

SECURE DATA STORAGE USING CRYPTOGRAPHY

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

By

N.Pramodh (37110522)
S.Sai Pranav Tej (37110667)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND
TECHNOLOGY (DEEMED TO BE
UNIVERSITY)

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

JEPPIAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600119

March - 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with “A” grade by NAAC I 12B Status by UGC I Approved by
AICTE

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of N.PRAMODH
(17SCS0310) S.SAI PRANAV TEJ ((17SCS0407) hWcarried out the project entitled
"SECURE DATA STORAGE USING CRYPTOGRAPHY" under my supervision from
November 2020 to March 2021.

Internal Guide

Dr. T. PREM JACOB, M.E., Ph.D.,

Head of the Department

Dr. S. VIGNESHWARI, M.E., Ph.D.,

Submitted for Viva voce Examination held on_____

Internal Examiner

External Examiner

DECLARATION

WE N.PRAMODH and S.SAI PRANAV TEJ hereby declare that the Project Report entitled **SECURE DATA STORAGE USING CRYPTOGRAPHY** done by us under the guidance of **Dr.T.PREM JACOB M.E., Ph.D**, is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree COMPUTER SCIENCE AND ENGINEERING.

DATE:

PLACE:CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. SASIKALA, M.E., Ph.D.**, Dean, School of Computing and **Dr. L. LAKSHMANAN, M.E., Ph.D.**, and **Dr. S. VIGNESHWARI, M.E., Ph.D.**, Heads of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.T.PREM JACOB M.E., Ph.D.**, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion our project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

Abstract

Cyphertext-policy attribute-based encryption (CP-ABE) has been a preferred encryption technology to solve the challenging problem of secure data sharing in cloud computing. The shared data files generally have the characteristic of multilevel hierarchy, particularly in the area of healthcare and the military. However, the hierarchy structure of shared files has not been explored in CP-ABE. In this paper, an efficient file encryption scheme is proposed in cloud computing. The layered access structures are integrated into a single access structure, and then, the hierarchical files are encrypted with the integrated access structure. The cipher text components related to attributes could be shared by the files. Therefore, both cipher text storage and time cost of encryption are saved. Moreover, the proposed scheme is proved to be secure under the standard assumption.

TITLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
1	INTRODUCTION	1
	1.1 CLOUD COMPUTING	
	1.2 CLOUD COMPUTING MODELS	
	1.2.1 SOFTWARE AS A SERVICE	
	1.2.2 PLATFORM AS A SERVICE	
	1.2.3 INFRASTRUCTURE AS A SERVICE	
	1.3 UNDERSTANDING PUBLIC & PRIVATE CLOUDS	2
	1.3.1 PUBLIC CLOUD	
	1.3.2 PRIVATE CLOUD	
	1.3.3 ON-PREMISE CLOUD	
	1.3.4 EXTERNALLY HOSTED PRIVATE CLOUD	
	1.3.5 HYBRID CLOUD	
	1.4 CLOUD COMPUTING BENEFITS	3
	1.4.1 REDUCED COST	
	1.4.2 INCREASED STORAGE	
	1.4.3 FLEXIBILITY	
	1.5 CLOUD COMPUTING CHALLENGES	
	1.5.1 DATA PROTECTION	
	1.5.2 DATA RECOVERY AVAILABILITY	4
	1.5.3 MANAGEMENT CAPABILITY	
	1.5.4 REGULATORY & COMPLIANCE RESTRICTIONS	
	1.6 INTRODUCTION	
	1.6.1 OBJECTIVES	5
2	LITERATURE REVIEW	6
3	3.1 EXISTING EXISTING SYSTEM	9
	3.1.1 DISADVANTAGES	
	3.2 PROPOSED SYSTEMS	
	3.2.1 ADVANTAGES	10
4	SYSTEM ARCHITECTURE	11
	4.1 INFRASTRUCTURE	
	4.2 DESIGN STRUCTURE	
	4.2.1 INPUT DESIGN	
	4.2.2 OUTPUT DESIGN	12

5	SYSTEM SPECIFICATION	13
6	SYSTEM IMPLEMENTATION	14
7	UML AND DATA FLOW DIAGRAMS	15
	7.1 DATA FLOW DIAGRAM	16
CHAPTER NO	TITLE	PAGE NO
	7.2 UML DIAGRAM	17
	7.3 USE CASE DIAGRAM	18
	7.4 CLASS DIAGRAM	19
	7.5 SEQUENCE DIAGRAM	20
	7.6 ACTIVITY DIAGRAM	21
8	SOFTWARE ENVIRONMENT	22
	8.1 JAVA TECHNOLOGIES	
	8.1.1 JAVA PLATFORM	
	8.1.2 JAVA VIRTUAL MACHINE	
	8.1.3 JAVA CLASS LIBRARY	
	8.1.4 LANGUAGES SUPPORTED BY .NET	
	SYSTEM ARCHITECTURE	25
	8.2 IMPLEMENTATION	
	8.3 SQL SERVER	26
	8.4 SQL SERVER MANAGEMENT STUDIO	
	8.4.1 TABLES	27
	8.4.1.1 DESIGN VIEW	
	8.4.1.2 DATASHEET VIEW	
	8.4.2 QUERY	28
9	CONCLUSION	29
	SCREENSHOT	
	REFERENCES	32
	APPENDIX	34

CHAPTER 1

INTRODUCTION

1.1 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.

1.2 Cloud Computing Models

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

1.2.1 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, Sales force, Microsoft, Zoho, etc.

1.2.2 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.

1.2.3 Infrastructure as a Service (IaaS):

IaaS provides basic storage and computing capabilities as standardized services over the network. Servers, storage systems, networking equipment, data center 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..

1

1.3 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.

1.3.1 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.

1.3.2 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:

1.3.3 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.

1.3.4 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.

1.3.5 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.

2

1.4 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:

1.4.1 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.

1.4.2 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.

1.4.3 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.

1.5 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:

1.5.1 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.

1.5.2 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.

1.5.3 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.

1.5.4 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.

1.6 INTRODUCTION

CLOUD computing is main purpose of data storing and retrieving data in data owners or users so data owners upload some sensitive data's in cloud so only valid user retrieve the data so we design attribute based data searching in cloud computing, become a research hot-spot due to its distinguished long-list advantages (e.g. convenience, high scalability). One of the most promising cloud computing applications is on-line data sharing, such as photo sharing in On-line Social Networks among more

than one billion users and on-line health record system. A data owner (DO) is usually willing to store large amounts of data in cloud for saving the cost on local data management. Without any data protection mechanism, cloud service provider (CSP), however, can fully gain access to all data of the user. This brings a potential security risk to the user, since CSP may compromise the data for commercial benefits. Accordingly, how to securely and efficiently share user data is one of the toughest challenges in the scenario of cloud computing.

