A Key Management Technique for Cloud Storage Using Semi Trusted Policy Preservation

[1]Rupa Kesavan, [2]Vijayaraja L

[1]Assistant Professor, [2] Assistant Professor
[1]Computer Science Engineering, [2]Electrical and Electronics Engineering
[1] Prince Shri Venkateshwara Padmavathy Engineering, Chennai, India
[2] Sri Sairam Institute of Technology, Chennai, India

Abstract : One of the major challenge in cloud computing is providing security for the data. The main aim of this project is to provide a security for data in cloud and also to propose a proper key management system using key sharing technique with multiple key managers. To avoid unauthorized access to cloud data, access control mechanism must be enforced. Moreover, data leakage and data privacy strategies must be employed so that only authorized users can access and utilize data. Encryption techniques provide a solution to ensure privacy and confidentiality of stored data. Key management becomes a prime issue in the case of encryption. Cryptographic keys need to be stored and protected. Compromise or failure of a key storage facility may lead to the loss of data. Another issue in security for cloud data is that there is a man-in-the-middle attack. The intruder may access the user files illegally due to improper authentication system. This can be overcome by using proper authentication between user and cloud. The security concerns of outsourcing data to public clouds, serves as a motivation to work for the development of data security technique. We aim for a technique capable of addressing the aforementioned critical issues. We proposed a data security scheme that uses key manager servers for the management of cryptographic keys

IndexTerms **- Data Security for Cloud Environment with Semi-Trusted Third Party (DaSCE), Digital Encryption Standard (DES) and GFS (Google File System).**

I. INTRODUCTION

Outsourcing data to a third-party administrative control entails serious security concerns. Data leakage may occur due to attacks by other users and machines in the cloud. Wholesale of data by cloud service provider is yet another problem that is faced in the cloud environment. Consequently, high-level of security measures is required. Digital technology is changing and it is becoming more and more important to adopt ways that increase time and cost efficiency. Furthermore, with reports indicating that mobile usage is consistently growing, the benefits of cloud computing for businesses become clear[1]. The cloud is evolving rapidly, encompassing bigger and better things; the time when it was only used to store data is now long gone and it has emerged as a multifaceted ecosystem that allows management and usage of resources in an efficient manner ,allows scaling of data and allows outsourcing of data across different mobile devices.

Data being the principal asset for organizations needs to be secured. Especially, when data must enter a public cloud. To avoid unauthorized access to cloud data, access control mechanism must be enforced. The construction of cloud-enabled mobile applications can take advantage of existing resources. Mobile application in cloud platform could be regarded as a special IoT application in CoT. With the wide employment of IoT application, the physical devices allow users to share the mobile services conveniently [2]. Along with the extensive use and fast improvement of web information technology, an urgent need is generated from end users to develop or configure web-based mobile applications rapidly especially in cloud environment. End users are suggested to implement their customized applications by reusing existing IT resources including system components, services and databases on a cloud platform. On the other hand, to meet the continuously changing personal requirements for business purpose, these mobile applications are required not only to be developed rapidly, but also to be adjusted easily.

Storing data in a third party's cloud system causes serious concern over data confidentiality. Constructing a secure storage system that supports multiple functions is challenging when the storage system is distributed and has no central authority. This paper use threshold proxy re-encryption scheme and integrate it with a decentralized erasure code such that a secure distributed storage system is formulated. The distributed storage system not only supports secure and robust data storage and retrieval, but also lets a user forward his data in the storage servers to another user without retrieving the data back. The main technical contribution is that the proxy re-encryption scheme supports encoding operations over encrypted messages as well as forwarding operations over encoded and encrypted messages. This method fully integrates encrypting, encoding, and forwarding [3].

The later requirement is especially important because related personal or business requirements are always changing. However, different from other distributed IoT applications, Mobile Services in cloud computing are restricted by very limited resources which bring out some important considerations as following: (1) Data contents and related I/O operations are the main consideration for mobile services development. Complex logics such as image processing or scientific calculation are less used in such situation, or these computing-massive operations are always carried out in cloud platform other than front devices with limited computing ability. (2) Resources are limited both from available services and IT components.

And the developing process of a mobile service is somewhat in a lightweight service composition or component configuration manner on a relatively simple business requirement. (3) Cooperation between mobile devices is very important for complementing intelligent interaction. Contextual data are required to support decision making from an interaction from different devices.

And rules for dynamic behaviors are important for intelligent applications. Research challenges in mobile application development cover different levels from identification and communication to distributed system and distributed intelligence [4]. Due to the highly complex system and a huge quantity of heterogeneous data exist, related information and relations are too complicated for data representation and processing. Information integration and intelligent interaction cannot be archived effectively by means of data integration. From the view of data processing, these challenges could be divided into different stages of data abstracting, data gathering, data integration, and intelligent interaction. Therefore IoT application not only needs to realize data integration but also requires realizing intelligent interaction [5]. Aim to facilitate mobile services development so as to rapidly construct an intelligent application with adaptive features when changes occurs, a model-driven service configuration architecture is provided for web-based mobile application development in cloud computing. Starting from the point of resource configuration of cloud platform, a meta-model covering multi-view business models and system components is provided for model abstraction and management [6]. Based on a formal representation language, automatic model transformation and service integration are realized in three patterns. Furthermore, a related development platform is also provided for verification.

The contributions can be concluded as following points: (1) A united meta-Model which integrates CIMs and PIMs is proposed as a referenced structure to encapsulate and manage business and IT resources for application implementation in cloud platform. Thus, by modeling process, function, organization, and data, different views of business information are connected with executable services by a united information framework [7]. (2) Three development patterns of Role-driven, Data driven, and Process-driven are given as service configuration. Based on common service resources in cloud platform, these patterns provide a best practice for rapid mobile service development. Since formal language is also introduced in the process, the adjustment of mobile application based on three patterns can be carried out quickly by means of relation reasoning. (3) Dynamic behaviors between different mobile devices and users can also be configured based on the semantic relations between IT components [8]. By means of resource-based contextual structure, intelligent interaction can be realized for complex application with adaptive actions.

