ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

A Modified Hierarchical Attribute-Based Encryption Access Control Method for Mobile Cloud Computing

B.Vineela
Department Of Computer Science & Engineering,
Avanthi Institute of Engineering & Technology,
Cherukupalli, Vizianagaram, A.P – 531162, India.

Mr.S.Keshava Rao
Department Of Computer Science & Engineering,
Avanthi Institute of Engineering & Technology,
Cherukupalli, Vizianagaram, A.P – 531162, India.

ABSTRACT:

Cloud computing is an Internet-based computing pattern through which shared resources are provided to devices on demand. Its an emerging but promising paradigm to integrating mobile devices into cloud computing, and the integration performs in the cloud based hierarchical multi-user data-shared environment. With integrating into cloud computing, security issues such as data confidentiality and user authority may arise in the mobile cloud computing system, and it is concerned as the main constraints to the developments of mobile cloud computing. In order to provide safe and secure operation, a hierarchical access control method using modified hierarchical attribute-based encryption (M-HABE) and a modified three-layer structure is proposed in this paper. In a specific mobile cloud computing model, enormous data which may be from all kinds of mobile devices, such as smart phones, functioned phones and PDAs and so on can be controlled and monitored by the system, and the data can be sensitive to unauthorized third party and constraint to legal users as well. The novel scheme mainly focuses on the data processing, storing and accessing, which is designed to ensure the users with legal authorities to get corresponding classified data and to restrict illegal users and unauthorized legal users get access to the data, which makes it extremely suitable for the mobile cloud computing paradigms.

1. INTRODUCTION:

CLOUD computing extends the existing capabilities of Information Technology (IT) since cloud adaptively provides storage and processing services such as SaaS, IaaS, and PaaS that dynamically increase the capacity and add capabilities without investing in new infrastructure or licensing new software. However, the data access control (DAC) issue of cloud computing systems has been escalated by the surge in attacks such as collusion, wiretapping and distort, so that DAC must be designed with sufficient resistance [2]. DAC issues are mainly related to the security policies provided to the users accessing the uploaded data, and the techniques of DAC must specify their own defined security access policies and the further support of policy updates, based on which each valid user can have access to some particular sets of data whereas invalid users are unauthorized to access the data. One approach to alleviate attacks is to store the outsourcing data in encrypted form [1]. However, due to the normally semi trusted cloud and its arrangement issues of administration rights, cloud-based access control approaches with traditional encryption are no longer applicable to cloud storage systems.

1. In this paper, two attacks are firstly constructed on the vulnerabilities of revocation security in DAC-MACS and EDAC-MACS. By the first attack, the revoked user can eavesdrop to obtain other users’ Key Update Keys to update its Secret Keys, and then it can obtain proper Token to decrypt any se- cret information as a no revoked user as before. In addition, by the second attack, the revoked user can intercept the Cipher text Update Key to retrieve its ability to decrypt any secret information as a no revoked user as before.

Cite this article as: B.Vineela & Mr S.Keshava Rao, "A Modified Hierarchical Attribute-Based Encryption Access Control Method for Mobile Cloud Computing", International Journal & Magazine of Engineering, Technology, Management and Research, Volume 5, Issue 9, 2018, Page 81-91.

2. Secondly, we propose a new extensive DAC-MACS scheme, denoted as the NEDAC-MACS, to withstand above two attacks and support more secure attribute revocation. We modify some DAC-MACS's algorithms, and perform the vital cipher- text update communication between cloud server and AAs with some more secure algorithms [3]. Our NEDAC-MACS scheme mainly includes two improvements on the DAC-MACS at Secret Key Generation phase and Attribute Revocation phase, and it can run correctly according to the correctness proof of NEDAC-MACS.

3. Then, formal cryptanalysis of the NEDAC-MACS is described to prove that the proposed NEDAC-MACS can guarantee collusion resistance, secure attribute revocation, data confidentiality, and provable security against static corruption of authorities based on the random oracle model.

4. Finally, performance analysis of our NEDAC-MACS are conducted by making an efficiency comparison among related CP-ABE schemes to testify that the NEDAC-MACS is security-enhanced without reducing more efficiency. The major overhead of decryption is also securely outsourced to the cloud servers, and the overall overheads of storage, communication and computation of the NEDAC-MACS are superior to that of DACC and relatively same as that of DAC-MACS.

RELATED WORK:

Data Access Control: A plurality of data access control systems based on the promising CP-ABE technique are proposed to construct the efficient, secure, fine grained and revocable access schemes. S.Ruj et al. (2011) proposed a distributed access control scheme in clouds (DACC) that supported attribute revocation. In DACC, one or more key distribution centers (KDCs) distributed keys to data owners and users. Technically, it requires not only forward security but more indispensable backward security in context of the attribute revocation [5]. However, DACC supported attribute revocation with vulnerable forward security. J.Hur et al. (2011) proposed an attribute-based DAC scheme with efficient revocation in cloud storage systems, whereas it was designed only for the cloud systems with single trusted authority. In addition, the above two schemes both require data owners to reencrypt the out- sourced ciphertext after revocation.