4

Cipher text-policy attribute-based encryption (CP-ABE), has turned to be an important encryption technology to tackle the challenge of secure data sharing. In a CP-ABE, user's secret key is described by an attribute set, and cipher text is associated with an access structure. DO is allowed to define access structure over the universe of attributes. A user can decrypt a given cipher text only if his/her attribute set matches the access structure over the cipher text. Employing a CP-ABE system directly into a cloud application that may yield some open problems. Firstly, all users' secret keys need to be issued by a fully trusted key authority (KA). This brings a security risk that is known as key escrow problem. By knowing the secret key of a system user, the KA can decrypt all the user's cipher texts, which stands in total against to the will of the user. Secondly, the expressiveness of attribute set is another concern. As far as we know, most of the existing CP-ABE schemes can only describe binary state over attribute, In this paper, the weighted attribute is introduced to not only extend attribute expression from binary to arbitrary state, but also to simplify access policy. Thus, the storage cost and encryption cost for a cipher text can be relieved.

1.6.1 Objectives

Data owner encrypts and uploads the generated ciphertext to CSP. User downloads and decrypts the interested cipher text from CSP. The shared files usually have hierarchical structure. Attribute-based encryption has been attracted much more attentions since it can keep data privacy is one of feasible schemes which has much more flexibility and is more suitable for general applications.

CHAPTER 2

LITERATURE REVIEW

- 1. Title: A Security Model for the Enhancement of Data Privacy in Cloud Computing
Published in: 2019 Amity International Conference on Artificial Intelligence (AICAI)**

As we all are aware that internet acts as a depository to store cyberspace data and provide as a service to its user. cloud computing is a technology by internet, where a large amount of data being pooled by different users is stored. The data being stored comes from various organizations, individuals, and communities etc. Thus, security and privacy of data is of utmost importance to all of its users regardless of the nature of the data being stored. In this research paper the use of multiple encryption technique outlines the importance of data security and privacy protection. Also, what nature of attacks and issues might arise that may corrupt the data; therefore, it is essential to apply effective encryption methods to increase data security.

- 2. B. Mehul, D. Prayas, R. Lalit & K. Rohini, "Secure File Storage In Cloud Computing Using Hybrid Encryption Algorithm", International Journal of Computer Engineering and Applications, 9(6), 2018**

Now a day's cloud computing is used in many areas like industry, military colleges etc to storing huge amount of data. We can retrieve data from cloud on request of user. To store data on cloud we have to face many issues. To provide the solution to these issues there are n number of ways. Cryptography and steganography techniques are more popular now a day's for data security. Use of a single algorithm is not effective for high level security to data in cloud computing. In this paper we have introduced new security mechanism using symmetric key cryptography algorithm and steganography. In this proposed system AES, blowfish, RC6 and BRA algorithms are used to provide block wise security to data. All algorithm key size is 128 bit. LSB steganography technique is introduced for key information security. Key information contains which part of file is encrypted using by which algorithm and key. File is splited into eight parts. Each and every part of file is encrypted using different algorithm.

6

All parts of file are encrypted simultaneously with the help of multithreading technique. Data encryption Keys are inserted into cover image using LSB technique. Stego image is send to valid receiver using email. For file decryption purpose reverse process of encryption is applied.

3. Hybrid Algorithm for Cloud Data Security Richa Singla¹ , Richa Dutta

Published in csjournal in Volume-10, Number-2 Jan-June 2017

Cloud computing has shifted the concept of computing from local hard disk to remote data centers, providing a lot of benefits. But, in spite of the many potential blessings of cloud computing, this new prototype is also connected with additional risks which cloud consumers, people or organization's leasing cloud resources need to be concerned about. This research paper proposed a hybrid algorithm which comprises of MD5 hash with AES security algorithms to assure the clients about their data security in cloud. The hybrid algorithm almost guarantee for data security in today's scenario.

4. Secure cloud auditing over encrypted data by Sarah and Depali (2016) Published in: 2016 International Conference on Communication and Electronics Systems (ICCES)

Cloud computing is a popular technology which permits storing and accessing data over Internet instead of storing it on local machine's hard drive. Cloud enables users to store their data on cloud without fearing about its accuracy and reliability. However storing data on cloud imposes certain security challenges. Outsourcing data in cloud result in data owners losing physical control over their

data. Certain Cloud Service Providers (CSPs) may operate dishonestly with the cloud users' data, they may sneak the data from cloud and sell it to third parties in order to earn profit. Thus even though outsourcing data on cloud is inexpensive and reduces long duration storage and maintenance complexity, there is least assurance of data integrity, privacy, security and availability on cloud servers. A number of solutions have been recommended to solve the security issues in cloud. This paper focuses on the integrity verification strategy for outsourced data. The proposed scheme combines the encrypting mechanism along with integrity verification strategy. The encrypting scheme used here is public key cryptographic algorithm like ElGamal and SHA-256 hash function is employed for ensuring data storage correctness on untrusted server.

**5. Achieving cloud security using third party auditor, MD5 and identity-based encryption
Published in: 2016 International Conference on Computing, Communication and Automation
(ICCCA) by Bhale et.al**

In recent years, Cloud computing has enjoyed a tremendous rise in popularity. It provides facilities for users to use cloud applications or platform without the need of any software installation and access the data or application from anywhere, with the help of internet. The cloud computing is divided into three fragments: Application, Service and Platform. Each of fragment provides different cloud services for business and individuals. Many organizations use this cloud service without any software, they run all their applications on the cloud and use on demand services. In Cloud, consumers, store their personal files or data on cloud server and consumers use that data or files whenever needed. Many consumers store or place their personal data on the cloud, so security and privacy are very important issue in cloud. These two issues can lead to a number of security concerns related to data transmission, integrity control, access control, identity management, logging and auditing, etc. Yet, research in the area of cloud computing receiving great attention from industry, academia and government. Since these domains have numerous complex issues, there are multiple open problems for research and opportunities for making noteworthy contributions, one of these issues is the data transmission in cloud computing. This proposal is concerned to overcome the security trade-off and improve the performance of data transmission and increase the security through Third Party Auditor and Identity Based Encryption.

CHAPTER 3

3.1 Existing System

- In existing proposed symmetric encryption algorithm is used for encryption.
- Data owner outsource the encrypted file in cloud by using encryption algorithm.
- But is less efficient compared to another encryption algorithm.
- In Existing System time and cost for encryption is high.
- No any special multiple hierarchical files are used.
- Decryption system time and computation cost are very high.

3.1.1 Disadvantages:

- It provided a requirement of verifiability to the decryption of ABE, but their scheme doubled the size of the underlying ABE ciphertext and the computation costs.
- Low capacity of users.
- Possible to some person misbehave to data access

- Low efficiency.

3.2 Proposed System

- An efficient encryption scheme based on layered model of the access structure is proposed in cloud computing, which is named blowfish encryption scheme.
- The contributions of our scheme are three aspects.
- The layered model of access structure to solve the problem of multiple hierarchical files sharing. The files are encrypted with one integrated access structure.
- Secondly, we also formally prove the security of blowfish encryption that can successfully resist chosen plaintext attacks (CPA). The Key for the encrypted text is generated using RSA algorithm.
- Thirdly, the simulation results show that blowfish & RSA scheme has low storage cost and computation complexity in terms of encryption and decryption.

3.2.1 Advantages

- CP-ABE feasible schemes which has much more flexibility and is more suitable for general applications.
- Multiple hierarchical files sharing is resolved using layered model of access structure.
- In proposed system both cipher text storage and time cost of encryption are saved.
- The proposed scheme has an advantage that users can decrypt all authorization files by computing secret key once. Thus, the time cost of decryption is also saved if the user needs to decrypt multiple files.
- The computation cost of decryption can also be reduced if users need to decrypt multiple files at the same time.

