Improved Information Retrieval Technique for Cryptographic Data Storage

Shilpa Soni

Post-Graduation (M.tech)
Patel Group of Institution of Management

Abstract: **Cryptography is a technique to secure the data and their sensitivity and privacy from other users. That is a low cost and easy implementable technique for achieving these goals. Therefore different services and applications are consuming these techniques. In this presented work the cryptographic cloud technology is studied. The cryptographic cloud provides the security of data over the data during the data storage, retrieval and outsourcing. But that data is not searchable due to cryptographic data transformation. In this context the presented work provides the technique to secure the data using cryptographic and also make the data searchable using the keywords available in the data stored on cloud server. In order to develop such application first a hybrid cryptographic algorithm is implemented which usages the MD5 hash generation algorithm. That algorithm is used to generate the cryptographic key for encryption and the blowfish algorithm is implemented for computing the cryptographic data. On the other hand for making the data searchable first the text mining technique is used which extract the essential keywords form the data. Then for security purpose the keywords are stored in data base using the MD5 hash. During the search, data first indexed using the inverted index and then bloom filter is used for finding the most probable index. In further the data precisely searched using the KNN algorithm. That process improves the performance of search accuracy and resource consumption during the search. The observed results demonstrate the efficient and acceptable performance of implemented system.**

Keywords**: Data Security, Encryption, Decryption, Blowfish, Cryptography, Secure Search, MD5**

I. INTRODUCTION

Cryptographic technique is a popular security technique that is implemented in different applications for securing information from unauthorized access. These techniques are not only secured that are low cost and easy to implement. In cryptographic technique the data is transformed in an unreadable format for security purpose and only the person who has the correct recovery pin can access the data. In the similar ways the cloud storage providers also adopt the cryptographic technique for securing the user's data. But the cryptographic data is not suitable for searching or information retrieval purpose. Therefore in this presented work a secure cryptographic search technique is proposed for implementation and design that helps to search information more accurately.

The proposed work first demonstrates how the cryptographic storage is developed for securing the data on cloud. Therefore a web based application is developed that help users to upload or preserve data over remote server in cryptographic manner. In this context a hybrid cryptographic algorithm is also prepared. In addition of that for performing accurate search over the encrypted data a secure data model is prepared that help user to apply different keywords and search the files from the cryptographic storage. Based on the extracted results the performance of the proposed system is also computed and reported.

II. PROPOSED WORK

This section provides the detailed description of the proposed system and their functional aspects. Thus first the overview of the system is described and then the system architecture is provided. Finally the entire system is summarized using the algorithm steps.

A. System Overview

Cryptographic cloud ensures the cloud users the data is secured in the cloud environment. Additionally not any anonymous user can access the file. But the cryptographic technique transforms the actual data into unreadable format which is not accessible with the normal information retrieval processes. Therefore a search methodology is required that help to find the data which can be used for cryptographic data search. In literature a different kind of keyword based cryptographic search processes are available but either these techniques are not much secure or they are not much efficient. Therefore the proposed work is intended to design a new search technique for cryptographic information retrieval.

The proposed cryptographic search technique is working in two key scenario. In first scenario the cryptographic cloud is prepared using the hybrid cryptographic technique. That technique includes the implementation of the blowfish cryptographic algorithm and MD5 hash generation algorithm. This cryptographic technique helps to encrypt the user data to secure over the cloud from unauthorized users. At the same time from the user data essential keywords are also recovered. These keywords are preserved separately over the database for performing search. But to secure the original keywords from the network attackers these keywords are preserved in data base using hash codes. To prepare the keyword hashes the MD5 algorithm is again used during storage of keywords. In the second part of the system implementation the search methodology is implemented. In this context the first the data is arranged in inverted index. This index is passed over the bloom filter with the user query hash codes. The bloom filter helps to identify the set of files where the probability of keyword available. That helps to reduce the search time of the cryptographic search

algorithm. Finally the identified set of file's keyword and the user query keywords are processed with the KNN (k-nearest neighbor) algorithm for providing the accurate list of files where the keywords are available. This section provides the basic details of the proposed methodology. The next section provides the detailed description of the proposed system.

