

COLLABORATIVE DATA SHARING AND SECURE ACCESS OVER SHARED DATA

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

By

GUNNAM PAVAN VAMSI (37110255)

GUNNAM HEMANTH KUMAR REDDY (37110254)



**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

JEPPIAAR NAGAR, RAJIV GANDHI SALAI,

CHENNAI - 600 119

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 Mr. **GUNNAM PAVAN VAMSI (Reg.No:37110255)** who carried out the project entitled " **COLLABORATIVE DATA SHARING AND SECURE ACCESS OVER SHARED DATA**" under my supervision from November 2020 to March 2021.

Internal Guide
Dr.S.Prince Mary,M.E.,Ph.D.,

Head of the Department

Submitted for Viva voce Examination held on_____

Internal Examiner

External Examiner

DECLARATION

I **GUNNAM PAVAN VAMSI (37110255)** hereby declare that the Project Report entitled **COLLABORATIVE DATA SHARING AND SECURE ACCESS OVER SHARED DATA** done by me under the guidance of **Dr.S.Prince Mary,M.E.,Ph.D.**, at **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** is submitted in partial fulfillment of the requirements for the award of Bachelor ofEngineering / Technology degree In **CSE**

DATE:

PLACE:

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.S.Prince Mary, M.E.,Ph.D.**, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my 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

The ability to enforce and adaptive access controls on centralized hosted information while at the same time maintaining database privacy is a clear goal for many users and organizations. Designing efficient dynamic access control system in the data storage is still challenging. Preventing and Maintaining the access policy on the data is another one concern in the storage methodologies. In this project we enable an efficient dynamic access control over the centrally stored data for maintaining the access permissions for the entity. To prevent the overhead of protecting the access control of the data we introduced a security parameters and revocation of the control. It deals with the problem of performing security in central data after the revocation of the particular set of roles or set of users.

LIST OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	i
	LIST OF FIGURES	viii
	LIST OF ABBREVIATIONS	x
1	INTRODUCTION	1
	1.1 DATA SHARING	1
	1.2 BENEFITS OF DATA SHARING	1
	1.3 SECURE ACCESS CONTROL	2
	1.4 WHY IS ACCESS CONTROL IMPORTANT	2
	1.5 TYPES OF ACCESS CONTROLS	3
	1.6 IMPLEMENTING ACCESS CONTROL	4
2	LITERATURE SURVEY	5
3	3.1 SCOPE OF THE PROJECT	9
	3.2 OBJECTIVE	9
	3.3 INPUT AND OUTPUT	9
	3.4 EXISTING STSTEM	10
	3.5 PROPOSED SYSTEM	10
	3.5.1 ADVANTAGES OF PROPOSED SYSTEM	11
	3.6 HARDWARE AND SOFTWARE REQUIREMENTS	11
4	MATERIALS AND METHODS	12
	4.1 SYSTEM ARCHITECTURE	12
	4.2 IMPLEMENTATION OF AES ALGORITHM	13
	4.3 E-R DIAGRAMS	14

4.4	INTRODUCTION OF DFD'S	17
4.4.1	DFD SYMBOLS	18
4.4.2	CONSTRUCTING A DFD	18
4.4.3	SAILENT FEATURES OF DFD'S	19
4.4.4	DATA FLOW DIAGRAMS	19
4.5	INTRODUCTION TO USE CASE DIAGRAM	21
4.5.1	USE CASE DIAGRAMS	22
4.6	INTRODUCTION OF CLASS DIAGRAM	24
4.6.1	PURPOSE OF CLASS DIAGRAM	24
4.6.2	CLASS DIAGRAM	24
5	RESULTS AND DISCUSSION	25
5.1	OUTPUT SCREENSHOTS	25
5.2	DATABASE SCREENSHOTS	28
6	CONCLUSION	30
	REFERENCES	31
	APPENDIX	32

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NO
4.1	SYSTEM ARCHITECTURE	12
4.2	DATA OWNER E-R DIAGRAM	15
4.3	DATA USER E-R DIAGRAM	15
4.4	ADMIN E-R DIAGRAM	16
4.5	CLOUD E-R DIAGRAM	16
4.6	DATA OWNER DFD	19
4.7	DATA USER DFD	20
4.8	ADMIN DFD	20
4.9	CLOUD DFD	21
4.10	DATA OWNER USE CASE DIAGRAM	22
4.11	DATA USER USE CASE DIAGRAM	22
4.12	ADMIN USE CASE DIAGRAM	23
4.13	CLOUD USE CASE DIAGRAM	23
4.14	CLASS DIAGRAM	24
5.1	HOME PAGE	25
5.2	ENCRYPTION KEY MAIL	25
5.3	FILE WRITE STATUS	26
5.4	ENCRYPTED FILES IN CLOUD	26

