International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-1, October, 2019

Secure Ranked Multi-Keyword Search based on Modified Blowfish Algorithm and AVL Tree in Untrusted Cloud Environment

Rosy Swami, Prodipto Das

Abstract: Advancement in cloud services growing day by day motivates a huge number of data proprietors to store their valuable data in cloud servers. For privacy issue, datasets that contain sensitive information are encrypted first and then outsourced into the cloud server. In this paper, ranked multi keyword search based on AVL tree, especially for several data owners, has been proposed. To avoid unauthorized access, the proposed work encrypts the data using a Modified Blowfish (MB) algorithm before outsourcing. MB algorithm introduces # operation instead of normal OR operation in conventional blowfish algorithm. This MB algorithm provides robustness against any intruding whereas the conventional blowfish algorithm insecure for many applications. To achieve a proficient search, every data owner's index based on AVL tree is encrypted by way of additive order and the privacy-preserving family is formed. The cloud server is then permitted to combine these indexes effectually without knowing the index content. An Iterative Deepening Depth First Search (IDDFS) procedure is used to discover the matching file for the data user request. Finally, our proposed work provides secure data outsourcing to cloud server compared to the other existing methods. The proposed work considers three metrics for the performance evaluation process, viz. precision, index construction time and search time.

Keywords: AVL tree, Cloud server, Data owner, Data user, Iterative Deepening Depth First Search algorithm, Modified Blowfish algorithm

I. INTRODUCTION

With the improvement of cloud services, data proprietors are getting motivated in outsourcing their data into the cloud server to achieve better access and storage facility at a low cost. Encrypting the data before outsourcing into the cloud is considered as a general approach for protecting data privacy. Even though encryption protects the data against unauthorized access, but at the same time, it also activates inconvenience for the authorized users in accessing the encrypted data at large. As a result, much research is being carried out so as to quickly retrieve the information from the huge pool of data using some keyword-based search techniques. It has become a challenge for the researchers to give an efficient multi-keyword search model. Privacy-preserving conjunctive keyword search system over encrypted cloud data cares the update operations dynamically. The index structure is constructed on the basis of Multi-Attribute Tree (MAT) and an effective search procedure which is known as search MAT algorithm is introduced [1]. In order to enhance the efficiency of the text searching the index structure based on the Hierarchical Agglomerative Clustering tree index (HAC-tree) is proposed. To encrypt the index of HAC tree and query vector, this method uses the secure inner product algorithm. In this, Non-candidate Pruning Depth First Algorithm is used to search the corresponding file in the tree which prunes the sub-tree which does not contain any search result [2]. To increase the relevance of the searched keyword to the cloud file, the coordinate matching along with inner product similarity is introduced. Reverse data structure to permit users to accomplish dynamic operations on document collection is proposed, which perform either inserting or deleting. The sparse matrix is used to encrypt the index matrix and query vector to enhance efficiency [3]. To provide privacy for both cloud service providers as well as data users, the new Oblivious Multiple Keyword Search (OMKS) is proposed. The proposed protocol support multiple keyword searches such as conjunctive keyword search and disjunctive keyword search. In the disjunctive keyword search, it apprehends in a simple way that it sends the values of the keyword to the server in the query. In the conjunctive keyword search, the addition of all keywords values is used as the fresh keywords values involved in the calculation. By using these two searches this method achieves efficient search and matched ciphertext [4]. The multi-keyword tree-based search scheme is proposed to provide security to the sensitive information of the data owners. The document collection in the cloud environment is achieved through the hierarchical K-Means method. To generate an encrypted index as well as query vectors, the vector space model is used and to achieve efficient search, DFS algorithm is used. The secure KNN algorithm is used to encrypt the query vectors. The clustering of documents is performed using bisecting k-means clustering [5]. Context-aware search is introduced to make semantic search smart. The proposed method first introduces the Conceptual Graphs (CG) as a knowledge representation tool. Two schemes are proposed based on CG. This method converts original CG into their corresponding linear form with few modifications and it matches them to numerical vectors. Ranked multi keyword search over encrypted data in the cloud is introduced on the basis of two threat models. To resolve the problem in the privacy-preserving smart semantic search based on CGs, the proposed scheme uses PRSCG and PRSCG-TF schemes [6].

