A PROXY RE-ENCRYPTION APPROACH TO SECURE DATA SHARING IN THE INTERNET OF THINGS BASED ON BLOCKCHAIN

[1]UPPARI SAI RAHUL, [2]BANDI SRI RAM, [3]EMBADI ROHITH KUMAR, [4]BELLALA VINAY KUMAR

[1,2,3,4]Student
BHARATH INSTITUTE OF HIGHER EDUCATION AND RESEARCH

***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 issue in cloud computing is security.**

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.

OVERVIEW OF CLOUD COMPUTING:

Cloud computing is a computing paradigm, where a large pool of systems are connected in private or public networks, to provide dynamically scalable infrastructure for application, data and file storage. With the advent of this technology, the cost of computation, application hosting, content storage and delivery is reduced significantly.
Cloud computing is a practical approach to experience direct cost benefits and it has the potential to transform a data center from a capital-intensive set up to a variable priced environment.
The idea of cloud computing is based on a very fundamental principal of „reusability of IT capabilities'. The difference that cloud computing brings compared to traditional concepts of "grid computing", "distributed computing", "utility computing", or "autonomic computing" is to broaden horizons across organizational boundaries

Cloud Computing Models

Cloud Providers offer services that can be grouped into three categories.

Software as a Service (SaaS):

In this model, a complete application is offered to the customer, as a service on demand. A single instance of the service runs on the cloud & multiple end users are serviced. On the customers‟ side, there is no need for upfront investment in servers or software licenses, while for the provider, the costs are lowered, since only a single application needs to be hosted & maintained. Today SaaS is offered by companies such as Google, Salesforce, Microsoft, Zoho, etc.

Platform as a Service (Paas):

Here, a layer of software, or development environment is encapsulated & offered as a service, upon which other higher levels of service can be built. The customer has the freedom to build his own applications, which run on the provider‟s infrastructure. To

meet manageability and scalability requirements of the applications, PaaS providers offer a predefined combination of OS and application servers, such as LAMP platform (Linux, Apache, MySql and PHP), restricted J2EE, Ruby etc. Google‟s App Engine, Force.com, etc are some of the popular PaaS examples.

Infrastructure as a Service (Iaas):

IaaS provides basic storage and computing capabilities as standardized services over the network. Servers, storage systems, networking equipment, data centre space etc. are pooled and made available to handle workloads. The customer would typically deploy his own software on the infrastructure. Some common examples are Amazon, GoGrid, 3 Tera, etc.

Understanding Public and Private Clouds

Enterprises can choose to deploy applications on Public, Private or Hybrid clouds. Cloud Integrators can play a vital part in determining the right cloud path for each organization.

Public Cloud

Public clouds are owned and operated by third parties; they deliver superior economies of scale to customers, as the infrastructure costs are spread among a mix of users, giving each individual client an attractive low-cost, “Pay-as-you-go” model. All customers share the same infrastructure pool with limited configuration, security protections, and availability variances. These are managed and supported by the cloud provider. One of the advantages of a Public cloud is that they may be larger than an enterprises cloud, thus providing the ability to scale seamlessly, on demand.

Private Cloud

Private clouds are built exclusively for a single enterprise. They aim to address concerns on data security and offer greater control, which is typically lacking in a public cloud. There are two variations to a private cloud:

On-premise Private Cloud:

On-premise private clouds, also known as internal clouds are hosted within one‟s own data center. This model provides a more standardized process and protection, but is limited in aspects of size and scalability. IT departments would also need to incur the capital and operational costs for the physical resources. This is best suited for applications which require complete control and configurability of the infrastructure and security. –

Externally hosted Private Cloud:

This type of private cloud is hosted externally with a cloud provider, where the provider facilitates an exclusive cloud environment with full guarantee of privacy. This is best suited for enterprises that don’t prefer a public cloud due to sharing of physical resources.

Hybrid Cloud

Hybrid Clouds combine both public and private cloud models. With a Hybrid Cloud, service providers can utilize 3rd party Cloud Providers in a full or partial manner thus increasing the flexibility of computing. The Hybrid cloud environment is capable of providing on-demand, externally provisioned scale. The ability to augment a private cloud with the resources of a public cloud can be used to manage any unexpected surges in workload.

Cloud Computing Benefits

Enterprises would need to align their applications, so as to exploit the architecture models that Cloud Computing offers. Some of the typical benefits are listed below:

1. Reduced Cost

There are a number of reasons to attribute Cloud technology with lower costs. The billing model is pay as per usage; the infrastructure is not purchased thus lowering maintenance. Initial expense and recurring expenses are much lower than traditional computing.

2. Increased Storage

With the massive Infrastructure that is offered by Cloud providers today, storage & maintenance of large volumes of data is a reality. Sudden workload spikes are also managed effectively & efficiently, since the cloud can scale dynamically.

3. Flexibility

This is an extremely important characteristic. With enterprises having to adapt, even more rapidly, to changing business conditions, speed to deliver is critical. Cloud computing stresses on getting applications to market very quickly, by using the most appropriate building blocks necessary for deployment.