B. System Architecture

The proposed system is described using figure 2.1 and figure 2.2. In the figure 2.1 the uploading process of the files from the client end is described. In addition of that the process taken place during the upload event of file is also demonstrated using figure 2.1. In the similar ways the search methodology is described in the figure 2.2. The different involved components of the model are defined as:

a. File Upload

The components of the file upload process are described in this section:

Input file: user wishes to upload a file on the cloud server. Therefore user selects a file from the local machine and using the system provision the data is uploaded to the server.

Pre-process: as the file selected through the system a copy of file provided to this phase. In this phase the data is tried to reduce for finding the essential keywords from file. In this context first the unwanted symbols (such as , . ;) are removed from the file. That process reduces the content in small amount. In next process the stop words are reduced from the data. To reduce the stop words from the input file an additional stop word list is provided to system and the system replace all the stop words from the data with the blank space. That process reduces the data much frequently and only important data and words are remaining in the input file.

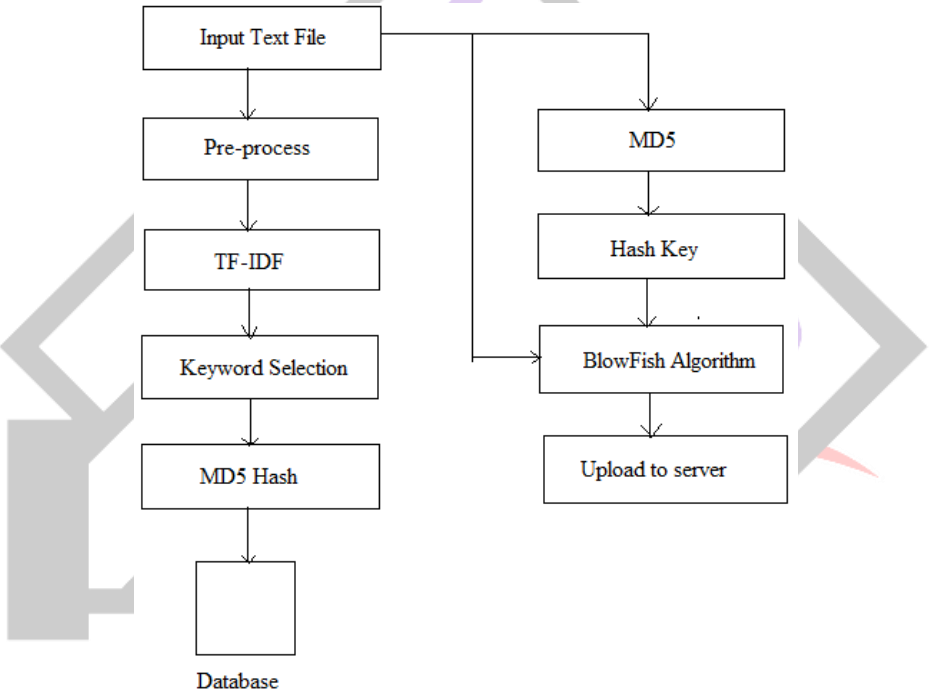

Figure 2.1 File Upload Scenario

TF-IDF: after removing the stop words and special characters from the file, individual words importance is computed. Therefore the word frequency of each word is calculated. To compute the word frequency the following formula is used.

$$word\ frequency = \frac{number\ of\ times\ word\ appeared\ in\ content}{total\ number\ of\ words\ in\ document}$$

After computing the word frequency of each word the words and frequency is provided in next phase for process.

Keyword Selection: in this phase the essential words are selected from all the remaining words available in previous data. Therefore a fixed length of data is needed to be prepared. In this experiment the 20 words length is considered for keyword selection. Therefore the 20 words which having the higher word frequency is selected and provided to the next phase.

MD5 Hash: the MD5 algorithm is a hash generation algorithm which accepts a different length of string and generates a fixed length 128 bit of fixed string. The extracted keywords are processed using the MD5 algorithm and for all the keywords the hash codes are generated. These hash code is preserved in the database for feather use during the search.

Database: the database contains the extracted keywords from file in the form of hashes. In addition of that the file name which is stored on the server.

