Chapter 1

INTRODUCTION

1.1 INTRODUCTION

Imagine yourself in the world where the users of today's Internet world don't have to run, install or store their application or data on their own computers, imagine the world where every piece of our information or data would reside on the Cloud(Internet). As a metaphor for the Internet, "the cloud" is a familiar cliché, but when combined with "computing," the meaning gets bigger and fuzzier. Some analysts and vendors define cloud computing narrowly as an updated version of utility computing: basically virtual servers available over the Internet. Others go very broad, arguing anything you consume outside the firewall is "in the cloud," including conventional outsourcing.

1.2 CLOUD COMPUTING-THE CONCEPT

Cloud Computing is internet based development and use of computer technology. It is a style of computing in which IT-related capabilities are provided as a service, allowing users to access technology enabled services from the internet without the knowledge of expertise or control over technology infrastructure that supports them.

Cloud computing comes into focus only when we think about what IT always needs: a way to increase capacity or add capabilities on the fly without investing in new infrastructure, training new personnel, or licensing new software. Cloud computing encompasses any subscription-based or pay-per-use service that, in real time over the Internet, extends Its existing capabilities.

Cloud computing is at an early stage, with a motley crew of providers large and small delivering a slew of cloud-based services, from full-blown applications to storage services to spam filtering. Yes, utility-style infrastructure providers are part of the mix, but so are SaaS (software as a service) providers such as Salesforce.com. Today, for the most part, IT must plug into cloud-based services individually, but cloud computing aggregators and integrators are already emerging.

The National Institute of Standards and Technology's definition of cloud computing identifies "five essential characteristics":

On-demand self-service. A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider.

Broad network access. Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, tablets, laptops, and workstations).

Resource pooling. The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand.

Rapid elasticity. Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear unlimited and can be appropriated in any quantity at any time.

Measured service. Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.

1.3 ARCHITECTURE

Cloud architecture, the system architecture of the software systems involved in the delivery of cloud computing, typically involves multiple cloud components communicating with each other over a loose coupling mechanism such as a messaging queue. Elastic provision implies intelligence in the use of tight or loose coupling as applied to mechanisms such as these and others.

These components typically consist of a front end platform (fat client, thin client, mobile device), back end platforms (servers, storage), a cloud based delivery, and a network (Internet, Intranet, Intercloud). Combined, these components make up cloud computing architecture.

A cloud computing sample architecture:

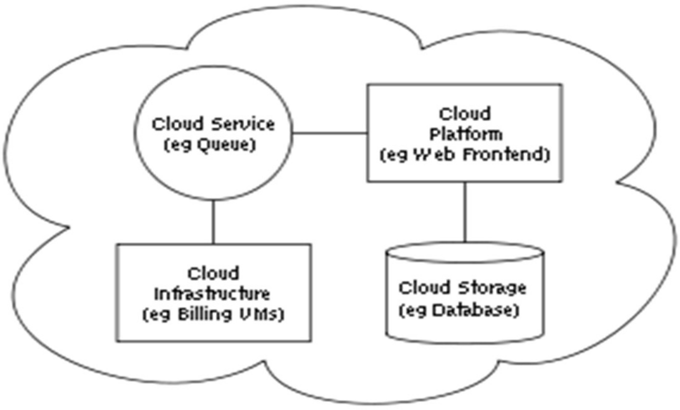


Fig. 1.3.1 A sample cloud architecture

1.4 BENEFITS OF CLOUD COMPUTING:

- Minimized Capital expenditure
- Location and Device independence
- Utilization and efficiency improvement
- Very high Scalability
- High Computing power

1.5 SECURITY A MAJOR CONCERN:

- Security concerns arising because both customer data and program are residing in Provider Premises.
- Security is always a major concern in Open System Architectures.

1.6 DATA CENTRE SECURITY

- Professional Security staff utilizing video surveillance, state of the art intrusion detection systems, and other electronic means.
- When an employee no longer has a business need to access datacenter his privileges to access datacenter should be immediately revoked.
- All physical and electronic access to data centers by employees should be logged and audited routinely.
- Audit tools so that users can easily determine how their data is stored, protected, used, and verify policy enforcement.

1.7 DATA LOCATION:

- When user uses the cloud, user probably won't know exactly where the data is hosted.
- Data should be stored and processed only in specific jurisdictions as defined by the user.
- Provider should also make a contractual commitment to obey local privacy requirements on behalf of their customers.
- Data-centered policies that are generated when a user provides personal or sensitive information, that travels with that information throughout its lifetime to ensure that the information is used only in accordance with the policy.

1.8 INFORMATION SECURITY:

- Security related to the information exchanged between different hosts or between hosts and users.
- This issues pertaining to secure communication, authentication, and issues concerning single sign on and delegation.
- Secure communication issues include those security concerns that arise during the communication between two entities.

- These include confidentiality and integrity issues. Confidentiality indicates that all data sent by users should be accessible to only “legitimate” receivers, and integrity indicates that all data received should only be sent/modified by “legitimate” senders.
- **Solution:** public key encryption, X.509 certificates, and the Secure Sockets Layer (SSL) enables secure authentication and communication over computer networks.

In the current era of digital world, various organizations produce a large amount of sensitive data including personal information, electronic health records, and financial data. The local management of such huge amount of data is problematic and costly due to the requirements of high storage capacity and qualified personnel. Therefore, storage-as-a-service offered by cloud service providers (CSPs) emerged as a solution to mitigate the burden of large local data storage and reduce the maintenance cost by means of outsourcing data storage.

Since the data owner physically releases sensitive data to a remote CSP, there are some concerns regarding confidentiality, integrity, and access control of the data. For verifying data integrity over cloud servers, researchers have proposed provable data possession technique to validate the intactness of data stored on remote sites.

A number of PDP protocols have been presented to efficiently validate the integrity of data. Proof of irretrievability was introduced as a stronger technique than PDP in the sense that the entire data file can be reconstructed from portions of the data that are reliably stored on the servers. Commonly, traditional access control techniques assume the existence of the data owner and the storage servers in the same trust domain. This assumption, however, nolonger holds when the data are outsourced to a remote CSP, which takes the full charge of the outsourced data management, and resides outside the trust domain of the data owner.

A feasible solution can be presented to enable the owner to enforce access control of the data stored on a remote non trusted CSP. Through this solution, the data are encrypted under a certain key, which is shared only with the authorized users. The unauthorized users, including the CSP, are unable to access the data because they do not have the decryption key. This general solution has been widely incorporated into existing schemes, which aim at providing data storage security on non-trusted remote servers.

Another class of solutions utilizes attribute-based encryption to achieve fine-grained access control. Different approaches have been investigated that encourage the owner to outsource the data, and offer some sort of guarantee related to the confidentiality, integrity, and access control of the outsourced data.

These approaches can prevent and detect malicious actions from the CSP side. On the other hand, the CSP needs to be safeguarded from a dishonest owner, who attempts to get illegal compensations by falsely claiming data corruption over cloud servers. This concern, if not properly handled, can cause the CSP to go out of business.

Chapter 2

LITERATURE SURVEY

Literature survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, the next step is to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system the above consideration are taken into account for developing the proposed system.

Following are some references taken during the development of the system:

Enabling Public Verify ability and Data Dynamics for Storage Security in Cloud Computing: [5]

This paper describes that "Cloud Computing has been envisioned as the next generation architecture of IT Enterprise. It moves the application software and databases to the centralized large data centers, where the management of the data and services may not be fully trustworthy. This unique paradigm brings about many new security challenges, which have not been well understood. This work studies the problem of ensuring the integrity of data storage in Cloud Computing. It first identifies the difficulties and potential security problems of direct extensions with fully dynamic data updates from prior works and then shows how to construct an elegant verification scheme for seamless integration of these two salient features in our protocol design.

Providing Dynamic Data Storage and Indirect Mutual Trust in Clouds: [2]

Storage-as-a-Service (SaaS) offered by cloud service providers (CSPs) is a paid facility that enables organizations to outsource their data to be stored on remote servers. Thus, SaaS reduces the maintenance cost and mitigates the burden of large local data storage at the organization's end. A data owner pays for a desired level of security and must get some compensation in case of any misbehavior committed by the CSP. On the other hand, the CSP needs a protection from any false accusation that may be claimed by the owner to get illegal compensations. This paper proposes a cloud-based storage scheme that allows the data owner to benefit from the facilities offered by the CSP and enables indirect mutual trust between them. The proposed scheme has four important features: First one, it allows the owner to outsource sensitive data to a CSP, and perform full block-level dynamic operations on the outsourced data, i.e., block modification, insertion, deletion, and append. Second one, it ensures that authorized users (i.e., those who have the right to access the owner's file) receive the latest version of the outsourced data. Next one, it enables indirect mutual trust between the owner and the CSP. Last one, it allows the owner to grant or revoke access to the outsourced data.

Replicated Data Integrity Verification in Cloud: [7]

Data replication is a commonly used technique to increase the data availability in cloud computing. Cloud replicates the data and stores them strategically on multiple servers located at various geographic locations. Since the replicated copies look exactly similar, it is difficult to verify whether the cloud really stores multiple copies of the data. Cloud can easily cheat the owner by storing only one copy of the data. Thus, the owner would like to verify at regular intervals whether the cloud indeed possesses multiple copies of the data as claimed in the SLA.

In general, cloud has the capability to generate multiple replicas when a data owner challenges the CSP to prove that it possesses multiple copies of the data. Also, it is a valid assumption that the owner of the data may not have a copy of the data stored locally. So, the major task of the owner is not only to verify that the data is intact but also to recover the data if any deletions/corruptions of data are identified. Since, the replicas are to be stored at diverse geographic locations; it is safe to assume that a data loss will not occur at all the replicas at the same time.

Chapter 3

SYSTEM REQUIREMENTS

3.1 HARDWARE REQUIREMENTS:

System	: Pentium IV 2.4 GHz.
Hard Disk	: 40 GB.
Monitor	: 15 inch VGA Color.
Mouse	: Logitech Mouse.
Ram	: 512 MB
Keyboard	: Standard Keyboard

3.2 SOFTWARE REQUIREMENTS:

Operating System	: Windows 7
Coding Language	: C#
Database	: Microsoft SQL Server 2008
Server	: IIS7
I.D.E.	: Visual Studio 2010

3.3 FUNCTIONAL REQUIREMENTS

In software engineering, a functional requirement defines a function of a system or its component. A function is described as a set of inputs, the behavior, and outputs.

3.3.1 CSU

Register: Cloud service user register by entering the required details after Registration they are authenticated by cloud service provider**.**

Login: Registered User Login through valid Username and Password.

Upload: User Uploads particular file**.**

Details: User can view details of the files uploaded by them.

Files: User can view their files and also others.

Profile: User can view their profile.

Modification: User can view modified files**.**

Requisition By TPA: User Views Requested key from TPA and response the key.

Requested Files: View TPA Requested files.

3.3.2 CSP

Register: Register Requested Cloud Service User and TPA/TTP.

User Details: View Cloud Service User Details.

Modified Files: View files which are modified by cloud service user.

TPA: View all the Registered TPA's.

Pending Files: View the Request Pending files.

3.3.3 TPA/TTP

Login: User Logins through Username and Password.

Files: View all the files uploaded from users

Download: Selects Particular file and downloads by entering user key or Request for key is sent to cloud service user.

Modify: once the files are downloaded from CSU .It can be modified like add, edit etc.

Reupload: After Modifying the files are again uploaded to cloud service user.

3.4 NON FUNCTIONAL REQUIREMENTS

This attributes will specifies the system characteristics with respect to their functionalities in terms of reliability, availability, security, and maintainability.

3.4.1 Reliability

Since the system has admin as well as user authentication process, it is trustworthy and due to systematic operation it is reliable in nature.

3.4.2 Availability

The application will be available on any .NET platform

3.4.3 Security

The application provides complete security for Security system through user credential since the comparison of the previously stored credential is done with the presently entered. Where the comparison is done based on stored credential and allows the user to access the application.

3.4.4 Maintainability

Since we are using the .NET platform to support our application no maintenance is very easy and economical also.

3.4.5 Interoperability

Is the ability of making systems and organizations to work together (inter-operate).

3.4.6 Usability

Is the ease of use and learn ability of a human-made object. The object of use can be a software application, website, book, tool, machine, process, or anything a human interacts with.

3.4.7 Extensibility

Is a system design principle where the implementation takes future growth into consideration.

CHAPTER 4

SYSTEM ANALYSIS

System Analysis is a detailed study of the various operations performed by a system and their relationships within and outside of the system. Here the key question is- what all problems exist in the present system? What must be done to solve the problem? Analysis begins when a user or manager begins a study of the program using existing system.