CHAPTER 4**SYSTEM ARCHITECTURE****4.1 INTRODUCTION**

Design is a multi- step that focuses on data structure software architecture, procedural details, algorithm etc... and interface between modules. The design process also translate the requirements into presentation of software that can be accessed for quality before coding begins. Computer software design change continuously as new methods; better analysis and border understanding evolved. Software design is at relatively early stage in its revolution.

Therefore, software design methodology lacks the depth, flexibility and quantitative nature that are normally associated with more classical engineering disciplines. However techniques for

software designs do exist, criteria for design qualities are available and design notation can be applied.

4.2 DESIGN STRUCTURE

4.2.1 INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

What data should be given as input?

How the data should be arranged or coded?

The dialog to guide the operating personnel in providing input.

Methods for preparing input validations and steps to follow when error occur.

4.2.2 OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

The output form of an information system should accomplish one or more of the following objectives.

Convey information about past activities, current status or projections of the Future.

Signal important events, opportunities, problems, or warnings.

Trigger an action.

Confirm an action.

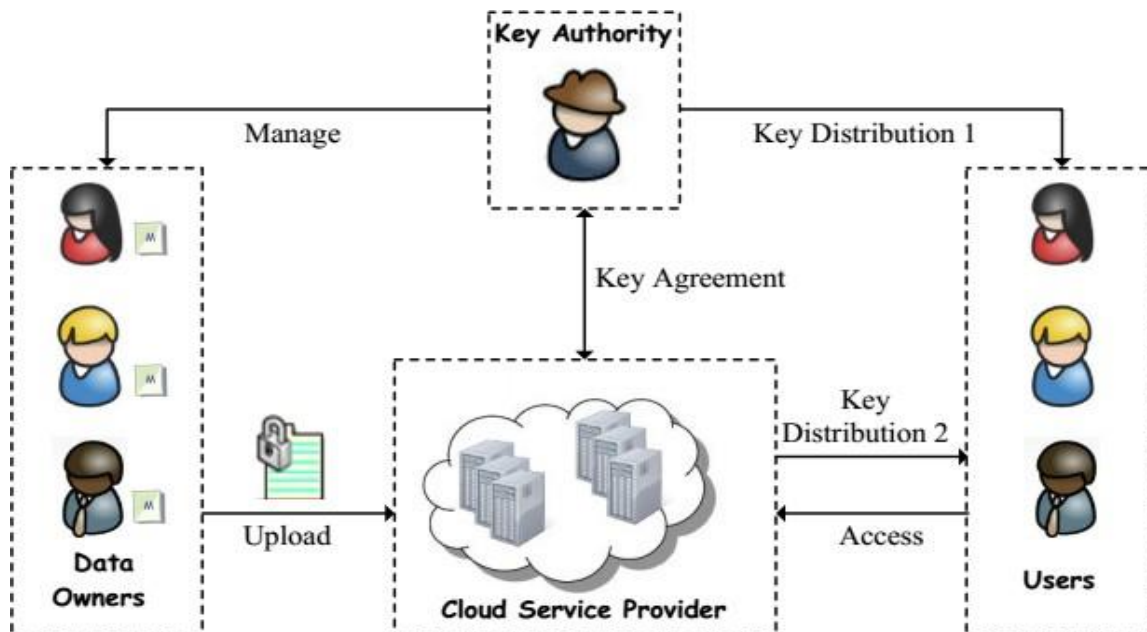


Fig 4.1 Architecture Diagram

12

CHAPTER 5

SYSTEM SPECIFICATION

5.1 Hardware Requirements:

System : Pentium IV 2.4 GHz.

Hard Disk : 40 GB.

Monitor : 15 VGA Colour.

Ram : 512 Mb.

5.2 Software Requirements:

Operating system : Windows 7/10.

Coding Language : Java

Tool : Netbeans

Database : MySql

CHAPTER 6

SYSTEM IMPLEMENTATION

6.1 PROJECT MODULES

- . User Registration
- . Uploading a file
- . Key Authority
- . Cloud
- . Download a file from the cloud

6.2 MODULE DESCRIPTION

6.2.1 User Registration

Data owner will have to register initially to get access to the profile. Data Owner will upload the file to the cloud server in the encrypted format. Random encryption key generation is happening while uploading the file to the cloud. Encrypted file will be stored on the cloud.

6.2.1 Uploading a file

The data owners will upload the file into the cloud system. They are in charge of defining access structure and executing data encryption operation. Before uploading the file, the plain text is converted into the cipher text using the encryption algorithm which is Blowfish algorithm and the key is generated using RSA algorithm. Once they got the key means they upload the data owner file in the cloud based on the attribute.

6.2.2 Attribute Authority:

Authority will have to provide the key, as per the user's key request. Every user's request will have to be raised to authority to get access key on mail. There are two complementary forms of attribute-based encryption. One is key-policy attribute-based encryption (KP-ABE) and the other is ciphertext-policy attribute-based encryption (CPABE). In a KP-ABE system, the decision of access policy is made by the key distributor instead of the encipherer, which limits the practicability and usability for the system in practical applications.

6.2.3 Cloud Server:

Cloud server will have the access to files which are uploaded by the data owner

Cloud server needs to decrypt the files available under their permission.

Furthermore data user will have to decrypt the data to access the original text by providing the respective key. File has been decrypted successfully and provided for consumer.

6.2.4 download a file from the cloud

When the user requests a file to be downloaded first the Key has to be generated. After getting the key for the respective file, the user enters the attributes of that particular file to download. The cloud provides cipher text which will be decrypted using the decryption algorithm and the shared the secret key. Finally the plain text is obtained.

CHAPTER 7

UML AND DATA FLOW DIAGRAM

System Design involves identification of classes their relationship as well as their collaboration. In objector, classes are divided into entity y classes and control classes. The Computer Aided Software Engineering (CASE) tools that are available commercially do not provide any assistance in this transition. CASE tools take advantage of Meta modeling that is helpful only after the construction of the class diagram. In the FUSION method some object-oriented approach likes Object Modeling Technique (OMT), Classes, and Responsibilities. Collaborators (CRC), etc, are used. Objector used the term” agents” to represent some of the hardware and software system. In Fusion method, there is no requirement phase, where a user will supply the initial requirement document. Any software project is worked out by both the analyst and the designer. The analyst creates the user case diagram. The designer creates the class diagram. But the designer can do this only after the analyst creates the use case diagram. Once the design is over, it is essential to decide which software is suitable for the application

1. 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.
2. 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.
3. 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.
4. 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.

