

SECURE AUTHENTICATED KEY MANAGEMENT PROTOCOL FOR CLOUD COMPUTING

A Project Report Submitted to

Jawaharlal Nehru Technological University, Kakinada.

In partial fulfillment of the requirements for the award of the Degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

R. CHANDRA KOUSHIK (18NA1A0523)

Under the Esteemed Guidance of

Mr. B. ANIL KUMAR, M. Tech

Asst.Professor



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING.

LINGAYAS INSTITUTE OF MANAGEMENT AND TECHNOLOGY

Approved by AICTE, New Delhi, Recognized by Govt. of A.P., &
Affiliated to Jawaharlal Nehru Technological University, Kakinada.
Madalavarigudem, Nunna, Vijayawada-521 212. (Krishna District) [A.P.]
2021-2022

LINGAYAS INSTITUTE OF MANAGEMENT AND TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



CERTIFICATE

This is to certify that the project report entitled “**SECURE AUTHENTICATED KEY MANAGEMENT PROTOCOL FOR CLOUD COMPUTING**” was successfully completed by **R. CHANDRA KOUSHIK (18NA1A0523)** in partial fulfillment for the award of the degree of **Bachelor of Technology in Computer Science and Engineering** by **Jawaharlal Nehru Technological University Kakinada, Andhra Pradesh** during the academic year **2021-2022**

Project Guide

Mr. B. ANIL KUMAR

Head of The Department

Mr. B. ANIL KUMAR, M. Tech

External Examiner

DECLARATION

We Hereby declare that the project report titled **“SECURE AUTHENTICATED KEY MANAGEMENT PROTOCOL FOR CLOUD COMPUTING”** is a bonafide work carried out in Lingaya’s Institute of Management and Technology, Madalavarigudem, Vijayawada, during the academic year 2021-2022 in partial fulfillment of the requirement for the award of degree of **Bachelor of Technology** in **Computer Science and Engineering**.

Project Associate

R. CHANDRA KOUSHIK (18NA1A0523)

ACKNOWLEDGEMENT

We would like to express our gratitude to our beloved Chairman **Mr. GADDE RAJLING** for his great motivation and encouragement towards project work.

We also extend our sincere gratitude to our Principal **Dr.Y. SUDHEER BABU** for his encouragement and facilities provided during the course of the project. We would like to express our heart full gratitude to our lab technicians, who helped us in all aspects of lab work.

We wish to express our sincere gratitude to **Mr. B. ANIL KUMAR, Head of the Department, Computer Science and Engineering** for supporting us in the completion of this project.

We would like to articulate our profound gratitude and indebtedness to our project guide **Mr. B. ANIL KUMAR**, Asst. Professor, Computer Science and Engineering, who has been constant motivation and guiding factor throughout the project time. It has been a great pleasure for us to get an opportunity to work under his guidance and complete the project work successfully.

We would like to thank the faculty of department of **Computer Science and Engineering**, Lingaya's Institute of Management and Technology for consistently supporting our work throughout the project period with all patience and for giving valuable inputs where ever required for betterment and result oriented work.

Project Associate

R. CHANDRA KOUSHIK (18NA1A0523)

ABSTRACT

With the maturity of cloud computing technology in terms of reliability and efficiency, a large number of services have migrated to the cloud platform. To convenient access to the services and protect the privacy of communication in the public network, three-factor Mutual Authentication and Key Agreement protocols for multi-server architectures gain wide attention. However, most of the existing three-factor MAKKA protocols don't provide a formal security proof resulting in various attacks on the related protocols, or they have high computation and communication costs. And most of the three-factor MAKKA protocols haven't a dynamic revocation mechanism, which leads to malicious users cannot be promptly revoked.

INDEX

S.NO	CONTENT	PAGE NO
CHAPTER-1	INTRODUCTION	1-6
	1.1 Introduction	2-5
	1.2 Problem Statement	6
	1.3 Objective of the project	6
	1.4 Scope of project	6
CHAPTER-2	LITERATURE SURVEY	7-10
CHAPTER-3	SYSTEM ANALYSIS	11-16
	3.1 Existing System	12
	3.1.1 Disadvantages of Existing System	12
	3.2 Proposed System	12
	3.2.1 Advantages of Proposed System	13
	3.3 Software Requirement Specification	13-14
	3.3.1 SRS Introduction	13-14
	3.4 Software Requirements	14
	3.5 Hardware Requirements	14
	3.6 Algorithms	15-16
CHAPTER-4	SYSTEM DESIGN	17 – 27
	4.1 Input Design	18
	4.2 Output Design	19
	4.3 Flow Chart	20

	4.4 System Architecture	21
	4.5 Data Flow Diagram	22-23
	4.6 UML diagrams	24-27
CHAPTER-5	SOFTWARE ENVIRONMENT	28 - 30
CHAPTER-6	IMPLEMENTATION AND RESULTS	31 – 49
	6.1 Output Screens	31 –41
	6.2 Coding	42 – 49
CHAPTER- 7	SYSTEM TESTING	50 – 55
	7.1 Introduction	51
	7.2 System Testing	51 - 55
CHAPTER-8	CONCLUSION	56 – 57
CHAPTER-9	REFERENCES	58 – 59
CHAPTER-10	VISIBLE PROJECT WORK OUTPUT	60 – 64
	10.1 Domain Technology	61 – 64

LIST OF DIAGRAMS

S.NO	DIAGRAMS	PAGE
1.	1.1 Cloud Services	2
2.	1.2 Cloud Computing Services	4
3.	1.3 Cloud platforms	5
4.	3.6 AES design	15
5.	4.3 Flow chart diagram	20
6.	4.4 Architecture of the cloud system	21
7.	4.5.1 Data Flow Diagram at level -0	22
8.	4.5.2 Data Flow Diagram at level -1	23
9.	4.5.3 Data Flow Diagram at level -2	23
10.	4.6.1 Use Case Diagram	25
11.	4.6.2 Class Diagram	26
12.	4.6.3 Sequence Diagram	27

LIST OF SCREENSHOTS

S.NO	SCREENSHOTS	PAGE
1.	5.1 Tomcat Server	30
2.	6.1.1 Home page	32
3.	6.1.2 User Login Page	33
4.	6.1.3 User Verification Page	34
5.	6.1.4 User File Upload Page	35
6.	6.1.5 User File Verification page	36
7.	6.1.6 Registration Page	37
8.	6.1.7 Cloud Login Page	38
9.	6.1.8 Cloud Files Page	39
10.	6.1.9 Trust Manager Login Page	40
11.	6.1.10 List of Users in Cloud Services Page	41
12.	7.1.1 User login result	55
13.	7.1.2 Data Base result	55
14.	7.2.3 Data Base Test Cases	55
15.	7.2.4 User Login Test Cases	55

CHAPTER – 1

1. INTRODUCTION

Cloud computing:

Cloud computing transforms IT infrastructure into a utility: It lets you ‘plug into’ infrastructure via the internet, and use computing resources without installing and maintaining them on-premises.

What is Cloud Computing?

Cloud computing is on-demand access, via the internet, to computing resources applications, servers, data storage, development tools, networking capabilities, and more hosted at a remote data center managed by a cloud services provider. The CSP makes these resources available for a monthly subscription fee or bills them according to usage.

Compared to traditional on-premises IT, and depending on the cloud services you select, cloud computing helps do the following:

Lower IT costs:

Cloud lets you offload some or most of the costs and effort of purchasing, installing, configuring, and managing your own on-premises infrastructure.

Improve agility and time-to-value:

With cloud, your organization can start using enterprise applications in minutes, instead of waiting weeks or months for IT to respond to a request, purchase and configure supporting hardware, and install software. Cloud also lets you empower certain users specifically developers and data scientists to help themselves to software and support infrastructure.

Scale more easily and cost-effectively:

Cloud provides elasticity instead of purchasing excess capacity that sits unused during slow periods, you can scale capacity up and down in response to spikes and dips in traffic. You can also take advantage of your cloud provider’s global network to spread your applications closer to users around the world.

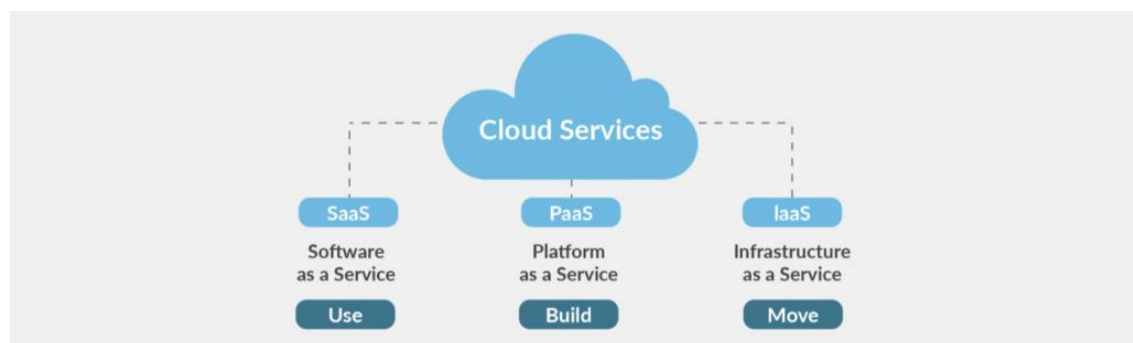


Fig: 1.1 Cloud Services

Cloud computing services:

IaaS (Infrastructure as a Service), PaaS (Platform as a Service), and SaaS (Software as a Service) are the three most common models of cloud services, and it's not uncommon for an organization to use all three. However, there is often confusion among the three and what's included with each:

SaaS (Software as a Service):

SaaS also known as cloud-based software or cloud applications is application software that's hosted in the cloud and that you access and use via a web browser, a dedicated desktop client, or an API that integrates with your desktop or mobile operating system. In most cases, SaaS users pay a monthly or annual subscription fee; some may offer 'pay as you go' pricing based on your actual usage.

❖ In addition to the cost savings, time-to-value, and scalability benefits of cloud, SaaS offers the following:

Automatic upgrades:

With SaaS, you take advantage of new features as soon as the provider adds them, without having to orchestrate an on-premises upgrade.

Protection from data loss:

Because your application data is in the cloud, with the application, you don't lose data if your device crashes or breaks. SaaS is the primary delivery model for most commercial software today there are hundreds of thousands of SaaS solutions available, from the most focused industry and departmental applications, to powerful enterprise software database and AI (artificial intelligence) software.

PaaS (Platform as a Service):

PaaS provides software developers with on-demand platform hardware, complete software stack, infrastructure, and even development tools for running, developing, and managing applications without the cost, complexity, and inflexibility of maintaining that platform on-premises. With PaaS, the cloud provider hosts everything servers, networks, storage, operating system software, middleware, databases at their data center. Developers simply pick from a menu to 'spin up' servers and environments they need to run, build, test, deploy, maintain, update, and scale applications. Today, PaaS is often built around containers, a virtualized compute model one step removed from virtual servers. Containers virtualize the operating system, enabling developers to package the application with only the operating system services it needs to run on any platform, without modification and without need for middleware.

IaaS (Infrastructure as a Service):

IaaS provides on demand access to fundamental computing resources physical and virtual servers, networking, and storage over the internet on a pay as you go basis. IaaS enables end users to scale and shrink resources on an as-needed basis, reducing the need for high, up-front capital expenditures or unnecessary on-premises or ‘owned’ infrastructure and for overbuying resources to accommodate periodic spikes in usage. IaaS was the most popular cloud computing model when it emerged in the early 2010s. While it remains the cloud model for many types of workloads, use of SaaS and PaaS is growing at a much faster rate.



Fig:1.2 Cloud Computing Services

Cloud security

Traditionally, security concerns have been the primary obstacle for organizations considering cloud services, particularly public cloud services. In response to demand, however, the security offered by cloud service providers is steadily outstripping on-premises security solutions. Nevertheless, maintaining cloud security demands different procedures and employee skillsets than in legacy IT environments. Some cloud security best practices include the following:

Shared responsibility for security:

Generally, the cloud provider is responsible for securing cloud infrastructure and the customer is responsible for protecting its data within the cloud but it's also important to clearly define data ownership between private and public third parties.

Data encryption:

Data should be encrypted while at rest, in transit, and in use. Customers need to maintain full control over security keys and hardware security module.