With the development of cloud computing, data security becomes more and more important in cloud computing. The paper analyses the basic problem of cloud computing data security. With the analysis of HDFS architecture, we get the data security requirement of cloud computing and set up a mathematical data model for cloud computing. Finally a data security model for cloud computing was built. The model used three-level defense system structure, in which each floor performs its own duty to ensure that the data security of cloud layers. HDFS is used in large-scale cloud computing in typical distributed file system architecture, its design goal is to run on commercial hardware, due to the support of Google, and the advantages of open source, it has been applied in the basis of cloud facilities. HDFS is very similar to the existing distributed file system, such as GFS (Google File System); they have the same objectives, performance, availability and stability. Thus the model discusses the cloud computing environment with the safety issues through analyzing a cloud computing framework-HDFS's security needs [9].

The FADE is a light-weight and scalable technique that assures the deletion of files from cloud when requested by the user. It is practical, implementable, and readily deployable cloud storage system that focuses on protecting deleted data with policy-based file assured deletion. FADE is built upon standard cryptographic techniques, such that it encrypts outsourced data files to guarantee their privacy and integrity, and most importantly, assuredly deletes files to make them unrecoverable to anyone (including those who manage the cloud storage) upon revocations of file access policies. In particular, the design of FADE is geared toward the objective that it acts as an overlay system that works seamlessly atop today's cloud storage services. However, during our analysis, FADE fell short on issues of security of keys and authentication of participating parties. In this existing process there is a man-in-the-middle between client and key manager [10]. The intruder can intercept user policy and send modified policy to KM. It mainly focus on deleting he files from cloud when the user requested. Key management becomes a prime issue in the case of encryption. Compromise or failure of a key storage facility may lead to loss of data. Man-in-the-middle attacks cannot be prevented.

In this paper, section II describes the work carried out to to develop and customize mobile applications, section III gives a brief explanation of framework and section IV explains the architecture, section V gives methods to implementation and section VI presents the screen shoots of the process.

II. USER TO DEVELOP AND CUSTOMIZE MOBILE APPLICATION

There are many ideas and technologies for end users to develop and customize mobile applications. On the view of development process, we divide related researches into four areas:

Cataloging of user and policy setting.
File upload and policy file creation.
File download.
Policy revocation and policy renewal.

We proposed a data security scheme that uses key manager servers for the management of cryptographic keys. Shamir's (k,n) threshold scheme is used for the management of keys that uses k shares of n to rebuild the key. Therefore, cryptographic keys must be stored in robust manner and a single point of failure should not affect the availability of data. Proposed system makes use of both symmetric and asymmetric keys. Out of the key pair, only public key is transmitted to the client. For secure transmission of keys, a secret key is established between client and KM.

2.1 CATALOGING OF USER AND POLICY SETTING

In this module user has to register to become a member in cloud, once they registered user has to choose some attributes (e.g. name, email, address etc..,) and also give some user defined attributes to encrypt their policy file which is created while file uploading process. This Attribute Based Encryption performed using Elgamal algorithm.

In registration, we will give some personal information such as name, email id, phone number, password, city, state and country. Password should be only 4 characters.That 4 characters is converted to 7 digit hash code using hashcode() function.
For example: If 3522673 is the 7 digit hash code and the encryption keys are split, then 1st key is given 3rd key manager, 2nd key is given to 5th key manager and so on. Then we have to select the default attributes and user defined attributes. Any number of default attributes can be selected i.e. we can select either all or few of the default attributes. For example: Email id, phone number and city only can be selected. Then the user defined attributes have to be specified. These may be any private information of the user so that no third person can knew it. For example: favorite things or person name can be given. Both of the default attributes and the user defined attributes have to be provided to cloud when we want to download the file and to renew the policy file.

2.2 FILE UPLOAD AND POLICY FILE CREATION

When data must be uploaded to the cloud, the client requests the *KM* to generate a public/private key pair. The said is done by sending a policy file to the *KM*. The *KM* generates the key pair, associate that with the policy file, and sends the public part of the key to the client. After completing the registration process, authentication process will be performed between user and key manager using Diffie-Hellman key exchange Algorithm. This algorithm is used to exchange the key secretly. It is secure because of the complexity to compute discrete logarithms.

STEPS:
1. Choose a prime number 'q'
2. Assume 'a' such that 'a' is a primitive root of 'q' and a < q
3. Select such that Xa <q
4. Calculate Ya

$$Ya = aXa \bmod q$$

5. Select such that Xb <q
6. Calculate Yb

$$Yb = aXb \bmod q$$

Key for User A, k = (Yb)Xa mod q
Key for User B, k = (Ya)Xb mod q

Now the user A and B gets their respective secret keys. After that user will encrypt their file using secret key which is provided by cloud, based on user attributes and then it will uploaded into cloud. Simultaneously the policy file will also be generated and it contains username, filename and access permission, by default user access permission will be allowed.

Rivest-Shamir-Adleman (RSA) algorithm is used to generate keys for the key manager. The public part of the key is encrypted and assigned to the key managers. Digital Encryption Standard (DES) is a efficient algorithm which is used to encrypt the user files which is to be uploaded into the cloud and also to encrypt the policy file. Now user breaks up secret key into n shares (S1, S2…Sn) by using Shamir's key sharing technique and user encrypts their i-th key share with public key of i-th key manager[11].

2.3 FILE DOWNLOAD

File download module describes the downloading of the file from cloud. If the user need to download their file, then they will send request to Key-Manager with appropriate attributes. The attributes include both default and user defined attributes. Key-Manager will check the attributes provided by the user whether they match with the existing attributes and decrypt the appropriate user's policy file and check the user's file access permission for authenticated user [12].