During analysis, data collected on the various files, decision points and transactions handled by the present system. The success of the system depends largely on how clearly the problem is defined, thoroughly investigated and properly carried out through the choice of solution. A good analysis model should provide not only the mechanisms of problem understanding but also the frame work of the solution. Thus it should be studied thoroughly by collecting data about the system. Then the proposed system should be analyzed thoroughly in accordance with the needs.

Storage as a Service (SaaS) facilitates cloud applications to scale beyond their limited servers. SaaS allows users to store their data at remote disks and access them anytime from any place. Cloud storage systems are expected to meet several rigorous requirements for maintaining users' data and information, including high availability, reliability, performance, replication and data consistency; but because of the conflicting nature of these requirements, no one system implements all of them together.

4.1 EXISTING SYSTEM

Commonly, traditional access control techniques assume the existence of the data owner and the storage servers in the same trust domain. This assumption, however, no longer holds when the data is outsourced to a remote CSP, which takes the full charge of the outsourced data management, and resides outside the trust domain of the data owner. Existing research close to our work can be found in the areas of integrity verification of outsourced data, cryptographic file systems in distributed networks, and access control of outsourced data.

4.1.1 DISADVANTAGES OF EXISTING SYSTEM

CSP needs to be safeguarded from a dishonest owner, who attempts to get illegal compensations by falsely claiming data corruption over cloud servers. This concern, if not properly handled, can cause the CSP to go out of business.

4.2 PROPOSED SYSTEM

We propose a cloud-based storage scheme that allows the data owner to benefit from the facilities offered by the CSP and enables indirect mutual trust between them. The proposed scheme has four important features:

(i) It allows the owner to outsource sensitive data to a CSP, and perform full block-level dynamic operations on the outsourced data, i.e., block modification, insertion, deletion, and append,

(ii) It ensures that authorized users (i.e., those who have the right to access the owner's file) receive the latest version of the outsourced data,

(iii) It enables indirect mutual trust between the owner and the CSP, and

(iv) It allows the owner to grant or revoke access to the outsourced data. We discuss the security issues of the proposed scheme. Besides, we justify its performance through theoretical analysis and experimental evaluation of storage, communication, and computation overheads.

4.2.1 ADVANTAGES OF PROPOSED SYSTEM

(i) It allows a data owner to outsource the data to a CSP, and perform full dynamic operations at the block-level, i.e., it supports operations such as block modification, insertion, deletion, and append.

(ii) It ensures the newness property, i.e., the authorized users receive the most recent version of the outsourced data.

(iii) It establishes indirect mutual trust between the data owner and the CSP since each party resides in a different trust domain; and

(iv) It enforces the access control for the outsourced data

4.3 FEASIBILITY STUDY

The feasibility study proposes one or more conceptual solution to the problem set of the project. In fact, it is an evaluation of whether it is worthwhile to proceed with project or not.

Feasibility analysis usually considers a number of project alternatives, one that is chosen as the most satisfactory solution. These alternatives also need to be evaluated in a broad way without committing too many resources. Various steps involved in feasibility analysis are:

1. To propose a set of solution that can realize the project goal. These solutions are usually descriptions of what the new system should look like.
2. Evaluation of feasibility of such solutions. Such evaluation often indicates shortcomings in the initial goals. This step is repeated as the goals are adjusted and the alternative solutions are evaluated.

Four primary areas of interest in feasibility study are:

4.3.1 ECONOMIC FEASIBILITY

The purpose of the economic feasibility assessment is to determine the positive economic benefits to the organization that the proposed system will provide. It includes quantification and identification of all the benefits expected. This assessment typically involves a cost/ benefits analysis.

Cloud solutions deliver real benefits to businesses, particularly from increased cost efficiency, improved flexibility and enhanced resilience. Cloud provides computing power, data storage and the associated security and connectivity without the need for the customer to buy any hardware, O/S licenses, and management tools or provide additional support teams which means almost zero capital expenditure.

4.3.2 TECHNICAL FEASIBILITY

The technical feasibility assessment is focused on gaining an understanding of the present technical resources of the organization and their applicability to the expected needs of the proposed system. It is an evaluation of the hardware and software and how it meets the need of the proposed system.

The underlying technology focuses on maximising the utilisation of computing power and disk storage resulting in less wasted resource.

Customers need only to buy what they require today rather than buying an IT solution that they will have to grow into overtime and then provision extra resource as and when it is needed.

4.3.3 OPERATIONAL FEASIBILITY

Operational feasibility is a measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development.

The operational feasibility assessment focuses on the degree to which the proposed development projects fits in with the existing business environment and objectives with regard to development schedule, delivery date, corporate culture, and existing business processes.

To ensure success, desired operational outcomes must be imparted during design and development. These include such design-dependent parameters such as reliability, maintainability, supportability, usability, producibility, disposability, sustainability, affordability and others. These parameters are required to be considered at the early stages of design if desired operational behaviors are to be realized.

A system design and development requires appropriate and timely application of engineering and management efforts to meet the previously mentioned parameters. A system may serve its intended purpose most effectively when its technical and operating characteristics are engineered into the design. Therefore operational feasibility is a critical aspect of systems engineering that needs to be an integral part of the early design phases.

Chapter 5

SYSTEM DESIGN

5.1 DATA FLOW DIAGRAMS

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. Often they are a preliminary step used to create an overview of the system which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design). A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of processes, or information about whether processes will operate in sequence or in parallel (which is shown on a flowchart).

5.1.1 CSU

A cloud service user is the organization or the person who seeks to take advantage of the cloud computing services provided by the service provider. The user accesses services through a browser over the internet. The user needs to first get registered with the cloud service provider. If a registered user wishes to access the cloud services, then he needs to proceed with the login steps by entering the username and password assigned to him. Once logged in the user is presented with an interface in which there are various options which can be selected. The following are some of the tabs which a user can select:

1. File Uploaded: Carries all the files uploaded by user.
2. File Details: It contains all the properties about a file.
3. Profile: Carries all the details about a user.
4. Modified files: It contains all the modified files.
5. Requested Files: It carries all the requested files.

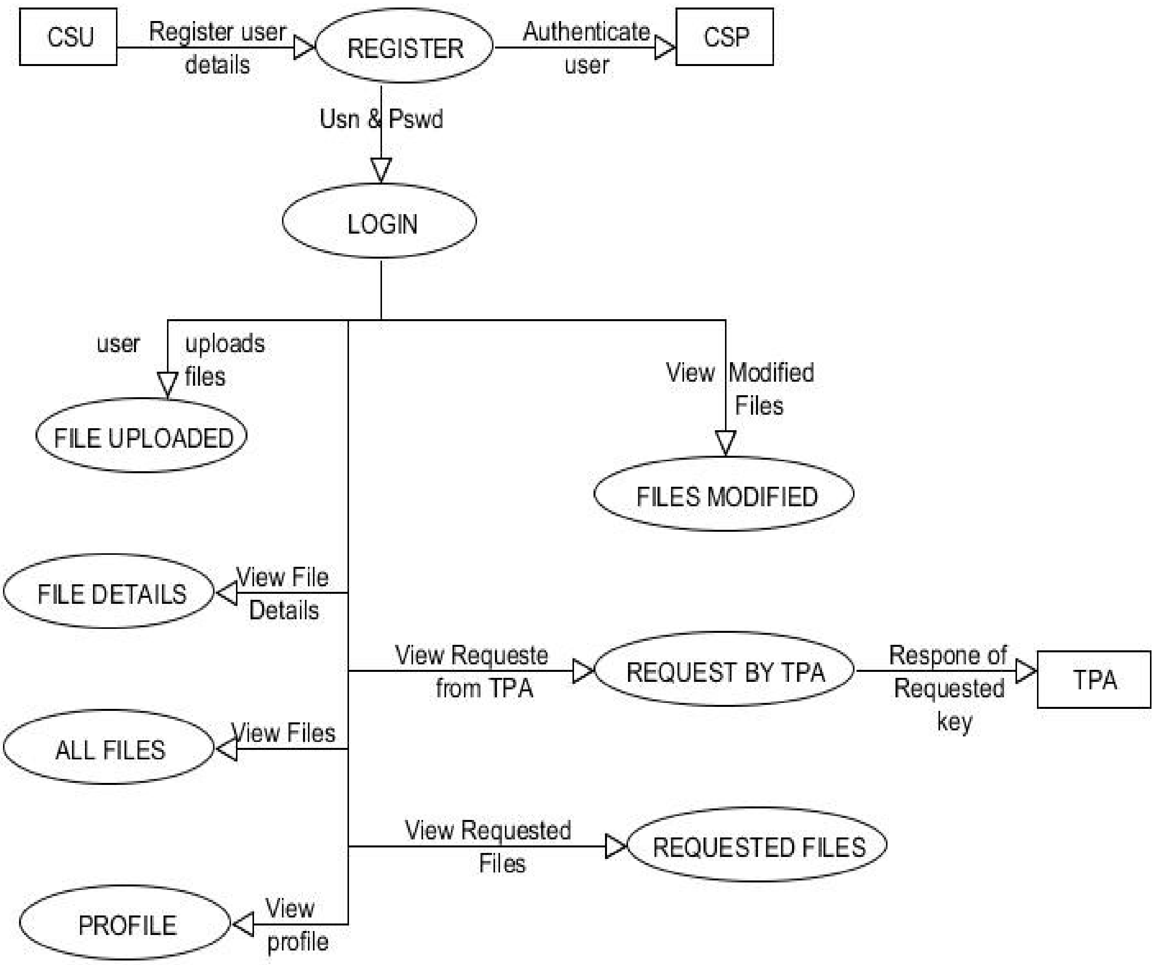


Fig. 5.1.1 Flow diagram for a Cloud Service User

5.1.2 CSP

The cloud service provider is the one who provides the customers with the cloud computing services. Through outsourcing data storage scenario, data owners delegate the storage and management of their data to a CSP in exchange for pre-specified fees metered in GB/month. Such outsourcing of data storage enables owners to store more data on remote servers than on private computer systems. Moreover, the CSP often provides better disaster recovery by replicating the data on multiple servers across multiple data centers achieving a higher level of availability. Thus, many authorized users are allowed to access the remotely stored data from different geographic locations making it more convenient for them. The CSP is responsible for the registration of any user. This can be done by contractual agreements. Also the TPAs need to save a deal with the CSP before registering themselves for the services. The CSP can view the registered users and TPAs through the interface provided. CSP has the right to reject the registration of any user if any terms or conditions are not met with. The following are some of the details with which a CSP can work with:

1. User Details: CSP can see all the detail about user.
2. Modified files: Carries all the modified files.
3. Registered TPA: CSP can see all the registered TPA.
4. Pending Files: It can find all the details about pending files.

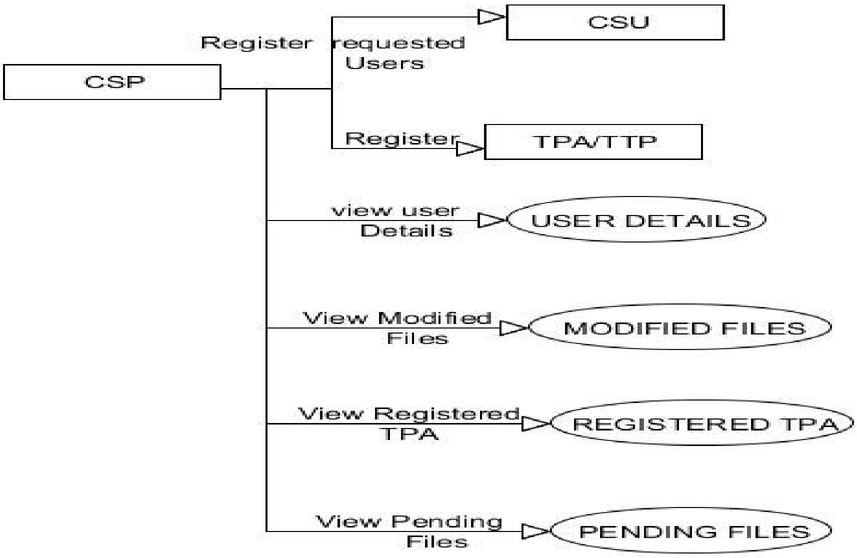


Fig 5.1.2 Cloud Service Provider

5.1.3 TPA/TTP

Trusted Third Party or Third Party Authenticator a trusted third party (TTP) is an entity which facilitates interactions between two parties who both trust the third party. The Third Party reviews all critical transaction communications between the system components, based on the ease of creating fraudulent digital content. In TTP models, the relying parties use this trust to secure their own interactions. Here the third party acts as the entity between the cloud service provider and the cloud service user. We are using the acronyms TPA and TTP interchangeably. The TPA needs to first get registered with the cloud service provider. After the confirmation of registration, the TPA can sign in and proceed with the file verification activities. Below are some of the actions which can be taken by the TPA.

