Chapter 1

INTRODUCTION

This chapter introduces the basic concepts of cloud computing including its characteristics, service models, deployment models, and security challenges which need to be addressed. It also presents the problem statement, motivation and scope of work. It finally ends with a brief description of how this thesis is organized.

1.1 Cloud Computing

A cloud can be defined as a pool of virtual computing resources. The resources are configurable in nature and are available in the form of storage, services, networks, servers or applications etc. These resources are wrapped up and are made available to the cloud users in the form of services which can be accessed remotely. Cloud computing is a model which provides an on-demand access through internet to these sharable resources [1]. It enables the end user to make use of the above mentioned resources without even having any knowledge of their whereabouts.

The cloud service provider (CSP) is responsible for making the services provided by cloud reach the end user via some interface. The access to the resources can be provisioned or released dynamically with the nominal interaction of service provider. Granting and revoking access to the pool of resources dynamically is one of the essential features of cloud computing.

Billing models of cloud is generally similar to those of public utilities. One of the benefits of cloud computing is its pay-for- use model in which the users need to pay only for the services they use.

The cloud computing model comprises of five essential characteristics, four deployment models and three service models as depicted in the Figure 1.

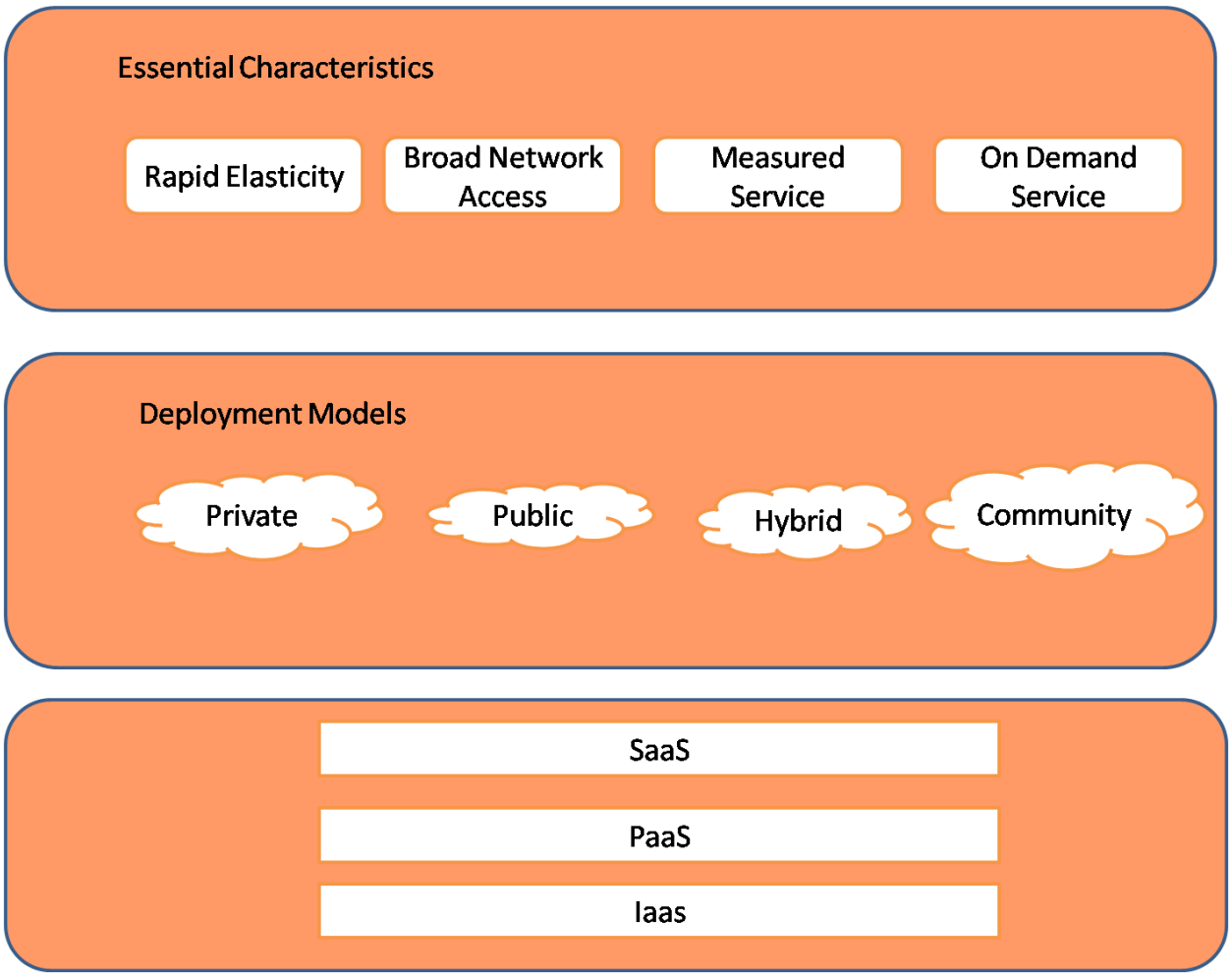

Fig. 1: Cloud Computing Model [2]

1.2 Essential Features

The NIST has defined four essential features of cloud. These are explained in the following section.

1.2.1 On-demand self-service

The user of cloud must be given access to the cloud services as per his requirements. Provisioning access to any of the services should be performed automatically without any need of human interaction.

1.2.2 Broad Network Access

Cloud users belong to varying domains. Different users support different platforms giving rise to a heterogeneous system model. The access provided to the various services should be according to some standard mechanisms so as to fulfill the needs of users with varying environments.

1.2.3 Resource Pooling

The computing resources are stored or pooled over the cloud so as to support multi-tenant model. Access rights to these resources are assigned and reassigned dynamically depending upon their availability. Location transparency is also ensured. Clients do not have any information regarding the exact physical location of these resources but they might possess some information at a certain level of abstraction.

1.2.4 Rapid Elasticity

The services provided by the cloud are provisioned and released elastically to ensure scalability of the system. Total services available generally seem to be unlimited and are provided in the requested amount of quantity.

1.2.5 Measured Service

It is the responsibility of cloud to control and optimize resource allocation so that each user gets his fare share of the resources. Cloud service providers are also required to monitor, require and control the usage of resources. Users are granted access to the resources based on their subscription and they are charged according to the amount in which the resources have been used by them.

1.3 Cloud Deployment Models

There are four deployment models according to NIST's definition of cloud. These models can be put to use as per the needs of an organization. The deployment models are shown in Figure. 2.

1.3.1 Private Cloud

Private cloud infrastructures are meant to be cater an organization exclusively. It does support multiple consumers. It could be managed either by the organization itself or can be

contracted to a trusted third party and can be hosted internally or externally. Setting up a private cloud requires a noteworthy level and degree of engagement to virtualize the business environment. The decisions regarding the existing resources are also required to be re-evaluated. Successful deployment is likely to be proved more beneficial in business growth. Security issues will not be of major concern if deployed carefully.

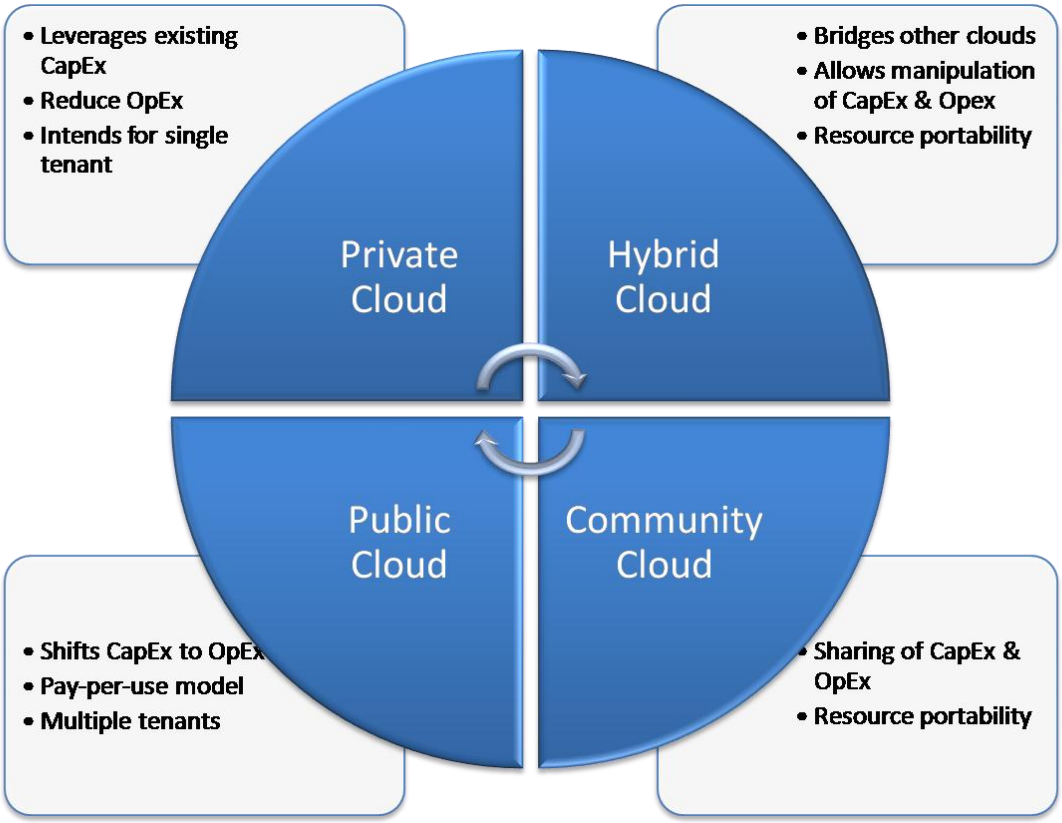

Fig. 2: Cloud Deployment Models [2]

1.3.2 Community Cloud

Community cloud infrastructure provides services to a group of users who belong to a community with common interests and goals. The example includes compliance consideration. This infrastructure can be maintained and operated by a single or many organizations or can be

contracted to third party. It supports fewer people than public cloud but more than private cloud, thereby saving some costs.

1.3.3 Public Cloud

Public cloud is open for general public use. It can be managed and operated by any organization be it private or government. The services provided by them are either free or based on pay per use. Examples of public cloud are Google, Amazon, AWS. It support comparatively more number of users and offer them services via internet.

1.3.4 Hybrid Cloud

It is a combination of any two or more types of cloud models which are compelled together but each model remains independent in itself. This model does not rely on internet connectivity for offering services. Also, it offers high degree of fault tolerance. The architecture of hybrid cloud requires both on-premises resources (online resources) and off-site (remote) server-based cloud infrastructure.

Hybrid clouds provide the flexibility of in-house applications along with scalability but lacks in security, certainty and flexibility of in-house applications.

1.4 Cloud Service Models

Cloud provides services to its clients on the basis of its three service models. These models are Iaas, Paas, and Saas with Iaas being the most fundamental model. Other two modes are built upon Iaas model while abstracting its details. The service models are shown in Figure. 3.

1.4.1 Infrastructure as a Service (IaaS)

It is the most elementary service model. All the virtual computing resources are hosted by a third

Party and access to them is given via internet. The resources can be anything like network, storage, processing. Delegating storage to the cloud server is very common these days. Many business organizations move their confidential data to the server thereby reducing their storage costs. These resources provide an environment to let the users deploy and run their software. The

virtual machines, offered as service, are run as guests by hypervisor. The users cannot control the infrastructure model but they do have control over the deployed applications.

Examples of IaaS include Amazon Cloud Formation (and underlying services such as Amazon EC2), HP Cloud, Windows Azure Virtual, etc.