If the attributes did not match, the KM will not allow to do further process. Otherwise if the attributes matches, the Key-Manager will decrypt user secret key by using their own private key and they provide decrypted i-th share to the requested user. The process starts with the client downloading the data from the cloud. To decrypt the file, we need the key that is encrypted with secret key. The secret key is encrypted with private key parameter received from *KM*.

The snippet for downloading image content:

```
request.setAttribute("getimage", filename);
```

```
RequestDispatcher rd=request.getRequestDispatcher("ImageContent.jsp");
rd.forward(request, response);
```

The snippet for downloading file content:

```
request.setAttribute("getcontent", decrypedcontent.replace(",", ""));
RequestDispatcher rd=request.getRequestDispatcher("FileContent.jsp");
rd.forward(request, response);
```

2.4 POLICY REVOCATION AND POLICY RENEWAL

In this module the user can do revocation and renewal of policies. For policy revocation user can send revocation request to the key-manager. Revocation is the process of removing all the policies that the user set before. Once the user policy revocation request is send to key-manager, they delete all the policies of the user. In policy renewal the cloud will allow to renew the user existing policy for particular file. Once he/she got approval from cloud, then user will renew their policy. For creating new policy the key manager ask the existing attribute as well as new attributes through which identify the old policy was identified and replaced by new one. Now the key manager will generate new set of key and encrypt the user's policy file by using new user's policy.

III. FRAMEWORK

For the purpose of supporting both data integration and intelligent interaction for mobile application, some major requirements are given from information processing phases such as device identification, abstracting, data transformation and integration and interaction.

3.1 DOMAIN INTRODUCTION

Cloud computing involves deploying groups of remote servers and software networks that allow centralized data storage and online access to computer services or resources. Clouds can be classified as public, private or hybrid. Cloud computing relies on sharing of resources to achieve coherence and economies of scale, similar to a utility over a network. At the foundation of cloud computing is the broader concept of converged infrastructure and shared services shown in Fig 3.1.

Cloud computing offer their service according to several fundamental models.

- Platform as a service.
- Software as a service.
- Infrastructure as a service

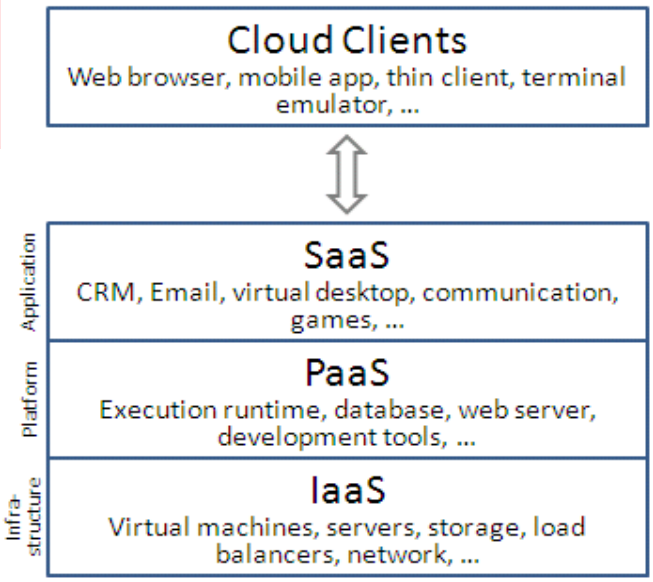

Fig 3.1: Service models

3.2 PLATFORM AS A SERVICE

In the PaaS, cloud providers deliver a computing platform, typically including operating system, programming language execution environment, database, and web server. Application developers can develop and run their software solutions on a cloud platform without the cost and complexity of buying and managing the underlying hardware and software layers. The advantages to PaaS are primarily that it allows for higher-level programming with dramatically reduced complexity; the overall development of the application can be more effective, as it has built-in infrastructure; and maintenance and enhancement of the application is easier.

3.3 SOFTWARE AS A SERVICE

In the software as a service (SaaS), users are provided access to application software and databases. Cloud providers manage the infrastructure and platforms that run the applications. In the SaaS model, cloud providers install and operate application software in the cloud and cloud users access the software from cloud clients. This eliminates the need to install and run the application on the cloud user's own computers, which simplifies maintenance and support.

3.4 INFRASTRUCTURE AS A SERVICE

Infrastructure as a Service (IaaS) is a form of cloud computing that provides virtualized computing resources over the Internet. In an IaaS model, a third-party provider hosts hardware, software, servers, storage and other infrastructure components on behalf of its users. Cloud applications are different from other applications in their scalability—which can be achieved by cloning tasks onto multiple virtual machines at run-time to meet changing work demand. Load balancers distribute the work over the set of virtual machines. This process is transparent to the cloud user, who sees only a single access point. There are three types of cloud as shown in Fig 3.2.

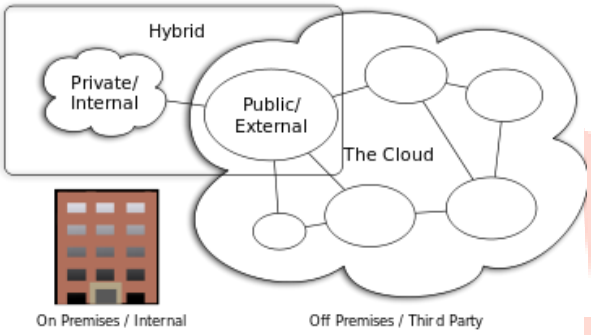

Fig 3.2: Cloud computing types

PRIVATE CLOUD

Private cloud is cloud infrastructure operated solely for a single organization, whether managed internally or by a third-party, and hosted either internally or externally.