Hash key: on the other end the original file is again used with the MD5 algorithm which generates the 128 bit of hash key. This 128 bit key is produced in next phase for encryption of file.

Blowfish algorithm: the blowfish is one of the popular cryptographic algorithms which are used here for encrypting the input file and to store data on server. In this context the MD5 generated hash code of the file and the original file is provided to the algorithm that algorithm encrypts the file and produce it to the next phase.

Upload to server: finally the encrypted file is uploaded to the cloud server for use.

b. Search Process

In this phase the process of search is described and their components which are used for preparing the search methodology is discussed.

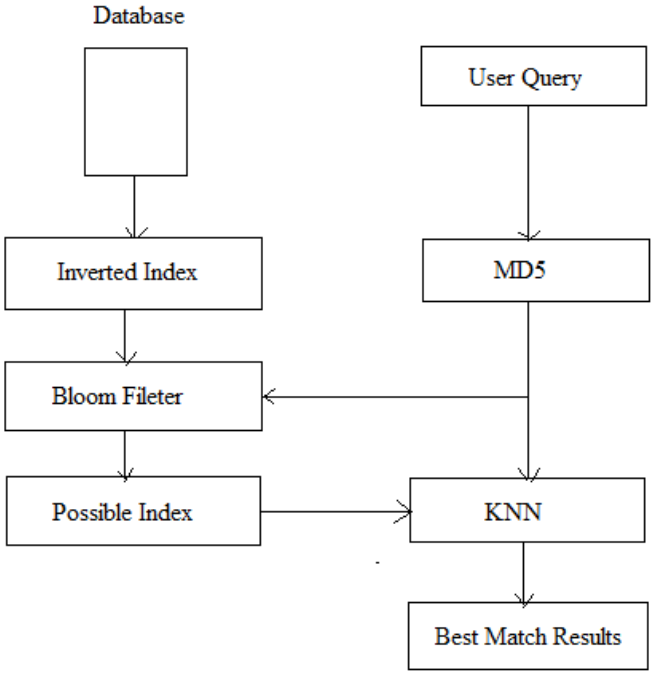

Figure 2.2 Search Process

Database: the prepared database in last phase is used here for performing search. Here the term database is used for the database table which contains the extracted keywords hash codes and the storage which is used to upload the file by users.

Inverted index: the files and the keywords stored in data base are first transformed in an efficient data structure. That data structure helps to find the files according to the index. An example of inverted index is defined using the table 2.1. This table contains the keywords in first column and the second column stores the file names which are hosted in server before.

Keyword	Files
K1	F1, f2, f5, f7
K2	F1, f3, f6

Table 2.1 inverted index

User query: that is the second input by the user. Using this input user tries to search the data from server. The user query is a string of keywords passed on the user input provision in the system for finding the relevant data.

MD5: the user input keywords are first processed using the MD5 algorithm for generating the hash code for each keyword in the user query.

Bloom filter: in this phase the inverted index table and the user query hashes are produced for finding the indexes where the data is stored.

Possible index: the bloom filter returns the possible indexes from the table where the keywords are possible.

KNN search: to return the precise outcome the KNN search on the selected index data and the user query data is performed. That result accurate list of files, which contains the user given keywords.

Best match results: that is the provision which contains the list of files which contains the user query outcomes.

C. Proposed Algorithm

This section provides the algorithms steps to explain the data upload technique and the user search outcomes using the previously defined methodology.

Table 2.2 File Upload

Input: user file F Output: keywords list K_{list}, file upload to server F_s
Process: 1. $R = readFile(F)$ 2. $D_n = preprocessData(R)$ 3. $for(i = 1; i \leq n; i++)$ a. $Fr = ComputeFrequency(D_i)$ 4. $end\ for$ 5. $K_{list} = SelectTopFrequentWords(Fr)$ 6. $key = MD5.genrateHash(F)$ 7. $F_s = BlowFish.Encrypt(F, key)$ 8. Return F_s

Table 2.3 File Search

Input: user query Q, Keyword database D, File storage S Output: list of files F_{list}
Process: 1. $K_n = ExtractKeywords(Q)$ 2. $H_n = MD5.genrateHash(K_n)$ 3. $Inv = CreateInvertedIndex(D, S)$ 4. $FP_m = bloomFilter.Find(H_n, Inv)$ 5. $for(j = 1; j \leq m; j++)$ a. $temp = KNN.search(H_j, FP_j)$ b. $F_{list}.Add(Temp)$ 6. End for 7. Return F_{list}

III. RESULT ANALYSIS

The implementation of the Secure Cryptography Data Search *(SCDS)* technique is described in previous chapter. This chapter provides the detailed understanding about the experimental evaluation and performance computation. Therefore essential parameters which are used for evaluation are listed with their observations.