1. Login: TTA/TPA has to login first using user name and password.
2. All Files: It can view all the files that has been uploaded.
3. Download: TPA/TTP can download the files to verify them.
4. Reupload files: It can also reupload the modified files.

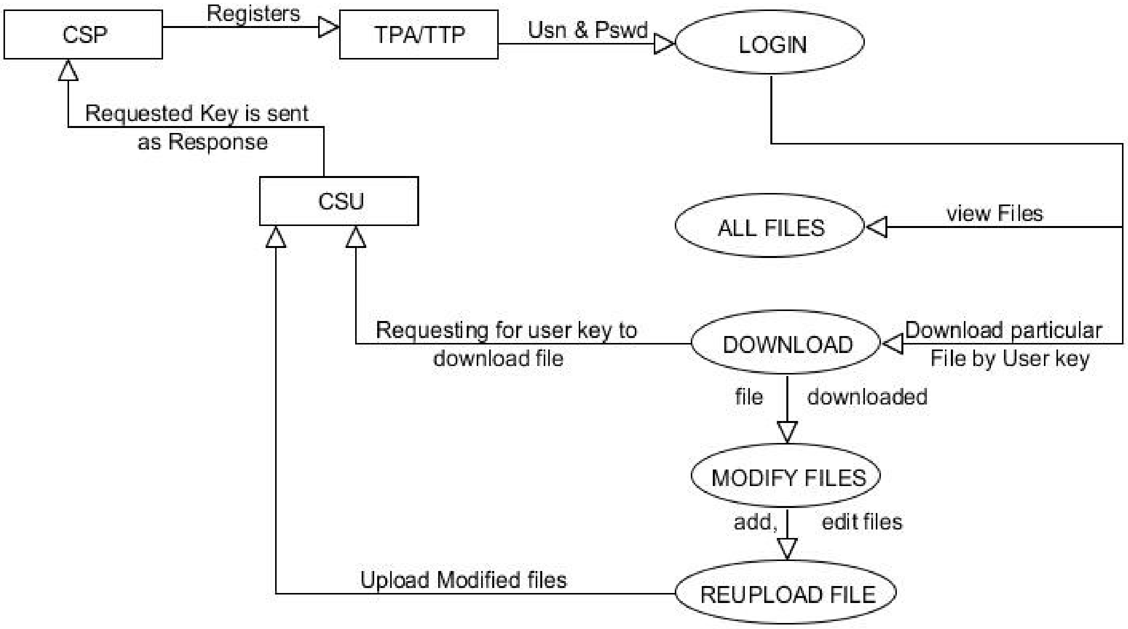


Fig 5.1.3 Flow diagram for Trusted Third Party

5.2 USE CASE DIAGRAM

A use case diagram at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as well. The use case diagrams for the CSU, CSP and the TPA are as follows. The use case diagrams show the activities that each of the system component can provide when they are logged in.

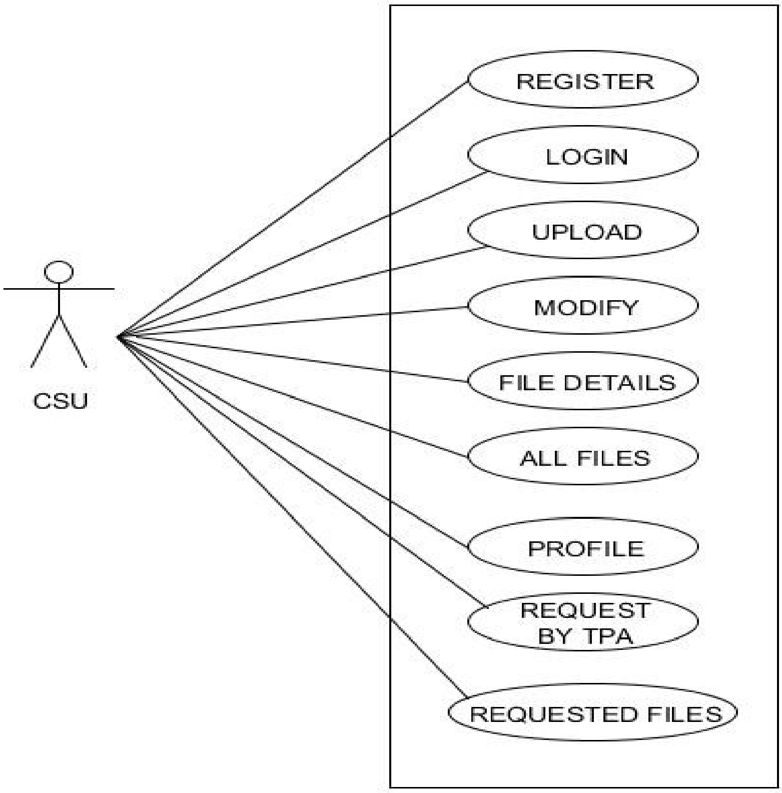


Fig 5.2.1 - Use case diagram for the cloud service user

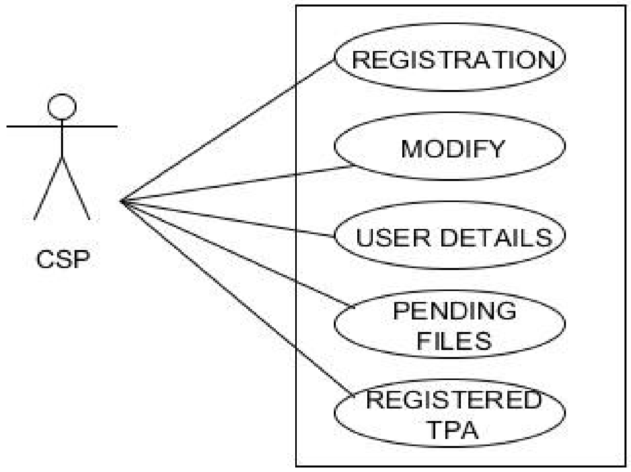


Fig 5.2.2 - Use case diagram for the cloud service provider.

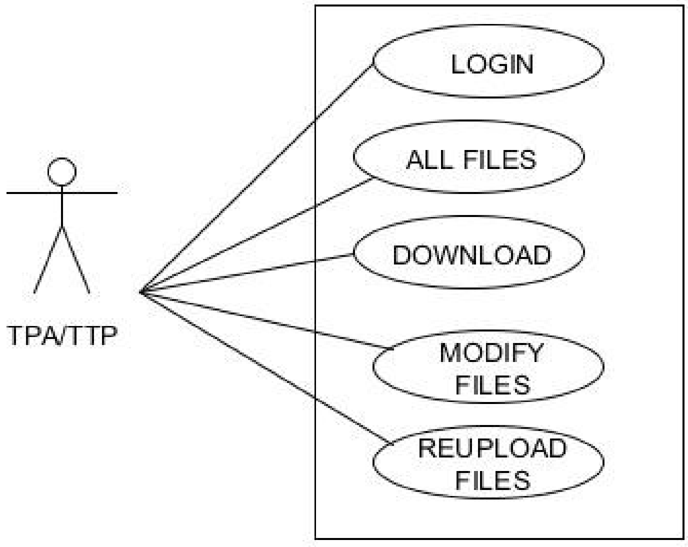


Fig 5.2.3 - Use case diagram for the trusted third party

5.3 E-R DIAGRAM

In software engineering, an entity–relationship model (ER model) is a data model for describing the data or information aspects of a business domain or its process requirements, in an abstract way that lends itself to ultimately being implemented in a database such as a relational database. The main components of ER models are entities (things) and the relationships that can exist among them. Entity–relationship modeling was developed by Peter Chen and published in a 1976 paper. However, variants of the idea existed previously, and have been devised subsequently such as super type and subtype data entities and commonality relationships

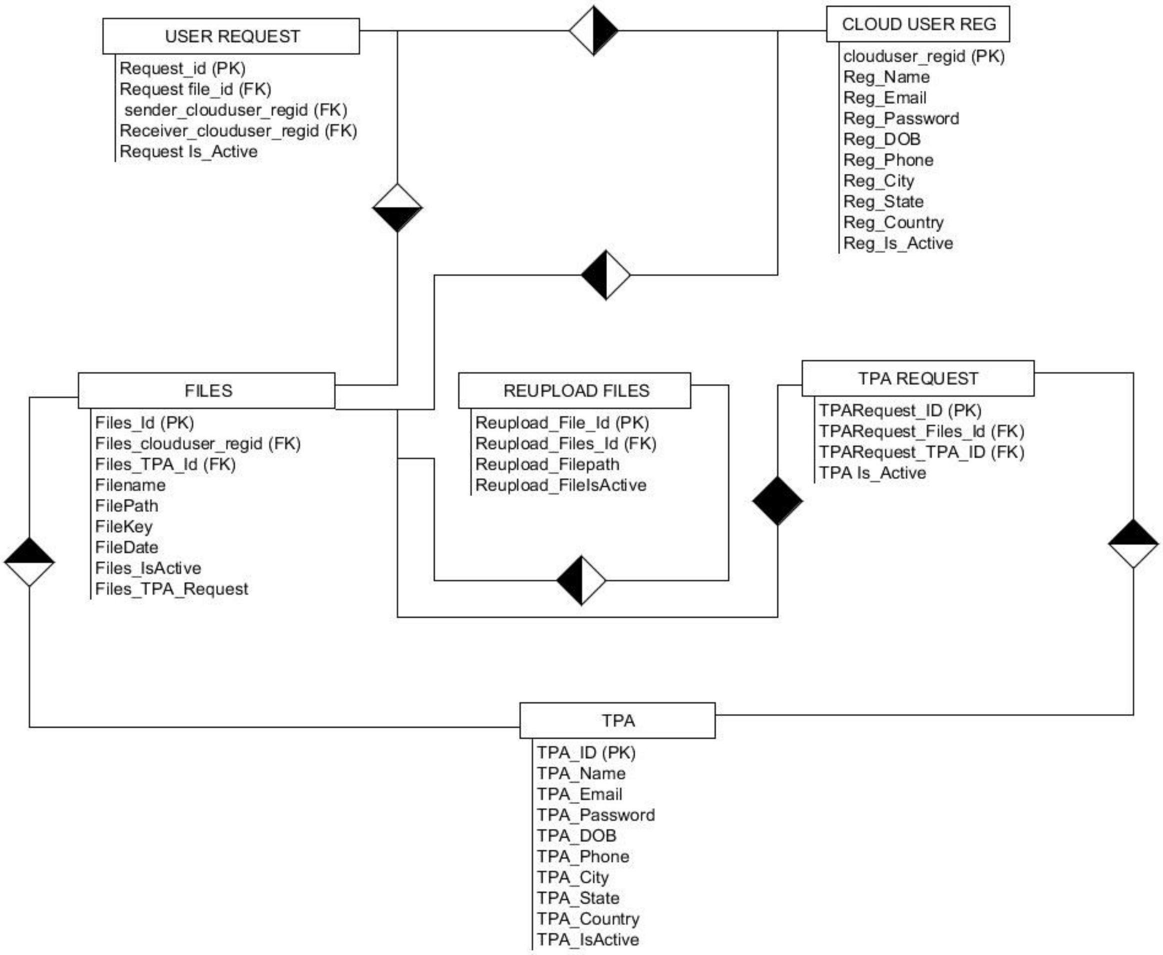


Fig 5.4.1 E.R. diagram

Chapter 6

IMPLEMENTATION

Implementation is the realization of an application, or execution of a plan, idea, model, design, specification, standard, algorithm, or policy.

6.1 MODULES DESCRIPTION

The cloud computing storage model considered in this work consists of three main components:

The implementations are based on these modules and consist of the separate functionalities of these components:

6.1.1 CLOUD SERVICE USERS

DATA OWNER:

In this module if a owner of data has to store data on a cloud server. He/she should register their details first. These details are maintained in a Database. Then he has to upload the file in a file database. The files which are stored in a database are in an encrypted form. Authorized users can only decode it.

DATA USER:

In this module if a user wants to access the data which is stored in a cloud server. He/she should register their details first. These details are maintained in a Database.

The Cloud Service User organizes and generates sensitive data. The CSU has to first login to his account and if he doesn't have his account he has to register his account first into the Cloud Service Provider.

