Project Report
on

Fuzzy Keyword Search Over Encrypted Data In A Cloud

Submitted by:

Anju N S	**EPAMECS009**
Chaythanya S K	**EPAMECS022**
Hari K	**EPAMECS030**
Rekha N	**EPAMECS045**

under the guidance of

Mr. Irshad M
Asst. Professor, Dept. Of Computer Science And Engineering

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GOVERNMENT ENGINEERING COLLEGE
SREEKRISHNAPURAM
PALAKKAD

May 19, 2016

CERTIFICATE

GOVERNMENT ENGINEERING COLLEGE
SREEKRISHNAPURAM
PALAKKAD - 678633

CS09 805(P) Project

PROJECT REPORT

This is to certify that this project report entitled **Fuzzy Keyword Search Over Encrypted Data In A Cloud** submitted by **Hari K** to the Department of Computer Science and Engineering, Government Engineering College, Sreekrishnapuram, Palakkad - 678633, in partial fulfillment of the requirement for the award of B.Tech Degree in Computer Science and Engineering is a bonafide record of the work carried out by him.

Mr. Irshad M	Mr. Dileesh E D	Dr. Rafeeque P C
Guide	**Co-ordinator**	**HOD of CSE**

Place : Sreekrishnapuram
Date : May 19, 2016

Acknowledgement

At the very outset, I would like to extend my sincere and heartfelt obligation towards the Almighty, who kept me healthy and gave me the strength and wisdom to pursue this endeavour.

I am highly indebted to my guide, **Mr. Irshad M**, Assistant Professor, Department of Computer Science and Engineering, Government Engineering College, Sreekrishnapuram for his exemplary guidance and constant supervision as well as for providing all the necessary information regarding the project.

I would like to thank **Dr. Rafeeque P C**, Asst. Professor and Head of the Department, Computer Science and Engineering, Government Engineering College, Sreekrishnapuram for giving me this opportunity and for his constant encouragement and guidance throughout the progress of this project.

I express my heartfelt gratitude to **Mr. Dileesh E D**, Asst. Professor, Department of Computer Science and Engineering, Government Engineering College Sreekrishnapuram, for his cordial support, valuable information and guidance, which helped me in completing this task through various stages.

My thanks and appreciations also go to my friends for sharing so many wonderful moments.They have willingly helped me out with their abilities.

I take this opportunity to thank all the other faculties of Computer Science and Engineering Department,Government Engineering College, Sreekrishnapuram and all other people who have helped me in many ways for the successful completion of this work.

Last but not the least, I thank my parents for their constant encouragement and support without which this project would not have been possible.

Abstract

Cloud computing is a technology that uses the internet and central remote servers to maintain data and applications. Cloud computing allows consumers and businesses to use applications without installation and access their personal files at any computer with internet access. This technology allows for much more efficient computing by centralizing storage, memory, processing and bandwidth.

As cloud computing becomes more prevalent more and more sensitive data are being centralized into the cloud. As a result, the biggest concerns about cloud computing are security and privacy. If a client can login from any location to access data and applications, it's possible that the client's privacy could be compromised. Privacy preservation is one of the major hurdles for a cloud user, especially when the users data that reside in local storage is outsourced and computed in a cloud. The Cloud Service Provider has full control over the infrastructure of cloud including lower level of system stack and system hardware. In a few cases the service provider is not fully trusted, but still we need the service.

Therefore, some method should be employed to protect the user data and user queries from an unauthorized person in the cloud environment. Thus, before sending data to the cloud, it must be encrypted to protect from unsolicited access. Encryption is the easiest route to ensure privacy of the data. But a search operation on such data is a very challenging task.

There are many searching technique which were implemented in the cloud but these techniques support only exact keyword search.

The proposed system implements a verifiable fuzzy keyword search in semi-honest but curious cloud.

Also to ensure the correctness of the stored data , a third party auditor is implemented. A third party auditor is an independent trusted entity. Clients can turn to Third party auditor (TPA) to check the uprightness of outsourced data.

Keywords : Fuzzy Keyword Search, Third Party Auditor, Cloud

Contents

List of Figures

List of Algorithms

Chapter 1

Introduction

1.1 Overview

As cloud computing becomes prevalent, more and more sensitive information are being centralized into the cloud, such as emails, personal health records, government documents, etc. By storing their data into the cloud, the data owners can be relieved from the burden of data storage and maintenance so as to enjoy the on-demand high quality data storage service. However, the fact that data owners and cloud server are not in the same trusted domain may put the outsourced data at risk, as the cloud server may no longer be fully trusted. It follows that sensitive data usually should be encrypted prior to outsourcing for data privacy and combating unsolicited accesses.

However, data encryption makes effective data utilization a very challenging task given that there could be a large amount of outsourced data files. Moreover, in cloud computing, data owners may share their outsourced data with a large number of users. The individual users might want to only retrieve certain specific data files they are interested in during a given session. One of the most popular ways is to selectively retrieve files through keyword-based search instead of retrieving all the encrypted files back which is completely impractical in cloud computing scenarios. Such keyword-based search technique allows users to selectively retrieve files of interest and has been widely applied in plaintext search scenarios, such as Google search. Unfortunately, data encryption restricts users ability to perform keyword search and thus makes the traditional plaintext search methods unsuitable for cloud computing. Besides this, data encryption also demands the protection of keyword privacy since keywords usually contain important information related to the data files. Although encryption of keywords can protect keyword privacy, it further renders the traditional plaintext search techniques useless in this scenario.

To securely search over encrypted data, searchable encryption techniques have been developed in recent years. Searchable encryption schemes usually build up an index for each keyword of interest and associate the index with the files that contain the keyword. By integrating the trapdoors of keywords within the index information, effective keyword search can be realized while both file content and keyword privacy are well-preserved. Although allowing for performing searches securely and effectively, the existing searchable encryption techniques do not suit for cloud computing scenario since they support only exact keyword search. That is, there of minor typos and format inconsistencies. It is quite common that users searching input might not exactly match those pre-set keywords due to the possible typos, or their lack of exact knowledge about the data.

Here, the focus is on enabling effective yet privacy preserving fuzzy keyword search in cloud computing. Fuzzy keyword search greatly enhances system usability by returning the matching files when users searching inputs exactly match the predefined keywords or the closest possible matching files based on keyword similarity semantics, when exact match fails. More specifically, edit distance is used to quantify keywords similarity and develop a novel technique, i.e., a gram based technique, for the construction of fuzzy keyword sets. This technique eliminates the need for enumerating all the fuzzy keywords and the resulted size of the fuzzy keyword sets is significantly reduced. Based on the constructed fuzzy keyword sets, an efficient fuzzy keyword search scheme is proposed.

To ensure the integrity of the files stored in the cloud and independent third party auditor is also implemented. TPA or the Third Party Auditor is an entity who has expertise and capabilities that clients do not have. He is trusted to assess and expose risk of cloud storage services on behalf of the clients upon request. Any registered user can request the TPA to check the validity of a file. TPA is trusted and hence itself will not make user data vulnerable to threat.

1.2 Purpose

The purpose of this document is to give a detailed description about the proposed application. It will illustrate the purpose and complete requirements for the development of the system. It will also explain the design and implementation along with algorithms used if any. Finally it will provide an evaluation of the application.

1.3 Problem Statement

The aim of the project is to set up an ideal private cloud that would allow clients to upload, delete and search effectively without compromising the privacy of the files. Security and privacy are two major concerns of a cloud user. Hence, for the protection of data privacy, sensitive data usually have to be encrypted before outsourcing. The major challenge is to implement an efficient searching mechanism in encrypted data stored in a cloud. In this system a fuzzy keyword search scheme is used, which greatly enhances system usability by returning the matching files when users searching inputs exactly match the predefined keywords or the closest possible matching files based on keyword similarity semantics, when exact match fails. The proposed system introduces a third party auditor to audit users data whenever required. Any user (not just the data owner) can challenge the cloud server for the correctness of the stored data via the third party. The third party auditor keeps no private information while auditing thus user data remains safe and secure.

1.4 Scope of the Project

The scope of the project is confined to cloud computing. In the current scenario almost every internet user is part of the cloud architecture either directly or indirectly. As a result a huge amount of data is being outsourced by every user. Since the cloud service provider is not a trusted entity, the data needs to be encrypted before outsourcing. The problem with encrypting data is that it makes effective data utilization cumbersome, given that there could be a large amount of outsourced data files.

Moreover, in cloud computing, data owners may share their outsourced data with a large number of users. The individual users might want to only retrieve certain specific data files they are interested in during a given session. One of the most popular ways is to selectively retrieve files through keyword-based search instead of retrieving all the encrypted files back which is completely impractical in cloud computing scenarios. Our system implements an efficient searching technique using fuzzy keywords, which returns the user with relevant files even when the closest possible match fails.