5.5	DECRYPTIONKEY MAIL	27
5.6	DECRYPTED FILE	27
5.7	DATA USERS REGISTRATION DATABASE	28
5.8	DATA OWNERS REGISTRATION DATABASE	28
5.9	UPLOADED FILES DATABASE	29

LIST OF ABBREVIATIONS

AES	ADVANCED ENCRYPTION STANDARD
UML	UNIFIED MODELING LANGUAGE
DFD	DATA FLOW DIAGRAM
ER	ENTITY RELATIONSHIP
DAC	DISCRETIONARY ACCESS CONTROL
PII	PERSONALLY IDENTIFIABLE INFORMATION

CHAPTER 1

INTRODUCTION

The massive scale, the speed of ingesting and processing, and the characteristics of the data that must be dealt with at each stage of the process present significant new challenges when designing solutions. The goal of most data sharing systems is to surface insights and connections between the sharable entities. Depending upon the idea of the information being broke down, there are lawful limitations hindering such organizations to re-appropriate the capacity and control of a portion of the datasets, particularly while including individual data. To better address the data sharing within an organization the methodology provided in this project will be efficient. The access control of the data will be given according to the designation of the data users which is registered on the system. The access control mechanism is introduced for the theme that only the authorized user of the particular data must the access based upon the accepted or revoked positions.

1.1 DATA SHARING

The ability to share the same data resource with multiple applications or users. It implies that the data are stored in one or more servers in the network and that there is some software locking mechanism that prevents the same set of data from being changed by two people at the same time. Data sharing is a primary feature of a database management system (DBMS)

Data sharing is not a new thing. Individuals, organisations, and governments have been sharing information since before computers and networks existed. Over the last decade, however, progress in digital literacy and skills, technology, and the adaptation of legislative frameworks to the digital space have enabled data to be shared faster and at an unprecedented scale

1.2 BENEFITS OF DATA SHARING

- Organisations and public authorities can share more data between them, and can do it in a way that is secure, fair, lawful and respectful of the rights of those that the data concerns.

- Combining data from different sources can increase the performance and value of services by orders of magnitude. It enables better research and development, and the delivery of better products.
- Sharing data enables unprecedented collaboration and data-driven decision-making, informing policy and amplifying social impact.

1.3 SECURE ACCESS CONTROL

Access control is a security technique that regulates who or what can view or use resources in a computing environment. It is a fundamental concept in security that minimizes risk to the business or organization.

There are two types of access control: physical and logical. Physical access control limits access to campuses, buildings, rooms and physical IT assets. Logical access control limits connections to computer networks, system files and data.

To secure a facility, organizations use electronic access control systems that rely on user credentials, access card readers, auditing and reports to track employee access to restricted business locations and proprietary areas, such as data centers. Some of these systems incorporate access control panels to restrict entry to rooms and buildings, as well as alarms and lockdown capabilities, to prevent unauthorized access or operations.

Access control systems perform identification authentication and authorization of users and entities by evaluating required login credentials that can include passwords, personal identification numbers (PINs), biometric scans, security tokens or other authentication factors. Multifactor authentication (MFA), which requires two or more authentication factors, is often an important part of a layered defense to protect access control systems.

1.4 WHY IS ACCESS CONTROL IMPORTANT?

The goal of access control is to minimize the security risk of unauthorized access to physical and logical systems. Access control is a fundamental component of security compliance programs that ensures security technology and access

control policies are in place to protect confidential information, such as customer data. Most organizations have infrastructure and procedures that limit access to networks, computer systems, applications, files and sensitive data, such as personally identifiable information (PII) and intellectual property.

Access control systems are complex and can be challenging to manage in dynamic IT environments that involve on-premises systems and cloud services. After some high-profile breaches, technology vendors have shifted away from single sign-on (SSO) systems to unified access management, which offers access controls for on-premises and cloud environments.

1.5 TYPES OF ACCESS CONTROL

The main models of access control are the following:

Discretionary access control (DAC)

This is an access control method in which owners or administrators of the protected system, data or resource set the policies defining who or what is authorized to access the resource. Many of these systems enable administrators to limit the propagation of access rights. A common criticism of DAC systems is a lack of centralized control.

Mandatory access control (MAC)

This is a security model in which access rights are regulated by a central authority based on multiple levels of security. Often used in government and military environments, classifications are assigned to system resources and the operating system (OS) or security kernel. It grants or denies access to those resource objects based on the information security clearance of the user or device. For example, Security Enhanced Linux (SE Linux) is an implementation of MAC on the Linux OS.

Role-based access control (RBAC)

This is a widely used access control mechanism that restricts access to computer resources based on individuals or groups with defined business functions -- e.g., executive level, engineer level 1, etc. -- rather than the identities of individual users. The role-based security model relies on a complex structure of role

assignments, role authorizations and role permissions developed using role engineering to regulate employee access to systems. RBAC systems can be used to enforce MAC and DAC frameworks.