The C# code for registering the account is :

```
protected voidUser_Register(object sender, EventArgs e)
   {
```

```
DataTabledt = new DataTable();
dt = DataAccess.GetInstance().Cloud_UserReg_addCloudUserInfo(TxtName.Text.Trim(),
TxtEmail.Text.Trim(), TxtPassword.Text.Trim(),TxtDOB.Text.Trim(), TxtPhone.Text.Trim(),
TxtCity.Text.Trim(), TxtState.Text.Trim(), TxtCountry.Text.Trim());
if (dt.Rows.Count> 0)
      {
if (dt.Rows[0]["ret"].Equals(1))
         {
string User = TxtEmail.Text;
string[] words = User.Split('@');

stringsubPath = Server.MapPath("~/temp/" + words[0] + "/");
boolisExists = System.IO.Directory.Exists(subPath);

if (!isExists)
            {
System.IO.Directory.CreateDirectory(subPath);
            }
Utils.ShowAlertMessage("Registered Successfully");
Response.Redirect("~/CloudUserLogin.aspx");
         }
else
         {
Utils.ShowAlertMessage("Email have been registered already!!");
         }
      }
   }
```

The user enters the details asked in the form, and these details are stored in the database. If the user doesn't enter a field, an error message is displayed.

If the user is already registered he needs to login to his account.

C# code for login is:

```
protected void User_Login(object sender, EventArgs e)
   {
DataTabledt = new DataTable();
dt = DataAccess.GetInstance().CloudUserReg_Authencate(TxtEmail.Text.Trim(),
TxtPassword.Text.Trim());
if(dt.Rows.Count> 0)
        {
Session["Email"] = TxtEmail.Text;
Response.Redirect("UploadFile.aspx");
        }
else {
Utils.ShowAlertMessage("Incorrect Email and Password");
        }
   }
```

Upon login, the details are verified from the database, if they are found to be matching, the user successfully logs in, else it shows an error.

When the user login successfully then he can upload the file.

The code for uploading a file is:

```
protected void BtnSubmit_Click(object sender, EventArgs e)
   {
stringFileName = TxtFileName.Text.Trim();
uploadF up = new uploadF();
stringUploadFile = up.savefile(FileUp);
int TPA = Convert.ToInt32(DropDownTPA.SelectedValue);
string Key = GetKeyGenerate(9);
stringfileName = Path.GetFileNameWithoutExtension(UploadFile);
stringfileExtension = Path.GetExtension(UploadFile);
string input = Server.MapPath("~/temp/") + fileName + fileExtension;
string output = Server.MapPath("~/temp/") + fileName + "_enc" + fileExtension;
FileUp.SaveAs(input);
this.Encrypt(input, output, Key);
stringuploadFilepath = fileName + "_enc" + fileExtension;
string User = Session["Email"].ToString();
if (DataAccess.GetInstance().Files_add_File(User, TPA, FileName, uploadFilepath, Key))
        {
Utils.ShowAlertMessage("File Uploaded Successfully");
TxtFileName.Text = "";
DropDownTPA.ClearSelection();
        }
else
        {
Utils.ShowAlertMessage("Server Down Try Later");
        }
    }
```

The user can also do the following work:

- View all the approved files by the TTP as well as the pending files.
- Send request as well as accept the request for the respective file from other cloud users or TTP respectively.

```
protected void GridFileViewRowCommand(object sender, GridViewCommandEventArgs e)
    {
if (e.CommandName.Equals("Accept"))
        {
if
(DataAccess.GetInstance().TPARequest_Update_TPA_Request(Convert.ToInt32(e.CommandArgument)))
            {
MailSending mail = new MailSending();
GridViewRow row = (GridViewRow)(((ImageButton)e.CommandSource).NamingContainer);
```

```
 Label LblUserName = (Label)row.Cells[0].FindControl("LblUserName");
HiddenFieldHiddenKey = (HiddenField)row.Cells[1].FindControl("HiddenKey");
Label LblFileName = (Label)row.Cells[2].FindControl("LblFileName");
Label LblEmail = (Label)row.Cells[3].FindControl("LblEmail");
HiddenFieldHiddenCloudUserName =
(HiddenField)row.Cells[3].FindControl("HiddenCloudUserName");
string body = "Cloud User : " + HiddenCloudUserName.Value.ToString() + "\\n File Name : " +
LblFileName.Text + "\\n secret Key : " + HiddenKey.Value.ToString();
mail.SendMail(LblEmail.Text, body, "Cloud User Accepted File Request");
LoadGrid();
Utils.ShowAlertMessage("Key has been sent to TPA");
            }
        }
    }
```

The requests are being sent to the owner of the data, who then decides whether or not to allow the corresponding user to access the data. If so decided, the owner shares the key with the user.

- It can update the pre-uploaded file as well as update his own personal information.

```
protected void GridFileView_RowCommand(object sender, GridViewCommandEventArgs e)
    {
if (e.CommandName.Equals("Edit"))
        {
GridViewRow row = (GridViewRow)(((ImageButton)e.CommandSource).NamingContainer);
HiddenFieldHiddenKey = (HiddenField)row.Cells[0].FindControl("HiddenKey");
Label lblTPANAme = (Label)row.Cells[0].FindControl("LblTPAName");
Label LblFileName = (Label)row.Cells[0].FindControl("LblFileName");
Label LblEmail = (Label)row.Cells[0].FindControl("LblEmail");
HiddenFieldHiddenCloudUserName =
(HiddenField)row.Cells[0].FindControl("HiddenCloudUserName");
Response.Redirect("EditFileDetails.aspx?ID=" + e.CommandArgument + "&Key=" +
HiddenKey.Value + "&TPAName=" + lblTPANAme.Text + "&FileName=" +
LblFileName.Text + "&Email=" + LblEmail.Text + "&UserName=" +
HiddenCloudUserName.Value);


        }
    }
```

The updated info is made to reflect back in the database for future reference. The updated files are reuploaded in the server.

6.1.2 TTP (TRUSTED THIRD PARTY):

In this module TTP has monitors the data owners file by verifying the data owner's file and stored the file in a database .Also TTP checks the CSP(CLOUD SERVICE PROVIDER), and find out whether the CSP is authorized one or not.

It is an entity who is trusted by all other system components, and has expertise and capabilities to detect and specify dishonest parties.

TTP has to login in to his account. Upon first time login, the TTP needs to get registered. This registration needs to be approved by the CSP.

TPA login code:

```
protected void TPA_Login(object sender, EventArgs e)
    {
DataTabledt = new DataTable();
dt = DataAccess.GetInstance().TPA_Authenticate(TxtUserName.Text.Trim(),
TxtPassword.Text.Trim());
if (dt.Rows.Count> 0)
        {
Session["TPAEmail"] = TxtUserName.Text;
Response.Redirect("TPAGetFiles.aspx");
        }
else
```

```
        {
Utils.ShowAlertMessage("Invalid Username and Password");
        }
    }
```

The TTP has following work to do:

- Has to approve the fresh uploaded files by the cloud user as well as approve the files updated by the cloud user and after verifying the respective file it again uploads the same file and which becomes trusted data as well as visible to other users too.
- Keep the record of the file details so that it is able to determine the guilty party at the time of a dispute.

```
public void LoadGrid()
    {
string Email = Session["TPAEmail"].ToString();
DataTabledt = new DataTable();
dt = DataAccess.GetInstance().Files_Get_TPA_Files(Email);
if (dt.Rows.Count> 0)
        {
GridFileView.DataSource = dt;
GridFileView.DataBind();
        }
    }
protected void GridFileView_RowCommand(object sender, GridViewCommandEventArgs e)
    {
if (e.CommandName.Equals("Send"))
        {
if
(DataAccess.GetInstance().TPARequest_add_TPA_Request(Convert.ToInt32(e.CommandArgu
ment)))
            {
```

```
LoadGrid();
Utils.ShowAlertMessage("Request Sent");
          }
       }
   }
```

- TTP can send the request for the files which he need verification to the user, and receive the decryption key of that file via email, from that user.

File request by TPA:

```
public void View_request_files()
   {
string Email = Session["TPAEmail"].ToString();
DataTabledt = new DataTable();
dt = DataAccess.GetInstance().Files_Get_TPA_Requsted_Sent_Files(Email);
if (dt.Rows.Count> 0)
       {
GridFileView.DataSource = dt;
GridFileView.DataBind();
       }
   }
```

File request received by TPA:

```
public voidRequest_received()
   {
string Email = Session["TPAEmail"].ToString();
DataTabledt = new DataTable();
dt = DataAccess.GetInstance().Files_Get_TPA_Requst_Accepted_Files(Email);
if (dt.Rows.Count> 0)
       {
GridFileView.DataSource = dt;
GridFileView.DataBind();
```

```
        }
    }
protected void GridFileView_RowCommand(object sender, GridViewCommandEventArgs e)
    {
if (e.CommandName.Equals("Download"))
        {
Response.Redirect("EnterKeyToDownload.aspx?File_Id=" +
Convert.ToInt32(e.CommandArgument));
        }
    }
```

The file request goes to the owner of the file. The file owner has to send the key to the TTP in order for its file to get verified.

6.1.3 CSP (CLOUD SERVICE PROVIDER)

A Cloud Service Provider (CSP) who manages cloud servers and provides storage space on a rent basis in its infrastructure to store the owner's files and make them available for authorized users.

A CSP who manages cloud servers and provides paid storage space on its infrastructure to store the owner's files and make them available for authorized users.

The CSP has following work to do:

- CSP can check the user registration as well as can approve the user to avail the cloud facility or reject the user if he does not meet certain conditions and can also view all the registered users on the Cloud and can remove them if their term period is over.

Code for approving a new user:

```
protected void grvADview_RowCommand(object sender, GridViewCommandEventArgs e)
    {
if (e.CommandName.Equals("Approve"))
        {
GridViewRow row = (GridViewRow)(((ImageButton)e.CommandSource).NamingContainer);
HiddenField Email = (HiddenField)row.Cells[0].FindControl("hndemail");
if
(DataAccess.GetInstance().Cloud_UserReg_Approve_Cloud_User(Convert.ToInt32(e.Command
Argument)))
         {
MailSending mail = new MailSending();
mail.SendMail(Email.Value,"You Have Sucessfully Registered to Cloud","Cloud Service
Provider");
Utils.ShowAlertMessage("User Approved");
Response.Redirect("CheckUserReg.aspx");
         }
      }
```

- Same as the procedure for the cloud users the CSP can do the same with the TTP registration approval as well as view all verified TTP.

Code for approving the TTP:

```
protected void Approve_TTP(object sender, GridViewCommandEventArgs e)
   {
if (e.CommandName.Equals("Approve"))
      {
GridViewRow row = (GridViewRow)(((ImageButton)e.CommandSource).NamingContainer);
         Label Email = (Label)row.Cells[0].FindControl("LblEmail");
if (DataAccess.GetInstance().Cloud_TPA_Approve(Convert.ToInt32(e.CommandArgument)))
         {
MailSending mail = new MailSending();
mail.SendMail(Email.Text, "You Have Sucessfully Registered to Cloud", "Cloud Service
Provider");
Utils.ShowAlertMessage("TPA Approved");
Response.Redirect("CheckTPA.aspx");
         }
      }
   }
```

- CSP can see list of all the files which are uploaded on his cloud server and by which user. He can also view the list of all the files which are pending by the TTP to be approved as well as see the list of files which are still pending to be approved by the TTP.

Code for viewing the files:

```
protected void GridFileView_RowCommand(object sender, GridViewCommandEventArgs e)
   {
if (e.CommandName.Equals("Download"))
       {
GridViewRow row = (GridViewRow)(((ImageButton)e.CommandSource).NamingContainer);
HiddenFieldHiddenKey = (HiddenField)row.Cells[0].FindControl("HiddenKey");
stringpname = e.CommandArgument.ToString();
stringfileName = Path.GetFileNameWithoutExtension(pname);
stringfileExtension = Path.GetExtension(pname);

string input = Server.MapPath("~/temp/") + fileName + fileExtension;
string output = Server.MapPath("~/temp/") + fileName + "_dec" + fileExtension;

if ((System.IO.File.Exists(output)))
           {


           }
else
           {
this.Decrypt(input, output, HiddenKey.Value);
           }

Response.ContentType = "text/txt";
Response.AppendHeader("Content-Disposition", "attachment; filename=" + pname);
Response.TransmitFile(output);
Response.End();

if (output != null || output != string.Empty)
            {
if ((System.IO.File.Exists(output)))
                {
System.IO.File.Delete(output);
                }
            }
        }
```

6.2 OTHER IMPLEMENTATION ASPECTS

6.2.1 ENCRYPTION AND DECRYPTION OF THE FILES

We are providing each file with a unique ID which includes the time (HH:MM:SS) and date (DD/MM/YYYY) at the moment when the file is uploaded which keep all the files unique from each other.

Encryption algorithm that we used is AES using a key length as 9 which is alpha-numeric, which is generated using a key generation algorithm.

