International Journal of Research
Available at https://edupediapublications.org/journals

e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 15
May 2018

# An Efficient Search Scheme over Encrypted Data on Mobile Cloud

[1]SUNDARAPALLI VIJAYA DURGA M.Tech(CSE)

Email : svdurga27@gmail.com

Chaitanya Institute Of Science & Technology, Madhavapatnam, Kakinada

[2]VENUTHURU PRADEEP M.Tech(CSE)

ASSOCIATE PROFESSOR

Email : pradeepvenuthuru@gmail.com

Chaitanya Institute Of Science & Technology, Madhavapatnam, Kakinada

**Abstract-** Cloud storage is currently a trending technology owing to huge storage provided at negligible cost. Cloud services are prone to attacks due to lack of centralized control. To maintain data security and privacy, data should be stored in an encrypted format on cloud storage. Mobile Cloud Storage (MCS) provides storage solutions to mobile device users by enabling storage and retrieval of data on cloud through wireless communication. MCS incurs new challenges due to limitation of mobile devices in terms of bandwidth, computational capability and payable traffic fee. Encrypted search on mobile cloud results in huge processing overhead. To overcome the aforementioned drawbacks, we proposed a traffic and energy efficient encrypted multiple keywords ranked search for mobile device users. The proposed architecture offloads computation from mobile devices to cloud server and optimizes the communication trip. Multiple keyword search narrows down the result set providing the most relevant data. We have testified the results by graphical analysis in terms of time complexity and battery consumption and found multiple keywords ranked search method efficient and user friendly.

**Keywords** – Encrypted Search, Energy Efficiency, Mobile Cloud Storage, Multiple Keyword Search.

## I. INTRODUCTION

Cloud storage is a storage model in which massive data can be stored on the cloud and it can be accessed anytime, anywhere by a user across a network. Mobile Cloud Storage (MCS) is a form of cloud storage that enables the mobile device users to store and retrieve data on the cloud through wireless communication [1]. The data privacy issue is paramount in cloud storage system, so the sensitive data is encrypted by owner before outsourcing onto the cloud, and data users retrieve the interested data by encrypted search scheme. These traditional data encryption methods cannot be imported directly in MCS due to limited computing and battery capacities of mobile devices. Therefore, MCS is in need of an efficient encrypted search scheme that focuses on bandwidth and energy efficiency. TEES (Traffic and Energy Saving Encrypted Search) architecture was introduced to satisfy the MCS requirements. This method uses Ranked Keyword Search as a data encrypted search scheme. The TEES architecture is shown in figure 1 and figure 2.

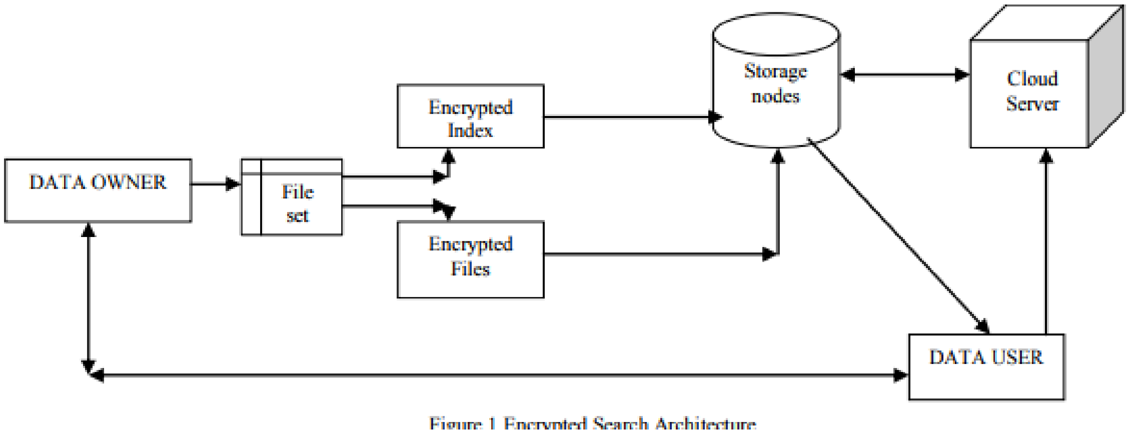


Figure 1 Encrypted Search Architecture

TEES architecture has a data owner and data user. Data owner uploads the files and index table in an encrypted format on the cloud server. This process is shown in Figure 2

