

BANKING CLOUD SECURITY APPLICATION

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Computer Science

by

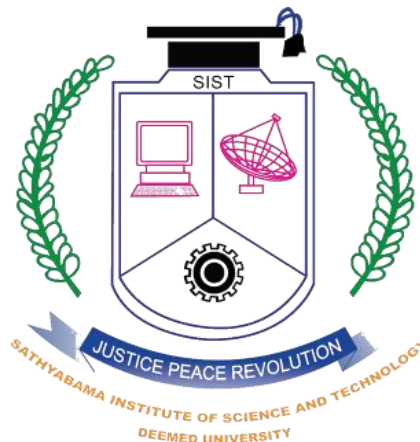
ASHOK KUMAR A

(41290004)

&

YUGASUBAVENKATESH S

(41290064)



**DEPARTMENT OF COMPUTER
SCIENCE SCHOOL OF SCIENCE AND
HUMANITIES**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade “A++” by NAAC

JEPPIAR NAGAR, RAJIV GANDHI SALAI, CHENNAI 600 119

APRIL 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited "A++" Grade by NAAC | 12B Status by UGC I Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **ASHOK KUMAR A (41290004)** and **YUGASUBAVENKATESH S (41290064)** who carried out the project entitled “**BANKING CLOUD SECURITY APPLICATION**” under our supervision from November 2023 TO April 2024.

Internal Guide

Ms. ANCY A,

M.SC.,Assistant Professor

Dean And Head of The Department

Dr. REKHA CHAKRAVARTHI M.E., Ph.D.

Submitted for Viva voce examination held on

26/04/2024

Internal Examiner

External Examiner

DECLARATION

We **ASHOK KUMAR A (41290004)** and **YUGASUBAVENKATESH S (41290064)** hereby declare that the Project Report entitled “**BANKING CLOUD SECURITY APPLICATION**” done by under the guidance of **Ms. ANCY A** Department of Computer Science at **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY**, Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai-600 119 is submitted in partial fulfilment of the requirements for the award of Bachelor of Science degree in Computer Science.

DATE: 26/04/2024

**ASHOK KUMAR A
YUGASUBAVENKATESH S**

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATES

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to Board of Management **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey our thanks to **Dr. REKHA CHAKRAVARTHI M.E., Ph.D. Dean of Arts and Science** for providing us necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to our project guide **Ms. ANCY A** Department of Computer Science for their valuable guidance, suggestions and constant encouragement paved way for the successful completion of our project work.

We wish to express our thanks to all Teaching and Non-teaching staff member of the Department of Computer Science who were helpful in many ways for the completion of the project

ASHOK KUMAR A

YUGASUBAVENKATESH S

ABSTRACT

Nowadays, large amounts of data are stored with cloud service providers. Third- party auditors (TPAs), with the help of cryptography, are often used to verify this data. Cloud Data Auditing Techniques with a Focus on Privacy and Security. It aims to provide a resource based on-demand. It avoids online usage burden of accessing data through internet. Cloud storage supports to maintain data securely in cloud. Cloud is interconnected with group of computers, which is used to store information and run their applications in cloud platform. Through cloud computing, we can access any file, document of user from anywhere in the world. Mainly, cloud can be used for cost savings, high scalability and large storage space. But a major issues in cloud computing is security.

TABLE OF CONTENTS

CH.NO	TITLE	PAGE NO.
	ABSTRACT	
1	LIST OF FIGURES	
2	INTRODUCTION	1
	LITERATURE SUREY	3
	2.1 Inferences from Literature Survey	3
	2.2 Existing System	6
3	SYSTEM ANALYSIS	8
	3.1 Risk Analysis of the Project	8
4	SYSTEM SPECIFICATION	11
	4.1 Software specification	11
	4.2 Hardware specification	11
5	DESCRIPTION OF PROPOSED SYSTEM	12
	5.1 Architecture / Overall Design of Proposed System	20
	5.2 Description of Software for Implementation and system	21
6	SYSTEM TESTING	25
	6.1 Types of tests	25

CH.NO	NAME OF THE TOPIC	PAGENO.
7	MODULE DESCRIPTION	28
8	CONCLUSION	29
9	APPENDIX	30
	9.1 Coding	30
	9.2 Screenshots	35
10	REFERENCE	39

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO.
4.1.1	Class Diagram	14
4.1.2	Usecase Diagram	15
4.1.3	Sequence Diagram	16
4.1.4	Component Diagram	17
4.1.5	Activity Diagram	28
4.2	System Architecture Diagram	21

CHAPTER 1

INTRODUCTION

Storing large amounts of data with cloud service providers (CSPs) raises concerns about data protection. Data integrity and privacy can be lost because of the physical movement of data from one place to another by the cloud administrator, malware, dishonest cloud providers, or other malicious users who might distort the data.¹ Hence, saved data corrections must be verified at regular intervals.

Nowadays, with the help of cryptography, verification of remote (cloud) data is performed by third-party auditors (TPAs). TPAs are also appropriate for public auditing, offering auditing services with more powerful computational and communication abilities than regular users.³ In public auditing, a TPA is designated to check the correctness of cloud data without retrieving the entire dataset from the CSP. However, most auditing schemes don't protect user data from TPAs; hence, the integrity and privacy of user data are lost. Our research focuses on cryptographic algorithms for cloud data auditing and the integrity and privacy issues that these algorithms face. Many approaches have been proposed in the literature to protect integrity and privacy; they're generally classified according to data's various states: static, dynamic, multi owner, multiuser, and so on.

We provide a systematic guide to the current literature regarding comprehensive methodologies. We not only identify and categorize the different approaches to cloud data integrity and privacy but also compare and analyze their relative merits. For example, our research lists the

strengths and weaknesses of earlier work on cloud auditing, which will enable researchers to design new methods. Although related topics such as providing security to the cloud are beyond this article's scope, cloud data auditing requires explicit attention.

It represents the fusion of cutting-edge technologies in finance. Blockchain, renowned for its robust security and decentralized ledger system, intertwines with cloud-based banking, leveraging the scalability and accessibility of cloud computing. This integration promises a paradigm shift in financial operations by ensuring airtight security measures while enabling seamless transactions. The utilization of blockchain within cloud banking not only guarantees a high level of protection for sensitive financial data but also fortifies the system against potential faults or disruptions. This title encapsulates a forward-looking approach to banking, embracing innovation to deliver a trustworthy, resilient, and efficient platform for conducting transactions in the modern digital age.

