PRAGMATIC REVIVAL OF DOCUMENTS IN CLOUD COMPUTING UTILIZING ATTRIBUTE-BASED ENCRYPTION

[1]V. Swathi, [2]A. Sri Harsha, [3]J. Sri Sai Jyothsna, [4]N. Vinay, [5]S. Sai Dutt
[1]Assistant Professor, [2]Bachelor of Technology, [3]Bachelor of Technology, [4]Bachelor of Technology, [5]Bachelor of Technology
[1]Computer Science Engineering
[1]Dhanekula Institute of Engineering & Technology, Vijayawada, AP, India

Abstract: Retrieving documents from cloud in an efficient and secure manner is one of the important issues in cloud computing. There are many searchable encryption schemes already proposed, few of them support efficient retrieval over the documents which are encrypted based on their attributes. Previously single keyword Boolean search schemes, single keyword ranked search schemes and multi-keyword Boolean search schemes are proposed to retrieve the documents from cloud. Now, the idea is to enhance this process by considering a hierarchical attribute-based encryption scheme designed for document collection. Compared with the ciphertext-policy attribute-based encryption (CP-ABE) schemes, both the ciphertext storage space and time costs of encryption/decryption are saved. Here a structure named attribute-based retrieval features (ARF) tree is constructed for the document collection based on document's attributes. An index is used for this ARF tree to perform searching and retrieving operations. A depth-first search algorithm for the ARF tree is designed to improve the search efficiency which can be further improved by parallel computing. This scheme can also be applied to other datasets by modifying the ARF tree slightly. A series of experiments are performed to illustrate the security and efficiency of the proposed scheme.

Index Terms: CP-ABE, ARF, parallel computing

INTRODUCTION

An intuitive approach is encrypting the documents first and then out sourcing the encrypted documents to the cloud. A large number of searchable document encryption schemes have been proposed in the literatures. However, all these schemes cannot support effective, flexible and efficient document search because of their simple functionalities. Privacy-preserving multi-keyword ranked document search schemes are more promising and practical. However, all the documents in these schemes are organized by a coarse-grained access control mechanism, i.e., each authorized data user can access all the encrypted documents. As an example, the whole IEEE Explore Digital Library can be accessed by all the authorized organizations (e.g., the universities) at present and this cannot satisfy the data owners and users in the future. In this paper, a new situation is considered. A data user may want to access part of the library (e.g., computers and data related papers) and intuitively she wants to pay less money compared with the data users who want to access the whole library. In other words, in the document collection, each document can be accessed only by a set of specific data users. In this case, we need to design a fine-grained access control mechanism for the documents, and it is more reasonable compared with the present method.

A practical hierarchical attribute-based document collection encryption scheme is proposed in which the documents are organized and controlled based on attributes. The proposed scheme can greatly decrease the storage and computing burdens. We map the documents to vectors in which both the keywords and associated attributes are considered. The ARF tree is proposed to organize the document vectors and support time-efficient document retrieval. In addition, a depth-first search algorithm is designed.

Existing System:

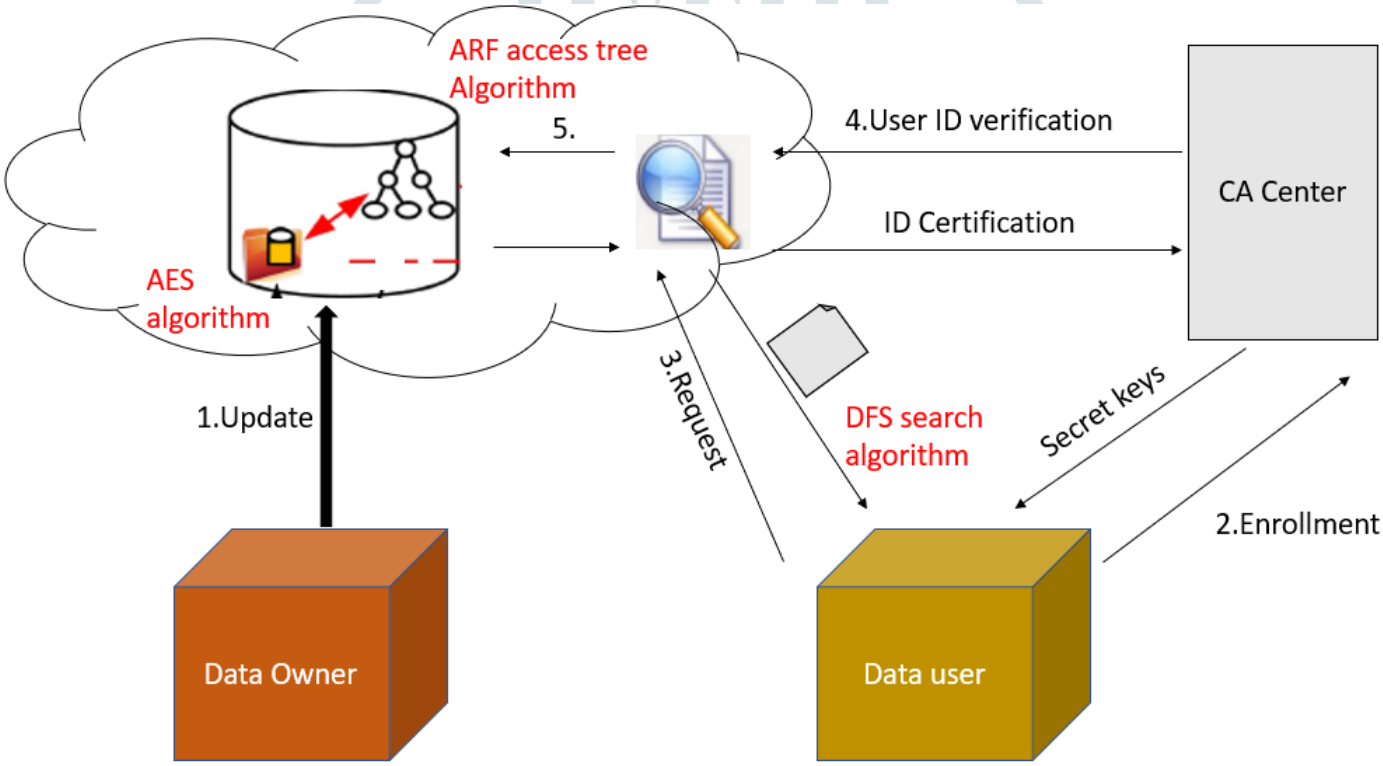
- A large number of searchable document encryption schemes have been proposed in the literatures, including single keyword Boolean search schemes, single keyword ranked search schemes and multi-keyword Boolean search schemes.

- However, all these schemes cannot support effective, flexible and efficient document search because of their simple functionalities.

Proposed System:

- A practical hierarchical attribute-based document collection encryption scheme is proposed in which the documents are organized and controlled based on attributes. The proposed scheme can greatly decrease the storage and computing burdens.
- Here we use DFS algorithm to retrieve the documents from cloud which is efficient than the single keyword searching and multiple keyword searching techniques.
- Specifically, the proposed encryption scheme will perform well in both time and storage efficiency.
- The search proportion of ARF tree is about 5% to 10% efficient to that of non-hierarchical attribute based mechanism.

ARCHITECTURE DIAGRAM:



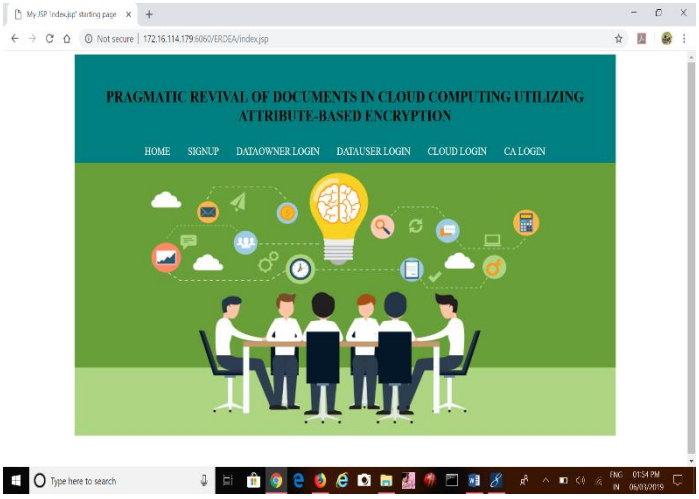
ALGORITHMS

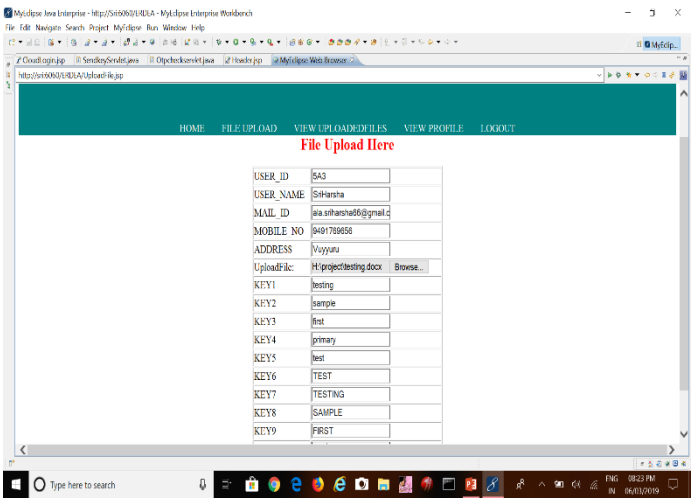
Algorithm 1:- Building Access Structure.
Input: Document collection $F = \{F1; F2; \cdots; FN\}$ with attribute sets $\{att(F1); att(F2); \cdots; att(FN)\}$
Output: A set of access trees ST
1: Sort the files in F in descending order based on the number of their attributes and obtain $F' = \{F'1; F'2 \cdots; F'N\}$;
2: $ST = null$;
3: for $i = 1 : N$ do
4: $S = att(Fi)$;
5: Scan the access trees in order;
6: for the scanned access tree T in ST do
7: if node Y in T matches S, i.e., $TY(S) = 0$ then
8: Insert the identifier of $F'i$ into node Y;
9: break;

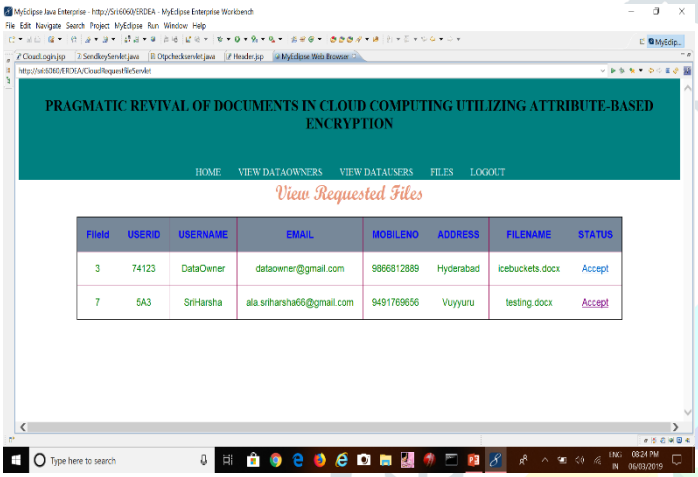
10: else if node *X* in *T* covers *S*, i.e., *TX*(*S*) = 1 then
11: Build a new node *Z* and let the created node *Z* be the child of *X*, and further the leaf nodes associated with *S* are inserted to *Z*; meanwhile, the leaf nodes are deleted from *X*;
12: Insert the identifier of *F'i* into the new node *Z*;
13: break;
14: end if
15: end for
16: if the identifier of *F'i* has not been inserted into an access tree then
17: Build a new access tree for *F'i* based on its attributes and insert the identifier of *F'i* to the root node;
18: Insert the tree to *ST* ;
19: end if
20: end for node *X*

