Privacy Preserving And Dynamic Key Management For Key Aggregate System In Large Cloud Auditing System

Mr. SIRAJDDOLA NADAF
Lecturer
Dept of Computer Science and Engineering.
Government Polytechnic BILAGI, Karnataka, India.

ABSTRACT - Data sharing is an important functionality in cloud storage. In this paper, we show how to securely, efficiently, and flexibly share data with others in cloud storage. We describe new public-key cryptosystems that produce constant-size ciphertexts such that efficient delegation of decryption rights for any set of ciphertexts are possible. The novelty is that one can aggregate any set of secret keys and make them as compact as a single key, but encompassing the power of all the keys being aggregated. In other words, the secret key holder can release a constant-size aggregate key for flexible choices of ciphertext set in cloud storage, but the other encrypted files outside the set remain confidential. This compact aggregate key can be conveniently sent to others or be stored in a smart card with very limited secure storage. We provide formal security analysis of our schemes in the standard model. We also describe other application of our schemes. In particular, our schemes give the first public-key patient-controlled encryption for flexible hierarchy, which was yet to be known.

1. INTRODUCTION

Cloud computing give a expandable environment for increasing quantity of data in different Peers and method that work on a variety of applications on-demand self service. The main advantage of cloud computing is that data are being centralized and out sourced in clouds. This type of outsourcing storage space in clouds has become a new revenue increase by providing a cheap price, extendable, location without depending on other operating systems for supervision clients' information. CSS ease load on storage space supervision and safeguarding. These security problems take place from the subsequent motive: the cloud communications are a great deal for trustworthy than individual work out procedure. However, they are still facing some sort of inside and outside intimidation; there exist a variety of inspiration for CSP to act faithlessly towards the cloud consumer; in addition, disagreement infrequently experience from a be short of belief on CSP. So, actions may not be identified by the cloud consumer, conflicts may possibly

effect from client own offensive operations. So, it is important for cloud resource providers to suggest proficient inspection facility to make sure the integrity and accessibility of the data which is in database.

Inside this project, we bring in a lively inspection facility for truthfulness proof of un trusted and contract out storage space. In this audit method, we provide new audit system be able to carry active data actions and well-timed uncharacteristic findings with the assist of other practices, like fragmentation, casual samples, and indexing and hashing tables. in addition, we put forward well-organized approach based on enquiry and episodic substantiation getting better act of audit tasks. The sample is also implementing to assess the practicality of our planned loom. Our tentative out puts not only legalize the efficacy of our approaches, nevertheless it give you an idea about our system has a a good deal for lesser working out fee, as well as a tiny the additional storage for truthfulness proof.

1.1 Purpose of the project

In this project, vital facts that have to be deal i.e. promise the customer about integrity of data in the cloud environment. As the data is actually not reachable to the user cloud ought to offer consumer to confirm if the truthfulness or integrity of his data is keep

1.2 Motivation

As the cloud storage technologies has become the need of an hour, Securing data in cloud storage has become a great deal last few years and as so lots of R & D are going on in order to accomplish stronger security and because the data is actually not easy, get to the user the cloud have to make available to the user, make sure if the integrity data is protected.

2. EXISTING SYSTEM

Considering data privacy, a traditional way to ensure it is to rely on the server to enforce the access control after authentication, which means any unexpected privilege escalation will expose all data. In a shared-tenancy cloud computing environment, things become even worse.

Regarding availability of files, there are a series of cryptographic schemes which go as far as allowing a third-party auditor to check the availability of files on behalf of the data owner without leaking anything about the data, or without compromising the data owners anonymity. Likewise, cloud users probably will not hold the strong belief that the cloud server is doing a good job in terms of confidentiality.

A cryptographic solution, with proven security relied on number-theoretic assumptions is more desirable, whenever the user is not perfectly happy with trusting the security of the VM or the honesty of the technical staff.