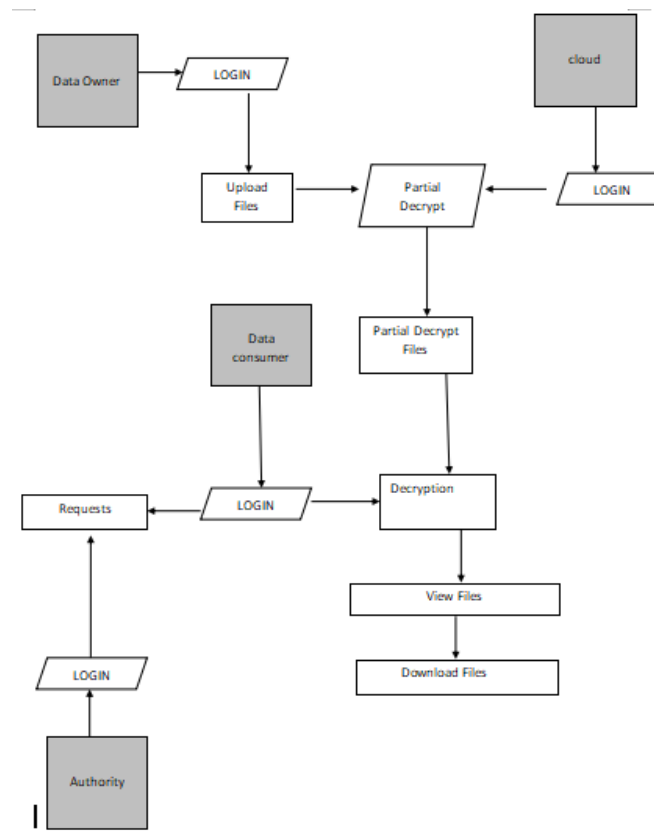


Fig 7.1 Data Flow Diagram

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.

GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

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.

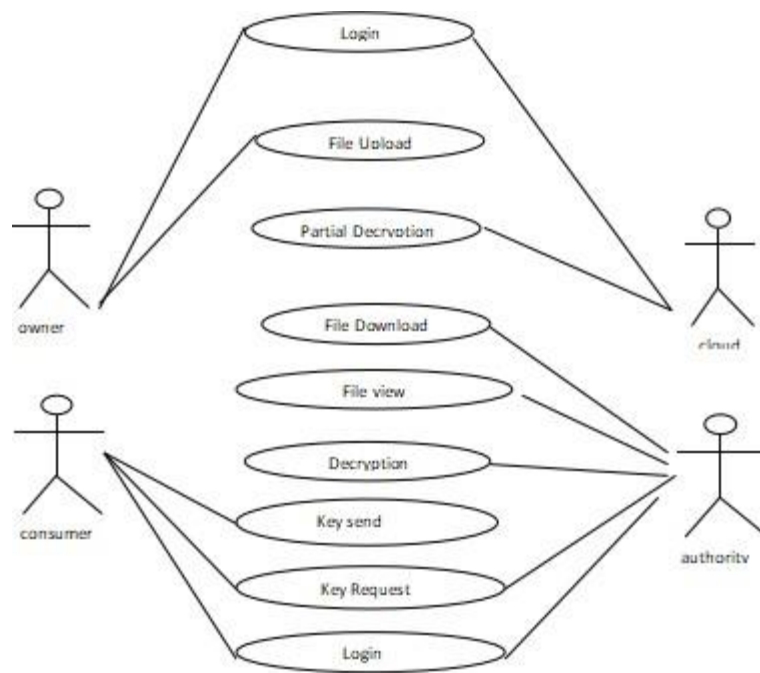


Fig 7.2 USE CASE Diagram

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.

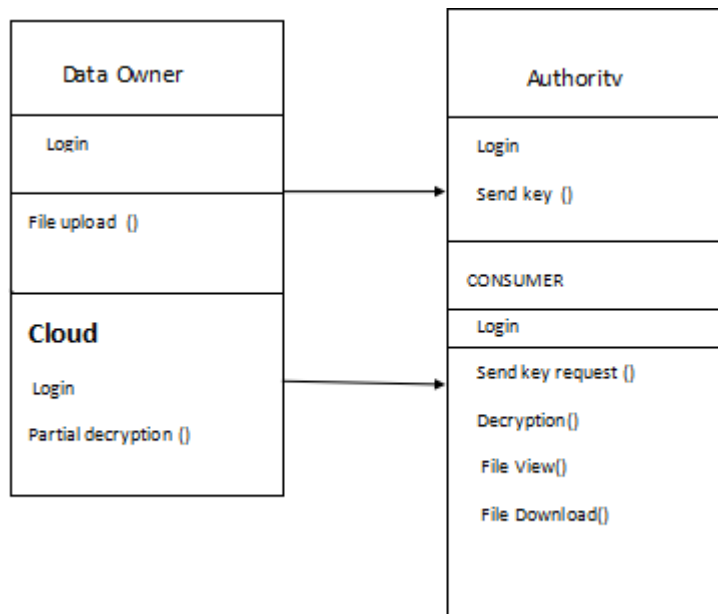


Fig 7.3 CLASS Diagram

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 diagrams.