Chapter 2

Literature Survey

2.1 Fuzzy Keyword Search

2.1.1 Plaintext Fuzzy Keyword Search

Recently, the importance of fuzzy search has received attention in the context of plaintext searching in information retrieval community. They addressed this problem in the traditional information-access paradigm by allowing user to search without using try-and-see approach for finding relevant information based on approximate string matching. The approximate string matching algorithms among them can be classified into two categories: on-line and off-line. The on-line techniques, performing search without an index, are unacceptable for their low search efficiency, while the off-line approach, utilizing indexing techniques, makes it dramatically faster. A variety of indexing algorithms, such as suffix trees, metric trees and q-gram methods, have been presented. At the first glance, it seems possible for one to directly apply these string matching algorithms to the context of searchable encryption by computing the trapdoors on a character base within an alphabet. However, this trivial construction suffers from the dictionary and statistics attacks and fails to achieve the search privacy.

2.1.2 Searchable Encryption

Traditional searchable encryption has been widely studied in the context of cryptography. Among those works, most are focused on efficiency improvements and security definition formalizations. The first construction of searchable encryption that each word in the document is encrypted independently under a special two-layered encryption construction. Another technique proposed was to use Bloom filters to construct the indexes for the data files. For each file, a Bloom filter containing trapdoors of all unique words is built up and stored on the server. To search for a word, the user generates the search request by computing the trapdoor of the word and sends it to the server. Upon receiving the request, the server tests if any Bloom filter contains the trapdoor of the query word and returns the corresponding file identifiers. To achieve more efficient search, other approaches, where a single encrypted hash table index is built for the entire file collection we proposed. In the index table, each entry consists of the trapdoor of a keyword and an encrypted set of file identifiers whose corresponding data files contain the keyword. As a complementary approach, a public-key based searchable encryption scheme. In this construction, anyone with the public key can write to the data stored on the server but only authorized users with the private key can search. As an attempt to enrich query predicates, conjunctive keyword search, subset query and range query over encrypted data, have also been proposed. Note that all these existing schemes support only exact keyword search, and thus are not suitable for cloud computing.

2.1.3 Other Works

Private matching, as another related notion, has been studied mostly in the context of secure multiparty computation to let different parties compute some function of their own data collaboratively without revealing their data to the others. These functions could be intersection or approximate private matching of two sets, etc. The private information retrieval is an often-used technique to retrieve the matching items secretly, which has been widely applied in information retrieval from database and usually incurs unexpectedly high computation complexity.

2.2 Cloud Computing

Cloud computing, also on-demand computing, is a kind of Internet-based computing that provides shared processing resources and data to computers and other devices on demand. It is a model for enabling ubiquitous, on-demand access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services),which can be rapidly provisioned and released with minimal management effort. Cloud computing and storage solutions provide users and enterprises with various capabilities to store and process their data in third-party data centres. It relies on sharing of resources to achieve coherence and economy of scale, similar to a utility (like the electricity grid) over a network.

Cloud computing promises several attractive benefits for businesses and end users. Three of the main benefits of cloud computing include:

Self-service provisioning End users can spin up computing resources for almost any type of workload on-demand.

Elasticity Companies can scale up as computing needs increase and then scale down again as demands decrease.

Pay per use Computing resources are measured at a granular level, allowing users to pay only for the resources and workloads they use.

Cloud computing services can be private, public or hybrid.

Private Cloud Private cloud services are delivered from a business' data center to internal users. This model offers versatility and convenience, while preserving management, control and security. Internal customers may or may not be billed for services through IT chargeback.

Public Cloud In the public cloud model, a third-party provider delivers the cloud service over the Internet. Public cloud services are sold on-demand, typically by the minute or the hour. Customers only pay for the CPU cycles, storage or bandwidth they consume. Leading public cloud providers include Amazon Web Services (AWS), Microsoft Azure, IBM/SoftLayer and Google Compute Engine.

Hybrid Cloud Hybrid cloud is a combination of public cloud services and on-premises private cloud with orchestration and automation between the two. Companies can run mission-critical workloads or sensitive applications on the private cloud while using the public cloud for bursty workloads that must scale on-demand. The goal of hybrid cloud is to create a unified, automated, scalable environment which takes advantage of all that a public cloud infrastructure can provide, while still maintaining control over mission-critical data.

A cloud provides various types of services, some of the important services are SaaS, PaaS and IaaS.

Software as a Service(SaaS It is a model of software deployment whereby according to the demand of the customer a provider provides licensed application for the specified time.

Platform as a Service (PaaS) Generates all facilities required to support the complete cycle of construction and delivery of web-based applications wholly available in Internet with built in services so there is no need of downloading software or special installations by developers.

Infrastructure as a Service (IaaS) It provides resources, such as servers, connections, storage and other necessary tools to construct an application design according to the need of organizations, making it quick, easy and economically viable.

2.3 Metal as a Service(MAAS)

Metal as a Service, or MAAS, is a provisioning construct created by Canonical, developers of Ubuntu, a Linux-based operating system. MAAS is designed to help facilitate and automate the deployment and dynamic provisioning of hyperscale computing environments such as big data workloads and cloud services.

MAAS serves as a layer underneath Infrastructure-as-a-Service (IaaS) and works with Juju to coordinate applications and workloads, deploying hardware and services that can dynamically scale up and down.

2.4 Juju

Juju is an open source service orchestration management tool developed by Canonical Ltd., Juju allows software to be quickly deployed, integrated and scaled on a wide choice of cloud services or servers.

The central mechanism behind Juju is called Charms. Charms can be written in any programming language that can be executed from the command line. A charm is a collection of YAML configuration files and a selection of "hooks". A hook is a naming convention to install software, start/stop a service, manage relationships with other charms, upgrade charms, scale charms, configure charms, etc. Charms can have many properties. Charm helpers allow boiler-plate code to be automatically generated hence accelerating the creation of charms.

2.5 Openstack

Openstack is a set of software tools for building and managing cloud computing platforms for public and private clouds. Backed by some of the biggest companies in software development and hosting, as well as thousands of individual community members, many think that Openstack is the future of cloud computing. Openstack is managed by the Openstack Foundation, a non-profit organisation which oversees both development and community-building around the project.

Openstack lets users deploy virtual machines and other instances which handle different tasks for managing a cloud environment on the fly. It makes horizontal scaling easy, which means that tasks which benefit from running concurrently can easily serve more or less users on the fly by just spinning up more instances. For example, a mobile application which needs to communicate with a remote server might be able to divide the work of communicating with each user across many different instances, all communicating with one another but scaling quickly and easily as the application gains more users.

And most importantly, Openstack is open source software, which means that anyone who chooses to can access the source code, make any changes or modifications they need, and freely share these changes back out to the community at large. It also means that Openstack has the benefit of thousands of developers all over the world working in tandem to develop the strongest, most robust, and most secure product that they can.

Chapter 3

Requirement Analysis

3.1 Introduction

Requirements analysis and validation is a process of refinement, modelling and specification of the already discovered user requirements. This chapter describes the method of requirement elicitation employed and the user requirements thus gathered. The requirements are also finalized after validation using a suitable method.

3.2 Method of Requirement Elicitation

A questionnaire based method was carried out for eliciting the user requirements.

Approach

To analyze the user requirements a set of questions were prepared to conduct an online survey. Listed below were the questions:

1. Do you understand the term cloud?
2. Grade your level of cloud computing knowledge.
3. Do you currently use any of these public cloud services?
4. What are the services you currently enjoy in the above mentioned cloud services?
5. Do you think a college private cloud is necessary?
6. Are you convinced of the privacy and security of your files in cloud?
7. Whom do you think should upload the files?
8. Would you like to share you uploaded files with others?
9. Would you like your search results to be classified?

10. Should your files be returned during a search?
11. How would you like the search engine to be?
12. Would you like to edit the files within the cloud?
13. Would you like an android app to manage your cloud?
14. How would you like to make sure of the privacy/correctness of your files?

3.3 Overall Description

This section mainly deals with the application structure and its various interfaces.

3.3.1 Project Perspective

The product to be developed could be classified as an alternative to existing solutions. But at the same time it is a self contained project. As mentioned in the Scope of the product, existing systems do not implement a fuzzy keyword search combined with third-party auditing. Given below is a simple diagrammatic representation of the product.