CHAPTER 2

LITERATURE SURVEY

Literature survey is the most important step in any kind of research. Before we start developing, we need to study the previous papers of our domain which we are working and on the basis of study we can predict or generate the drawback and start working with the reference of previous papers.

2.1 INFERENCES FROM LITRERATURE SURVEY

TITLE	AUTHOR/YEAR	DESCRIPTION
A surveyon auditing techniques used for preserving privacy of data stored on cloud	Riya Sudam Bote, Shikha Virendar Chandel / 2022	Providing security to the stored data on the cloud is one of the important challenges in cloud computing. Encrypted data which is stored on the cloud may be viewed or modified by the cloud service provider. To overcome this problem many techniques have been developed but, those cannot guarantee accurately about the security of the stored data. These modifications of the data by the service provider or by others should also be known to the data owner. For such purpose, data tagging technique can be used to audit the data. Auditing is done by using Third Party Auditor tpa.

TITLE	AUTHOR/YEAR	DESCRIPTION
Cloud Security Auditing Challenges and Emerging Approaches	Venkata Sasank Pagolo Ganapathi Panda / 2021	IT auditors collect information on an organization's information systems, practices, and operations and critically analyze the information for improvement. One of the primary goals of an IT audit is to determine if the information system and its maintainers are meeting both the legal expectations of protecting customer data and the company standards of achieving financial success against various security threats. These goals are still relevant in the newly emerging cloud computing model of business, but they need customization. There are clear differences between cloud and traditional IT security auditing. In this article, the authors explore potential challenges unique to cloud security auditing; examine additional challenges specific to particular cloud computing domains such as banking, medical, and government sectors; and present emerging cloud-specific security auditing approaches and provide critical analysis.

TITLE	AUTHOR/YEAR	DESCRIPTION
Dynamic-Hash-Table Based Public Auditing for Secure Cloud Storage	Muruga Gurjar, Parth Naik / 2018 Description	Cloud storage is an increasingly popular application of cloud computing, which can provide ondemand outsourcing data services for both organizations and individuals. However, users may not fully trust the cloud service providers (CSPs) in that it is difficult to determine whether the CSPs meet their legal expectations for data security. Therefore, it is critical to develop efficient auditing techniques to strengthen data owners' trust and confidence in cloud storage. In this paper, we present a novel public auditing scheme for secure cloud storage based on dynamic hash table (DHT), which is a new two-dimensional data structure located at a third parity auditor (TPA) to record the data property information for dynamic auditing. Differing from the existing works, the proposed scheme migrates the authorized information from the CSP to the TPA.

TITLE	AUTHOR/YEAR	DESCRIPTION
Cloud Data Auditing Techniques with a Focus on Privacy and Security	OmkarS ,Shruthika H , Amit A / 2019	Nowadays, large amounts of data are stored with cloud service providers. Third-party auditors (TPAs), with the help of cryptography, are often used to verify this data. However, most auditing schemes don't protect cloud user data from TPAs. A review of the state of the art and research in cloud data auditing techniques highlights integrity and privacy challenges, current solutions, and future research directions.

2.2 EXISTING SYSTEM

While Cloud Computing makes these advantages more appealing than ever, it also brings new and challenging security threats towards users' outsourced data. Since cloud service providers (CSP) are separate administrative entities, data outsourcing is actually relinquishing user's ultimate control over the fate of their data. As a result, the correctness of the data in the cloud is being put at risk due to the following reasons. Although the infrastructures under the cloud are much more powerful and reliable than personal computing devices, they are still facing the broad range of both internal and external threats for data integrity. There do exist various motivations for CSP to behave unfaithfully towards the cloud users regarding their outsourced data status. For examples, CSP might reclaim storage for monetary reasons by discarding data that has not been or is rarely accessed, or even hide data loss incidents to maintain a reputation. In short, although outsourcing data to the cloud is economically attractive for long-term large-scale storage, it does not immediately offer any guarantee on data integrity and availability. This problem, if not properly addressed, may impede the success of cloud architecture.

Abuse and Nefarious Use of Cloud Computing:

IaaS providers offer their customers the illusion of unlimited compute, network, and storage capacity often coupled with a 'frictionless' registration process where anyone with a valid credit card can register and immediately begin using cloud services. Some providers even offer free limited trial periods. By abusing the relative anonymity behind these registration and usage models, spammers, malicious code authors, and other criminals have been able to conduct their activities with relative impunity. PaaS providers have traditionally suffered most from this kind of attacks; however, recent evidence shows that hackers have begun to target IaaS vendors as well. Future areas of concern include password and key cracking, DDOS, launching dynamic attack points, hosting malicious data, botnet command and control, building rainbow tables, and CAPTCHA solving farms.

Insecure Interfaces and APIs:

Cloud computing providers expose a set of software interfaces or APIs that customers use to manage and interact with cloud services. Provisioning, management, orchestration, and monitoring are all performed using these interfaces. The security and availability of general cloud services is

dependent upon the security of these basic APIs. This introduces the complexity of the new layered API; it also increases risk, as organizations may be required to relinquish their credentials to third parties in order to enable their agency.

Malicious Insiders:

The threat of a malicious insider is well-known to most organizations. This threat is amplified for consumers of cloud services by the convergence of IT services and customers under a single management domain, combined with a general lack of transparency into provider process and procedure.

Shared Technology Issues:

IaaS vendors deliver their services in a scalable way by sharing infrastructure. Often, the underlying components that make up this infrastructure were not designed to offer strong isolation properties for a multi-tenant architecture.

Data Loss or Leakage:

There are many ways to compromise data. Deletion or alteration of records without a backup of the original content is an obvious example.

Account or Service Hijacking:

Account or service hijacking is not new. Attack methods such as phishing, fraud, and exploitation of software vulnerabilities still achieve results

CHAPTER 3

SYSTEM ANALYSIS

3.1 RISK ANALYSIS OF THE PROJECT

The software requirement specification gives the system specification in which process requirements are presented in an easily understandable way. Thus it contains all the inputs required, processes in the system and outputs produced by the system.

Software Requirements Specification plays an important role in creating quality software solutions. Specification is basically a representation process. Requirements are represented in a manner that ultimately leads to successful software implementation.