User identity and access management:

Customer and IT teams need full understanding of and visibility into network, device, application, and data access.

Cloud use cases

Disaster recovery and business continuity have always been a natural for cloud because cloud provides cost-effective redundancy to protect data against system failures and the physical distance required to recover data and applications in the event of a local outage or disaster. All of the major public cloud providers offer Disaster-Recovery-as-a-Service (DraaS).

Anything that involves storing and processing huge volumes of data at high speeds and requires more storage and computing capacity than most organizations can or want to purchase and deploy on-premises is a target for cloud computing. Examples include:

- ❖ Big data analytics
- ❖ Internet of Things (IoT)

Artificial intelligence particularly machine learning and deep learning applications. For development teams adopting Agile or DevOps to streamline development, cloud offers the on-demand end-user self-service that keeps operations tasks such as spinning up development and test servers from becoming development bottleneck.



Fig: 1.3 Cloud Platforms

1.1 PROBLEM DEFINITION

Problem: Analyzing the Cloud Data that has been stored in the cloud server. There will be a high risk in two-factor protocols, namely identity, password, and three-factor protocols, namely identity, password, biometrics.

1.2 OBJECTIVES OF THE PROJECT

- ❖ The results produced by this protocol can help in secure the data in cloud.
- ❖ This protocol uses an AES algorithm for providing better encryption to the cloud data in the server.
- ❖ This system gives us the complete representation of cloud data, which is easy to understand and can help in security.

1.3 SCOPE OF THE PROJECT

In the future assessment, we can add a wide variety of techniques which will help in better security of cloud data in any server. Further we can make many developments to the project for much secure encryption and make the data more secure from further attacks which will help in resolve problems of cloud servers. In effective way we can make the user add some tasks or he can share the data more redundantly between the users which will help the trust manager a little bit. Trust manager plays a key role in this project that he keeps a look of the things in data base as well as the users and data servers.

CHAPTER – 2

2. LITERATURE SURVEY

- 1) Opinion S. Sood, J. Antin, and E. Churchill, “Profanity use in online communities,” in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, 2012, pp. 1481–1490.

As user-generated Web content increases, the amount of inappropriate and/or objectionable content also grows. Several scholarly communities are addressing how to detect and manage such content: research in computer vision focuses on detection of inappropriate images, natural language processing technology has advanced to recognize insults. However, profanity detection systems remain flawed. Current list-based profanity detection systems have two limitations. First, they are easy to circumvent and easily become stale - that is, they cannot adapt to misspellings, abbreviations, and the fast pace of profane slang evolution. Secondly, they offer a one-size fits all solution; they typically do not accommodate domain, community and context specific needs. However, social settings have their own normative behaviors - what is deemed acceptable in one community may not be in another. In this paper, through analysis of comments from a social news site, we provide evidence that current systems are performing poorly and evaluate the cases on which they fail. We then address community differences regarding creation/tolerance of profanity and suggest a shift to more contextually nuanced profanity detection systems.

- 2) S. Rojas-Galeano, “On obstructing obscenity obfuscation,” ACM Transactions on the Web (TWEB), vol. 11, no. 2, p. 12, 2017.

Obscenity (the use of rude words or offensive expressions) has spread from informal verbal conversations to digital media, becoming increasingly common on user-generated comments found in Web forums, newspaper user boards, social networks, blogs, and media-sharing sites. The basic obscenity-blocking mechanism is based on verbatim comparisons against a blacklist of banned vocabulary; however, creative users circumvent these filters by obfuscating obscenity with symbol substitutions or bogus segmentations that still visually preserve the original semantics. The number of potential obfuscated variants is combinatorial, yielding the verbatim filter impractical. Here we describe a method intended to obstruct this anomaly inspired by sequence alignment algorithms used in genomics, coupled with a tailor-made edit penalty function. The method only requires to set up the

vocabulary of plain obscenities; no further training is needed. Its complexity on screening a single obscenity is linear, both in runtime and memory, on the length of the user-generated text. We validated the method on three different experiments. The first one involves a new dataset that is also introduced in this article; it consists of a set of manually annotated real-life comments in Spanish, gathered from the news user boards of an online newspaper, containing this type of obfuscation. The second one is a publicly available dataset of comments in Portuguese from a sports Web site. In these experiments, at the obscenity level, we observed recall rates greater than 90%, whereas precision rates varied between 75% and 95%, depending on their sequence length (shorter lengths yielded a higher number of false alarms). On the other hand, at the comment level, we report recall of 86%, precision of 91%, and specificity of 98%. The last experiment revealed that the method is more effective in matching this type of obfuscation compared to the classical Levenshtein edit distance. We conclude discussing the prospects of the method to help enforcing moderation rules of obscenity expressions or as a preprocessing mechanism for sequence cleaning and/or feature extraction in more sophisticated text categorization techniques.

3) Evaluating E. Wulczyn, N. Thain, and L. Dixon, “Ex machina: Personal attacks seen at scale,” in Proceedings of the 26th International Conference on World Wide Web. International World Wide Web Conferences Steering Committee, 2017, pp. 1391–1399.

The damage personal attacks cause to online discourse motivates many platforms to try to curb the phenomenon. However, understanding the prevalence and impact of personal attacks in online platforms at scale remains surprisingly difficult. The contribution of this paper is to develop and illustrate a method that combines crowdsourcing and machine learning to analyze personal attacks at scale. We show an evaluation method for a classifier in terms of the aggregated number of crowd-workers it can approximate. We apply our methodology to English Wikipedia, generating a corpus of over 100k high quality human-labeled comments and 63M machine-labeled ones from a classifier that is as good as the aggregate of 3 crowd-workers, as measured by the area under the ROC curve and Spearman correlation. Using this corpus of machine-labeled scores, our methodology allows us to explore some of the open questions about the nature of online personal attacks.

This reveals that the majority of personal attacks on Wikipedia are not the result of a few

malicious users, nor primarily the consequence of allowing anonymous contributions from unregistered users.

4) Mining A. Schmidt and M. Wiegand, “A survey on hate speech detection using natural language processing,” in Proceedings of the Fifth International Workshop on Natural Language Processing for social media. Association for Computational Linguistics, Valencia, Spain, 2017, pp. 1–10.

This paper presents a survey on hate speech detection. Given the steadily growing body of social media content, the amount of online hate speech is also increasing. Due to the massive scale of the web, methods that automatically detect hate speech are required. Our survey describes key areas that have been explored to automatically recognize these types of utterances using natural language processing. We also discuss limits of those approaches.

CHAPTER - 3

3.SYSTEM ANALYSIS

3.1 EXISTING SYSTEM:

- ❖ Earlier MAKKA protocols are designed for single-server architecture. As Internet users grow exponentially, the number of cloud servers rendering different services has also grown significantly.
- ❖ For the single-server architecture, it is difficult for users to maintain a variety of passwords for each server. To improve user experience, many scholars propose more flexible MAKKA protocols for multi-server environments. Combined with the unified management features of the cloud platform, such protocols can be conveniently applied. users and cloud servers only need to register in the registration center (RC) to mutual authentication and key agreement.

3.1.1 DISADVANTAGES OF EXISTING SYSTEM:

- ❖ In the multi-server environments, the MAKKA protocols can be further divided into two categories, two-factor MAKKA protocols, namely identity, password, and three-factor MAKKA protocols, namely identity, password, biometrics.
- ❖ The works in have shown that the password- based MAKKA protocols suffer from several attacks such as guessing password attack.

3.2 PROPOSED SYSTEM:

We propose a dynamic revocable three-factor mutual authentication and key agreement (3DRMAKA) protocol which has more comprehensive functions, reliable security and relatively higher execution efficiency. Our contribution can be summarized as follows:

- ❖ Our scheme achieves the user's dynamic management. In our protocol, users can be dynamically revoked to promptly prevent attacks from malicious users. Without a dynamic revocation mechanism, RC can't punish malicious users in a timely manner. This may result in such malicious users still active in the network to communicate with other servers.
- ❖ In the random oracle, we provide a formal proof of the proposed protocol based on BDH, CDH and Schnorr signatures unforgeability assumptions. We show that the proposed protocol is mutual authentication secure and authenticated key agreement secure.

- ❖ Our protocol has a good execution efficiency. Especially on the client side, the computation cost of our scheme is the lowest in the related existing protocols. This shows that our protocol is more suitable for device mobiles with limited computing resource. And, to prove that the protocol is technically sound, we programmatically simulate the proposed protocol.

3.2.1 ADVANTAGES OF PROPOSED SYSTEM:

- ❖ Proposed protocol can meet the demands of multi-server architectures such as anonymity, non-traceability, resistance password guessing attack and smart card extraction attack.
- ❖ On the other hand, the MAKKA protocol is also widely used in other environments, such as Passive Internet of Things.

3.3 SOFTWARE REQUIREMENT SPECIFICATION

3.3.1 INTRODUCTION TO SOFTWARE REQUIREMENTS SPECIFICATION:

Once the problem is analyzed and the essentials understood, the requirements must be specified in the requirement specification document. For requirement specification in the form of document, some specification language has to be selected. The requirements documents must specify all functional and performance requirements, the formats of inputs, outputs and any required standards, and all design constraints that exists due to political, economic environmental, and security reasons. The phase ends with validation of requirements specified in the document. The basic purpose of validation is to make sure that the requirements specified in the document, actually reflect the actual requirements or needs, and that all requirements are specified. Validation is often done through requirement review, in which a group of people including representatives of the client, critically review the requirements specification.

Software Requirement or Role of SRS

- ❖ A condition of capability needed by a user to solve a problem or achieve an objective.
- ❖ A condition or capability that must be met or possessed by a system to satisfy a contract, standard, specification, or other formally imposed document.

Software requirements are dealing with the requirements of the proposed system, that is, the capabilities that system, which is yet to be developed, should have. It is because dealing with specifying a system that does not exist in any form that the problem of requirements becomes complicated. The basic goal of the requirement phase is to produce the SRS, which describes the complete external behavior of the proposed software.

3.4 SOFTWARE REQUIREMENTS:

Operating system	:	Windows XP Service Pack 3 (SP3).
Coding Language	:	Java/J2EE (JSP, Servlet).
Database	:	HeidiSQL 9.4.0.5125.
Front End	:	J2EE 8.0.

3.5 HARDWARE REQUIREMENTS:

Processor	:	Intel Core i3 with 2.4 GHz.
Hard Disk	:	20 GB.
RAM	:	4 GB.

3.6 ALGORITHMS

ADVANCED ENCRYPTION STANDARD ALGORITHM:

AES includes three block ciphers:

- ❖ AES-128 uses a 128-bit key length to encrypt and decrypt a block of messages.
- ❖ AES-192 uses a 192-bit key length to encrypt and decrypt a block of messages.
- ❖ AES-256 uses a 256-bit key length to encrypt and decrypt a block of messages.

Each cipher encrypts and decrypts data in blocks of 128 bits using cryptographic keys of 128, 192 and 256 bits, respectively. Symmetric, also known as secret key, ciphers use the same key for encrypting and decrypting. The sender and the receiver must both know and use the same secret key.

The government classifies information in three categories: Confidential, Secret or Top Secret. All key lengths can be used to protect the Confidential and Secret level. Top Secret information requires either 192- or 256-bit key lengths. There are 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys. A round consists of several processing steps that include substitution, transposition and mixing of the input plaintext to transform it into the final output of ciphertext.