2.1 DISADVANTAGE OF EXISTING SYSTEM

1. The costs and complexities involved generally increase with the number of the decryption keys to be shared.

2. The encryption key and decryption key are different in public key encryption.

3. There is no system to generate unique key to access multiple files.

4. The costs and complexities involved generally increase with the number of the decryption keys to be shared.

5. The encryption key and decryption key are different in public key encryption.

6. Identity based encryption instead attributes based encryption

In this paper, we study how to make a decryption key more powerful in the sense that it allows decryption of multiple cipher texts, without increasing its size. Specifically, our problem statement is "To design an efficient public-key encryption scheme which supports flexible delegation in the sense that any subset of the cipher texts (produced by the encryption scheme) is decryptable by a constant-size decryption key (generated by the owner of the master-secret key)." We solve this problem by introducing a special type of public-key encryption which we call key-aggregate cryptosystem (KAC). In KAC, users encrypt a message not only under a public-key, but also under an identifier of cipher text called class. That means the cipher texts are further categorized into different classes. The key owner holds a master-secret called master-secret key, which can be used to extract secret keys for different classes. More importantly, the extracted key have can be an aggregate key which is as compact as a secret key for a single class, but aggregates the power of many such keys, i.e., the decryption power for any subset of ciphertext classes.

PROBLEM DEFINITION AND PROPOSED MODEL

3.1 Problem Statement

In current system present diverse inspiration for CSP to act faithlessly towards the cloud consumer; also, disagreement rarely experience be short of trust on CSP. as a result, behaviours may not be recognized by the cloud consumer, yet difference of opinion may outcome from the customers unacceptable operations. Thus, obligatory for CSP to put forward an competent audit facility to check the integrity and ease of use of the data.

3.2 Proposed Model

Proposed system overcomes the issues of existing system we bring in a lively inspection task for integrity proof of un-trusted users and storage spaces. New audit systems, which can support energetically, check the truthfulness of data.

SYSTEM REQUIREMENT

These are the following system software, hardware functional and non functional requirements

4.1 System Requirement Specification

Table: 4.1 Summaries of SRS

Functional	Control the file access at cloud server, Proxy Server authenticates Users, Data, Multiple key aggregations, File Privacy Management, Automatic Group Member Revocation.
Non- Functional	Data Owner never monitors the Cloud activities
External interface	LAN , WAN, Routers
Performance	Finding File Hacker Information, File Sharing efficiency fairness between Cloud Server and Remote User, Revocation of the File Hackers in the cloud
Attributes	File Management, Public auditing, Proxy Server, Cloud Server, Data Owner, Remote Users, Public Verifier, User Revocation, User Un Revocation

4.2 Functional Requirements

- The proprietor or Group component uploads the files to cloud server machine.
- The Cloud server has to allow the suitable isolated users by verifying the Digital Signature. if the isolated users malicious then he has to revoke in the public verifier. The Hacker details will be registered by the cloud machine server. The cloud server will generate the Shamir Secret Key to verify and authorize the end user.
- The Public Verifier has to uphold the fault localization (User revocation and un revocation) and has to keep an eye on the Cloud Server machine actions.
- The isolated user has to correct combined aggregated privet key and file name, Digital Signature. If anybody incorrect then he/she is observe as malicious user.
- File administration, Public auditing, Proxy Server, Cloud Server, Data proprietor, isolated Users, Public Verifier, client Revocation, Un Revocation client.

4.3 Non – Functional Requirements

The key non-functional requirements are:

- Security: The computer machine have to permit a protected communication among PS and CS, customer and File proprietor
- Energy effectiveness: The power consumed by the Users to obtain the File information from the CS machine
- Trustworthiness: Machine has to steadfast and must not mortify the act of the live machine and must not guide to the lynching of the machine.

USER INTERFACE

5. GUI Components

JButton, JLabel, JTextField, JTextArea, JFrame, JTabbedPane, JScrollPane, Container.

JButton: JButton is used to transmit, plain, hop add up, obtain data set

JLabel : It displays data small text string

JTextField JTextField is element that allows the editing of a single line of text.

JTextArea JTextArea multi-line area that demonstrates plain text. In the development environment, it is used to transmit the data and to collect the data. The user enters the message to send the data.