Revised Manuscript Received on October 30, 2019.

* Correspondence Author
Rosy Swami*, Dept. of Computer Science, Assam University, Silchar, India. Email: dam.rosy@gmail.com
Prodipto Das, Dept. of Computer Science, Assam University, Silchar, India. Email: prodiptodas@gmail.com

The compound concept semantic similarity evaluation method is projected to quantify the similarity between the compound concepts. This method integrates both secure K nearest neighbor
scheme and CCSS with Locality Sensitive Hashing Function, thus proposing the Semantic Compound Keyword Search (SCKS). The goal of secure KNN scheme is to steadily recognize the K-Nearest points in the encrypted databank to a provided encrypted query. This proposed method not only achieves semantic-based search but at the same time also performs a multi-keyword search and ranks the searched result [7]. A new semantic search scheme built on the idea of the hierarchical semantic relationship between concepts in the encrypted datasets is proposed. To enhance the effectiveness of the search, the tree based index structure to bring together all the document index structure is introduced. This proposed method uses two numbers of cloud servers, one used for storing the datasets of data owners and returned ranked results and the other used for computing the similarity scores between the document and query and sending the scores to the first server [8].

The major objective of the proposed work is to secure outsourcing of information to the cloud server. The main contributions are highlighted below:

- ➢ Index based on AVL tree for each data owner document is constructed, and MB algorithm is used for encryption before outsourcing it into the cloud server.
- ➢ Each encrypted AVL index tree is merged in the cloud by the server without knowing the sensitive information in the data owner dataset.
- ➢ IDDFS algorithm is used to search the top k ranked result for given data user trapdoors, which improves the searching efficiency.

The rest of this paper is systematized as follows, section II describes the related work, in which existing methods and its drawbacks are presented. In section III the proposed work is discussed. In section IV performance evaluation of the work is illustrated. Finally in section V conclusion and future work is presented.

II. RELATED WORK

To achieve efficient conjunctive query process, [9] presented an unusual tree-based data arrangement called Virtual Binary (VB) tree. This method creates a tree totally kept in a hash table for effective updating and improved scalability. The main objective is to arrange indexing components into the VBT tree in a top-down model, keeping neither any of the tree branches nor any of the tree nodes. In VB Tree, to traverse the children items into a tree item, this method uses the binary track based function to find the children items both left and right. The limitation of this paper, VB Tree tends to become unbalanced when data owners pieces of information are increased. To avoid loss of data usability, [10] proposed the ranked multi-keyword searchable encryption. The proposed method gives back the upper k results as an answer to a data user request. The system does not depend on a predefined dataset and has provisions for keywords in random languages. The system supports multiple users to go through flexible search authorization in addition to time controlled revocation. In the encryption process, data owners can assign different weights to the keywords based on their importance. In the search process, the cloud server computes relevance scores for each keyword. The limitation of this paper, the data user can use only one trapdoor to search multiple owner documents which may reduce the searching efficiency. To secure data outsourcing in the cloud environment [11] proposed an exceptional index structure and traversal algorithm on the basis of tree. The proposed traversal algorithm makes the same query to yield various paths on the index. This method proposes the multi-keyword based top k search system based on the idea of partition in which set of tree-based indexes are created for all documents. This algorithm with the random traversal policy makes cloud server to traverse in a random fashion on the index and returns various outcomes for the same query within the meantime. In this proposed scheme data owner has control over the level of query unlinkability without forfeiting accuracy. The proposed traversal algorithm takes more time to search the document in the cloud server. To enhance integrity verification of search, [12] introduced the conjunctive keyword searchable encryption scheme with an authentication mechanism. The proposed scheme depends on the dynamic searchable encryption scheme and it adapts the Merkel tree. It uses bilinear map accumulator to verify the rightness of set operations. The proposed method supports the conjunctive keyword as input for conjunctive search. This method has formal mathematical proof under oracle model to claim that the proposed method is secure against the adaptive chosen unauthorized access. The proposed Merkel tree is unbalanced when the data owner document increases. To support dynamic update operation, [13] proposed a multi-phrase ranked search over encrypted data. Symmetric searchable encryption permits data user to retrieve keyword over encrypted data without the need for decrypting the data. This method uses the inverse index to keep a record of the location of keywords and to investigate whether the expression appears. To provide security to the relevance score, ranking of the results and computation of relevance score is performed at the user end. The special structure of the index makes the proposed system dynamic in nature. In this method, the data owner has the advantage of updating the cloud data at very minimal cost. The relevance score is computed in the client side, so sensitive information of the data owner can be access by the unauthorized users. In [14] an efficient search result approach is proposed based on the split keyword fuzzy and synonym search. The wildcard technique is used in this process to store the keyword in index safely. The major contributions of this proposed scheme are to support synonym queries. When the user enters the keyword to search data, the user gets the fuzzy matching in addition to files which consists of synonym of some predefined keywords. The synonym based keyword search is authenticated by using an index based tree structure. The limitations of this paper, synonym search takes more time to retrieve the appropriate results. [15] proposed the method which stores the encrypted data in the cloud server and supports the keyword searching through the encrypted database.