PUBLIC CLOUD

A cloud is called a "public cloud" when the services are rendered over a network that is open for public use. Public cloud services may be free.

HYBRID CLOUD

Hybrid cloud is a composition of two or more clouds (private, community or public) that remain distinct entities but are bound together, offering the benefits of multiple deployment models. Hybrid cloud service as a cloud computing service that is composed of some combination of private, public and community cloud services, from different service providers.

IV. SYSTEM ARCHITECTURE

A system architecture or systems architecture is the conceptual design that defines the structure and/or behavior of a system shown in Fig 4.1. An architecture description is a formal description of a system, organized in a way that supports reasoning about the structural properties of the system. In cloud environment, users can upload and download files. First, the user has to register as a member to the cloud. While registering the user has to enter the default attributes like name, password, email id, phone number, city, state and country. Then the user can select any of default attributes which would be asked while downloading or renewing the policy. Then the user has to enter some user defined attributes of their own choice. These attributes also have to be given while downloading files and renewing policy. The next operation is to upload files to cloud. For that we have to choose the file to be uploaded. The secret key is to be generated for authentication purpose. Policy file is generated and encrypted simultaneously. The keys are split by Shamir's key sharing technique. For downloading the file, the user has to send request to the KM and also should provide the user and default attributes. After checking for the valid user, the KM will send the key for decrypting the file to the user. The user can also revoke and renew the existing policy. Revocation is the process of removing the existing files. For revoking policy, the user has to send request to cloud, and the cloud will revoke the selected files. For renewal of files, the user has to provide the old attributes as well the new attributes to the cloud.

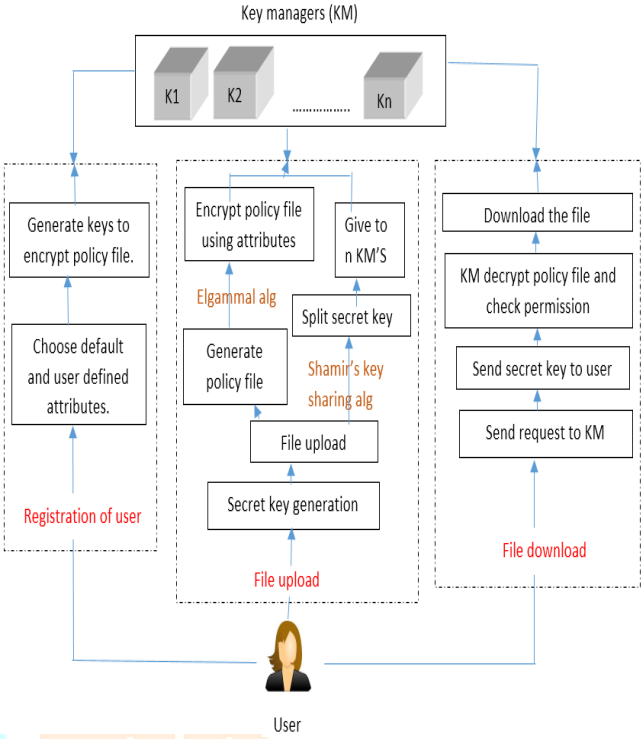

Fig 4.1System architecture

V. METHODS USED FOR IMPLEMENTATION

Methods used for implementation are ,

5.1. executeQuery():
Method used to access values backend and perform operations in frontend.

5.2. executeUpdate()
Method used to access values frontend and perform operations in backend.

5.3. readObject()
Method used to read the object files.

5.4. writeObject()
Method used to write the object files.

5.5. accept()
Method used to accept the client request.

5.6. public static Connection getConnection()
This method is used to provide database connection by creating class name, driver manager and jdbc connection.

5.7. While making a **JDBC connection** we go through the following steps :

a) Register the database driver by using
Class.forName(\" driver class for that specific database\")

b) Create a database connection using
Connectioncon= DriverManager.getConnection(url,username,password)

c) Create a query using
Statement stmt = Connection.Statement("select * from TABLE NAME\")

d) Exceute the query
stmt.exceuteUpdate()

5.8. getAddr():
It is the method used to return the IP address of that specified site.

Input: File name that includes the IP address of the site.

Output: IP address of that site taken from the file.

5.9. request.getParameter(variableName)
It determines that the value of the attribute "txtUserName" of the html form field is assigning to the String type variable "variableName".

5.10. request.getRequestDispatcher()
The RequestDispatcher method is used to forward or include response of a resource in a servlet.

5.11. Clear():
Method Clear the text area of the register form.

5.12. policyKey(String username)
This method is used to get the policy key.

Input: User name should be given as input.

Output: It will return the key information.

5.13. createKey(String user)

This method is used to generate the secret key for authentication.
Input: The user name is given as input
Output: The secret key will be stored in the variable result2.
5.14. updatepolicy(String name, String filename)
This method is used to renew the existing policy.
Input: The user name and the file name should be given as input.
Output: The existing policy will be removed and new policy is updated.
5.15. encrypt(String mess,String passWord1)
This method encrypt the given message to a different form.
Input: The message to be encrypted and the password is given as input.
Output: Encrypted text is displayed.
5.16. decrypt(String cipher,String passWord1)
This method decrypt the encrypted cipher to the original form.
Input: The cipher to be decrypted and the password is given as input.
Output: Decryption is performed and original message is displayed.

