A Theoretical Analysis for relevancy of files in Enhanced Ranking Based Cloud Search with Improved Metadata Storage

Rajpreet Kaur and Manish Mahajan
1 Research scholar, CGC College of engineering, Landran (Mohali), Punjab, India
2 Head of Department, CGC College of engineering, Landran (Mohali), Punjab, India

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.

Keywords: Cloud data privacy, SSE, Ranking, Porter Stemming, Boyer Moore algorithm

1. 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.

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

A simple model for ranked searching over encrypted cloud data is shown below in figure 1.

Fig. 1 Architecture for secure ranked search scheme

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.

1.2 Order Preserving Symmetric Encryption
OPSE is basically an encryption scheme which preserves numerical order of plain text of data while it is in encrypted form. It uses cryptographic basics and permutation functions.

2. 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. 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. Boldyreva [2] first gave a cryptographic primitive of OPSE and implemented it for secure search framework.

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.

Including these some literature related to Information Retrieval (IR) is also surveyed. Basic introduction to IR technologies is given in [21]. A thorough study of improved Porter Stemming algorithm is given by Ramasubramanian et al. in [22].

3. Proposed Model of Work
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.

3.1 Basic design of system
3.1.1 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.

Fig. 2 Flowchart for Metadata creation

In first step of metadata creation main three procedures of information retrieval are considered. These three 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”.

**Porter Stemmer Algorithm:** A simple algorithm for suffix stripping is discussed below:

**Step 1:** Plurals and past participles
SSES → SS, for example, Possesses → Possess
(*verb*)ING → verb, example, going → go
Step 2: adjective → noun, noun → verb, and
Noun → adjective, example, Realization → realize,
Darkness → dark, Operational → operate etc
Step 3: (m>0)ICATE → IC, example,
Authenticate → Authentic
Step 4: (m>1)al → , example, arrival → arrive
(m>1)ance → , example, Allowance → Allow
Step 5: (m>1)y → , example, Furry → fury
(m>1 and *d and *L) → Single letter, example, Control → Control

3.1.2 Searching encrypted data
In searching Boyer Moore’s algorithm is implemented. **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. This algorithm contains three clever ideas- the right to left scan, the bad character rule, good suffix shift rule.

\[
\text{Score}(t, D) = \text{TF}(t, D) \times \text{IDF}(t)
\]
Here \( t \) is term/keyword and \( D \) is Document.

**Table 1: Boyer Moore’s Algorithm**

<table>
<thead>
<tr>
<th>Preprocessing stage</th>
<th>{Preprocessing stage}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Given the pattern ( P ), Compute ( L'(t) ) and ( l'(t) ) for each position ( i ) of ( P ), and compute ( R(x) ) for each character ( x \in \Sigma ).</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Search stage</th>
<th>{Search stage}</th>
</tr>
</thead>
<tbody>
<tr>
<td>While ( k \leq m ) do</td>
<td></td>
</tr>
<tr>
<td>( i := n );</td>
<td></td>
</tr>
<tr>
<td>( n := k );</td>
<td></td>
</tr>
<tr>
<td>While ( i &gt; 0 ) and ( P(i) = T(h) ) do</td>
<td></td>
</tr>
<tr>
<td>( i := i - 1 );</td>
<td></td>
</tr>
<tr>
<td>( h := h - 1 );</td>
<td></td>
</tr>
<tr>
<td>end;</td>
<td></td>
</tr>
<tr>
<td>if ( i = 0 ) then</td>
<td></td>
</tr>
<tr>
<td>begin</td>
<td></td>
</tr>
<tr>
<td>report an occurrence of ( P ) in ( T ) ending at position ( k ).</td>
<td></td>
</tr>
<tr>
<td>( k := k + n - l'(2) );</td>
<td></td>
</tr>
<tr>
<td>end</td>
<td></td>
</tr>
<tr>
<td>Else</td>
<td></td>
</tr>
<tr>
<td>Shift ( P ) (increase ( k )) by the maximum amount determined by (extended) bad character rule and the good suffix rule.</td>
<td></td>
</tr>
<tr>
<td>End.</td>
<td></td>
</tr>
</tbody>
</table>

4. 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

**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 2 given below.

Table 2: Metadata Size

<table>
<thead>
<tr>
<th></th>
<th>base</th>
<th>advance</th>
</tr>
</thead>
<tbody>
<tr>
<td>For 15 Files</td>
<td>72</td>
<td>48</td>
</tr>
</tbody>
</table>

Searching Time: Improvement in searching time is shown in graph in figure 5 after using, Boyer Moore’s algorithm. Table 3 represents the values given by timer implemented in programming of algorithm.