Cloud Computing Challenges:

Despite its growing influence, concerns regarding cloud computing still remain. In our opinion, the benefits outweigh the drawbacks and the model is worth exploring. Some common challenges are:

Data Protection

Data Security is a crucial element that warrants scrutiny. Enterprises are reluctant to buy an assurance of business data security from vendors. They fear losing data to competition and the data confidentiality of consumers. In many instances, the actual storage location is not disclosed, adding onto the security concerns of enterprises. In the existing models, firewalls across data centers

(owned by enterprises) protect this sensitive information. In the cloud model, Service providers are responsible for maintaining data security and enterprises would have to rely on them

Data Recovery and Availability

All business applications have Service level agreements that are stringently followed. Operational teams play a key role in management of service level agreements and runtime governance of applications. In production environments, operational teams support appropriate clustering and Fail over Data Replication System monitoring (Transactions monitoring, logs monitoring and others) Maintenance (Runtime Governance) Disaster recovery Capacity and performance management if, any of the above-mentioned services is under-served by a cloud provider, the damage & impact could be severe.

Management Capabilities

Despite there being multiple cloud providers, the management of platform and infrastructure is still in its infancy. Features like „Auto-scaling" for example, are a crucial requirement for many enterprises. There is huge potential to improve on the scalability and load balancing features provided today.

Regulatory and Compliance Restrictions

In some of the European countries, Government regulations do not allow customer's personal information and other sensitive information to be physically located outside the state or country. In order to meet such requirements, cloud providers need to setup a data center or a storage site exclusively within the country to comply with regulations. Having such an infrastructure may not always be feasible and is a big challenge for cloud providers.

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.

As users no longer physically possess the storage of their data, traditional cryptographic primitives for the purpose of data security protection cannot be directly adopted. In particular, simply downloading all the data for its integrity verification is not a practical solution due to the expensiveness in I/O and transmission cost across the network. Besides, it is often insufficient to detect the data corruption only when accessing the data, as it does not give users correctness assurance for those un accessed data and might be too late to recover the data loss or damage. Considering the large size of the outsourced data and the user's constrained resource capability, the tasks of auditing the data correctness in a cloud environment can be formidable and expensive for the cloud users. Moreover, the overhead of using cloud storage should be minimized as much as possible, such that a user does not need to perform too many operations to use the data (in additional to retrieving the data). In particular, users may not want to go through the complexity in verifying the data integrity. Besides, there may be more than one user accesses the same cloud storage, say in an enterprise setting. For easier management, it is desirable that cloud only entertains verification request from a single designated party.

DISADVANTAGES

- 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

- Unknown Risk Profile

. Security by obscurity may be low effort, but it can result in unknown exposures. It may also impair the in-depth analysis required highly controlled or regulated operational areas.

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.

REQUIREMENT & ANALYSIS

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.

HARDWARE AND SOFTWARE REQUIREMENTS

- Database : MySQL
- Operating System : Windows95/98/2000/XP
- Processor : Pentium 4 processor
- RAM : 1 GB RAM
- Hard Disk : 80 GB Hard Disk Space

Literature survey:

1)Cloud Data Auditing Techniques with a Focus on Privacy and Security:

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) A survey on auditing techniques used for preserving privacy of data stored on cloud:

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). TPA stores data information of the data owner and challenges to the cloud server, depending upon the data owner request. With the help of such mechanism, TPA can convince both, data owner and cloud server.

3)Auditing in Cloud Computing Solutions with OpenStack:

This presentation will walk through how auditing works in a Cloud environment. We will touch upon things like Cloud Auditing Data standard (CADF), the auditing challenges in a distributed cloud platform like OpenStack and how they are overcome using by CADF.