In this modified cuckoo filter is used to form the set of membership operations and also used to store some extra information associated with each inserted element in the set. To improve the traceability in a multi-keyword search, [16] proposed the Multi-User Searchable Encryption (MUSE) scheme. In this proposed scheme an asymmetric group key management protocol is used to obtain the group encryption key. In this, each user in the group consisting of their own secret key and each user generate trapdoor without getting any help from the other users. The proposed scheme enables the user to search using a single keyword, and this becomes a limitation of this paper. To overcome the keyword search with access control problem, [17] proposed the Keyword Search with Access Control (KSAC) scheme. The proposed KSAC method utilizes a new cryptographic primitive called Hierarchical Predicate Encryption (HPE) to enforce access control and to perform the multi-field query search. In this paper, the privacy of each data user query is enhanced by adding noise in the query.

III. PROPOSED WORK

A. System Overview

The proposed work enhances the security through MB algorithm encryption of data user document before outsourcing it into the cloud server.

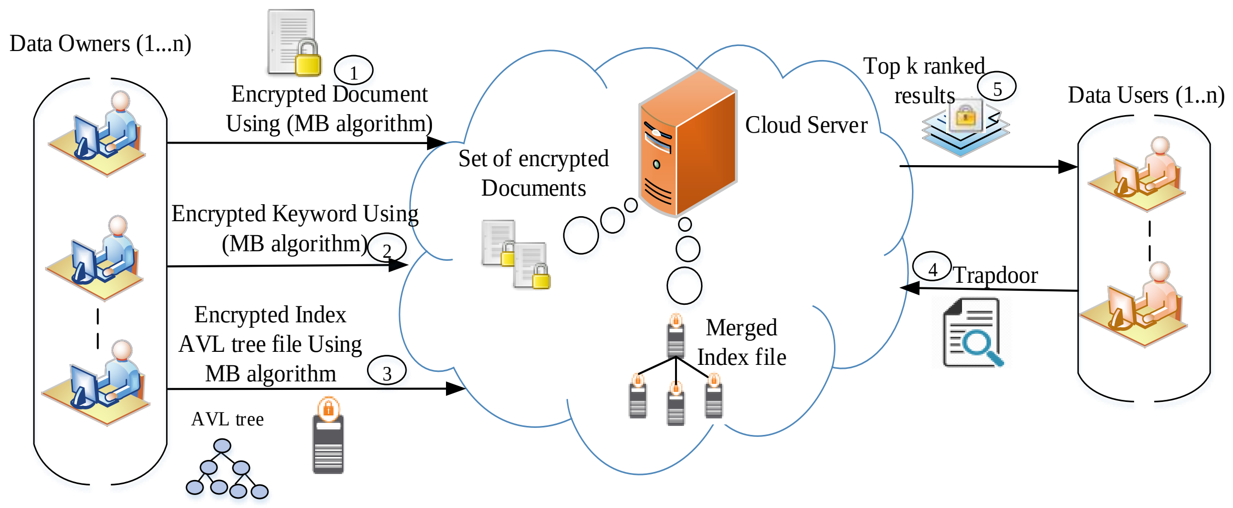
Data Owners (1...n)
1
Encrypted Document Using (MB algorithm)
Encrypted Keyword Using (MB algorithm)
2
Encrypted Index AVL tree file Using MB algorithm
3
AVL tree
Set of encrypted Documents
Cloud Server
Merged Index file
Top k ranked results
5
4
Trapdoor
Data Users (1..n)

Fig. 1. Architecture for the proposed work

The above fig. 1 illustrates the architecture of the proposed work which contains Data owner (1 to n) and cloud server and Data user (1 to n). Data owners form AVL index tree and encrypt it using MB algorithm. Encrypted AVL index tree files are merged in the cloud server. Data users generated trapdoor for searching keyword in the cloud server. The proposed work IDDFS algorithm searches the keyword in cloud server and it returns top k ranked results to the data users. Data owners provide secret key to the authorized data users to decrypt the document.