JScrollPane

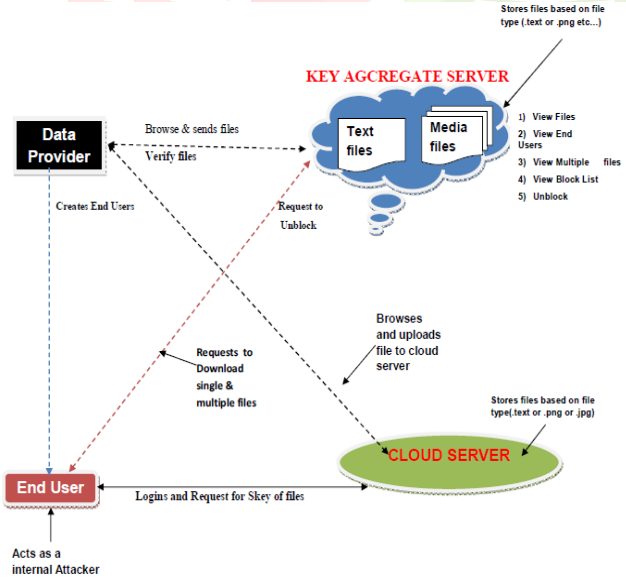
Provides a scrollable view of a light weight component. A JScrollPane manages a viewport, optional vertical and horizontal scroll bars, and optional row and column heading viewports.

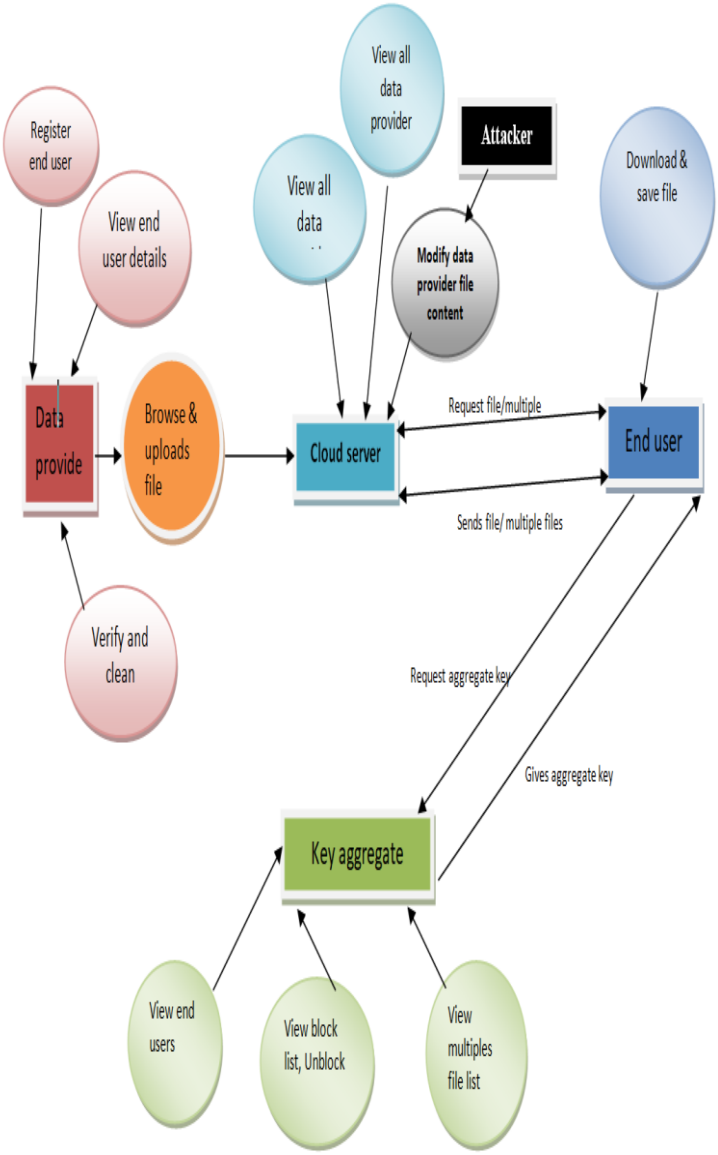
JTabbedPane : This allows user exchange data between a group of elements by clicking on a tab.