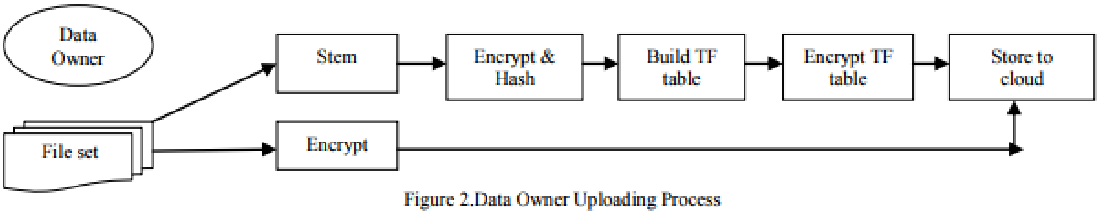


Figure 2.Data Owner Uploading Process

The data owner first performs stemming of each word in a document to retain the word stem. The term obtained is then encrypted and hashed. It is stored in index table. The index table is encrypted and stored on cloud server. The index table is a Term Frequency (TF) table [2]. It keeps a record of the number of times a term appears in a document. Data owner also encrypts the files and stores it on the cloud server. Order Preserving Encryption (OPE) is used as an encryption algorithm [3].

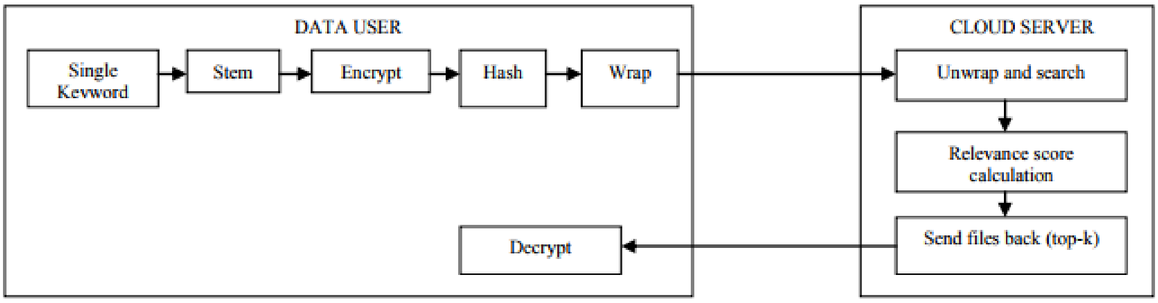


Figure 3.Data User Search and Retrieval Process

A data user can access a file only after being authenticated by the data owner. Search and retrieval has the following steps as shown in figure 3:

1. An authenticated user enters a single keyword.

2. The keyword is then stemmed, encrypted and hashed by the user to match its entry in index table.

3. Noise is also added to this hashed keyword to make it unrecognisable. wrap() method is used for this purpose.

4. Encrypted keyword is then sent to the cloud server.

5. The cloud server receives the encrypted keyword. It unwraps the keyword and searches for it in the index.

6. The cloud server calculates the relevance scores and accordingly sends the top-k relevant files to the mobile data user.