B. Encryption

In this phase, data owners encrypt the documents using MB algorithm. The Blowfish algorithm being affected by many intruders, MB algorithm is proposed which provides robustness to the intruders. MB algorithm has four states which are used to convert the plain text into cipher text.

Thefig. 2 illustrates the truth table for '#' operation. Three inputs to the # operation should be converted from 32 bits to the 16 bits. In the new MB algorithm, symbol # operation is used instead of XOR operation. In MB algorithm two keys are used instead of one key in each round of the original blowfish algorithm. Three inputs are used in the proposed MB Algorithm. The first input is used to represent the table number which is used to compute the result among the 4 tables given in figure 2. The other two inputs represent the row and column number in the specified table in which cross point of them gives the result. The first key K1 will be used in each round of the original blowfish. The first key K1 is used with the xL and Pi to produce the next left part. And then second key K2 is used with the F(xL) and xR to produce the right part.

#0	0	1	2	3
0	3	2	1	0
1	2	3	0	1
2	1	0	3	2
3	0	1	2	3

#1	0	1	2	3
0	0	1	2	3
1	1	0	3	2
2	2	3	0	1
3	3	2	1	0

#2	0	1	2	3
0	2	3	0	1
1	3	2	1	0
2	0	1	2	3
3	1	0	3	2

#3	0	1	2	3
0	1	0	3	2
1	0	1	2	3
2	3	2	1	0
3	2	3	0	1

Fig. 2. Truth table for '#' operation

For example, binary number (32 bits):

10010111010100101010011110100010 01

Will be converted to the number:

2 1 1 3 1 1 0 2 2 2 1 3 2 2 0 2 1

In this 32 bits are converted to 16 bits for each input in the # operation. After converting, # operation is performed according to the table illustrated in figure 2 producing the encrypted result.

The reverse of this encryption operation provides the decryption result which gives the original document.

C. Building AVL tree Index

To improve the efficiency of the proposed work, the AVL tree is introduced. An AVL tree is defined as a self-balancing binary search tree which is named after inventors Adelson-Velsky and Landis. In AVL tree, balance factor is calculated for every node in the tree. Balance factor for node i is expressed as,

Balance factor (i) = Height (RS (i))-Height (LS (i)) (1)

Where RS represents the Right Sub-tree and LS represents the Left Sub-tree of the node (i). The balance factor for the node has three different types that are given by,

- Balance factor>0 means right sub-tree is heavy
- Balance factor=0 means the tree is balanced
- Balance factor<0 means left sub-tree is heavy

Based on these three types AVL tree balances the node which in turn improves the searching efficiency. In this scheme, all the data owners build their own secure AVL tree and outsource them into the cloud server. Each node in the AVL tree stores the vector V which contains the elements that are relevance score. The index tree is built from the set of document file D = $(f_1,f_2\ldots f_n)$. The node in the index is defined as follows,

$I_{node} = \{id, f_{id}, V, L_{node}, R_{node}\}$ (2)

Where id and f_{id} denote the id of the node and id of the file respectively. L_{node} represents the indicators to the left child of the node. R_{node} represents the indicators to the right child of the node. To build the AVL tree for the entire files data owner creates a node for each data file f as,

$I_{node,i} = \{id_i, f_{id,i}, V_i, L_{node,i}, R_{node,i}\}$ (3)

By considering the above formula, the proposed scheme generates a node for each file in its document.

D. Merging of the Index tree

The cloud server is having index tree set S= $\{S_1, S_2\ldots\ldots\ldots S_n\}$ generated from several data owners. Cloud server combines a set of index tree into one index tree $S_{ubmerged}$. Cloud server merges the set of index tree using two steps that are given below,

- At first, cloud server sets an empty linked-list and then inserts the root R_i which denotes the root of S_i, i ∈ [1, n] to the linked list.
- Cloud server considers this constructed linked list as inputs and gets the root of the merged tree. Using these two steps cloud server merges the given set of index tree into one index tree.