Container : A generic Abstract Window Toolkit (AWT) container object and it is storage place where component include components.

6.ARCHITECTURE DIAGRAM

Fig . Design of the proposed model





Data Flow diagram

Data Flow Diagram:

A data-flow diagram (DFD) is a graphical representation of the "flow" of data through an information system. DFDs can also be used for the visualization of data processing (structured design). On a DFD, data items flow from an external data source or an internal data store to an internal data store or an external data sink, via an internal process DFD provides no information about the timing of processes, or about whether processes will operate in sequence or in parallel.

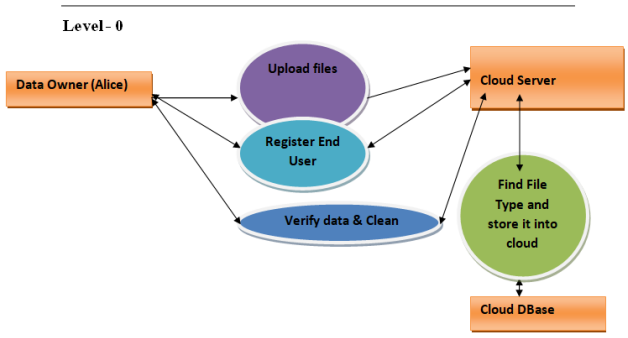
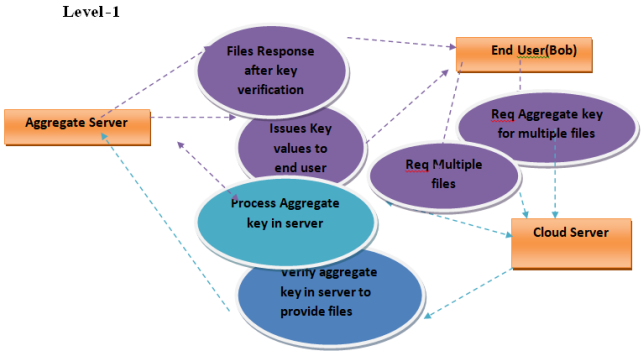


Fig.6.2.2 Data Flow diagram



Data Flow diagram

7. Use Case Diagram

It is graphical presentations of data owner, key aggregate server, cloud server, and end user, along with this their work, job.

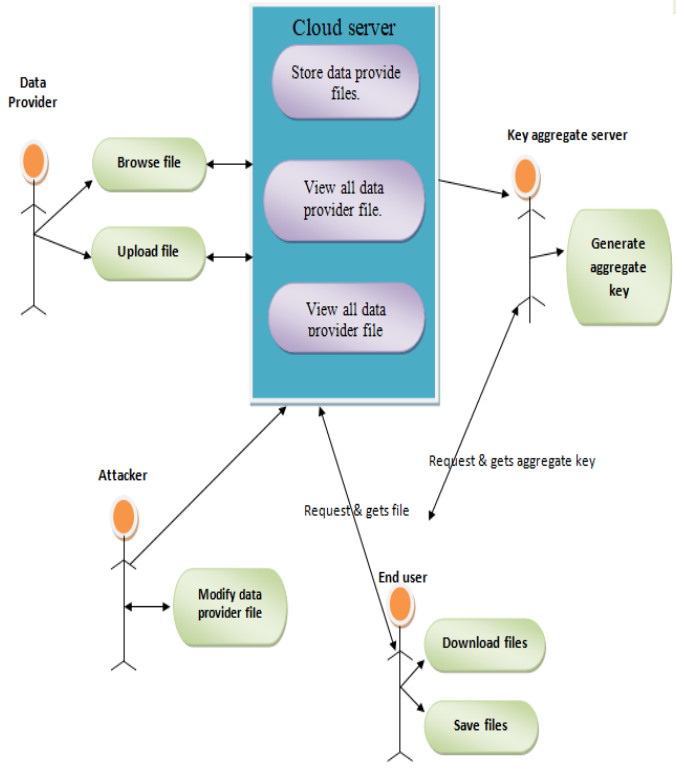
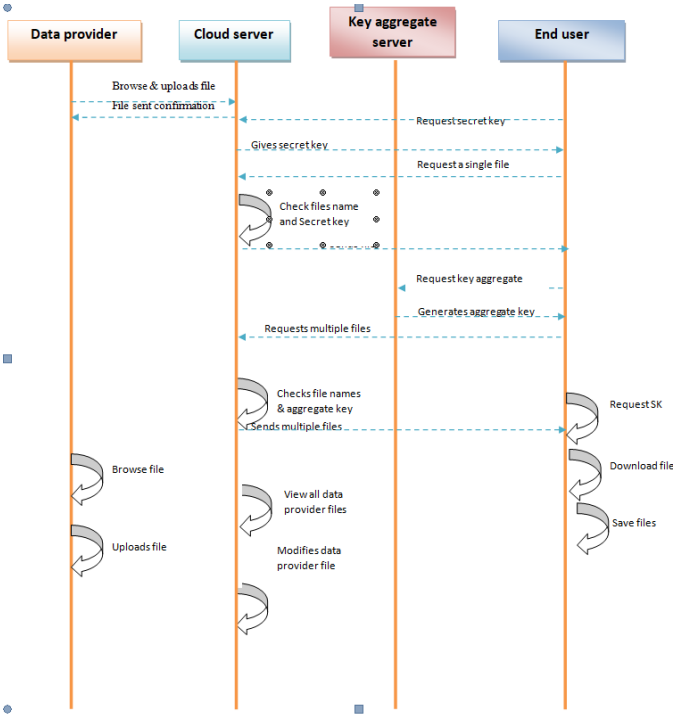