7. The data user decrypts the files

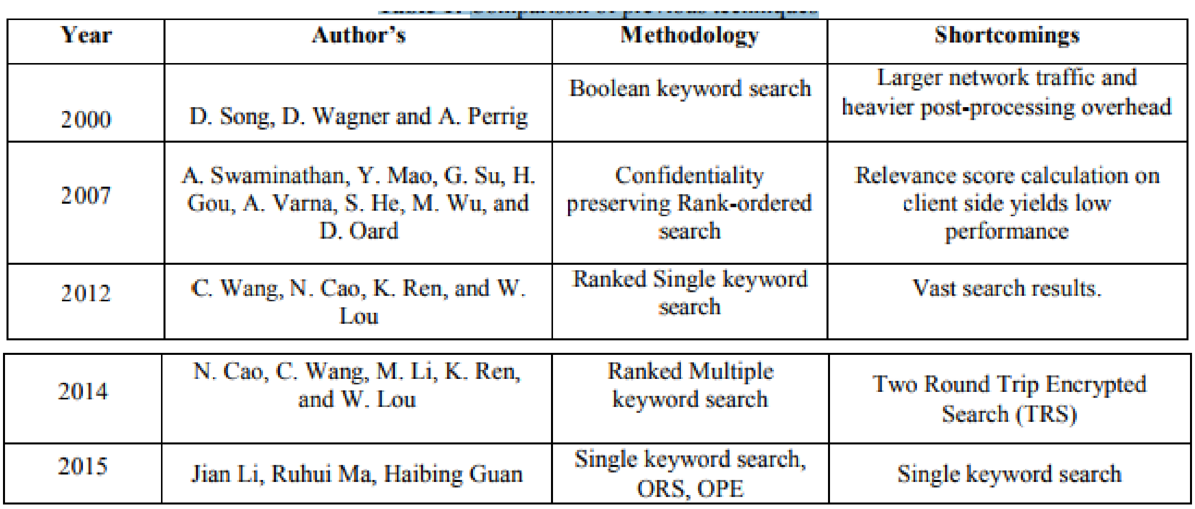
TEES architecture has simplified the search and retrieval process for encrypted data on mobile cloud. The computation and relevance score calculation is performed by the cloud server thus saving the energy consumption of mobile devices.

## II. RELATED WORK

Before 2000, there was hardly any work aimed to provide a solution for searching on encrypted data. In 2000, D. Song, D. Wagner and A. Perrig proposed the different techniques for searching operation over encrypted data [4]. These techniques for remote searching on encrypted data were provided with security proofs and have a number of crucial advantages. All these techniques were based on Boolean keyword search. Boolean keyword search is not suitable for cloud storage since it sends all matching files to the clients, and therefore incur a larger amount of network traffic and a heavier post-processing overhead for the mobile devices. TF-IDF is a statistic which reflects how important a word is to a document in a collection [5]. Y. Chang and M. Mitzenmacher provided keyword search scheme, but it does not send back the most relevant files [6]. R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu proposed a one-to-one mapping OPE which will lead to Statistics Information Leak Control [3]. A. Swaminathan, Y. Mao, G. Su, H. Gou, A. Varna, S. He, M. Wu, and D. Oard proposed a confidentiality-preserving rank-ordered search [7]. This scheme displays low performances as the relevance scores are computed on the client side, increasing its workload. C. Wang, N. Cao, J. Li, K. Ren, and W. Lou presented a secure ranked keyword search over encrypted cloud data [8]. However, in their work the terms are closely related to the files which could lead to potential information leak.

In 2015, Jian Li, Ruhui Ma, Haibing Guan proposed TEES: An Efficient Search Scheme over Encrypted Data on Mobile Cloud [9]. TEES architecture was introduced to create a traffic and energy efficient encrypted keyword search tool over mobile cloud storages. It offloaded the relevance scores calculation to the cloud server reducing the burden on the mobile clients. It also shortened the retrieval process so that the data user can receive the most relevant files within only one communication. However, TEES architecture uses Single keyword search thus yielding far too coarse results. To improve the search result accuracy as well as to enhance the user searching experience, it is necessary to support multiple keyword searches to narrow down the results. Multi-keyword is potentially the future main stream encrypted search scheme with higher searching accuracy. Table 1 compares all the previous techniques and mentions their shortcomings