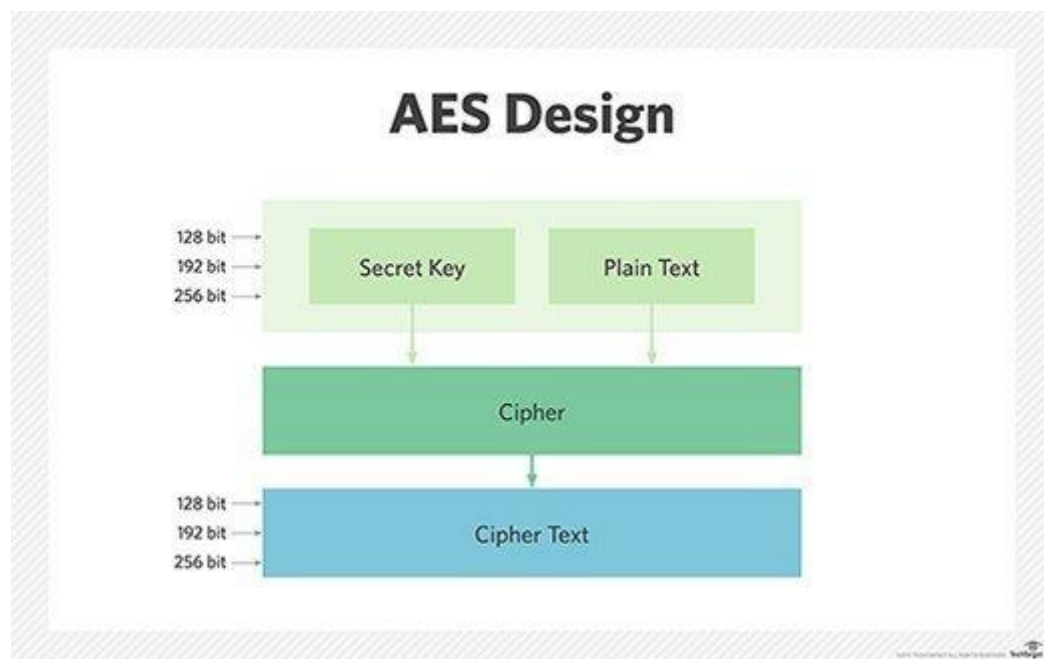


Fig- 3.6 AES Design

AES uses 128-, 192- or 256-bit keys to encrypt and decrypt data.

The AES encryption algorithm defines numerous transformations that are to be performed on data stored in an array. The first step of the cipher is to put the data into an array, after which the cipher transformations are repeated over multiple encryption rounds.

The first transformation in the AES encryption cipher is substitution of data using a substitution table. The second transformation shifts data rows. The third mixes columns. The last transformation is performed on each column using a different part of the encryption key. Longer keys need more rounds to complete.

CHAPTER- 4

4. SYSTEM DESIGN

4.1 INPUT DESIGN

Input Design plays a vital role in the life cycle of software development, it requires very careful attention of developers. The input design is to feed data to the application as accurate as possible. So, inputs are supposed to be designed effectively so that the errors occurring while feeding are minimized. According to Software Engineering Concepts, the input forms or screens are designed to provide to have a validation control over the input limit, range and other related validations.

This system has input screens in almost all the modules. Error messages are developed to alert the user whenever he commits some mistakes and guides him in the right way so that invalid entries are not made. Let us see deeply about this under module design.

Input design is the process of converting the user created input into a computer-based format. The goal of the input design is to make the data entry logical and free from errors. The error in the input are controlled by the input design. The application has been developed in user-friendly manner. The forms have been designed in such a way during the processing the cursor is placed in the position where must be entered. The user is also provided with in an option to select an appropriate input from various alternatives related to the field in certain cases.

Validations are required for each data entered. Whenever a user enters an erroneous data, error message is displayed and the user can move on to the subsequent pages after completing all the entries in the current page.

OBJECTIVES

- ❖ Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

4.2 OUTPUT DESIGN

The Output from the computer is required to mainly create an efficient method of communication within the company primarily among the project leader and his team members, in other words, the administrator and the clients. The output of VPN is the system which allows the project leader to manage his clients in terms of creating new clients and assigning new projects to them, maintaining a record of the project validity and providing folder level access to each client on the user side depending on the projects allotted to him. After completion of a project, a new project may be assigned to the client. User authentication procedures are maintained at the initial stages itself. A new user may be created by the administrator himself or a user can himself register as a new user but the task of assigning projects and validating a new user rest with the administrator only.

The application starts running when it is executed for the first time. The server has to be started and then the internet explorer is used as the browser. The project will run on the local area network so the server machine will serve as the administrator while the other connected systems can act as the clients. The developed system is highly user friendly and can be easily understood by anyone using it even for the first time.

4.3 FLOW CHART DIAGRAM:

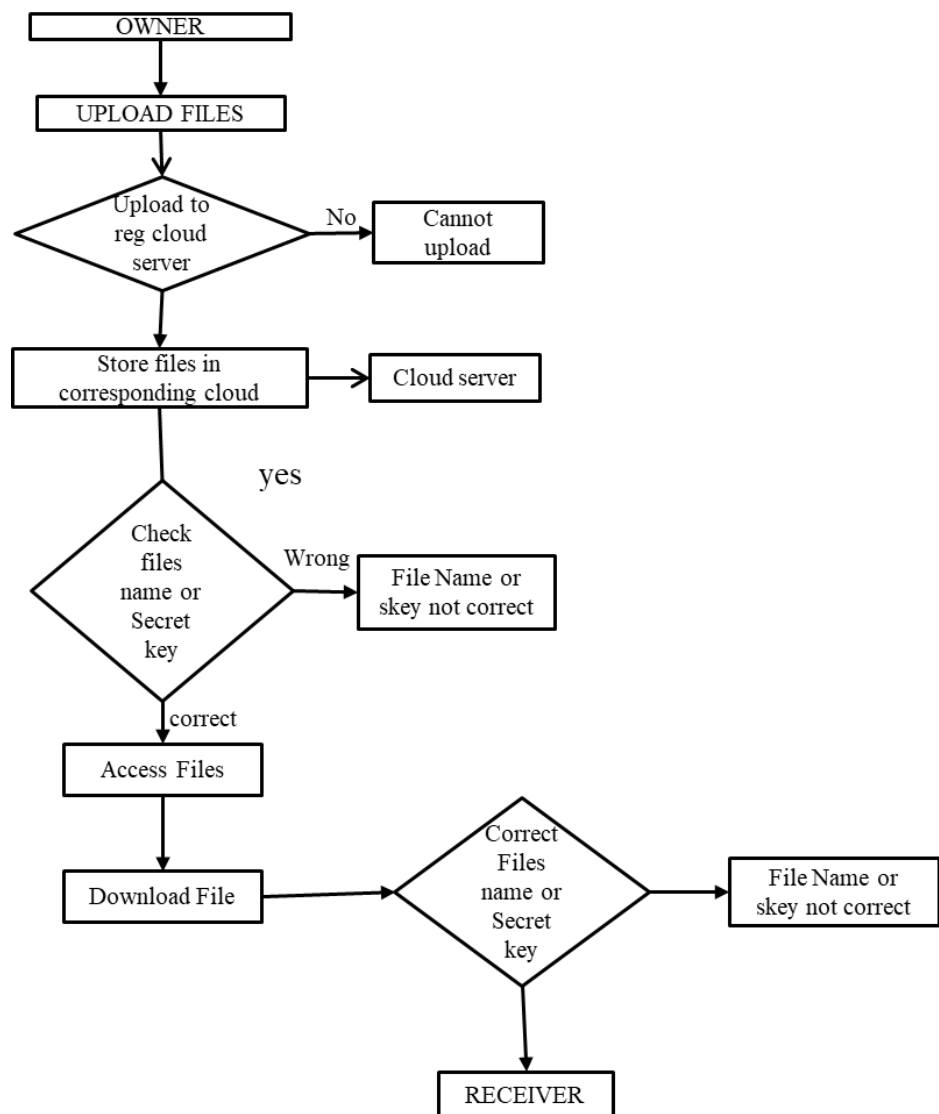


Fig- 4.3 Flow chart diagram

4.4 SYSTEM ARCHITECTURE

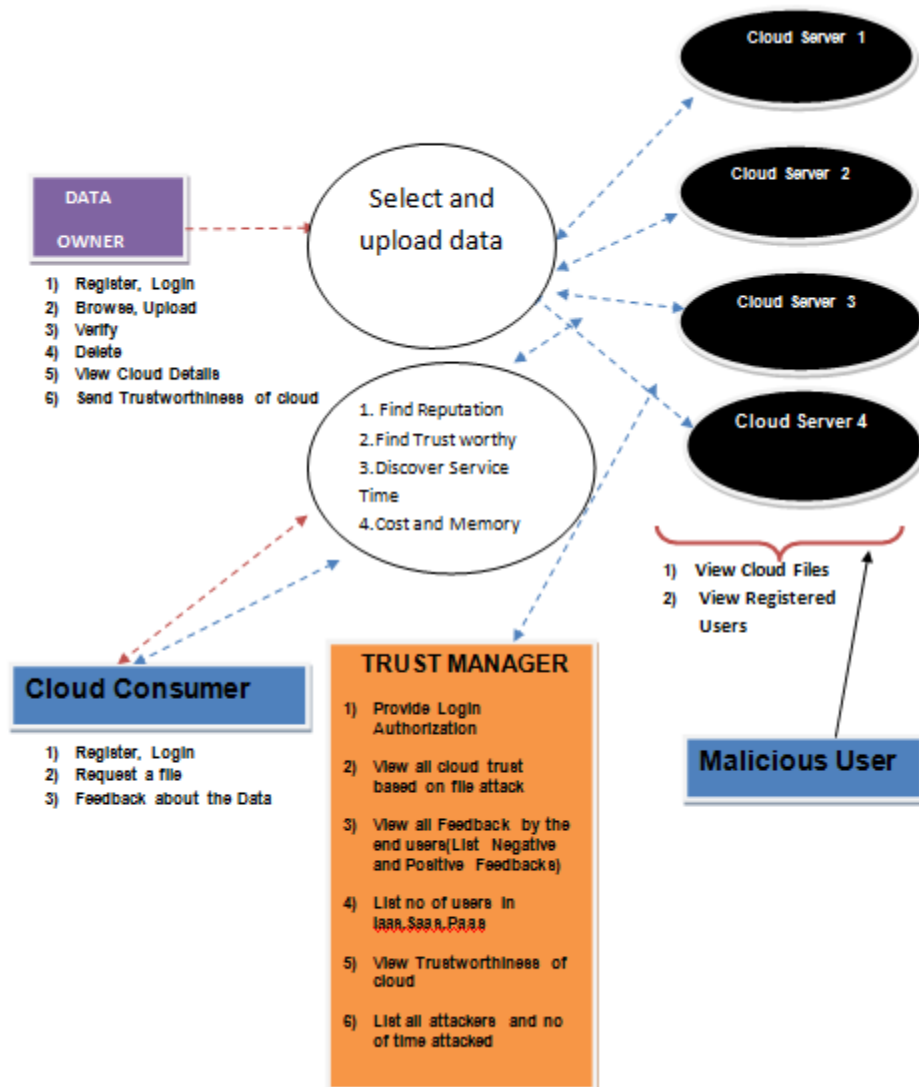


Fig 4.4: Architecture of the cloud system

4.5 DATA FLOW DIAGRAM:

- ❖ The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
- ❖ The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
- ❖ DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
- ❖ DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