Fig.6.3 Use Case diagram

8. Sequence Diagram

It is flow of data between data supplier, cloud server, key aggregate server and end consumer



8. Class Diagram:

A Class figure is a sort of relations figure that shows how classes interact data owner, cloud server machine, end user etc.

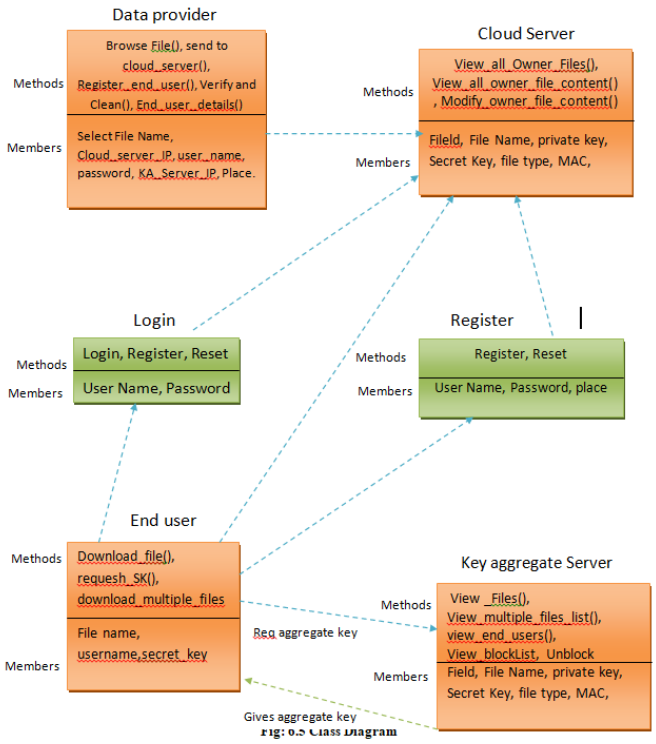


Fig: 6.5 Class Diagram

9. SYSTEM IMPLEMENTATION

- **Cloud Server**

  The cloud server supervises a cloud machine to offer data storage facility. Data proprietor encrypts the files and stores them in the cloud machine for sharing with consumers. To access the collective files, consumers download encrypted files which they want from the cloud and then CS will decrypt files. The cloud will generate the aggregate key if the end user requests multiple files at the same time to access.

- **Key Aggregate Server**

  The Key Aggregate Server is responsible to generate the master key for multiple files and this is responsible for permit access permission for multiple files.

- **Multiple files Key Aggregation**

  In this module gets a aggregate key for multiple files it can be text file or Multimedia (Video, Image) files this is been provided by the data provider to end-user master-secret key called Aggregate key.