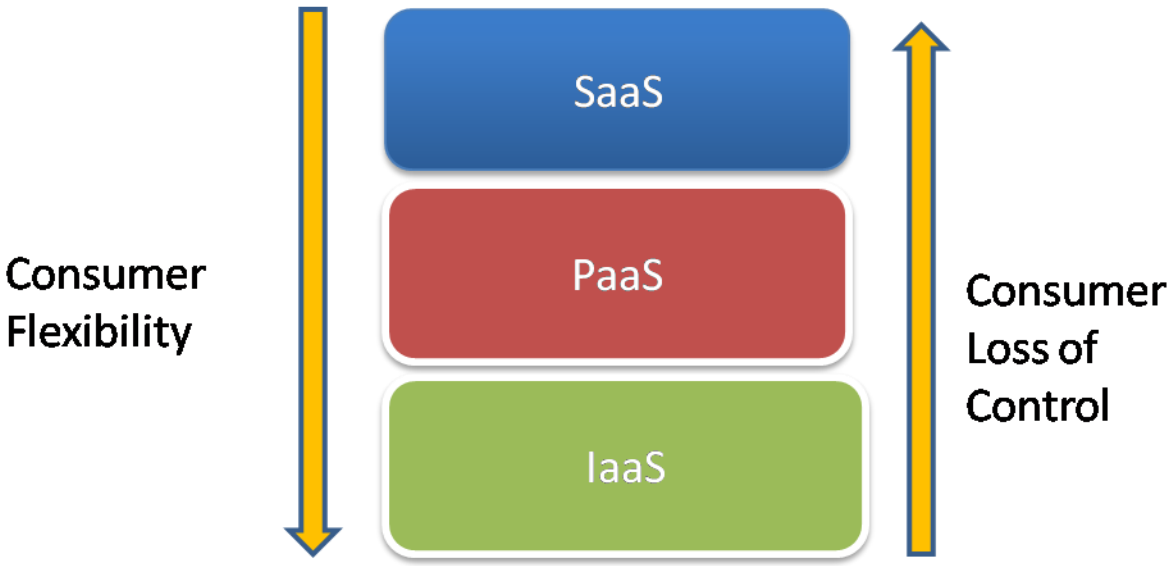

Fig. 3: Cloud Service Models

1.4.2 Platform as a Service (Paas)

In PaaS model, platforms are delivered as services. The platform can be either of web server, database, operating systems, and execution environments of programming languages. The application developers can build their applications over these platforms without having to worry about buying the required platforms hence reducing clients' expenses. Storage provided by some clouds can even be scaled automatically according to the needs of users so users need not worry about allocating resources manually.

Examples of PaaS : Amazon Elastic Beanstalk, Cloud Foundry, Google App Engine, Windows Azure Compute etc.

1.4.3 Software as a service (SaaS)

In SaaS model, applications are deployed onto the cloud. Clients are given access to these application software as its service. End users need not worry about the underlying infrastructure architecture or platforms on which the applications server. These applications can be accessed through any interface like web browsers or program interface. The users have no control over the underlying infrastructure like operating system, servers, storage but they might be given some support for user-specific functions.

Examples of SaaS include: Google Apps, Microsoft Office 365, and Onlive.

1.5 Security Challenges

Cloud computing is being heavily used by a large number of organizations for out-sourcing their critical data. Along with the opportunities it offers, a number of security challenges come into existence. Security of cloud can be measured against the following parameters[3].

1.5.1 Confidentiality

One of the services provided by cloud is storage. Organizations are moving a large amount of their data storage to cloud so as to mitigate their storage cost. The data outsourced to cloud contains critical information whose confidentiality must be ensured. To provide confidentiality data is generally stored in encrypted form. To ensure confidentiality data must be protected from unauthorized access.

1.5.2 Integrity

Integrity can be compromised if data stored on cloud is given access to unauthorized users. The illicit user can modify, replay or delay the data. This could lead to a huge business loss. There is a possibility of a number of domains which attracts the attackers. Depending upon the type of attack and service model adopted, integrity varies. In SaaS model, data stored in cloud needs to be protected. In other two models, underlying architecture, platform and configuration files need to be protected. If integrity of these files is compromised then not only these two models suffer but also the services provided by them suffer.

1.5.3 Availability

Availability is compromised if the server fails to deliver its services as per the expectations. This could happen if the sever gets suspended or spoofed. One of the common targets of availability attacks includes attack on Domain Name System (DNS). This is due to the fact that broad network access is one of the essential features of cloud computing.

1.5.4 Security Management

One of the key features of cloud computing is its on-demand delivery. The needs and requests of users keep on changing; therefore there must be a mechanism to address these frequent changes with immediate effect. As the scope of cloud computing increases, management of security issues becomes more complex.

1.6 Access control

The security challenges mentioned in the above section come into existence because of the unauthorized access by illicit users. The information stored on cloud is highly confidential and therefore access to them must be given only by the legitimate users.

Access control is a mechanism of granting/revoking rights to/from users in order to ensure confidentiality and integrity of the critical data stored on the cloud. The first step in the authorization process is login to the system via some safe authentication mechanism. Access control mechanism controls the activities that a user is allowed to perform. The access rights include read access, write access, delete access etc. Once the user is authenticated he is allowed to access the data stored on the server according to his access rights.

It is the responsibility of the data owner to grant access rights according to its policies based on certain policies. Once it generates the rights, it becomes the responsibility of the server to ensure that only the authorized person is granted access.

For a successful cloud model, all the above mentioned security challenges must be addressed. Access control mechanism, if deployed appropriately, can be of great advantage in meeting the security goals.

1.7 Performance Evaluation

Access control mechanisms are validated against the parameters like security, dynamic data sharing, fine-grained access control, accountability and scalability.

1.7.1 Security

Security is concerned with confidentiality and integrity of the data stored on the cloud server. To enforce security data must be encrypted using a robust cipher. The encryption of data protects it from malicious server. Access control mechanism must ensure that only the authorized users get to access the decrypted data and only they can decrypt it. The data outsourced to cloud must not be tempered with. There must be a mechanism to ensure that the data has not been tempered with.

1.7.2 Dynamic Data Sharing

Dynamic data sharing means access control mechanism must not be restricted by the number of data sharers. The cipher text stored on cloud must be independent of the data sharers. This ensures the scalability of the entire model. If the system does not support dynamic data sharing, each a time new user enters or leaves the system, the cipher text would need to be updated. To ensure scalability of the system, access control mechanism must support dynamic data sharing.

1.7.3 Fine Grained Access Control

Fine grained access control ensures that only the authorized users get to access the resources and that too in the mode for which they have been given access rights. It means that if a client has been granted read access rights then he would not be able to perform anything on resources, but reading.

1.7.4 Accountability

It is required to control and manage the amount of resources being accessed by the consumers. The server must maintain a log of all the records.

1.7.5 Scalability

Access control mechanism must ensure that the cipher text placed on the cloud is independent of the number of data sharers. Any user should be allowed to join or leave the group depending on his needs and this should not affect the data stored on cloud.

1.8 Motivation

Cloud computing is an emerging paradigm with offers a number of computing resources as its service. One of the services provided is storage facility. The cloud model is encouraging a number of organizations to move its storage to a trusted third party cloud server. This helps them in reducing their storage costs which can be put to use for other resources. The data uploaded on server is needed to be shared among the heterogeneous users.

With the storage facilities it provides, there arise a number of security threats. Since the data stored is crucial for the organization, it must be protected against any attack. The confidentiality and integrity of data must be protected.

There is a need of access control mechanism that addresses the security threats in the most efficient manner. The techniques proposed so far are not able to provide the entire essential solutions. All the features have not been attained simultaneously. Also, some techniques are too complex to achieve access control with limited resources.

With this motivation, this thesis aims at proposing an access control mechanism which meets the required security criteria in an optimum possible way. The proposed mechanism achieves access control with the help of authorization tickets while ensuring user's anonymity.

1.9 Problem Statement

This thesis aims at achieving a solution for access control mechanism in cloud while retaining the confidentiality and integrity of data. It ensures that the data is given access only to the trusted users while protecting its access from any malicious server.

In this work, access control is achieved with the help of authorization certificates which is issued by the data owners to its clients [4]. Data owner is responsible for issuing authorization certificates to the data sharers or clients. Data sharer maintains a log of all the certificates issued by him. This record can be later used to modify or revoke the certificate. Brief information about the certificates is also sent to the server. Server uses this information to check whether the certificate is issued by him or not. Only the owner of the certificates is allowed to access the files. The certificate is comprised of various attributes which would be used by the server to validate the client. After receiving the request from the data sharer, client validates the user and depending upon his access rights lets him access the file. The server also maintains a record of certificates revoked.

Problem of the thesis can be stated as:

Achieving access control in cloud which maintains the confidentiality and integrity of files stored on the cloud server. While doing so anonymity of the client is also ensured.

1.10 Scope of Work

The thesis provides a solution to the access control problem for cloud. The access control, as proposed in the thesis is achieved via authorization permits, also referred to as authorization certificates. The implementation is divided into three modules, one for each, data owner, data sharer and cloud server. Data owner can be thought of as an organization or an individual who moves his storage to the cloud. He has the authority to grant access rights to the clients or data sharers. He issues permits to the clients. Only the clients with valid authorization permit are allowed to access the storage. His access rights can be revoked anytime by the owner. Also, the identity of the data sharer is kept from the cloud server. Finally, the proposed scheme is checked against all the essential features of an ideal access control mechanism.

Scope of work can be summarized as:

- Design an interface for data owner, using which he uploads his files on the server and revokes the access rights of clients.
- Design an interface for client, using which he requests for access permit.

- Design the format of the certificate which must contain all the essential attributes required for his validation.
- Issue the authorization certificates, its validation on the client side.
- Requests the server to access files using certificate. Server performs validation check on the client.
- Evaluate the client based on his attributes and decides whether to let the client access the file or not.

1.11 Thesis Organization

The remaining chapters of the thesis are organized as follows:

Chapter 2 provides a detailed description of access control techniques put forth so for. It gives an insight to the advantages and disadvantages of the already existing techniques.

Chapter 3 describes security architecture of cloud computing.

Chapter 4 presents the proposed mechanism and framework in detail.

Chapter 5 shows the implementation and results.

Chapter 6 provides the security analysis of the proposed algorithm.

Chapter 7 concludes the thesis along with the future work.

Chapter 2

Literature Survey

This chapter provides a detailed description of the techniques put forth so far. Access control in cloud has fascinated a number of researchers. There exist a number of techniques for achieving access control in cloud. Each technique has some advantages and disadvantages. The objective of this chapter is to define the existing mechanisms in detail and provide a comparison among them.

The existing techniques can be categorized broadly into three categories.

2.1 Distributing Decryption Key Directly

The simplest way of achieving access control is to directly distribute the decryption key among the authorized users. This method is known as Distributing Decryption Key Directly (**DDKD**). In this technique, the data owner encrypts the file using a secret key and uploads the encrypted file on to the server. The confidentiality of file is maintained since the file is stored in encrypted form. Now the data owner distributes the key among the legitimate users. User can fetch the stored file from the server. If he has the authorization rights (i.e. if he possesses the key) he can decrypt it using the shared secret key.

This mechanism is simple to implement but suffers from many disadvantages. The server does not check the authenticity of the user before rendering the file. Although the file is encrypted it should not be given to any random user. If some point of time, if the key gets compromised confidentiality no longer remains intact.

Also, there is no way of ensuring the integrity of data. A malicious server can modify the data anytime and the clients will never get to know about this. This issue can be resolved by storing the message digest along with the encrypted file.