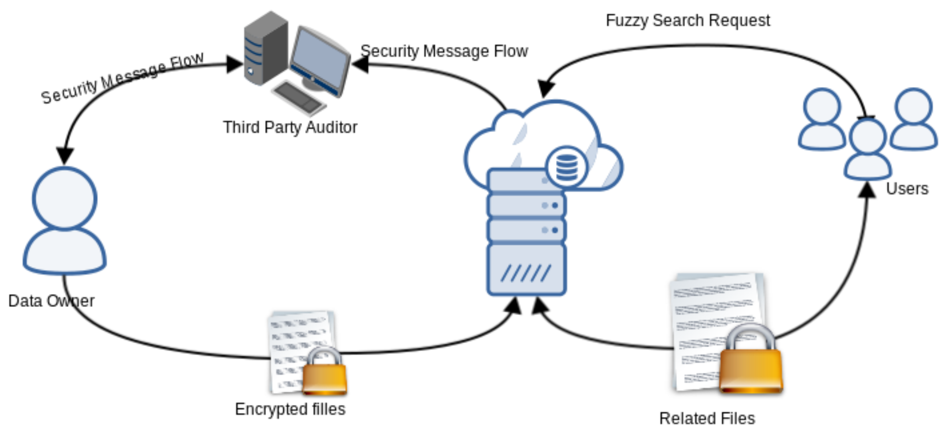

Figure 3.1: Fuzzy Keyword Search over Encrypted Data In a Cloud

3.3.2 Product Functions

- Search and download public files
- Upload files
- Delete uploaded files
- Audit the cloud server using a Third Party Auditor

3.3.3 User Classes and Characteristics

Based on the product functions mentioned above there are 4 classes of users:

1. Data User: This is a general class of all registered users.This class of users can upload and search files as well as invoke the auditor.They may not be able to download,edit or delete a file without the permission of its owner.

2. Data Owner: Owner is the most important users as most of the functionalities are developed with respect to him.He is a subset of Data Users.Data Owner can upload,edit,delete and download his files.He may invoke the auditor as well.Only data Owner can view the actual contents of a file.

3. Cloud Service Provider: CSP is the one who provides the storage service.He will act as the admin and hence can view the user profiles and encrypted files.In the proposed product we assume CSP to be semi-honest and curious.

4. Third Party Auditor: An entity, which has expertise and capabilities that clients do not have, and is trusted to assess and expose risk of cloud storage services on behalf of the clients upon request.Any registered user can request the TPA to check the validity of a file.TPA is trusted and hence itself will not make user data vulnerable to threat.

3.3.4 Operating Environment

The system requires the following hardware and software environments for development and to function properly:

3.3.5 Hardware Requirements

- A minimum of 7 PCs with at least
  - 1 PC having 1 processor, 2GB memory, and 5GB storage
  - 1 PC having 1 processor, 512MB memory, and 5GB storage
  - 1 PC having 1 processor, 2GB memory, and 10GB storage
- Switch
- Intranet and Internet connectivity

3.3.6 Software Requirements

Following are the software required for the development of the project:

- Ubuntu Server Images
- Openstack

3.3.7 Design And Implementation Constraints

As mentioned in the previous section, the cloud is built and managed using Openstack. Openstack Networking (neutron) architecture mainly requires three nodes, two of which require more than one network interface. Since none of the PCs currently available has more than one Ethernet interface, the requirement is expected to be fulfilled by a wireless LAN interface in the PCs or by running virtual machines where network interfaces are configurable.

3.3.8 Assumptions and Dependencies

The project is expected to be built on free software which were mentioned in software requirements,especially the hypervisors.So if in the future these software turn proprietary, then it could effect the development of the product. Apart from software, it is assumed that the amount of hardware dedicated to the project is feasible. Also an uninterrupted power and network connectivity is assumed.

3.4 User Requirements

3.4.1 External Interface Requirements

User Interfaces

Users can access the Secure Cloud using a web interface.The web page will have the standard forms for User Register and Sign In.Once signed in the Data User/Owner will be provided buttons to enable uploading of files,viewing user owned files and a search bar to search for public files.

Files owned by User will be listed in a table with delete and download buttons.Also there would be an option to request TPA services.On every page there would be the Settings and Logout links.

Hardware Interfaces

To set up a basic cloud environment, the three-node networking architecture(neutron) is followed:

- The basic controller node runs the management portions of Compute and Networking, Networking plug-in, and the dashboard. It also includes supporting services such as a database,message broker, and Network Time Protocol (NTP).
- The network node runs the Networking plug-in. It had agents in Layers 2 and 3 that provision and operate tenant networks. Layer 2 services include provisioning of virtual networks and tunnels. Layer 3 services include routing, Network Address Translation (NAT), and Dynamic Host Configuration Protocol (DHCP). This

node also handles external (internet) connectivity for tenant virtual machines or instances.

- The compute node runs the hypervisor portion of Compute, which operates tenant virtual machines or instances. By default Compute uses KVM as the hypervisor. The compute node also runs the Networking plug-in and layer 2 agent which operate tenant networks and implement security groups.

Software Interfaces

The private Secure cloud is set up using Ubuntu OS and Openstack Cloud Software Platform.Openstack provides many tools for managing a private cloud and providing IaaS.Since the proposed project provides Storage as a Service, Openstack was the best choice.Nova,Cinder,Keystone,Swift,Glance and Horizon are some of the software interfaces provided by Openstack to manage various features.

Openstack uses MaaS-Metal as a Service-which turns bare metal-servers and nodes-into an elastic cloud-like resource. This means that the cloud administrator just need tell MaaS about the machines it should manage and it will boot them, check that the hardware's okay, and have them waiting till they are needed. The administrator can then pull nodes up, tear them down and redeploy them at will; just as with virtual machines in the cloud. When a particular service is ready to be deployed, MaaS gives Juju the nodes it needs to power the service. Juju is used to manage how systems and services are deployed automatically in the cloud.

Communication Interfaces

The user interacts with the system either to search for a file or to upload/edit his file. Both these interactions are handled through web pages. The back end programming handles further processing of the request. The users need no internet connectivity since the search is implemented within a private network. File transfer between users and the server is carried out by FTP. Openstack uses Network Time Protocol (NTP) for clock synchronization between computer systems over the network.

3.4.2 Functional Requirements

Functional requirements will analyze the services provided to the users by the product:

Data Owner

1. File Browser: The Data Owners who have files to be outsourced can select the files of their choice using the Browse/Open File option.

2. Public/Private: Once selected, the owner can decide if the file should be Public or Private.Only a Public file will be returned during a search.Private files will be visible only to the Data Owner.

3. Encrypt the file: Using this option, the user can encrypt the file. Once encrypted, user can check it by viewing the file which would now show encrypted text.

4. Upload: Finally the user can upload the file to the cloud using the upload option.It is required that the user checks through each of the previous options for a successful upload.

5. Keywords: Whenever the user upload a public file he/she will be asked to enter few keyword that best describes the file contents.Keywords are collected because not only the file but also its name and keywords are encrypted.

6. File type: If the privacy of the file is set as public, the data owner is asked to specify the type of file that is being uploaded. Setting the file type for eg .mp3/.txt/.pdf/.pptx would help the developer provide a more efficient and classified search, thus keeping the search results minimum and relevant.

Data User

1. View File: This would enable all the users to view the file content, only that it would be encrypted.

2. Search: CSP will perform the fuzzy keyword search and return a list of files tagged with the keyword.User can only download these files.

3. Auditing:The user can request TPA to audit any public file and thus check the credibility of the CSP.

4. Download: Search would result in a list of files which the user can download.The downloaded file will be decrypted.

TPA

1. TPA Log in: The third party auditor will have his own user id and password with which he can log in and manage the requests.

2. Approve request: This would allow the TPA to verify the specified file by sending a challenge to the CSP and verifying its result.

Cloud Admin/CSP

1. Admin Login: Admin can log in with his respective username and password. In the Admin Page, all the uploaded(encrypted) file list,user details,approved/pending user requests can be viewed. He also verifies, manages and responds to TPA challenge queries.

2. View User Requests: The Admin has a list of user requests which has to be approved for the user to upload files.

3. View Uploaded Files: The Admin can view the entire list of uploaded files along with the user details.
4. View TPA Requests: The Admin can view a list of pending TPA requests and service them one by one.

3.4.3 Non-Functional Requirements

Safety Requirements

The database is set up on the controller node. Any damage to the controller node poses a serious threat to data.

Security Requirements

Since the project is intended to address the security concerns in cloud environments, it basically implements all the usual security techniques. All users of the cloud maintains an account with the CSP/ Cloud admin. The TPA as mentioned above is a separate entity that unlike the user or the data owner has special privileges.

It is ensured that the TPA does not keep any private information while auditing thus user data remains safe and secure.

3.5 Project Requirements

3.5.1 Software Quality Attributes

Software quality attributes are the parameters that are used to measure and quantify to what extent the system is rated along each of them. Each of these features will be studied and tested during the different phases of development of the application.

Scalability

The application should be scalable in the sense more features can be added after the implementation of the first version.In future, the system can be upgraded so that the data owner can edit his uploaded files.It can also be modified such that any user can edit public files stored in the cloud provided he has the right. When the resources are added, the total throughput of the system should be able to accommodate that change.

Maintainability

Maintainability is an obvious quality that every software should be having for it to be efficient. The associated factors were studied during the initial phase and were found to be feasible.