A. Precision

Precision measure is the ratio of the number of correct positive results and number of all positive results. It measures the exactness of any search query. The higher the precision means that less false positives (FP), whereas the lower precision means that more the false positives are. Here, we are showing precision rate formula:

$$\text{Precision Rate} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

-TP is the number of true positives

-FN is the number of false positives

The evaluation of secure cryptographic data search i.e. SCDS model is demonstrated using figure 3.1 and table 3.1. In this figure, X-axis show the number of different experiments performed and Y-axis depict the precision rate of the algorithms. Blue line show proposed approach using Blowfish Cipher. In this, we have demonstrated and analysis graph value on the basis of tabular form. By this result, cloud storage security is increases during performing the execution in different time. Additionally, performance is most efficient and accurate in terms of the precision rate other than other tradition algorithm.

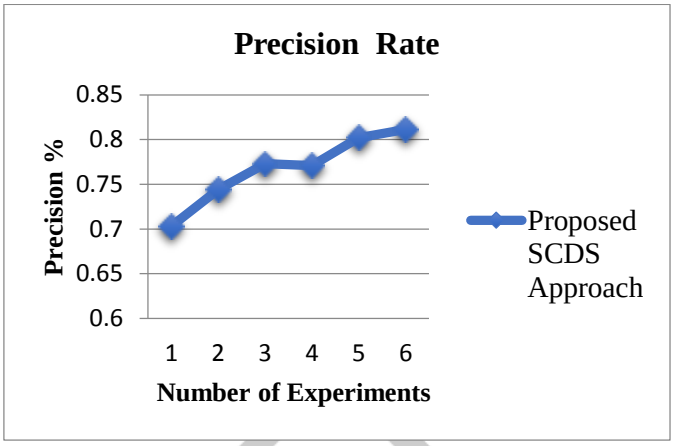

Figure 3.1 Precision Rate

Table 3.1 Numeric Values for Precision Rate

Number of Experiments	Proposed SCDS Technique
1	0.7032
2	0.7441
3	0.7730
4	0.7708
5	0.8021
6	0.8113

B. Recall

Recall is the ratio of the number of correct positive results and number of positive results that should have been returned. It measures the completeness or accuracy of the searching of the query. Higher the recall means that small false negatives (FN), whereas lower the recall is more false negatives it leads to. In this, following recall rate formula used to calculate Performance of algorithm.

$$\text{Recall Rate} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

-TP is the number of true positives

-FN is the number of false negatives

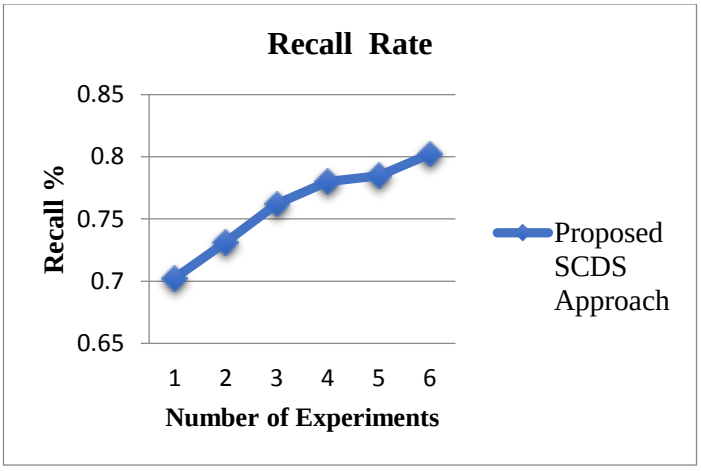

Figure 3.2 Recall Rate

The figure 3.2 and table 3.2 shows the recall rate of implemented SCDS approach. In order to show the performance of the system the X-axis contains the different execution of the project and the Y axis shows the performance in terms of recall rate percentage. The performance of the proposed data search algorithm is given using the blue line along with their tabular form. The performance of the proposed model is effective and efficient during different execution and reducing with the amount of data increases. Thus the presented model is more efficient and accurate of data security with search keyword according to their indexing.