Encryption code:

```
private void Encrypt(string inputFilePath, string outputfilePath, string EncryptionKey)
    {
        //string EncryptionKey = "MAKV2SPBNI99212";
using (Aesencryptor = Aes.Create())
        {
            Rfc2898DeriveBytes pdb = new Rfc2898DeriveBytes(EncryptionKey, new byte[] {
0x49, 0x76, 0x61, 0x6e, 0x20, 0x4d, 0x65, 0x64, 0x76, 0x65, 0x64, 0x65, 0x76 });
encryptor.Key = pdb.GetBytes(32);
encryptor.IV = pdb.GetBytes(16);
using (FileStreamfsOutput = new FileStream(outputfilePath, FileMode.Create))
            {
using (CryptoStreamcs = new CryptoStream(fsOutput, encryptor.CreateEncryptor(),
CryptoStreamMode.Write))
                {
using (FileStreamfsInput = new FileStream(inputFilePath, FileMode.Open))
                    {
int data;
while ((data = fsInput.ReadByte()) != -1)
cs.WriteByte((byte)data);
}
                }
            }
        }
    }
```

Key Generation Code

```
public stringKey_Generate(int length)
    {
char[] chars = "1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZ".ToCharArray();
string password = string.Empty;
        Random random = new Random();
for (int i = 0; i < length; i++)
        {
int x = random.Next(1, chars.Length);
if (!password.Contains(chars.GetValue(x).ToString()))
password += chars.GetValue(x);
else
i--;
        }
return password;
   }
```

Decryption Code

Decrypt method accepts the key from the User/TTP sent to him via email and uses the same key to decrypt the specific file.

Code to decrypt the file:

```
private void Decrypt(string inputFilePath, string outputfilePath, string EncryptionKey)
   {

using (Aesencryptor = Aes.Create())
      {
         Rfc2898DeriveBytes pdb = new Rfc2898DeriveBytes(EncryptionKey, new byte[] {
0x49, 0x76, 0x61, 0x6e, 0x20, 0x4d, 0x65, 0x64, 0x76, 0x65, 0x64, 0x65, 0x76 });
encryptor.Key = pdb.GetBytes(32);
encryptor.IV = pdb.GetBytes(16);
using (FileStreamfsInput = new FileStream(inputFilePath, FileMode.Open))
         {
using (CryptoStreamcs = new CryptoStream(fsInput, encryptor.CreateDecryptor(),
CryptoStreamMode.Read))
            {
using (FileStreamfsOutput = new FileStream(outputfilePath, FileMode.Create))
               {
int data;
while ((data = cs.ReadByte()) != -1)
                  {
fsOutput.WriteByte((byte)data);
                  }
               }
            }
         }
```

```
    }
  }
```

6.2.2 TRIGGERING THE EMAIL

This segment helps in generating and triggering the email when the file request is accepted but user to download the file or to verify the file, using the fixed email id i.e. “bits.clouds@gmail.com” .

```
public void SendMail(string emailiduser, string Body, string subjecttext)
  {
stringemailid = emailiduser;//to user
string Subject = subjecttext;
stringmailtext = Body;
System.Net.Mail.MailMessagemsg = new System.Net.Mail.MailMessage();
msg.To.Add(emailid);
     //   msg.To.Add(emailidinfo);
msg.From = new MailAddress("bits.clouds@gmail.com", Subject,
System.Text.Encoding.UTF8);
msg.Subject = Subject;
msg.SubjectEncoding = System.Text.Encoding.UTF8;
msg.Body = mailtext;
msg.BodyEncoding = System.Text.Encoding.UTF8;
msg.IsBodyHtml = true;
msg.Priority = MailPriority.High;

     //Add the Creddentials
SmtpClient client = new SmtpClient();
client.Credentials = new System.Net.NetworkCredential("bits.clouds@gmail.com",
"4ni10cs045");
client.Port = 25;//or use 587
client.Host = "smtp.gmail.com";
client.EnableSsl = true;
objectuserState = msg;
try
        {
client.Send(msg);
        }
catch (Exception ex)
        {
        }
    }
```

6.2.3 FILE UPLOADING AND SAVING IN THE CLOUD SERVER:

This method helps in uploading different types of files and also approves which type of files can be uploaded into the cloud server.

```
public string savefile(FileUploadFileUpload)
    {
stringfname = "";
if (FileUpload.HasFile)
        {
string filename = DateTime.Now.ToString("MMddyyyyHHmmss");
filename += Path.GetFileName(FileUpload.FileName);
string extension = System.IO.Path.GetExtension(filename);
try
            {
                if (extension == ".txt" || extension == ".doc" || extension == ".DOC" || extension ==
".docx" || extension == ".DOCX" || extension == ".TXT")
                {
FileUpload.SaveAs(System.Web.HttpContext.Current.Server.MapPath("~/temp/") + filename);
return filename;
                }
            }
catch (Exception ex)
            {
Utils.ShowAlertMessage("Upload status: The file could not be uploaded.");
            }
        }
else
        {
Utils.ShowAlertMessage("Select File to Upload");
        }
return "file missing";
    }
```

Chapter 7

TESTING

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include, but are not limited to the process of executing a program or application with the intent of finding software bugs (errors or other defects).

7.1 SYSTEM TESTING:

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

7.2 TYPES OF TESTS:

7.2.1 UNIT TESTING:

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

7.2.2 INTEGRATION TESTING:

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

7.2.3 FUNCTIONAL TEST:

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

7.2.4 BLACK BOX TESTING:

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

7.3 TEST STRATEGY AND APPROACH:

Field testing will be performed manually and functional tests will be written in detail.

7.3.1 TEST OBJECTIVES:

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

7.3.2 FEATURES TO BE TESTED

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

7.4 ACCEPTANCE TESTING:

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

7.5 TEST CASES

Test case ID	Test case name	Test case description	Test steps				Test status P/F
			Step	I/p given	Expected o/p	Actual o/p	
TC01	Registration Of user	To verify that the user has entered all the required Details	Enter Required details	User details	Registration is Successful.	Registration is Successful.	Pass
	Registration Of user	To verify that the user has entered all the required Details	Enter Required details	User details	Registration is Successful.	Error Enter all the required details	Fail
TC02	Login	To check whether the User has entered valid usn&pswd	Enter the Username & Password	Valid Username & password	Login Successfully	Login Successfully	Pass
	Login	To check whether the User has entered valid usn&pswd	Enter the Username & Password	Invalid Username & Password	Login Successfully	Error message Enter the valid Usn&Pswd	Fail
TC03	File Upload	Validate the Uploaded file is correct	Upload text,doc file	Upload file	File Uploaded Successfully	File Uploaded Successfully	Pass
	File Upload	Validate the	Upload	Upload file	File	Error	Fail

		Uploaded file is correct	image, video, audio &pdf file		Uploaded Successfully	Message Upload Correct format file	
TC04	Modified Files	To view the modified file	Modified files by ttp can be checked	Select modified files	Modified Files are displayed To the user	Files are viewed By the user	Pass
	Modified Files	To view the modified file	If files are not available	Select modified files	Modified Files are displayed to the user	Error message No files	Fail
TC05	All files details	To view the number of files and its details	View all files & details	Select files & details option	Files and its complete details are displayed	Files and its complete details are displayed	Pass
	All files details	To view the number of files and its details	If files are not available	Select files & details option	Files and its complete details are displayed	Error message No files Exists	Fail
TC06	Requested key & Files	To check that the requested key & files are displayed	View requested files & keys From TPA	Select requested file & key option	Requested files & keys from the TPA are displayed to the user	Requested files & keys from the TPA are displayed to the user	Pass

	Requested key & Files	To check that the requested key & files are displayed	If there is no request sent from the TPA	Select requested file & key option	Requested files & keys from the TPA are displayed to the user	Error Message No Request is found	Fail
TC07	Register Requested User	To confirm the Registration of Requested user	Register requested User	Confirm Registration	User Registered Successfully	User Registered Successfully	Pass
	Register Requested User	To confirm the Registration of Requested user	If there is no requested user	Confirm Registration	User Registered Successfully	Error Message No requested user	Fail
TC08	View file,user&tpa	To check & view that the User details, modified , pending files and registered TPA exists	View available files user, &TPa	Select the respective option	Pending, Modified files ,User details & Registered Tpa's are displayed	Pending, Modified files ,User details & Registered Tpa's are displayed	Pass
	View file,user&tpa	To check & view that the User details, modified , pending files and registered TPA exists	If files ,users &Tpa are not existing	Select the respective option	Pending, Modified files ,User details & Registered Tpa's are displayed	Error Message Not found	Fail
TC09	Download	To check that	Using key	Enter the key &	File	File	Pass

		the TPA can download user uploaded files	file can be downloaded	Download	downloaded successfully	downloaded Successfully	
	Download	To check that the TPA can download user uploaded files	If key is not sent from User or invalid key	Enter the key or without key & Download	File downloaded successfully	Error Enter the valid key	Fail
TC10	Modification By TPA	To check that the downloaded files are modified (add,edit,update)	User uploaded File is downloaded & modified	Modify(add,edit& update)	File Modified Successfully	File Modified Successfully	Pass
	Modification By TPA	To check that the downloaded files are modified (add,edit,update)	If file is not downloaded	Modify(add,edit& update)	File Modified Successfully	Error No files to modify	Fail

CONCLUSION

Outsourcing data to remote servers has become a growing trend for many organizations to alleviate the burden of local data storage and maintenance. In this project we have studied different aspects of outsourcing data storage: block-level data dynamic, newness, mutual trust, and access control.

We have proposed a cloud-based storage scheme which supports outsourcing of dynamic data, where the owner is capable of not only archiving and accessing the data stored by the CSP, but also updating and scaling this data on the remote servers. Through outsourcing data storage scenario, data owners delegate the storage and management of their data to a CSP in exchange for pre-specified fees metered in GB/month. Such outsourcing of data storage enables owners to store more data on remote servers than on private computer systems. Moreover, the CSP often provides better disaster recovery by replicating the data on multiple servers across multiple data centers achieving a higher level of availability. Thus, many authorized users are allowed to access the remotely stored data from different geographic locations making it more convenient for them. Moreover, in case of dispute regarding data integrity/newness, a TTP is able to determine the dishonest party.

FUTURE ENHANCEMENT

In the following, we identify some critical security and privacy issues in cloud computing that need immediate attention for ubiquitous adoption of this technology:

Authentication and identity management:

By using cloud services, user can easily access their personal information and make it available to various services across the Internet. An identity management (IDM) mechanism can help authenticate users and services based on credentials and characteristics. An IDM system should be able to protect private and sensitive information related to users and processes. However, multi- tenant cloud environments can affect the privacy of identity information and isn't yet well understood.

Trust management and policy integration:

Although multiple service providers coexist in cloud and collaborate to provide various services, they might have different security approaches and privacy mechanisms. Hence, we must address heterogeneity among their policies. Cloud service providers might need to compose multiple services to enable bigger application services. Therefore, mechanisms are necessary to ensure that such a dynamic collaboration is handled securely and that security breaches are effectively monitored during the interoperation process.

REFERENCES

[1] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z.Peterson, and D. Song, “Provable Data Possession at UntrustedStores,”Proc. 14th ACM Conf. Computer Comm. Security,pp. 598-609, 2007.

[2] F. Sebe´, J. Domingo-Ferrer, A. Martinez-Balleste, Y. Deswarte, and J.-J. Quisquater, “Efficient Remote Data Possession Checking in Critical Information Infrastructures,”IEEE Trans. Knowledge Data Eng.,vol. 20, no. 8, pp. 1034-1038, Aug. 2008.

[3] G. Ateniese, R.D. Pietro, L.V. Mancini, and G. Tsudik, “Scalable and Efficient Provable Data Possession,”Proc. Fourth Int’l Conf. Security Privacy Comm. Networks,pp. 1-10, 2008.

[4] C. Erway, A. Ku¨pc¸u ¨,C.Papamanthou,andR.Tamassia, “Dynamic Provable Data Possession,” Proc. 16th ACM Conf. Computer Comm. Security,pp. 213-222, 2009.

[5] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, “Enabling Public Verifiability and Data Dynamics for Storage Security in Cloud Computing,”Proc. 14th European Conf. Research ComputerSecurity,pp. 355-370, 2009.