Another issue that arises in this technique is the revocation of access rights. If the owner wants to revoke the access right from a user, he has to generate a new key, encrypt the file using the new

key, remove the previous version of file from cloud, upload the new file and distribute the new key among the users all over again i.e. repeating the entire process again.

Repeating the entire process again is not only time consuming, it restricts the scalability of the system. It does not support dynamic data sharing. This means every time the owner wants to remove a sharer, the system has to be rebuilt again which is neither convenient nor feasible for a large system.

2.2 Key-policy attribute based encryption

Attribute Based Encryption technique lets the user decrypts the file as long as the uses possess the required set of attributes. This technique is able to achieve fine grained access control. **Key-policy attribute based encryption (KP-ABE)** was introduced by Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters[5]. In this technique the cipher text is associated with a number of attributes and the private key is associated with an access structure of tree format.

For example, access structure associated with the key of user1 is a AND b. Access structure associated with user2 is b AND c. The cipher text is labeled with the attribute b. Given this scenario, neither of the two users would be able to decrypt the cipher text.

A user possesses the key associated with the access structure A, he would be able to derive the key for access structure B only if, B is more restrictive than A.

Access tree

The internal nodes of tree represent threshold gates (AND, OR). Associated with each internal node x are two values (threshold value k_x and no. of children num_x).

$$0 < k \leq num$$

Value of k_x is 1 if the threshold gate is OR else k_x is equal to the number of child nodes.

Leaf nodes represent the attributes. Associated with them is threshold value k. Value of k is 1 if the attribute represented by it belongs to the set of attributes.

User is able to decrypt to decrypt the cipher text if and only if the attributes associated with the cipher text satisfies the access structure of key. Consider access structure associated with the client's key (*t)* (Figure 3).

If the attribute associated with the cipher text is X, the user would not be able to decrypt it. If the attributes associated with cipher text are X and Z, the user would decrypt it.

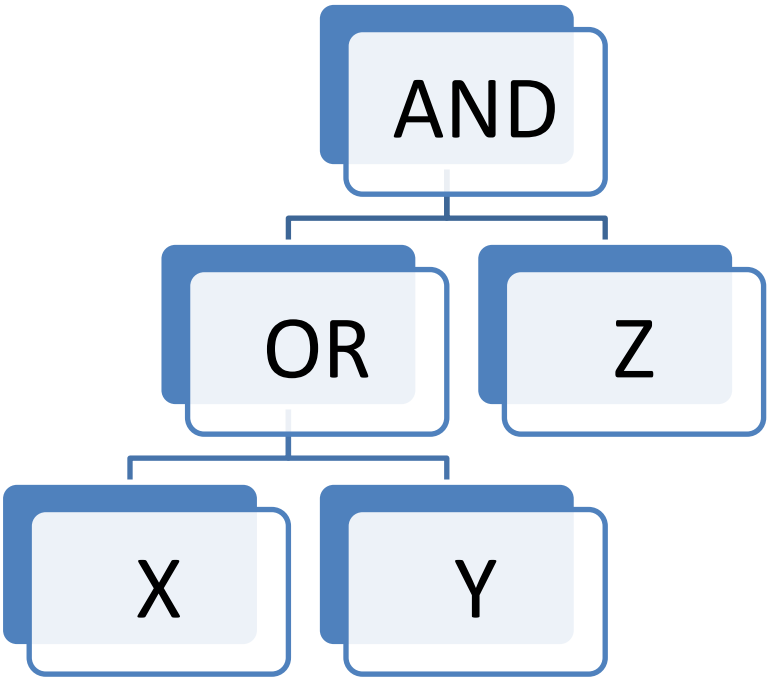

Fig. 4: Example of access structure

The access tree checks if it satisfies the attributes associated with the cipher text. The leaf nodes returns 1 if the attribute represented by it belongs to the set of attributes associated with the cipher text. Internal nodes return 1 if at least k children return 1 else it returns 0.

For a set of attributes, Υ access tree returns 1 if the access structure is met by the requirements. In this case user is able to decrypt the cipher text.

This technique supports fine grained access control mechanism but is difficult to implement. It involves complex bilinear pairing. Access revocation is also difficult to implement as it would require re-encryption. Therefore, this does not support dynamic data sharing.

2.3 Cipher text-policy attribute based encryption

It lets only those users decrypt files, who possess some credentials .The technique was proposed by John Bethencourt, Amit Sahai and Brent Waters. In this technique, the private key of data owner is associated with a number of attributes; the access structure is enforced on the cipher text by the encrypter[6]. The users who possess the required set of attributes would be able to decrypt it.

Access structure is represented in the form of access tree as described in the above section. Internal nodes are represented by threshold gates and leaves are associated with the attributes. In this technique encryption algorithm takes into consideration the access structure which is to be imposed on the cipher text. This is how it controls the access of cipher text only to the legitimate user. The key generation function considers the set of attributes that describes the key and generates the decryption key(S). The decryption algorithm takes the cipher text and the decryption key as its input. If the key satisfies the access structure enforced on the cipher text, it would decrypt the cipher text.

It is the responsibility of the data owner to represent policy in the form of access structure. The key is then associated with a number of attributes. The policy decided by the data owner must be satisfied by the key in order to decrypt the files.

For instance, consider the policy: (Names: x OR Y) AND (age>25). An access tree corresponding to the policy is taken into consideration by the encrypting mechanism. Only the clients whose attributes satisfy this would be able to decrypt it.

The technique is very close to the key-policy technique described above. It does prevents collusion attack. Although the technique is able to achieve fine grained access control, it is difficult to implement because of the complex bilinear pairing.

2.4 Profile Based Access Control

In profile based access control mechanism, the cloud service provider (CSP) is responsible for creating profiles of users[7]. The profile of a user reflects its characteristics. Characteristics like type of data that the user wants to access and for how long.

Cloud service provider maintains a list of the user profiles in the following format.

Table 1: List of Profiles

USRGRP	USRID	Data	USRGRP Date & Time	USR Date & Time
1	001 030 123 442	P	02-11-12 10:50	04-11-12, 11:00 22-01-13, 10:15 08-12-12, 11:10 19-04-13, 06:00
2	119 231 134 191	Q	14-08-13 11:10	14-08-13, 1400 27-11-13, 1810 03-11-13, 01:00 08-12-14, 01:10

CSP groups the users based on the type of operation they performs. If a request comes to access a file, CSP first checks the list of profiles. If the user is in the list, he is given access to the file. If the user is new, a new profile for him will be created based on the authorization policies set up by the data owner. The new user first sends the authorization request to the server. The server collects the relevant information and generates the profile. The server then contacts the data owner for his access rights. Server finally records his data rights in its logs and also sends a copy of it in encrypted format to the user.

Two operations can be performed on the list of profiles. These are insertion and deletion. The deletion corresponds to the revocation of access rights.

The described technique does not require the data owner to be online all the time. It also decreases the access time.

2.5 FADE

FADE[8] stands for Fine Grained Access Control Assured Deletion. In this technique, the active files are attached with some access policies. The user is granted access to the files if he satisfies those policies. It also incorporates policy-based file deletion. FADE is a generalization of time-based file deletion. In time-based file deletion the file gets inaccessible after a specified amount of time. The data owner encrypts the file using a secret key. This key is encrypted using a control key. This control key is deleted after some time. Since we don't have the control key, we cannot obtain the key used for file encryption, making the file inaccessible.

In policy-based file deletion, each file is associated with a single or a group of policies. Associated with each policy is a control key. The file is encrypted with a data key. The data key, in turn is encrypted using the control key of policy attached to it. When the policy gets revoked, the control key associated to it is removed leaving the data key inaccessible.

Fade describes two operations on files.

2.5.1 File Upload

File is encrypted using a random key, K. A pair of public-private RSA keys is generated in accordance with the policies associated with the files. The public key is made known to all the entities in the system while the secret key is retained by the key manager. These keys are used as control keys. Data owner then generates another random key, S. It then encrypts the random key K using S. S is further encrypted using public part of the control key. It then uploads the encrypted file, encrypted key and encrypted $(S)^e$.

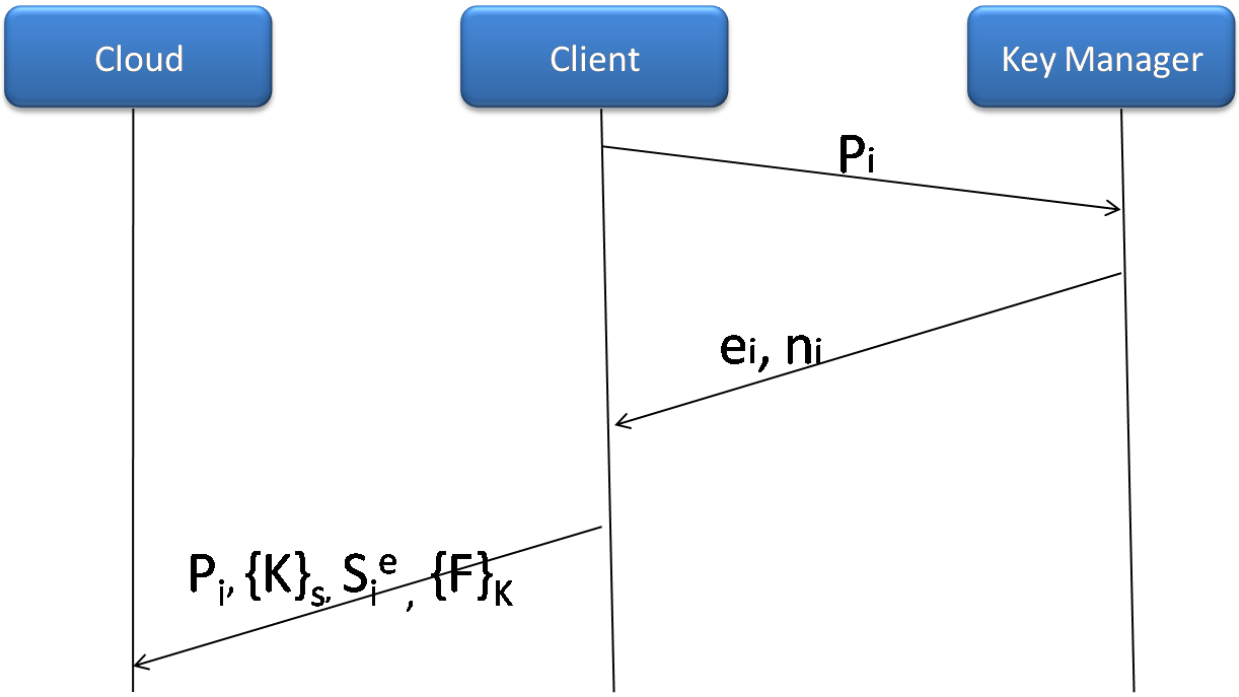

Fig. 5: File upload in FADE

2.5.2 File Download

The client generates a random number R. It calculates $(SR)^e$ and sends it to the key manager. Key manager uses his private key to evaluate *SR*. It sends back *SR* to the client. Client then finds out S. It then decrypts to obtain K using S and finally decrypts the file using K.

To delete the files, key manager removes private part of control key associated with the file access policy. In the absence of control key, data key cannot be obtained and hence file cannotbe decrypted. The file thus becomes inaccessible.