Requirements may be specified in a variety of ways. However there are some guidelines worth following

- Representation format and content should be relevant to the problem.
- Information contained within the specification should be nested.

Requirement analysis enables the system engineer to specify software function and perform, indicate software's interface with the other system elements, and establish constraints that software must meet. Requirement analysis allows the analyst to refine the software allocation and build models of the data, functional and behavioral domains that will be treated by software.

The first step is to understand the user's requirement within the framework of the organization's objectives and the environment in which the system is installed. Considerations are given to the user to carry on with the work within the organization's specified objectives.

Using swings in java we will develop the proposed system. The proposed application can be implemented by taking minimum of three systems into consideration. The server is implemented in one system TPA is implemented in another system and client can be implemented from n no of systems it can be implemented on any operating system like windows or linux. The client system will store data like files images on to server through TPA. The TPA will store the metadata information of file on TPA where as server will store files as well as metadata data information about the files. Whenever the client asks for the verification of files on cloud the TPA will check for the data integrity on the server. This application demands minimum three systems should be connected within a network.

FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

CHAPTER 4

SYSTEM SPECIFICATION

4.1 SOFTWARE SPECIFICATION

Server and Client Side Technology

AWT and Swings.

Database

MySQL

Operating System

Windows10/11

Coding language

Java

4.2 HARDWARE SPECIFICATION

Processor

Intel ® Pentium ®

Speed

1.6 GHz and Above

Hard Disk

80GB

CHAPTER 5

DESCRIPTION OF PROPOSED SYSTEM

PROPOSED SYSTEM

The proposed system can be summarized as the following three aspects:

1. We motivate the public auditing system of data storage security in Cloud Computing and provide a privacy-preserving auditing protocol, i.e., our scheme supports an external auditor to audit user's outsourced data in the cloud without learning knowledge on the data content.
2. To the best of our knowledge, our scheme is the first to support scalable and efficient public auditing in the Cloud Computing. In particular, our scheme achieves batch auditing where multiple delegated auditing tasks from different users can be performed simultaneously by the TPA.
3. We prove the security and justify the performance of our proposed schemes through concrete experiments and comparisons with the state-of-the-art.

ADVANTAGES

- Novel automatic and enforceable logging mechanism in the cloud.
- Proposed architecture is platform independent and highly decentralized, in that it does not require any dedicated authentication or storage system in place.
- Provide a certain degree of usage control for the protected data after these are delivered to the receiver
- The results demonstrate the efficiency, scalability, and granularity of our approach. We also provide a detailed security analysis and discuss the reliability and strength of our architecture.

CLASS DIAGRAM AUDITING FOR SECURE DATA STORAGE IN CLOUD

This class diagram contains four classes that are Server, TPA, Client and Database. Server will perform operations like it maintains client details & session information, stores details & files and generate graphs. Client will perform operations like registration, login, upload files, download files, verify documents, add blocks, delete blocks. TPA will perform operations like take file size, divide file into blocks, maintain metadata information, send response and verification message. And this diagram shows the relationship between this classes.

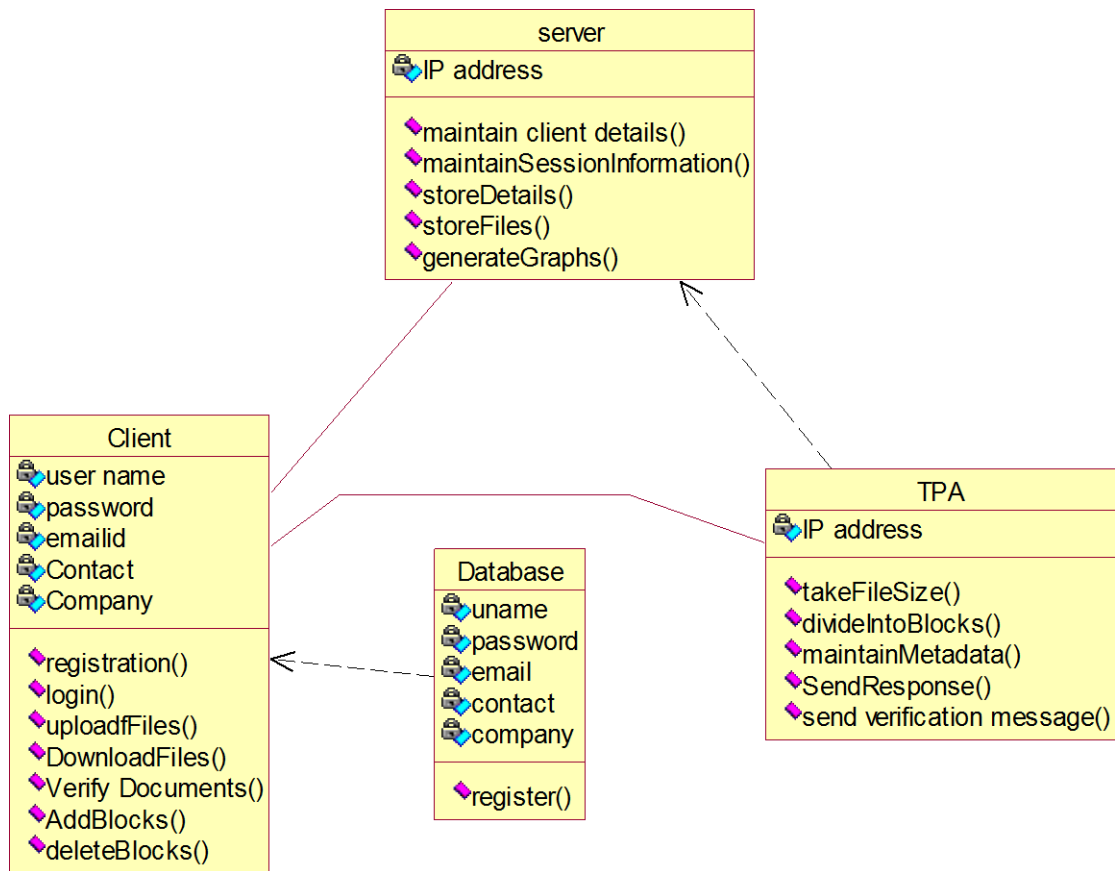


FIG 4.1.1 CLASS DIAGRAM

USECASE DIAGRAM:

This use case diagram contains three actors that are Server, TPA, Client and. Server will perform operations like it maintains client details & session information, stores details & files and generate graphs. Client will perform operations like registration, login, upload files, download files, verify documents, add blocks, delete blocks. TPA will perform operations like take file size, divide file into blocks, maintain metadata information, send response and verification message. And this diagram shows the use cases of each actor and relationship between this actors and use cases.