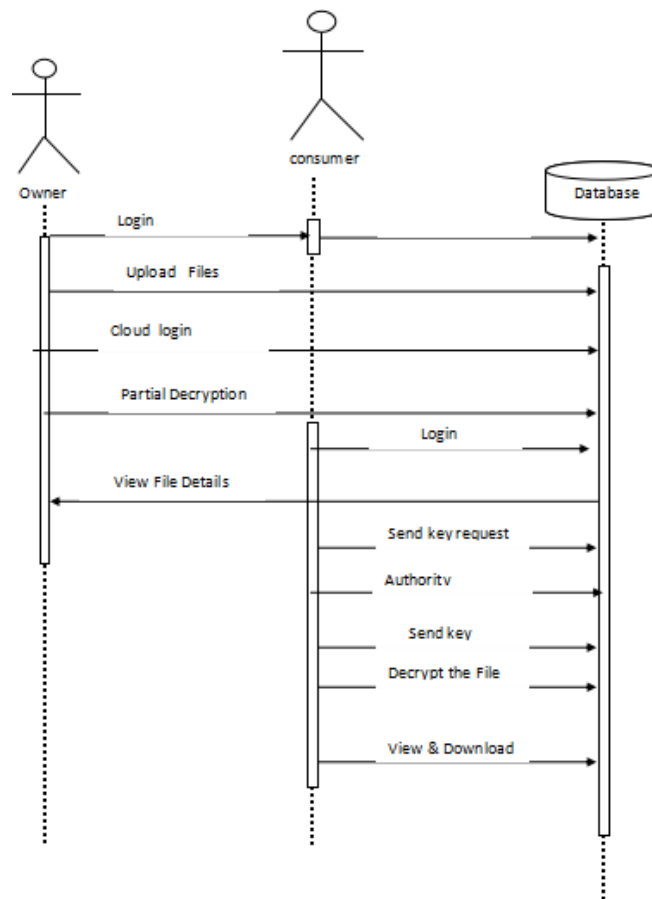


Fig 7.4 SEQUENCE Diagram

ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

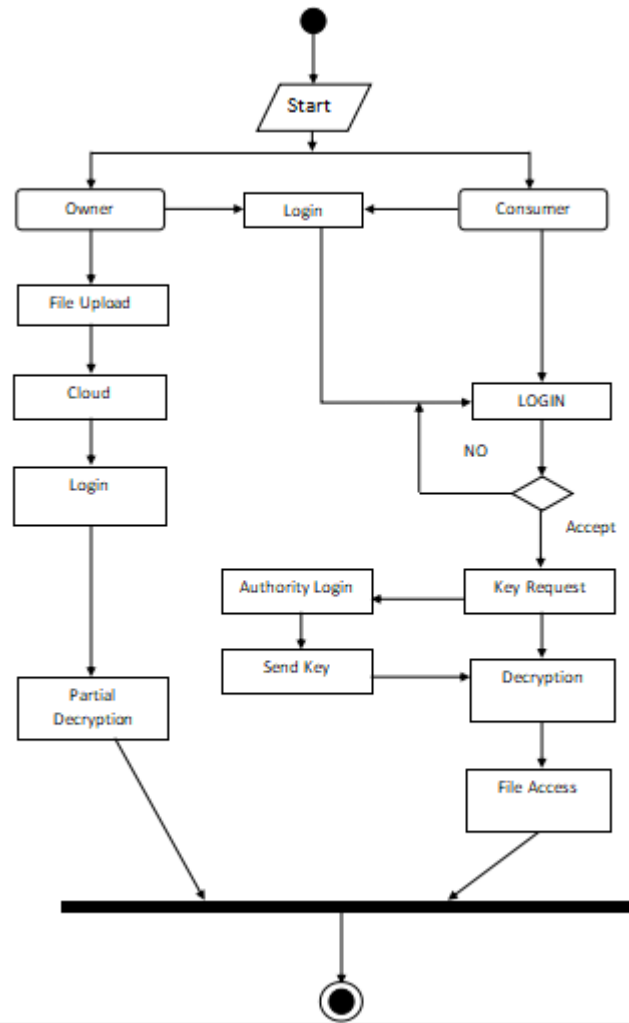


Fig 7.5 ACTIVITY Diagram

Java technology is both a programming language and a platform. Java is a general-purpose computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that code that runs on one platform does not need to be recompiled to run on another. Java applications are typically compiled to byte code that can run on any Java virtual machine (JVM) regardless of computer architecture. Java is, as of 2015, one of the most popular programming languages in use, particularly for client-server web applications, with a reported 9 million developers.

Java was originally developed by James Gosling at Sun Microsystems (which has since merged into Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++, but it has fewer low-level facilities than either of them. The original and reference implementation Java compilers, virtual machines, and class libraries were originally released by Sun under proprietary licences. As of May 2007, in compliance with the specifications of the Java Community Process, Sun relicensed most of its Java technologies under the GNU General Public License. Others have also developed alternative implementations of these Sun technologies, such as the GNU Compiler for Java (byte code compiler), GNU Class path (standard libraries), and Iced Tea-Web (browser plugin for applets).

8.1.1 JAVA PLATFORM

One design goal of Java is portability, which means that programs written for the Java platform must run similarly on any combination of hardware and operating system with adequate runtime support. This is achieved by compiling the Java language code to an intermediate representation called Java byte code, instead of directly to architecture-specific machine code. Java byte code instructions are analogous to machine code, but they are intended to be executed by a virtual machine (VM) written specifically for the host hardware. End users commonly use a Java Runtime Environment (JRE) installed on their own machine for standalone Java applications, or in a web browser for Java applets.

22

Standardized libraries provide a generic way to access host-specific features such as graphics, threading, and networking.

A major benefit of using byte code is porting. However, the overhead of interpretation means that interpreted programs almost always run more slowly than programs compiled to native executable would. Just-in-Time (JIT) compilers were introduced from an early stage that compiles byte codes to machine code during runtime. Java is platform independent. But as Java virtual machine must convert Java byte code into machine language which depends on the operating system being used, it is platform dependent.

8.1.2 JAVA VIRTUAL MACHINE

The heart of the Java platform is the concept of a "virtual machine" that executes Java byte code programs. This byte code is the same no matter what Time) compiler within the Java Virtual Machine, or JVM. The JIT compiler translates the Java byte code into native processor instructions at run-time and caches the native code in memory during execution. The use of byte code as an intermediate language permits Java programs to run on any platform that has a virtual machine available.

The use of a JIT compiler means that Java applications, after a short delay during loading and once they have "warmed up" by being all or mostly JIT-compiled, tend to run about as fast as native programs. Since JRE version 1.2, Sun's JVM implementation has included a just-in-time compiler instead of an interpreter. Although Java programs are cross-platform or platform independent, the code of the Java Virtual Machines (JVM) that executes these programs is not. Every supported operating platform has its own JVM.

8.1.3 JAVA CLASS LIBRARY

The Java Class Library (JCL) is a set of dynamically loadable libraries that Java applications can call at run time. Because the Java Platform is not dependent on a specific operating system, applications cannot rely on any of the platform-native libraries. Instead, the Java Platform provides a comprehensive set of standard class libraries, containing the functions common to modern operating systems.

JCL serves three purposes within the Java Platform:

Like other standard code libraries, they provide the programmer a well-known set of useful facilities, such as container classes and regular expression processing.

The set of classes is pretty comprehensive, providing collections, file, screen, and network I/O, threading, and so on, as well as XML and database connectivity.

The class library is subdivided into a number of sets (or namespaces), each providing distinct areas of functionality, with dependencies between the namespaces kept to a minimum.

8.1.4 LANGUAGES SUPPORTED BY .NET

The multi-language capability of the .NET Framework and Visual Studio .NET enables developers to use their existing programming skills to build all types of applications and XML Web services. The .NET framework supports new versions of Microsoft's old favorites Visual Basic and C++ (as VB.NET and Managed C++), but there are also a number of new additions to the family.

Visual Basic .NET has been updated to include many new and improved language features that make it a powerful object-oriented programming language. These features include inheritance, interfaces, and overloading, among others. Visual Basic also now supports structured exception handling, custom attributes and also supports multi-threading.

Visual Basic .NET is also CLS compliant, which means that any CLS-compliant language can use the classes, objects, and components you create in Visual Basic .NET.

Managed Extensions for C++ and attributed programming are just some of the enhancements made to the C++ language. Managed Extensions simplify the task of migrating existing C++ applications to the new .NET Framework. C# is Microsoft's new language. It's a C-style language that is essentially "C++ for Rapid Application Development". Unlike other languages, its specification is just the grammar of the language. It has no standard library of its own, and instead has been designed with the intention of using the .NET libraries as its own.

The library provides an abstract interface to tasks that would normally depend heavily on the hardware and operating system, such as network access and file access.

Some underlying platforms may not support all of the features a Java application expects. In these cases, the library implementation can either emulate those features or provide a consistent way to check for the presence of a specific feature.