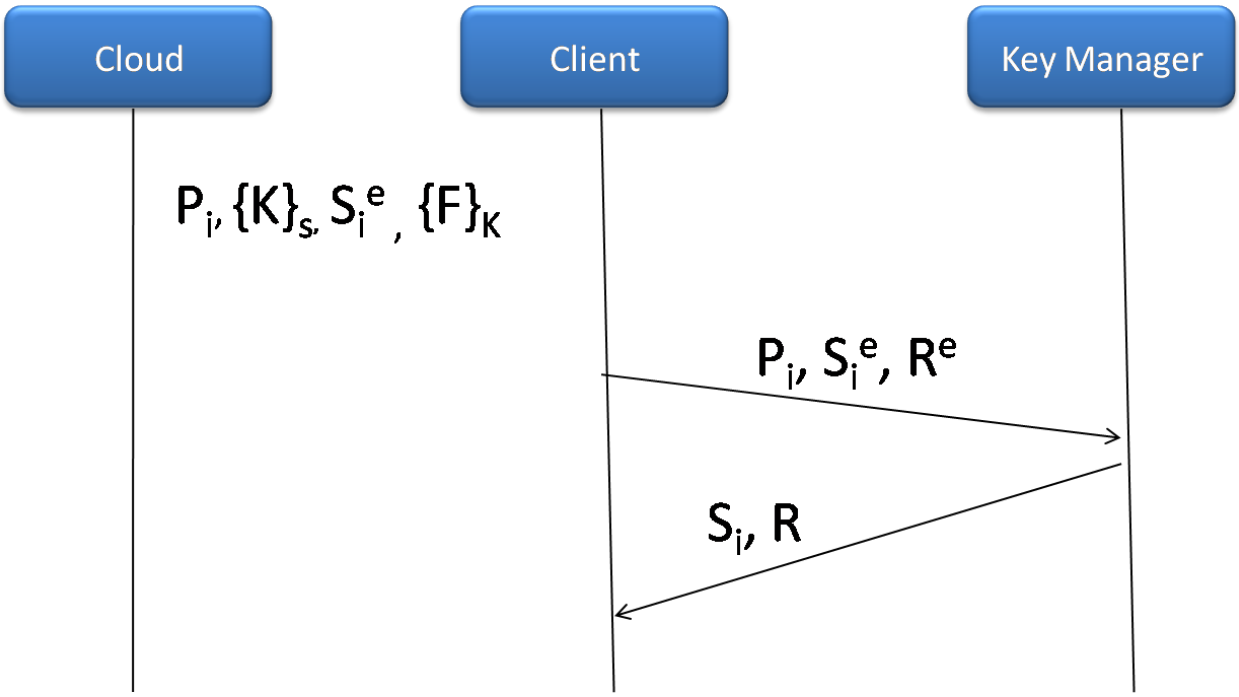

Fig. 6: File Download in FADE

2.6 MTBAC

MTBAC stands for a mutual trust based access control mechanism [9]. This technique is built upon the trust relationship between the users and the cloud. It takes into account the trust degree of users' behavior and trust degree of the cloud. The trust degree of cloud is calculated using bee colony mechanism.

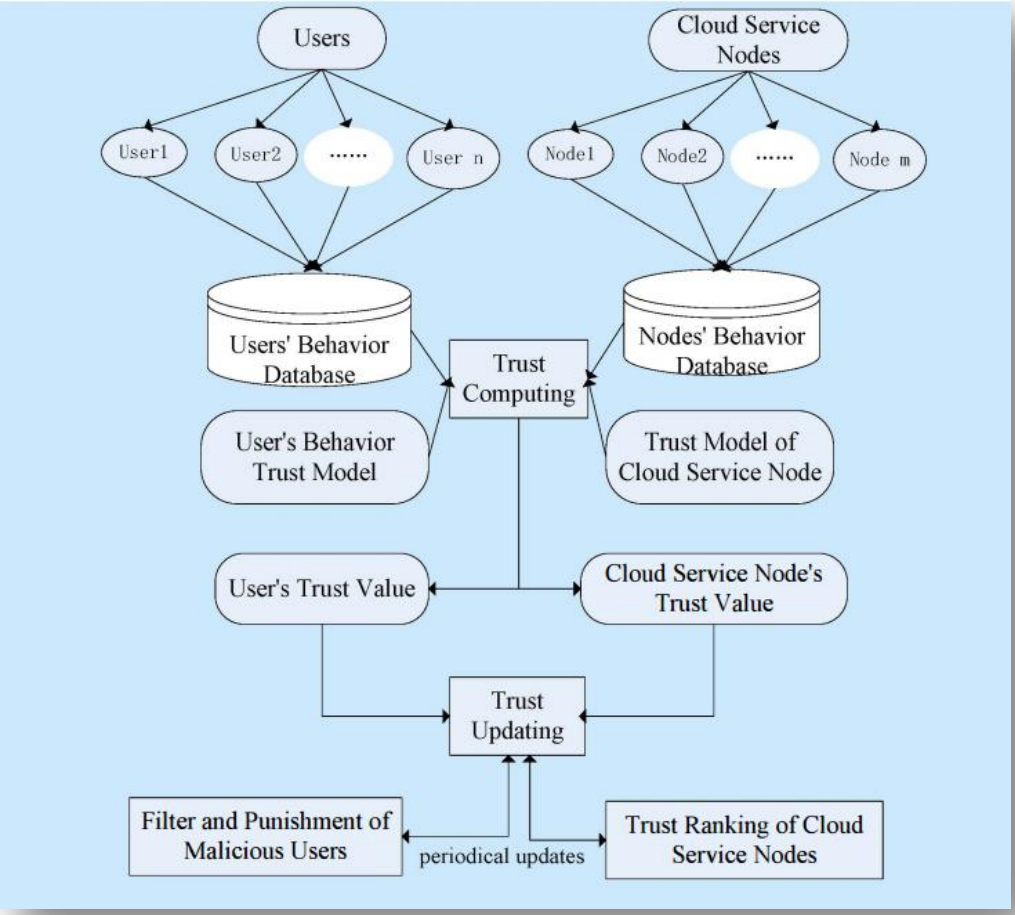

Fig. 7: Mutual trust structure

The algorithm can be summed up as:

Step 1: The user requests to access a resource.

Step 2: Cloud server gathers the information about the characteristics of the user's behavior. It then evaluates the degree of trust of user's behavior. The user's characteristics involve the type of resources user requests, frequency of access, duration for which resource is being accessed. The characteristics can be divided into the categories as shown in Figure: 7.

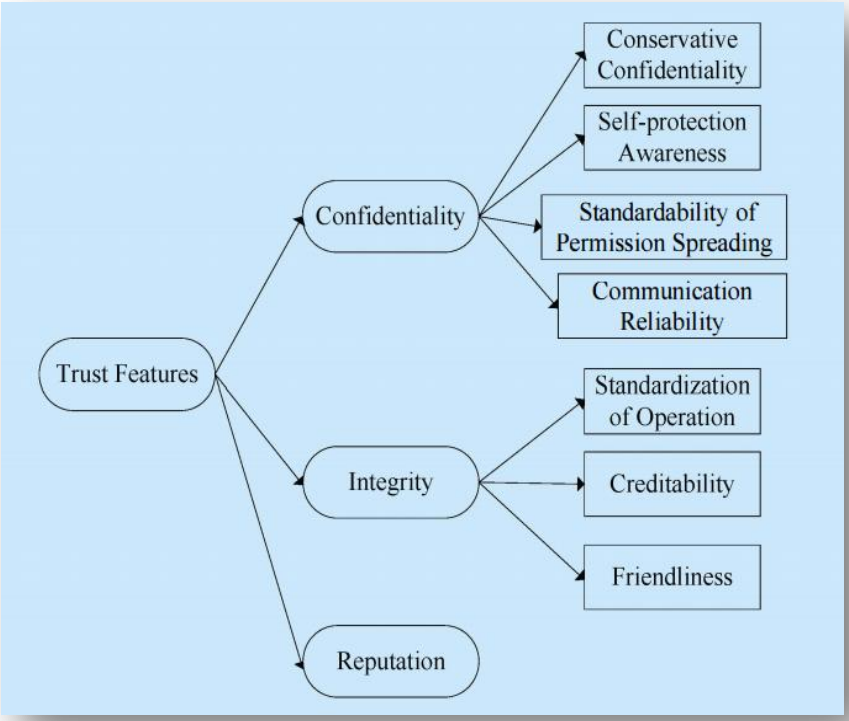

Fig. 8: Trust attributes

Step 3: The trust degree computed in the above step is compared to a threshold value. If it is less than threshold value, go to step 6.

Step 4: Choose the best service node using their trust values.

Step 5: The best chosen node renders the file.

Step 6: Stop.

The technique described above protects the system from malicious users. It improves availability.

2.7 Trust and Role based Access Control

Trust and role based mechanism incorporates trust model in the traditional role based access control [10]. In traditional role based model, organization introduces a number of roles, grants them permissions and assigns them to the users [11]. In trust based model, trust relationship is calculated between client and the cloud and among the clouds itself. Trust degree of user's behavior is evaluated based on the user's characteristics. Trust degree of cloud is evaluated by other clouds depending upon the services it provides to its users. The roles identified by the organizations are associated with the trust values. The algorithm proposed is shown in Figure. 9.

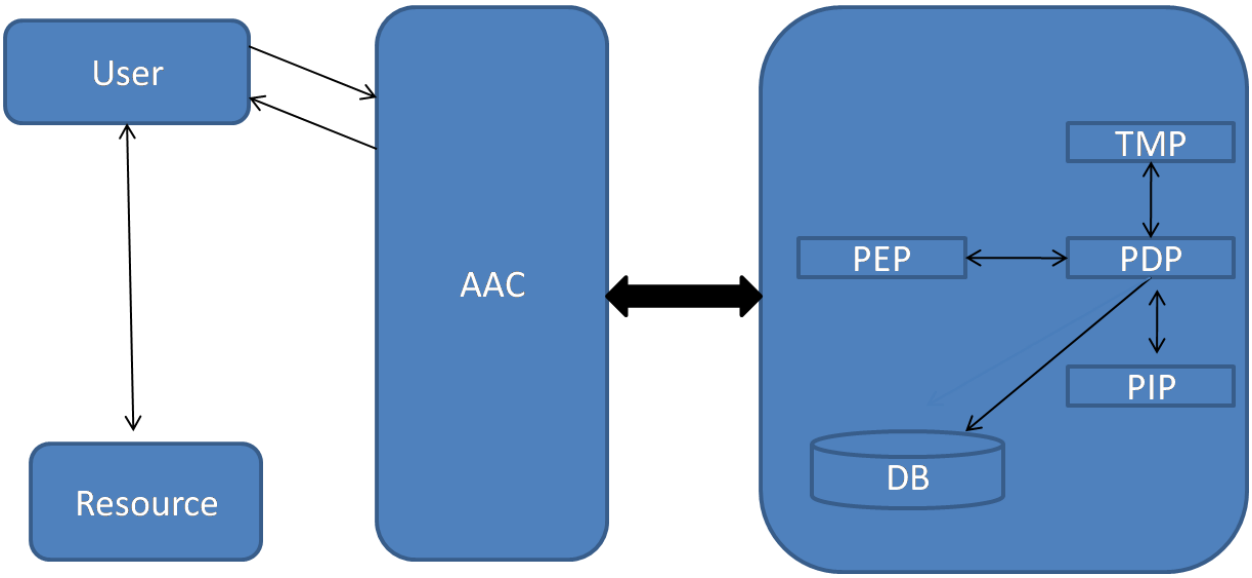

Fig. 9: Framework of trust and role based mechanism

According to the algorithm, user first makes a request to the authorization center. AAC forwards his request further to policy decision module, which decides whether the user should be given access to the resources or not. If the decision is positive, a certificate is issued to the user. User then uses this certificate to make requests. Finally, AAC modifies and stores the trust values.