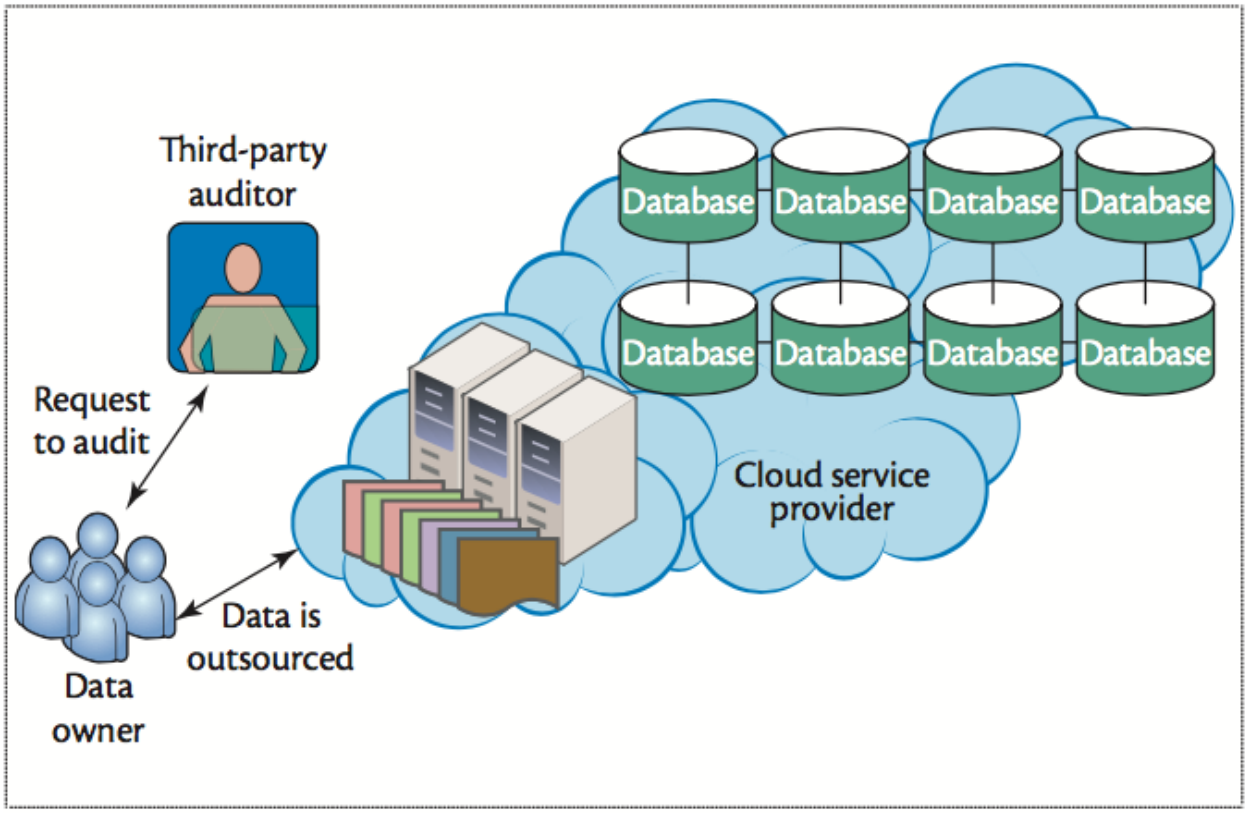
4)Cloud Security Auditing: Challenges and Emerging Approaches:

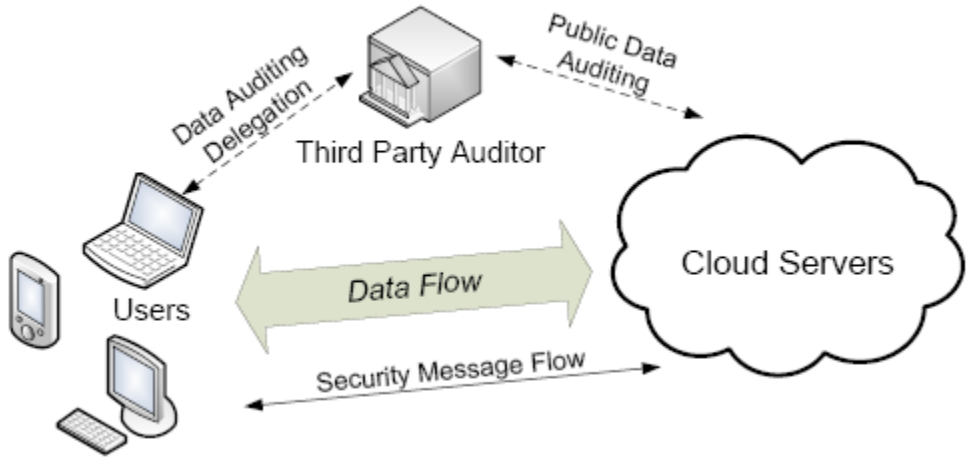
IT auditors collect information on an organization's information systems, practices, and operations and critically analyze the information for improvement. One of the primary goals of an IT audit is to determine if the information system and its maintainers are meeting both the legal expectations of 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.

5)Dynamic-Hash-Table Based Public Auditing for Secure Cloud Storage:

Cloud storage is an increasingly popular application of cloud computing, which can provide on-demand 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, and thereby significantly reduces the computational cost and communication overhead. Meanwhile, exploiting the structural advantages of the DHT, our scheme can also achieve higher updating efficiency than the state-of-the-art schemes. In addition, we extend our scheme to support privacy preservation by combining the homomorphic authenticator based on the public key with the random masking generated by the TPA, and achieve batch auditing by employing the aggregate BLS signature technique. We formally prove the security of the proposed scheme, and evaluate the auditing performance by detailed experiments and comparisons with the existing ones. The results demonstrate that the proposed scheme can effectively achieve secure auditing for cloud storage, and outperforms the previous schemes in computation complexity, storage costs and communication overhead.

ARCHITECTURE:





MODULES

The system is proposed to have the following modules:

- Admin Module
- TPA module
- User Module
- Block Verification Module
- Block Insertion Module
- Block Deletion

ADMIN MODULE

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

TPA MODULE

TPA 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

BLOCK VERIFICATION MODULE
User can check that the uploaded file is modified by any one or not (like server area)

BLOCK INSERTION MODULE
In the block insertion module user can insert the new block

BLOCK DELETION MODULE
In the Block Deletion Module user can delete the Block.

