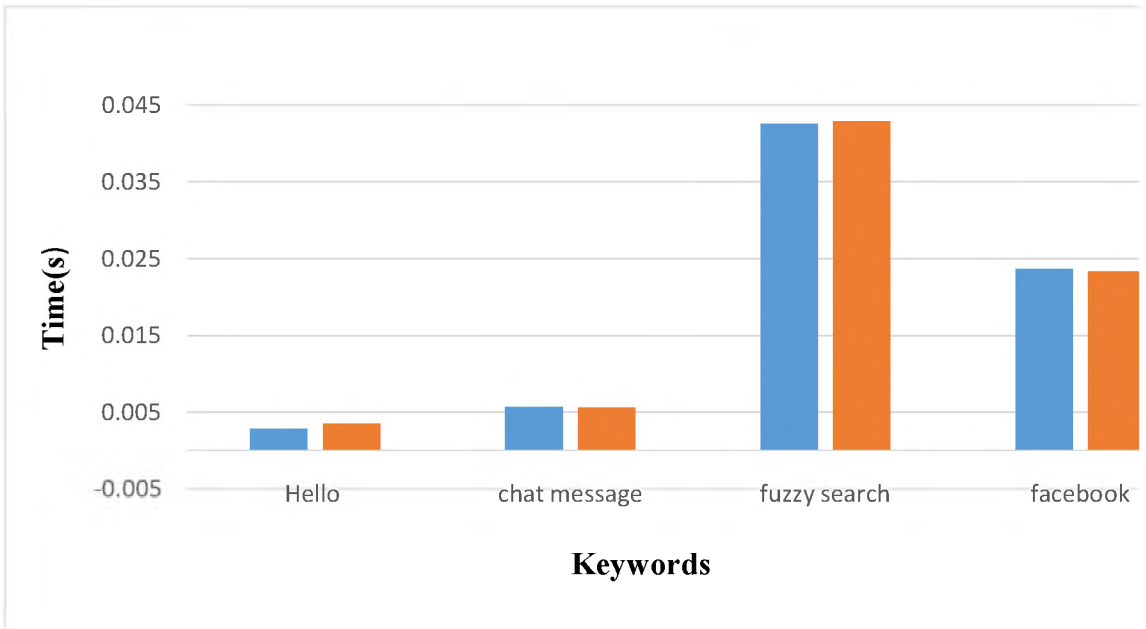

Figure 7.5 Comparison of Encryption and Decryption Time for Contribution Four

The search time for the keywords with varying file sizes such as 10, 25, 50, 75 and 100 shown in figure 7.6.

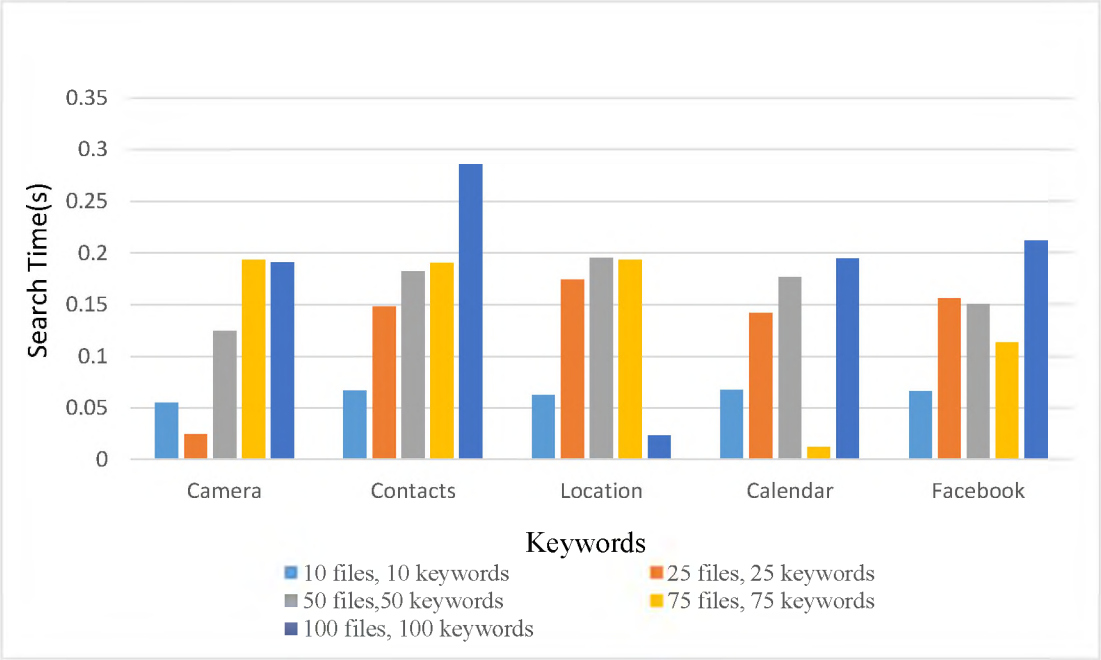

Figure 7.6 Comparison of search Keywords for Contribution Four

7.7 Chapter Summary

In this chapter, the proposed RFMKS method for privacy preserving fuzzy ranked multi keyword search on encrypted mobile device data in cloud storage is discussed. Ranking capability is incorporated to the search scheme which enables the users to retrieve all the possible relevant matches. An efficient ranking algorithm which will rank files on the basis of fuzzy multiple keyword search is discussed. The proposed RFMKS method is efficient and improves the user searching experience in cloud computing environment.