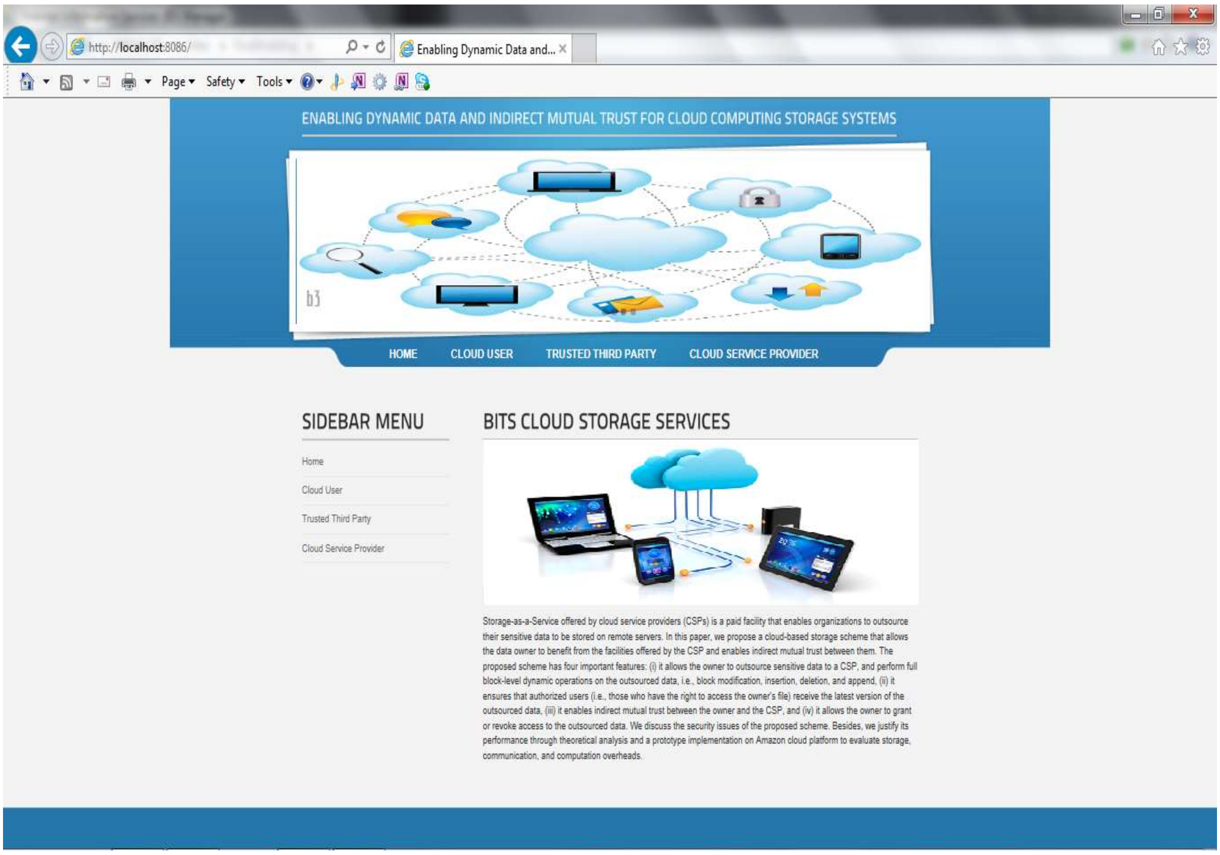
[6] A.F. Barsoum and M.A. Hasan, “Provable Possession andReplication of Data over Cloud Servers,” Technical Report 2010/32, Centre for Applied Cryptographic Research

[7] R. Curtmola, O. Khan, R. Burns, and G. Ateniese, “MR-PDP:Multiple-Replica Provable Data Possession,”Proc. 28th Int’l Conf.Distributed Computing Systems,pp. 411-420, 2008.

[8] A.F. Barsoum and M.A. Hasan, “On Verifying Dynamic MultipleData Copies over Cloud Servers,” Technical Report 2011/447, CryptologyEprint Archive, http://eprint.iacr.org/, 2011.

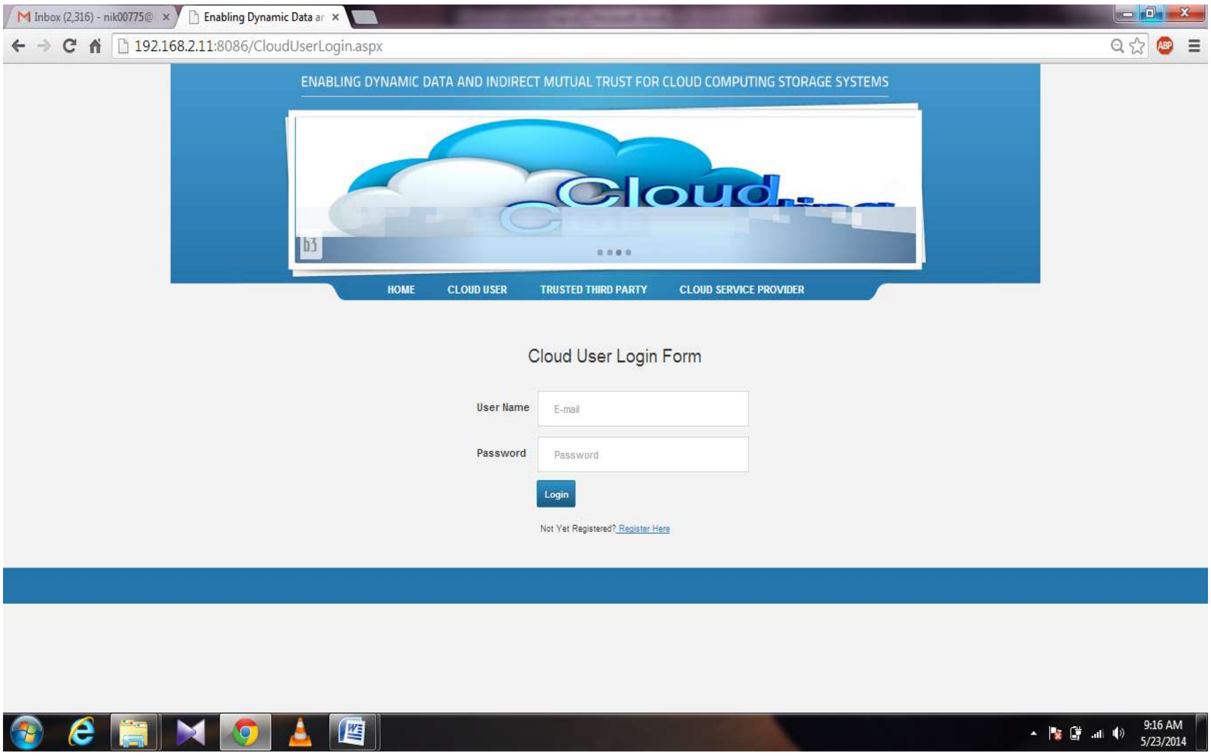
[9] K.D. Bowers, A. Juels, and A. Oprea, “HAIL: A High-Availabilityand Integrity Layer for Cloud Storage,” Proc. 16th ACM Conf.Computer Comm. Security,pp. 187-198, 2009.

SCREENSHOTS



Storage-as-a-Service offered by cloud service providers (CSPs) is a paid facility that enables organizations to outsource their sensitive data to be stored on remote servers. In this paper, we propose a cloud-based storage scheme that allows the data owner to benefit from the facilities offered by the CSP and enables indirect mutual trust between them. The proposed scheme has four important features: (i) it allows the owner to outsource sensitive data to a CSP, and perform full block-level dynamic operations on the outsourced data, i.e., block modification, insertion, deletion, and append, (ii) it ensures that authorized users (i.e., those who have the right to access the owner's file) receive the latest version of the outsourced data, (iii) it enables indirect mutual trust between the owner and the CSP, and (iv) it allows the owner to grant or revoke access to the outsourced data. We discuss the security issues of the proposed scheme. Besides, we justify its performance through theoretical analysis and a prototype implementation on Amazon cloud platform to evaluate storage, communication, and computation overheads.

S1. The landing page for the cloud services interface



Inbox (2,316) - nik00775@
Enabling Dynamic Data ar
192.168.2.11:8086/CloudUserLogin.aspx
ENABLING DYNAMIC DATA AND INDIRECT MUTUAL TRUST FOR CLOUD COMPUTING STORAGE SYSTEMS
Cloud
HOME
CLOUD USER
TRUSTED THIRD PARTY
CLOUD SERVICE PROVIDER
Login
9:16 AM
5/23/2014

