I.J. Modern Education and Computer Science, **2016, 1, 40-47**
Published Online January 2016 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijmecs.2016.01.06

Enhanced Ranking Based Cloud Searching with Improved Metadata Storage: A Case Study for Relevancy of Files

Rajpreet kaur
Research Scholar, CGC, Landran (Mohali), Punjab, India
E-mail: Preet.billing00@gmail.com

Manish Mahajan
Head of Department, Computer Science & Engineering, CGC, Landran (Mohali), Punjab, India
E-mail: Cec.manish@gmail.com

Abstract—With the outgrowth of cloud computing, a large amount of private information is stored over cloud servers, which is in encrypted format. But searching over encrypted data is very difficult. Earlier search schemes were based on Boolean search through keywords. But don't consider relevance of files. After that ranked search comes into its role, which uses searchable symmetric encryption (SSE). To achieve more practical and efficient design method was further modified to "Order preserving symmetric encryption" (OPSE), which uses primitives and indexed metadata files used in ranked SSE. In this proposed work further enhancements are done to reduce storage space for encrypted metadata using Porter Stemming method. Improvements in retrieval time are also done by using Boyer Moore's searching algorithm.

Index Terms—Private information, Ranked search, Metadata, Porter Stemming, Boyer Moore's algorithm.

I. INTRODUCTION

In simple words, cloud computing can be defined as a computing model which provides users access to a large shared pool of resources. From where, user can get hardware and software resources by paying according to their needs. It releases the user from the burden of hardware installation and maintenance. Only thing that is needed is internet connection. Different cloud service providers are available (like Google, Amazon, Rackspace Cloud, and Windows Azure etc) which provides services like networking, storage, applications etc. Cloud computing is a software through which you can take hardware and software resources on rent. Basically, cloud computing is based on pay-as-you-use model.

A. Searching cloud data

As cloud computing is a dominant platform in information technology, more and more information is being uploaded over cloud servers. These information files contain private records (for example personal photographs) of individuals and confidential information (e.g. Military records, bank account of person) which must be secure in encrypted form while stored over cloud servers. But searching these encrypted files is very difficult task. Earlier schemes were merely based on keyword search and return results only according to presence of query keyword. This increases network traffic by returning all files that contain search keyword. Also, it is not necessary that all the returned results are relevant to user's requirements. Hence technique is not quite effective.

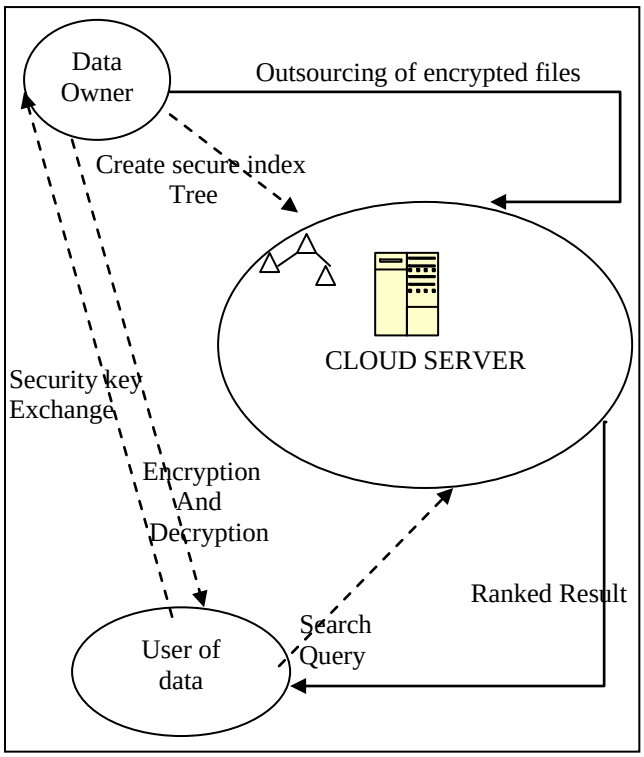

Fig.1. Architecture for Secure Ranked Search Scheme

By using ranked keyword search, results are returned in a ranked order. Results are according to similarity with search query.

A simple model for ranked searching over encrypted

cloud data is shown above in figure 1.

In cloud based search scheme for encrypted data there are basically three types of roles. First is cloud owner, which creates his own information files and outsource the files onto cloud in encrypted format. Along with files, data owner also creates secure inverted index for each unique keyword in the file. Using this index scheme an index tree is generated, which helps in creating metadata of files. This metadata helps in defining the relationship of various keywords with each other. Metadata should also be in encrypted form.