2. LITERATURE SURVEY:

CLOUD computing extends the existing capabilities of Information Technology (IT) since cloud adaptively provides storage and processing services such as SaaS, IaaS, and PaaS that dynamically increase the capacity and add capabilities without investing in new infrastructure or licensing new software. However, the data access control (DAC) issue of cloud computing systems has been escalated by the surge in attacks such as collusion, wiretapping and distort, so that DAC must be designed with sufficient resistance. DAC issues are mainly related to the security policies provided to the users accessing the uploaded data, and the techniques of DAC must specify their own defined security access policies and the further support of policy updates, based on which each valid user can have access to some particular sets of data whereas invalid users are unauthorized to access the data [6]. One approach to alleviate attacks is to store the outsourcing data in encrypted form. However, due to the normally semi trusted cloud and its arrangement issues of administration rights, cloud-based access control approaches with traditional encryption are no longer applicable to cloud storage systems.A plurality of data access control systems based on the promising CP-ABE technique are proposed to construct the efficient, secure, fine grained and revocable access schemes. S.Ruj et al. (2011) proposed a distributed access control scheme in clouds (DACC) that supported attribute revocation. In DACC, one or more key distribution centers (KDCs) distributed keys to data owners and users [7].

3. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM:

Senders encrypt message with certain attributes of the authorized receivers. The ABE based access control method uses several tags to mark the attributes that a specific authorized user needs to possess. The users with certain tag sets can get access to the specific encrypted data and decrypt it. Lots of paper introduced the scheme about the attribute based encryption access control method in the cloud computing. In the mobile cloud computing environment, there are tremendous data which needs to be processed and marked with attributions for the convenient attributing access before storing. At the same time, the hierarchical structure of the application users need an authentication center entity to control their attributes.

3.2 PROPOSED SYSTEM:

In the proposed scenario, users with different privilege levels have different rights to access the part of sensing data coming from the mobile devices. Therefore, one same data has to be encrypted into ciphertext once, which ought to be able to be decrypted multiple times by different authorized users. In this paper, a hierarchical access control method using a modified hierarchical attribute-based encryption (M-HABE) and a modified three-layer structure is proposed. Differing from the existing paradigms such as the HABE algorithm and the original three-layer structure, the novel scheme mainly focuses on the data processing, storing and accessing, which is designed to ensure the application users with legal access authorities to get corresponding sensing data and to restrict illegal users and unauthorized legal users get access to the data, the proposed promising paradigm makes it extremely suitable for the mobile cloud computing based paradigm. What should be emphasized is that the most important highlight of all in the proposed paper can be described as that the modified three-layer structure is designed for solving the security issues illustrated above.

ADVANTAGES OF PROPOSED SYSTEM

- One ciphertext can be decrypted by several keys.
- Both precise level description and user attribute should be supported in the access structure of the method.
- The keys in the authentication center ought to have the same hierarchical structure just as the structure of users privilege levels.

SYSTEM REQUIREMENTS

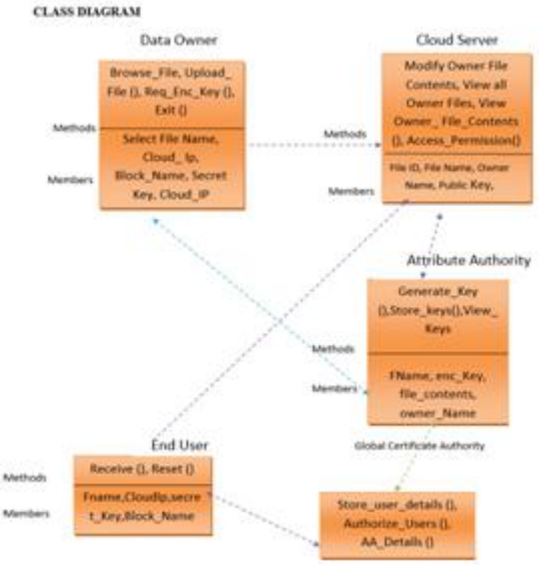
HARDWARE REQUIREMENTS:

- System: Pentium Dual Core.
- Hard Disk: 120 GB.
- Monitor: 15’’ LED
- Input Devices: Keyboard, Mouse
- Ram : 1GB.