Table 3.2 Numeric Values for Recall Rate

Number of Experiments	Proposed SCDS Technique
1	0.7023
2	0.7314
3	0.7622
4	0.7801
5	0.7847
6	0.8022

C. F-Measure

F-Measure or F1 Score is the harmonic mean of the Precision and Recall often used as a weighted average for balancing quality vs. quantity of true positives selection of an algorithm given by:

$$\mathrm{F-Measure} = 2 * \frac{\text{Precision x Recall}}{\text{Precision} + \text{Recall}}$$

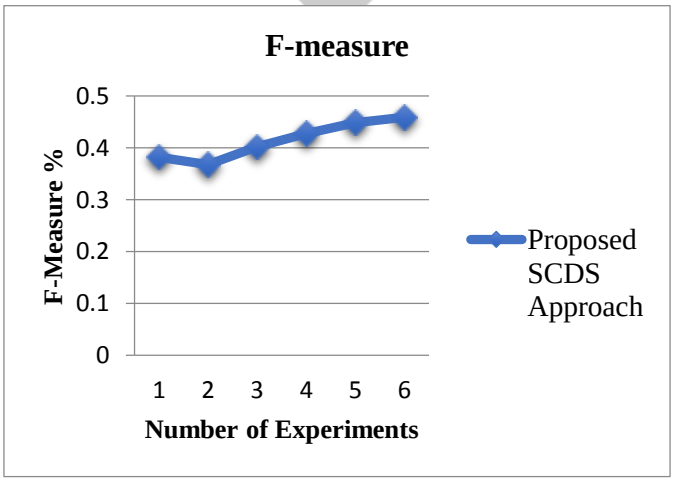

Figure 3.3 F-Measure

Figure 3.3 and table 3.3 shows the performance of Proposed Security Model in terms of f-measures parameter. To demonstrate the performance of the system the X axis shows the runs for data execution and the Y axis shows the obtained performance in terms of f-measures. Additionally, blue line represents the SCDS performance. According to the obtained results the performance of the proposed system is much stable and enhancing security of end user data using blowfish algorithm. In addition of that the results are in more progressive manner as the amount of data base is increases. Thus the obtained results are adaptable and efficient for the user data security.

Table 3.3 Tabular Values of F-Measure

Number of Experiments	Proposed SCDS Technique
1	0.7026
2	0.7376
3	0.7674
4	0.7754
5	0.7932
6	0.8066

D. Time Consumption

The amount of time required to process the execution of algorithm is known as the time consumption. That can be computed using the following formula:

$$\text{Time Consumed} = \text{End Time} - \text{Start Time}$$

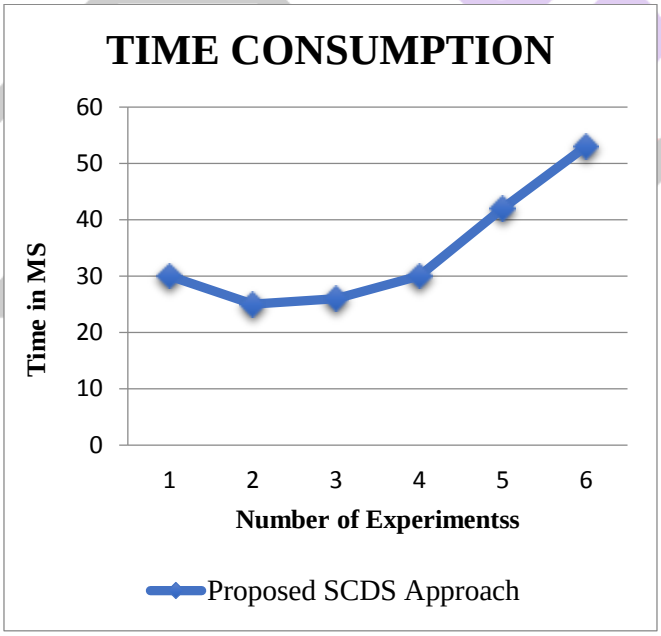

Figure 3.4 Time Consumption

The time consumption of the proposed algorithm is given using figure 3.4 and table 3.4. In this diagram the X axis contains different execution of experiments and the Y axis contains time required to process the algorithm. According to the given results analysis the performance of the proposed SCDS approach shows the small amount of time requirement. But the amount of time is increases in similar manner as the amount of data for analysis is increases.

Table 3.4 Tabular Values of Time Consumption

Number of Experiments	Proposed SCDS Technique
1	30
2	25
3	26
4	30
5	42
6	53

E. Memory Consumption

Memory consumption of the system also termed as the space complexity in terms of algorithm performance. This can be estimated using the following formula:

$$\text{Memory Consumption} = \text{Total Memory} - \text{Free Memory}$$

The amount of memory consumption depends on the amount of data reside in the main memory, therefore that affect the computational cost of an algorithm execution. The performance of the implemented cryptographic data search algorithm is given using figure 3.5 and table 3.5. For reporting the performance the X- axis contains the number of runs by executing algorithms and the Y axis shows the respective memory consumption during execution in terms of kilobytes (KB). According to the obtained results the performance of algorithm of system memory is increasing whenever we increase amount of input data. Therefore, in our SCDS approach less amount of memory consumed even input data is increasing.

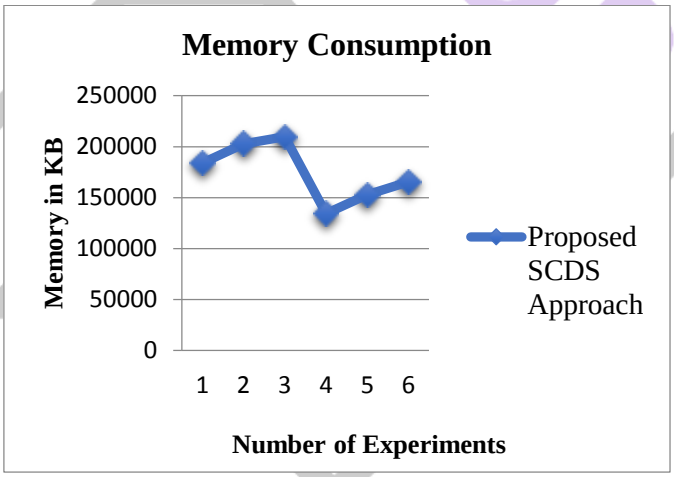

Figure 3.5 Memory Consumption

Table 3.5 Numeric Values for Memory Consumption

Number of Experiments	Proposed SCDS Technique
1	184149
2	202978
3	209636
4	134282
5	152803
6	165407

IV. CONCLUSION

This chapter provides the summary of the overall work performed for designing, implementation and analyzing the proposed technique of secure data search technique. The conclusion made here on the basis of experiments performed and the future extension of the work is also included.

A. Conclusion

The cloud is a huge infrastructure which provides the computational platforms and efficient storage for hosting the data. But the cloud storage providers are outsource their data to other cloud storage servers for reducing the data maintenance cost and the management effort. Due to this the data owner and service providers both are worried for privacy and security of data. In order to improve user trust the cloud service providers offer the cryptographic cloud for secure data storage. But due to the cryptographic process the data is not searchable for normal process of text keyword search. In this context the proposed technique offer a secure cryptographic data search process. That works on the basis of the data keywords which are associated with the text files which are hosted on the cloud servers.