Adaptability

Adaptability to the running environment, in this case, the adaptability of the different operating systems with the underlying hardware features should be satisfied by the proposed system. Proposed system relies highly on the smooth running of the software interfaces that facilitate the setting up of cloud.Users may use an operating system and browser of their choice.

Security

Security is the key aspect of the proposed system. Authentication is required for the Data owner, user and the TPA.Also all the files stored in the cloud server needs to be encrypted and verified.

Availability

The features of the system will be available ready to use whenever the user needs as long as there is power.

Testability

Testability is the parameter which holds the key to identifying whether the system satisfies the other quality attributes. The proposed system is testable at any stage after the implementation of the basic application.

3.6 Requirements Validation

Initial list of user requirements were analyzed and validated based on features like cost, design, implementation etc. and were categorized as functional and non-functional requirements. The requirements analysis were reviewed for clarity, completeness and consistency. This section illustrates the techniques used for analysis and validation and their results.

3.6.1 Validated Requirements

The idea behind validating the requirements is to negotiate a list that is acceptable to both, the users and the developers.The objective is to create a consensus list based on factors like cost, ease of implementation, etc. which is then set aside for later action.

3.6.2 Review of Requirement Analysis

The requirements and problems analysed are validated by the information gathered from the survey, related to the proposed requirements. The most common problems were found and recorded.

Clarity

The requirements proposed were analysed for clarity. This was done by consulting with the users to verify if the validated set of requirements were not diverging from what the customers intended to have. Some of the initial requirements that were listed by the customers, being difficult from an implementation point of view, were excluded from the final list.

Completeness

The final set of user requirements were then evaluated for its completeness. Incompleteness can occur due to the lack of proper communication between the customers and the developers. In our project, no such cases were identified. Thus the validated user requirements were reviewed to be complete.

Consistency

As the final step of review of requirement analysis, the user requirements were checked for consistency. Initially, different user needs may be conflicting in nature. The process of analysis and validation avoids this ambiguity. To verify this, consistency check was done.

Chapter 4

Design

This chapter provides a written description of a software product, that a software designer writes in order to give a software development team overall guidance to the architecture of the software project. It is intended to provide a high level overview of our software design as well as detailed design description of some major components of our system and the initial stages and it shows how the software system will be structured to satisfy the requirements. It is the primary reference for code development and, therefore, it must contain all the information required by a programmer to write code.

4.1 Overall System Design

This section defines the structure, behaviour, and views of the system,a formal description and representation of a system. A system architecture comprises system components, the externally visible properties of those components, and the relationships between them. It provides a plan from which products can be procured, and systems developed, that will work together to implement the overall functionalities.

4.1.1 Overview

This document is a description of the stages, states, classes, communications, and transitions, involved in the project. They have been explained in detail with the help of the following diagrams:

- ER diagram gives the overall view of the different entities and the relationships between the entities. ER diagrams in the UML modeling carry an important role for database design part.
- Use case diagram is a portrayal of the different types of users of a system and the various ways in which they interact with the system.
- Sequence diagram shows how processes operate with one another and in what order.

- Activity diagram has been used to describe the business and operational step by step workflows of the components in the system.
- A data flow diagram 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.

4.2 Class Diagram

A class diagram is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

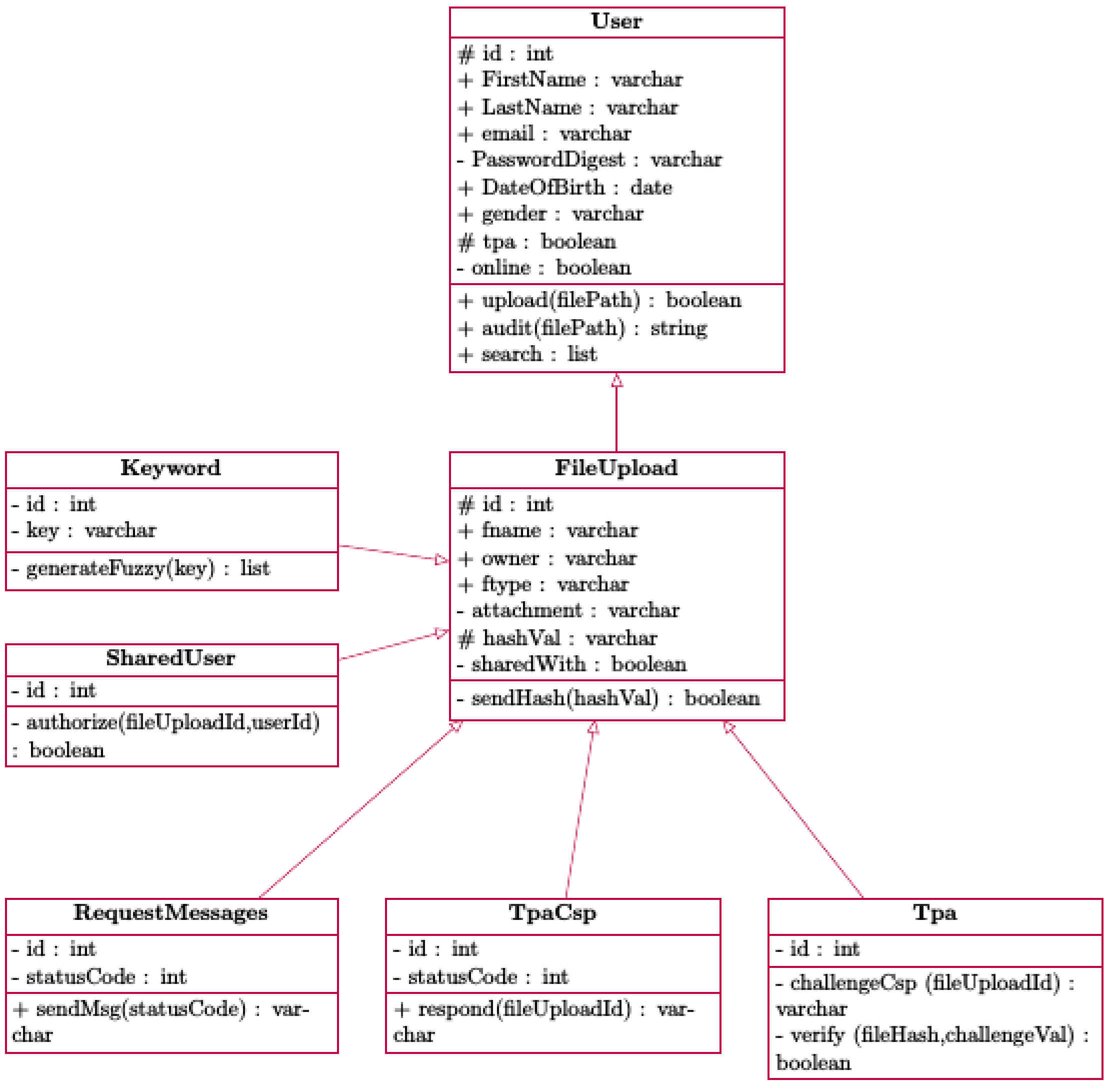

Figure 4.1: Class Diagram

4.3 Use Case Views

A use case defines a goal-oriented set of interactions between external actors and the system under consideration.

4.3.1 Data Owner

The data owner logs into his account and browses his local machine for a file he wants to upload.Once he has selected the file, he can either keep the file private or make it available for the search engine by keeping it public. If the file is kept public, He has to specify the type of the file, pdf, ppt, doc, etc. Before finishing the upload operation he has to provide keywords for his file. The file is then encrypted and send to the CSP for further processing.