UMLS
4.2.1. 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.

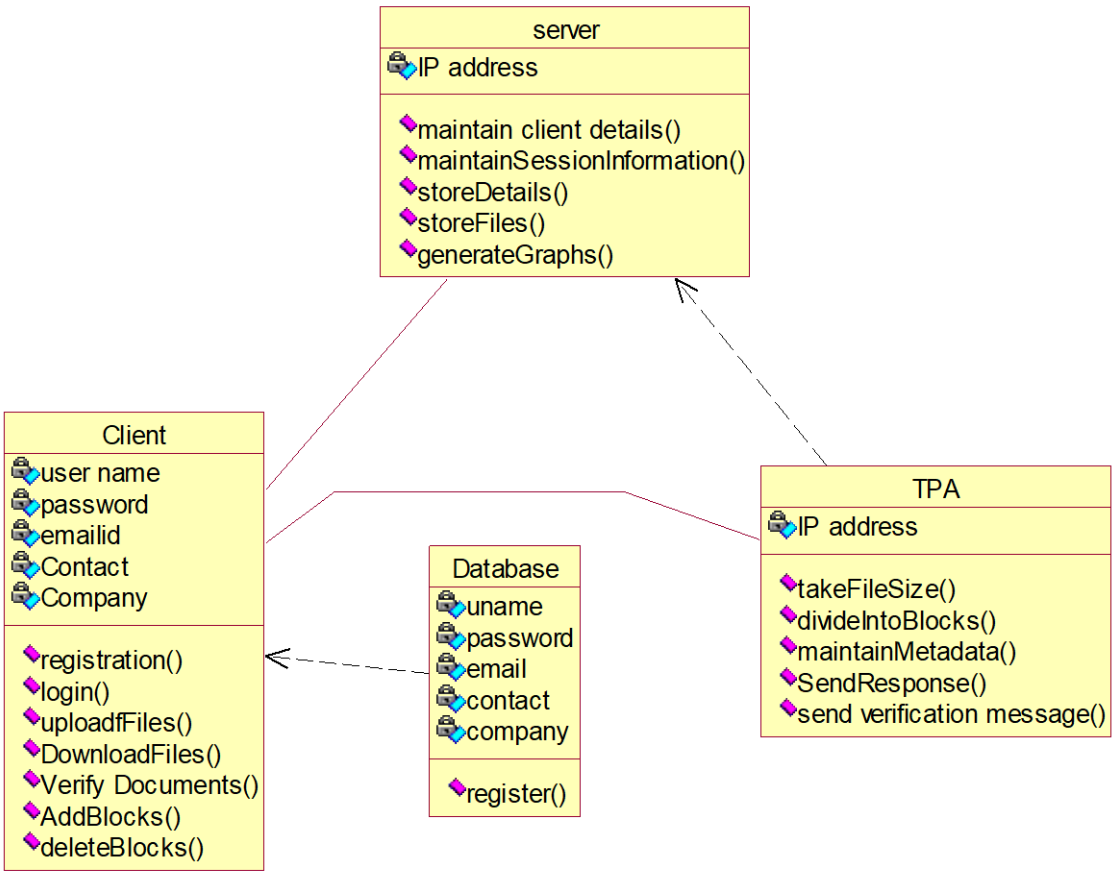
- User Model View
- This view represents the system from the user's perspective.
- The analysis representation describes a usage scenario from the end-users perspective.
- Structural model view
- In this model the data and functionality are arrived from inside the system.
- This model view models the 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 aspect 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.