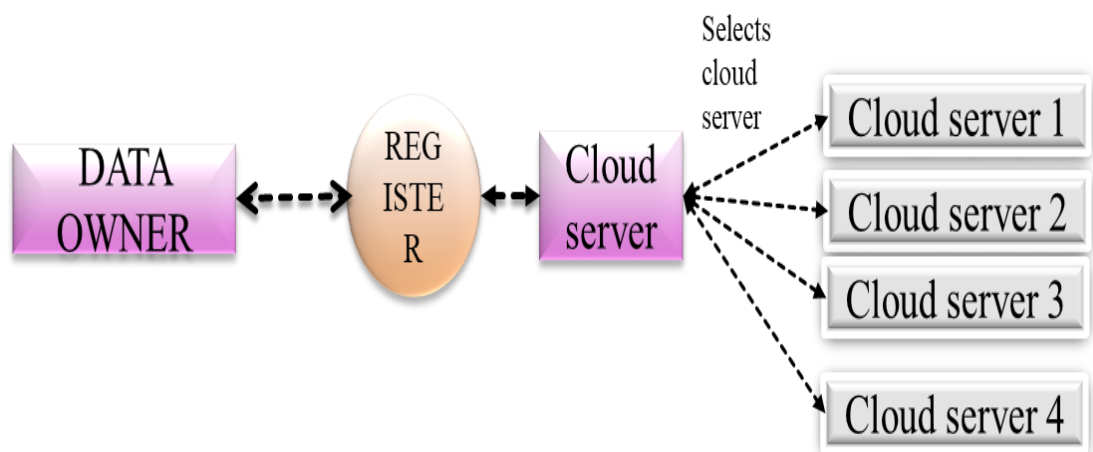


Fig- 4.5.1 Data Flow Diagram at level -0

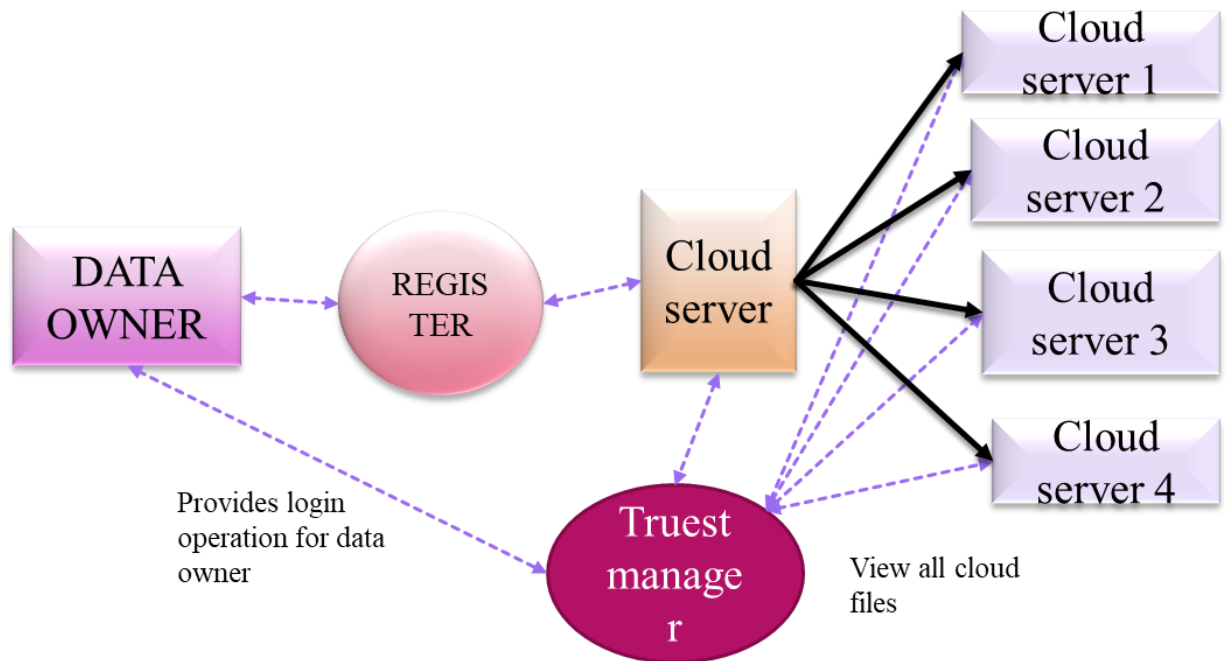


Fig- 4.5.2 Data Flow Diagram at level -1

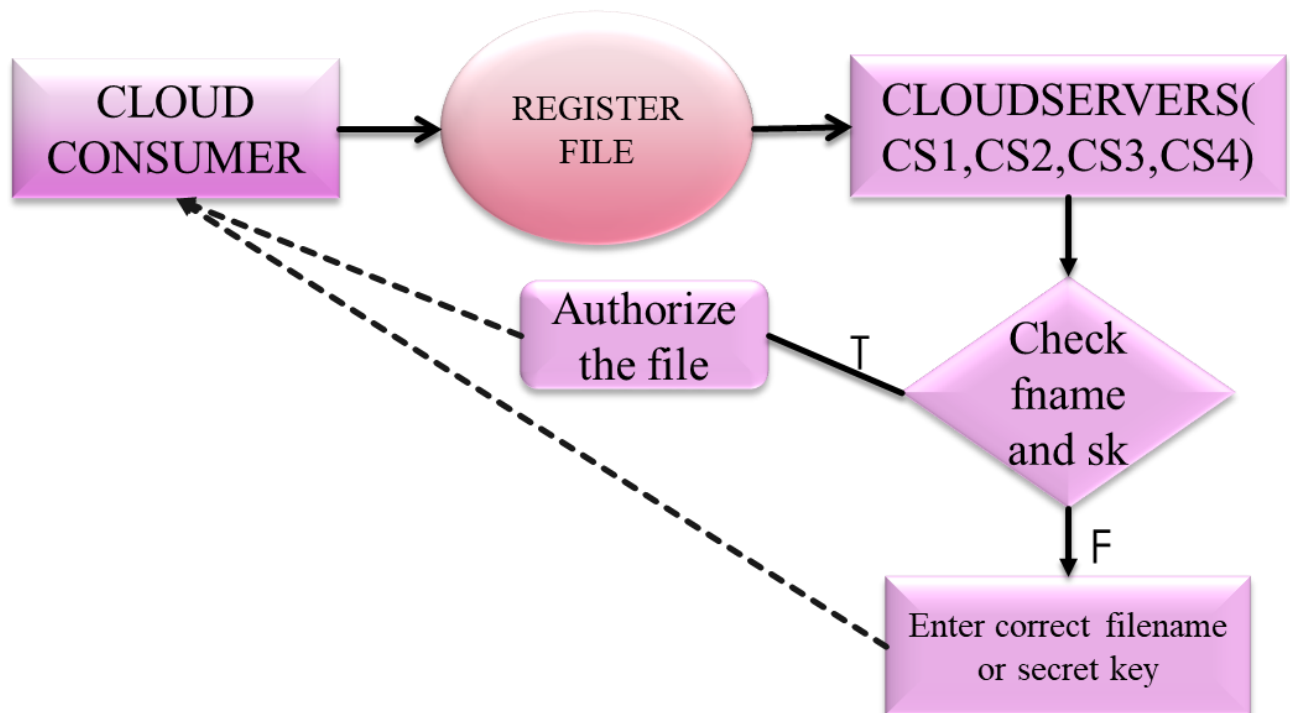


Fig- 4.5.3 Data Flow Diagram level -3

4.6 UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of Object-Oriented Computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML. The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML is a very important part of developing objects-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

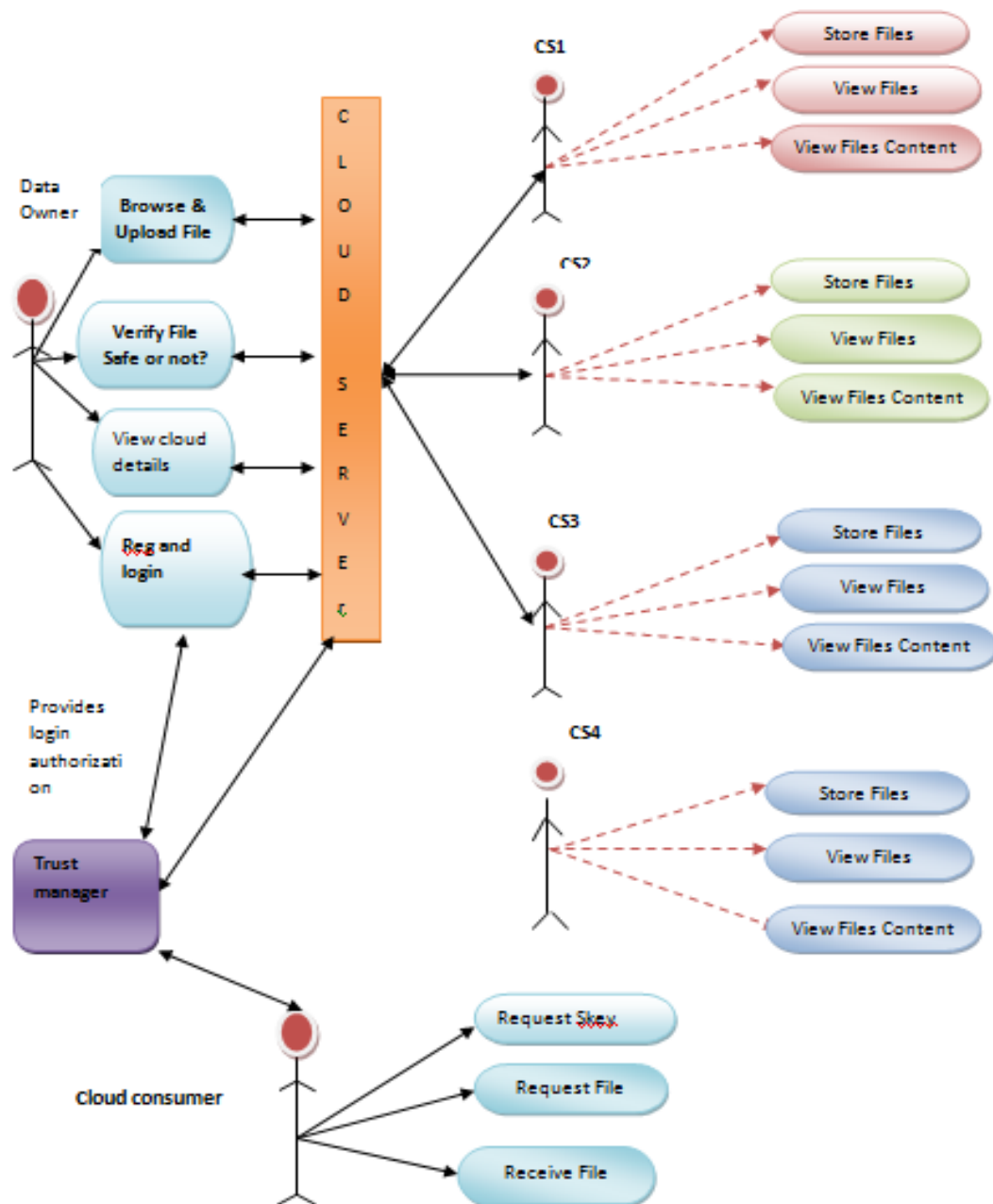


Fig -4.6.1 Use case Diagram

CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) 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 the classes. It explains which class contains information.

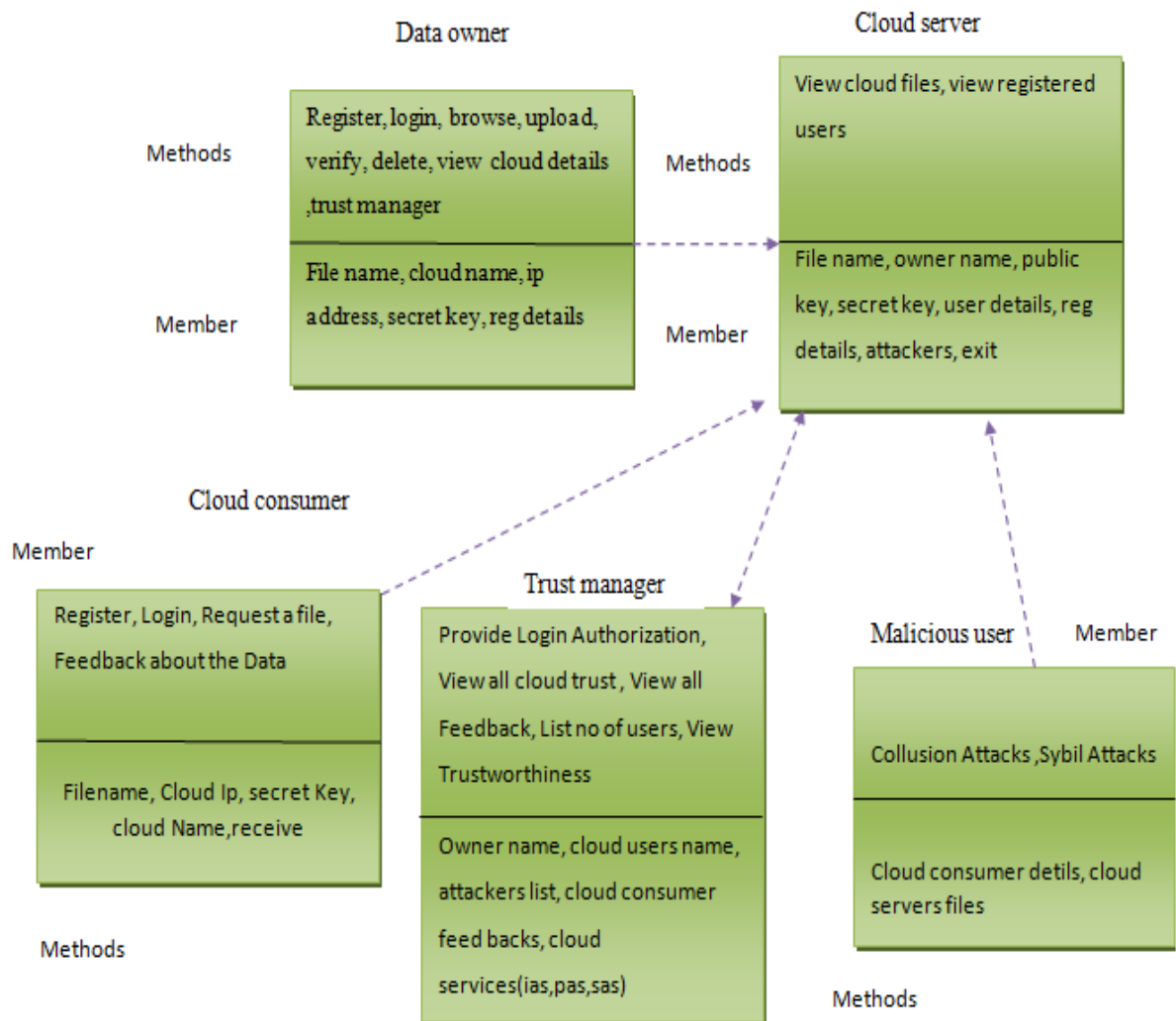


Fig: 4.6.2 Class Diagram

SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling 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. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagram

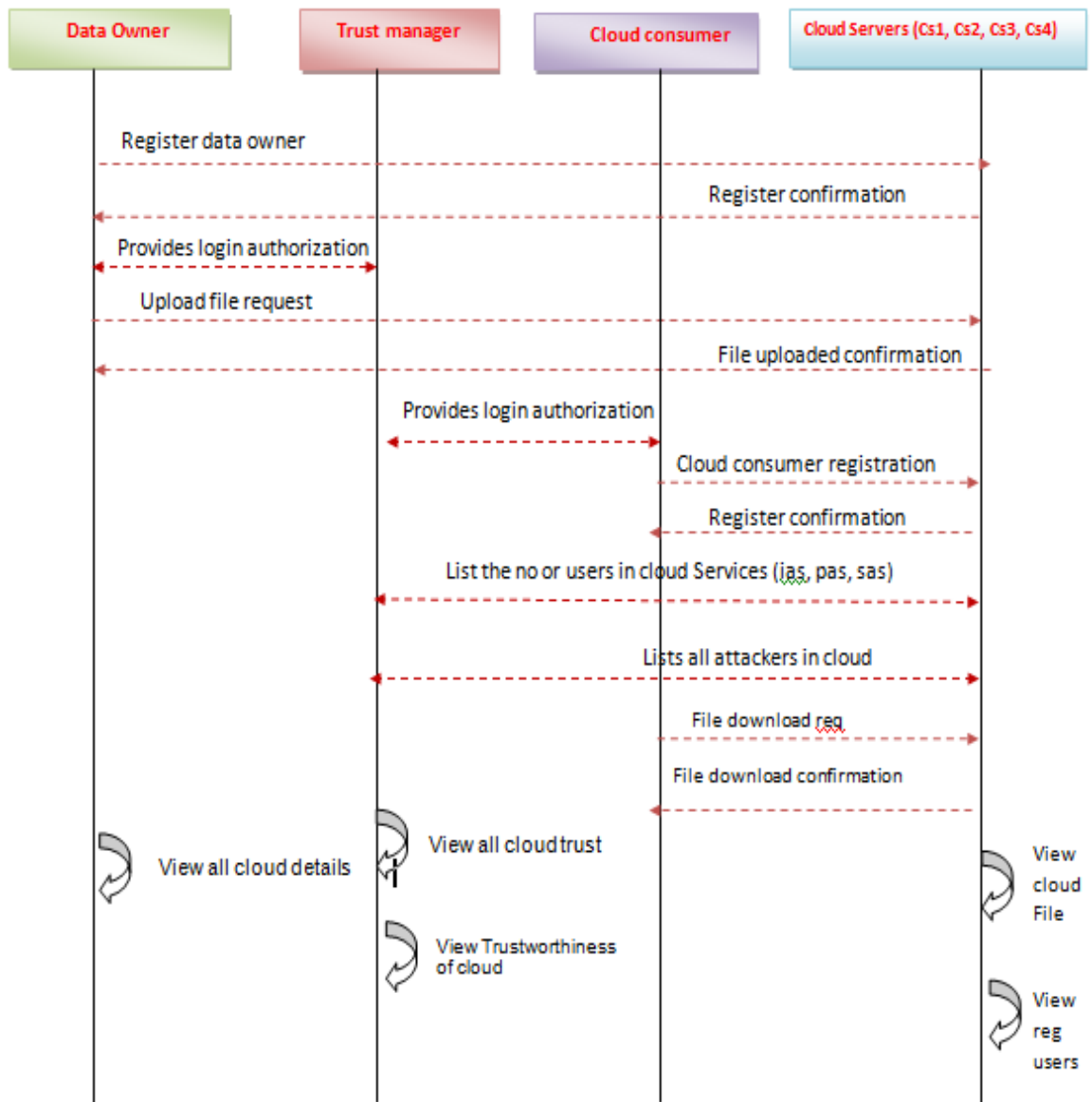


Fig: 4.6.3 Sequence Diagram

CHAPTER - 5

5. SOFTWARE ENVIRONMENT

Client Server:

Over view:

With the varied topic in existence in the fields of computers, Client Server is one, which has generated more heat than light, and also more hype than reality. This technology has acquired a certain critical mass attention with its dedication conferences and magazines. Major computer vendors such as IBM and DEC, have declared that Client Servers is their main future market. A survey of DBMS magazine revealed that 76% of its readers were actively looking at the client server solution. The growth in the client server development tools from \$200 million in 1992 to more than \$1.2 billion in 1996.

Client server implementations are complex but the underlying concept is simple and powerful. A client is an application running with local resources but able to request the database and relate the services from separate remote server. The software mediating this client server interaction is often referred to as MIDDLEWARE.

The typical client either a PC or a Work Station connected through a network to a more powerful PC, Workstation, Midrange or Main Frames server usually capable of handling request from more than one client. However, with some configuration server may also act as client. A server may need to access other server in order to process the original client request.

The key client server idea is that client as user is essentially insulated from the physical location and formats of the data needs for their application. With the proper middleware, a client input from or report can transparently access and manipulate both local database on the client machine and remote databases on one or more servers. An added bonus is the client server opens the door to multi-vendor database access indulging heterogeneous table joins.

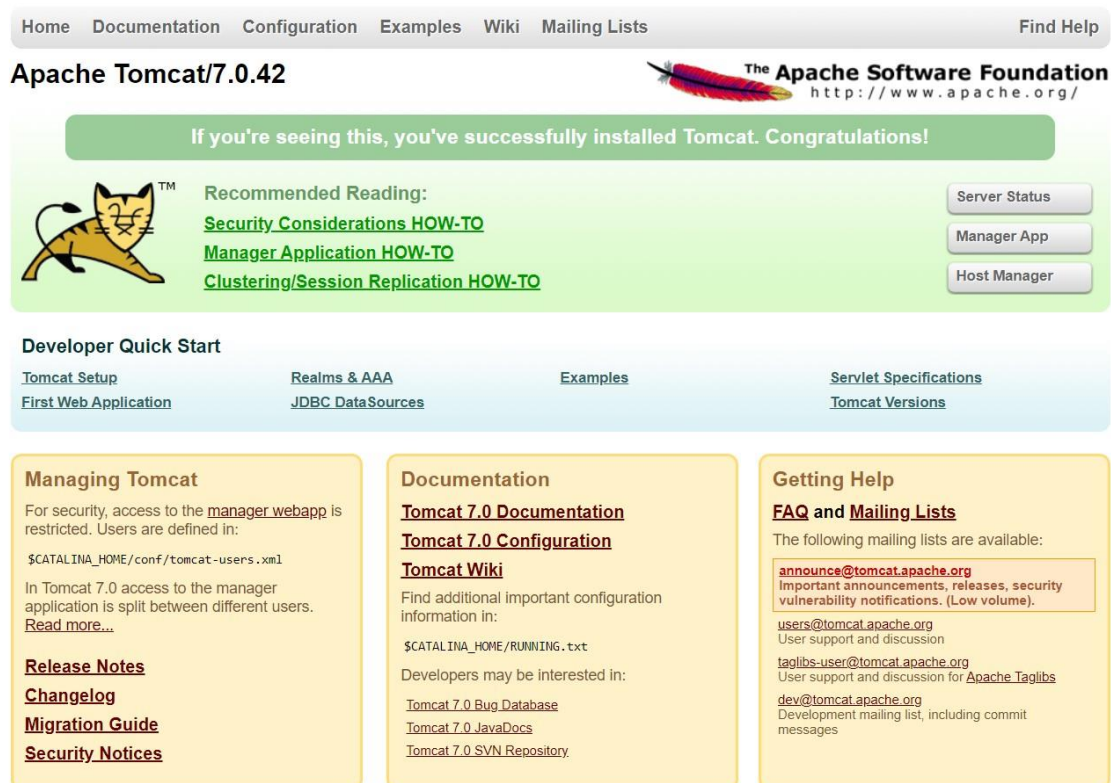
What is a client Server?

Two prominent systems in existence are client server and file server systems. It is essential to distinguish between client servers and file server systems. Both provide shared network access to data but the comparison ends there! The file server simply provides a remote disk drive that can be accessed by LAN applications on a file-by-file basis. The client server offers full relational database services such as SQL-Access, Record modifying, Insert, delete with full relational integrity backup/ restore performance for high volume of transactions, etc. the client server middleware provides a flexible interface between client

and server, who does what, when and to whom.

Tomcat 6.0 web server

Tomcat is an open-source web server developed by Apache Group. Apache Tomcat is the servlet container that is used in the official Reference Implementation for the Java Servlet and Java Server Pages technologies. The Java Servlet and Java Server Pages specifications are developed by Sun under the Java Community Process. Web Servers like Apache Tomcat support only web components while an application server supports web components as well as business components. To develop a web application with jsp/servlet install any web server like JRun, Tomcat etc. to run your application.