By considering this procedure to merge the set of index tree provides security to the data owners' documents, because, cloud server does not know about the sensitive information in the index file while merging the set of the index file.

E. Relevance Score Generation

Vector space model with TF×IDF model is proposed to generate the relevance score. The proposed vector space model supports information retrieval more. In this model, TF denotes the "Term Frequency" which is defined as the frequency of a given keyword present in the file. IDF denotes the "Inverse Document Frequency" which is defined as the logarithm computation of the total number of files divided by the number of files containing the given keyword. For a given data file set f= $\{f_1,f_2,\ldots,f_n\}$ and keyword set k=$\{k_1,k_2,\ldots,k_n\}$, the relevance score is computed using below expression,

$$RS(f_i,k_i) = \frac{1}{|f_i|}\left(1+\ln f(f_i,k_i)\ln\left(1+\frac{n}{f(k_i)}\right)\right)$$ (4)

Where $RS(f_i,k_i)$ denotes the Relevance Score of a set of file (f_i) and keywords (k_i). $|f_i|$ represents the length of the given file. $f(f_i,k_i)$ represents the count of the keyword k_i in the file f_i. n represents the total number of the file. $f(k_i)$ denotes the number of the file contains the keywordk_i. The vector of the relevance score for keywords in the file f_i is denoted by,

$$V_i = RS(f_1,k_1), RS(f_2,k_2)\ldots\ldots.RS(f_n,k_n)$$ (5)

The vector file of relevance score length is the same for all the files.

F. Encryption of Relevance Score

The proposed work introduces the additive order as well as privacy-preserving function in order to provide access to data owners to encrypt their relevance score by considering different function. This scheme also permits the cloud server to correctly rank the search result files.

$$AOPPF(x) = \sum_{0\le j,k\le\tau} G_{j,k}.h(x,j).h(y,k) + R_a$$ (6)

Using the above expression relevance scores of different data owners files are encrypted. $G_{j,k}$ represents the factor of $h(x,j).h(y,k)$. τ represents the degree of $AOPPF(x)$. The function $h(x,.)$ is used to preserve the order of relevance score x. The function $h(y,.)$ is used for handling the data owners ID y. R_a represents the disturbing factor.

G. Generation of Trapdoor

In the multi-user environment, it is not possible to connect to each data owner to generate the trapdoor. For generating trapdoor, two conditions must be satisfied that are given below,

- Trapdoors generated in data users without communicating with others.
- For same searched keyword different trapdoors will be generated at different times.

By satisfying the above two conditions trapdoor is generated in the data user. We assume data user D_i wants to search keyword k_i for that it computes the trapdoor as,

$$TD_{k_i} = a^{m_{D_i,k} \cdot H(k_i) \cdot l_d}, a^{m_{D_i,k} \cdot l_d} \quad (7)$$

Where l_d is a random number. We assume $TD_1 = a^{m_{D_i,k} \cdot H(k_i) \cdot l_d}$ and $TD_2 = a^{m_{D_i,k} \cdot l_d}$. From which, $TD_{k_i} = (TD_1, TD_2)$ and after generation of trapdoor data user outsource it to the cloud server.

H. Keyword Search

After receiving the search request cloud server translates the trapdoor into a search vector. After the translation, it calls the IDDFS algorithm to search the matching files and return the top-k related file. The proposed IDDFS algorithm is a combination of both Breadth First Search (BFS) algorithm and Depth First Search (DFS). IDDFS algorithm searches the index in AVL index tree based on the pre-order traversal algorithm which is shown in below figure 3.