Rule-based access control

This is a security model in which the system administrator defines the rules that govern access to resource objects. Often, these rules are based on conditions, such as time of day or location. It is not uncommon to use some form of both rule-based access control and RBAC to enforce access policies and procedures

Attribute-based access control (ABAC)

This is a methodology that manages access rights by evaluating a set of rules, policies and relationships using the attributes of users, systems and environmental conditions.

We used Discretionary access control (DAC) in our project

1.6 IMPLEMENTING ACCESS CONTROL

Access control is a process that is integrated into an organization's IT environment. It can involve identity management and access management systems. These systems provide access control software, a user database, and management tools for access control policies, auditing and enforcement.

When a user is added to an access management system, system administrators use an automated provisioning system to set up permissions based on access control frameworks, job responsibilities and workflows.

The best practice of least privilege restricts access to only resources that employees require to perform their immediate job functions.

CHAPTER 2

LITERATURE SURVEY

1 Title : CLOUD DATA SECURITY USING AES ALGORITHM

Author : Prof. S. Delfin, Rachana Sai. B, Meghana J.V, Kundana Lakshmi. Y,

Sushmita Sharma Popular security experts decrypt the most demanding feature of cloud computing security. Cloud computing allows both large and small companies to have the opportunity to use Internet-based services .so that they can lower the start-up costs, capital expenditures, access applications only if there is a need, use services on pay as-you use basis, and quickly lower or increase ability. As the authors offer the years of unparallel skill and knowledge, they discuss all the highly challenging topics like data ownership, privacy protections, data mobility, standard of service and service levels, bandwidth costs, data protection, and support. Advanced Encryption Standard is one on the most regular and mostly symmetric block cipher algorithm. This algorithm has a specific form to encrypt and decrypt subtle data and is put in all hardware and software. It is highly tough to hackers to get the actual data when encrypting by AES. Till date there is no proof to crack this algorithm. AES has the capacity to deal with 3 dissimilar key sizes such as AES 128, 192 and 256 bit. Each of its code has 128 bit

2 Title: Crypt-DAC: Cryptographically Enforced Dynamic Access Control in the Cloud

Author: Saiyu Qi, Yuanqing Zheng

Enabling cryptographically enforced access controls for data hosted in untrusted cloud is attractive for many users and organizations. However, designing efficient cryptographically enforced dynamic access control system in the cloud is still challenging. In this paper, we propose Crypt-DAC, a system that provides practical cryptographic enforcement of dynamic access control. Crypt-DAC revokes access permissions by delegating the cloud to update encrypted data. In Crypt-DAC, a file is encrypted by a symmetric key list which records a file key and a sequence of 3 revocation keys. In each revocation, a dedicated administrator uploads a new revocation key to the cloud and requests it to

encrypt the file with a new layer of encryption and update the encrypted key list accordingly. Crypt-DAC proposes three key techniques to constrain the size of key list and encryption layers. As a result, Crypt-DAC enforces dynamic access control that provides efficiency, as it does not require expensive decryption/re-encryption and uploading/re-uploading of large data at the administrator side, and security, as it immediately revokes access permissions. We use formalization framework and system implementation to demonstrate the security and efficiency of our construction

3 Title: cryptographically Enforced Access Control for User Data in Untrusted Clouds, in NSDI

Author: F.Wang, J.Mickens, N.Zeldovich, & V.Vaikuntanathan, Sieve Modern

web services rob users of low-level control over cloud storage—a user’s single logical data set is scattered across multiple storage silos whose access controls are set by web services, not users. The consequence is that users lack the ultimate authority to determine how their data is shared with other web services. In this paper, we introduce Sieve, a new platform which selectively (and securely) exposes user data to web services. Sieve has a user-centric storage model: each user uploads encrypted data to a single cloud store, and by default, only the user knows the decryption keys. Given this storage model, Sieve defines an infrastructure to support rich, legacy web applications. Using attribute-based encryption, Sieve allows users to define intuitively understandable access policies that are cryptographically enforceable. Using key homomorphism, Sieve can re encrypt user data on storage providers in situ, revoking decryption keys from web services without revealing new keys to the storage provider. Using secret sharing and two-factor authentication, Sieve protects cryptographic secrets against the loss of user devices like smartphones and laptops. The result is that users can enjoy rich, legacy web applications, while benefiting from cryptographically strong controls over which data a web service can access

4 Title: Attribute-based encryption for fine-grained access control of encrypted data

Author: V. Goyal, O. Pandey, A. Sahai, and B. Waters

As more sensitive data is shared and stored by third-party sites on the Internet, there will be a need to encrypt data stored at these sites. One drawback of encrypting data, is that it can be selectively shared only at a coarse-grained level (i.e., giving another party your private key). We develop a new cryptosystem for fine-grained sharing of encrypted data that we call Key-Policy Attribute-Based Encryption (KP-ABE). In our cryptosystem, ciphertexts are labelled with sets of attributes and private keys are associated with access structures that control which ciphertexts a user is able to decrypt. We demonstrate the applicability of our construction to sharing of audit-log information and broadcast encryption. Our construction supports delegation of private keys which subsumes Hierarchical Identity-Based Encryption (HIBE).