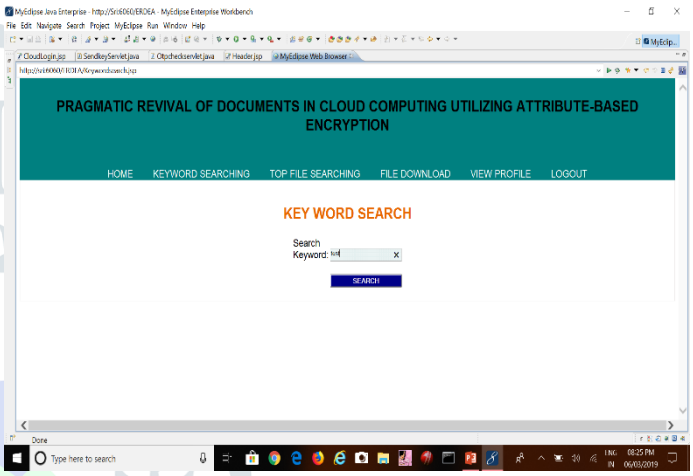
Algorithm 2:- Depth First Search.
Input: an ARF tree with root *r*, a query vector *VQ*, an attribute vector *V'Q* of the data user
Output: The most *k* relevant legal document vectors
1: $u \leftarrow r$;
2: while *u* is not a leaf node do
3: for all the child nodes *v* of node *u* do
4: Calculate the relevance scores between *v* with *VQ* by *RScore*(*v; VQ*);
5: Check whether the attribute set *Av,min* is covered by *V'Q* by comparing *Length*(*Av,min* $\wedge$ *V'Q*) and *Length*(*Av,min*);
6: $u \leftarrow$ the most relevant child node whose attributes are covered by *V'Q*;
7: end for
8: end while
9: Select the most relevant *k* document vectors in the leaf node *u* whose attributes are covered by *V'Q* and construct *RList*;
10: *Stack*.push(*r*);
11: while *Stack* is not empty do
12: $u \leftarrow$ *Stack*.pop();
13: if the node *u* is not a leaf node then
14: if *RScore*(*Vu,max; VQ*) > *kth Score* and *Length*(*Au,min* $\wedge$ *V'Q*) = *Length*(*Au,min*) then
15: Sort the child nodes of *u* in ascent order based on the relevant scores with *VQ* whose attribute sets are covered by *V'Q*;
16: Push the children of *u* into *Stack* in order, i.e., the most relevant child is latest inserted into *Stack*;
17: end if
18: else
19: Calculate the relevance scores between the document vectors in the leaf node with *VQ* and compare their attributes with *V'Q*;
20: Update *RList*;
21: end if
22: end while
23: return *RList*

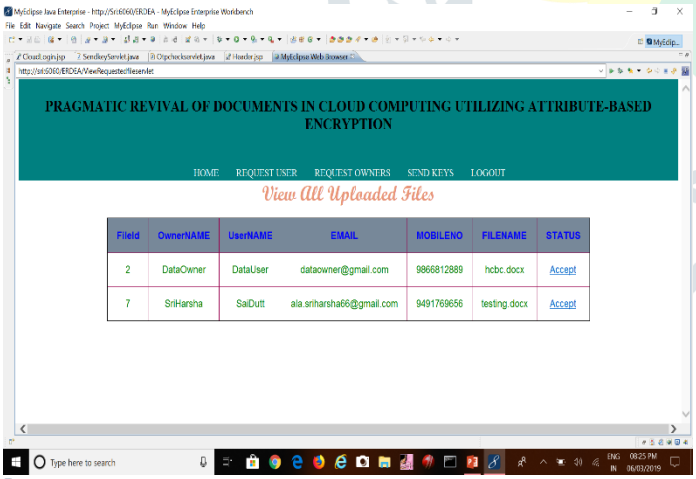
OUTPUTS:

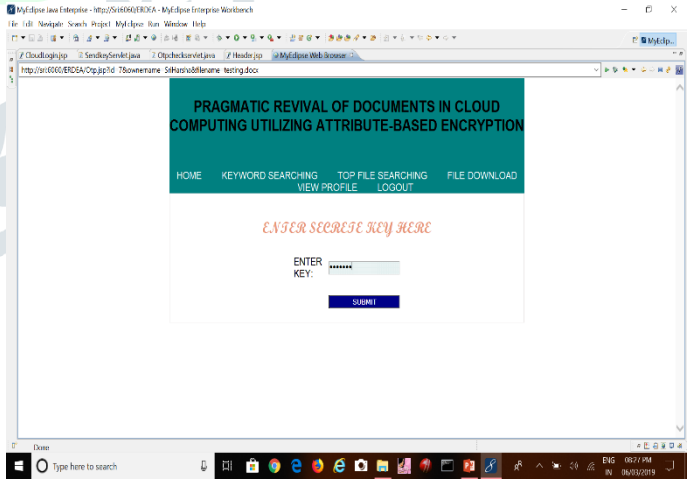


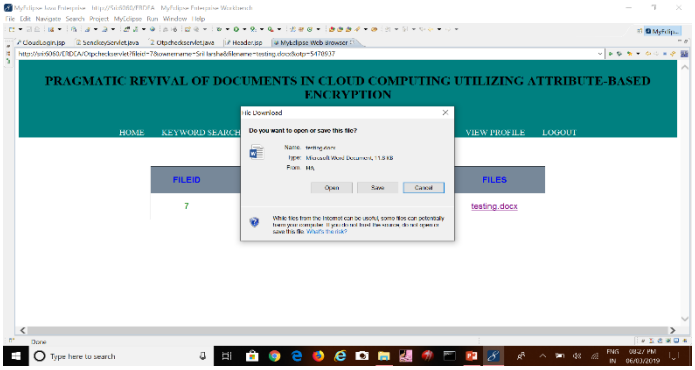

HOME FILE UPLOAD VIEW UPLOADEDFILES VIEW PROFILE LOGOUT
File Upload Here
USER_ID 5A3
USER_NAME SriHarsha
MAIL_ID ala.sriharsha66@gmail.c
MOBILE_NO 9491769656
ADDRESS Vuyyuru
UploadFile: H:\project\testing.docx Browse...
KEY1 testing
KEY2 sample
KEY3 first
KEY4 primary
KEY5 test
KEY6 TEST
KEY7 TESTING
KEY8 SAMPLE
KEY9 FIRST