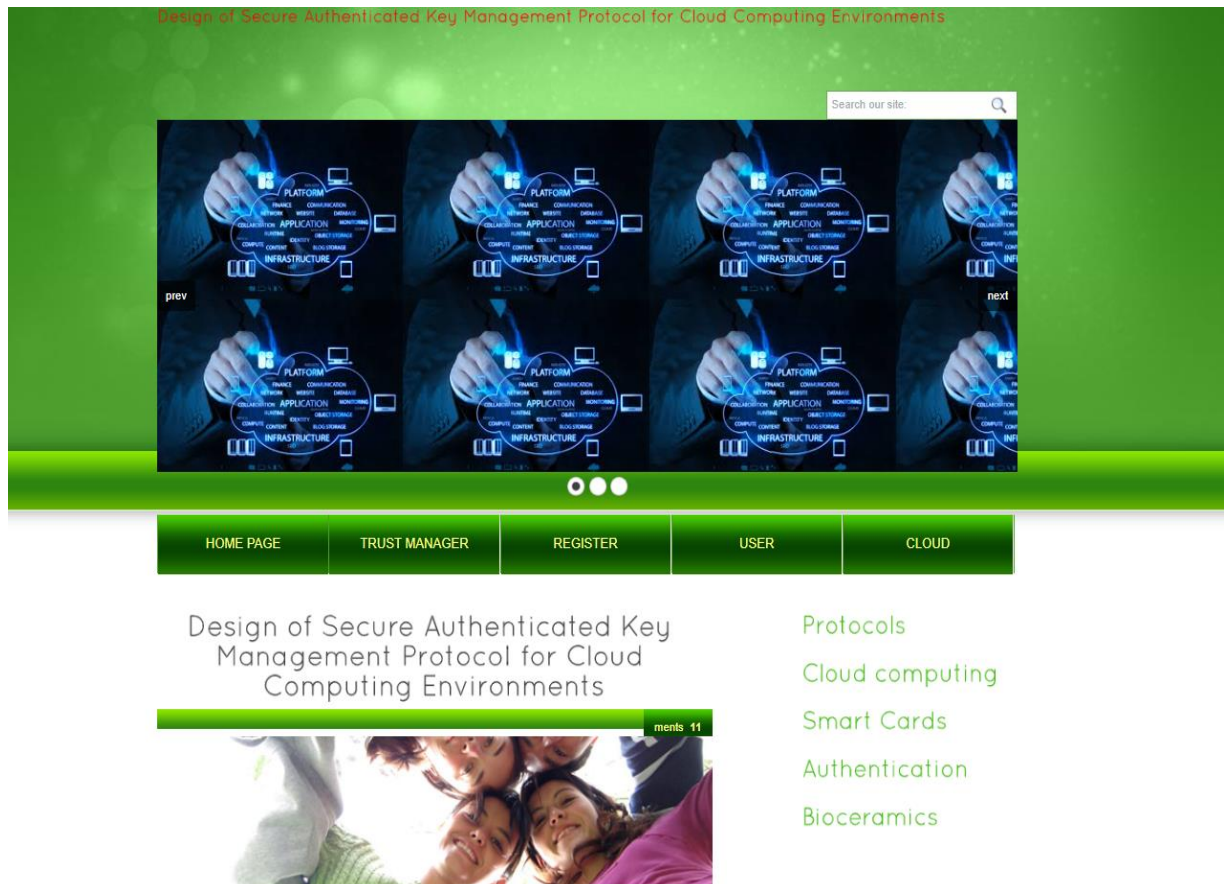
The screenshot displays the Apache Tomcat 7.0.42 web interface. At the top, a navigation bar includes links for Home, Documentation, Configuration, Examples, Wiki, Mailing Lists, and Find Help. The main heading is "Apache Tomcat/7.0.42" next to the Apache Software Foundation logo and URL. A green banner states: "If you're seeing this, you've successfully installed Tomcat. Congratulations!". Below this, a "Recommended Reading" section lists links for "Security Considerations HOW-TO", "Manager Application HOW-TO", and "Clustering/Session Replication HOW-TO", accompanied by a Tomcat cat logo. To the right are buttons for "Server Status", "Manager App", and "Host Manager". A "Developer Quick Start" section provides links for "Tomcat Setup", "First Web Application", "Realms & AAA", "JDBC DataSources", "Examples", "Servlet Specifications", and "Tomcat Versions". The bottom of the page features three yellow boxes: "Managing Tomcat" with security and user management information; "Documentation" with links to 7.0 documentation, configuration, wiki, and additional resources; and "Getting Help" with FAQ, mailing lists, and contact information.

Fig: 5.1 Tomcat server

CHAPTER - 6

6. IMPLEMENTATIONS AND RESULTS


6.1 SCREEN SHOTS



6.1.1 HOME PAGE

Design of Secure Authenticated Key Management Protocol for Cloud Computing Environments

Search our site:



HOME PAGE TRUST MANAGER REGISTER USER CLOUD

User Login Details

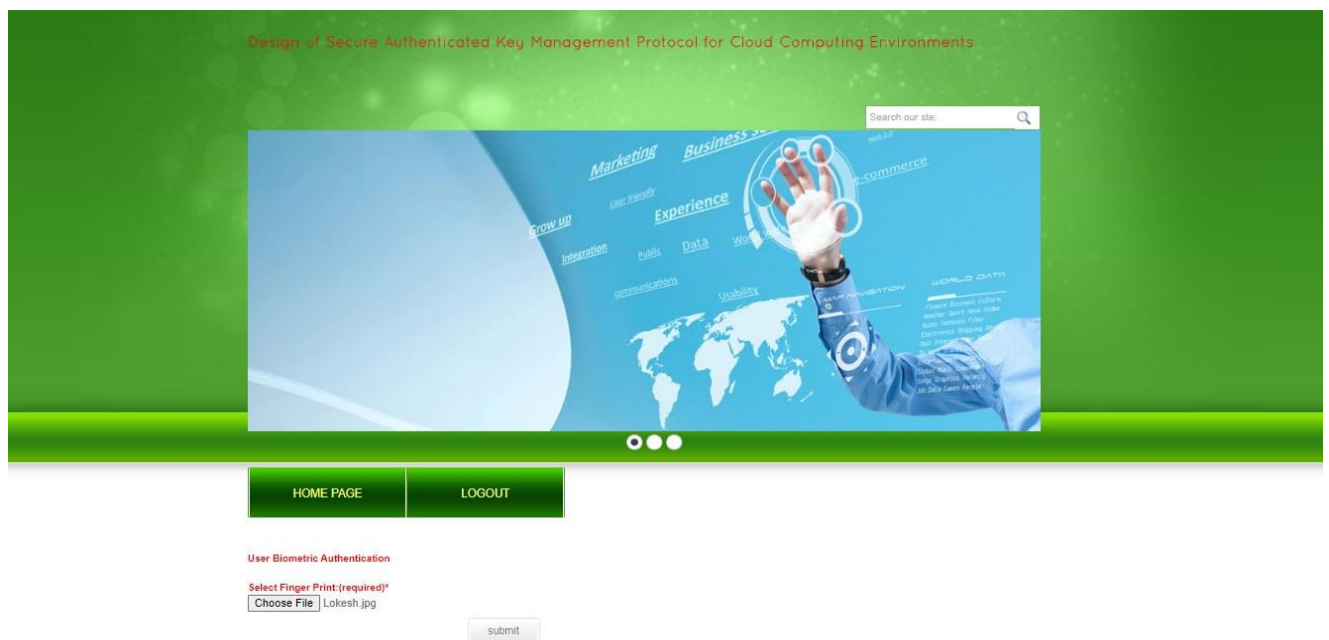
Name (required)

Password (required)

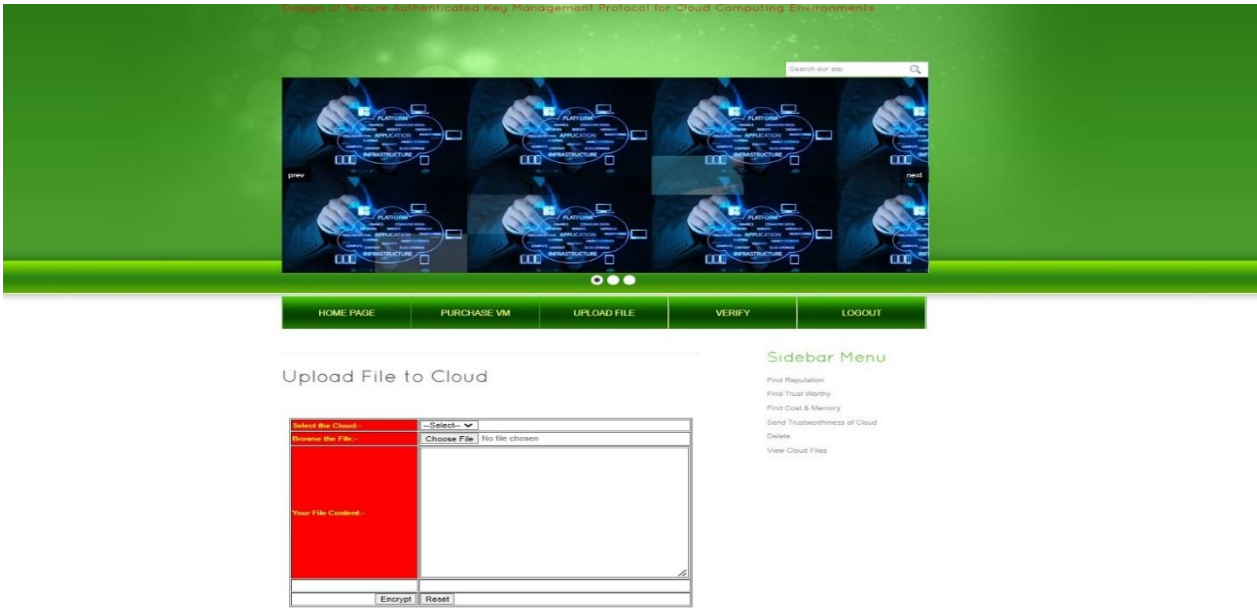
Select the User Type

Select the Cloud Server

6.1.2USER LOGIN PAGE



6.1.3 USER VERIFICATION PHASE



6.1.4 USER FILE UPLOAD PAGE

●●●

HOME PAGEPURCHASE VMUPLOAD FILEVERIFYLOGOUT

Verify the File in Cloud

Select the Cloud:-

--Select--

Enter the Filename:-

Submit

Reset

Sidebar Menu

Find Reputation

Find Trust Worthy

Find Cost & Memory

Send Trustworthiness of Cloud

Delete

View Cloud Files

6.1.5 USER FILE VERIFICATION PAGE

36

HOME PAGE	TRUST MANAGER	REGISTER	USER	CLOUD
-----------	---------------	----------	------	-------

User Registration

Name (required)

Password (required)

DOB(dd/mm/yyyy)

Email Address (required)

Mobile NO(10 Digits)

Location

Select User Type

--Select--

Select Services

--Select--

Select Cloud

--Select--

Choose Photo:*

Choose File No file chosen

Select Finger Print:(required)*

Choose File No file chosen

submit

Index Terms

Protocols,
Cloud computing,
Smart cards,
Authentication,
Biocremics,

6.1.6 REGISTRATION PAGE

HOME PAGE	TRUST MANAGER	REGISTER	USER	CLOUD
-----------	---------------	----------	------	-------

Cloud Login Details

Name (required)

cs1

Password (required)

...

Select the Cloud

--Select--

submit

Menu Operations

Home

6.1.7 CLOUD LOGIN PAGE

HOME PAGE	LIST ALL FILES	LIST ALL USERS	LIST ALL VMS	LOGOUT
-----------	----------------	----------------	--------------	--------

View Cloud Files

Owner	Cloud	File Name	Hash	Public Key	Private Key	Date & Time
kumar	CS1	otnet.txt	-11a263b1e4ee7bdce125577954f240fad3730	[B@4fa9f041	[B@134e3e85	20/12/2019
allen	CS1	pavan.txt	5fe4d1399f0708a16f3deacdcd7001af02d6c35e	[B@f2d9ee	[B@db2121	11/05/2022
lokesh	CS1	sample test.txt	-4433782ceedc7de1223e8ee499836f4c1457ea5c	[B@11fad02	[B@108154f	11/05/2022
lokesh	CS1	sample.txt	-4c5b889f054eab52dbd3774714836f52deb2711d	[B@3377d	[B@171d315	11/05/2022

6.1.8 CLOUD FILES PAGE

Design of Secure Authenticated Key Management Protocol for Cloud Computing Environments



TMS Login Details

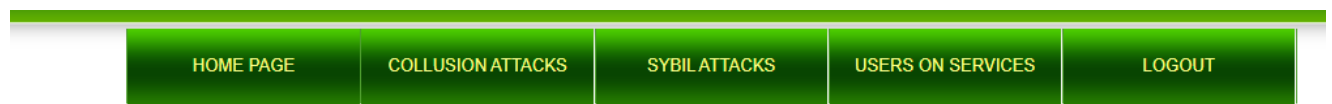
Name (required)

Password (required)

Sidebar Menu

Home

6.1.9 TRUST MANGER LOGIN PAGE



View Users on Services


Select the Service

Menu Operations

- List +ve Feedbacks
- List -ve Feedbacks
- List All Users
- List All Content Attackers
- List All User Downloads



View Users on Services

User Image	User Name	DOB	E-Mail	Mobile	Location	User Type	Service Type	Cloud
	allen	15/05/2001	admin	9087654321	568908	Data Owner	SaaS	CS1