Architecture Diagram

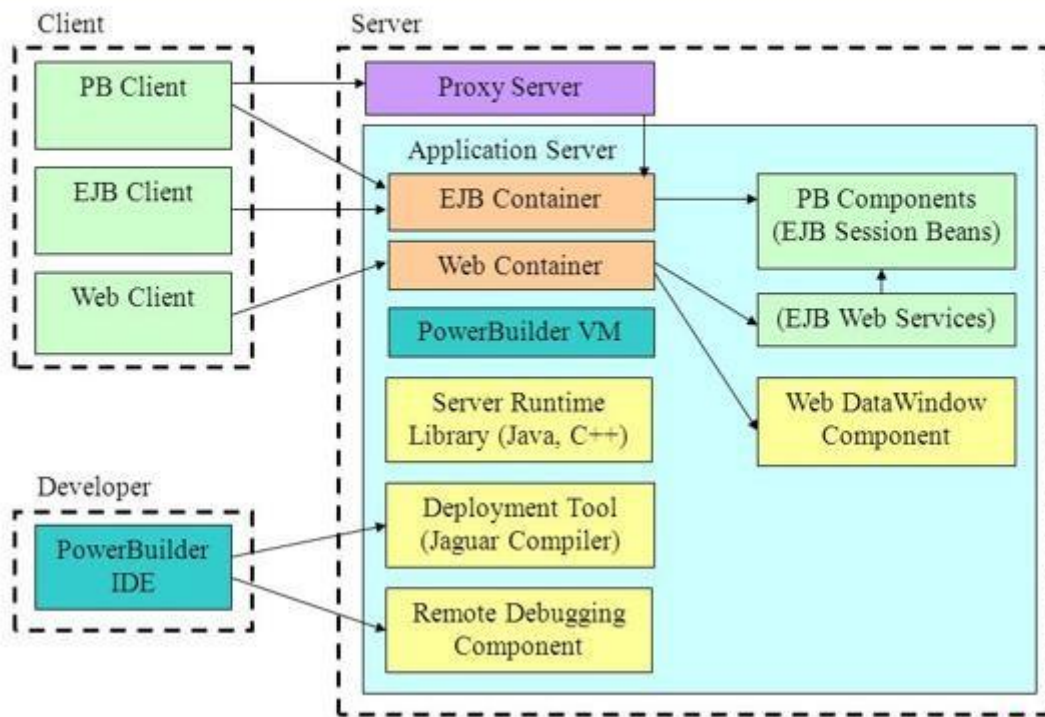


Fig 8.1 ARCHITECTURE Diagram

8.2 IMPLEMENTATIONS

Oracle Corporation is the current owner of the official implementation of the Java SE platform, following their acquisition of Sun Microsystems on January 27, 2010. This implementation is based on the original implementation of Java by Sun. The Oracle

Implementation is available for Microsoft Windows, Mac OS X, Linux and Solaris. Because Java lacks any formal standardization recognized by Ecma International, ISO/IEC, ANSI, or other third-party standards organization, the Oracle implementation is the de facto standard.

The Oracle implementation is packaged into two different distributions: The Java Runtime Environment (JRE) which contains the parts of the Java SE platform required to run Java programs and is intended

for end users, and the Java Development Kit (JDK), which is intended for software developers and includes development tools such as the Java compiler, Java doc, Jar, and a debugger. Open JDK is another notable Java SE implementation that is licensed under the GNU GPL.

25

The implementation started when Sun began releasing the Java source code under the GPL. As of Java SE 7, Open JDK is the official Java reference implementation. The goal of Java is to make all implementations of Java compatible. Historically, Sun's trademark license for usage of the Java brand insists that all implementations be "compatible". This resulted in a legal dispute with Microsoft after Sun claimed that the Microsoft implementation did not support RMI or JNI and had added platform-specific features of their own.

8.3 SQL SERVER

Microsoft is the vendor of SQL Server. We have different editions of SQL Server, where SQL Server Express is free to download and use. SQL Server uses T-SQL (Transact-SQL). T-SQL is Microsoft's proprietary extension to SQL. TSQL is very similar to standard SQL, but in addition it supports some extra functionality, built in functions, etc.

T-SQL expands on the SQL standard to include procedural programming, local variables, and various support functions for string processing, date processing, mathematics, etc. SQL Server consists of a Database Engine and a Management Studio (and lots of other stuff which we will not mention here). The Database engine has no graphical interface - it is just a service running in the background of your computer (preferable on the server). The Management Studio is graphical tool for configuring and viewing the information in the database. It can be installed on the server or on the client (or both).

SQL Server Management Studio can also be used to create a new database, alter any existing database schema by adding or modifying tables and indexes, or analyze performance. It includes the query windows which provide a GUI based interface to write and execute queries.

8.4 SQL SERVER MANAGEMENT STUDIO

SQL Server Management Studio is a GUI tool included with SQL Server for configuring, managing, and administering all components within Microsoft SQL Server. The tool includes both script editors and graphical tools that work with objects and features of the server. As mentioned earlier,

version of SQL Server Management Studio is also available for SQL Server Express Edition, for which it is known as SQL Server Management Studio Express.

A central feature of SQL Server Management Studio is the Object Explorer, which allows the user to browse, select, and act upon any of the objects within the server. It can be used to visually observe and analyze query plans and optimize the database performance, among others.

26

SQL Server Management Studio can also be used to create a new database, alter any existing database schema by adding or modifying tables and indexes, or analyze performance. It includes the query windows which provide a GUI based interface to write and execute queries.

When creating SQL commands and queries, the “Query Editor” (select “New Query” from the Toolbar) is used (shown in the figure above). With SQL and the “Query Editor” we can do almost everything with code, but sometimes it is also a good idea to use the different Designer tools in SQL to help us do the work without coding (so much).

Create a new Database

It is quite simple to create a new database in Microsoft SQL Server. Just right-click on the “Databases” node and select “New Database...” The OLAP Services feature available in SQL Server version 7.0 is now called SQL Server 2000 Analysis Services. The term OLAP Services has been replaced with the term Analysis Services. Analysis Services also includes a new data mining component. The Repository component available in SQL Server version 7.0 is now called Microsoft SQL Server 2000 Meta Data Services. References to the component now use the term Meta Data Services.

The term repository is used only in reference to the repository engine within Meta Data Services SQL-SERVER database consist of six type of objects,

They are,

1. TABLE
2. QUERY
3. FORM
4. REPORT
5. MACRO

8.4.1 Table

A database is a collection of data about a specific topic.

We can work with a table in two types,

8.4.1.1 Design View

To build or modify the structure of a table we work in the table design view. We can specify what kind of data will be hold.

8.4.1.2 Datasheet View

To add, edit or analyses the data itself we work in tables data-sheet view mode.

8.4.2 Query

A query is a question that has to be asked the data. Access gathers data that answers the question from one or more table. The data that make up the answer is either dynaset (if you edit it) or a snapshot (it cannot be edited).Each time we run query, we get latest information in the dynaset. Access either displays the dynaset or snapshot for us to view or perform an action on it, such as deleting or updating.

CHAPTER 9

CONCLUSION

Data confidentiality is very important during transmission of information from client to server. Privacy is achieved with creating password of our file or generate code word of information (which ready to transmit), that easily understandable by those users who wants to share it. It is possible through information encryption. This paper presented explains encryption techniques available and used, which have strengths in term of security and computational performance. This hybrid methodology is more secure because this is a robust technology which provides better security than others. Concept of compression of plain text first and then encryption is good to provide more security, and it is best in term of file size and encryption execution time.

FUTURE ENHANCEMENT

Issues such as scalability in key management, flexible access and efficient user revocation, have remained the most important challenges toward achieving fine-grained, cryptographically enforced data access control. For improving the limitations of the above technique, we proposed the algorithm is crucial in the second stage, the randomly generated key provides more security than the conventional encryption system. In future we can enhance the project with multimedia like image, video.