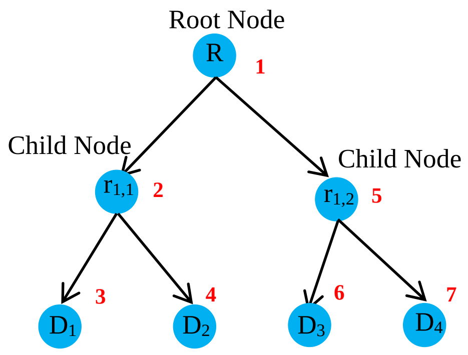

Fig. 3. IDDFS algorithm on AVL tree

Above fig. 3 illustrates the IDDFS algorithm searching on AVL tree. IDDFS algorithm follows the preorder traversal searching method which traverses in the root, left child node and right child node which is shown in above figure 3. Cloud server computes below expression for a given trapdoor TD_{k_i} and each encrypted word in the keyword set $\widehat{k_i}$ which is expressed below,

$$e(EK_1, TD_1) = e(a^{m_{D_i,k} \cdot H(k_i) \cdot l_d}, a^{m_{D_i,k} \cdot l_d}) = e(a,a)^{m_{D_i,k} \cdot H(k_i) \cdot l_d, m_{D_i,k} \cdot l_d} \quad (8)$$

The cloud server judges whether k_i belongs to the keywords set if the following equation is found to be true and set the query vector to 1.

$$e(EK_1, TD_2) = e(a^{m_{D_i,k} \cdot l_d}, a^{m_{D_i,k} \cdot H(k_i) \cdot l_d}) = e(a,a)^{m_{D_i,k} \cdot l_d, m_{D_i,k} \cdot H(k_i) \cdot l_d}, \quad (9)$$

The above two equations are computed in cloud server side after getting the search request trapdoor. It searches the file using IDDFS algorithm. This algorithm searches the file based on the pre-order traversal method and sends the top k ranked result to the data users.

IV. PERFORMANCE EVALUATION

In the performance evaluation section, simulation setup for the proposed work is discussed and comparative analysis with the existing method is described. This section is further divided into two subsections that are namely Simulation Setup and Comparative Analysis.

A. Simulation Setup

To implement the proposed method, Java NetBeans 8.0 software tool is used. During simulation more number of text documents are outsourced to the cloud server by the data owners which are in the form of .doc.

Table- I: Simulation Requirement

Parameters	Values
Number of Documents	700
Number of Owners	70-80
Number of Keywords	20
Number of Users	700-800

The above table I illustrate the parameters that are required in the proposed work implementation.

B. Comparative Analysis

In this section, simulation results obtained from the implementation is compared with the existing method discussed in papers RGMTS [11], IVCKSE [12] and SKFS [14]. Three performance metrics are considered to compare the proposed work with existing work. The comparative analysis considers the following three metrics specifically index construction time, precision and search time to prove the proposed work is better than existing methods.

1) Precision

Precision is defined as the ratio of the number of documents that are retrieved correctly to the total number of documents retrieved in top k ranked results.

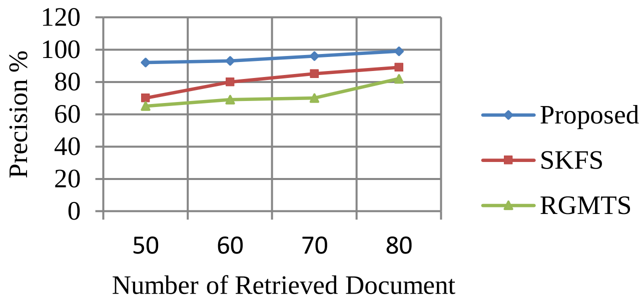

Fig. 4. Comparison on Precision

In fig. 4 precision result of the proposed work is compared with the existing SKFS and RGMTS method. From the simulation results, it can be concluded that the precision percentage of the proposed work is more compared with the SKFS and RGMTS methods. The proposed work achieves the maximum percentage of 99% in precision results which is very high compared to the existing SKFS and RGMTS methods.

2) Index Construction Time

Index construction time is defined as the time taken to complete the construction of the index for data owners' documents.

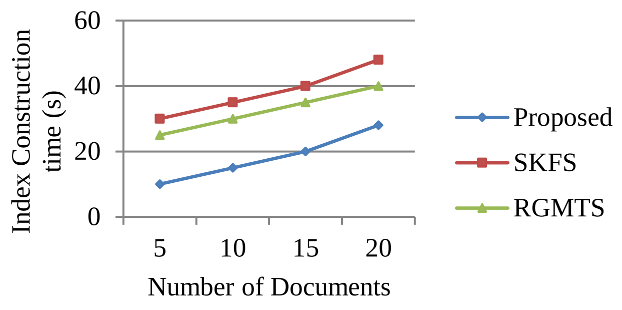

Fig. 5. Comparison on Index Construction Time

In above fig. 5, the index construction time of the proposed work is compared with the existing SKFS and RGMTS methods. Form the simulation result, it can be concluded that the proposed method construct index within a small amount of time whereas the existing methods SKFS and RGMTS require more amount of time to construct the index.