- **END User**

  In this part, the user is able to only right to use the data file by means of the encrypted key. Files it can be Text File or Multimedia File. The user can access the multiple files from Cloud Server via Key Aggregate Server using Aggregate Key. Therefore wicked malicious persons possibly will plan with dig up sensitive files by using un authorized key.

1. AES - File Encryption and decryption
2. SHA1 - Secured Hash Algorithm - for generating Digital Signature
3. Aggregate Key Generation algorithm - To generate an unique key for multiple file request
4. RSA - To generate Public Key for accessing file

Data provider

The data supplier uploads records to the CS. For the safety reason the proprietor encrypts the file and then store in to the cloud. The proprietor be able to have competent of operate the encrypted data file. The proprietor set up the public system parameter via Setup and produces an aggregate key/master-secret key pair via Key Aggregate Server.

Modules:

1. Browse
2. Send To Cloud Server
3. Register End Users
4. Verify and Clean
5. End User Details

REGISTRATION OF END USER

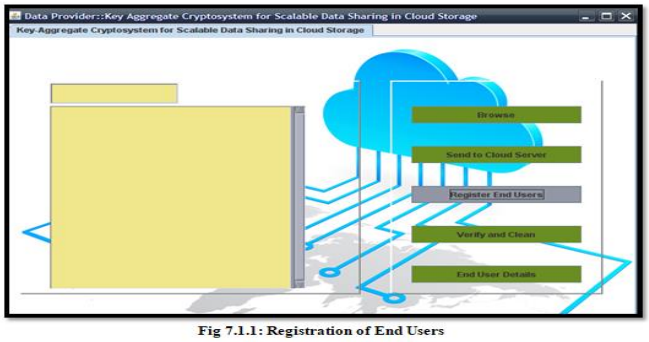
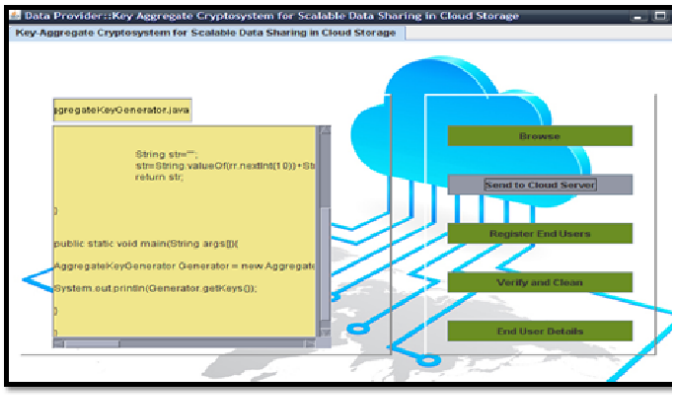