SCREENSHOT:

Secure Cloud Storage Cryptography in Cloud Computing

Home

Data Owner

Data Consumer

Authority

Registration

Cloud

Abstract

In the cloud, for achieving access control and keeping data confidential, the data owners could adopt attribute-based encryption to encrypt the stored data. Users with limited computing power are however more likely to delegate the task of the decryption to the cloud servers to reduce the computing cost. As a result, attribute-based encryption with delegation emerges. Still, there are caveats and questions remaining in the previous relevant works. For instance, during the delegation, the cloud servers could tamper or replace the delegated ciphertext and respond a forged computing result with malicious intent. They may also cheat the eligible users by responding them that they are ineligible for the purpose of cost saving. Furthermore, during the encryption, the access policies may not be flexible enough as well. Since policy for general circuits enables to achieve the strongest form of access control, a construction for realizing circuit ciphertext-policy attribute-based hybrid encryption with verifiable delegation has been considered in our work. In such a system, combined with verifiable computation and encrypt-then-mac mechanism, the data confidentiality, the fine-grained access control and the correctness of the delegated computing results are well guaranteed at the same time. Besides, our scheme achieves security against chosen-plaintext attacks under the k -multilinear Decisional Diffie-Hellman assumption. Moreover, an extensive simulation campaign confirms the feasibility and efficiency of the proposed solution.

29

Secure Cloud Storage Cryptography in Cloud Computing

Home

Authority

Data Owner

Data Consumer

Owner Login

User id:

Password:

Login

FE34589D6DA687A22
9E78D8C885E1G 43
44HK4 1D1F917C14

Secure Cloud Storage Cryptography in Cloud Computing

Home

Authority

Data Owner

Data Consumer

Authority Login

User id:

Password:

Login

7E34589D6D45A7422
9E78D8C88E1G1 K1
44HK4 1D1F017CE4

Secure Cloud Storage Cryptography in Cloud Computing

Home

Data Owner

Data Consumer

Data Owner Registration:

User id:

Password:

Address:

Age:

Phone:

Email-Id:

Submit

Secure Cloud Storage Cryptography in Cloud Computing

Home

Authority

Data Owner

Data Consumer

Cloud Login

User id:

Password:

Login

REFERENCES

- [1] M.Z. Abdullah & Z.J Khaleefah, "Design and implement of a hybrid cryptography textual system", International Conference on Engineering and Technology , pp. 1-6. doi: 10.1109/ICEngTechnol.2017.8308141 , 2017
- [2] D. Bhole, A. Mote & R. Patil, "A new security protocol using hybrid cryptography algorithms", International Journal of Computer Sciences and Engineering 4(2), pp. 18-22, 2016.
- [3] B. Bindu, K. Lovejeet & L. Pawan, "Secure File Storage In Cloud Computing Using Hybrid Cryptography Algorithm", International Journal of Advanced Research in Computer Science 9(2), 2017.
- [4] C. Biswas, U. D. Gupta and M. M. Haque, "An Efficient Algorithm for Confidentiality, Integrity and Authentication Using Hybrid Cryptography and Steganography", International Conference on Electrical, Computer and Communication Engineering, pp. 1-5, 2019.
- [5] S. Bulusu., & K. Sudia. A Study on Cloud Computing Security Challenges. 2013
- [6] M.M Chauhan. "An implemented of hybrid cryptography using elliptic curve cryptosystem (ECC) and MD5". 2016 International Conference on Inventive Computation Technologies (ICICT),Coimbatore, 1-6. doi: 10.1109/INVENTIVE.2016.7830092 doi:10.1109/ECACE.2019.8679136, 2016.
- [7] A. Chauhan and J. Gupta, "A novel technique of cloud security based on hybrid encryption by Blowfish and MD5," 2017 4th International Conference on Signal Processing, Computing and Control (ISPCC), Solan, 2017, pp. 349-355. doi: 10.1109/ISPCC.2017.8269702
- [8] S. Garima, and S. Naveen. "Triple Security of Data in Cloud computing". International Journal of Computer Science and Information Technologies. 5(4), 2014
- [9] M. Harini., K. P.Gowri, C. Pavithra, & M.P Selvarani. "A novel security mechanism using hybrid cryptography algorithms". IEEE International Conference on Electrical, Instrumentation and Communication Engineering. doi:10.1109/iceice.2017.8191910, 2017
- [10] N Jirwan, A Singh & S Vijay, "Review and Analysis of Cryptography Techniques", Inter. J.Sci. Engineer. Res. 4(3): 1-6, 2019
- [11] D Kahn . The Codebreakers - The Story of Secret Writing: New York Times, 1967.
- [12] P Kumar & S. B. Ran'a. "Development of modified AES algorithm for data security". Optik-Int, J. Light Electron Opt., 127 (4), pp. 2341-2345, 2016

[13] S. Kuswaha, S. Waghmare & P. Choudhary. "Data Transmission using AES-RSA Based Hybrid Security Algorithms", International Journal on Recent and Innovation Trends in Computing and Communication (3)4, 2015.

32

[14] B. Mehul, D. Prayas, R. Lalit & K. Rohini, "Secure File Storage In Cloud Computing Using Hybrid Encryption Algorithm", International Journal of Computer Engineering and Applications,9(6),2018

[15] M. Mondal & K.S Ray. "Review on DNA Cryptography". arXiv preprint arXiv:1904.05528. Retrieved from <https://arxiv.org/abs/1904.05528?context=cs>, 2019.

[16] P. V. Maitri and A. Verma, "Secure file storage in cloud computing using hybrid cryptography algorithm", International Conference on Wireless Communications, Signal Processing and Networking pp. 1635-1638. doi:10.1109/WiSPNET.2016.7566416, 2016.

[17] B. Pooja. Optimization of Cryptography Algorithms in Cloud Computing. International Journal of Computer Trends and Technology,6(2), 2017.

[18] C. Ponnusamy and P. Deepalakshmi, "Design of Secure Storage for Health-care Cloud using Hybrid Cryptography", 2018 Second International Conference on Inventive Communication and Computational Technologies, April 2018.

[19] Y. Sharma, H. Gupta & S.K Khatri, "A Security Model for the Enhancement of Data Privacy in Cloud Computing", Amity International Conference on Artificial Intelligence pp.898-902. doi: 10.1109/AICAI.2019.8701398, 2019.

[20] V.K Soman & V. Natarajan, "An enhanced hybrid data security algorithm for cloud", International Conference on Networks & Advances in Computational Technologies, pp. 416-419. doi: 10.1109/NETACT.2017.8076807, 2017.

APPENDIX

SOURCE CODE

Index.jsp

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <meta name="description" content="" />
    <meta name="keywords" content="" />
    <title>Cipher_Text_Policy</title>
    <!--      <link      href="http://fonts.googleapis.com/css?family=Open+Sans"      rel="stylesheet"
type="text/css" />-->
    <link rel="stylesheet" type="text/css" href="style.css" />
  </head>
  <body>
    <div id="wrapper">
      <div id="header">
        <div id="logo">
          <h3>Secure Cloud Storage Cryptography in Cloud Computing</h3>
        </div>
        <div id="menu">
          <ul>
            <p class="first current_page_item"><a href="index.jsp">Home</a></p>
            <p><a href="ownerLogin.jsp">Data Owner</a></p>
            <p><a href="consumerLogin.jsp">Data Consumer</a></p>
            <p><a href="domainLogin.jsp">Authority</a></p>
            <p><a href="registration.jsp">Registration</a></p>
            <p><a href="cloudLogin.jsp">Cloud</a></p>

          </ul>
          <br class="clearfix" />
        </div>
      </div>
    </div>

```