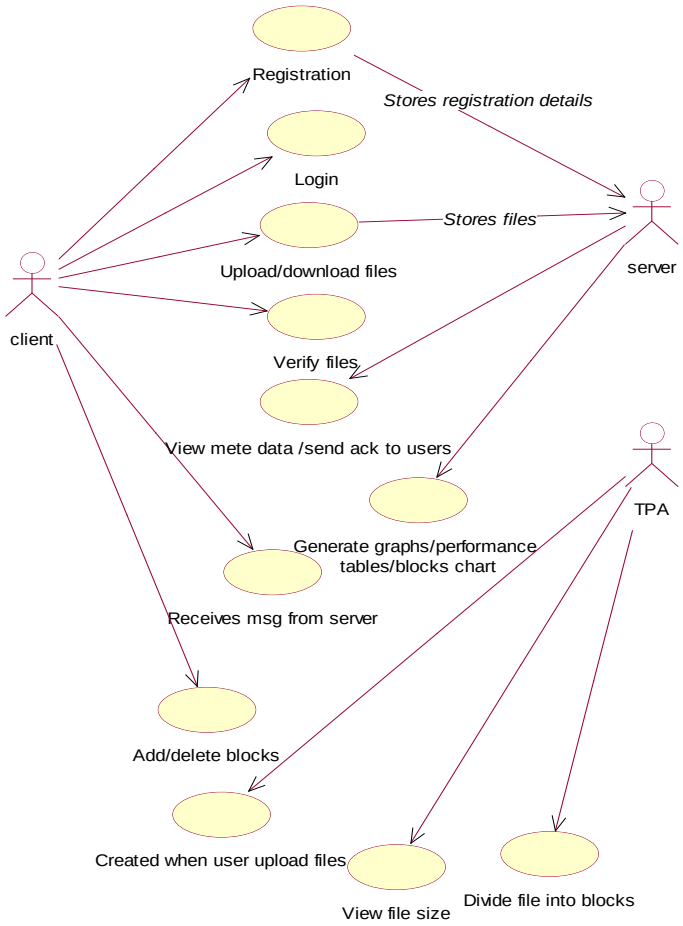
CLASS DIAGRAM AUDITING FOR SECURE DATA STORAGE IN CLOUD

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.



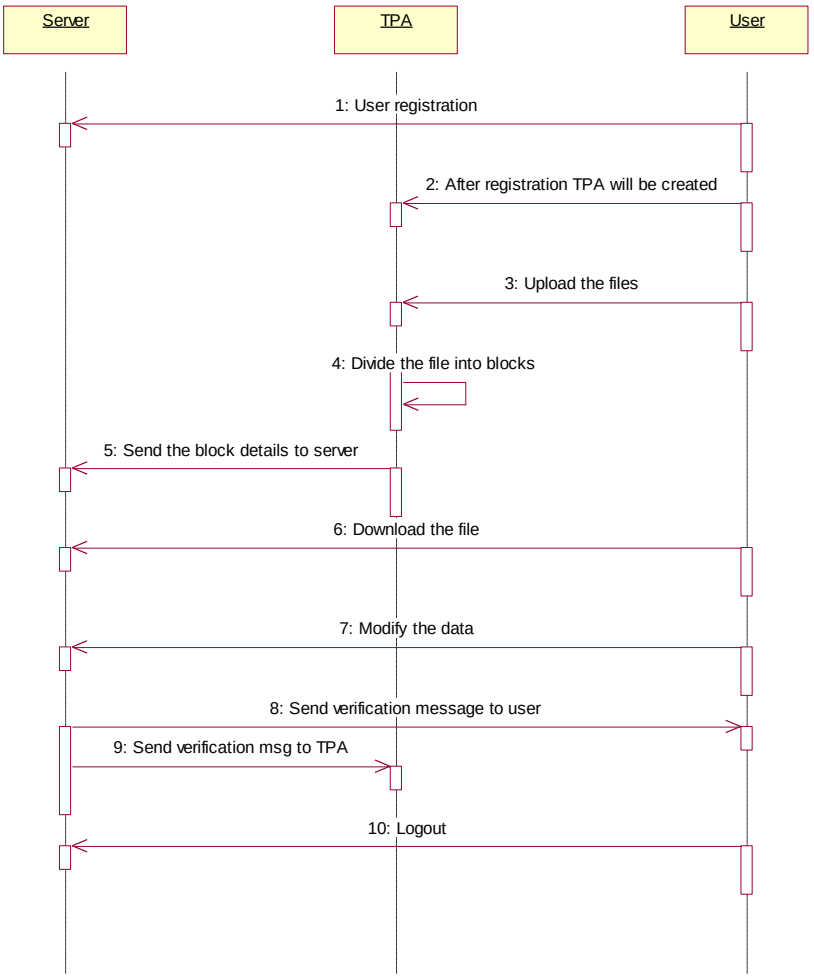
USECASE DIAGRAM:

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.