VI. SCREEN SHOTS TO EXPLAIN THE PROCESS

The various process involved in cloud security system is shown through screenshots from Fig. 6.1 to Fig. 6.20

Fig. 6.1 Cloud

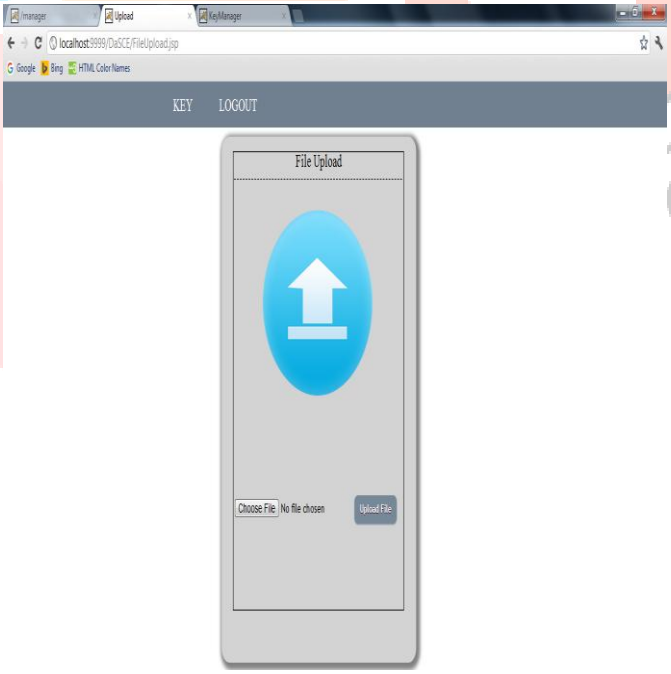

Fig. 6.2 File Upload

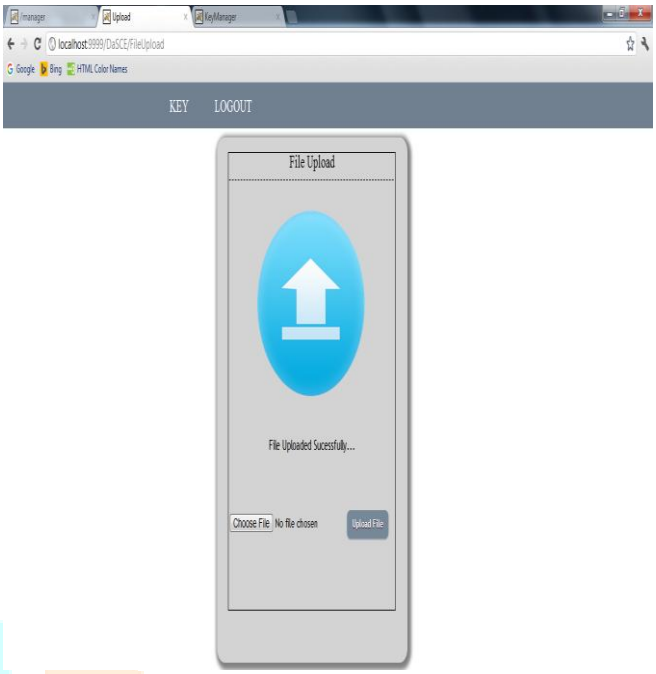

Fig. 6.3 File Upload Completed

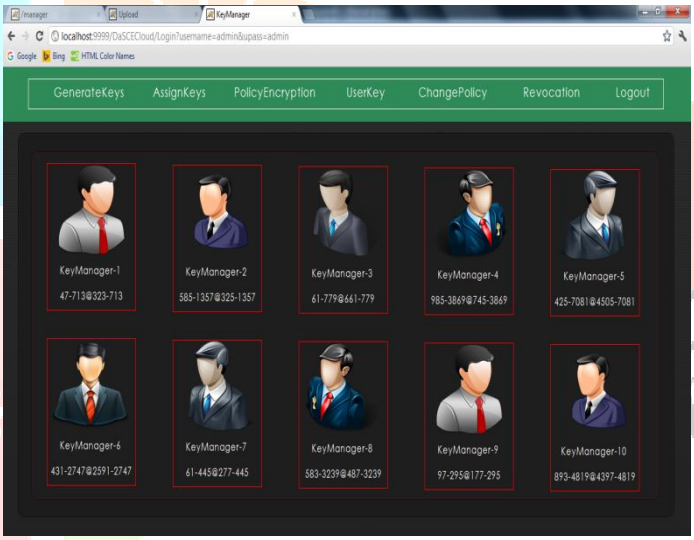

Fig. 6.4 Policy File Encryption Customer selection

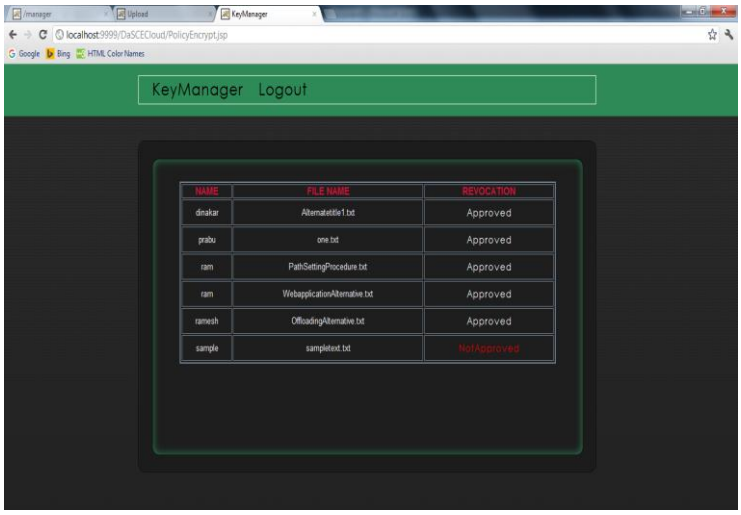

Fig. 6.5 Policy File Encryption

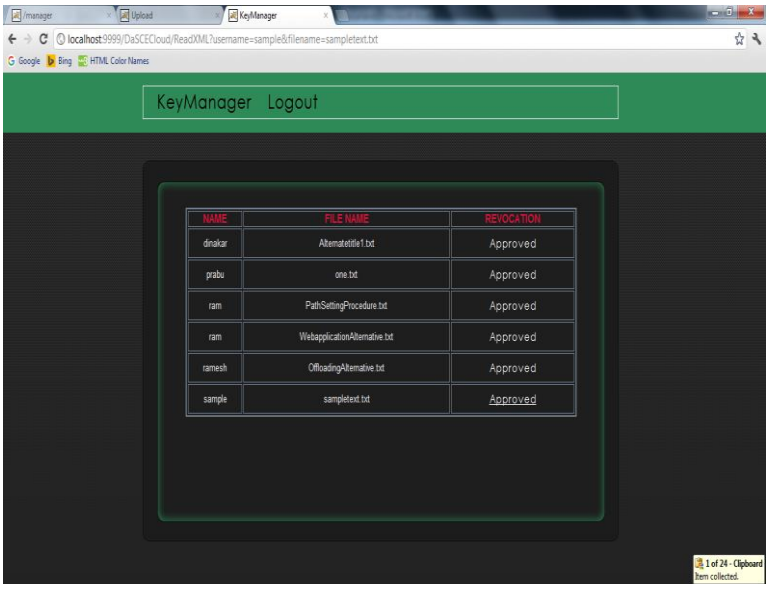

Fig. 6.6 Policy File Encryption Approval

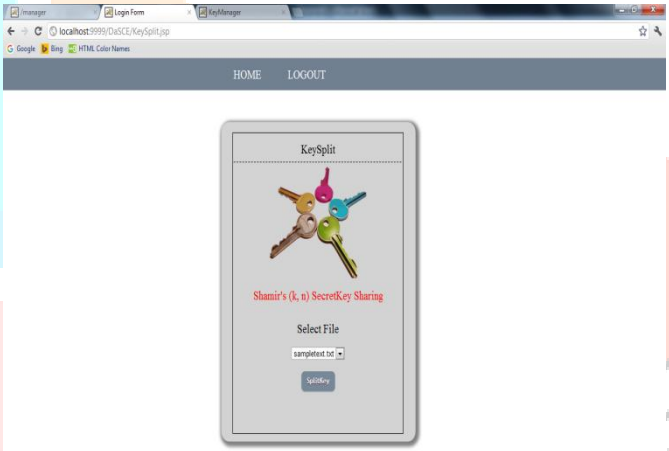

Fig. 6.7 Key Splliting Name Selection

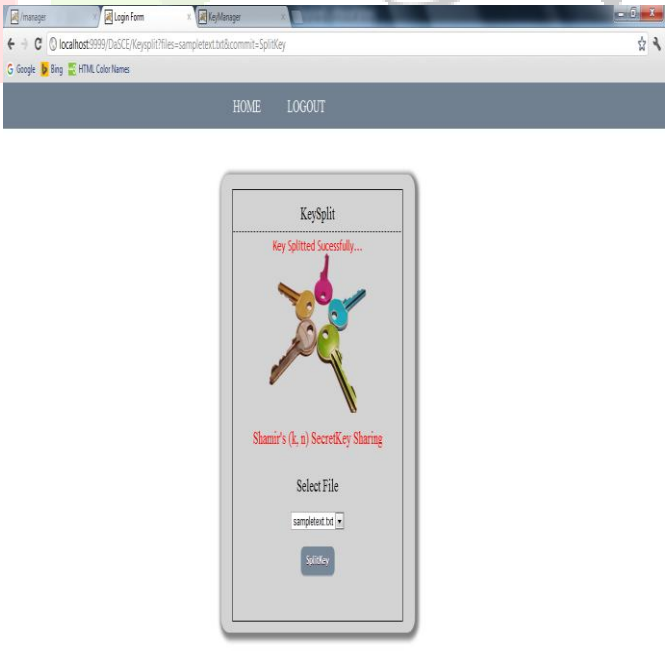

Fig. 6.8 Key Splliting File Selection

Fig. 6.9 Key Splliting File Download

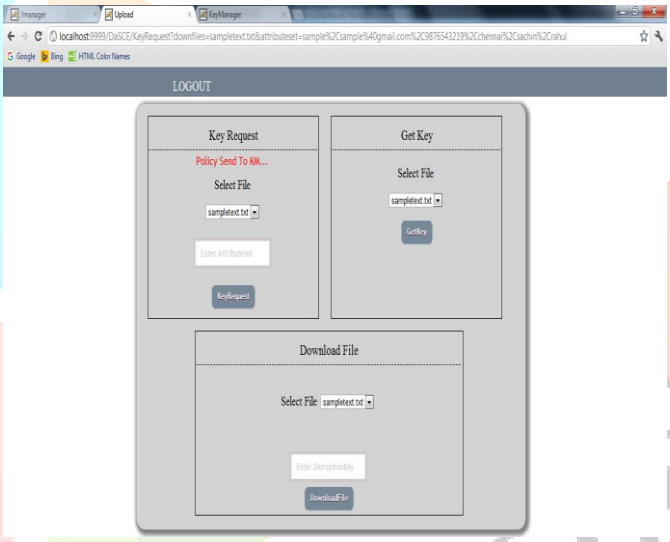

Fig. 6.10 Send Key Request To KeyManager

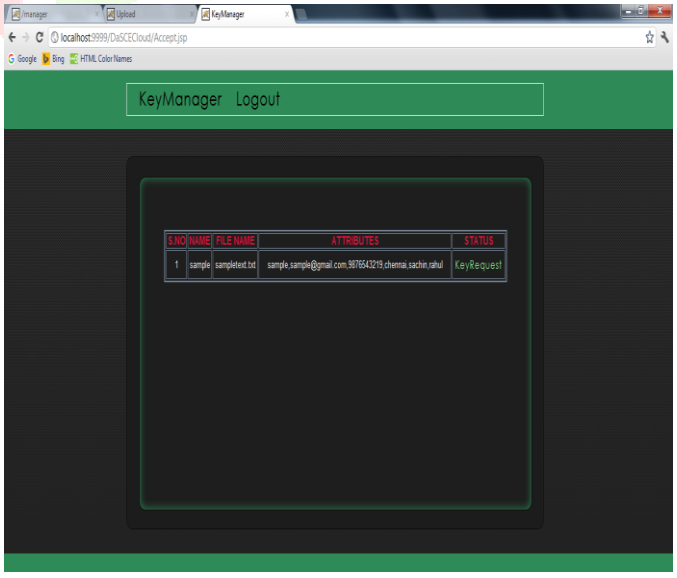

Fig. 6.11 Generate Keys For Requested User

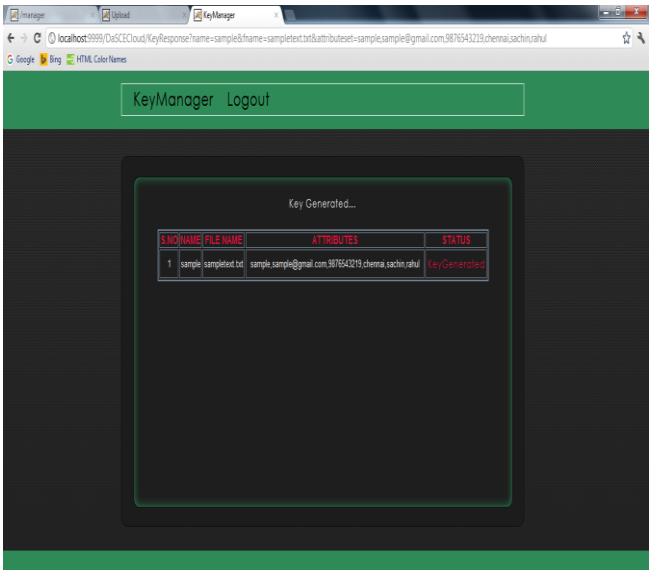

Fig 6.12 Key Selection

Fig. 6.13 Get Key From KM

Fig. 6.14 File Selection for Key From KM

Fig. 6.15 File Selection for Key From KM

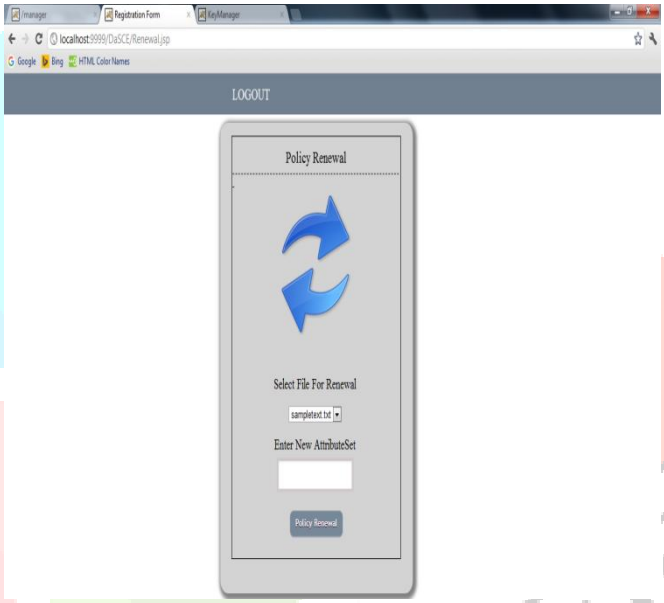