Second role is named as cloud server. All the encrypted files and index tree is stored at cloud server. Most of the computation of generating ranks and calculating relevance of files is done at cloud server. When user generates a search trapdoor with a given keyword, server checks the presence of keyword in files and then checks similarity of search keyword with files using index and metadata stored. After calculating relevance of files server generates the list of results in encrypted format.

In the end, role of user comes into play. User generates a search trapdoor by sending a query keyword to server. Results are returned in ranked order. But these results are still in encrypted form. User decrypts these results using his public key.

B. Order Preserving Symmetric Encryption

OPSE is basically an encryption scheme which preserves numerical order of plaint text of data while it is in encrypted form. It uses cryptographic basics and permutation functions.

II. Related Work

Ren et al. [18] proposed similar secure per-file index, where an index including trapdoors of all unique words was constructed for each file. In index based scenarios, unique index is created for each file and each document. Here, authors have paid attention to many privacy related issues.

Xia et al. [12] described that secure semantic expansion based search over encrypted cloud data. This work is concerned with searchable encryption techniques for secure outsourced data. Author also considers order preserving symmetric encryption (OPSE) for preserving order of data in encrypted form. In this research first of all searchable symmetric encryption is described. After that implementation for OPSE is given. Result analysis elaborates security of scheme and benefits of preserving orders of results. Boldyreva [2] first gave a cryptographic primitive of OPSE and implemented it for secure search framework. One advantage of this approach over others is that along with similarity based search, results are produced as ranked list. So finding appropriate results became more efficient.

Some practical techniques of searching encrypted data are explained by Song et al. [8]. This work elaborated that for security proposes, functionality is often sacrificed. So, there is a tradeoff between security and efficiency. As the use of encryption methods and index structures increased the storage and computational overloads.

Cao et al. and Yang et al. [4, 13] proposed schemes for multi- keyword ranked search. Boneh et al. [4] proposed first public key based searchable encryption (PEKS). Li et al. [7] exploited edit distance as similarity metric of keywords to construct fuzzy set as indexes. In a fuzzy set not only exactly matched results are considered, but also the values which are nearly similar to query keyword.

Including these some literature related to Information Retrieval (IR) is also surveyed. Basic introduction to IR technologies is given in [21]. This work explains some basic search techniques for information retrieval and storage. A thorough study of improved Porter Stemming algorithm is given by Ramasubramanian et al. in [22]. Porter stemmer is generally an algorithm for efficient storage of text.

III. Proposed Model of Work

A. Problems in basic system

Above mentioned ranked SSE technique provides an efficient way for retrieval of secure cloud data. But SSE creates its own database while generating index structures and metadata. Hence, a large storage need arises for metadata creation. This storage problem can be reduced using information retrieval techniques like "Porter Stemming method" and "Stop words removal".

One more issue is that in SSE search scheme is slow because it matches every single alphabet and character for matching terms. Here, search time can be reduced by using "Boyer Moore's algorithm" which search complete string every time.

B. Basic design of system

In implementation of this work two types of algorithm are implemented. First algorithm is for Metadata creation and second is for fast searching. After these two steps ranks are generated for results by giving score to matched files.

1. Metadata creation

In this step user uploads data in encrypted format. Along with this, metadata is also created. Storage space for metadata is reduced by some strategies like Porter stemming, stop words removal, removing repeated words.

In first step of metadata creation storage procedures of information retrieval are considered. These IR procedures are discussed below:

- **Removing Stop words**
  - Function words don't bear useful information for IR like of, in, the, about, with, I etc.
  - Stoplist- Prepositions, articles, pronouns, some adverbs come into this category.
  - Removing of these stop words usually improves efficiency of IR.

- **Porter stemming**

"Stemming" is a word used in IR to describe the process of reducing inflected (derived) words to their root form or stem. For example, all the words computer, computing, computed, computational etc can be stemmed to "comput".

The diagram given below in figure 2 explains the steps for metadata creation. Metadata creation starts with data owner. Data owner upload his file to the cloud server in encrypted form. Along with encrypted data files index trees which have unique ids and mappings for each word in the file are also uploaded. At cloud server metadata is created from index trees. Metadata is also in the encrypted form. Metadata storage can be reduced by removing repeated terms and using porter stemming method.