Table 1: Comparison of previous techniques



| Year | Author's | Methodology | Shortcomings |
|---|---|---|---|
| 2000 | D. Song, D. Wagner and A. Perrig | Boolean keyword search | Larger network traffic and heavier post-processing overhead |
| 2007 | A. Swaminathan, Y. Mao, G. Su, H. Gou, A. Varna, S. He, M. Wu, and D. Oard | Confidentiality preserving Rank-ordered search | Relevance score calculation on client side yields low performance |
| 2012 | C. Wang, N. Cao, K. Ren, and W. Lou | Ranked Single keyword search | Vast search results. |
| 2014 | N. Cao, C. Wang, M. Li, K. Ren, and W. Lou | Ranked Multiple keyword search | Two Round Trip Encrypted Search (TRS) |
| 2015 | Jian Li, Ruhui Ma, Haibing Guan | Single keyword search, ORS, OPE | Single keyword search |

## III. PROPOSED METHODOLOGY

The TEES architecture has created a traffic and energy efficient encrypted keyword search tool over mobile cloud storages. The architecture however suffers from two major drawbacks:

1. TEES supports only single keyword encrypted search.

2. TEES has low encryption security as OPE (Order Preserving Encryption) is used.

We proposed a modified architecture for search and retrieval on mobile cloud. This architecture eliminates the drawbacks of TEES architecture. It supports multiple keywords producing a narrow and accurate result set. The experimental results prove that the battery consumption and time required to search documents with multiple keywords is less than single keyword search. The proposed architecture uses an advanced and efficient AES (Advanced Encryption Standard) algorithm to enhance the security. The proposed architecture is shown in Figure 4 and Figure 5

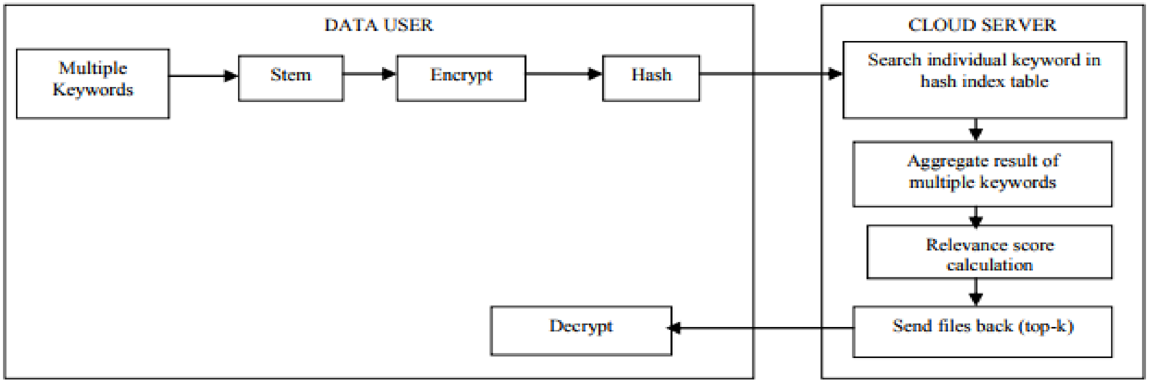

DATA USER
Multiple Keywords
Stem
Encrypt
Hash
Decrypt
CLOUD SERVER
Search individual keyword in hash index table
Aggregate result of multiple keywords
Relevance score calculation
Send files back (top-k)

Figure 4.Modified Search and Retrieval Process between Data User and Cloud Server

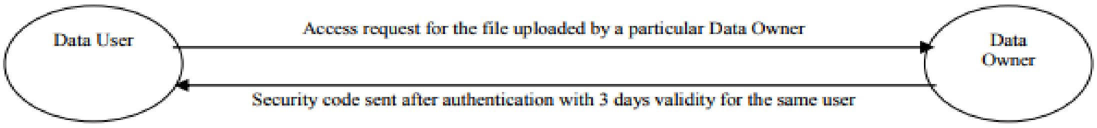

Data User
Access request for the file uploaded by a particular Data Owner
Security code sent after authentication with 3 days validity for the same user
Data Owner

Figure 5.Data user Authentication by Data Owner

The data owner uploads the encrypted file set and encrypted hash index table in the similar manner as shown in figure 2. The modifications are done in search and retrieval process between mobile data user and cloud server. The modified process is as follows:

1. The data user may enter ‘n’ keywords in the search box.

2. Each keyword is stemmed, encrypted and hashed to place its entry in index table.

3. These encrypted keywords are then sent to the cloud server.

4. The cloud server searches individual keyword entry in hash index table.

5. The cloud server computes the aggregation of multiple keywords.

6. The cloud server calculates tf-idf for the occurrence of the aggregated result of multiple keywords.

7. It picks the position of the files corresponding to a keyword or a set of keywords and returns the top-k files to the mobile data user without decrypting them.