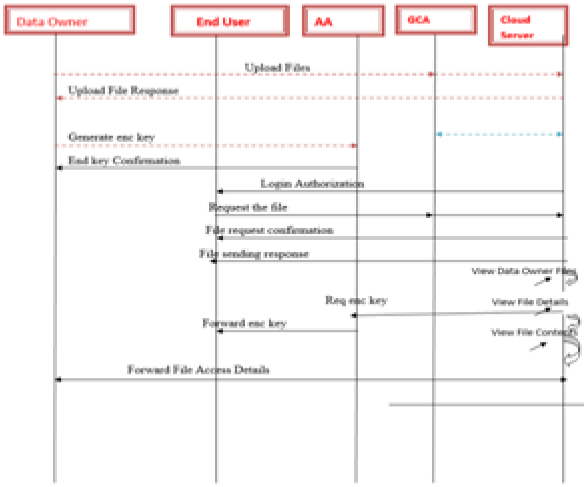
SOFTWARE REQUIREMENTS:

- Operating system: Windows 7.
- Coding Language: JAVA- JSP,JAVASCRIPT
- Tool: Eclipse -- Galileo
- Database: MYSQL

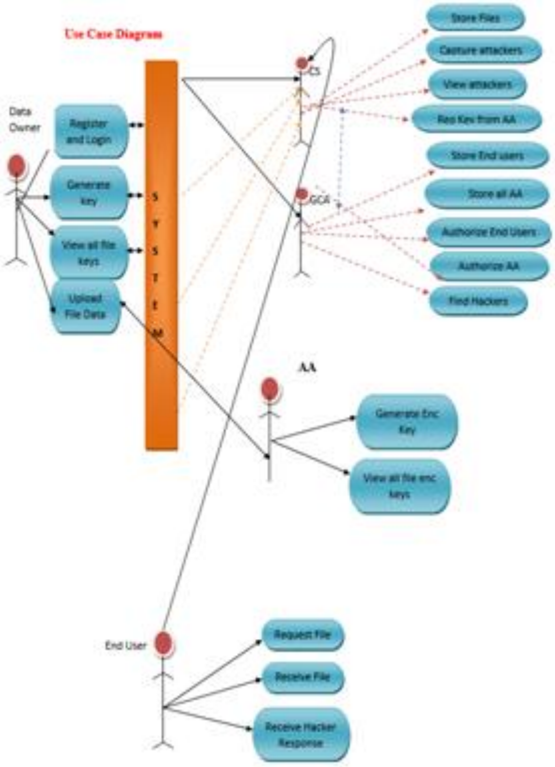
CLASS DIAGRAM


CLASS DIAGRAM
Data Owner
Browse_File, Upload_File (), Req_Enc_Key (), Exit ()
Select File Name, Cloud_Ip, Block_Name, Secret Key, Cloud_IP
Methods
Members
Cloud Server
Modify Owner File Contents, View all Owner Files, View Owner_File_Contents (), Access_Permission()
File ID, File Name, Owner Name, Public Key,
Methods
Members
Attribute Authority
Generate_Key (),Store_keys(),View_Keys
FName, enc_Key, file_contents, owner_Name
Methods
Members
End User
Receive (), Reset ()
Fname,CloudIp,secret_Key,Block_Name
Methods
Members
Global Certificate Authority
Store_user_details (), Authorize_Users (), AA_Details ()