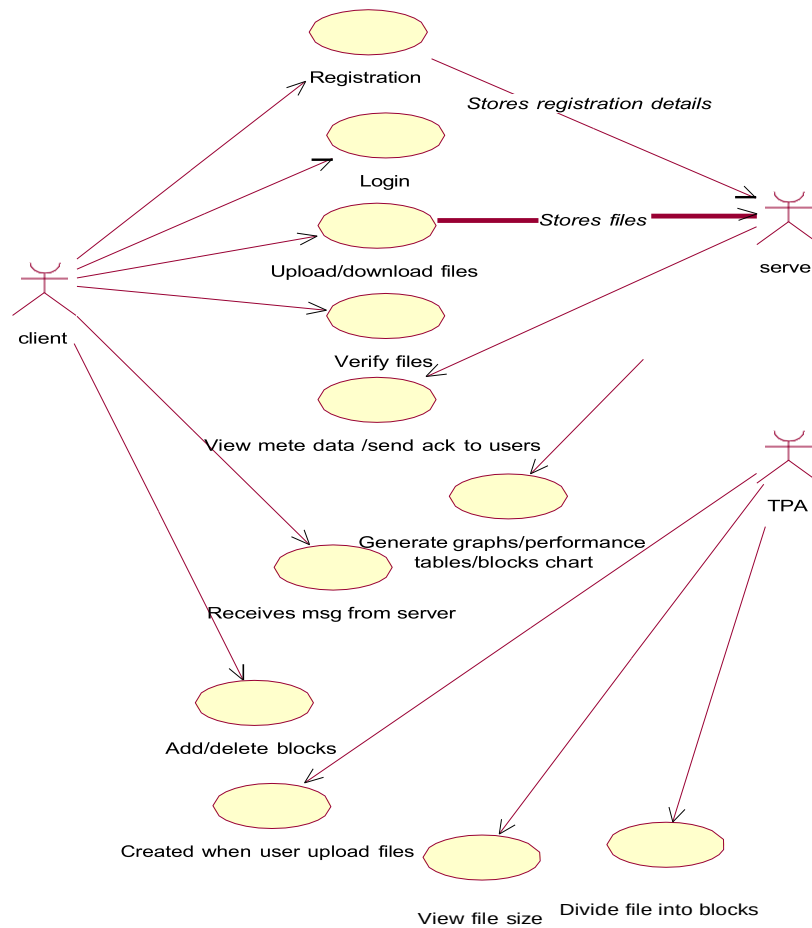


FIG 4.1.2 USECASE DIAGRAM

SEQUENCE DIAGRAM:

This sequence diagram contains three objects that are Server, TPA, Client and. Server will perform operations like it maintains client details & session information stores details & files and generate graphs. Client will perform operations like registration, login, upload files, download files, verify documents, add blocks, delete blocks. TPA will perform operations like take file size, divide file into blocks, maintain metadata information, send response and verification message. And this diagram shows the sequence of actions performed between these objects.

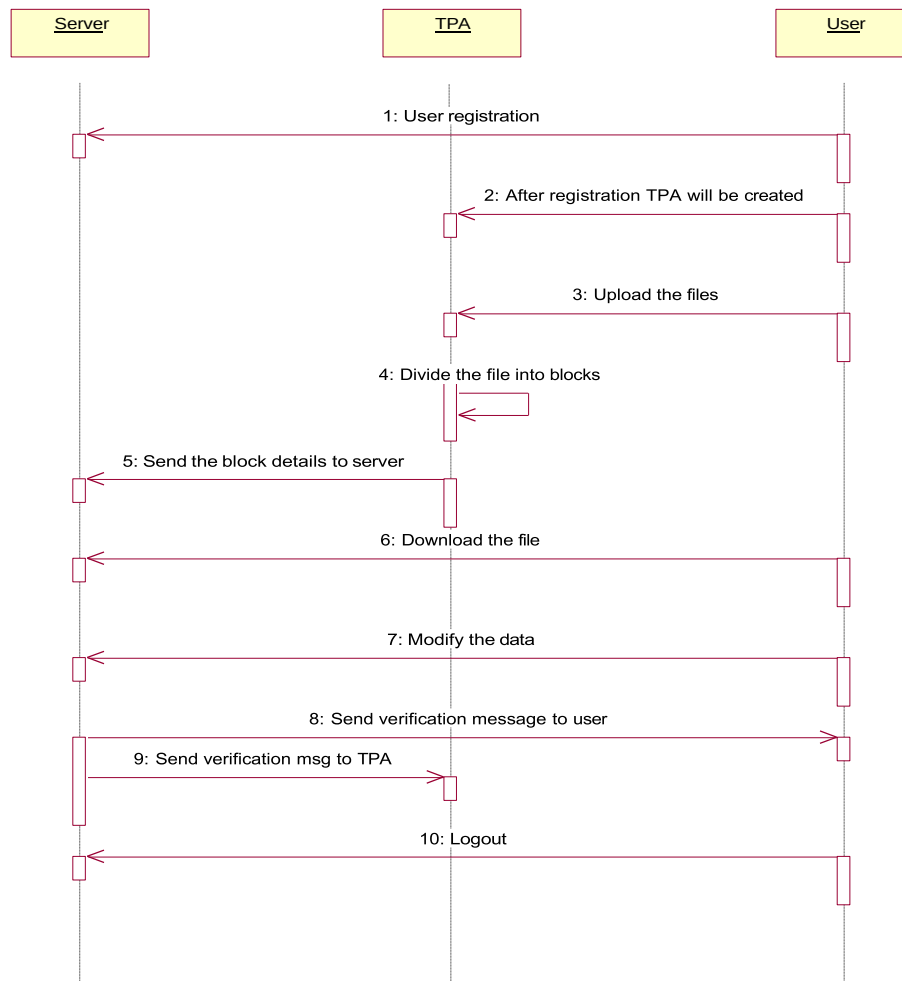


FIG 4.1.3 SEQUENCE DIAGRAM

COMPONENT DIAGRAM:

This component diagram contains three components that are Server, TPA, Client and. Server will perform operations like it maintains client details & session information stores details & files and generate graphs. Client will perform operations like registration, login, upload files, download files, verify documents, add blocks, delete blocks. TPA will perform operations like take file size, divide file into blocks, maintain metadata information, send response and verification message. And this diagram shows the actions performed by these components.