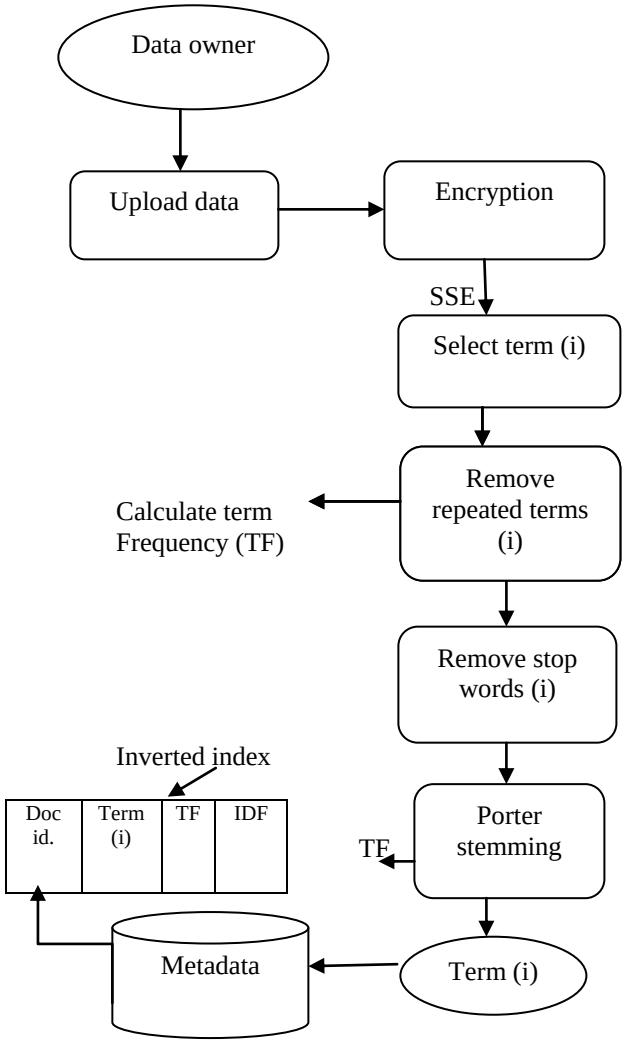

Fig.2. Flowchart for Metadata Creation

- **Metadata creation algorithm**

As shown in above figure metadata is created for uploaded files. Two main Parameters Term frequency (TF) and Inverse domain frequency (IDF) are generated which are used later for score generation in further implementation. Next to this, complete algorithm for metadata creation is shown in table.

Table 1. Algorithm for Metadata Creation

```
BEGIN
Term(i) ← null;
TF ← 0;
{
[Data DocID] ←uploaded Doc by user
EncData ← AES(Data); //Data Encryption
For i = 1 to n //where n is no. of terms in the document

Term(i) = EncData
Count= count_Term(i)
TF=count  // Term Frequency
// Remove repeated terms
If TF >.= 1
        Remove term(i) upto TF==1
End
//Remove stop words
If Term(i)==stopwords
        Remove Term(i)
End
// Porter's Stemmer
define Step_1a as (
    [Term (i) ] among (
      'sses' (<-'ss')
      'ies'  (<-'i')
      'ss'   ()
      's'    (delete)
    )
  )
define Step_1b as (
    [Term (i) ] among (
      'eed'  (R1 <-'ee')
      'ed'
      'ing'  (delete)
    )
  )

Metadata() ← Term(i)
End of for loop
}

END
```

2. Searching encrypted data

After metadata creation searching is next step. When user wants to search some text, a query is generated by giving search keyword. With the help of metadata a relationship of words similar to query keyword is identified. In searching Boyer Moore's algorithm is implemented. Boyer Moore is a string search algorithm, in which complete search keyword is considered as pattern and compared with words (strings) in each document or stored file. All the files are isolated and no information of plain text is revealed. Hence, approach is

secure as well as efficient. Work flowchart is given below.