2.8 Access Control Matrix

Access control matrix is one of the fundamental access control mechanisms. It maintains the access rights in a two-dimensional grid. The column represents users and the row represents resources. The cell corresponding to a user and resources contains the type of access. Example of a typical ACM is shown in table 2.

The traditional ACM method suffers with some disadvantages. Some of them were addressed in [12].

Table 2: Example of ACM

Subject	File1	File2	File3	File4
User1	Read	Write	Own	-
User2	Write	-	Own	-
User3	Own	-	-	Read
User4	Read	Read	Read	Own

They used the concept of data hiding to decrease the response time. Suppose a user makes a request to access a file for which he has no permission. He makes a request to the server. Server checks CAN and reply with the negative response. Meanwhile, the client was doing nothing but waiting for the response. This waiting time can be eliminated using data hiding concept. Consider table 1. We can hide file4 from user1. If he does not know about its existence, it would not make any request to access it, thereby reducing the waiting time.

We can also restrict user from making a request for which is not authorized to. Consider user 3 makes request to access file 4 in write mode. The implementation must be done in such a way that the user is not able to generate the request in unauthorized mode.

2.9 Risk based Access Control

As described in [13], incorporating risk in access control enhances the flexibility and scalability of the system. They introduced three risk components in their research:

- Risk Engine
- Risk quantification Web Services

- Risk Policies

All the techniques described in the above sections are able to achieve access control with varying degree of effectiveness. Some of them are easy to implement but are not promising when it comes to meeting the essential security parameters. Some of them are more effective are too complex to implement.

Therefore, we have proposed an access control mechanism which promises to satisfy all the required security parameters while using the resources optimally and is not too complex to simulate. The technique that we have proposed makes use of authorization certificates to prove their authenticity.

2.10 Summary of the existing mechanisms

Table 3 presents a brief summary of all the techniques that have been put forth in the field of access control mechanism.

Table 3: Access Control Techniques' Summarization

Access Control Technique	Year of Establishment	Key Features
KP-ABE	2006	Associates cipher-text with policies and an access structure with key. Complex bilinear pairing.
CP-ABE	2007	Associates key with a policy and enforces access structure on cipher-text. Involves bilinear pairing.
Trust & Role Based	2011	Incorporates trust degree in the traditional RBAC

FADE	2012	Files are attached with some policies & users are granted access only if they satisfy the policies.
ACM	2012	Makes use of improved version of access control time with reduced response time.
Profile-Based	2013	CSP generates users' profile based on their characteristics & grants access right accordingly.
MTBAC	2014	Takes into account trust degree of user's behavior & trust degree of cloud.
Risk-Based	2014	Takes into consideration risk components. Scalable

Chapter 3

Security Architecture of Cloud Computing

This chapter aims at emphasizing the security needs in the cloud architecture. It highlights the security architecture introduced by NIST. It also gives an insight to the various challenges that must be addressed by the cloud service provider in order to build the trust of consumers. It finally concludes with the security recommendation list.

Cloud computing is a buzz in today's IT industry. It lets the organization move its storage to a trusted third party server. By doing so, organization need not worry about storing the big chunks of data as the cloud server takes the full responsibility of it. Cloud servers are responsible for storing data of multiple heterogeneous clients. This invites a bunch of questions for the data owner:

- How is it ensured that data of one client does not interfere with the data of another client?
- Will the confidentiality of data be maintained?
- How reliable are the services offered by the cloud?

The following figure depicts the result of a survey conducted in [14].

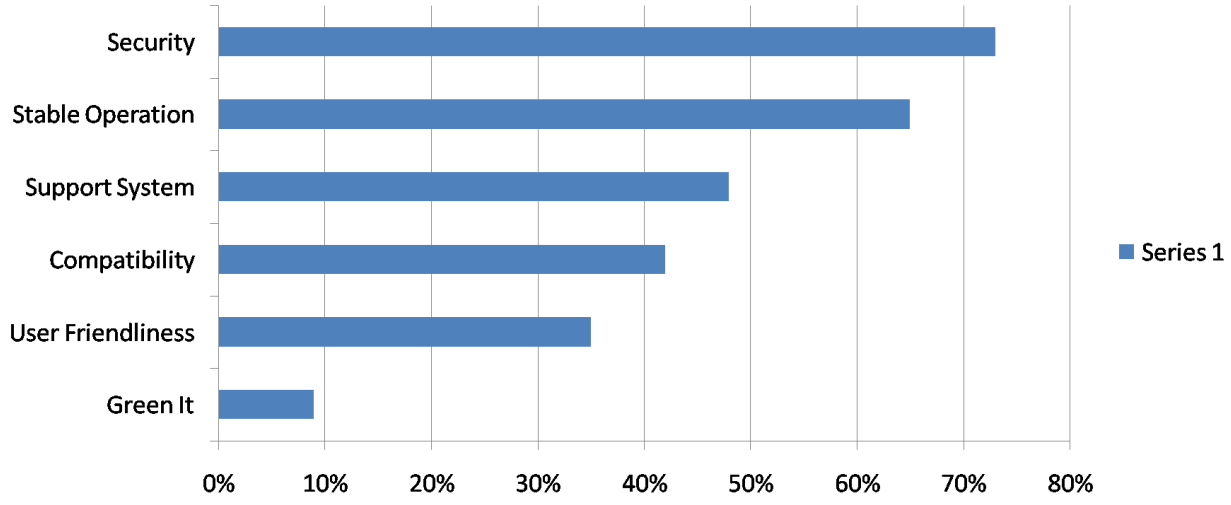

Fig. 10: User's apprehension in cloud computing

The security architecture of the cloud is built upon the underlying cloud architecture. Figure.11 illustrates the cloud architecture upon which the security architecture is constructed.

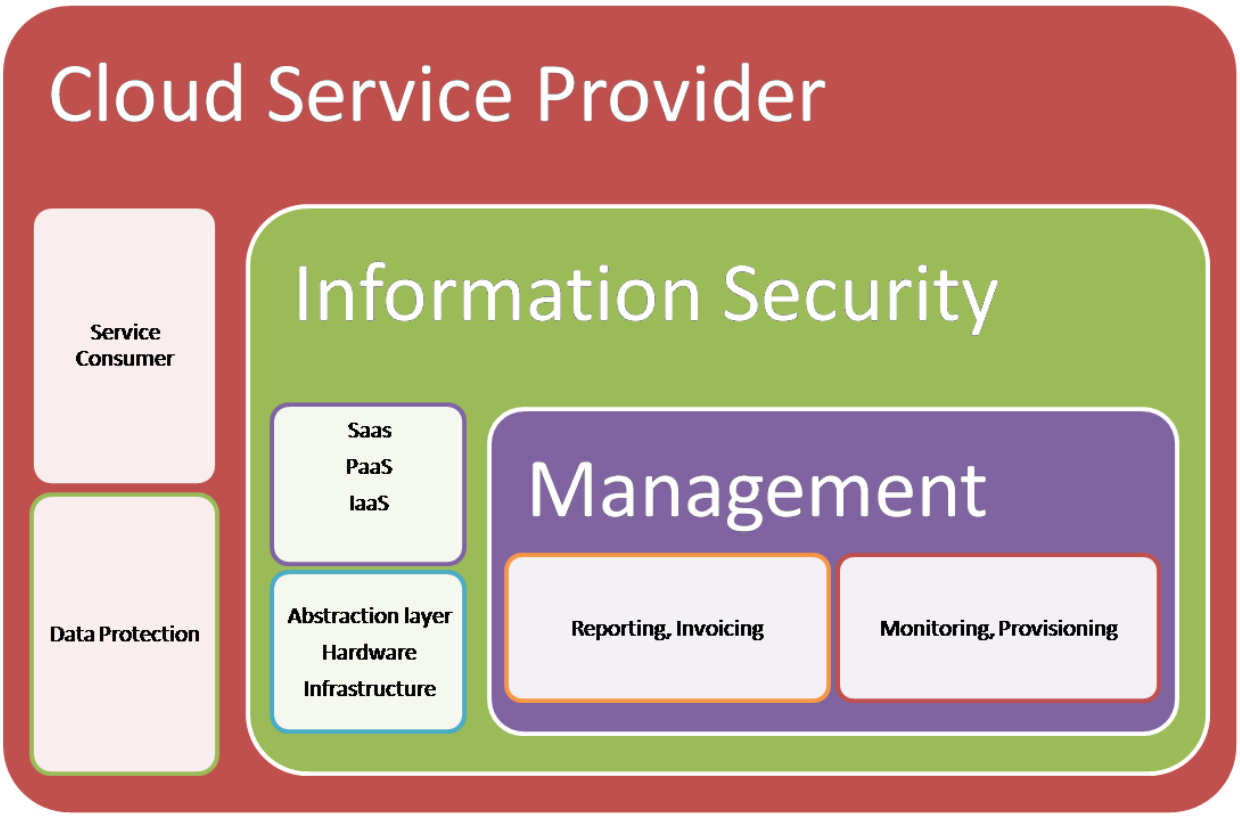

Fig. 11: Reference architecture for cloud security

It can be inferred from the above figure that the cloud service provider, CSP holds the responsibility of addressing the following duties:

- The first and foremost responsibility of the service provider is to present the customers the types of services it offers.

- Provisioning and de-provisioning of virtual resources like storage facilities, load balancers etc.
- Billing responsibilities. Consumers are given access to the resources made available by cloud server on pay per use bases. So, it is the responsibility of the cloud server to monitor the usage of resources.
- Monitoring and controlling of resources so as to detect any fault or failure in any of the resources and rectifying them so that the guaranteed level of service is maintained.

Consumers are required to choose among the service providers. This decision is dependent on the trust values of these service providers. It is important for a service provider to address the various security issues and gets itself certified in compliance with ISO 27001 based on IT, ISO 27001 or other established standard. Information Security Management System(ISMS) is a key requirement for any CSP. In order to provide the requisite reliability, the system must adhere to ISMS.

The following figure depicts the security architecture as defined by NIST.