PRAGMATIC REVIVAL OF DOCUMENTS IN CLOUD COMPUTING UTILIZING ATTRIBUTE-BASED ENCRYPTION
HOME VIEW DATAOWNERS VIEW DATAUSERS FILES LOGOUT
View Requested Files
FileId USERID USERNAME EMAIL MOBILENO ADDRESS FILENAME STATUS
3 74123 DataOwner dataowner@gmail.com 9866812889 Hyderabad icebuckets.docx Accept
7 5A3 SriHarsha ala.sriharsha66@gmail.com 9491769656 Vuyyuru testing.docx Accept


PRAGMATIC REVIVAL OF DOCUMENTS IN CLOUD COMPUTING UTILIZING ATTRIBUTE-BASED ENCRYPTION
HOME KEYWORD SEARCHING TOP FILE SEARCHING FILE DOWNLOAD VIEW PROFILE LOGOUT
KEY WORD SEARCH
Search Keyword:
SEARCH


PRAGMATIC REVIVAL OF DOCUMENTS IN CLOUD COMPUTING UTILIZING ATTRIBUTE-BASED ENCRYPTION
HOME REQUEST USER REQUEST OWNERS SEND KEYS LOGOUT
View All Uploaded Files
FileId OwnerNAME UserNAME EMAIL MOBILENO FILENAME STATUS
2 DataOwner DataUser dataowner@gmail.com 9866812889 hcbc.docx Accept
7 SriHarsha SaiDutt ala.sriharsha66@gmail.com 9491769656 testing.docx Accept


PRAGMATIC REVIVAL OF DOCUMENTS IN CLOUD COMPUTING UTILIZING ATTRIBUTE-BASED ENCRYPTION
HOME KEYWORD SEARCHING TOP FILE SEARCHING FILE DOWNLOAD VIEW PROFILE LOGOUT
ENTER SECRETE KEY HERE
ENTER KEY:
SUBMIT


PRAGMATIC REVIVAL OF DOCUMENTS IN CLOUD COMPUTING UTILIZING ATTRIBUTE-BASED ENCRYPTION
File Download
Do you want to open or save this file?
Open Save Cancel
FILEID FILES
7 testing.docx

CONCLUSION:

WORK DONE:

we consider a new encrypted document retrieval scenario in which the data owner wants to control the documents in fine-grained level. To support this service, we first design a novel hierarchical attribute-based document encryption scheme to encrypt a set of documents together that share an integrated access structure. Further, the ARF tree is proposed to organize the document vectors based on their similarities. At last, a depth-first search algorithm is designed to improve the search efficiency for the data users which is extremely important for large document collections. The performance of the approach is thoroughly evaluated by both theoretical analysis and experiments.

REFERENCES:

1. K. Ren, C. Wang, and Q. Wang, "Security challenges for the public cloud," IEEE Internet Computing, vol. 16, pp. 69–73, Jan. 2012.Hogpracha, W., & Vongpradhip, S. (2015, July). Recognition system for QR code on moving car. In Computer Science & Education (ICCSE), 2015 10th International Conference on (pp. 14-18). IEEE.
2. Y. Miao, J. Ma, X. Liu, Q. Jiang, J. Zhang, L. Shen, and Z. Liu, "Vcksm: Verifiable conjunctive keyword search over mobile e-health cloud in shared multi-owner settings," Pervasive and Mobile Computing, vol. 40, pp. 205–219, 2017.
3. C. Chen, X. Zhu, P. Shen, J. Hu, S. Guo, Z. Tari, and A. Zomaya, "An efficient privacy-preserving ranked keyword search method," IEEE Transactions on Parallel and Distributed Systems, vol. 27, pp. 951–963, Apr. 2016.
4. Java Complete Reference by Herbert Shield.
5. Database Programming with JDBC and Java by George Reese.
6. Java and XML by Brett McLaughlin.