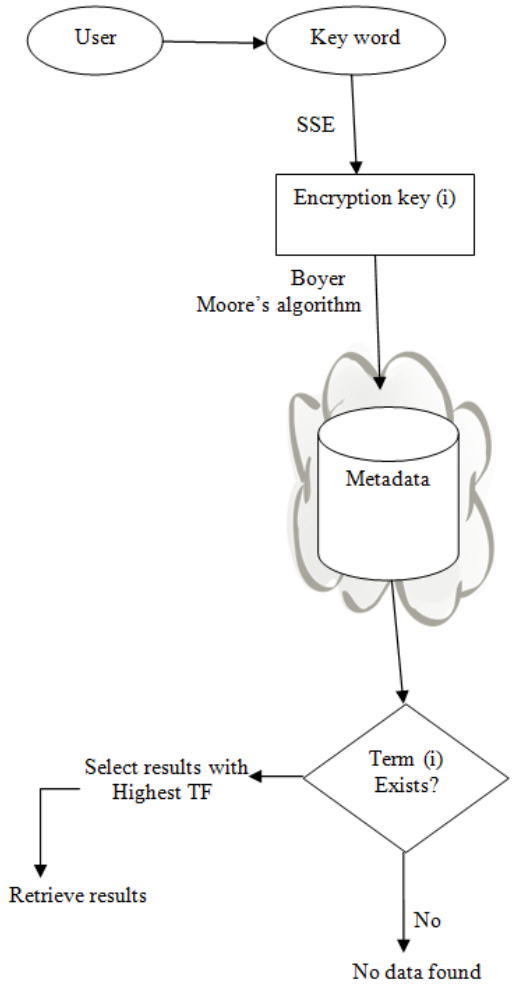

Fig.3. Flow Chart for Search Procedure

- **Boyer Moore's Algorithm**

This string matching algorithm starts matching at the end of pattern string P rather than the beginning. When a mismatch is found, this allows the shift to be increased by more than one. Algorithm uses three basic ideas which are right to left scan, the bad character rule, good suffix shift rule.

3. Score Generation

After searching, results are scored for ranking them. Weight (score) generation can be calculated using following factors.

Term frequency (TF): It is the frequency of a term/keyword in a document. The higher the TF, higher is the score for document.

Document frequency (DF): It is the number of documents containing that term.

Inverse domain frequency (IDF): Its value can be obtained by dividing the number of files containing the term. Commonly used weighting formula is,

$$Score(t, D) = TF(t, D) * IDF(t) \quad (1)$$

Here t is term/keyword and D is Document.

Table 2. Searching Algorithm

```
BEGIN
Keyword ← enter by user
EncKey ← SSE(Keyword)
For i= 1 to s //where s is the size of metadata
Searched Key = BOYER MOORE's (Metadata)

//Boyer Moore's
Given Data pattern P ← EncKey
for each position i to P
Compute M'(i) and m'(i)
End
For each character x
Compute R(x)
End
//Search Stage
K:= n;
While k <= size(Metadata) do
        I=n;
        N=K;
        While i>0 && P(i) =T(h) do
                I=i+1
                H=h+1
        End of while
If i == 0
Report an occurrence of P in T ending at position k
K=k+n-m'(2)
End of if
End of while
END
```

IV. Result Analysis

In this section, results of implementation are evaluated for efficient retrieval and improved storage space for encrypted cloud data. For implementation of above given work firstly web based environment is generated using C# in visual studio and SQL server 2008. After that this web based work is converted into local cloud using Microsoft Windows Azure (Platform as a service (PAAS) cloud).Result analysis is done on the basis of following factors:

- Metadata Size
- Searching Time
- Precision
- Recall
- F- measure

A. Metadata Size

For storing metadata of files in encrypted form a large storage is required. This storage size can be improved to greater extent using IR strategies like stop words and

Porter Stemmer. Further a table is given in which metadata size of 15 files is given for both base SSE scheme and enhanced method. Graph clearly shows the change in Metadata size. The size of metadata can be calculated from database storage by giving SQL query, Sp_spaceused tablename.

The SQL query given above generates size of metadata, which is shown in table 3 given below.

Table 3. Metadata Size

◦	base	advance
For 15 Files	72	48

The values shown in table give storage space of metadata for both basic method and enhanced method. A graphical representation given in figure Fig.4., clarifies the difference.