Fig. 6.16 User Disclaimer

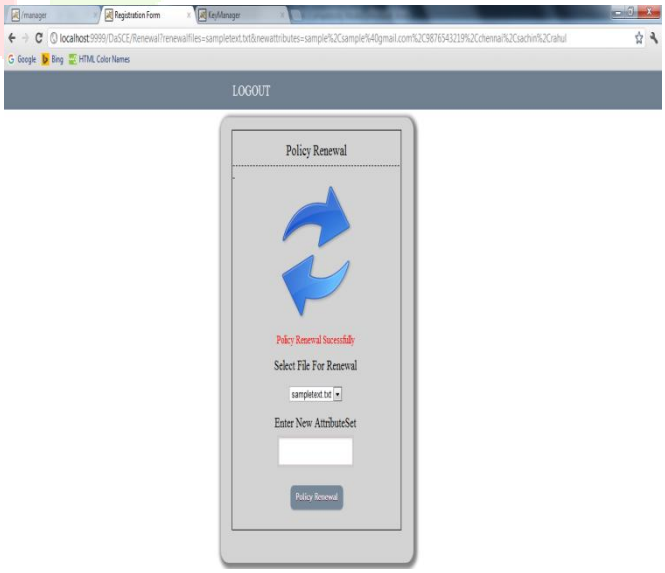

Fig.6.17 Policy Renewal

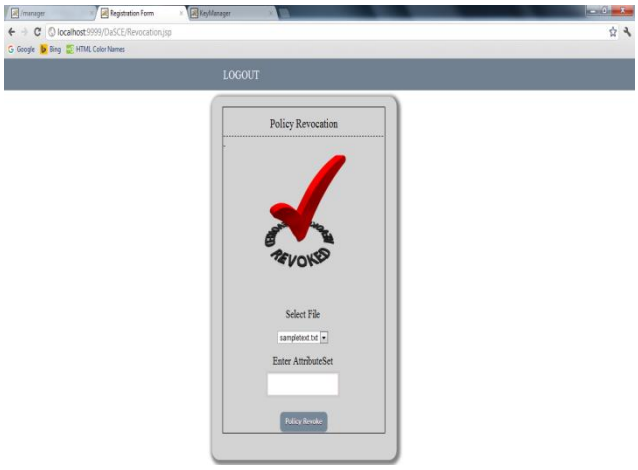

Fig. 6.18 Policy Revocation

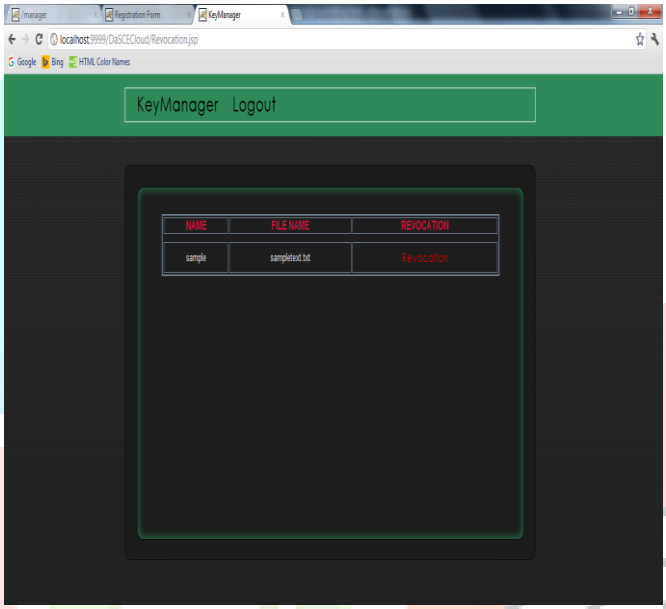

Fig. 6.19 Policy Revocation Customer Selection

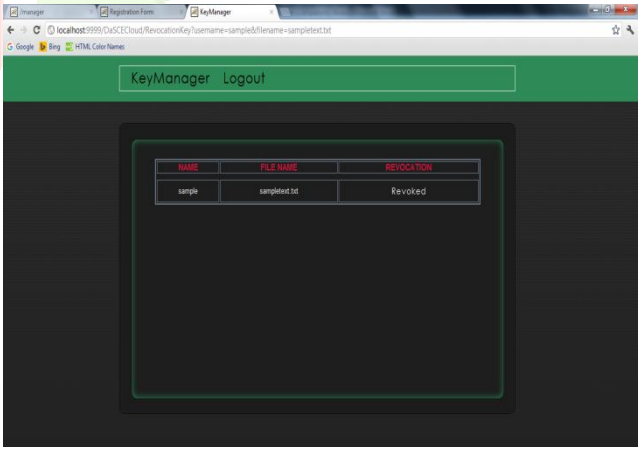

Fig. 6.20 Policy Revocation Sucessful

VI. CONCLUSION

Cloud is a collection of resources called as "resource pool". Users hire resources based on "pay-as-you-use" strategy. But security for data is the major challenge in cloud. This methodology ensures data confidentiality at a cloud infrastructure, as long as it is in use by the client. It also assures that data gets deleted and becomes unrecoverable after the user deletes it from the cloud. Enforcing access control to both data and key through validity of policies and mutual authentication between client and key managers, and client and cloud. Diffie-Hellman key exchange algorithm is used for mutual authentication of parties. Successful authentication and secret key establishment results in access to asymmetric keys that are used in subsequent cryptographic operations. The key management