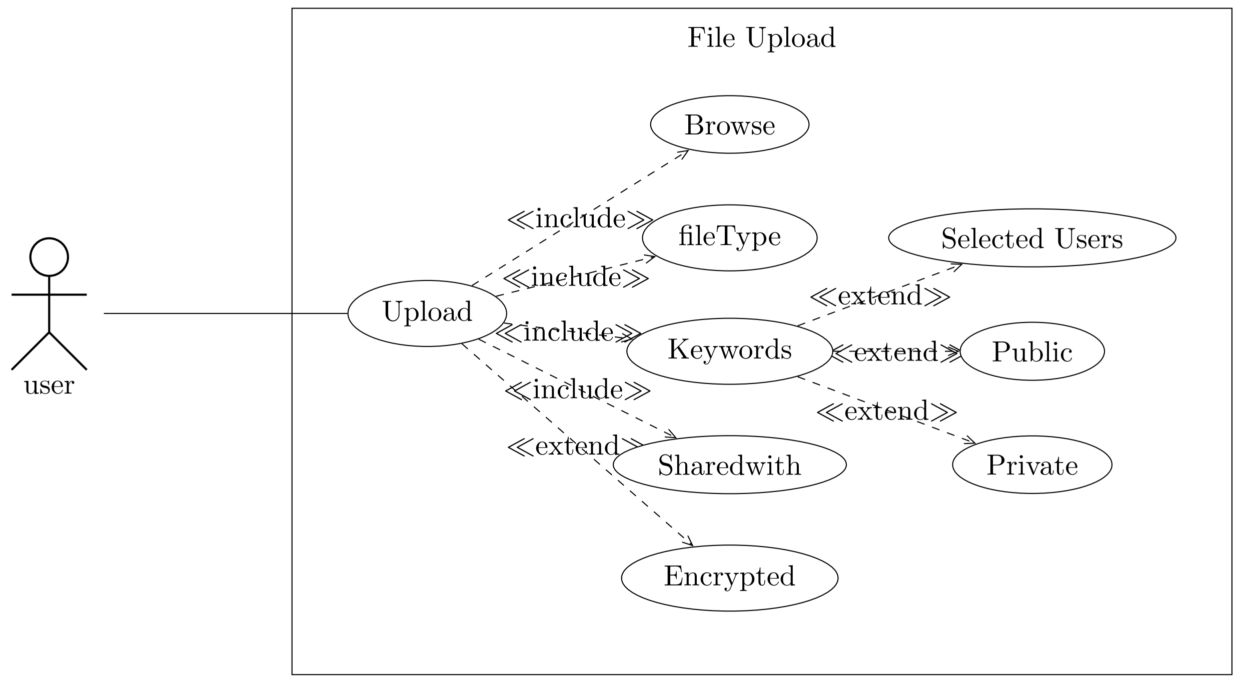

Figure 4.2: Use Case 1 - Data Owner Uploading a File

4.3.2 Data User

The user logs into his account and searches the cloud for a file. The search returns the matching files, and sorts it based on the file type, pdf, ppt, doc, etc. The user can challenge the integrity of the cloud server via the TPA.

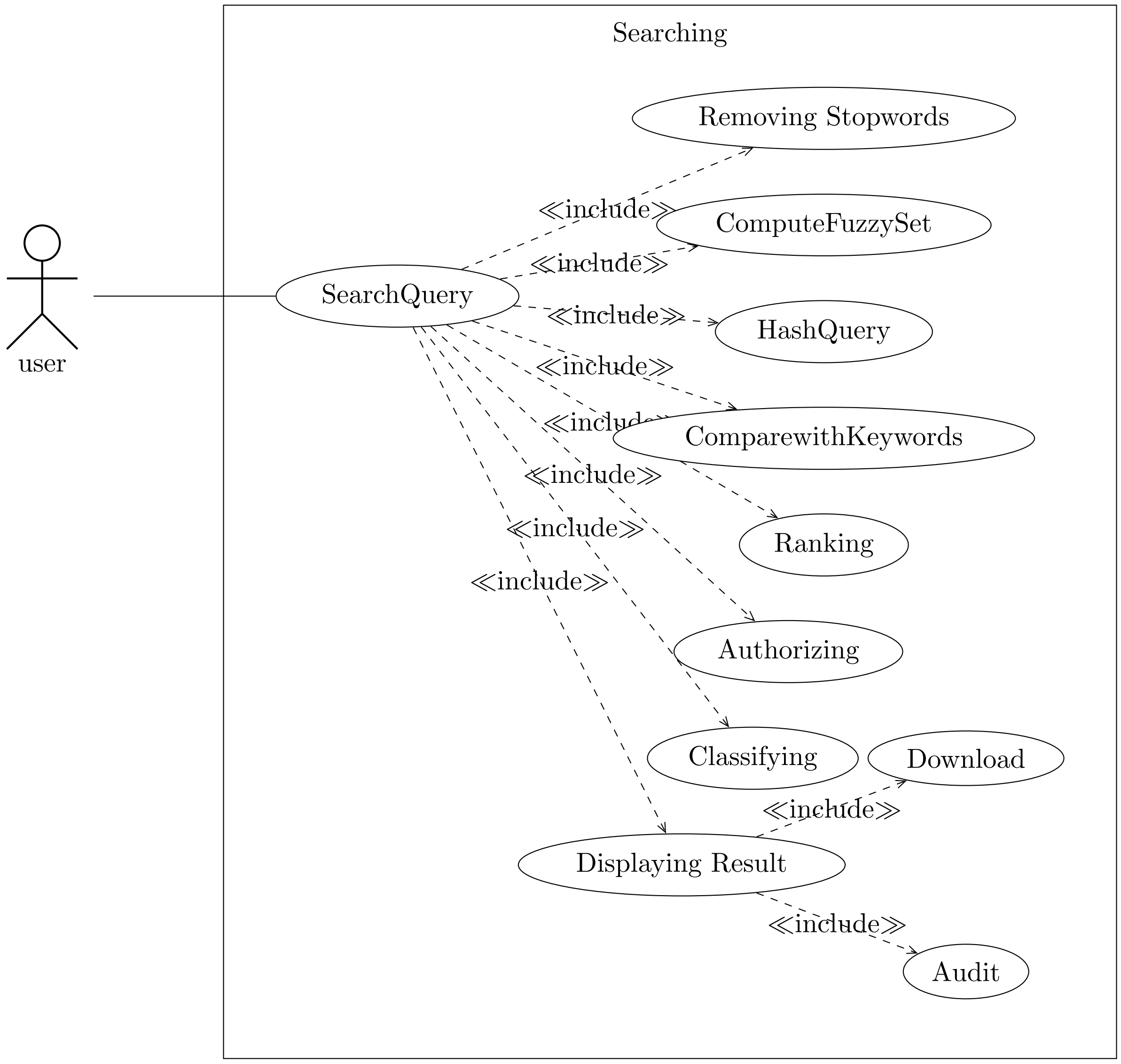

Figure 4.3: Use Case 2 - Data User Searching

4.3.3 Third Party Auditor

The data owner or the user requests the TPA to verify the integrity of the CSP. The TPA logs into his account, scans through the pending requests and verifies them. Once the requests are verified, the TPA challenges/queries the CSP for the validity of the file.

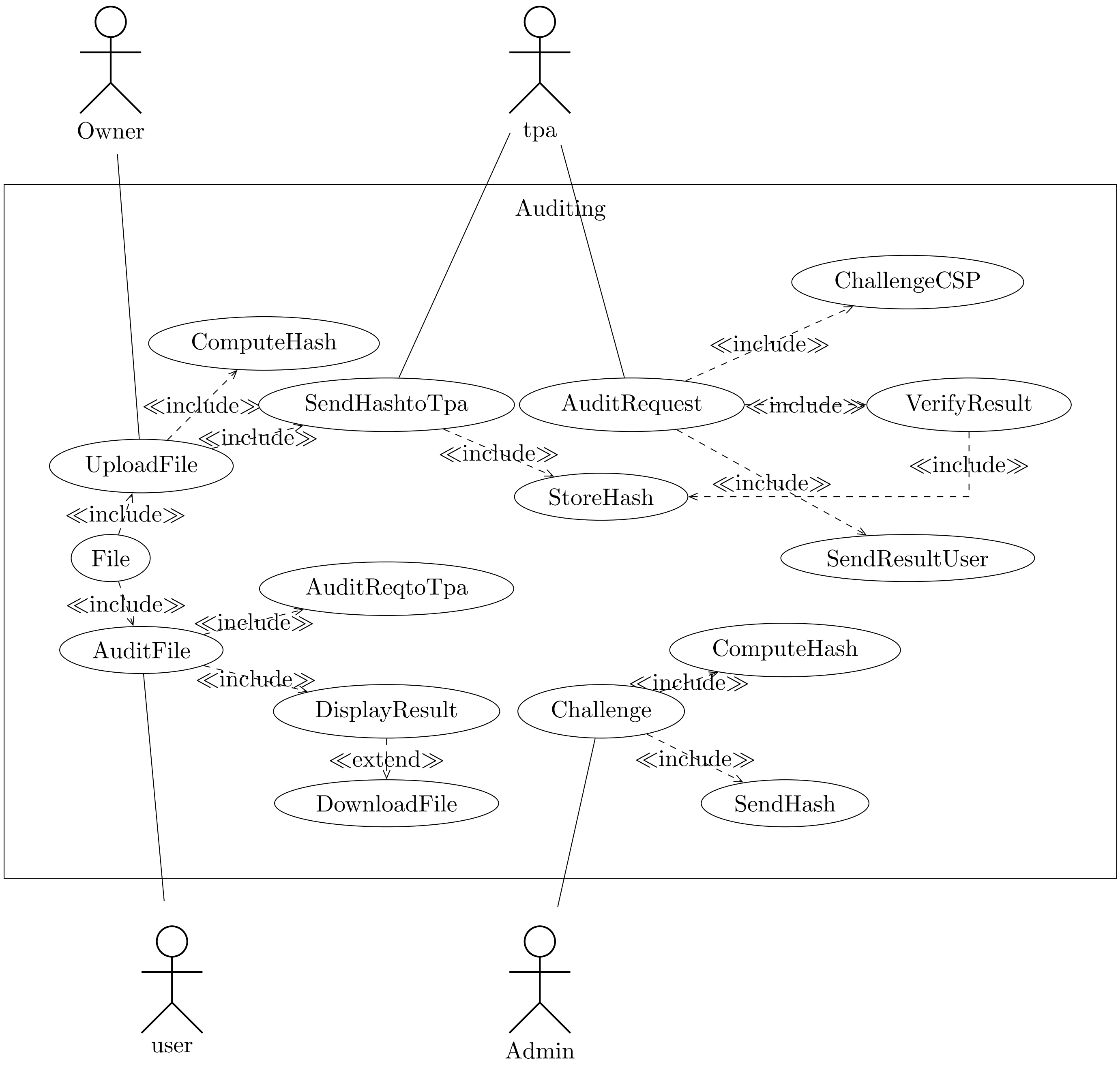

Figure 4.4: Use Case 3 - Third Party Auditing

4.3.4 CSP

The cloud admin logs into his account, and can view all the uploaded files and user details. He can also view the pending TPA requests. The admin responds to the challenge raised by the TPA.