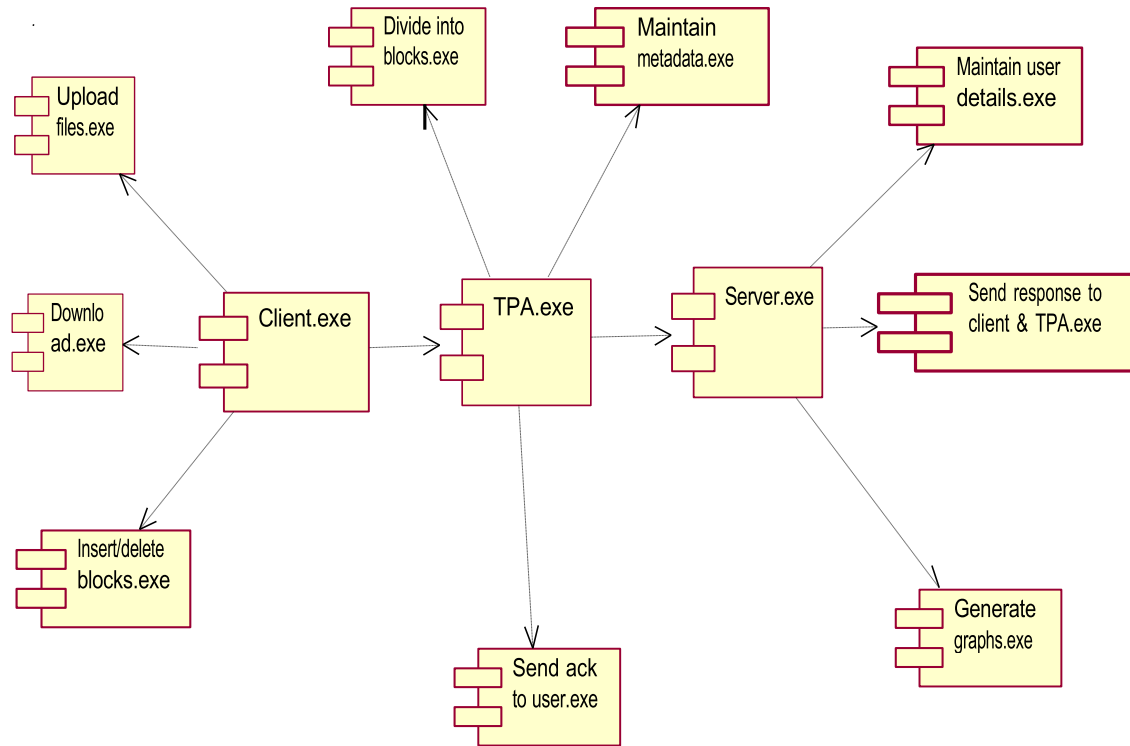


FIG 4.1.4 COMPONENT DIAGRAM

ACTIVITY DIAGRAM:

This activity diagram contains three activities that are Server, TPA, Client and. This diagram shows the flow of control between these activities.