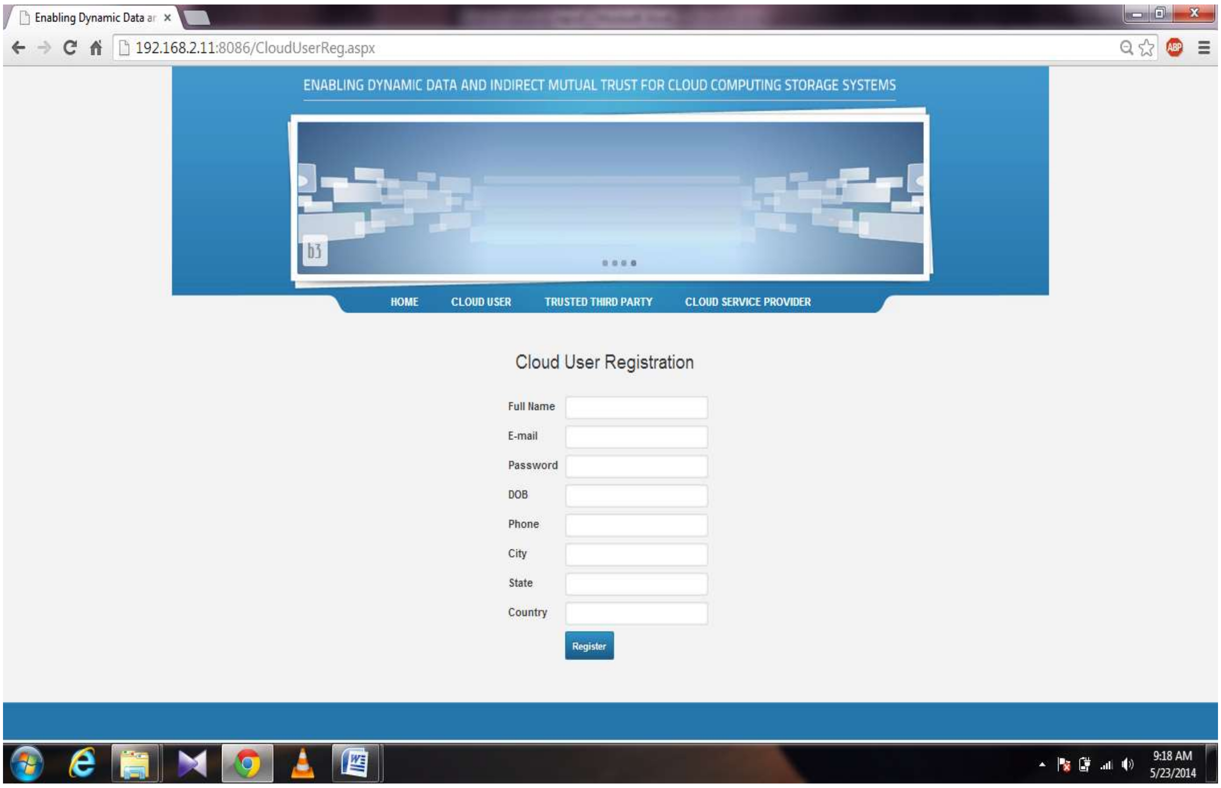
Cloud User Login Form

User Name E-mail

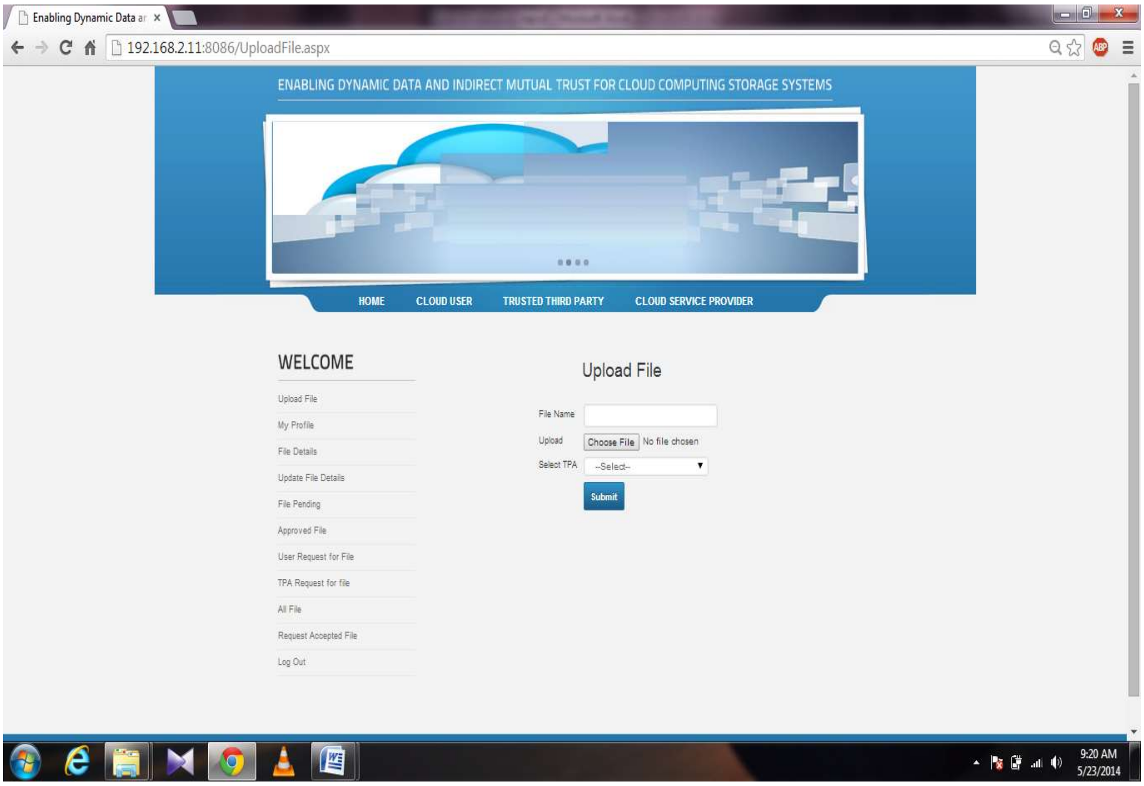
Password Password

Not Yet Registered? Register Here

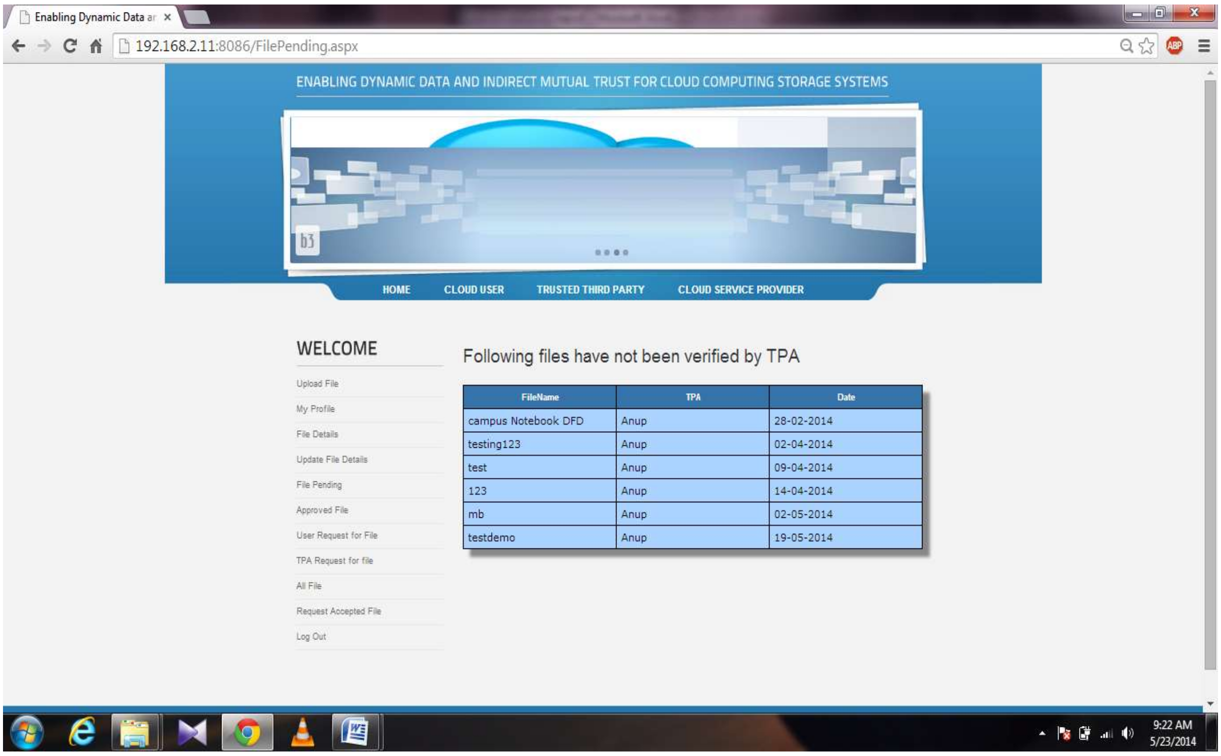
S2. Cloud User login page



S3. User Registration Form



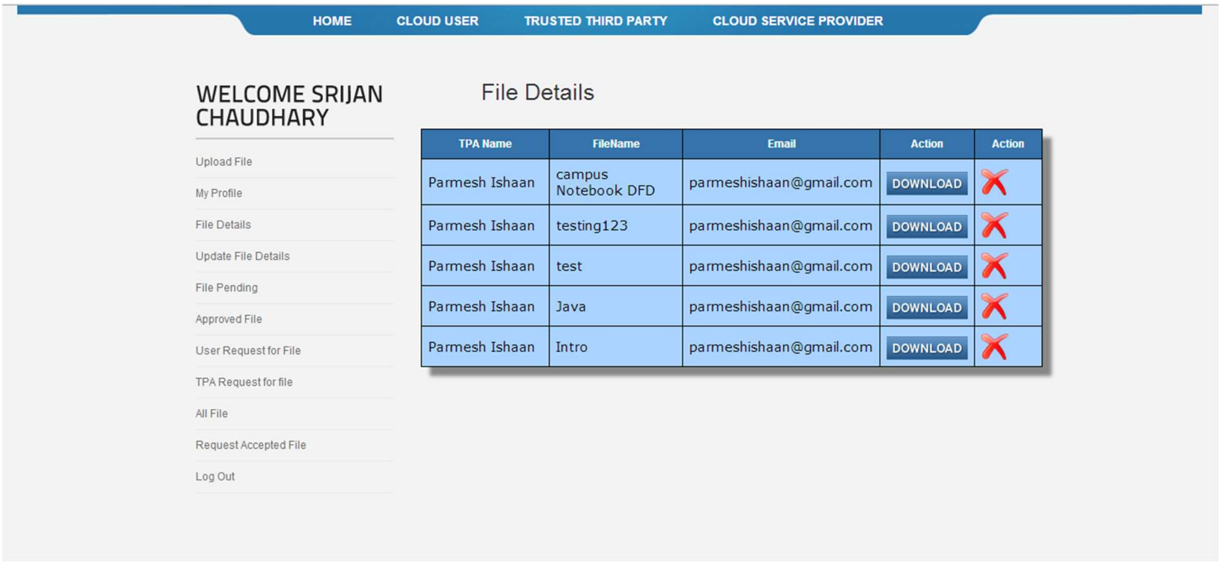
S4.File Uploading



Following files have not been verified by TPA

FileName	TPA	Date
campus Notebook DFD	Anup	28-02-2014
testing123	Anup	02-04-2014
test	Anup	09-04-2014
123	Anup	14-04-2014
mb	Anup	02-05-2014
testdemo	Anup	19-05-2014

S5. File to be verified by TPA

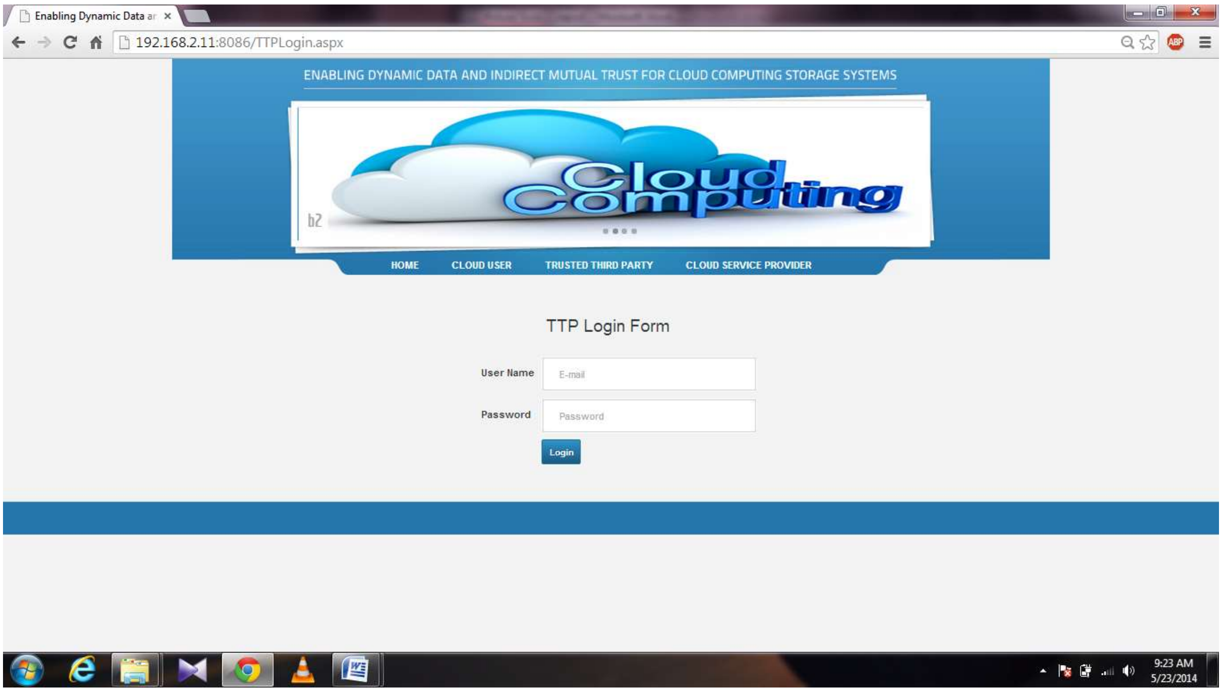


WELCOME SRIJAN CHAUDHARY

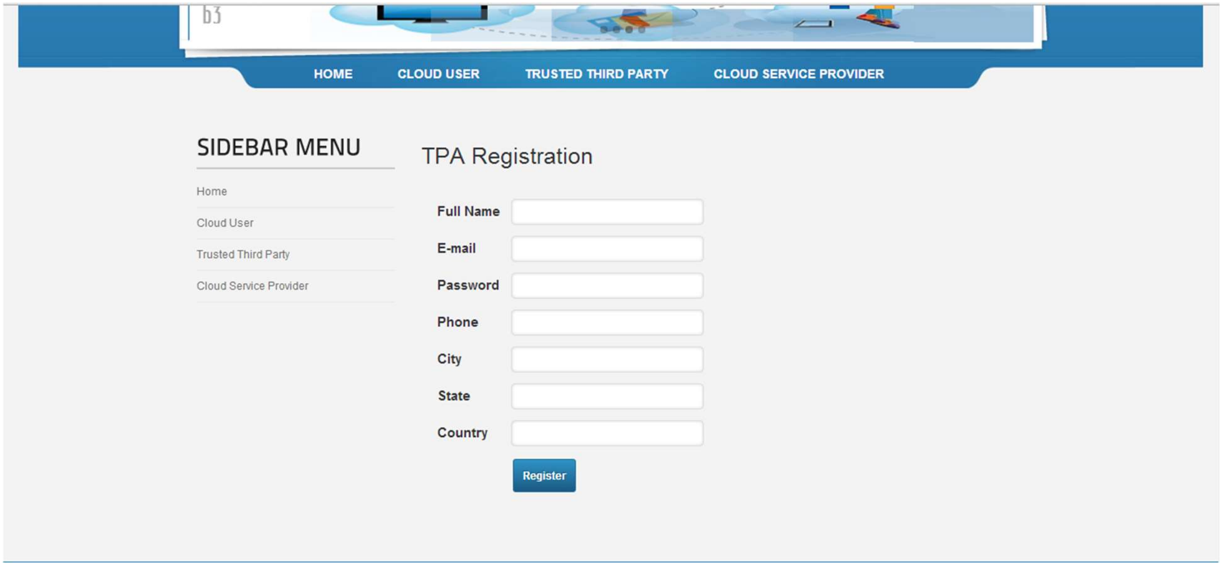
File Details

TPA Name	FileName	Email	Action	Action
Parmesh Ishaan	campus Notebook DFD	parmeshishaan@gmail.com	DOWNLOAD	✗
Parmesh Ishaan	testing123	parmeshishaan@gmail.com	DOWNLOAD	✗
Parmesh Ishaan	test	parmeshishaan@gmail.com	DOWNLOAD	✗
Parmesh Ishaan	Java	parmeshishaan@gmail.com	DOWNLOAD	✗
Parmesh Ishaan	Intro	parmeshishaan@gmail.com	DOWNLOAD	✗

S6. Files uploaded by the user.



S7. TTP Landing and login page.