SEQUENCE DIAGRAM

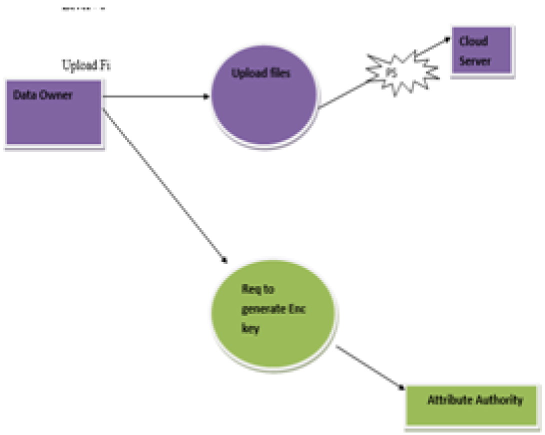


USE CASE DIAGRAM

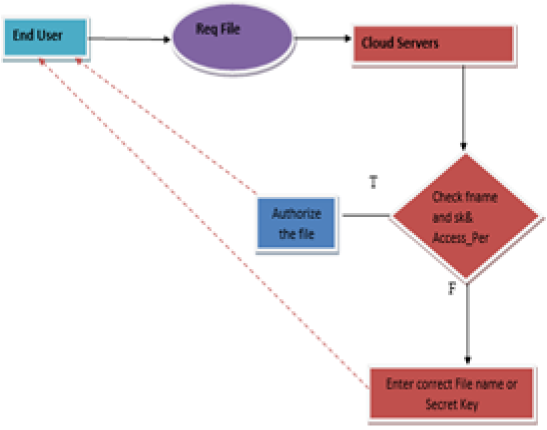


DATA FLOW DIAGRAM

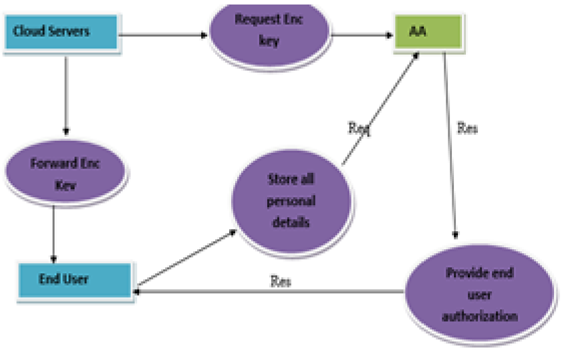
Level-0



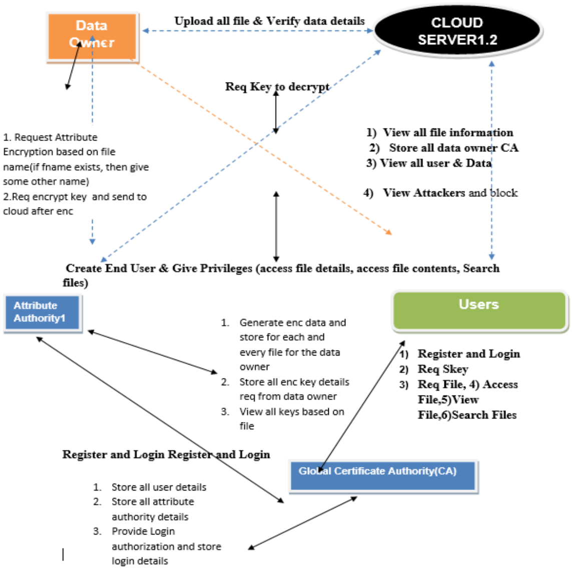
Level -1


End User
Req File
Cloud Servers
T
Authorize the file
Check fname and sk& Access_Per
F
Enter correct File name or Secret Key

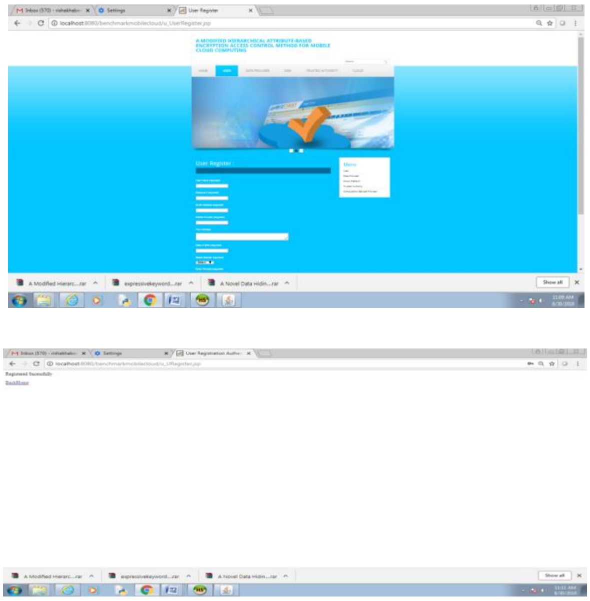
Level -2



ARCHITECTURE DIAGRAM


Data Owner
Upload all file & Verify data details
CLOUD SERVER1.2
Req Key to decrypt
1. Request Attribute Encryption based on file name(if fname exists, then give some other name)
2.Req encrypt key and send to cloud after enc
1) View all file information
2) Store all data owner CA
3) View all user & Data
4) View Attackers and block
Create End User & Give Privileges (access file details, access file contents, Search files)
Attribute Authority1
Users
1. Generate enc data and store for each and every file for the data owner
2. Store all enc key details req from data owner
3. View all keys based on file
1) Register and Login
2) Req Skey
3) Req File, 4) Access File,5)View File,6)Search Files
Register and Login Register and Login
Global Certificate Authority(CA)
1. Store all user details
2. Store all attribute authority details
3. Provide Login authorization and store login details

Registration form



Home page

A MODIFIED HIERARCHICAL ATTRIBUTE-BASED ENCRYPTION ACCESS CONTROL METHOD FOR MOBILE CLOUD COMPUTING
Cloud networking
Data base
Introduction