Fig 7.1.1: Registration of End Users

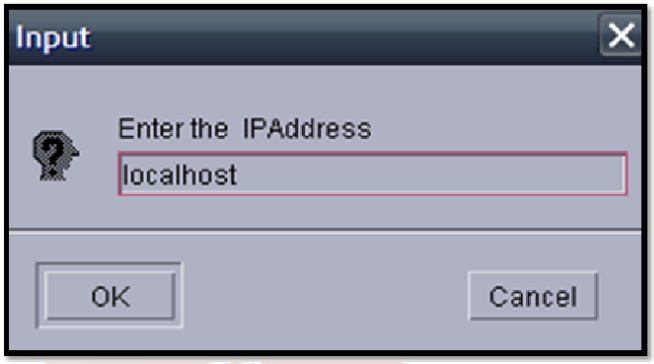
REGISATRAION OF SUCCESSFUL

Fig 7.1.2: Registration of End Users

SEND DATA TO GHE CLOUD SERVER



Send Data to Cloud Server



UPLOADING FILE TO THE SERVER

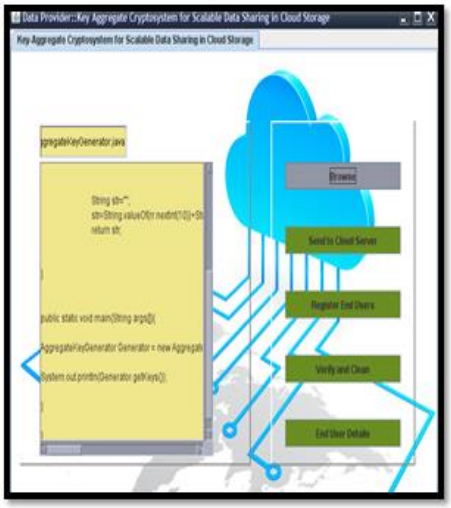
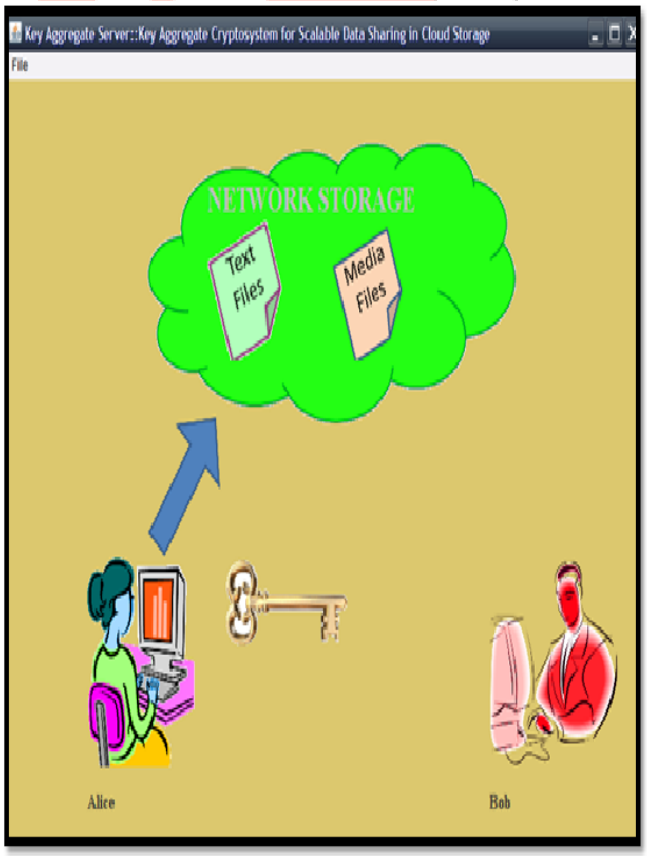


Fig 7.1.4 Uploading files to Cloud Server



Data is Uploading to Cloud Server

10. TESTING AND RESULTS

Test Id	Test Name	Input	Output	Expected Result	Status
1	Data Provider creates Remote user	Remote User Details	Registration Success	Registration Success and details stored in Key Agg Server	PASS
2	Data Provider browsing the data	file	Data stored	Data stored in encrypted form	PASS
	Data Provider browsing data	data	Data is not Present in directory	Display Data	FAIL
3	Data Provider adding data into server	Server IP address	Data stored in server	Data stored in server in encrypted form	PASS
	Data Provider adding data to server	Invalid server IP address	Data cannot stored in server	Data stored in server	FAIL
4	Data Provider adding data to KAS	Key Aggregate Server IP Address	Data stored in key Aggregate Server	Metadata stored in Key Aggregate Server	PASS
	Data Provider Adding data to Key Aggregate Server	Invalid KAS IP address	Data cannot stored in KAS	Metadata stored in Key Aggregate Server	FAIL
5	Data Provider viewing end user details	Key Aggregate server IP address	Details of End User	Details of End User	PASS
	Data Provider viewing end user details	Wrong IP address of Key Aggregate Server	Details of End User not available	Details of End User	FAIL
6	Data Provider verify and clean file	File name and server IP address	File is safe	File is safe	PASS
	Data Provider verify and clean file	Wrong file name and IP address	File not Found	File is safe	FAIL
7	Attacker modifies file in server	Some extra code	File modified alert to Attacker	File modified alert to Attacker	PASS
	Attacker modifies file in server	wrong server IP address	File not modified in server	File modified alert to Attacker	FAIL
	Attacker modifies file in server	wrong server IP address	File not modified in server	File modified alert to Attacker	FAIL
8	End User logging in	Enter Valid user name and Password	Login successful	Login Success and End User Module got activated	PASS