6.1.10 LIST OF USERS IN CLOUD SERVICES PAGE

6.2 CODING

- **ADMIN.html:**

```
<!DOCTYPE html>
<html>
<head>
<title>Design of Secure Authenticated Key Management Protocol for Cloud
Computing Environments</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<link href="css/style.css" rel="stylesheet" type="text/css" />
<link rel="stylesheet" type="text/css" href="css/coin-slider.css" />
<script type="text/javascript" src="js/cufon-yui.js"></script>
<script type="text/javascript" src="js/cufon-quicksand.js"></script>
<script type="text/javascript" src="js/jquery-1.4.2.min.js"></script>
<script type="text/javascript" src="js/script.js"></script>
<script type="text/javascript" src="js/coin-slider.min.js"></script>
<style type="text/css">
<!--
.style1 {
    font-size: 18px;
    color: #FF0000;
    font-weight: bold;
    font-style: italic;
}
.style2 {
    color: #FF0000;
    font-weight: bold;
}
.style3 {color: #FF0000}
.style4 {color: #0000FF}
.style5 {color: #003300}
-->
</style>
</head>
```

```

<body>
<div class="main">
  <div class="header">
    <div class="header_resize">
      <div class="logo">
        <h1><a href="index.html"><span class="style1">Design of Secure
Authenticated Key Management Protocol for Cloud Computing
Environments</span> </a></h1>
      </div>
      <div class="searchform">
        <form id="formsearch" name="formsearch" method="post" action="#">
          <span>
            <input name="editbox_search" class="editbox_search"
id="editbox_search" maxlength="80" value="Search our site:" type="text" />
          </span>
          <input name="button_search" src="images/search.gif"
class="button_search" type="image" />
        </form>
      </div>
      <div class="clr"></div>
      <div class="slider">
        <div id="coin-slider"><a href="#"></a><a href="#"></a> <a
href="#"></a> </div>
      <div class="clr"></div>
    </div>
    <div class="clr"></div>
    <div class="menu_nav">
      <ul>
        <li class="active"><a href="index.html"><span>Home
Page</span></a></li>
        <li class="active"><a href="admin.html">Trust Manager</a></li>

```

```

        <li class="active"><a
href="register.html"><span>Register</span></a></li>
        <li class="active"><a href="user.html"><span>USER</span></a></li>
        <li class="active"><a href="cloud.html"><span>CLOUD</span></a></li>

    </ul>
</div>
<div class="clr"></div>
</div>
</div>
<div class="content">
    <div class="content_resize">
        <div class="mainbar">
            <div class="article">

                <div class="mainbar">
                    <div class="article">
                        <p>&nbsp;</p>
                        <h2><font color="red">TMS Login Details</h2>
                        <div class="clr"></div>
                        <form action="admin_login.jsp" method="post" id="">
                            <ol>
                                <li class="active">
                                    <label for="name"><B>Name (required)</label>
                                    <input id="adminid" name="adminid" type="text" />
                                </li>
                                <li class="active">
                                    <label for="name"><B>Password (required)</label>
                                    <input id="pass" name="pass" type="password" />
                                </li>

                                <li class="active">
                                    <input type="image" name="imageField" id="imageField"
src="images/submit.gif" class="send" />

```

```

        <div class="clr"></div>
    </li>
</ol>
</form>
</div>
</div>
</div>
<div class="article">
    <h2>&nbsp;</h2>
    <div class="clr"></div>
</div>
</div>
<div class="sidebar">
    <div class="gadget">
        <h2 class="star"><span>Sidebar</span> Menu</h2>
        <div class="clr"></div>
        <ul class="sb_menu">
            <li class="active"><a href="#">Home</a></li>

        </ul>
    </div>
</div>
</div>
<div class="clr"></div>
</div>
<div class="fbg">
    <div class="fbg_resize">
        <div class="col c1">
            <h2>&nbsp;</h2>
        </div>
        <div class="col c2">
            <h2>&nbsp;</h2>
        </div>
        <div class="col c3">

```

```

        <h2>&nbsp;</h2>
    </div>
    <div class="clr"></div>
</div>
</div>
<div align=center></div>
</body>
</html>

```

•ADMIN. SQL:

```

SELECT CONNECTION_ID();

/* Connected. Thread-ID: 44 */

/* Unknown character set: 'utf8mb4' */

/* Characterset: utf8 */

SHOW STATUS;

SHOW VARIABLES;

SHOW DATABASES;

USE `ctrust`;

/* Entering session "local" */

SELECT `DEFAULT_COLLATION_NAME` FROM
`information_schema`.`SCHEMATA` WHERE `SCHEMA_NAME`='ctrust';

SHOW TABLE STATUS FROM `ctrust`;

SHOW FUNCTION STATUS WHERE `Db`='ctrust';

SHOW PROCEDURE STATUS WHERE `Db`='ctrust';

SHOW TRIGGERS FROM `ctrust`;

SELECT `DEFAULT_COLLATION_NAME` FROM
`information_schema`.`SCHEMATA` WHERE
`SCHEMA_NAME`='information_schema';

SHOW TABLE STATUS FROM `information_schema`;

SHOW FUNCTION STATUS WHERE `Db`='information_schema';

SHOW PROCEDURE STATUS WHERE `Db`='information_schema';

```

```

SHOW TRIGGERS FROM `information_schema`;

SHOW EVENTS FROM `information_schema`;

SELECT *, EVENT_SCHEMA AS `Db`, EVENT_NAME AS `Name` FROM
information_schema.`EVENTS` WHERE `EVENT_SCHEMA`='ctrust';

/* Access is denied */

USE `information_schema`;

USE `ctrust`;

SHOW CREATE TABLE `ctrust`.`cost`;

SHOW COLLATION;

SHOW ENGINES;

SELECT `DEFAULT_COLLATION_NAME` FROM
`information_schema`.`SCHEMATA` WHERE `SCHEMA_NAME`='mysql';

SHOW TABLE STATUS FROM `mysql`;

SHOW FUNCTION STATUS WHERE `Db`='mysql';

SHOW PROCEDURE STATUS WHERE `Db`='mysql';

SHOW TRIGGERS FROM `mysql`;

SELECT *, EVENT_SCHEMA AS `Db`, EVENT_NAME AS `Name` FROM
information_schema.`EVENTS` WHERE `EVENT_SCHEMA`='mysql';

USE `mysql`;

SHOW CREATE TABLE `mysql`.`columns_priv`;

SHOW CREATE TABLE `mysql`.`db`;

SHOW CREATE TABLE `mysql`.`user`;

SELECT `Host`, `User`, `Password`, `Select_priv`, `Insert_priv`,
`Update_priv`, `Delete_priv`, `Create_priv`, `Drop_priv`, `Reload_priv`,
`Shutdown_priv`, `Process_priv`, `File_priv`, `Grant_priv`,
`References_priv`, `Index_priv`, `Alter_priv`, `Show_db_priv`, `Super_priv`,
`Create_tmp_table_priv`, `Lock_tables_priv`, `Execute_priv`,
`Repl_slave_priv`, `Repl_client_priv`, `Create_view_priv`,
`Show_view_priv`, `Create_routine_priv`, `Alter_routine_priv`,
`Create_user_priv`, `Event_priv`, `Trigger_priv`, `Create_tablespace_priv`,

```



```

`ssl_type`, LEFT(`ssl_cipher`, 256), LEFT(`x509_issuer`, 256),
LEFT(`x509_subject`, 256), `max_questions`, `max_updates`,
`max_connections`, `max_user_connections` FROM `mysql`.`user` LIMIT
1000;

SHOW CREATE TABLE `mysql`.`user`;

USE `ctrust`;

SHOW CREATE TABLE `ctrust`.`admin`;

SELECT * FROM `ctrust`.`admin` LIMIT 1000;

SHOW CREATE TABLE `ctrust`.`admin`;

SHOW CREATE TABLE `ctrust`.`attackers`;

SELECT * FROM `ctrust`.`attackers` LIMIT 1000;

SHOW CREATE TABLE `ctrust`.`attackers`;

SHOW CREATE TABLE `ctrust`.`attackers1`;

SELECT * FROM `ctrust`.`attackers1` LIMIT 1000;

SHOW CREATE TABLE `ctrust`.`attackers1`;

SHOW CREATE TABLE `ctrust`.`attackers2`;

SELECT * FROM `ctrust`.`attackers2` LIMIT 1000;

SHOW CREATE TABLE `ctrust`.`attackers2`;

SHOW CREATE TABLE `ctrust`.`backup`;

SELECT `id`, LEFT(`oname`, 256), LEFT(`cname`, 256), LEFT(`fname`,
256), LEFT(`mac`, 256), LEFT(`key1`, 256), LEFT(`key2`, 256), LEFT(`pt`,
256), LEFT(`ct`, 256), LEFT(`dt`, 256) FROM `ctrust`.`backup` LIMIT 1000;

SHOW CREATE TABLE `ctrust`.`backup`;

SHOW CREATE TABLE `ctrust`.`cfiles`;

SELECT `id`, LEFT(`oname`, 256), LEFT(`cname`, 256), LEFT(`fname`,
256), LEFT(`mac`, 256), LEFT(`key1`, 256), LEFT(`key2`, 256), LEFT(`pt`,
256), LEFT(`ct`, 256), LEFT(`dt`, 256), LEFT(`tdelay`, 256),
LEFT(`throughput`, 256) FROM `ctrust`.`cfiles` LIMIT 1000;

SHOW CREATE TABLE `ctrust`.`cfiles`;

SHOW CREATE TABLE `ctrust`.`cloud`;

```

```

SELECT * FROM `ctrust`.`cloud` LIMIT 1000;

SHOW CREATE TABLE `ctrust`.`cloud`;

SHOW CREATE TABLE `ctrust`.`cost`;

SELECT * FROM `ctrust`.`cost` LIMIT 1000;

SHOW CREATE TABLE `ctrust`.`cost`;

SHOW CREATE TABLE `ctrust`.`feedback`;

SELECT `id`, LEFT(`oname`, 256), LEFT(`cname`, 256), LEFT(`feedback`,
256), LEFT(`dt`, 256) FROM `ctrust`.`feedback` LIMIT 1000;

SHOW CREATE TABLE `ctrust`.`feedback`;

SHOW CREATE TABLE `ctrust`.`user`;

SELECT `id`, LEFT(`uname`, 256), LEFT(`pwd`, 256), LEFT(`dob`, 256),
LEFT(`email`, 256), LEFT(`mobile`, 256), LEFT(`location`, 256),
LEFT(`utype`, 256), LEFT(`stype`, 256), LEFT(`cname`, 256),
LEFT(`imagess`, 256), `count`, LEFT(`sk`, 256), LEFT(`finger`, 256) FROM
`ctrust`.`user` LIMIT 1000;

SHOW CREATE TABLE `ctrust`.`user`;

SHOW CREATE TABLE `ctrust`.`vm`;

SELECT `id`, LEFT(`oname`, 256), LEFT(`cname`, 256), LEFT(`memory`,
256), LEFT(`cost`, 256), LEFT(`bw`, 256), LEFT(`dt`, 256) FROM
`ctrust`.`vm` LIMIT 1000;

SHOW CREATE TABLE `ctrust`.`vm`;

SHOW CREATE TABLE `ctrust`.`admin`;

```

CHAPTER - 7

7. SYSTEM TESTING

7.1 INTRODUCTION:

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 tests. Each test type addresses a specific testing requirement.

7.2 SYSTEM TESTING:

TESTING METHODOLOGIES:

The following are the Testing Methodologies:

- ❖ Unit Testing.
- ❖ Integration Testing.
- ❖ User Acceptance Testing.
- ❖ Output Testing.
- ❖ Validation Testing.
- ❖

Unit Testing:

Unit testing focuses verification effort on the smallest unit of Software design that is the module. Unit testing exercises specific paths in a module's control structure to ensure complete coverage and maximum error detection. This test focuses on each module individually, ensuring that it functions properly as a unit. Hence, the naming is Unit Testing. During this testing, each module is tested individually and the module interfaces are verified for the consistency with design specification. All-important processing paths are tested for the expected results. All error handling paths are also tested.

Integration Testing:

Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high order tests are conducted. The main objective in this testing process is to take unit tested modules and build a program structure that has been dictated by design.

The following are the types of Integration Testing:

1. Top-Down Integration:

This method is an incremental approach to the construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main program module. The module subordinates to the main program module are incorporated into the structure in either a depth first or breadth first manner.

In this method, the software is tested from main module and individual stubs are replaced when the test proceeds downwards.

2. Bottom-up Integration:

This method begins the construction and testing with the modules at the lowest level in the program structure. Since the modules are integrated from the bottom up, processing required for modules subordinate to a given level is always available and the need for stubs is eliminated. The bottom-up integration strategy may be implemented with the following steps:

- ❖ The low-level modules are combined into clusters into clusters that perform a specific Software sub-function.
- ❖ A driver the control program for testing is written to coordinate test case input and output.
- ❖ The cluster is tested.
- ❖ Drivers are removed and clusters are combined moving upward in the program structure

User Acceptance Testing:

User Acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required. The system developed provides a friendly user interface that can easily be understood even by a person who is new to the system.