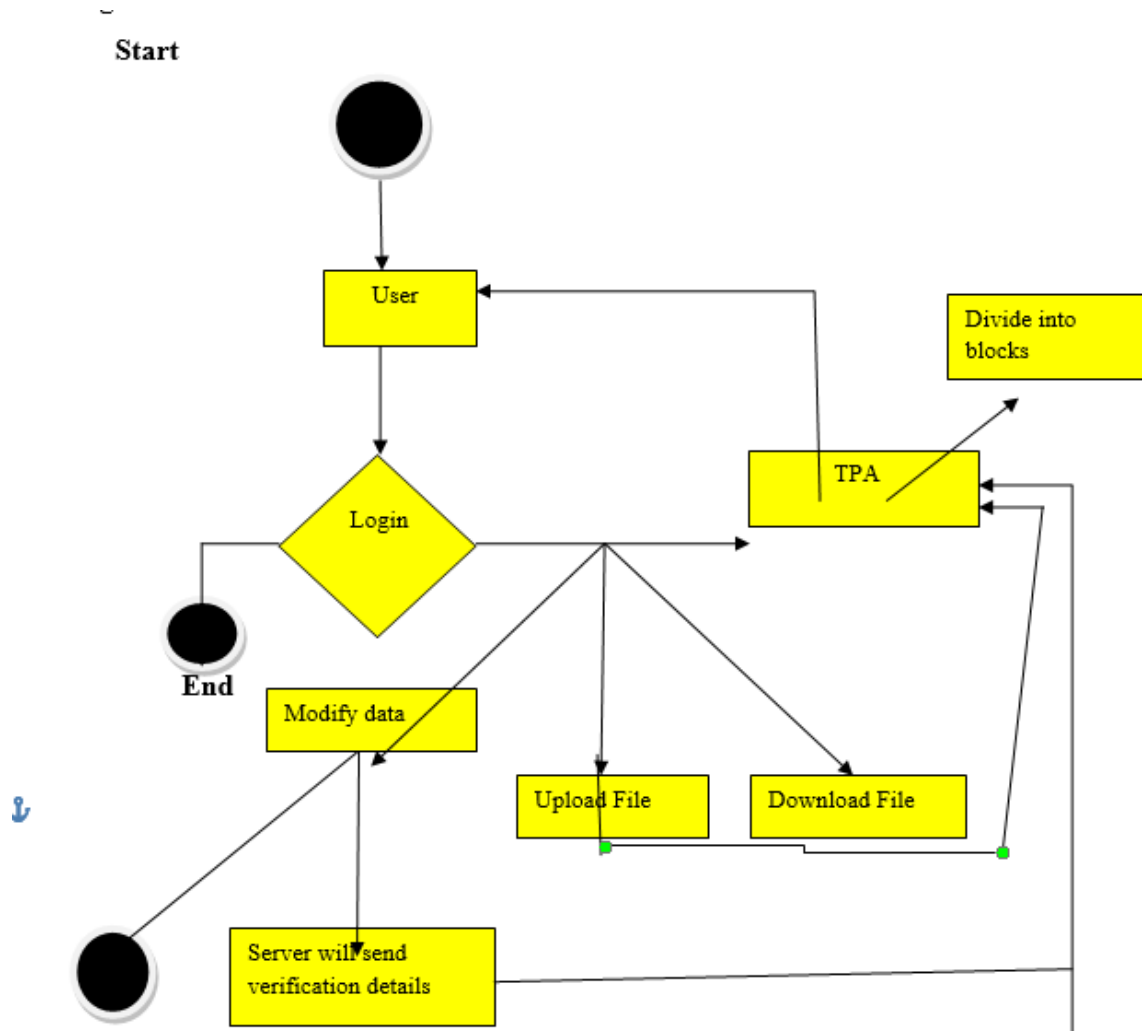


FIG 4.1.5 ACTIVITY DIAGRAM

UNIFIED MODELING LANGUAGE

The Unified Modelling Language allows the software engineer to express an analysis model using the modelling notation that is governed by a set of syntactic semantic and pragmatic rules.

A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows.

- **User Model View**

- This view represents the system from the user's perspective.
- The analysis representation describes a usage scenario from the end-users perspective.

- **Structural model view**

- In this model the data and functionality are arrived from inside the system.
- This model view models the static structures.

- **Behavioral Model View**

It represents the dynamic of behavioral as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

- **Implementation Model View**

In this the structural and behavioral as parts of the system are represented as they are to be built.

- **Environmental Model View**

In this the structural and behavioral aspects of the environment in which the system is to be implemented are represented.

UML is specifically constructed through two different domains they are:

- UML Analysis modeling, this focuses on the user model and structural model views of the system.

- UML design modeling, which focuses on the behavioral modeling, implementation modeling and environmental model views.

Use case Diagrams represent the functionality of the system from a user's point of view. Use cases are used during requirements elicitation and analysis to represent the functionality of the system. Use cases focus on the behavior of the system from external point of view.

Actors are external entities that interact with the system. Examples of actors include users like administrator, bank customer ...etc., or another system like central database.

5.1 ARCHITECTURE/OVERALL DESIGN OF PROPOSED SYSTEM SYSTEM ARCHITECTURE

Describing the overall features of the software is concerned with defining the requirements and establishing the high level of the system. During architectural design, the various web pages and their interconnections are identified and designed. The major software components are identified and decomposed into processing modules and conceptual data structures and the interconnections among the modules are identified. The following modules are identified in the proposed system.

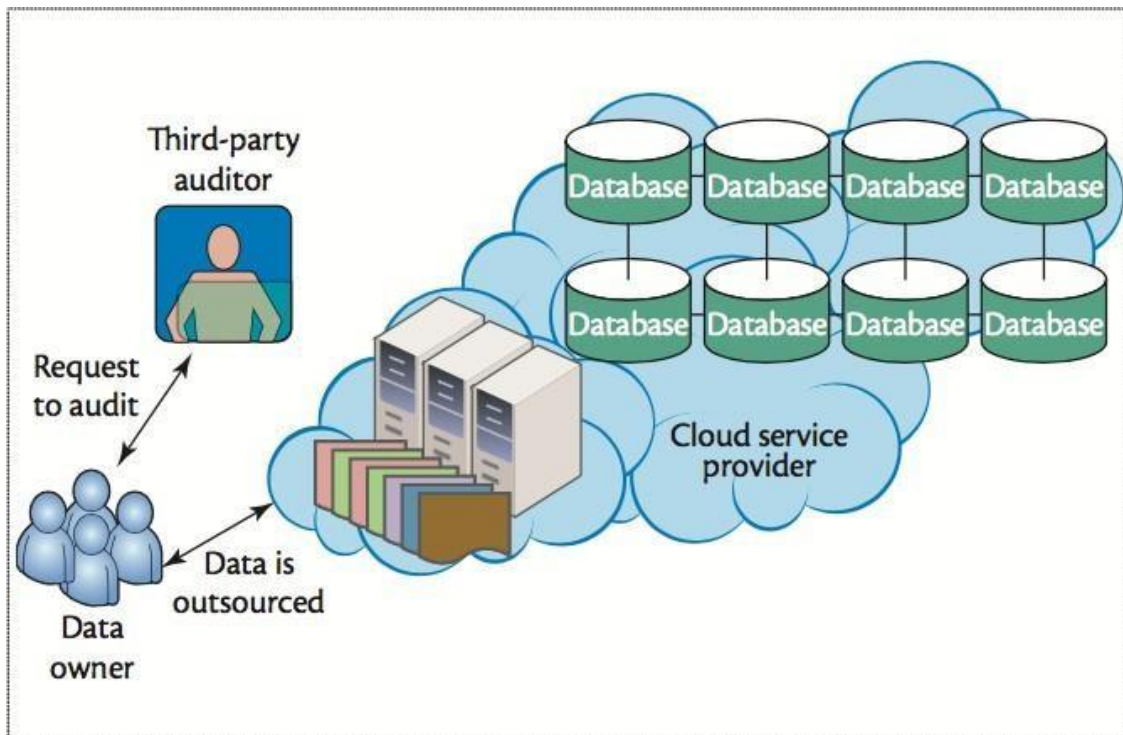


FIG 4.2 SYSTEM ARCHITECTURE DIAGRAM

5.2 DESCRIPTION OF SOFTWARE FOR IMPLEMENTATION AND SYSTEM

The purpose of the Software Requirement Specification is to produce the specification of the analysis task and also to establish complete information about the requirement, behavior and also the other constraint like functional performance and so on. The main aim of the Software Requirement Specification is to completely specify the technical requirements for the software product in a concise and in unambiguous manner.