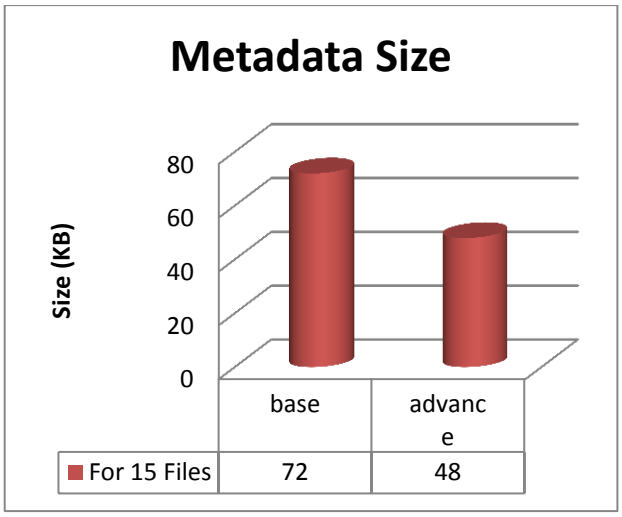

Fig.4. Graph for Metadata Size

B. Searching Time

Basic ranked searchable method used simple serial searching technique in which every letter of word is compared in every step. Enhanced approach uses Boyer Moore's search algorithm. Improvement in searching time is shown in graph in figure 5 after using enhanced searching mechanism. Table 4 represents the values given by timer implemented in programming of algorithm.

Table 4. Search time for two queries

◦	base time	advanced
Query 1	1.34	0.8
Query 2	1.63	0.98

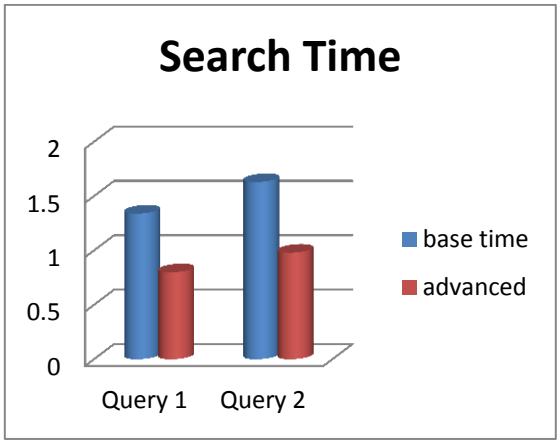

Fig.5. Graph Analysis for Search Time

C. Case studies

Other three factors precision, recall and F-measure depend upon theoretical and quantitative calculations. Two case studies are given for two searches of different query keywords. Assumptions are required for numbers of relevant results returned by both basic and enhanced methods. For evaluating these factors two case studies are considered:

Case study 1:

Total number of files= 15
Let Searching keyword = CLOUD
Retrieved results:
Result files for base method= 9
Result files for advanced method= 8
Let number of relevant results (A) = 7
Let total retrieved files is B = total files - retrieved files
For Base method,

$$B= 15-9=6$$

For advanced method,

$$B= 15-8=7$$

Let C= retrieved files – relevant files
For base method,

$$C= 9-7=2$$

For advanced method,

$$C= 8-7=1$$

$$\text{Recall}=\frac{A}{A+B} * 100 \quad (2)$$

$$\text{Precision}=\frac{A}{A+C} * 100 \quad (3)$$

In Base Method →

$$Recall = 7/(7+6) \rightarrow 7/13 * 100 \rightarrow 53.8$$
$$Precision = 7/(7+2) \rightarrow 7/9 * 100 \rightarrow 77.8$$

In Advance Method →

$$Recall = 7/(7+7) \rightarrow 7/14 * 100 \rightarrow 50$$
$$Precision = 7/(7+1) \rightarrow 7/8 * 100 \rightarrow 87.5$$

F-Measure can be calculated as,

$$F = \frac{2.Precision.Recall}{(Precision+Recall)} \quad (4)$$

For base method,

F-measure = 63.61155

For advanced method,

F-measure = 63.63636

Case Study 2:

Total number of files= 15
Let Searching keyword = PHOTO
Retrieved results:
Result files for base method= 7
Result files for advanced method= 6
Let number of relevant results (A) = 5
Let total retrieved files is B = total files - retrieved files
For Base method,

B= 15-7=8

For advanced method,

B= 15-6=9

Let C= retrieved files – relevant files
For base method,

C= 7-5=2

For advanced method,

C= 6-5=1

In Base Method →
As given in equation (2) and (3) respectively

$$Recall = 5/(5+8) \rightarrow 5/13 * 100 \rightarrow 38.46$$
$$Precision = 5/(5+2) \rightarrow 5/7 * 100 \rightarrow 71.42$$

In Advance Method →
Using equations (2) and (3)

$$Recall = 5/(5+9) \rightarrow 5/14 * 100 \rightarrow 35.71$$
$$Precision = 5/(5+1) \rightarrow 5/6 * 100 \rightarrow 83.33$$

F-Measure can be calculated using equation (4)
For base method,

F-measure = 49.9966

For advanced method,

F-measure = 49.9952

D. Precision

The value of precision can be defined as retrieved relevant documents over total retrieved documents. The value of precision is shown for basic and modified algorithms in table 5 given below. Graphical representation is given in figure Fig.6.