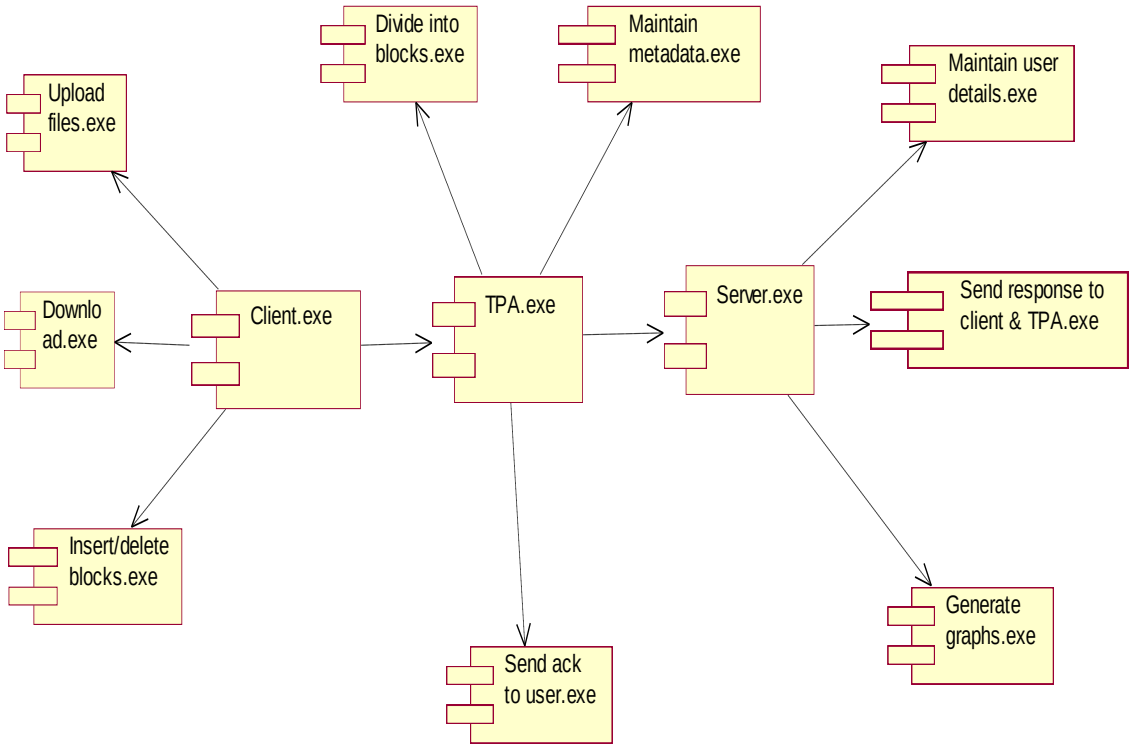
SEQUENCE DIAGRAM:

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.



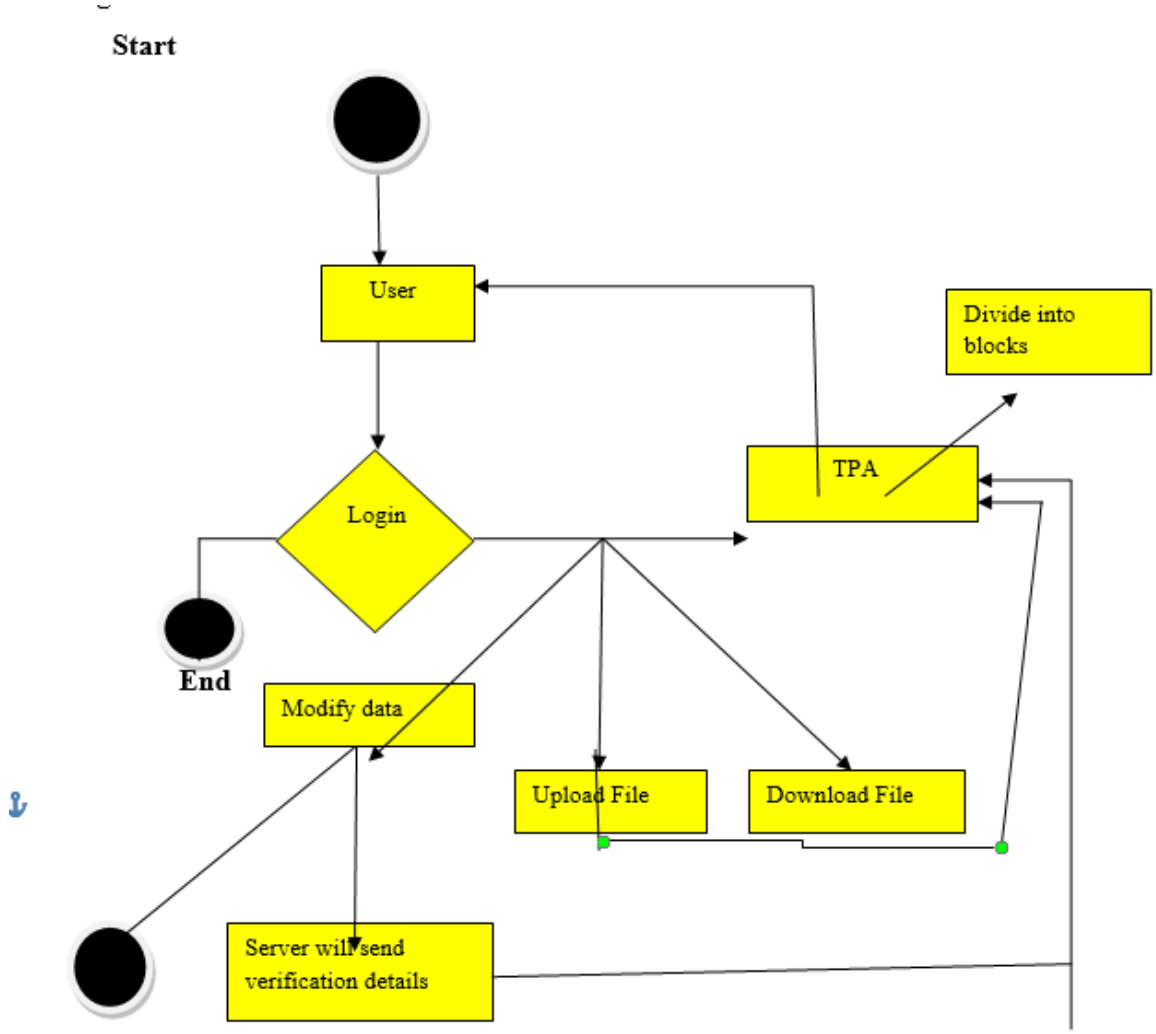
COMPONENT DIAGRAM:

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.