NETBEANS IDE

In this project the NetBeans is used as an IDE. NetBeans IDE Code combines the simplicity of a source code editor with powerful developer tooling, like Intelli Sense code completion and debugging. First and foremost, it is an editor that gets out of our way. The delightfully frictionless edit-build-debug cycle means less time fiddling with our environment, and more time executing on

our ideas. NetBeans Code supports macOS, Linux, and Windows - so we can hit the ground running, no matter the platform. At its heart, NetBeans Code features a lightning fast source code editor, perfect for day-to-day use. With support for hundreds of languages, NetBeans Code helps us be instantly productive with syntax highlighting, bracket-matching, auto-indentation, box-selection, snippets, and more. Intuitive keyboard shortcuts, easy customization and community-contributed keyboard shortcut mappings let us navigate our code with ease. For serious coding, we'll often benefit from tools with more code understanding than just blocks of text. NetBeans Code includes built-in support for Intelli Sense code completion, rich semantic code understanding and navigation, and code refactoring. And when the coding gets tough, the tough get debugging. Debugging is often the one feature that developers miss most in a leaner coding experience, so we made it happen. NetBeans Code includes an interactive debugger, so we can step through source code, inspect variables, view call stacks, and execute commands in the console. NetBeans also integrates with build and scripting tools to perform common tasks making everyday workflows faster. NetBeans has support for Git so we can work with source control without leaving the editor including viewing pending changes diffs. NetBeans Code includes a public extensibility model that lets developers build and use extensions, and richly customize their edit-build debug experience.

JAVA

Java is one of the world's most important and widely used computer languages, and it has held this distinction for many years. Unlike some other computer languages whose influence has waned with passage of time, while Java's has grown.

APPLICATION OF JAVA

Java is widely used in every corner of world and of human life. Java is not only used in softwares but is also widely used in designing hardware controlling software components. There are more than 930 million JRE downloads each year and 3 billion mobile phones run java.

Following are some other usage of Java:

1. Developing Desktop Applications
2. Web Applications like LinkedIn.com, Snapdeal.com etc

3. Mobile Operating System like Android
4. Embedded Systems
5. Robotics and games etc.

FEATURES OF JAVA

The prime reason behind creation of Java was to bring portability and security feature into a computer language. Beside these two major features, there were many other features that played an important role in moulding out the final form of this outstanding language. Those features are;

1. Simple

Java is easy to learn and its syntax is quite simple, clean and easy to understand. The confusing and ambiguous concepts of C++ are either left out in Java or they have been re-implemented in a cleaner way.

Eg: Pointers and Operator Overloading are not there in java but were an important part of C++.

2. Object Oriented

In java everything is Object which has some data and behaviour. Java can be easily extended as it is based on Object Model.

3. Robust

Java makes an effort to eliminate error prone codes by emphasizing mainly on compile time error checking and runtime checking. But the main areas which Java improved were Memory Management and mishandled Exceptions by introducing automatic Garbage Collector and Exception Handling.

4. Platform Independent

Unlike other programming languages such as C, C++ etc. which are compiled into platform specific machines. Java is guaranteed to be write- once, run-anywhere language.

On compilation Java program is compiled into byte code. This byte code is platform independent and can be run on any machine, plus this byte code format also provide security. Any machine

with Java Runtime Environment can run Java Programs.

5. Secure

When it comes to security, Java is always the first choice. With java secure features it enable us to develop virus free, temper free system. Java program always runs in Java runtime environment with almost null interaction with system OS, hence it is more secure.

6. Multi-Threading

Java multithreading feature makes it possible to write program that can do many tasks simultaneously. Benefit of multithreading is that it utilizes same memory and other resources to execute multiple threads at the same time, like While typing, grammatical errors are checked along.

7. Architectural Neutral

Compiler generates byte codes, which have nothing to do with a particular computer architecture, hence a Java program is easy to interpret on any machine.

8. Portable

Java Byte code can be carried to any platform. No implementation dependent features. Everything related to storage is predefined, example: size of primitive data types

10.) High Performance

Java is an interpreted language, so it will never be as fast as a compiled language like C or C++. But, Java enables high performance with the use of just-in-time compiler.

CHAPTER 6

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. 0Each test type addresses a specific testing requirement.

6.1 TYPES OF TESTS :

UNIT TESTING :

Unit testing involves the design of test cases that validate that the internal Program logic is functioning properly, and that program input produces 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 business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

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.

FUNCTIONAL TESTING :

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 is used to identified classes of valid input must be accepted. Invalid Input is used to identified classes of invalid input must be rejected. Functions is used to identified functions must be exercised. Output is used to identify classes of application outputs.

Systems/Procedures is used to 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.**

SYSTEM TESTING :

System testing ensures that the entire integrated software system meets

requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system

integration test. System testing is based on process descriptions and flows, emphasizing pre-Driven process links and integration points.

WHITE BOX TESTING :

White Box Testing is a testing in which the software tester has knowledge of the inner workings, structure and language of the software, or at least it's purpose. It is used to test areas that cannot be reached from a black box level.

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 requirement 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.

TEST STRATEGY AND APPROACH

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

1. All field entries must work properly.
2. Pages must be activated from the identified link.
3. The entry screen, messages and responses must not be delayed. Features to be tested.
4. Verify that the entries are of the correct format No duplicate entries should be allowed.
5. All links should take the user to the correct page.

INTEGRATION TESTING:

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications.

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.

CHAPTER 7

MODULES DISCRIPTION

MODULE DESCRIPTION:

The system is proposed to have the following modules:

- Clint Module
- TPA module
- Cloud Module

CLINT MODULE:

Admin is allowed to check which user registered and which data is stored in the cloud space area

.

TPA MODULE :

Check that data is modified or not if modified that information send to user.

USER MODULE:

User can register and he can login with his user id and password and he can upload the data to cloud space area.

CHAPTER 8

CONCLUSION

In this project, we propose a cloud auditing system for data storage security in Cloud Computing. We utilize the homomorphic linear authenticator and random masking to guarantee that the TPA would not learn any knowledge about the data content stored on the cloud server during the efficient auditing process, which not only eliminates the burden of cloud user from the tedious and possibly expensive auditing task, but also alleviates the users' fear of their outsourced data leakage. Considering TPA may concurrently handle multiple audit sessions from different users for their outsourced data files, we further extend our privacy preserving public auditing protocol into a multi-user setting, where the TPA can perform multiple auditing tasks in a batch manner for better efficiency. Extensive analysis shows that our schemes are provably secure and highly

