ABSTRACT
The cloud computing motivates the data owners to outsource their data to the cloud servers. Because of security of outsourced data, encrypt the data before sending to the cloud server. Earlier research works were focused on single keyword search, and some consider the multi-keyword search on encrypted data, but none of them consider the dynamic updating of the index and keywords in the documents. In this paper proposed the fast and secure multi-keyword search over the encrypted cloud data. This scheme uses a hybrid data structure, “skip graph bloom filter” for the fast index management. It also added an algorithm that handles the dynamic changes in the end user by keeping the status, version, and modification dates. Performance evaluation and analysis are done to demonstrate the effectiveness of proposed search scheme.

Keywords
Cloud Computing, Skip Graph, Secure Multi-keyword search, Bloom Filter

1. INTRODUCTION
Nowadays, Cloud computing became the most suitable platform for various computing services, which allows users to remotely store and retrieve their data into and from the cloud. Companies like Google, Microsoft, and Amazon successfully provide the several benefits of cloud computing to the users. A huge amount of technical and economic advantages of using cloud computing introduces the new challenges towards the privacy of user as well as the sensitive data. Data encryption is the only solution to provide the secure remote access to data and maintain the privacy of the user. The encryption of the outsourced data gives the new problem of information retrieval. The primary concern is to retrieve the data from the cloud server with relative ranking and most relevant result.

The various researches reported that the searchable encryption scheme as the main search scheme for search in encrypted data. Some constructive scheme for the search in encrypted data has been proposed in [4]. Because of the changes in user demands, only the keyword oriented searchable encryption scheme does not provide the efficient results.

Ranked search enables data users to find the most relevant information quickly, rather than burdensomely sorting through every match in the content collection” [12] [11] [13]. In the cloud environment ranked search is more preferred as it reduces the network traffic and available in the “pay-as-you-use” cloud service. In secure searching schemes, the search operation should not leak any information related to the searched query. Also, it helps in better the search accuracy and enhances the user searching experience. Some search scheme supports multi keyword search, but the obtained result was not relevant and efficient. It may require providing a set of multi keywords instead of only one, to yield relevant results. These keywords can help in narrow down results produce in the searching scheme. Coordinate matching as discussed in [6], helps in refining the most relevant similarity search. Thus it is followed by researchers and widely used to retrieve information in plain text.

The contribution of our proposed scheme is to build a Fast and Secure Multi-keyword Search scheme over Encrypted Cloud. Firstly a new Fast and Secure scheme is proposed, after that through experimental evaluation of the proposed scheme is proved on the Enron email datasets. In the section 2 related work is discussed, section 3 gives the detailed system more and security requirements. The comprehensive proposed scheme presents in section 4. Section 5, analyzed the performance of the proposed scheme. The conclusion is given in section 6.

2. RELATED WORD
The literature of searchable encryption is [1]-[9] is a helpful technique for performing the secure single keyword searching over Encrypted data on the cloud. The search cost of the given basic searchable encryption would be very high. Few researchers provide the multi-keyword ranked search. Recent work provides some solution to the secure search over encrypted data, but the problem is they only use for single keyword queries. So it becomes a challenge to support multi keyword search on encrypted data without sacrifices the privacy of the encrypted outsourced data over the cloud.

First time introduced in [10], the concept of privacy preserving ranked multi-keyword search scheme by modifying technique. The sequential and indexed based searching proposed by [1], but limited to small data sets and didn’t yield the required result in large data sets. Also, it becomes an overhead to generate and maintain indexes using various index based searching. The more suitable indexing scheme for large as well as small documents with more accurate results presented here [2]. In [8] a multi keyword search scheme, proposed which is based on fuzzy keyword searching. It supports the searching as well as allow efficient searching by handling minor typing errors.

Most of the research work focuses on multi-keyword ranked searches, support secure and privacy preserving searching. But none of them handle the updating and modification of the outsourced data by the data owner. These problems handled in our proposed model and we also cut the communication cost by utilizing efficient index building technique.

Copyright © 2018 Upinder Kaur et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License 4.0, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.
3. PROBLEM FORMULATION

3.1 System Model

The proposed system model consists of following entities, the cloud service provider, the data owner, and the registered users as shown in Fig. 1. The data owner uses the cloud services to outsource the data to the cloud server. The data owner provides a set of n no of documents \( d_1, d_2, d_3, \ldots, d_n \), then convert it into encrypted form \( E = \{E_1, E_2, E_3, \ldots, E_n\} \) before outsourced it on the cloud. Then data owner builds their own Index \( I = \{I_1, I_2, I_3, \ldots, I_n\} \) from available data sets D where \( I_i \) is generated for \( D_i \) and then outsourced it. The required documents searched when the registered user generates a query \( Q \), having the subset of \( W \) (where \( W \) denotes as the keywords in the dictionary). Then data owner Create the trapdoor entry \( T \) from \( Q \), after collecting the Trapdoor entry \( T \), the cloud server searches \( I \), then returned the result to the registered user.

If the registered user wants to perform the updating in the document, then we extend the keyword and update the index. We introduced "last modified," "status" and "version" to handle the dynamic updates.

![Diagram of System Architecture](image)

**Fig. 1 System Architecture for fast and secure multi-keyword searching over encrypted cloud data**

In our system can be implied by following operations.