TPA Registration

Full Name

E-mail

Password

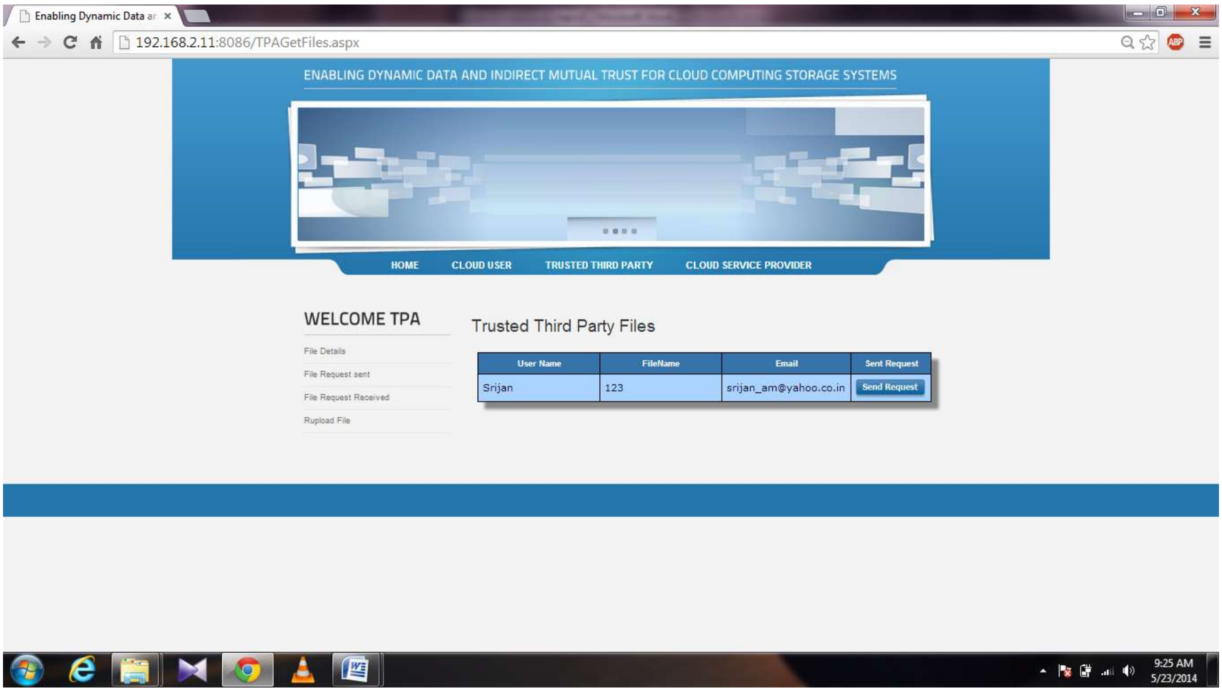
Phone

City

State

Country

S8. New TTP registration page.



Enabling Dynamic Data ar
192.168.2.11:8086/TPAGetFiles.aspx
ENABLING DYNAMIC DATA AND INDIRECT MUTUAL TRUST FOR CLOUD COMPUTING STORAGE SYSTEMS
HOME
CLOUD USER
TRUSTED THIRD PARTY
CLOUD SERVICE PROVIDER
9:25 AM
5/23/2014

WELCOME TPA

File Details

File Request sent

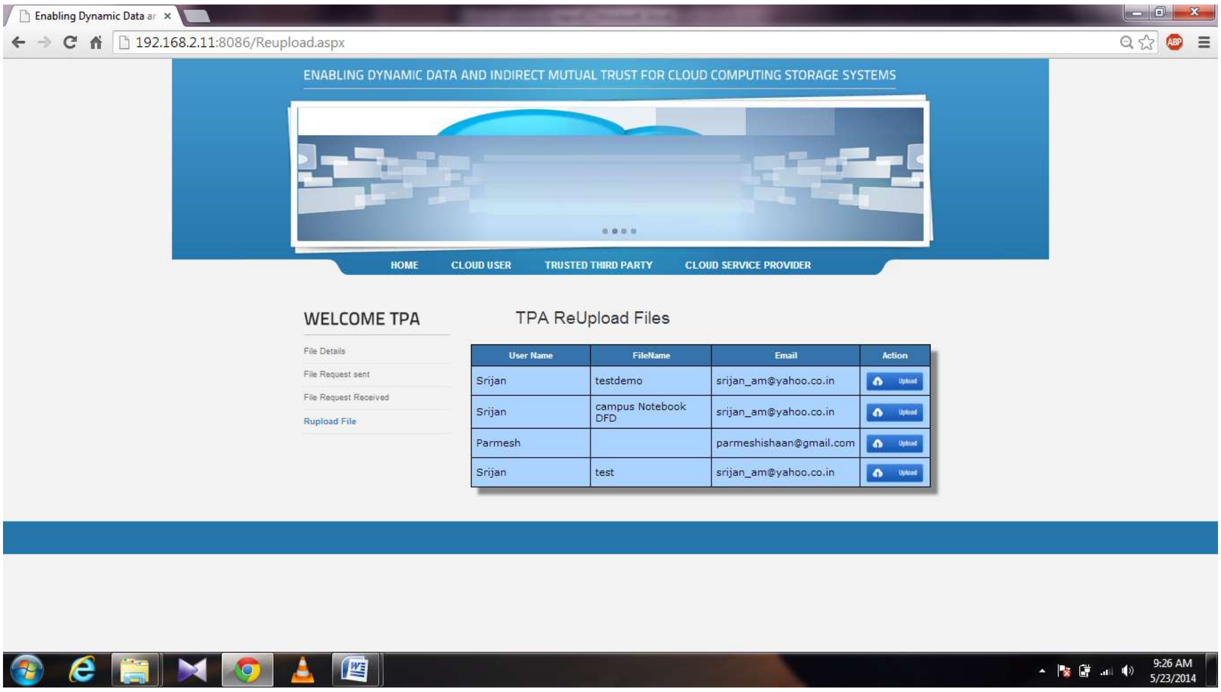
File Request Received

Rupload File

Trusted Third Party Files

User Name	FileName	Email	Sent Request
Srijan	123	srijan_am@yahoo.co.in	Send Request

S9. Registered TTP

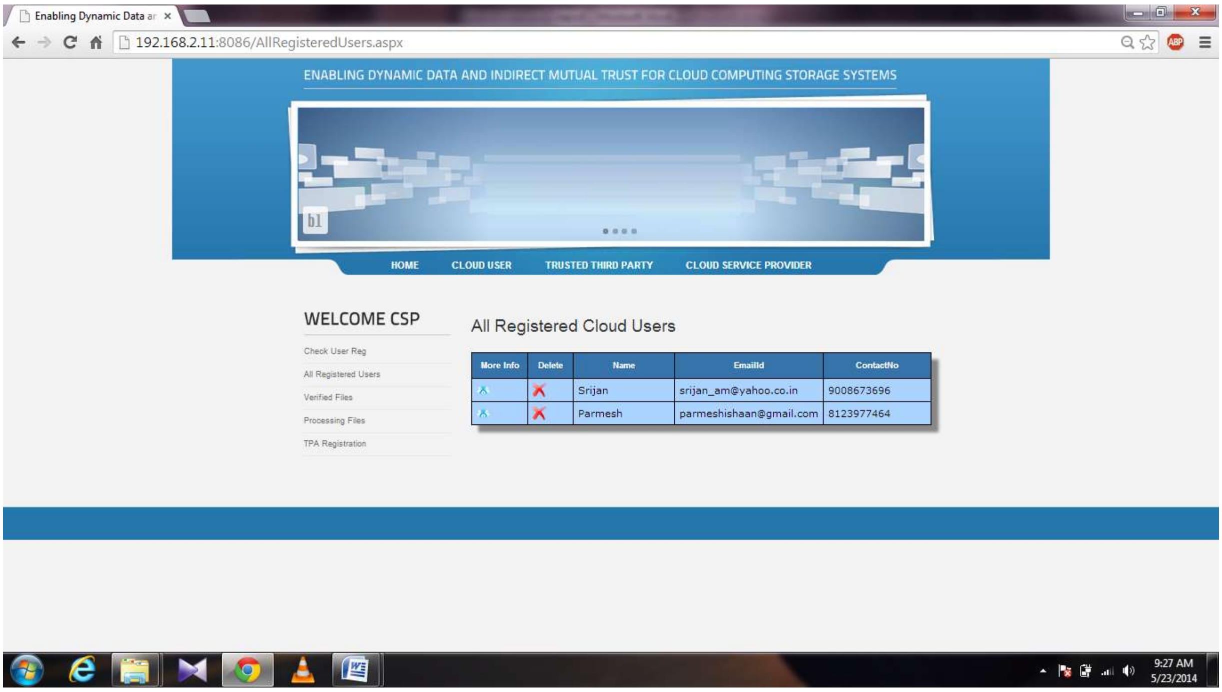


WELCOME TPA

TPA ReUpload Files

User Name	FileName	Email	Action
Srijan	testdemo	srijan_am@yahoo.co.in	Upload
Srijan	campus Notebook DFD	srijan_am@yahoo.co.in	Upload
Parmesh		parmeshishaan@gmail.com	Upload
Srijan	test	srijan_am@yahoo.co.in	Upload

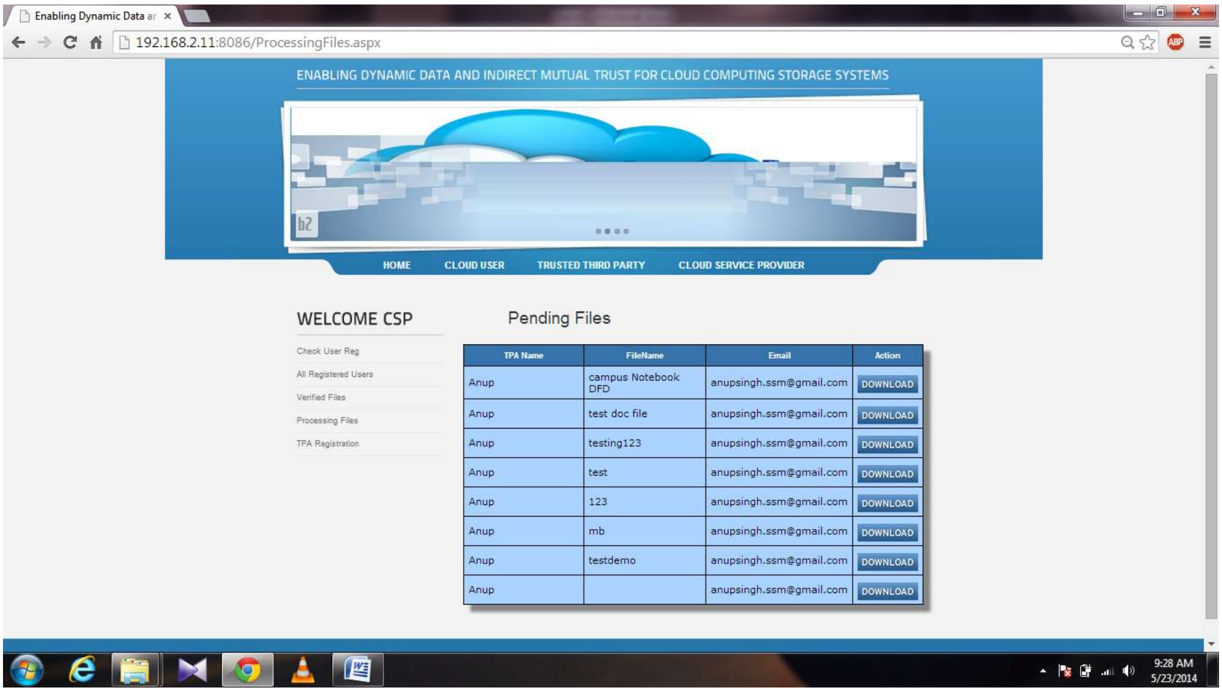
S10. The place where TTP can re-upload the file after verification.



All Registered Cloud Users

More Info	Delete	Name	EmailId	ContactNo
	✗	Srijan	srijan_am@yahoo.co.in	9008673696
	✗	Parmesh	parmeshishaan@gmail.com	8123977464

S11. CSP landing page. Here all registered users are seen.



Pending Files

TPA Name	FileName	Email	Action
Anup	campus Notebook DFD	anupsingh.ssm@gmail.com	DOWNLOAD
Anup	test doc file	anupsingh.ssm@gmail.com	DOWNLOAD
Anup	testing123	anupsingh.ssm@gmail.com	DOWNLOAD
Anup	test	anupsingh.ssm@gmail.com	DOWNLOAD
Anup	123	anupsingh.ssm@gmail.com	DOWNLOAD
Anup	mb	anupsingh.ssm@gmail.com	DOWNLOAD
Anup	testdemo	anupsingh.ssm@gmail.com	DOWNLOAD
Anup		anupsingh.ssm@gmail.com	DOWNLOAD

S12. Pending files which are not yet verified by the TTP.