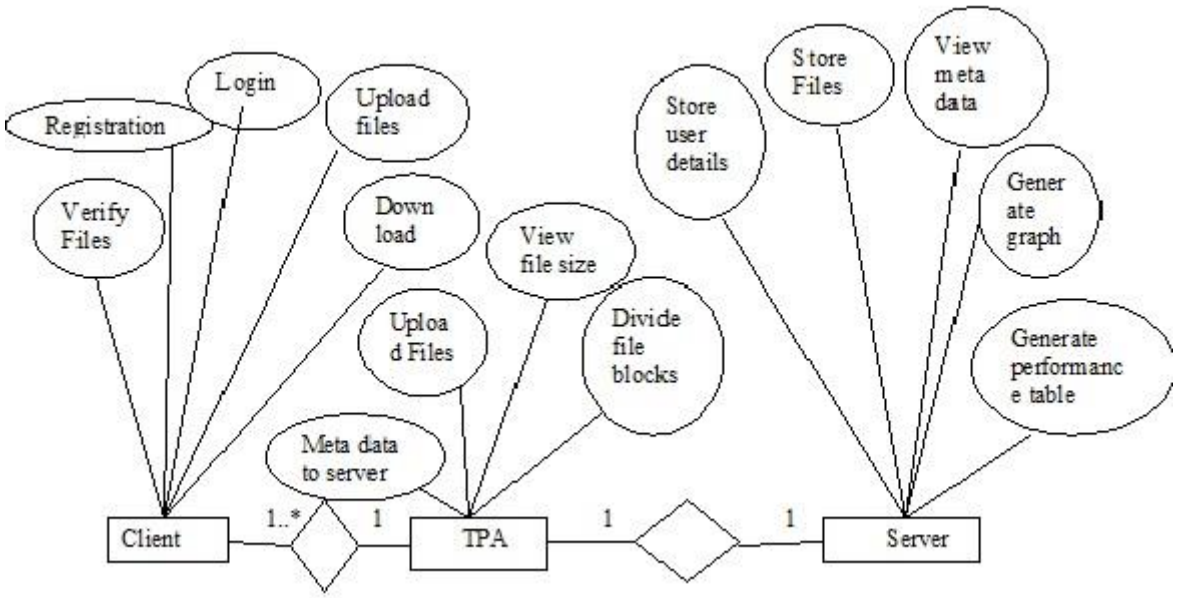
ACTIVITY DIAGRAM:

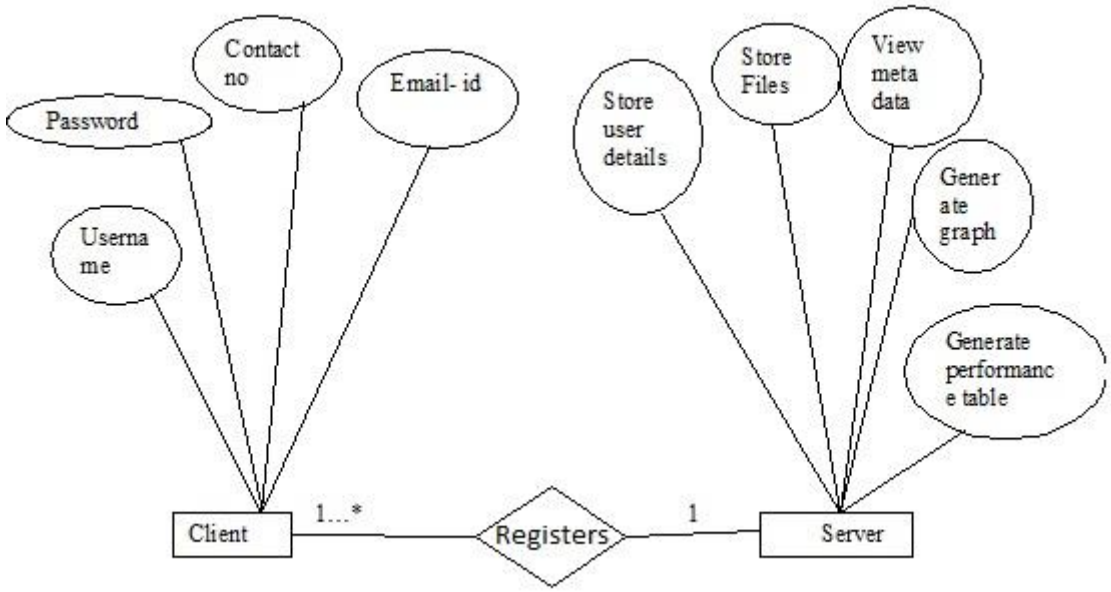
This activity diagram contains three activities that are Server, TPA, Client and. This diagram shows the flow of control between these activities.