	End User logging in	Enter invalid details	Login Failure	Login Success and End User Module got activated	FAIL
9	End User requesting Skey	File name	Requested Secrete Key	End User get Requested Skey	PASS
	End User requesting Skey	Wrong File name	End User not get Skey	End User get Requested Skey	FAIL
10	Receiving file	File Name and Secret key	Received file	File received	PASS
	Receiving file	Wrong File name or Secret key	Becomes attacker	File received	FAIL
11	Requesting Aggregate Key	No of files, File name and Key Aggregate Server IP Address	End user get Aggregate Key	End User will get Aggregate Key	PASS
	Requesting Aggregate Key	No of files, wrong file name or KAS IP address	End User not get Aggregate Key	End User will get Aggregate Key	FAIL
12	Receiving Multiple files	Files Name and aggregate key	Received file	File received	PASS
	Receiving Multiple files	Wrong File name or aggregate Key	File not received	File received	FAIL
13	Cloud Server viewing all owner files	----	Get owner file details	Get owner file details	PASS
	Cloud Server viewing all owner files	---	Cannot get owner file details	Get owner file details	FAIL
14	Cloud Server viewing owner file content	File name	Get content of file	Content received	PASS
	Cloud Server viewing owner file content	Wrong file name	Will not get content of file	Content received	FAIL
15	Cloud Server Modifying file	Some Extra content & server IP address	File modified alert to cloud server	File Modified alert to server	PASS
	Cloud Server Modifying file	Wrong server IP address	File will not modified	File Modified alert to server	FAIL

CONCLUSION AND FUTURE SCOPE

The Proposed system achieves the integrity of the users in cloud storage spaces at data owner side, this has been achieved with more numerical, cryptographic methods, which are reaching more flexible and habitually engage multiple keys for a single application. We think over how to "compress" secret keys in public-key cryptosystems which giving out of secret keys for dissimilar cipher text classes in cloud storage. No matter which one among the power set of classes, the delegate can always get an aggregate key of steady size. Our approach is more flexible than hierarchical key assignment which can only save spaces if all key-holders share a similar set of privileges.

Future Scope is that while downloading file or data alert message has been generated immediately to the Remote User if the file or data got tempered or modified. Another limitation in our work is the predefined bound of the number of maximum cipher text classes in cloud storage, the number of cipher texts usually grows rapidly. So we have to reserve enough cipher text classes for the future expansion. Otherwise, we need to expand the public-key as we described in this system

REFERENCES

[1] Dan Boneh, Xavier Boyen, eu-jin goh "Hierarchical identity based encryption with constant size ciphertext" by RYPT 2005 , volume 3493 of Lecture Notes in Computer Science,

[2] Mate Horavath Attribute-Based Encryption Optimized for Cloud Computing LNCS 8939, pp. 566{577 cSpringer-Verlag Berlin Heidelberg 2015.

[3] Giuseppe Ateniese, Kevin Fu,Improved Proxy Re-Encr Secure Distributed Storage ACM Transactions on Informati Issue 1, February 2006

[4] Tatsuaki Okamoto Achieving short ciphertexts or short secret-keys for adaptively secure general inner-product encryption Katsuyuki Takashima Mitsubishi Electric Vol 2536 july 11, 2012

[5] Boyang Wang,Sheremn M chow Storing shared data on the cloud via security-mediator ICDCS '13 Proceedings of the 2013 IEEE 33rd International Conference on Distributed Computing Systems Pages 124-133

[6] Jin LiA Hybrid Cloud Approach for Secure Authorized Deduplication Published in:Parallel and Distributed Systems, IEEE Transactions on (Volume:26 , Issue: 5)

[7] Patric P c A cloud enviroment for backup and data storage Published in: Electronics, Communications and Computers (CONIELECOMP), 2014 International Conference on

[8] Cheng-Kang Chu Key-Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage Parallel and Distributed Systems, IEEE Transactions on (Volume:25 , Issue: 2)