4.4 Sequence Diagram

A sequence diagram in a Unified Modelling Language(UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams typically, are associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

4.4.1 Overview

A sequence diagram shows, as parallel vertical lines, different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

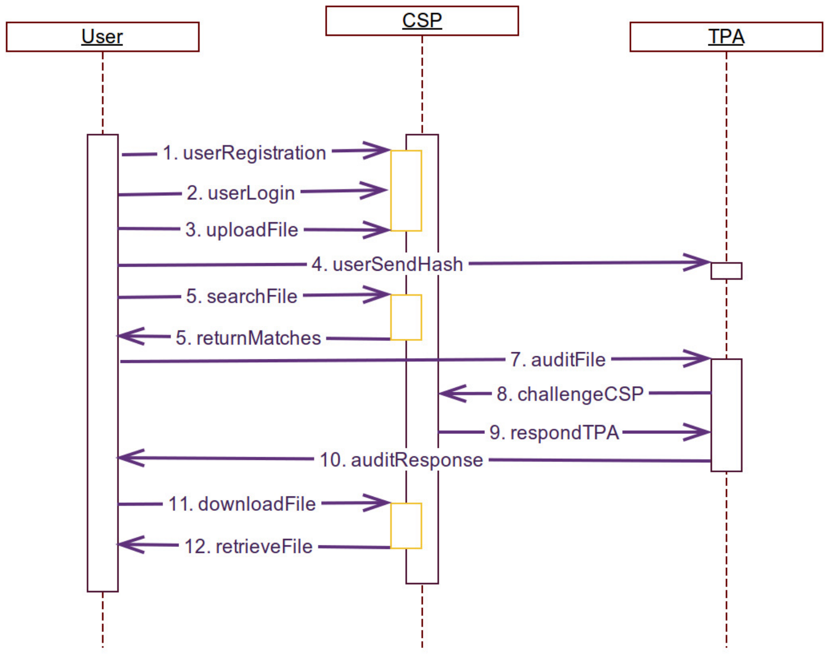

Figure 4.5: Sequence Diagram - Fuzzy Keyword Search over Encrypted Data In a Cloud

4.4.2 Description

The Data owner logs into his account and uploads files into the Cloud Service Provider's Database. TPA will be logged in always.Any data user can search for a file.Then CSP runs Fuzzy keyword algorithm and returns some relevant files to user which he can view or download.Now the user may request TPA to verify the file before downloading it.Then TPA would challenge the CSP which will inturn send a response.TPA analyses the response and pass on the response to the user.On the basis of the auditing user can download the file.

4.5 Activity Diagram

Activity diagrams are graphical representations of work flow of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams are intended to model both computational and organizational processes.