ER-DIAGRAMS

This ER-Diagram contains three entities 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 relationship between these entities.





SOFTWARE ENVIRONMENT

INTRODUCTION

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 weared 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.

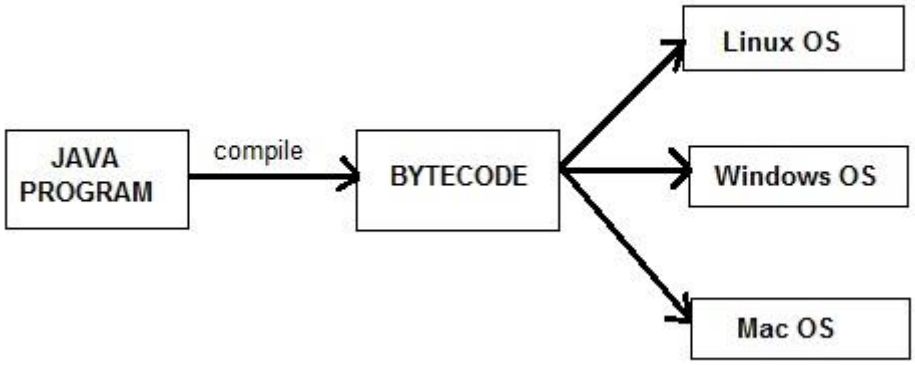
3) Robust

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

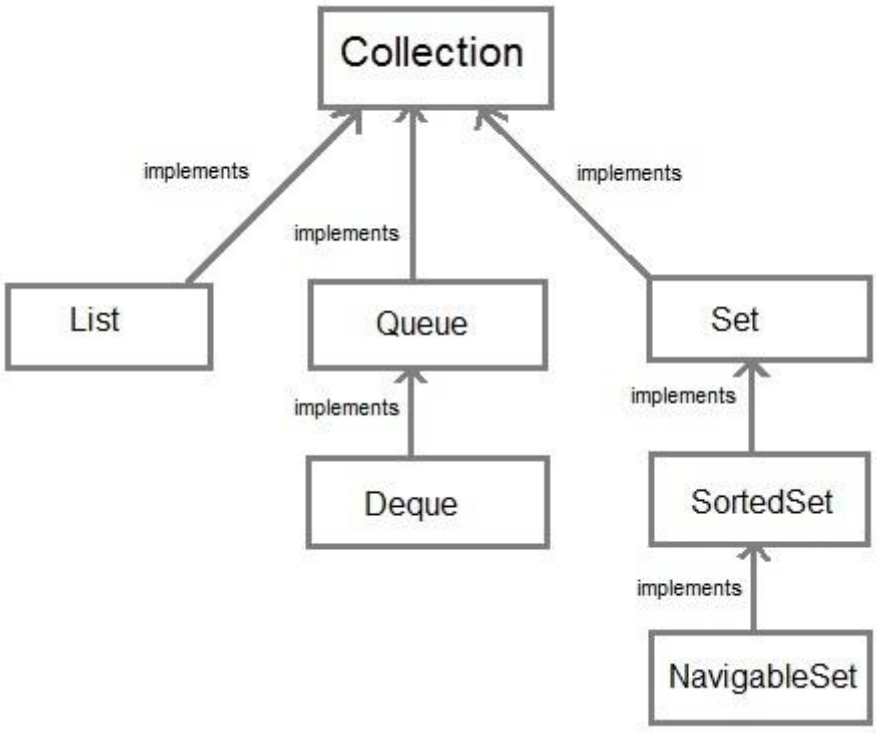
Java Byte code can be carried to any platform. No implementation dependent features. Everything related to storage is predefined, example: size of primitive data types

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.

COLLECTION FRAMEWORK

Collection framework was not part of original Java release. Collections was added to J2SE 1.2. Prior to Java 2, Java provided adhoc classes such as Dictionary, Vector, Stack and Properties to store and manipulate groups of objects. Collection framework provides many important classes and interfaces to collect and organize group of alike objects.



TESTING

TESTING PROCESS

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.

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.

Unit Testing

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.

Test Objectives

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.
4. Features to be tested
5. Verify that the entries are of the correct format
6. No duplicate entries should be allowed
7. 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.

ALPHA TESTING

In software development, alpha test will be a test among the teams to confirm that your product works. Originally, the term alpha test meant the first phase of testing in a software development process. The first phase includes unit testing, component testing, and system testing. It also enables us to test the product on the lowest common denominator machines to make sure download times are acceptable and preloads work.

BETA TESTING

In software development, a beta test is the second phase of software testing in which a sampling of the intended audience tries the product out. Beta testing can be considered "pre-release testing." Beta test versions of software are now distributed to curriculum specialists and teachers to give the program a & quot ; real-world & quota; test.

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 efficient.

REFERENCE:

[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.