Table 3: Search time for two queries

<table>
<thead>
<tr>
<th></th>
<th>base time</th>
<th>advanced</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query 1</td>
<td>1.34</td>
<td>0.8</td>
</tr>
<tr>
<td>Query 2</td>
<td>1.63</td>
<td>0.98</td>
</tr>
</tbody>
</table>

Other three factors precision, recall and F-measure depend upon theoretical and quantitative calculations. For evaluating these factors two case studies are considered:

**Case study 1:**
Total no. Of files= 15
Let Searching keyword = CLOUD

Retrieved results:
Result files for base method= 9
Result files for advanced method= 8
Let No. of relevant results (A) = 7
Let total retrieved files is $B = \text{total files} - \text{retrieved files}$
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,
$C= 8-7=1$
Recall=$\frac{A}{A+B} \times 100$, Precision=$\frac{A}{A+C} \times 100$

In Base Method →
Recall = $\frac{7}{7+6} \times 100 = 53.8$
Precision = $\frac{7}{7+2} \times 100 = 77.8$

In Advance Method
Recall = $\frac{7}{7+7} \times 100 = 50$
Precision = $\frac{7}{7+1} \times 100 = 87.5$

F-Measure can be calculated as,
$F = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$
For base method,
F-measure = 63.61155
For advanced method,
F-measure = 63.63636

**Case Study 2:**
Total no. Of files= 15
Let Searching keyword = PHOTO

Retrieved results:
Result files for base method= 7
Result files for advanced method= 6
Let No. of relevant results (A) = 5
Let total retrieved files is $B = \text{total files} - \text{retrieved files}$
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,
$C= 6-5=1$
Recall=$\frac{A}{A+B} \times 100$, Precision=$\frac{A}{A+C} \times 100$

In Base Method →
Recall = $\frac{5}{5+8} \times 100 = 38.46$
Precision = $\frac{5}{5+2} \times 100 = 71.42$
In advance method →
Recall = 5/(5 + 9) → 5/14 * 100 → 35.71
Precision = 5/(5 + 1) → 5/6 * 100 → 83.33

F-Measure can be calculated as,
\[ F = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{(\text{Precision} + \text{Recall})} \]
For base method, 
F-measure = 49.9966
For advanced method, 
F-measure = 49.9952

**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 4 given below. Graphical representation is given in figure 6.

Table 4: Precision values for two queries

<table>
<thead>
<tr>
<th></th>
<th>base</th>
<th>advance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query1</td>
<td>77.8</td>
<td>87.5</td>
</tr>
<tr>
<td>Query2</td>
<td>71.42</td>
<td>83.33</td>
</tr>
</tbody>
</table>

**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 5. Recall and precision are used for accurate measurement of precision. Graphical model is shown in figure 7.

Table 5: Recall values for two queries

<table>
<thead>
<tr>
<th></th>
<th>base</th>
<th>advance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query1</td>
<td>53.8</td>
<td>50</td>
</tr>
<tr>
<td>Query2</td>
<td>38.45</td>
<td>35.71</td>
</tr>
</tbody>
</table>

**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 6: Calculated F-Measure

<table>
<thead>
<tr>
<th></th>
<th>base</th>
<th>advance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query1</td>
<td>63.61155</td>
<td>63.63636</td>
</tr>
<tr>
<td>Query2</td>
<td>49.9966</td>
<td>49.9952</td>
</tr>
</tbody>
</table>

**Conclusions**

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.

**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**


**First Author** Rajpreet kaur is pursuing her master’s degree in computer science and engineering from cgc, college of engineering Landran (Mohali). She has completed her B. Tech degree from BBSBEC, Fatehgarh sahib. Area of her research is Cloud computing search. Her One review article is published in an international journal and one article is published in conference.