3) Search Time

Search time is defined as the time take to return the top k rank result to the data user. Search time of the proposed work is more efficient than other existing methods such as RGMTS and IVCKSE.

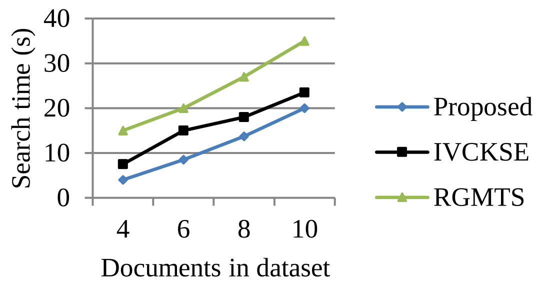

Fig. 6. Comparison on Search Time

The above fig. 6 depicts the comparison on the search time of the proposed work with the existing methods like IVCKSE and RGMTS schemes. From the simulation results of the proposed scheme, it can be concluded that the proposed work takes less time to search the user request keywords in cloud server whereas the existing methods like IVCKSE and RGMTS take more time to search user request in the cloud server. The proposed work requires a maximum of 20sec to search the user request in cloud server whereas IVCKSE method requires 24sec and RGMTS method require 35sec to search the user request in the cloud server. From which it can be proved that the proposed work is more efficient compared to the existing IVCKSE and RGMTS method.

V. CONCLUSION

In this paper, a secure multi keyword search based on AVL tree in untrusted cloud server is proposed. The main contribution of the proposal is to secure multi keyword search and data outsourcing in the untrusted cloud server. In the proposed work, MB algorithm is used to encrypt the data owner document, keyword, and indexes. The proposed MB algorithm provides more robustness against intruders compared to conventional Blowfish algorithm. In the proposed work AVL tree is constructed for data owners' index. The proposed AVL tree is a balanced binary search tree which calculate balance factor for each node in the AVL tree. If the tree is in unbalance condition, the proposed method balances the tree which is achieved through the computation of balance factor for each node in the AVL tree. After the construction of the AVL index tree, it is encrypted through the MB algorithm. After completing the encryption of document, keyword and AVL index tree, data owners outsource it into the cloud server. In cloud server, encrypted AVL index tree is merged to one index tree. While merging the multiple AVL index tree to one, cloud server doesn't know the information of the data owners which provides more security to data owner documents. Data user generates trapdoor for search request in the cloud server. After receiving trapdoor from the data user, the cloud server calls the IDDFS algorithm to search the index in the AVL index tree. IDDFS algorithm searches the index tree based on the preorder traversal method which traverses from the root node, left child node and the right child node of the AVL tree. Based on the relevance score computation, the proposed searching algorithm returns the top k ranked results to the data user. Finally, from simulation results it can be concluded that the proposed work is more effective in search time, precision and index construction time compared to the existing methods such as RGMTS, SKFS, and IVCKSE schemes.