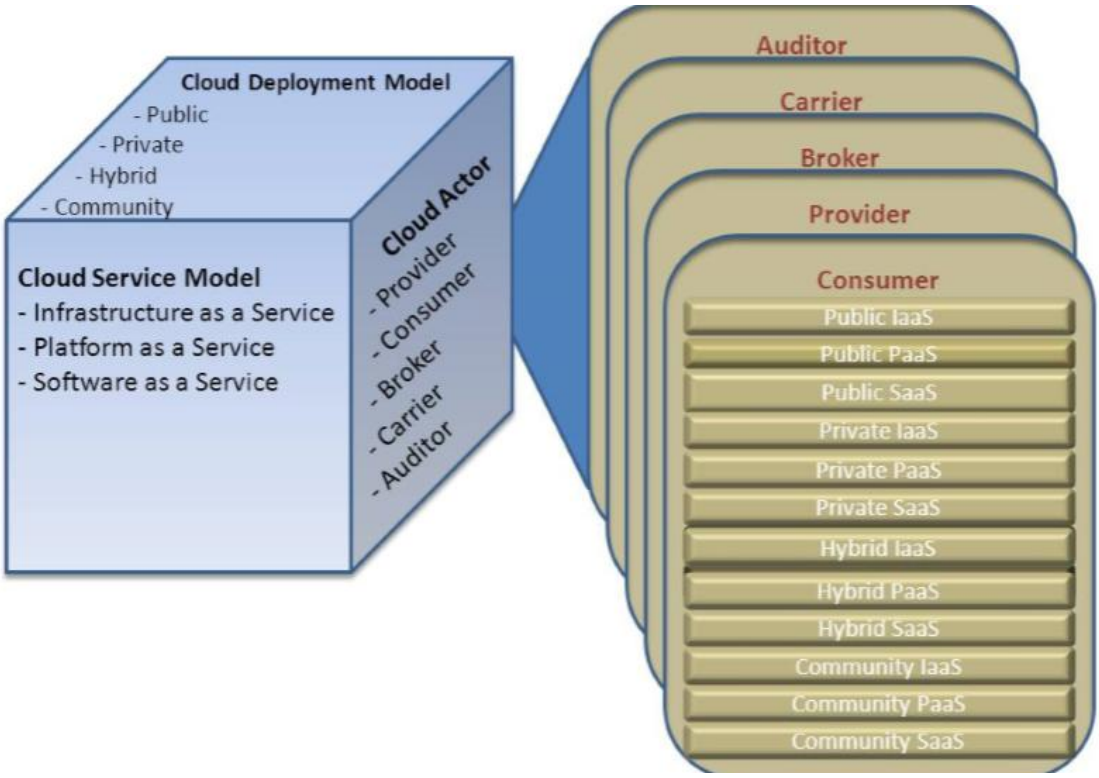

Fig. 12: NIST Security Reference Architecture [15]

For a cloud server to be fully secure, it must address the following security issues:

- Data Center Security
- Server Security
- Network Security
- Application and Platform Security
- Data Security
- Authentication

Table 4 lists down the security recommendations. These requirements are further subdivided based on the deployment model they belong to. Finally, they are assigned to one of the three categories.

Table 4: Security Recommendations

Security Management for Providers	Private ⇨			Public ⇨		
	B	C+	A+	B	C+	A+
Defined procedural model for all IT processes (e.g. as per ITIL, COBIT)	✓			✓		
Implementing a recognised information security management system (e.g. by BSI standard 100-2 (IT-Grundschutz), ISO 27001)	✓			✓		
Sustainably implementing an information security concept for the cloud	✓			✓		
Evidence of adequate information security (certification)		✓	✓		✓	✓
CSP has an adequate organisational structure for information security (including named contact persons to answer customers' security questions)	✓			✓		

- B: Basic Requirements
- C+: Confidentiality
- A+: Availability

Chapter 4

Proposed Algorithm and Framework

Access control has attracted a number of researchers and a number of algorithms have been put forth in this field. It is never possible to achieve complete protection against threats. The techniques described above suffer from one or many disadvantages.

In this thesis we have proposed an access control mechanism which tries to achieve the security parameters as discussed in chapter 1. The algorithm makes use of authorization certificates/tickets to achieve access control [4]. It consumes relatively less resources and is therefore easy to implement.

This chapter illustrates the algorithm proposed and system model being referenced.

4.1 Authorization Certificates

In the proposed algorithm, authorization certificates are the most fundamental unit. Its possession implies that the user has been granted a right to access a file in a mode as mentioned in the certificate. The certificates are issued by the data owners and are distributed to the clients who request to access the resources stored on the cloud. There is a restriction on the size of authorization certificates. Therefore, by using a single certificate we can grant access permissions only to a limited number of files. This requires the need of imposing restriction on the size of file names. Here we assume that length of names of every file is same.

Format of the authorization certificate used is shown in table 5. The certificate comprises of 6 fields as illustrated in the table.

Access Control Matrix

Access control matrix plots file names against the access modes. The access modes considered in this thesis are: read, modify and delete. In our example we have restricted the size of this field to 8. This means a certificate can grant access to the maximum of 8 files at a time. This field is used by the server to check if client has the access to a particular file.

Example of access control matrix is shown in table 6.

Table 5: Format of certificate

Access Control Matrix
Secret Information
Certificate Number
UID
Validity
Data Owner's Signature

Table 6: Access Control Matrix

File Names	Read	Modify	Delete
File1	✓		✓
File2		✓	
File3		✓	

Secret Information

Data owner uses a secret key to encrypt the files. Now, it is the responsibility of the data owner to make this secret key reach the clients safely. One way of doing so is to encrypt the secret key using client's public key so that only he is able to recover it. This has only one problem associated with it i.e. what if the secret key gets compromised. The entire mechanism is based on an assumption that the clients' key never gets compromised.

Certificate Number

Every certificate is uniquely numbered. When a request for issuing a certificate arises, data owner requests the cloud server for a unique certificate number. By giving the authority of issuing certificate numbers to cloud, we ensure that no two certificates contain a same certificate number. There are multiple data owners who want to upload their data to cloud. If they generate their own certificate number, there is a possibility that two data owners might generate same certificate number. Therefore, with the intention of making certificate number unique at cloud level, we outsource the task of generating certificate numbers to the server itself.

UID

It is a unique alphanumeric randomly generated identity embedded on every certificate by the data owner. The UID is also encrypted by client's public key ensuring that only he can retrieve it. Every certificate is issued with a unique identifier. This is used to ensure the anonymity of the user from the cloud. Also, this field is used to ensure the anonymity of the data sharer from the cloud service provider. UID varies with every certificate, so even if one UID gets compromised, it cannot get misused by the attacker. Data owner maintains the records of UIDs corresponding to certificate numbers. This record is used by him in access right revocation.

Validity

This field mentions the time until which the user has right to access a particular resource. It is completely the responsibility of the data owner to decide the validity period for a client. This field contains a date beyond which the users cannot access the files.

Data owner's Signature

This field ensures that that the certificate is not modified. Data owner digitally signs the certificate before issuing it to the customers. An RSA based method for digital signature is being used. The owner uses his private key to sign the certificate. The clients can use the owner's public key to decrypt the certificate. This not only authenticates the owner, also makes sure that the certificate is not tailored.

4.2 System Model

The underlying reference system model comprises of three entities:

- Cloud Service Provider
- Data Owner
- Data Sharer

The system model can be diagrammatically represented as shown in the next figure.

Data Owner uses the facilities provided by the cloud service provider to store its files on the server. It is his responsibility to grant or revoke access rights from the clients. The clients here are called as data sharers. Data owner gets to decide with whom he wants to share his files. It can adopt any policy to decide which user should get what access. These policies are not under the scope of this thesis.

Data Sharers are those who want to access the files stored by the owner on cloud.

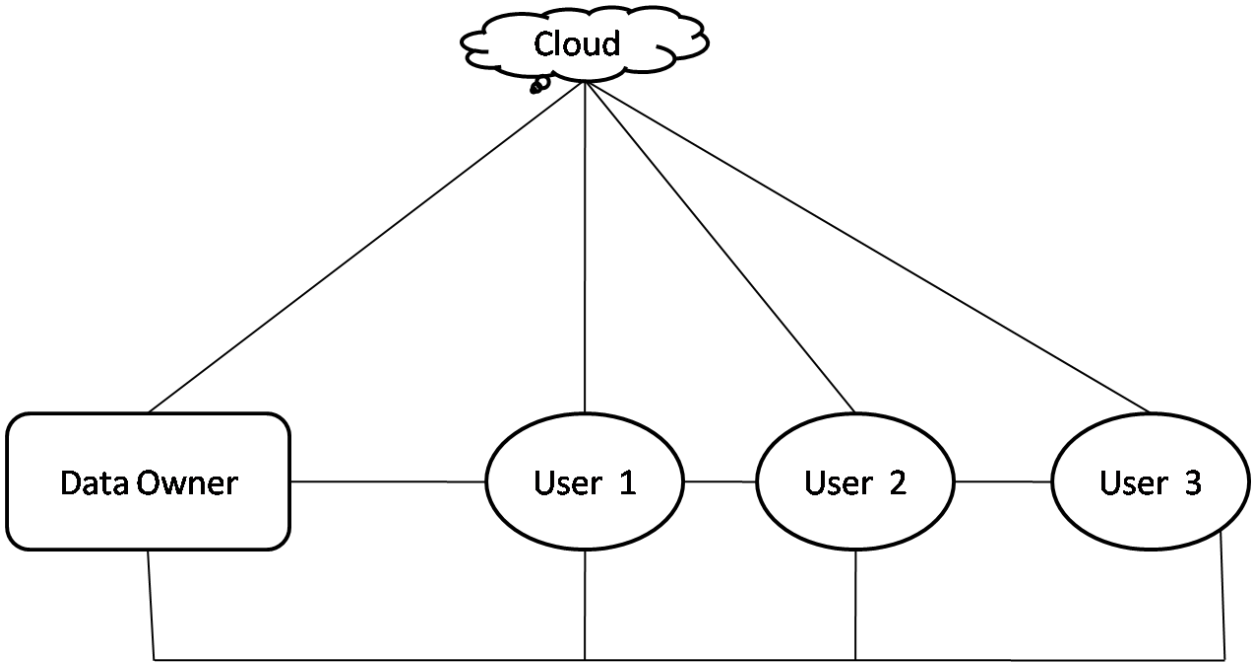

Fig. 13: System Model

4.3 Proposed Algorithm

The algorithm can be divided into four phases:

4.3.1 Set Up:- This algorithm is devised to be executed at data owner's side. It includes uploading the file on cloud server.

The algorithm can be described as:

Step 1 Generates a random secret key to encrypt the file.

Step 2 Encrypts the file with the key generated in Step 1 using any secure symmetric key cryptography.

Step 3 Calculate Message Digest of the file using whirlpool method.

Step 4 Uploads the file in encrypted format along with its digest on the server.

The following flowchart represents the above procedure.

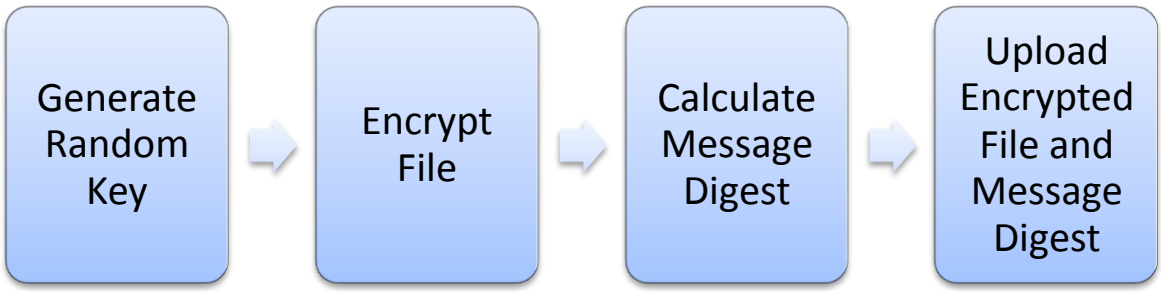

Fig. 14: File Upload

4.3.2 Request for Authorization certificate

This module is executed by the data sharer. Whenever a data sharer wants to access a file stored on the remote server, he is required to present his authorization certificate. To obtain the certificate, data sharer executes this module to request for the authorization certificate.

Figure.15 depicts the flow of information among the various entities.