Table 5. Precision Values for Two Queries

◦	base	advance
Query1	77.8	87.5
Query2	71.42	83.33

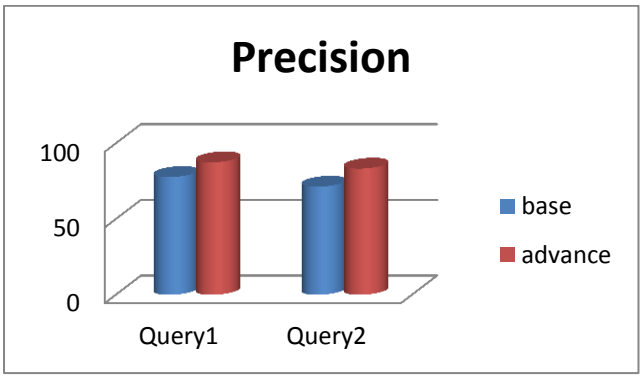

Fig.6. Graphical Representation of Precision

E. Recall

Its value can be described as retrieved relevant documents over relevant documents. Value of recall is calculated using case studies given above and are shown in table 6. Recall and precision are used for accurate measurement of precision. Graphical model is shown in figure 7.

Table 6. Recall Values for Two Queries

◦	Base	advance
Query1	53.8	50
Query2	38.46	35.71

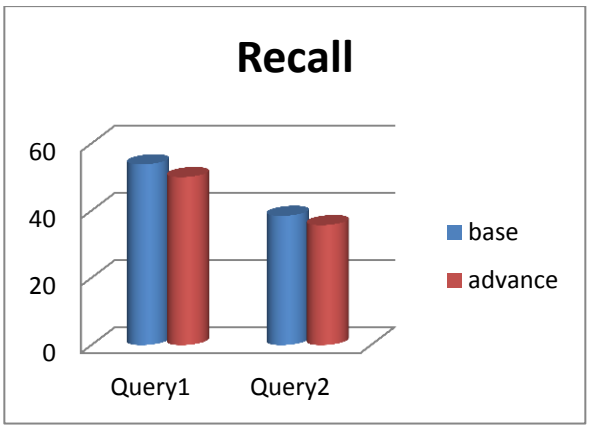

Fig.7. Graphical Representation for Recall

F. F-Measure

F-measure can be defined as overall performance measurement, which decides the importance of precision over recall. In advanced method F-measure improves to some extent.

Table 7. Calculated F-Measure

	base	advance
Query1	63.61155	63.63636
Query2	49.9966	49.9952

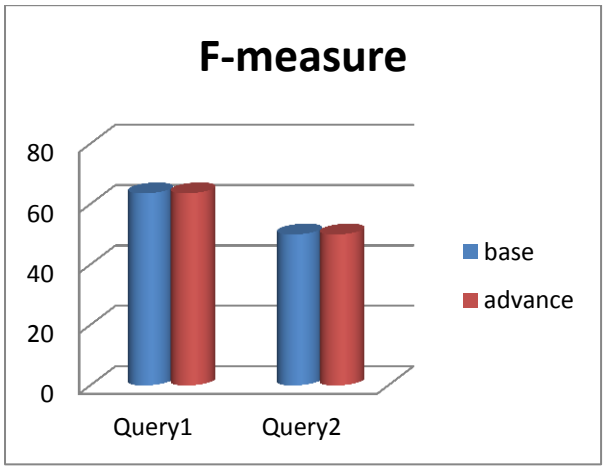

Fig.8. Graphical Representation of F-measure

Above graphs and results depicts that the performance in enhanced scheme improves in the form of metadata storage and speed of searching time.

V. Conclusion

The above discussed work attempts to solve the problem of efficient retrieval of data over encrypted cloud. When OPSE is used for encryption it allows effective RSSE (Ranked searchable symmetric encryption) to be designed. The method given is secure and also achieves the goal of ranked keyword search. In basic SSE scheme there was a large storage space required for metadata creation. An attempt is made to reduce storage space by using some strategies of information retrieval like Porter Stemming and stop words methods. The approach uses Boyer Moore algorithm which makes searching speed very fast. Results and study show that the enhanced approach greatly improves efficiency.