CHAPTER 9

APPENDIX

9.1 CODING

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Secure blockchain banking with error recovery process</title>
<!-- <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" /> -->
<link rel="stylesheet" href="layout/styles/layout.css" type="text/css" />
<script type="text/javascript" src="layout/scripts/jquery.min.js"></script>
<!-- <script type="text/javascript" src="layout/scripts/jquery.slidepanel.setup.js"></script>
<!-- Homepage Only Scripts -->
<script type="text/javascript" src="layout/scripts/jquery.cycle.min.js"></script>
<script type="text/javascript">
$(function() {
    $('#featured_slide').after('<div id="fsn"><ul id="fs_pagination">').cycle({
        timeout: 5000,
        fx: 'fade',
        pager: '#fs_pagination',
        pause: 1,
        pauseOnPagerHover: 0
    });
});
</script>
<!-- End Homepage Only Scripts -->
</head>
<body>

<!--
#####
```

```
##### -->
<div class="wrapper col1">
  <div id="header">
    <div id="logo">
      <h2><a href="index.html">Secure blockchain banking with</a><span>error recovery
process</span><h2>
    <p></p>
  </div>
  <div id="topnav">
    <ul>
      <li class="active"><a href="index.html">Home</a></li>
      <li><a href="Client.jsp">Client</a></li>
      <li><a href="TPALogin.jsp">TPA</a></li>
      <li><a href="CloudServer.jsp">Cloud Server</a></li>

    </ul>
  </div>
  <br class="clear" />
</div>
</div>
<!--
#####
##### -->
<div class="wrapper col2">
  <div id="featured_slide">
    <div class="featured_box"><a href="#"></a>
    <div class="floater">
      <h3>Secure blockchain banking with error recovery process</h3>
    </div>
  </div>
  <div class="featured_box"><a href="#"></a>
  <div class="floater">
```

<h2>Abstract</h2>

<p align="justify">In cloud computing, data owners host their data on cloud servers and users (data consumers) can access the data from cloud servers. Due to the data outsourcing, however, this new paradigm of data hosting service also introduces new security challenges, which requires an independent auditing service to check the data integrity in the cloud. Some existing remote integrity</p>

<p class="readmore">Continue Reading »</p>

</div>

</div>

<div class="featured_box">

<div class="floater">

<h2>Existing System</h2>

<p align="justify">To solve the data privacy problem, existing method is to generate an encrypted proof with the challenge stamp by using the Bilinearity property of the bilinear pairing, such that the auditor cannot decrypt it but can verify the correctness of the proof. Without using the mask technique, this method does not require any trusted organizer during the batch auditing for multiple clouds. Therefore, existing method can greatly reduce the computing loads of the auditor by moving it to the cloud server. </p>

<p class="readmore">Continue Reading »</p>

</div>

</div>

<div class="featured_box">

<div class="floater">

<h2>Proposed System</h2>

<p align="justify">To improve the draw backs of existing system a modified dynamic auditing protocol is introduced, This protocol contains 1.Time stamp value to verify the validity of data 2.Index table for dynamic owner as well as data. This system includes 4 modules
1. Multi cloud storages 2. Modified dynamic auditing 3.

<p class="readmore">Continue Reading
»</p>

</div>

</div>

</div>

</div>

</div>

<!--

#####

-->

<div class="wrapper col3">

<div id="homecontent">

<div class="fl_left">

<div class="column2">

</div>

<div class="column2">

<h2>Abstract</h2>

<p align="justify" style="text-decoration: blink; font-size: 15px;">

In cloud computing, data owners host their data on cloud servers and users (data consumers) can access the data from cloud servers. Due to the data outsourcing, however, this new paradigm of data hosting service also introduces new security challenges, which requires an independent auditing service to check the data integrity in the cloud. Some existing remote integrity checking methods can only serve for static archive data and thus cannot be applied to the auditing service since the data in the cloud can be dynamically updated. Thus, an efficient and secure dynamic auditing protocol is desired to convince data owners that the data are correctly stored in the cloud. This paper, first designs an auditing framework for cloud storage systems and proposes an efficient and privacy-preserving auditing protocol. Then, this will be extended for auditing protocol to support the data dynamic operations, which is efficient and provably secure in the random oracle model. Further the auditing protocol is extended to support batch auditing for both multiple owners and multiple clouds, without

using any trusted organizer. The analysis and simulation results show that the auditing protocols are secure and efficient, especially it reduce the computation cost of the auditor.

```
</p>

</div>
</div>
<div class="fl_right">

</div>
<br class="clear" />
</div>
</div>
<!--
#####
##### -->
<div class="wrapper col4">
  <div id="footer">

    </div>
    <br class="clear" />
  </div>
</div>
<!--
#####
##### -->

</body>
</html>
```

9.2 SCREENSHOTS



Abstract

In cloud computing, data owners host their data on cloud servers and users (data consumers) can access the data from cloud servers. Due to the data outsourcing, however, this new paradigm of data hosting service also introduces new security challenges, which requires an effective method to ensure the data integrity in the cloud.

localhost:8080/Secure_blockchain_banking_with_error_recovery_process-50/#

Home page



Secure blockchain banking with error recovery process



Welcome To Client Login

User Email	
Password	
Login	Reset

localhost:8080/Secure_blockchain_banking_with_error_recovery_process-50_/Client.jsp#

[Click Here to Register](#)

Login page



Secure blockchain banking with error recovery process

Upload File Here

File ID	7
Subject	
Choose File	Choose File No file chosen
Upload	

localhost:8080/Secure_blockchain_banking_with_error_recovery_process-50_/UploadFile.jsp#

File upload page



An Efficient and Secure Dynamic Auditing Protocol for Data Storage in Cloud Computing



File ID	File Name	Total Bytes	Date	Download
---------	-----------	-------------	------	----------

Download page

REFERENCES

- [1.] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Storage Security in Cloud Computing," Proc. IEEE INFOCOM '10, Mar. 2010.

- [2.] P. Mell and T. Grance, "Draft NIST Working Definition of Cloud Computing," <http://csrc.nist.gov/groups/SNS/cloudcomputing/index.html>, June 2009.

- [3.] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R.H. Katz, A. Konwinski, G. Lee, D.A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the Clouds: A Berkeley View of Cloud Computing," Technical Report UCB-EECS-2009-28, Univ. of California, Berkeley, Feb. 2009.

- [4.] Cloud Security Alliance, "Top Threats to Cloud Computing," <http://www.cloudsecurityalliance.org>, 2010.

- [5.] M. Arrington, "Gmail Disaster: Reports of Mass Email Deletions," <http://www.techcrunch.com/2006/12/28/gmail-disasterreportsof-mass-email-deletions/>, 2006.