Output Testing:

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in the specified format. Asking the users about the format required by them tests the outputs generated or displayed by the system under consideration. Hence the output format is considered in 2 ways – one is on screen and another in printed format.

Validation Checking:

Validation checks are performed on the following fields.

Text Field:

The text field can contain only the number of characters lesser than or equal to its size. The text fields are alphanumeric in some tables and alphabetic in other tables. Incorrect entry always flashes and error message.

Numeric Field:

The numeric field can contain only numbers from 0 to 9. An entry of any character flashes an error message. The individual modules are checked for accuracy and what it has to perform. Each module is subjected to test run along with sample data. The individually tested modules are integrated into a single system. Testing involves executing the real data information is used in the program the existence of any program defect is inferred from the output. The testing should be planned so that all the requirements are individually tested. A successful test is one that gives out the defects for the inappropriate data and produces and output revealing the errors in the system.

Preparation of Test Data:

Taking various kinds of test data does the above testing. Preparation of test data plays a vital role in the system testing. After preparing the test data the system under study is tested using that test data. While testing the system by using test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

Using Live Test Data:

Live test data are those that are actually extracted from organization files. After a system is partially constructed, programmers or analysts often ask users to key in a set of data from their normal activities. Then, the systems person uses this data as a way to partially test the system. In other instances, programmers or analysts extract a set of live data from the files and have them entered themselves. It is difficult to obtain live data in sufficient amounts to conduct extensive testing. And, although it is realistic data that will show how the system will perform for the typical processing requirement, assuming that the live data entered are in fact typical, such data generally will not test all combinations or formats that can enter the system. This bias toward typical values then does not provide a true system test and in fact ignores the cases most likely to cause system failure.

Using Artificial Test Data:

Artificial test data are created solely for test purposes, since they can be generated to test all combinations of formats and values. In other words, the artificial data, which can quickly be prepared by a data generating utility program in the information systems department, make possible the testing of all login and control paths through the program.

The most effective test programs use artificial test data generated by persons other than those who wrote the programs. Often, an independent team of testers formulates a testing plan, using the systems specifications. The package “Virtual Private Network” has satisfied all the requirements specified as per software requirement specification and was accepted.

USER TRAINING:

Whenever a new system is developed, user training is required to educate them about the working of the system so that it can be put to efficient use by those for whom the system has been primarily designed. For this purpose, the normal working of the project was demonstrated to the prospective users. Its working is easily understandable and since the expected users are people who have good knowledge of computers, the use of this system is very easy.

MAINTAINENCE:

This covers a wide range of activities including correcting code and design errors. To reduce the need for maintenance in the long run, we have more accurately defined the user's requirements during the process of system development. Depending on the requirements, this system has been developed to satisfy the needs to the largest possible extent. With development in technology, it may be possible to add many more features based on the requirements in future. The coding and designing are simple and easy to understand which will make maintenance easier.

TESTING STRATEGY:

A strategy for system testing integrates system test cases and design techniques into a well-planned series of steps that results in the successful construction of software. The testing strategy must co-operate test planning, test case design, test execution, and the resultant data collection and evaluation. A strategy for software testing must accommodate low-level tests that are necessary to verify that a small source code segment has been correctly implemented as well as high level tests that validate major system functions against user requirements.

Software testing is a critical element of software quality assurance and represents the ultimate review of specification design and coding. Testing represents an interesting anomaly for the software. Thus, a series of testing are performed for the proposed system before the system is ready for user acceptance testing.

SYSTEM TESTING:

Software once validated must be combined with other system elements (e.g., Hardware, people, database). System testing verifies that all the elements are proper and that overall system function performance is achieved. It also tests to find discrepancies between the system and its original objective, current specifications and system documentation.

ctrust.user: 2 rows total (approximately)

id	uname	puid	dob	email	mobile	location	utype	stype	cname	image	count	sk
5	allen	admin	15/05/2001	admin	9087654321	568908	Data Owner	SaaS	CS1	0xFFD8FFE000104A46494600010101006000600000FFD...	0	Rejected
6	lokesh	lokesh	12/12/1992	lokesh@gmail.com	9851465489	hyd	Cloud Consumer	Paas	CS2	0xFFD8FFE000104A46494600010100000100010000FFD...	0	Rejected

Fig: 7.2.1 User login result

Host: 127.0.0.1 Database: trustr Table: files Data Query

ctrust.files: 4 rows total (approximately)

id	uname	cname	fname	mac	key1	key2	pt	ct	dt
11	kumar	CS1	otnet.txt	-11a263b1e4ee7bdce125577954f240fad3730	[B@4fa9f041	[B@134e3e85	R690bmV0IGZlOGpGdHvZ3JhbWVpbmcbGfUz3VhZ2Jgd...	Ukc5MGJhYjR2x6SUFpZ2N5ZmM0poYlxcGjY2dR0Z1...	20/12/2019
12	allen	CS1	pavan.txt	5fe4d1399f0708a16f3deacdc7001a8f02d6c35e	[B@f2d9ee	[B@db2121	free cam aawastejst 2 or 5 minokdone...	ZnJZS8jYVW0gYVENChndhc3RDQgc3QgMBv0A1IG1pbq...	11/05/2022
13	lokesh	CS1	sample test.txt	-4433782cecd7de1223e8ee499836f4c1457ea5c	[B@11fad02	[B@108154f	verify the file first	dmVyaWZ5ZSRoZSBmaWlIGZpcnN0Q0Qo=	11/05/2022
14	lokesh	CS1	sample.txt	-4c5b889f054eab52dbd3774714836f53deb2711d	[B@3377d	[B@171d315	in cloud server maintain the data secure	aGkgV2xvdWQgc2YydmVvIGthaiW50YVWuH-RoZSBkYRh...	11/05/2022

Fig: 7.2.2 Data Base result

Users Names	Cloud Names	File Name	Result
Allen	CS1	otnet.txt	Success
Lokesh	CS1	pavan.txt	Success
Lokesh	CS1	sample test.txt	Success
Kumar	CS1	sample.txt	Success

7.2.3 Data Base Test Cases

Users Names	Cloud Names	Result
Allen	CS1	Success
Lokesh	CS1	Success
Lokesh	CS1	Success
Kumar	CS1	Success

7.2.4 User Login Test Cases

CHAPTER - 8

8. CONCLUSION

To resist the exhaustion of password attack on the two-factor MAKa protocols, a large number of three-factor MAKa protocols have been proposed. However, almost all three factor MAKa protocols don't provide formal proofs and dynamic user management mechanism. In order to achieve more flexible user management and higher security, this paper proposes a new three-factor MAKa protocol that supports dynamic revocation and provides formal proof.

Future Scope

The security shows that our protocol achieves the security properties of requirements from multi-server environments. On the other hand, through the comprehensive analysis of performance, our protocol doesn't sacrifice efficiency while improving the function. On the contrary, the proposed protocol has great advantages in terms of the total computation time.

CHAPTER - 9

9. REFERENCES

- [1] P. Tsantarliotis, E. Pitoura, and P. Tsaparas, “Defining and predicting troll vulnerability in online social media,” *Social Network Analysis and Mining*, vol. 7, no. 1, p. 26, 2017.
- [2] J. Cheng, C. Danescu-Niculescu-Mizil, J. Leskovec, and M. Bernstein, “Anyone can become a troll,” *American Scientist*, vol. 105, no. 3, p. 152, 2017.
- [3] S. Sood, J. Antin, and E. Churchill, “Profanity use in online communities,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2012, pp. 1481–1490.
- [4] S. Rojas-Galeano, “On obstructing obscenity obfuscation,” *ACM Transactions on the Web (TWEB)*, vol. 11, no. 2, p. 12, 2017.
- [5] Hate-Speech, “Oxford dictionaries,” retrieved August 30, 2017 from [https://en.oxforddictionaries.com/definition/hate speech](https://en.oxforddictionaries.com/definition/hate%20speech).
- [6] Z. Waseem and D. Hovy, “Hateful symbols or hateful people? Predictive features for hate speech detection on twitter.” in *SRW@ HLT-NAACL*, 2016, pp. 88–93.
- [7] P. Badjatiya, S. Gupta, M. Gupta, and V. Varma, “Deep learning for hate speech detection in tweets,” in *Proceedings of the 26th International Conference on World Wide Web Companion*. International World Wide Web Conferences Steering Committee, 2017, pp. 759–760.

CHAPTER -10

ANNEXURE – 1
DOMAIN TECHNOLOGIES

10. VISIBLE PROJECT WORK OUTPUT

10.1 DOMAIN TECHNOLOGY: CLOUD COMPUTING

Cloud computing transforms IT infrastructure into a utility: It lets you ‘plug into’ infrastructure via the internet, and use computing resources without installing and maintaining them on-premises.

What is Cloud Computing?

Cloud computing is on demand access, via the internet, to computing resources applications, servers, data storage, development tools, networking capabilities, and more hosted at a remote data center managed by a cloud services provider. The CSP makes these resources available for a monthly subscription fee or bills them according to usage.

Compared to traditional on-premises IT, and depending on the cloud services you select, cloud computing helps do the following:

Lower IT costs:

Cloud lets you offload some or most of the costs and effort of purchasing, installing, configuring, and managing your own on premises infrastructure.

Improve agility and time to value:

With cloud, your organization can start using enterprise applications in minutes, instead of waiting weeks or months for IT to respond to a request, purchase and configure supporting hardware, and install software. Cloud also lets you empower certain users specifically developers and data scientists to help themselves to software and support infrastructure.

Scale more easily and cost effectively:

Cloud provides elasticity instead of purchasing excess capacity that sits unused during slow periods, you can scale capacity up and down in response to spikes and dips in traffic. You can also take advantage of your cloud provider’s global network to spread your applications closer to users around the world.

Types of cloud computing?

Public cloud

Public cloud is a type of cloud computing in which a cloud service provider makes computing resources anything from SaaS applications, to individual virtual machines (VMs), to bare metal computing hardware, to complete enterprise grade infrastructures and development platforms available to users over the public internet. These resources might be accessible for free, or access might be sold according to subscription-based or pay-per-usage pricing models.

The public cloud provider owns, manages, and assumes all responsibility for the data centers, hardware, and infrastructure on which its customers' workloads run, and it typically provides high bandwidth network connectivity to ensure high performance and rapid access to applications and data.

Public cloud is a multi-tenant environment the cloud provider's data center infrastructure is shared by all public cloud customers. In the leading public clouds Amazon Web Services (AWS), Google Cloud, IBM Cloud, Microsoft Azure, and Oracle Cloud those customers can number in the millions.

Many enterprises are moving portions of their computing infrastructure to the public cloud because public cloud services are elastic and readily scalable, flexibly adjusting to meet changing workload demands. Others are attracted by the promise of greater efficiency and fewer wasted resources since customers pay only for what they use. Still others seek to reduce spending on hardware and on-premises infrastructures.

Private cloud

Private cloud is a cloud environment in which all cloud infrastructure and computing resources are dedicated to, and accessible by, one customer only. Private cloud combines many of the benefits of cloud computing including elasticity, scalability, and ease of service delivery with the access control, security, and resource customization of on-premises infrastructure.

A private cloud is typically hosted on-premises in the customer's data center. But a private cloud can also be hosted on an independent cloud provider's infrastructure or built on rented infrastructure housed in an offsite data center.

Many companies choose private cloud over public cloud because private cloud is an easier way (or the only way) to meet their regulatory compliance requirements. Others choose private cloud because their workloads deal with confidential documents, intellectual property, personally identifiable information (PII), medical records, financial data, or other sensitive data. By building private cloud architecture according to cloud native principles, an organization gives itself the flexibility to easily move workloads to public cloud or run them within a hybrid cloud (see below) environment whenever they're ready.

Hybrid cloud

Hybrid cloud is just what it sounds like a combination of public and private cloud environments. Specifically, and ideally, a hybrid cloud connects an organization's private cloud services and public clouds into a single, flexible infrastructure for running the organization's applications and workloads.

The goal of hybrid cloud is to establish a mix of public and private cloud resources and with a level of orchestration between them that gives an organization the flexibility to choose the optimal cloud for each application or workload and to move workloads freely between the two clouds as circumstances change. This enables the organization to meet its technical and business objectives more effectively and cost-efficiently than it could with public or private cloud alone.