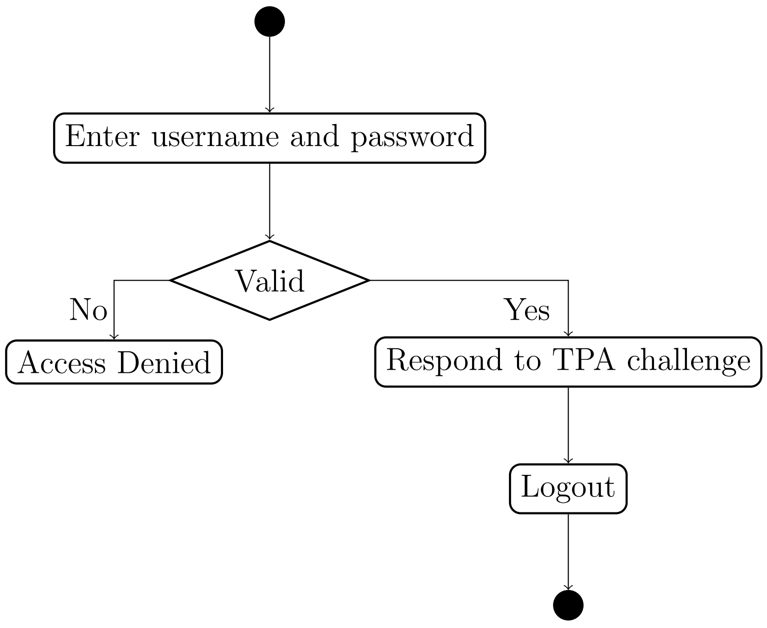

Figure 4.6: Activity Diagram - CSP

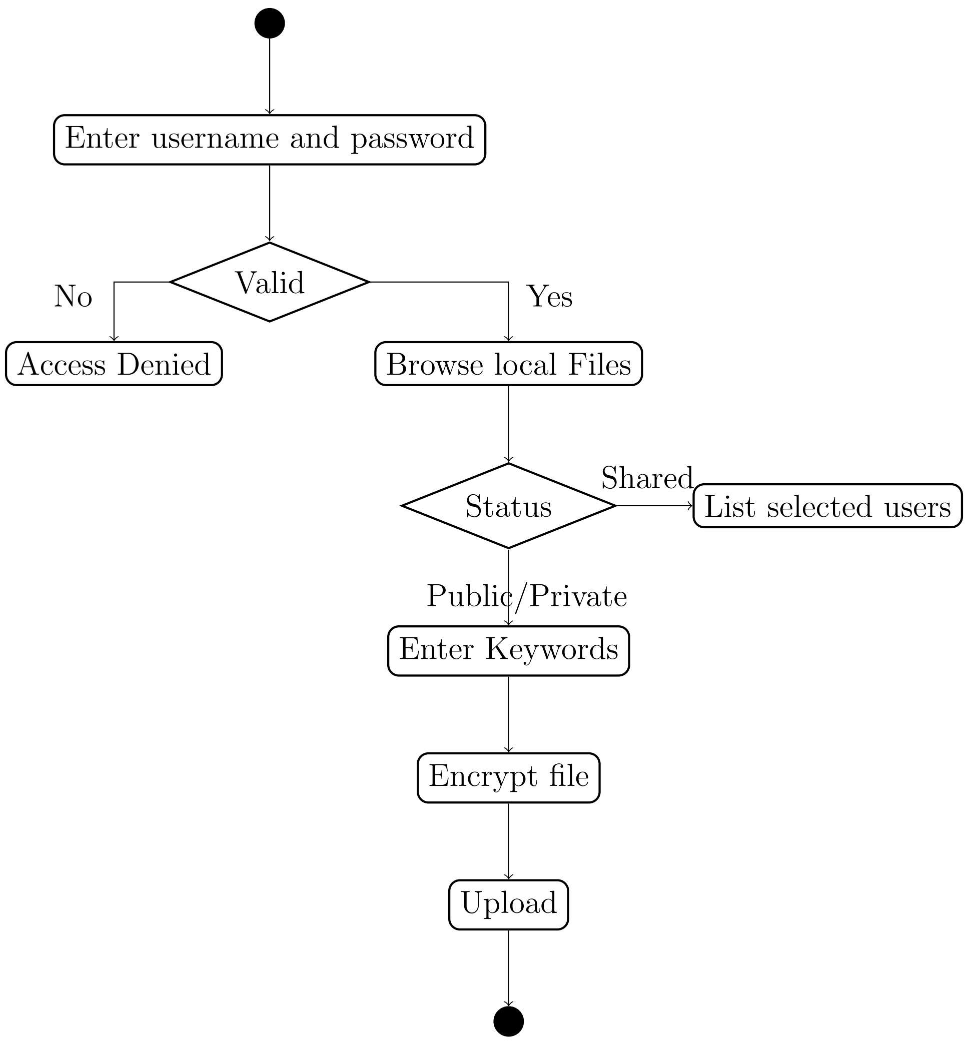

Figure 4.7: Activity Diagram - Uploading a File

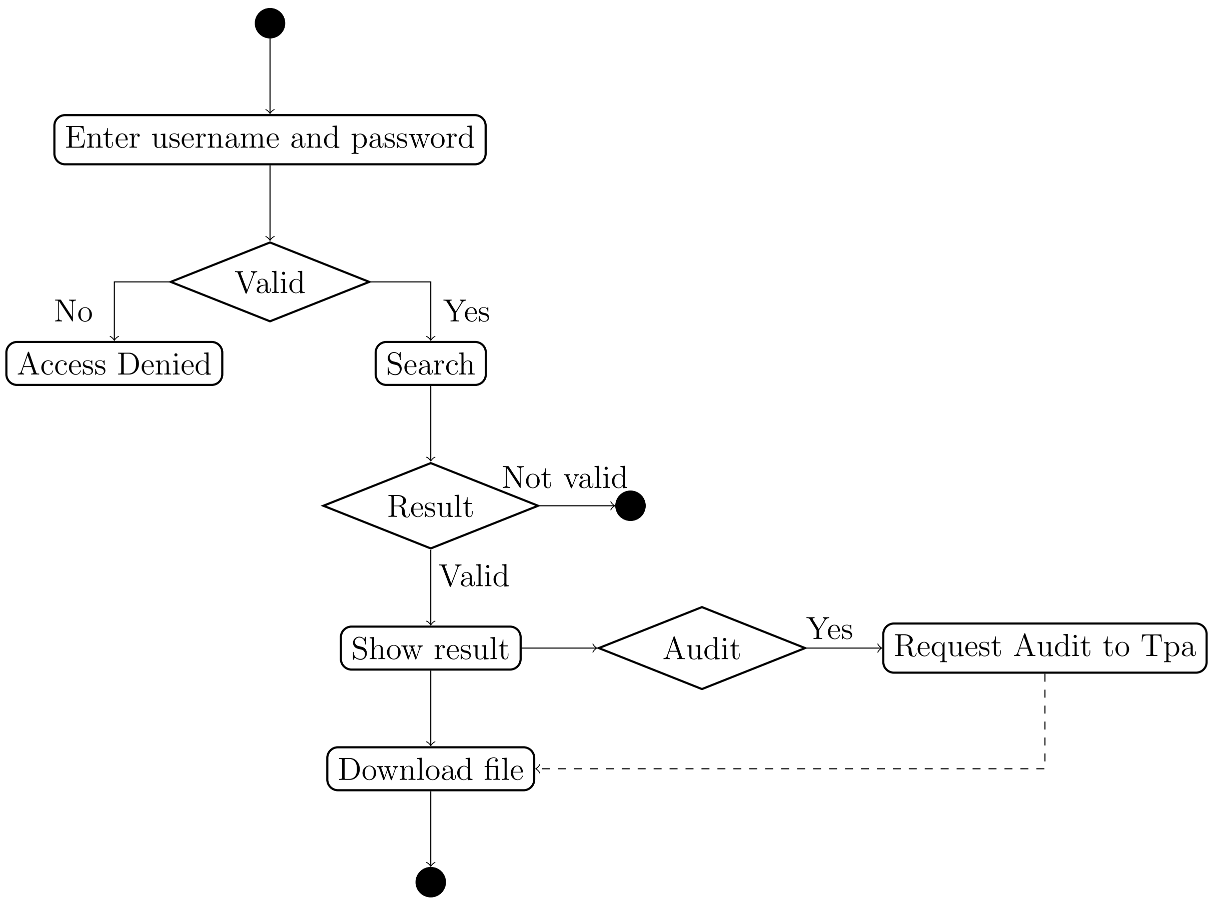

Figure 4.8: Activity Diagram - Searching a File

4.6 Data Flow Diagram

A data flow diagram (DFD) is a graphical representation of the flow of data through an information system, modelling 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 kinds 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.

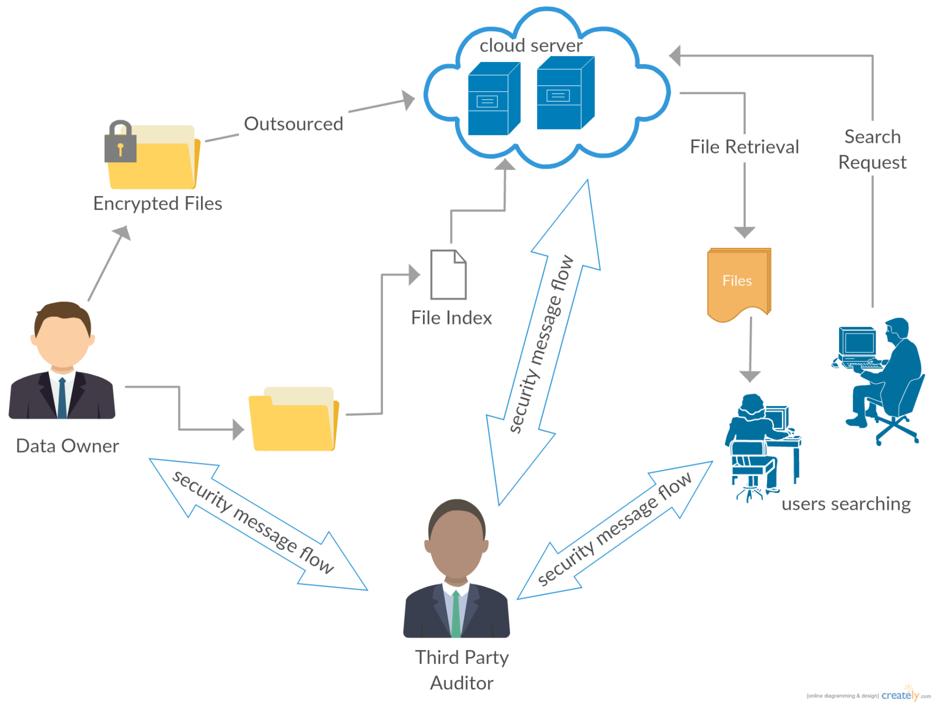

Figure 4.9: Data Flow Diagram - Fuzzy Keyword Search Over Encrypted Data in a Cloud

4.7 Database Design

This section provides the database description and their layouts. These are organised as different tables for storing the details regarding the user profiles, and the attributes of the user's files. The attributes of each of the tables, primary and foreign key constraints and the relationships between them are defined in the following sections.

4.7.1 Database Description

- Objective: The database includes the files uploaded by the users on which the search process is conducted. Every time a user uploads a file a set of keywords are provided which are also stored in the database. Database also includes details of all registered user.

- DBMS Environment:For this application, we are using the SQL database. SQL is used to manage data held in relational database management systems. Querying the database for authentication is done by using the declarative SELECT statement using optional keywords and clauses. Data manipulation is done through the commands such as INSERT INTO, DELETE FROM, UPDATE, MERGE etc.

Database Layout

The database storing the file details will have a table with the attributes given below.

File Database

- Encrypted File name
- Encrypted keywords
- Owner
- Public/Private
- Type
- Hash Value

User Data

- Name
- Username
- Password
- Email Id
- Date of Birth
- Contact Number

TPA

- File name
- Requested User
- Verified/Not verified
- Status

CSP

- Username
- Password
- Hashvalue

ER Diagram

ER diagram gives the overall view of the different entities and the relationships between the entities. ER diagrams in the UML modelling carry an important role for database design part.

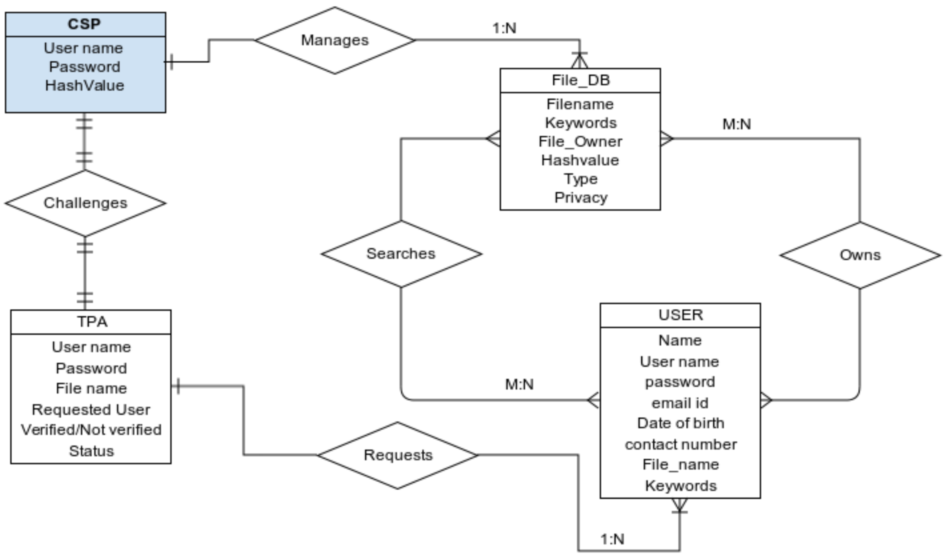

Figure 4.10: ER Diagram

4.8 Design Verification

Design verification is used to ensure that the product as designed is the same as the product as intended. In order to meet the customer expectations and avoid costly design modifications, appropriate selection of design verification method is essential.