VI. Future Scope

Above given technique greatly improves performance of search scheme. Following the current research, several possible directions are proposed for future work on ranked keyword search over encrypted data. The most attractive among them is the support for multiple keywords. New approaches still need to be designed to completely preserve the order when summing up scores for all the provided keywords. Another interesting direction is to combine advanced crypto techniques, such as attribute-based encryption to enable fine-grained access control in our multi-user settings.

References

[1] M. Bellare, A. Boldyreva, A. O'Neill, "Deterministic and efficiently searchable encryption", Advances in Cryptology-CRYPTO, Springer, Berlin/Heidelberg, (2007), pp. 535- 552.

[2] A. Boldreva, N. Chenette, Y. Lee, A. O'neill, "Order-preserving Symmetric encryption", Advances in Cryptology-EUROCRYPT 2009 Springer, Berlin/Heidelberg, (2009), pp. 224-241.

[3] D. Boneh, G. Di, R. Ostrovsky, G. Persiano, "Public key encryption with keyword search", Advances in Cryptology-Eurocrypt, Springer, Berlin/Heidelberg, (2004), pp 506–522.

[4] N. Cao, C. Wang, M. Li, K.. Ren, W. Lou, "Privacy-preserving multi-keyword ranked search over encrypted cloud data", Proceedings of IEEE INFOCOM.IEEE, Shanghai, China, (2011) pp 829–837.

[5] Y-C. Chang, M. Mitzenmacher, "Privacy preserving keyword searches on remote encrypted data", Applied Cryptography and Network Security. Springer, Berlin/Heidelberg, (2005), pp 442–455.

[6] R. Curtmola, J. Garay, S. Kamara, R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions", Proceedings of the 13th ACM conference on Computer and communications security.ACM, Alexandria, VA, USA, (2006), pp 79–88.

[7] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, W. Lou " Fuzzy keyword search over encrypted data in cloud computing", Proceedings of IEEE INFOCOM.IEEE, San Diego, CA, USA, (2010), pp 1–5.

[8] DX. Song, D. Wagner, A. Perrig, "Practical techniques for searches on encrypted data", Proceedings of IEEE Symposium on Security and Privacy, IEEE, Berkeley, California, (2000), pp 44–55.

[9] E. Stefanov, C. Papamanthou, E. Shi, "Practical Dynamic Searchable Encryption with Small Leakage", NDSS '14, San Diego, CA, USA, (2014).

[10] C. Wang, N. Cao, J. Li, K. Ren, W. Lou, "Secure ranked keyword search over encrypted cloud data", 30th IEEE International Conference on Distributed Computing Systems (ICDCS). IEEE, Genoa, Italy, (2010), pp 253–262.

[11] C. Wang, N. Cao, K. Ren, W. Lou, "Enabling Secure and Efficient Ranked Keyword Search over Outsourced Cloud Data", IEEE Trans Parallel DistribSyst23(8):1467–1479, (2012).

[12] Z. Xia, Y. Zhu, X. Sun and L. Chen, "Secure semantic expansion based search over encrypted cloud data supporting similarity ranking.", Journal of Cloud Computing, Springer 3.1, (2014), pp 1-11.

[13] C. Yang, W. Zhang, J. Xu, N. Yu, "A Fast Privacy-Preserving Multi-keyword Search Scheme on Cloud Data", International Conference on Cloud and Service Computing (CSC). IEEE, Shanghai, China, (2012), pp 104–110.

[14] S. Zerr , D. Olmedilla, W. Nejdl, "Zerber+r: Top-k Retrieval from a Confidential Index," Proc. EDBT '09, 2009.