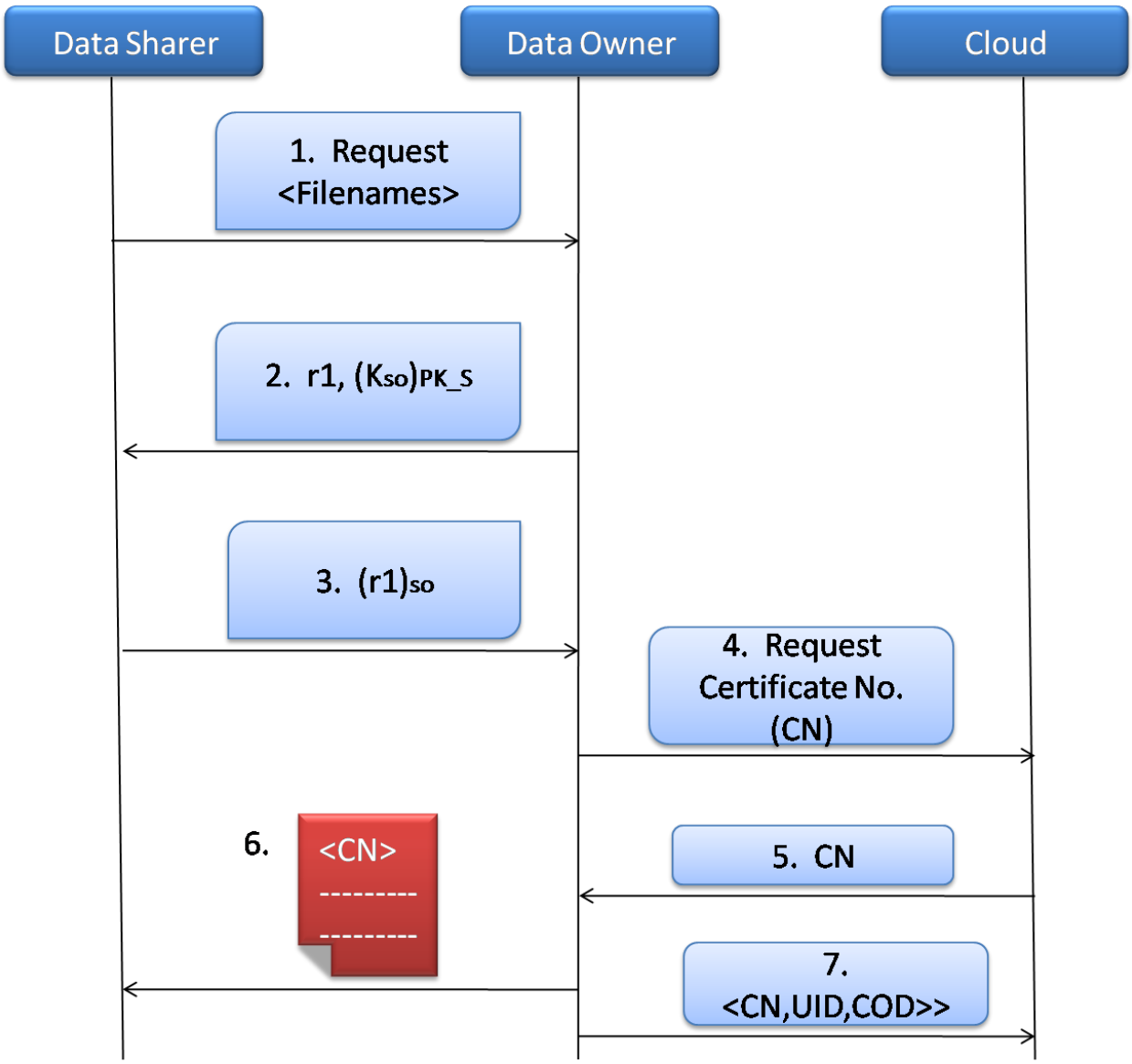

Fig. 15: Data Flow among Entities

1. Data sharer sends a "request" message along with the file names and access modes to the data owner.
2. Data owner generates a session key, encrypts it with the public key of sharer and sends it along with a random number (r1).

3. Data sharer decrypts the session key using his private key, encrypts r1 using the session key and sends to the owner.
4. Owner extracts the random number (r2) and compares it to the one he sent. If r1==r2, data owner checks his records to see if the user has been issued any certificate previously. If yes, it is checked if he can add more file names to the current certificate. If yes, modify the current certificate and goto step 7. Owner asks the cloud for a unique certificate number.
5. Cloud server replies with the unique certificate number.
6. Data owner generates the certificate:
    - Generates access control matrix
    - Inserts CN
    - Generates (UID)
    - Encrypts UID and key with session key
    - Calculates validity
    - Sign the certificate

It then sends the certificate to the user.

7. Data owner sends CN, UID, Certificate digest to the cloud server.

Following procedure takes place at client side:

- Client stores the certificate in its database
- It extracts key and UID using the session key and stores them separately

Data owner needs to be online only for issuing the certificates. Once the certificates have been issued data owner is no longer required to be online.

Figure 16 illustrates the flow chart of request process.

4.3.3 File Download

This module is devised for the data sharer. Using it, he is able to access the files in the requested modes. The modes in which a file can be accessed include: read, write and delete. The algorithm is described next.

Reading the file

Step 1 The data sharer extracts the certificate corresponding to the file he wants to access. He checks if the certificate is still valid. If no, removes the certificate from his database and requests for a new one with the help of "request certificate" module.

Step 2 He sends "read <file name>" request to the cloud server along with the certificate number.

Step 3 Cloud server checks his database for the certificate. If not found goto last step.

Step 4 Server checks if access to the file name is revoked in the revocation list. If the access rights to the file have been revoked it notifies the client and stop.

Step 5 Generate a random number, r and sends it to the sharer.

Step 6 Sharer calculates hash(r-1||UID) and sends to the cloud.

Step 7 Cloud server also computes hash(r-1||UID). It uses the UID stored in its database corresponding to the certificate number. If both match, sharer is validated. Else it stops.

Step 8 Sharer sends the certificate to the cloud.

Step 9 Cloud server calculates hash on the certificate and compares it with the value stored in its records. If both do not match goto step 13.

Step 10 It checks the validity on certificate. If not valid goto step 13.

Step 11 Checks if certificate contains file name with read access mode. If no, goto step 13

Step 12 Cloud server renders the encrypted file in "read-only" mode along with the message digest.

Step 13 Stop.

Steps 5-7 protect the algorithm against the replay attack. It is also responsible for validating (shown in the above figure) the data sharer.

Activities that take place at client side after receiving the file:

- After receiving the file, data sharer decrypts it using the secret key (provided to him by the owner in the certificate)
- Computes message digest.
- Compares this digest with the digest downloaded from server.
- If both match, it means file has not been interrupted else if they don't match, it means file has been modified. The sharer informs about this modification to the data owner.

Modifying the file

Algorithm used for modification of file is done using the similar algorithm with few changes. Before uploading the file to server, it encrypts the file using shared symmetric key. Message digest is computed.

Algorithm for modifying file:

Step 1 Encrypt the file using shared symmetric key.
Step 2 Calculate Message digest.
Step 3 Same as step 1 -10 from reading algorithm.
Step 4 Upload encrypted file along with message digest to the cloud server.
Step 5 Stop.

File Deletion

Data sharer can delete the file if they have the right to do so. For deleting the file, sharer needs to send "Delete" request to the server. Rest is the algorithm is same as reading algorithm. If everything satisfies the requirements cloud server deletes the file and notifies the owner.

4.3.4 Access Right Revocation

This phase is implemented at data owner side. It is the decision of data owner to revoke the access rights of any user at any point of time.

Data owner can either revoke the certificate completely or it can revoke all accesses to a file or an access mode of a file.

Algorithm to revoke access rights:

Step 1 Data owner creates "revoke" request.

Step 2 If data owner wants to revoke the access rights completely, he sends the certificate number to the cloud server.

Step 3 If access to few files has to be revoked, he sends CN and file names. In this case all the access rights to the file gets revoked.

Step 4 If access to file for a particular mode is to be revoked, he sends CN, file name and mode.

Cloud server maintains a revocation list in the following format

S.No	CN	File Name	Mode

.

Chapter 5

Implementation and Results

This chapter provides insight into the implementation details. The algorithm described in the above chapter has been implemented in java using eclipse IDE.

Figure.18 shows the interface used by the data owner to upload file on the cloud. It is also used for revoking access rights from the user.

It implements the set up algorithm as described in section 4.3.1.

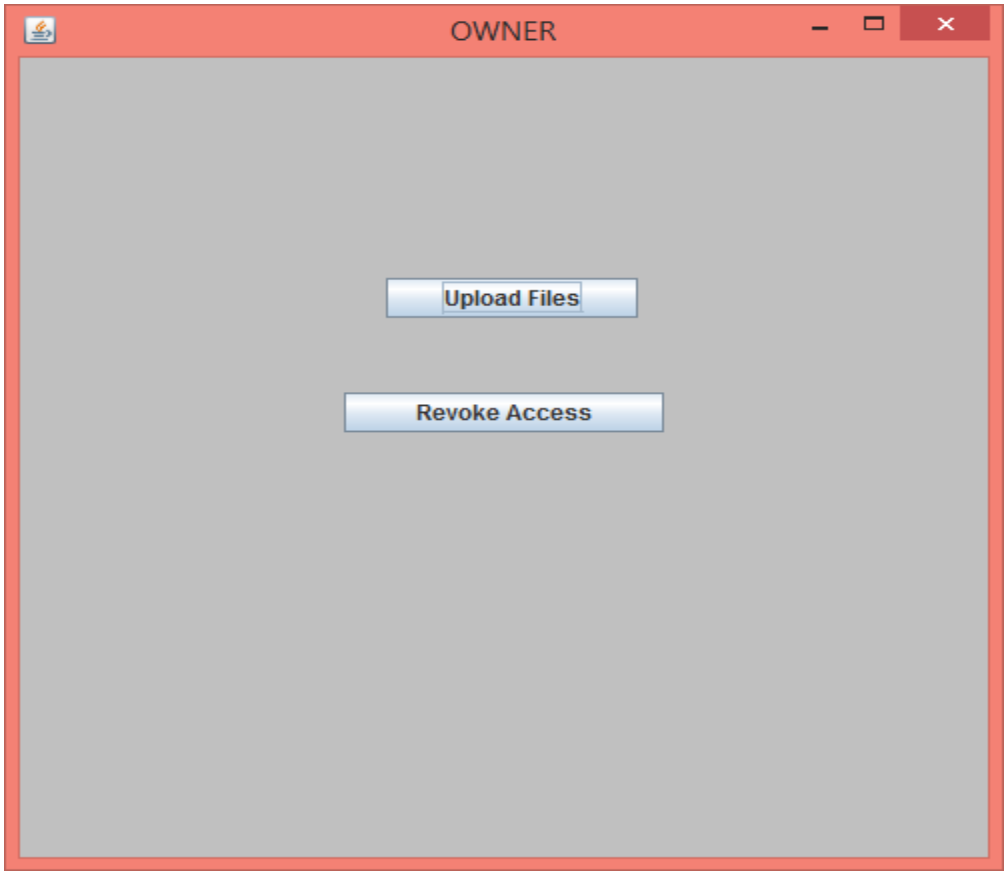

Fig. 18: UI for uploading files and revoking access

Upload button lets the owner choose file to upload on server. Figure.19 lets the data owner choose file from its local storage using the file chooser.