The following project activities were analyzed for accomplishing this goal.

- Requirement gathering
- Feasibility study
- Specification development
- Detailed design development

As per the initial requirements proposed by the user, the objective of the project was fixed. The main feature was providing a fuzzy keyword based searching.We also added the feature of a TPA to ensure credibility of the CSP. All these requirements were consolidated as a Software Requirements Specification document.

The database design described the tables required, their attributes and their primary keys. This was analysed and it is verified to be sufficient for the searching process.

As a formal method, a team review was performed to verify that the design correctly implements the specifications.

The process model that was adopted in our project was incremental process model.This model is more flexible and hence it is less costly to change scope and requirements.After the first increment, a core product will be delivered, which can already be used by the customer. Based on customer feedback, a plan will be developed for the next increments, and modifications are made accordingly. This process continues, with increments being delivered until the complete product is delivered.The final approach of this technique,i.e. the testing the reliability of the product, based on a formal specification will be conducted when a formal product is ready. This will ensure the reliability that is related to the input/output trajectories selected and tested.

Chapter 5

Implementation and Testing

5.1 Cloud Setup

A cloud was set up in the following manner:

1. Connect the server(with two NICs) to the netfacing router and the switch.
2. Configure Wake-on-LAN on all the client systems connected to the switch.
3. Install Ubuntu Server 14.04 on the server.
4. Configure static IPs(in different networks) for both NICs in the server.
5. Install MAAS on the server.
6. Create a MAAS superuser and log into the account.
7. Import Ubuntu Server boot images to MAAS server(for node enlistment and provisioning).
8. Configure the cluster facing interface from MAAS to manage DHCP and DNS for that interface.
9. Switch on the client machines and wait for them to enlist in MAAS.
10. Configure power control for each enlisted machine(Wake-on-LAN) in MAAS.
11. Commission the nodes.
12. Install Juju.
13. Deploy required Juju charms for the Openstack services.

5.2 Auditing Algorithm

When file is uploaded/edited, its hash is sent to the TPA and CSP.Whenever an authorised user requests an audit, the TPA challenges the CSP to compute the hash of the file and send it to the TPA.The TPA compares the hash value sent by the user and the CSP. If they match, the file is clean, else its corrupted.The result of the audit is sent to the user.

Algorithm 1 Audit Request

```
1: while 1 do
2:     Read $RequestMessages$
3:     if Type == $Audit\_request$ then
4:         Create new entry in $TpaCsp$ with $file\_id$ and type = $challenge\_csp$
5:     end if
6: end while
```

Algorithm 2 Verify

```
1: while 1 do
2:     Read $TpaCsp$
3:     if (Type == $Csp\_response$) then
4:         Compare $hash\_val$ returned by CSP with $hash\_val$ with TPA
5:         if Both values match then
6:             write in $RequestMessages$ auditing successful, $file\_secure$
7:         else
8:             write in $RequestMessages$ auditin successsful, $file\_insecure$
9:         end if
10:    end if
11: end while
```

5.3 Encryption

Encryption of files is done using the AES algorithm using a 256bit key in CBC mode. Files are encrypted at the time of upload and it is the hash of encrypted files that are sent to the CSP and TPA. Encryption is done using the *aes* gem in Rails.

5.4 Fuzzy Keyword and Search Algorithms

For efficient searching, the fuzzy sets of all the keywords(both while uploading and searching) are computed. Gram based technique is used to computed the fuzzy sets. The algorithm is given below.

Algorithm 3 Gram-Based Fuzzy Set Construction

```
1: procedure GRAMFUZZYSET($w_i, d$)
2:     if $d > 1$ then
3:         Call GramFuzzySet($w_i, d - 1$)
4:     end if
5:     if $d = 0$ then
6:         $S'_{w_i,d} = \{w_i\}$;
7:     else
8:         for ($k \leftarrow 1$ to $\mid S'_{w_i,d-1} \mid$ ) do
9:             for ($j \leftarrow 1$ to $2* \mid S'_{w_i,d}[k] \mid +1$) do
10:                Set fuzzykeyword as $S'_{w_i,d-1}[k]$;
11:                Delete the j-th character;
12:                if fuzzykeyword is not in $S'_{w_i,d-1}$ then
13:                    Set $S'_{w_i,d} = S'_{w_i,d} \cup \{fuzzykeyword\}$
14:                end if
15:            end for
16:        end for
17:    end if
18: end procedure
```

When a user enters his search queries, the fuzzy set of his keywords are computed and matched against the fuzzy sets stored in the database. The files with the highest match is returned first, followed by the second and so on. Before returning any file, it is verified that the user has permission to access the file.

Algorithm 4 Fuzzy Search

```
1: Read query from search bar
2: Remove stop words, special symbols etc.
3: Add keywords to Keyword set $K$.
4: for ($K_i$ in $K$) do
5:     Compute fuzzy set and append to $K_{fuzz}$
6: end for
7: for ($k_i$ in $K_{fuzz}$) do
8:     Find the files that the keyword appears in and append it to $File\_ids$
9: end for
10: Remove Duplicates from $File\_ids$
11: Remove unauthorised files from $File\_ids$
12: return $File\_ids$
```

5.5 Testing

Testing is an essential step in the development of any software where errors are detected and corrected. Here all the possible test cases are included to ensure that the system is implemented correctly. Following are the approaches undertaken in testing this project.

- Unit Testing
- Integration Testing
- System Testing

5.5.1 Unit Testing

In unit testing, each component is separately tested. The various components of this system to be tested independently are:

User The user module was tested by creating a new user. Additional functionalities like password reset, edit profile were also tested. All the functionalities were found to be working properly.

File Upload Tested by uploading a particular file from the system (text,doc, pdf etc) by browsing file location on the system, entering the keywords related to the file, typing in the owner name, file type etc and finally clicking the upload button. If the file gets stored in the corresponding location in the database and the metadata for the file is created, the test is said to be successful.

TPA Tested by sending an audit request from the client side, verified whether the request is received by the TPA. Once the message is received, a request is sent to the CSP requesting the hash value of the file, verified whether this request is received by the CSP.

CSP Tested whether the files are getting properly stored at the CSPs database. Also verified whether the CSP is sending the correct hash value when requested.

Search Tested the accuracy and relevance of the search results based on the keywords entered during the search and upload time. The results produced were further examined to check whether they are properly classified. Finally, the correctness of the ranking system was verified based on how well the keywords typed in during the search process match the already stored ones.

5.5.2 Integration Testing

User and TPA The user sends an audit request to the TPA, when the TPA completes its verification process, it sends a message to the user whether the file is corrupt or not. The test is successful when this message is received. The authenticity of the received message was verified during Unit testing the TPA module.

TPA and CSP When the TPA challenges the CSP, the CSP responds with the hash value of the corresponding file. The test is successful when the hash value is received. The authenticity of the received hash value was verified during unit testing the TPA module.

5.5.3 System Testing

In system testing, the whole application/system was tested together.This was done as follows:

1. A new user was created.
2. A new file was uploaded. While uploading the file, the owner name, file type, keywords associated with file were all specified.
3. Once the file was uploaded, it was examined whether the file was successfully saved in the CSPs database by checking whether the file was listed in the All Files tab.
4. Then, an audit request was sent by the user.
5. Afterwards, Logged in to TPA to verify whether the audit request has reached TPA (edit if needed :P). TPA then sends a request to the CSP to send the hash value of the stored file.
6. Later, Logged in to CSP to verify whether TPAs request message has been received. CSP responds with the hash value of the stored file.
7. Next, Verified whether the response is obtained by the TPA . TPA then compares both the hash values and responds to the user.
8. Finally, the message was obtained at the user end. This proves the correctness of the entire system.
9. When the user enters keywords in the search bar and clicks the search button relevant results were returned. The results were also ranked base on the number of keywords entered in the search bar that match with the already entered keywords.

Chapter 6

Conclusion

The project Fuzzy Keyword Search over Encrypted Data in Cloud has been completed. All the functional and non-functional requirements were met. A minimal cloud was implemented with the suggested features, so that anybody willing can build on it and extend the work.

Bibliography

[1] C. Ghezzi, M. Jazayeri, and D. Mandrioli, *Fundamentals of Software Engineering.* 2003.

[2] J. Li, Q. Wang, *et al.*, "Fuzzy keyword search over encrypted data in cloud computing", Illinois Institute of Technology.

[3] R. S. Pressman, *Software Engineering - A Practitioner's Approach.* 2001.

[4] J. Wang, H. Ma, *et al.*, "Efficient verifiable fuzzy keyword search over encrypted data in cloud computing", Xidian University.