8. The user selects the files and requests the data owner for file access as shown in figure 5.

9. The data owner sends a 4 digit security code to the registered e-mail id of user after authentication.

10. The data user enters the code and downloads the decrypted files. The data user decrypts these files in the mobile client and recovers the original data.

## IV. IMPLEMENTATION

We have used a data set of 500 documents. In order to achieve enhanced security and refined search results, we have implemented the modules with new routines. We have used a cloud server namely cloudserver.somee.com. Alternatively, a mobile data user can mention the IP address or the link of the cloud server to which the user wants to connect by executing the mobile cloud application also. After entering the respective link and pressing click button, the mobile data user is directed to the home page. The home page consists of Login option. Login has username and password fields. There is a Role field drop box containing three options: ‘Data user’, ‘Data Owner’ and ‘Admin’. Data Owner is the person who uploads documents which can have .pdf, .txt, or .doc extension. Data User is the person who accesses the uploaded documents by searching the documents based on multiple keywords. Data User is verified by data owner through security code. Data owner mails the security code to the data user. The code has two days validity. This means the data user can access the document anytime for two days without verification process. Admin can analyze the graphical results in terms of time complexity and battery consumption.

**Data User:** Every data user has to register for the first time. The registration details contain Email id field. This is used by data owner to send the security code for file access. Data user can search by entering keywords in the search box. Data user has three options for searching the file. The first option is for plain text file. This file is not encrypted before storing on cloud server. Second and third options are single keyword and multiple keyword searches. These files are encrypted before outsourcing on cloud server and decrypted after downloading. The search results are presented in the form of a table which contains the details of owner and the frequency of the keywords. If multiple keywords are entered, then the result is the aggregation of all the keywords in each file. We have used top-k algorithm based on tf-idf where the value of k is set to three. The data user can select any file. After selection, the request is received by the data owner. The data owner can view all the pending access requests to the uploaded files in Log record.

**Data Owner:**

The data owner home page has 2 options:

i) Upload Resources

ii) Log

The data owner log contains the data user request to access the files. If the data owner wishes to grant access, then it can select the particular user. On clicking the select option, the security code is generated and sent to the respective data user. This is used as a security measure for authentication. The data owner can enter a relevant keyword for the document to be uploaded. The data owner can upload .txt, .doc or .pdf files. The data user should enter the security code for getting authenticated by the data owner for the first time. The data user can access this code through registered e-mail id. As soon as the data user enters the code, the data user can access the file by clicking on the download option.

**Build Index table:** The data owner starts by collecting the files he wants to store into the cloud. The file is stored in encrypted format. AES encryption is used. The data owner builds a secure index by encrypting and hashing every term to fix its entry in the index table. The index table is then created by the data owner. Finally, the data owner encrypts the index and stores it into the cloud server, together with the encrypted file set. SHA256 is used as hashing algorithm to generate hash word. To find the top-k relevant files, a top-k ranking algorithm is used as shown below.

**Algorithm: top-k ranking algorithm**

**Input:** keyword w, k is set to three.

**Output:** topFiles

```
 if this request is sent by a “legal” user then
for each file Fc Є F do
Calculate Score(w, Fc).
end for
end if
if this request is sent by a overdue user then
for each file Fc Є F do
Calculate Score(w, Fc) but with a warning.
end for else
Return “No Permission”.
 end if
Rank the scores to get top-k files
topFiles = {topF1, topF2 ,....., topFk}.
return
topFiles.
```

where Fc is a certain file in the file set.