```

<div style="margin-top: 50px;margin-left: 100px;text-align: justify">
    </img>
</div>
<div id="inner">
    <%if (request.getParameter("msg") != null) {
        out.println(request.getParameter("msg"));
    }%>
</div>

</div>

</body>
</html>

```

34

uploadProcess.jsp

```

<%@page import="com.util.Utilities"%>
<%@page import="org.apache.commons.io.IOUtils"%>
<%@page import="org.apache.commons.fileupload.FileItem"%>
<%@page import="java.util.List"%>
<%@page import="java.util.Iterator"%>
<%@page import="org.apache.commons.fileupload.servlet.ServletFileUpload"%>
<%@page import="org.apache.commons.fileupload.disk.DiskFileItemFactory"%>
<%@page import="java.io.File"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <meta http-equiv="content-type" content="text/html; charset=utf-8" />
        <meta name="description" content="" />
        <meta name="keywords" content="" />
        <title>Cipher_Text</title>
        <!--
            <link href="http://fonts.googleapis.com/css?family=Open+Sans" rel="stylesheet"
type="text/css" />-->
        <link rel="stylesheet" type="text/css" href="style.css" />
    </head>
    <%
        String TMP_DIR_PATH = "c:\\tmp";
        File tmpDir = new File(TMP_DIR_PATH);
        String[] lines = null;
        int count = 0;
        String size = null;
        String key = null;
        String userid = null;
        String filename = null;
        String nameid="";
        String master="";
        String sec="";
        String pub="";
        try {
            DiskFileItemFactory fileItemFactory = new DiskFileItemFactory();

```

```

fileItemFactory.setSizeThreshold(1 * 1024 * 1024); //1 MB
fileItemFactory.setRepository(tmpDir);
ServletFileUpload uploadHandler = new ServletFileUpload(fileItemFactory);
List items = uploadHandler.parseRequest(request);
Iterator itr = items.iterator();
int a1;
int a2;

while (itr.hasNext()) {
    FileItem item = (FileItem) itr.next();
    if (item.isFormField()) {
        // if (item.getFieldName().equals("key_")) {
        //     key = item.getString();
        // }
    } else {
        String str = IOUtils.toString(item.getInputStream(), "UTF-8");
        filename = item.getName();
        //item.getInputStream().
        lines = str.split("\r\n|\r|\n");
        nameid= (String)session.getAttribute("userid");
        sec=Utilities.stringtoByte(filename+filename.length());
        pub=Utilities.stringtoByte(nameid);
        master=Utilities.stringtoByte(pub+sec);
        // a1=Utilities.stringtoByte(filename);
        // a2=Utilities.stringtoByte(filename.length()+""");
        // master= Integer.toBinaryString((a1&a2));
        // size = lines.length + """;
    }
}
} catch (Exception e) {
    e.printStackTrace();
}
%>
<body>
<div id="wrapper">
<div id="header">
<div id="logo">
<h3>Secure Cloud Storage Cryptography in Cloud Computing</h3>
</div>
<div id="menu">
<ul>
<p class="first current_page_item"><a href="#">Welcome Data Owner</a></p>
<p><a href="index.jsp">Log Out</a></p>
<br class="clearfix" />
</ul>
</div>
</div>
<div id="inner">
<div id="splash">
<%if (request.getParameter("msg") != null) {
    out.println(request.getParameter("msg"));

```

```

    }%>
    <span style="font-size: 22px;margin-left: 350px">Key Generation</span><br>
    <form method="post" action="UploadData">
    <input type="hidden" id="fileName" name="fileName" value="<%=filename%>">
    <table>
        <tr><td>Data Owners ID</td><td><input type="text" id="publicKey" name="publicKey" size="50"
value="<%=nameid%>"/></td></tr>
        <!--<tr><td>File Attribute 1:</td><td><input type="text" id="att1" name="att1" size="50"
value="<%=Utilities.stringtoByte(filename)%>"/></td></tr-->
        <tr><td>Random Encryption Key:</td><td><input type="text" id="att2" name="att2" size="50"
value="<%=Utilities.stringtoByte(lines.length+"")%>"/></td></tr>
        <tr><td>One Time Verified key:</td><td><input type="text" id="master" name="master" size="50"
value="<%= master%>"/></td></tr>
        <tr><td>Symmetric Key:</td><td><input type="text" id="secret" name="secret" size="50"
value="<%=sec%>"/></td></tr>
        <tr><td>File Content:</td><td><textarea id="block1" name="block1" value="" size="50"
style="width: 300px;height: 100px;">
            <%
                for (int i = count; i < lines.length; i++) {
                    if (lines[i].trim() == null || lines[i].trim() == "") {
                        continue;
                    }
                    out.println(lines[i]);
                }
            %>
        </textarea></td></tr>
        <tr><td><input type="submit" value="Upload"/></td></tr>
    </table>
    </form>
</div>

</div>

</div>

</body>
</html>

```

Decryptaction.jsp

```

<%@page import="util.Rsa"%>
<%@page import="com.util.DbConnector"%>
<%--<%@page import="com.util.Rsa"%>--%>
<%@page import="java.sql.Connection"%>
<%@page import="java.sql.Statement"%>
<%@page import="java.sql.ResultSet"%>
<%
// Connection conn = null;
// Statement smt = null;
// ResultSet rs = null;
String []array = request.getQueryString().split(",");
String msg= array[0];

```

```

String fileName=array[1];
// String receiver=array[2];
// String fileName=array[3];
// String user=array[4];
// String role=array[5];
// String partialKey = null;
// try{
//     conn = DbConnector.getConnection();
//     smt = conn.createStatement();
//     rs = smt.executeQuery("select pubkey from keydetails where key_owner='"+receiver+"' and requester
='"+sender+"'");
//     while(rs.next()){
//         partialKey = rs.getString("key_");
//         System.out.println("partial encr key is "+ partialKey);
//     }
// }catch(Exception ex){
//     ex.printStackTrace();
// }
// }
try{
    Connection con = DbConnector.getConnection();
    Statement smt1 = con.createStatement();
    int i = smt1.executeUpdate("update upload set data_='"+new Rsa().decrypt(msg)+"', status = 'YES' where
filename='"+fileName+"'and status ='NO'");
    if(i!=0){
        response.sendRedirect("partialdecrypt.jsp?success");
    }else{
        response.sendRedirect("partialdecrypt.jsp?failed");
    }
}catch(Exception ex){
    ex.printStackTrace();
}
}
%>

```

DbConnector.java

```

package com.util;

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

import java.sql.*;

/**
 *
 * @author sabari
 */
public class DbConnector {

```

```

// System.out.println("MySQL Connect Example.");

public static Connection getConnection() {
    Connection conn = null;
    String url = "jdbc:mysql://localhost:3307/";
    String dbName = "abevod";
    String driver = "com.mysql.jdbc.Driver";
    String userName = "root";
    String password = "root";
    try {
        Class.forName(driver).newInstance();
        conn = DriverManager.getConnection(url + dbName, userName, password);
        System.out.println("Connected to the database");
        //conn.close();
        //System.out.println("Disconnected from database");
    } catch (Exception e) {
        e.printStackTrace();
    }
    return conn;
}

public static void main(String arg[]){
    DbConnector.getConnection();
}
}

```