- **KeyGen**: This parameter is used by the data owner to generate a key \( K \).
- **BuildIndex**: Now a secure index \( I \) from the dataset \( D \) based on \( K \) build by the data owner and then encrypted the data before outsourced it.
- **TrapdoorGen**: The data owner generated the Trapdoor \( T \). The created Trapdoor \( T \) based on the query by Registered User having the terms from query \( Q \).
- **KeyExtend**: The data owner after any updating, extend the more keywords or reduce the existing keywords in the dictionaries (\( W \)).
- **Query**: The Cloud service provider performed the ranked query search using the Trapdoor \( T \) on build Index \( I \).
- **Update**: Data Owner allows updating the index and querying entries on the cloud server. It also maintains its update key \( P \).

3.2 Privacy Requirements

In the proposed model, assume that the cloud service provider as the untrusted entity. We believe the cloud service provider should not learn any information from the encrypted data, build the Index, the query generated by the data user. So the security requirements for the above scheme are:

- **Data Privacy**: for the data privacy, the data owner allows only the registered users to retrieve data from the cloud service provider, and the only data owner provides the authorization.
- **Index Privacy**: The secure index and the user query should not leak any information about the documents and the terms used in the query.
- **Trapdoor Privacy**: The cloud service provider is not allowed to generate their own trapdoor from the previous trapdoor information.

4. PROPOSED MODEL

The proposed model includes the following steps.

- **KeyGen**: This is initialized phase. Data owner generates a new key called Secret Key from the previously used keywords. It builds n-bit random vector \( S \) and product of two matrices \( M_1, M_2 \). Hence the secret key \( SK \) is a form of three tuples \( SK = \{M_1, M_2, S\} \)
- **BuildIndex**: This generates the Skip Graph Bloom Filter \( B_f \) from the data, set \( D_i \) where \( i=1,2,\ldots,n \).

We use multiple bloom filters and split them \( B_f \) to \( \{B_f', B_f'', \ldots\} \), with the Secret Vector \( S \). Get \( I \) by multiplying \( B_f' \) and \( B_f'' \), with matrices \( M'_1, P_1, M'''_1, P_2, M''''_1 \), and then Upload the Index to the Cloud Service Provider.

- **TrapdoorGen**: generate the bloom filter \( qB_f \) from the data user query \( Q \)

  Split the Query Bloom filter \( qB_f \) to \( \{ qB_f', qB_f'' \} \) with the secret vector \( S \)

  Get \( T \) by multiplying the \( qB_f' \) and \( qB_f'' \) with the matrices and then request the cloud service provider with the Trapdoor \( T \).

- **KeyExtend**: Data Owner allows the registered user to update the dictionary using keyExtend algo. In keyExtend algo., the matrices \( M_1, M_2, M_3, \ldots, M_n \) are used and these matrices will be added to the original matrices \( M_1 \) and \( M_2 \). This used for update operation by joining \( S_1 \) and \( S \) and copy the all the content in \( S' \).

- **Query**: Only the authorized data user allows searching the encrypted outsourced cloud data. When the data user wants to run a query, it will calculate the hash value for each search keyword. We use the user secret key to generate query results from index generated by the data owner. The resultant index is then returned to the cloud service provider to yield results.

- **Update**: The update operation handles the dynamic updating which includes insertion, deletion, and update. Then can add new keywords to the existing documents, deletes a keyword from the existing documents or replacing the keywords in the existing documents.
documents. All these required to regenerate the document index and send it back to the cloud server. We also keep the status of update like version, status, updated. We generate the bloom filter $B_{fj}$ for the new document $D_j$ where $j= n+1,n+2,\ldots,n+t$. Then we encrypt all bloom filter entries from $B_{fj}$ to $D_j$ and update the encrypted index entries to the cloud server.

5. PERFORMANCE EVALUATION

In this section, the performed experiments on the real data sets and synthetic data sets. Synthetic data sets are simulated data sets in the lab and calculated a real life dataset, Enron email dataset [14]. We use Enron email dataset because of various numbers of emails available. We randomly selected different emails from the dataset for each test. The proposed model is tested under simulated environment using CloudSim in Java Platform. The evaluation is based on the consideration of the fast index generation as compared to the existing schemes and fast trapdoor generation and the result of query generation.

5.1 Index Generation

As in the proposed scheme, use skip graph bloom filter, in which skip graph is used for indexing various data sets, and bloom filter are used for faster lookup. It also handles the dynamic update in the query as well as data set index. In contrast to the comparison with existing tree based index scheme, proposed skip graph bloom filter index outperforms well. It takes less time for index generation and lookup than other existing schemes. Fig. 2 shows the comparison between the time taken by our proposed scheme and other tree based index scheme.

5.2 Query Generation

The time calculation for query generation mainly depends upon the no. of keywords in the data set dictionary. This also noticed that the time cost for query search increases with increase in a number of keywords. So we consider the query generation w.r.t to a number of query words in a search, it is shown in Fig. 3. Bloom filter used in the index construction helps in cuts the communication cost as it depends on the length of the input keywords. So compare with existing PPMRSE our proposed scheme can achieve much lower query generation cost.

6. CONCLUSION

In this paper proposed a scheme i.e. skip graph and bloom filter for multi-keyword search on the encrypted cloud data. This paper prefers Java platform to test the proposed scheme. This provides the combination of skip graph and bloom filter for index generation and faster lookup. It also proposes a new scheme for dynamic updating in multi-keyword search scheme. Bloom filter helps in cuts down the time cost in query generation and trapdoor generation and index generation by reducing the length of input keywords. Also, the keyExtend function helps in dynamic updating of documents by data users.

In the future, will builds searching schemes for large databases under intense security threats.
7. REFERENCES