## V. RESULT ANALYSIS

### 5.1. Parameters for result analysis:

We have considered the following two parameters as performance metrics:

**5.1.1. Time Complexity:** The file search and retrieval time depends on the file size and network bandwidth. FSRT of our architecture is improved by introducing only a single communication trip, relevance score calculation offload and Multiple Keyword Ranked Search. Plain text search (PTS) requires least FSRT as it does not spend any time on stemming, encryption, hashing or wrapping. However, PTS is insecure.

**5.1.2. Battery Consumption:** We have calculated the energy consumption of mobile device by taking the difference between the battery percentage of device during searching and battery percentage after downloading. This is done for both single keyword and multiple keywords. The results are supported by graph of time complexity and battery consumption.

The admin can view these graphs through admin account. In our experiments, we have used a data set of 500 files with different sizes and a free cloud server from https://somee.com. For mobile client, we have used Android Studio.

We have also created an application for mobile device. The project can run on any mobile device by visiting the link: http://cloudserver.somee.com/. The following are the resources used for ASP.Net web hosting: 150MB storage, 5GB transfer ASP.Net 4.6/4.5/4.0/3.5/2.0 15MB MSSQL 2008R2/2012/2014 Free third level domain FTP access

**Testing data example: File 1: “java” keyword occurrence:**

1, java servlet pages:0 File 2: “java” keyword occurrence:

2, java servlet pages:1 File

3: “java” keyword occurrence: 3, java servlet pages:1 File

4: “java” keyword occurrence: 4, java servlet pages:1 File

5: “java” keyword occurrence: 5, java servlet pages:0 File

6: “java” keyword occurrence: 3, java servlet pages:3

A mobile data user searches for “java servlet pages” by using single keyword and multiple keyword approach. If single keyword is used i.e. java then the user will get file 5, file 4 and file 3 as top-3 files. This is inappropriate result as the highest occurrence of java servlet pages is in file 6 followed by file 2, 3 and 4. If multiple keyword is used i.e. java servlet pages then the user gets file 6, 2 and 3 which is a correct result

**5.2. Energy Consumption Graph**

As energy consumption is critical for mobile devices, we evaluate efficiency of our implementation with respect to energy consumption or battery consumption of a mobile device user as shown in figure 6. To calculate battery consumption, we have used the battery of the device. Before search operation, the amount of battery remaining is captured through sql query. Then, after search operation, the amount of battery remaining is captured and the battery consumption is the difference between these two values. If Android Studio is used, then by using Battor [10] software, we can accurately measure the energy consumption. Battor is a phone power monitor

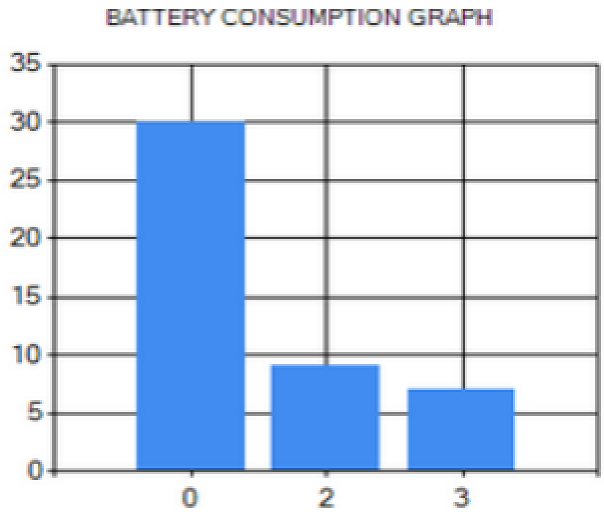


Figure 6.Battery consumption graph

Y-axis indicates the battery consumption in mA (milli amperes) which is one thousandth of an ampere, a measure for small electric currents.

0 represents Single keyword method

2 represents Multiple keyword method and

3 represent plain text search method

**5.3. Time Complexity Graph**

We compare the file search and retrieval time for single keyword search, multiple keyword search and plain text search as shown in figure 7. We tested the FSRT for different files. We observe that the FSRT of plain text search is the shortest since it does not have to perform any security computation. The FSRT of single keyword is same as multiple keywords.