REFERENCES

1. L. Zhang, Y. Zhang and H. Ma, "Privacy-Preserving and Dynamic Multi-Attribute Conjunctive Keyword Search Over Encrypted Cloud Data", IEEE Access, vol. 6, pp. 34214-34225, 2018.
2. Z. Xiangyang, D. Hua, Y. Xun, Y. Geng, and L. Xiao, "MUSE: An Efficient and Accurate Verifiable Privacy-Preserving Multi-keyword Text Search over Encrypted Cloud Data", Security and Communication Networks, vol. 2017, pp. 1-17, 2017.
3. L. Chen, L. Qiu, K-C. Li, W. Shi, and N. Zhang, "DMRS: an efficient dynamic multi-keyword ranked search over encrypted cloud data", Soft Computing, vol. 21(16), pp. 4829–4841, 2017.
4. R. Zhang, R. Xue, L. Liu, and L. Zheng, "Oblivious Multi-Keyword Search for Secure Cloud Storage Service", 2017 IEEE 24th International Conference on Web Services, pp. 269-276, 2017.
5. P. K. Samantaray, N. K. Randhawa, and S. L. Pati, "An Efficient Multi-keyword Text Search Over Outsourced Encrypted Cloud Data with Ranked Results", Computational Intelligence in Data Mining, pp. 31-40, 2018.
6. Z. Fu, F. Huang, K. Ren, J. Weng, and C. Wang, "Privacy-Preserving Smart Semantic Search Based on Conceptual Graphs Over Encrypted Outsourced Data", IEEE Transactions On Information Forensics And Security, vol. 12(8), pp. 1874-1884, 2017.
7. B. Lang, J. Wang, M. Li, and Y. Liu, "Semantic-based Compound Keyword Search over Encrypted Cloud Data", IEEE Transactions On Services Computing, 2018.
8. Z. Fu, L. Xia, X. Sun, A. X. Liu, and G. Xie, "Semantic-aware Searching over Encrypted Data for Cloud Computing", IEEE Transactions on Information Forensics and Security, vol. 13(9), pp. 2359-2371, 2018.
9. Z. Wu, and K. Li, "VBTree: forward secure conjunctive queries over encrypted data for cloud computing", The VLDB Journal, pp. 1–22, 2018.

10. Y. Yang, X. Liu, and R. H. Deng, "Multi-user Multi-Keyword Rank Search over Encrypted Data in Arbitrary Language", IEEE Transactions on Dependable and Secure Computing, 2018.
11. X. Ding, P. Liu, and H. Jin, "Privacy-Preserving Multi-keyword Top-*k* Similarity Search Over Encrypted Data", IEEE Transactions on Dependable and Secure Computing, 2018.
12. Y. Li, F. Zhou, Y. Qin, M. Lin, and Z. Xu, "Integrity-verifiable conjunctive keyword searchable encryption in cloud storage", International Journal of Information Security, vol. 17(5), pp. 549–568, 2018.
13. C. Guo, X. Chen, Y. Jie, Z. Fu, M. Li, and B. Feng, "Dynamic Multi-phrase Ranked Search over Encrypted Data with Symmetric Searchable Encryption", IEEE Transactions On Services Computing, 2018.
14. Raghavendra S, Girish S, Geeta C. M., R. Buyya, Venugopal K. R., S. S. Iyengar, and L. M. Patnaik, "Split keyword fuzzy and synonym search over encrypted cloud data", Multimedia Tools and Applications, vol. 77(8), pp. 10135–10156, 2018.
15. A. V. Vora, and S. Hegde, "Keyword-based private searching on cloud data along with keyword association and dissociation using cuckoo filter", International Journal of Information Security, pp. 1–15, 2018.
16. H. Wang, X. Dong, and Z. Cao, "Secure and efficient encrypted keyword search for multi-user setting in cloud computing", Peer-to-Peer Networking and Applications, pp. 1–11, 2018.
17. Z. Shen, J. Shu, and W. Xue, "Keyword Search with Access Control over Encrypted Cloud Data", IEEE Sensors Journal, vol. 17(3), pp. 858-868, 2017.

AUTHOR PROFILE

Rosy Swami has obtained her Integrated Masters in Computer Science (MCS) from Assam University, Silchar in 2005. Currently she is pursuing her Ph. D. with the Department of Computer Science, Assam University, Silchar. Her research interests are in the fields of Networking, Big Data Analytics and Algorithm and Model design in Cloud Computing.

Prodipto Das is an Assistant Professor in the Department of Computer Science, Assam University, Silchar. He has obtained Integrated Masters in Computer Science (MCS) from Assam University in 2003 as well as Ph.D. in Computer Science from the same University in 2011 respectively. He has completed his post doc research from Plymouth University, UK in 2012. He has teaching and research experience in the areas of Networks and Communication and in Image Processing. He has published several research papers in reputed international journals and conference proceedings. He has chaired a session of ICAEE 2011 in Dhaka, Bangladesh. He attended International Conferences in many places in India and abroad. He has written one book on Computer Graphics from New Delhi Publishers. He has also published an edited volume in Networks from Narosa Publications. He is a member of IEEE, ACM and senior member of IACSIT, China.