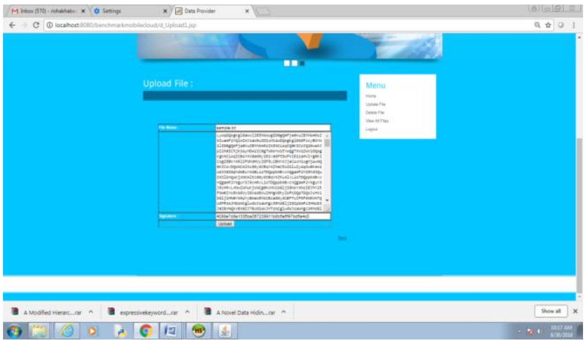
Data provider login

Home screen of data provider

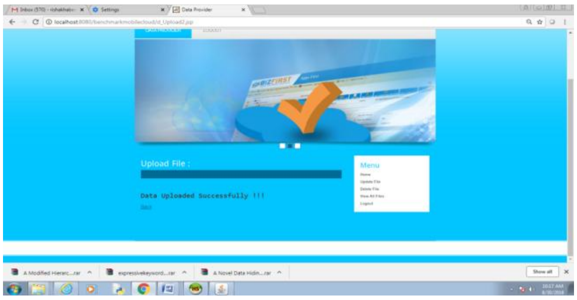
User login

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

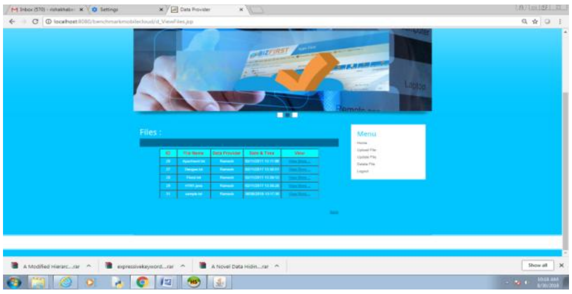
Attribute based encrption



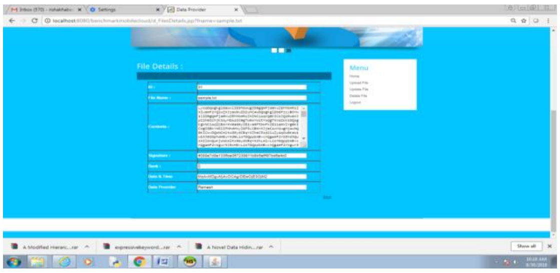
File uploaded successfully



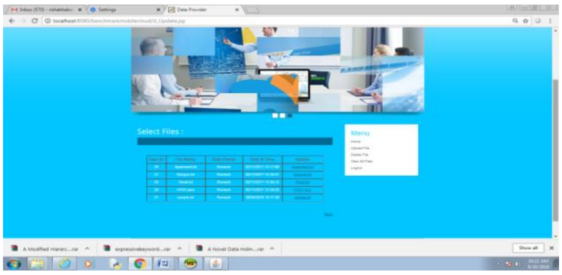
Delete files

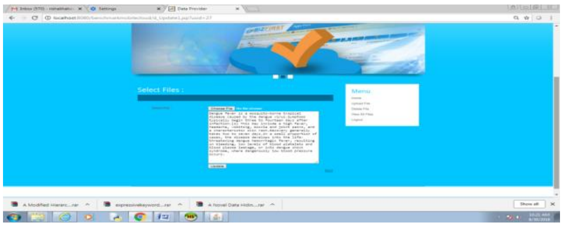


List of files uploaded with key

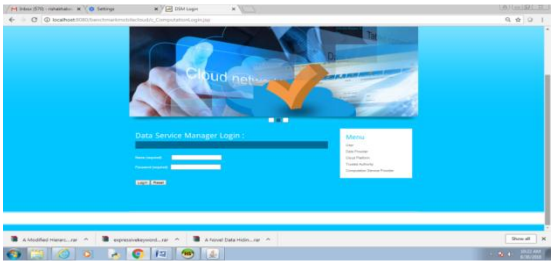


Editing file by data owner

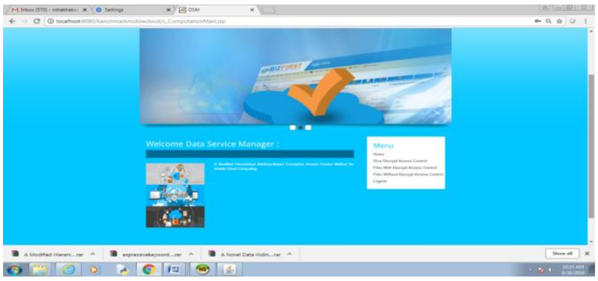




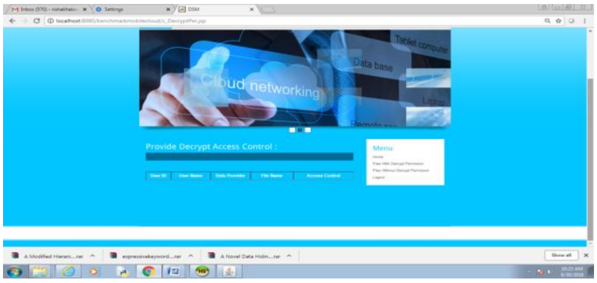
Data service manager



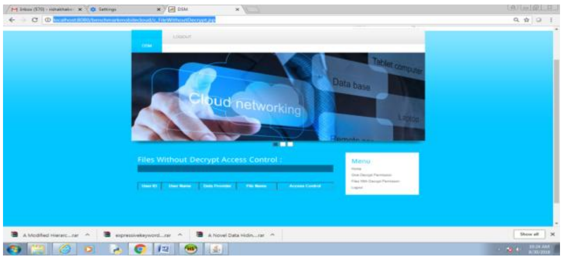
Data service manager home page



Granding decrypt acess control



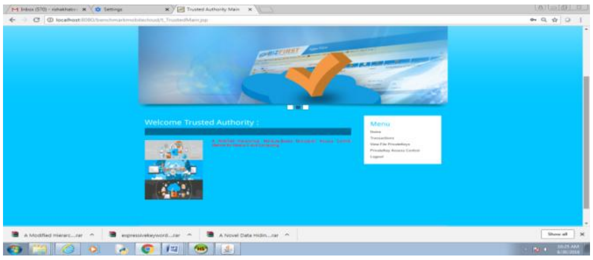
Files with decrypt permission



Files Without Decrypt Access Control :