was accomplished using (k, n) threshold secret sharing mechanism. On revocation of policies, access keys are deleted by the *KMs* that result in halting of the access to the data.

REFERENCES

[1] M. Ali, R.Dhamotharan, E. Khan, S. U. Khan, A. V. Vasilakos, K.Li, and A. Y. Zomaya, "SeDaSC: Secure Data Sharing in Clouds," IEEE Systems Journal, 2015.

[2] M. Ali, S. U. Khan, and A. V. Vasilakos, "Security in cloud computing: Opportunities and challenges," Information Sciences, Vol. 305, 2015, pp. 357-383.

[3] A. R. Khan, M. Othman, S. A. Madani, and S. U. Khan, "A survey of mobile cloud computing application models," IEEE Communications Surveys and Tutorials, 2013, 1-21.

[4] A. Juels and A. Opera, "New approaches to security and availability for cloud data," Communications of the ACM, Vol. 56, No. 2, 2013, pp. 64-73.

[5] Y. Tang, P. P. Lee, J. C. S. Lui, and R. Perlman, "Secure Overlay Cloud Storage with Access Control and Assured Deletion," IEEE Transactions on Dependable and Secure Computing, Vol. 9, No. 6, Nov. 2012, pp. 903-916.

[6] H. Lin and W. Tzeng, "A secure erasure code-based cloud storage system with secure data forwarding," IEEE Transactions on Parallel and Distributed Systems, vol. 23, no. 6, June 2012, pp. 995-1003.

[7] Shuhua Wu and Yuefei Zhu,"Improved Two-Factor Authenticated Key Exchange Protocol," The International Arab Journal of Information Technology, Vol. 8, No. 4, October 2011, pp. 430-439.

[8] H. Lin and W. Tzeng, "A secure decentralized erasure code for distributed network storage," IEEE Transactions on Parallel and Distributed Systems, vol. 21, no. 11, Nov. 2010, pp. 1586-1594.

[9] H. Takabi, J. B. D. Joshi, and G. J. Ahn, "Security and privacy challenges in cloud computing environments," IEEE Security and Privacy, Vol. 8, No. 6, 2010,pp. 24-31.

[10] S. Kamara and K. Lauter, "Cryptographic cloud storage," Financial Cryptography and Data Security, Springer Berlin Heidelberg, 2010, pp. 136-149.

[11] M. Kaufman,"Data security in the world of cloud computing," IEEE Security and Privacy, Vol. 7, No. 4, 2009, pp. 61-64.

[12] A. Yun, C. Shi, and Y. Kim, "On protecting integrity and confidentiality of cryptographic file system for outscored storage," Proceedings of 2009 ACM workshop on cloud computing security CCSA'09, pp. 67-76, 2009.