The proposed technique works on two main phases first the preparation of cryptographic data storage and keywords key word. Secondly implementation of the secure searches technique to identify the files which contains the data which is required to search. Therefore first the keywords are extracted from target input files. These extracted keywords are preserved separately on the database with the appropriate file mapping. In further to secure the files on server a cryptographic technique is implemented. The cryptographic technique includes the blowfish encryption with the MD5 hash code. To arrange the data and keywords for efficient search the inverted index concept is used. In the next phase the search system is implemented with the help of bloom filter. The bloom filter is basically works on the hash data thus the search query is processed using MD5 hash code and database which is available in hash code is produced for extracting the file names in server storage as the output. During the search of the data precision, recall and f-measure is calculated for demonstrating system performance. Similarly the time and space complexity is provided as efficiency parameter.

The proposed system is implemented with the help of JAVA technology. After the implementation of the proposed technique the performance of the search system is computed and the observations are concluded in table 4.1.

S. No.	Parameters	Remark
1	Precision	High precise results which increasing amount of data for performing search
2	Recall	Acceptable recall rate for different cryptographic file and search keywords
3	F-measures	Acceptable accuracy due to less fluctuation on results with the different key word
4	Time complexity	The less time is consumed for inverted index based search process
5	Space complexity	The memory usages of the current system is low due to implementation of bloom filter for search and indexing

Table 4.1 performance summary

According to the evaluated performance and described results in table 4.1 the proposed model is acceptable for utilizing with the real world applications.

B. Future Work

According to the obtained performance results the proposed cryptographic search technique efficient and able to produce precise results for user query. In near future the following improvements are proposed.

1. Current system is implemented only for the plain text files in near future more file formats are include for securing and retrieval

2. In current system the normal bloom filter is used which only distinguish the file is available on the set or not in near future more complex bloom filter is tried to implement

3. In near future need to explore more literature for improving the search time and enhancing the accuracy of the proposed system.

REFERENCES

[1] Zhang, Hongli, Zhigang Zhou, Xiaojiang Du, Panpan Li, and Xiangzhan Yu, "Practical and privacy-assured data indexes for outsourced cloud data", In Global Communications Conference (GLOBECOM), 2013 IEEE, pp. 671-676.
[2] I. Foster, Z. Yong, I. Raicu, and S. Lu, "Cloud Computing and Grid Computing 360-Degree Compared," in Grid Computing Environments Workshop, 2008. GCE '08, 2008, pp. 1-10
[3] M. Armbrust et al., "Above the Clouds: A Berkeley View of Cloud Computing," Univ. California, Berkeley, Tech. Rep. UCBEECS-2009-28, Feb. 2009.
[4] Sookhak, Mehdi, et al. "Remote data auditing in cloud computing environments: a survey, taxonomy, and open issues." ACM Computing Surveys (CSUR) 47.4 (2015): 65.
[5] J. F. Yang and Z. B. Chen, "Cloud Computing Research and Security Issues," 2010 IEEE International Conference on Computational Intelligence and Software Engineering (CiSE), Wuhan pp. 1-3.
[6] "Introduction to Cloud Computing", Dialiogic, available online at: https://www.dialogic.com/~/media/products/docs/whitepapers/12023-cloud-computing-wp.pdf
[7] Mather, Tim, Subra Kumaraswamy, and Shahed Latif, Cloud security and privacy: an enterprise perspective on risks and compliance "O'Reilly Media, Inc.", 2009.
[8] Goulding, J. Tony, Identity and Access Management for the Cloud: CA Technologies strategy and vision. Tech. Rep. May, CA Technologies, 2010.
[9] Pearson, Siani, "Taking account of privacy when designing cloud computing services", Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing IEEE Computer Society, 2009.
[10] Neha Rawat and Ratnesh Srivastava, "Data Security Issues in Cloud Computing", Open Journal of Mobile Computing and Cloud Computing, Volume 1, Number 1, August 2014
[11] Venkata Sravan Kumar, Maddineni Shivashanker Ragi, Security Techniques for Protecting Data in Cloud Computing, Master Thesis, Electrical Engineering November 2011
[12] John Harauz and Lori M. Kaufman, "Data Security in the World of Cloud Computing"
[13] Obrutsky, S. "Cloud Storage: Advantages, Disadvantages and Enterprise Solutions for Business", (2016).
[14] Wu, Jiyi, et al. "Cloud storage as the infrastructure of cloud computing", 2010 International Conference on Intelligent Computing and Cognitive Informatics (ICICCI), IEEE, 2010.