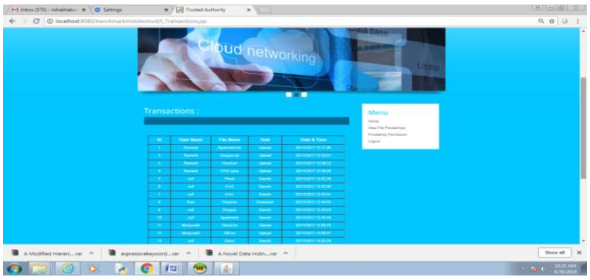
User ID	User Name	Data Provider	File Name	Access Control

Trusted authority home page



Welcome Trusted Authority :

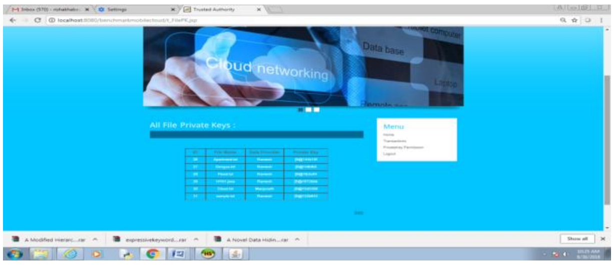
Transactions



Transactions :

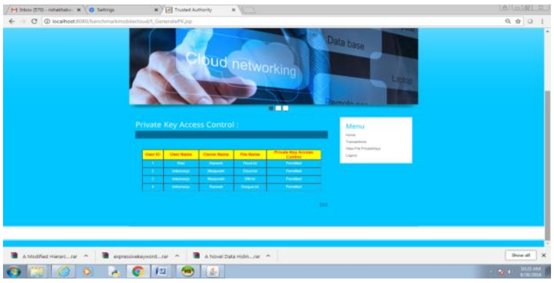
ID	User Name	File Name	Task	Date & Time
1	[illegible]	[illegible]	Upload	[illegible]
2	[illegible]	[illegible]	Upload	[illegible]
3	[illegible]	[illegible]	Upload	[illegible]
4	[illegible]	[illegible]	Upload	[illegible]
5	[illegible]	[illegible]	Search	[illegible]
6	[illegible]	[illegible]	Search	[illegible]
7	[illegible]	[illegible]	Search	[illegible]
8	[illegible]	[illegible]	Download	[illegible]
9	[illegible]	[illegible]	Search	[illegible]
10	[illegible]	[illegible]	Search	[illegible]
11	[illegible]	[illegible]	Upload	[illegible]
12	[illegible]	[illegible]	Upload	[illegible]
13	[illegible]	[illegible]	Search	[illegible]

File and private keys



All File Private Keys :

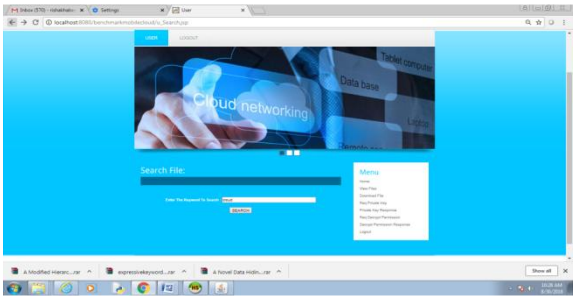
Private Key permissions



Private Key Access Control :

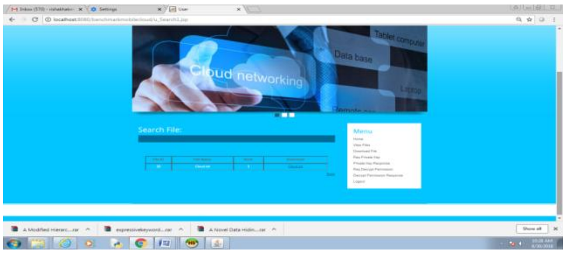
User ID	User Name	Owner Name	File Name	Private Key Access Control
1	[illegible]	[illegible]	[illegible]	[illegible]
2	[illegible]	[illegible]	[illegible]	[illegible]
3	[illegible]	[illegible]	[illegible]	[illegible]
4	[illegible]	[illegible]	[illegible]	[illegible]

User search for files



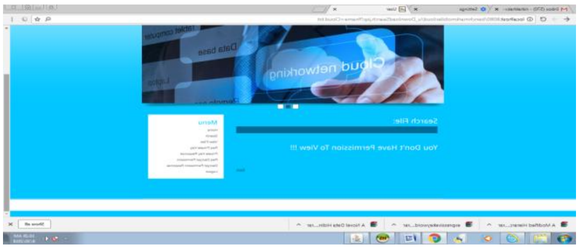
Search File:

File details



Search File:

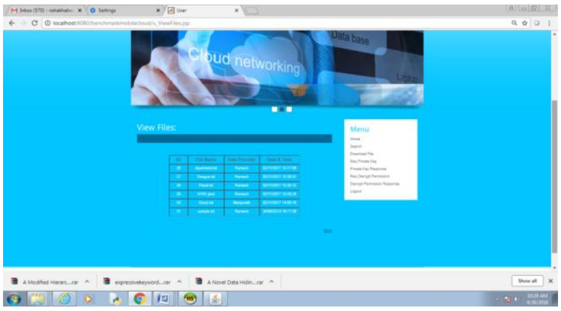
Checks for permissions



Search File:

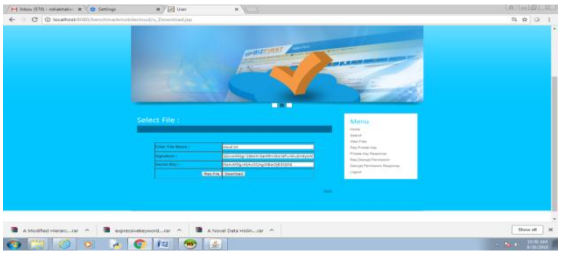
You Don't Have Permission To View !!!

All files uploaded details

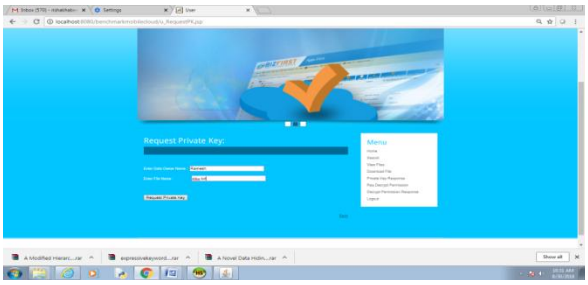


View Files:

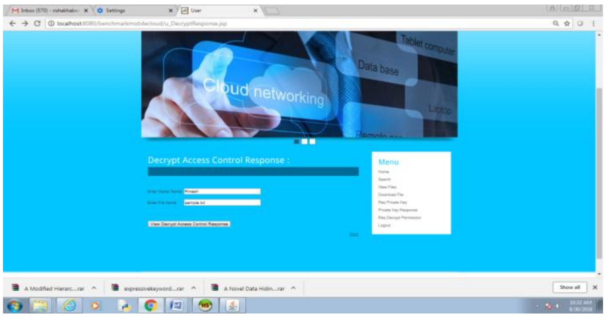
File download details


localhost:8080/benchmarkmobilecloud/u_Download.jsp
Select File :
Menu
Download
A Modified Hierarc....rar
expressivekeyword....rar
A Novel Data Hidin....rar
Show all

Request forprivate key for file download


localhost:8080/benchmarkmobilecloud/u_RequestPK.jsp
Request Private Key:
Menu
A Modified Hierarc....rar
expressivekeyword....rar
A Novel Data Hidin....rar
Show all

Decrypt file permissions


localhost:8080/benchmarkmobilecloud/u_DecryptResponse.jsp
Tablet computer
Data base
Cloud networking
Laptop
Decrypt Access Control Response :
Menu
sample.txt
View Decrypt Access Control Response
A Modified Hierarc....rar
expressivekeyword....rar
A Novel Data Hidin....rar
Show all

SYSTEM TESTING:

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

System Testing:

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing:

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing:

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

SYSTEM TESTINGMETHODOLOGIES

The following are the Testing Methodologies:

- Unit Testing.
- Integration Testing.
- User Acceptance Testing.
- Output Testing.
- Validation Testing.

Unit Testing

Unit testing focuses verification effort on the smallest unit of Software design that is the module. Unit testing exercises specific paths in a module’s control structure to ensure complete coverage and maximum error detection. This test focuses on each module individually, ensuring that it functions properly as a unit. Hence, the naming is Unit Testing. During this testing, each module is tested individually and the module interfaces are verified for the consistency with design specification. All important processing path are tested for the expected results. All error handling paths are also tested.

Integration Testing:

Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high order tests are conducted. The main objective in this testing process is to take unit tested modules and builds a program structure that has been dictated by design.

The following are the types of Integration Testing:

1. Top Down Integration:

This method is an incremental approach to the construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main program module. The module subordinates to the main program module are incorporated into the structure in either a depth first or breadth first manner. In this method, the software is tested from main module and individual stubs are replaced when the test proceeds downwards.

2. Bottom-up Integration:

This method begins the construction and testing with the modules at the lowest level in the program structure. Since the modules are integrated from the bottom up, processing required for modules subordinate to a given level is always available and the need for stubs is eliminated. The bottom up integration strategy may be implemented with the following steps:

- The low-level modules are combined into clusters that perform a specific Software sub-function.
- A driver (i.e.) the control program for testing is written to coordinate testcase input and output.
- The cluster is tested.
- Drivers are removed and clusters are combined moving upward in the program structure

The bottom up approaches tests each module individually and then each module is module is integrated with a main module and tested for functionality.