The file retrieval time only depends on the file size and network bandwidth. When offered a greater bandwidth, the scheme becomes more efficient.

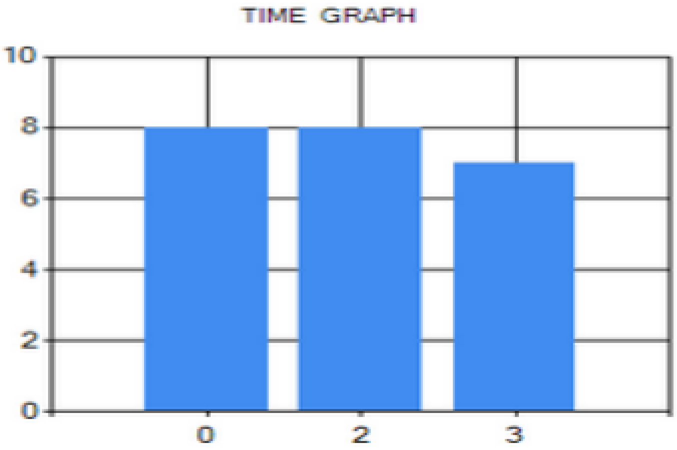


Figure 7.Time complexity graph

## VI. CONCLUSION

Mobile Cloud Storage (MCS) provides storage solutions to mobile device users by enabling storage and retrieval of data on cloud through wireless communication. But, MCS incurs new challenges due to limitations of mobile devices in terms of computational capacity, battery life, bandwidth and payable traffic fee. Due to these limitations, encrypted search over mobile cloud results in huge processing overhead. TEES architecture is an initial attempt to create a traffic and energy efficient encrypted keyword search tool over mobile cloud storages. TEES uses single keyword search scheme. Single keyword search produces a broad result set. Our proposed methodology narrow down the result set by using multiple keyword search scheme. This improves the search result accuracy and enhances user searching experience. The methodology and implementation details are supported by the graphical results. The time complexity and battery consumption of mobile device is minimized for multiple keyword searches as compared to single keyword search.

## REFERENCES

[1] D. Huang, “Mobile cloud computing,” IEEE COMSOC Multimedia Communications Technical Committee (MMTC) E-Letter, 2011.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent dirichlet allocation,” the Journal of machine Learning research, vol. 3, 2003, pp. 993–1022.

[3] R. Agrawal, J. Kiernan, R. Srikant and Y. Xu, “ Order preserving encryption for numeric data,” in Proceedings of the 2004 ACM SIGMOD international conference on Management of data. ACM, 2004, pp. 563-574.

[4] D. Song, D. Wagner, and A. Perrig, “Practical techniques for searches on encrypted data,” in Security and Privacy, 2000. S&P 2000. Proceedings. 2000 IEEE Symposium on. IEEE, 2000, pp. 44–55.

[5] A. Aizawa, “ An Information-theoretc perspective of tf-idf measures,” Information Processing and Management, 2003, vol. 39, pp. 45-65.

[6] Y. Chang and M. Mitzenmacher, “ Privacy preserving keyword searches on remote encrypted data,” in Applied Cryptography and Network Security. Springer, 2005, pp. 391-421.

[7] A. Swaminathan, Y. Mao, G. Su, H. Gou, A. Varna, S. He, M. Wu, and D. Oard, “ Confidentiality-preserving rank-ordered search,” in Proceedings of the 2007 ACM workshop on Storage security and survivability. ACM 2007, pp. 7-12.

[8] C. Wang, N. Cao, J. Li, K. Ren, and W. Lou,” Secure ranked keyword search over encrypted cloud data,” in Distributed Computing Systems (ICDCS), 2010 IEEE 30th International Conference on. IEEE, 2010, pp. 253-262.

[9] Jian Li, Haibing Guan, Ruhui Ma, “ TEES: An Efficient Search Scheme over Encrypted Data on Mobile Cloud,” in IEEE Transactions on Cloud Computing, 2015.

[10] A. Schulman, T. Schmid, P. Dutta, and N. Spring, “Demo: Phone power monitoring with battor.” MobiCom, 2011