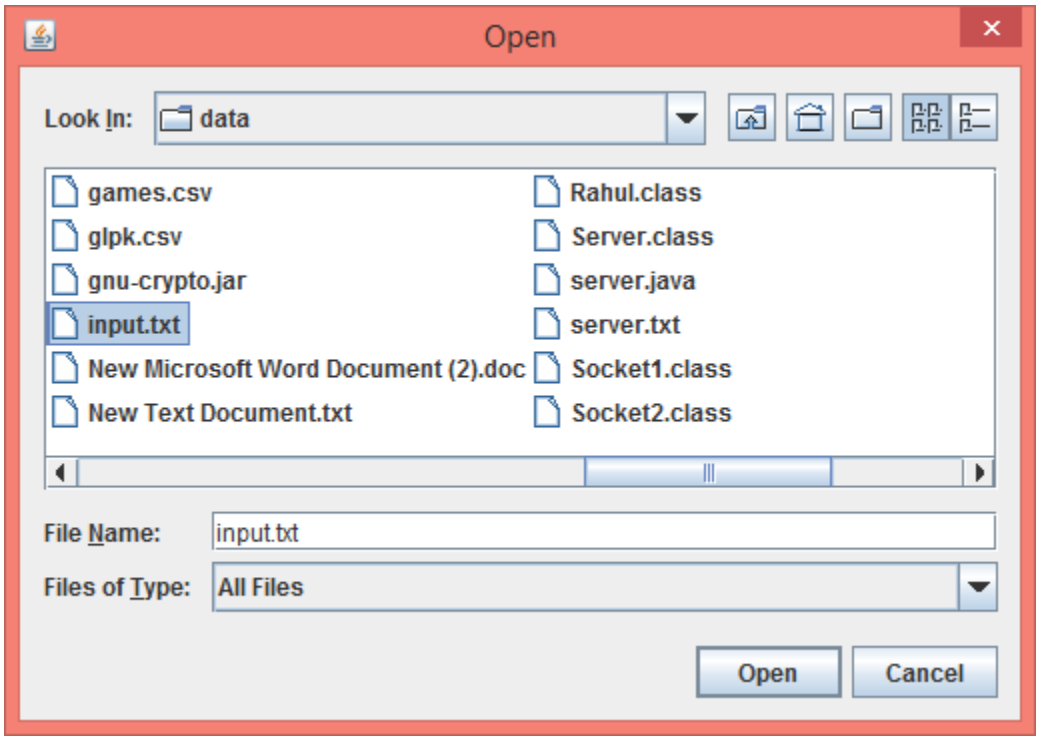

Fig. 19: File chooser

The file chosen is encrypted and uploaded on the cloud server using this upload module.

Once the set up is done, data owner is ready to issue authorization certificates. The data sharers use the interface in Figure.20 to request access to a file. “Request Authorization Certificate” in the figure lead data sharer to the next window (Figure.21) which lets him chooses the file names for which he wants access permissions.

After validating the requests, data owner issues the certificate in the described format.

Once the certificate is issued, data owner is not required to stay online any more. Data sharer uses the certificate issued to him to access files from the server. Figure.22 shows UI for accessing files.

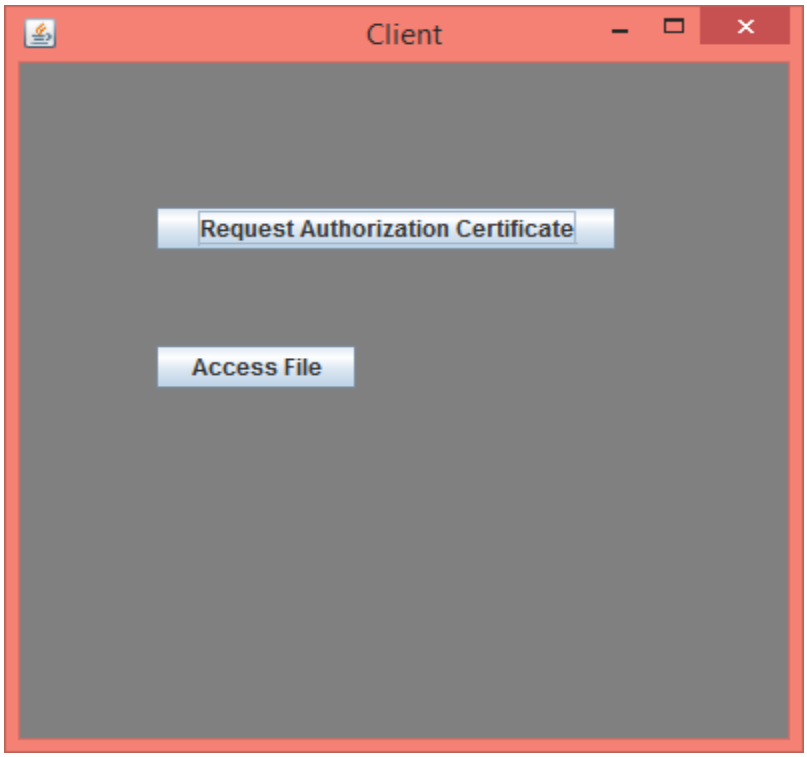

Fig. 20: UI for Data sharers

Fig. 21: UI for requesting authorization certificates

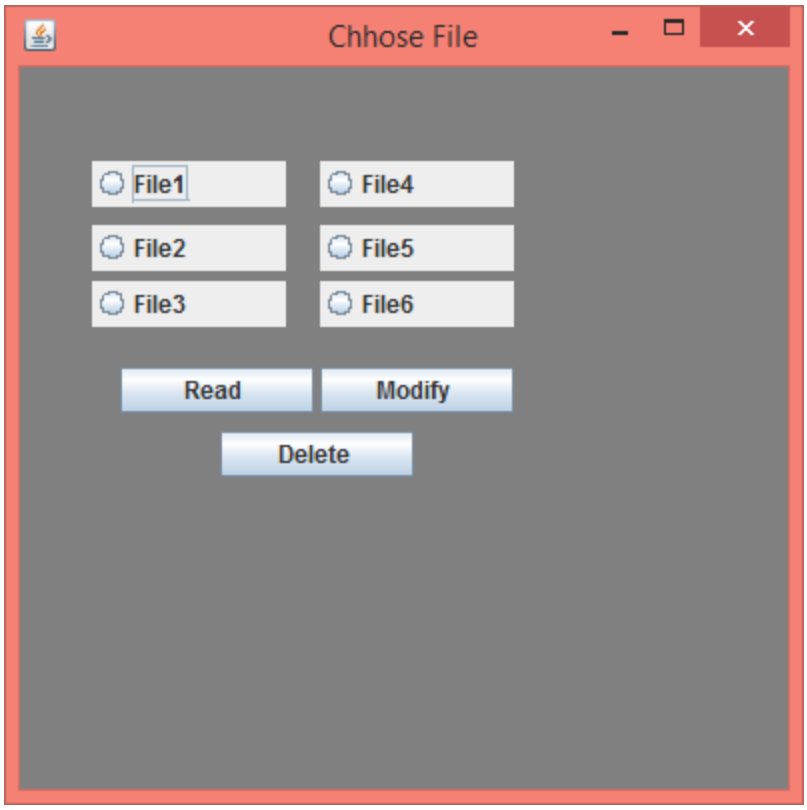

Fig. 22: UI to access files

Chapter 6

Security Analysis

Data stored on the remote cloud server is critical to the organization. It is important to protect it from an unauthorized user. It is not possible for any algorithm to be completely secure. The algorithm proposed ensures the following security features:

A. Security

Security is concerned with confidentiality and integrity of data. Data is uploaded in cipher-text form. Data owner uses a secure symmetric key encryption to encrypt the data. This key is transferred securely to the data sharers. Data owner uses the public key of sharer to encrypt the key. Since, key is shared only among the authorized users confidentiality of data is ensured. Data owner computes the message digest of the file encrypting it. Message digest is also uploaded with the encrypted file. Data sharer, on other side uses the secret key provided to him via authorization certificate to decrypt the file. He then evaluates its message digest and compares them. If both are same, it means file has not been modified else file has been modified. Data sharer notifies the owner about this modification.

B. Dynamic Data Sharing

It is not required to change the cipher text every time a user leaves the system. The proposed architecture is independent of the number of data sharers. Moreover, access rights can be granted or revoked as per the needs.

C. Fine-Grained Access Control

Data sharers use authorization certificate to show that they have the right to access the file. Authorization certificate specifies which user can access which resource in which mode. Data owner is responsible for distributing authorization certificates. Since in the proposed algorithm, data sharer is able to control the usage of files by users, we can say that fine-grained access control is achieved.

D. Accountability

In the proposed algorithm, data owner maintains a record of the certificates issued by it. A record is also maintained by the cloud server. Cloud server stores the certificate number, a random identifier generated the owner and the certificate digests. This information is needed while revoking the access rights.

E. Scalability

Performance of the system does not get degraded with the number of users. Adding a user only requires issue of a new authorization certificate. Size of cipher text is independent of the number of users. Therefore, we can say that our system is scalable.

F. Replay attacks

The proposed algorithm is safe against replay attacks. Every time data sharer wants to access a resource, he sends him the certificate no. Cloud server generates a random number and sends to the sharer. Sharer responds to the server with the hash of random number and UID. Since, server generates a new random number on every request no one can eavesdrop.

G. User's Anonymity

Data sharer uses the random UID generated by the data owner while requesting the cloud server. Since cloud has no information regarding the original identity of the client, we can say that user's anonymity is maintained.

Chapter 7

Conclusion

This chapter discusses the conclusions inferred from this research.

This research work proposes a new access control mechanism. Our algorithm uses authorization certificates to achieve access control in cloud computing. The proposed algorithm does not include any complex computation like bilinear pairing. It uses minimal resources and is easy to implement. The basic operations used in the algorithm include private key encryption, public key encryption and hash.

The algorithm adheres to the following essential parameters:

- Security
- Fine-grained access control
- Dynamic data sharing
- Accountability
- Scalability

It uses random identity generated by the data owner, therefore ensuring user's anonymity. It is also safe against the replay attacks.

REFERENCES

[1] Peter Mell ,Timothy Grance ,"The NIST Definition of Cloud Computing".

[2] Google images
www.google.com

[3] Hsin-Yi Tsai, Melanie Siebenhaar and André Miede, Yu-Lun Huang, Ralf Steinmetz, "Virtualization's Impact on Cloud Security", 2012.

[4] Xuanxia Yao, Xiaoguang Han, Xiaojiang Du, "A Lightweight Access Control Mechanism for Mobile Cloud Computing", IEEE INFOCOM Workshop on Mobile Cloud Computing, 2014.

[5] Vipul Goyal, Omkant Pandey, Amit Sahai, Brent Waters, "Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data".

[6] John Bethencourt, Amit Sahai, Brent Waters, "Ciphertext-Policy Attribute-Based Encryption", 2007.

[7] Suyel Namasudra, Samir Nath, Abhishek Majumder, "Profile Based Access Control Model in Cloud Computing Environment", 2013

[8] Yang Tang, Patrick P.C. Lee, John C.S. Lui, Radia Perlman, "Secure Overlay Cloud Storage with Access Control and Assured Deletion" 2012.

[9] LIN Guoyuan, WANG Danrul, BIE Yuyul, LEI Min, "MTBAC: A Mutual Trust Based Access Control Model in Cloud Computing", 2014.

[10] Wenhui Wang ,Jing Han ,Meina Song,Xiaohui Wang, "The Design of a Trust and Role Based Access Control Model in Cloud Computing", 2011.

[11] Kong Guangqian, Li Jianshi. "Research on RBAC-based separation of duty constraints, " Journal of InformatIOn and Computing Science, Vo1.20, 2007,pp. 235-24", 2007.

[12] ILANCHEZHIAN. J, VARAD HARASSU. V , RANJEETH. A , ARUN. K ," To Improve the Current Security Model and Efficiency in Cloud Computing Using Access Control Matrix", 2012.

[13] Daniel Ricardo dos Santos, Carla Merkle Westphall and Carlos Becker Westphall, "A Dynamic Risk-based Access Control Architecture for Cloud Computing".

[14] Masayuki Okuhara, Tetsuo Shiozaki, Takuya Suzuki, "Security Architectures for cloud computing", 2010.

[15] NIST Cloud Computing 6 Security Reference Architecture.