[15] N. Cao, C. Wang, M. Li, K. Ren, "Privacy-Preserving

Multi-Keyword Ranked Search Over Encrypted Cloud Data," IEEE INFOCOM, 2011, pp. 829–37.
[16] C. Wang, N. Cao, J. Li, K. Ren, "Secure Ranked Keyword Search Over Encrypted Cloud Data," Proc. ICDCS '10, 2010.
[17] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, "Fuzzy Keyword Search Over Encrypted Data in Cloud Computing," Proc. IEEE INFOCOM '10 Mini-Conf., San Diego, CA, Mar. 2010.
[18] C. Wang, K. Ren, S. Yu, "Achieving Usable and Privacy-Assured Similarity Search over Outsourced Cloud Data," Proc. IEEE INFOCOM '12, Orlando, FL, Mar. 2012.
[19] M. Li, S Yu, N. Cao, W. Lou, "Authorized Private Keyword Search over Encrypted Data in Cloud Computing," 31st Int'l. Conf. Distributed Computing Systems, 2011, pp. 383–92.
[20] M. Li, S. Yu, K. Ren, Y. Hou, W. Lou, "Toward Privacy-assured and searchable cloud data services", Network, IEEE, 27(4), (2013), pp. 56-62.
[21] W. Zhou, N. R. Smalheiser, & C. Yu, "A tutorial on information retrieval: basic terms and concepts", Jornal of biomedical discovery and collaboration, 1(1), (2006), pp. 2.
[22] C. Ramasubramanian, R. Ramya, "Effective pre-processing activities in text Mining using Improved Porter's Stemming Algorithm", International Journal of Advanced Research in Computer and Communication Engineering, 2(12), (2013), pp. 2278-1021.
[23] S. Goyal, "Public vs private vs hybrid vs community-cloud computing: A critical review", International Journal of Computer Network and Information Security (IJCNIS), (2014), pp 20-29, 6(3).
[24] D. Singh, J. Singh, and A. Chhabra, "Failures in cloud computing data centres in 3-tier cloud architecture", International Journal of Information Engineering and Electronic Business(IJIEEB), (2012), PP 1-8, 4(3).
[25] J. Singh, "Study of response time in cloud computing", International Journal of Information Engineering and Electronic Business (IJIEEB), 2014, pp 36-43, 6(5).
[26] A. Swaminathan, Y. Mao, G-M. Su, H. Gou, A.L. Vama, S. He, M. Wu and D.W. Oard, "Confidentiality-Preserving Rank-Ordered Search," Proc. Workshop Storage Security and Survivality, 2007.
[27] H. Hu, J. Xu, C. Ren, and B. Choi, "Processing Private Queries over Untrusted Data Cloud through PrivacyHomomorphism," Proc. IEEE 27th Int'l Conf. Data Eng. (ICDE), 2011.
[28] C. Liu, L. Zhu, L. Li, Y. Tan , "Fuzzy keyword search on encrypted cloud storage data with small index", IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS). IEEE, Beijing, China, (2011), pp 269–273.
[29] P. S. Hersarlo, "Security, privacy and trust challenges in cloud computing and solutions", International Journal of Computer Network and Information Security(IJCNIS), 2014, pp 34-40, 6(8).
[30] M. I. Alam, M. Pandey and S. S. Rautaray, "A comprehensive survey on cloud Computing", International Journal of Information Technology and Computer Science (IJITCS), 2015, pp 68-79, 7(2).
[31] M. Y. Saeed, M. N. A. Khan, "Data Protection Techniques for Building Trust in Cloud Computing", International Journal of Modern Education and Computer Science(IJMECS), 2015, pp 38-47, 7(8).
[32] A. Zia, M. N. A. Khan, "Identifying Key Challenges in Performance Issues in Cloud Computing" International Journal of Modern Education and Computer Science (IJMECS), 2012, pp 59-68, 4(10).
[33] M. Jabalameli, A. Arman, M. Nematbakhsh, "Improving the Efficiency of Term Weighting in Set of Dynamic Documents" International Journal of Modern Education and Computer Science (IJMECS), 2015, pp 42-47, 7(2).
[34] J. Bringer, H. Chabanne, "Embedding edit distance to enable private keyword search", Human-centric Comput Inf Sci, vol.2 (1), (2012), pp. 1–12.
[35] E. R. Daniel, "Cloud Search and Democratization of Information Retrieval", SIGIR'12, August 12-16, (2012), Portland, Oregon, USA, pp. 1022-1023
[36] J. Yu, P. Lu, Y. Zhu, G. Xue, and M. Li (2013), "Toward secure Multikeyword Top-k retrieval over encrypted cloud data", IEEE transactions on dependable and secure computing, (4), pp. 239-250.

Authors Profiles

Rajpreet Kaur is a student of masters in Computer science and engineering in Chandigarh group of colleges, College of Engineering, Landran (Mohali), Punjab. She has completed her Bachelor degree in computer science & engineering from Baba Banda Singh Bahadur Engineering College, Fatehgarh Sahib. Area of her research is searching in Cloud Computing. She has attended a national conference on recent trends in cloud computing (RTICC) in Chandigarh Group of Colleges, College of Engineering, Landran (Mohali).

Manish Mahajan is associate professor and head of department in Chandigarh group of colleges, College of Engineering, Landran (Mohali), Punjab. Areas of his interest are Image Processing, Cloud computing and Networking.