OTHER TESTING METHODOLOGIES

User Acceptance Testing

User Acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required. The system developed provides a friendly user interface that can easily be understood even by a person who is new to the system.

Output Testing

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in the specified format. Asking the users about the format required by them tests the outputs generated or displayed by the system under consideration. Hence the output format is considered in 2 ways – one is on screen and another in printed format.

Validation Checking

Validation checks are performed on the following fields.

Text Field

The text field can contain only the number of characters lesser than or equal to its size. The text fields are alphanumeric in some tables and alphabetic in other tables. Incorrect entry always flashes and error message.

Numeric Field

The numeric field can contain only numbers from 0 to 9. An entry of any character flashes an error messages. The individual modules are checked for accuracy and what it has to perform. Each module is subjected to test run along with sample data. The individually tested modules are integrated into a single system. Testing involves executing the real data information is used in the program the existence of any program defect is inferred from the output.

The testing should be planned so that all the requirements are individually tested. A successful test is one that gives out the defects for the inappropriate data and produces and output revealing the errors in the system.

Preparation of Test Data

Taking various kinds of test data does the above testing. Preparation of test data plays a vital role in the system testing. After preparing the test data the system under study is tested using that test data. While testing the system by using test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

USER TRAINING

Whenever a new system is developed, user training is required to educate them about the working of the system so that it can be put to efficient use by those for whom the system has been primarily designed. For this purpose the normal working of the project was demonstrated to the prospective users. Its working is easily understandable and since the expected users are people who have good knowledge of computers, the use of this system is very easy.

MAINTAINENCE

This covers a wide range of activities including correcting code and design errors. To reduce the need for maintenance in the long run, we have more accurately defined the user's requirements during the process of system development. Depending on the requirements, this system has been developed to satisfy the needs to the largest possible extent. With development in technology, it may be possible to add many more features based on the requirements in future. The coding and designing is simple and easy to understand which will make maintenance easier.

TESTING STRATEGY

A strategy for system testing integrates system test cases and design techniques into a well planned series of steps that results in the successful construction of software. The testing strategy must co-operate test planning, test case design, test execution, and the resultant data collection and evaluation .A strategy for software testing must accommodate low-level tests that are necessary to verify that a small source code segment has been correctly implemented as well as high level tests that validate major system functions against user requirements. Software testing is a critical element of software quality assurance and represents the ultimate review of specification design and coding. Testing represents an interesting anomaly for the software. Thus, a series of testing are performed for the proposed system before the system is ready for user acceptance testing.

CONCLUSION:

In this paper, we first give two attacks on DAC-MACS and EDAC-MACS for their backward revocation security. Then, a new effective data access control scheme for multi authority cloud storage systems (NEDAC-MACS) is proposed to withstand the two vulnerabilities in section 3 and thus to enhance the revocation security. NEDAC-MACS can withstand the two vulnerabilities even though the nonrevoked users reveal their received key update keys to the revoked user. In NEDAC-MACS, the revoked user has no chance to decrypt any objective ciphertext even if it actively eavesdrop to obtain an arbitrary number of nonrevoked users' Key Update Keys (KUK) or collude with some nonrevoked users or obtain any transmitted information such as Ciphertext Update Keys (CUK) . Then, formal cryptanalysis of NEDAC-MACS is presented to prove its improved security. Finally, the performance simulation shows the overall storage, computation, and communication overheads of the NEDAC-MACS are superior to that of DACC and relatively same as that of DAC-MACS.

REFERENCES:

[1] S. Subashini and V. Kavitha, “A survey on security issues in service delivery models of cloud computing,” J. Network and Computer Applications, vol. 34, no. 1, pp. 1-11, Jul. 2010

[2] K. Yang, X. Jia, and K. Ren, “DAC-MACS: Effective data access control for multi authority cloud storage systems,” IEEE Trans. Information Forensics and Security, vol. 8, no. 11, pp. 1790-1801, Nov. 2013

[3] Kan Yang and Xiaohua Jia, "Expressive, Efficient, and Revocable Data Access Control for Multi-Authority Cloud Storage," IEEE Trans. Parallel and Distributed Systems, vol.25, no.7, pp.1735-1744, July 2014

[4] A. Sahai and B. Waters, “Fuzzy identity-based encryption,” Proc. EU- ROCRYPT’ 05, pp. 457-473, 2005

[5] V. Goyal, O. Pandey, A. Sahai, and B. Waters, “Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data,” Proc. ACM Conf. Computer and Comm. Security, pp. 89-98, 2006

[6] J. Bethencourt, A. Sahai, and B. Waters, “Ciphertext-Policy Attribute- Based Encryption,” Proc.IEEESymp.Security& Privacy, pp. 321-334, 2007

[7] R. Ostrovsky, A. Sahai, and B. Waters, “Attribute-Based Encryption with Non-Monotonic Access Structures,” Proc. ACM Conf. Computer and Comm. Security, pp. 195-203, 2007.