5 Title: On the Practicality of Cryptographically Enforcing Dynamic Access Control Policies in the Cloud

Author: W. C. Garrison III, A. Shull, S. Myers, and, A. J. Lee

The ability to enforce robust and dynamic access controls on cloud-hosted data while simultaneously ensuring confidentiality with respect to the cloud itself is a clear goal for many users and organizations. To this end, there has been much cryptographic research proposing the use of (hierarchical) identity-based encryption, attribute-based encryption, predicate encryption, functional encryption, and related technologies to perform robust and private access control on untrusted cloud providers. However, the vast majority of this work studies static models in which the access control policies being enforced do not change over time. This is contrary to the needs of most practical applications, which leverage dynamic data and/or policies. In this paper, we show that the cryptographic enforcement of dynamic access controls on untrusted platforms incurs computational costs that are likely prohibitive in practice. Specifically, we develop lightweight constructions for enforcing role-based access controls (i.e., RBAC0) over cloud-hosted files using identity-based and traditional public-key cryptography. This is done under a threat model as close as possible to the one assumed in the cryptographic literature. We prove the correctness of these constructions, and leverage real-world RBAC datasets and recent techniques developed by the access control community to

experimentally analyze, via simulation, their associated computational costs. This analysis shows that supporting revocation, file updates, and other state change functionality is likely to incur prohibitive overheads in even minimally-dynamic, realistic scenarios. We identify a number of bottlenecks in such systems, and fruitful areas for future work that will lead to more natural and efficient constructions for the cryptographic enforcement of dynamic access controls. Our findings naturally extend to the use of more expressive cryptographic primitives (e.g., HIBE or ABE) and richer access control models (e.g., RBAC1 or ABAC).

CHAPTER 3

AIM AND SCOPE

3.1 SCOPE OF THE PROJECT

We aim to provide access control for the cloud-hosted file data. To stores encrypted data on the cloud, but never reveals the decryption keys to the cloud. This protects the confidentiality of the file data. To uses cryptography to enforce access control so that users can only read file data according to their access permissions. For writing permission enforcement, our system relies on the cloud provider to validate write privileges of users prior to file updates. This project can be applied in the areas where the confidential data requires more privacy as well as to maintain reliability on the data. The Organization facing one of the problems in the digital word to maintain the cloud hosted data. The data may be prone to some of the attacks or outsourced by the cloud provider itself. That's where the access control concept will get into the area to provide set of access permissions for the users or roles as well as write and read permissions to the cloud

3.2 OBJECTIVE

- To maintain the data within an organization is sharable.
- To store and maintain the data in an efficient way.
- To provide data secured in case of unauthorised users.

3.3 INPUT AND OUTPUT

The major inputs and outputs and major functions of the system are follows:

Input

- The entities such as data owners and users must register the account for login. All the user details have been stored the data in our database for the purpose of identifying the entities.
- The data owner uploads the necessary details to involve in the storing operation.
- The data uploaded by the owners must be accepted by the cloud in order to write data.

Output

- Only the authorized data users can able to read that data.
- The data will be accessed by the entities having the key

3.4 EXISTING SYSTEM

In existing system, incur high overhead when access control policies need to be changed in practice. If a decryption key is revealed to an entity, anytime they can able to access the data even after they revoked. There must be a scenario to get those data from the cloud and again the owner wants to encrypt again using one another unique key. At a first glance, the revocation of a user's permission can be done by revoking his access to the keys with which the files are encrypted. This solution, however, is not secure as the user can keep a local copy of the keys before the revocation. To prevent such a problem, files have to be re-encrypted with new keys. This requires the file owner to download the file, re-encrypt the file, and upload it back for the cloud to update the previous encrypted file, incurring prohibitive communication overhead at the file owner side.

3.5 PROPOSED SYSTEM

In proposed system, the company's contract with a commercial cloud provider to outsource enterprise storage. There are three types of entities in our system model: a cloud provider, an access control administrator, and a large number of users. The cloud provider is responsible for the data storage and management. The data includes file data of users in the company, as well as policy data regulating access policies for these files. Both the policy/file data are encrypted prior to being uploaded to the cloud provider. The access control administrator is responsible for managing access policies of the file data. It assigns/revokes access permissions by creating, updating, and distributing cryptographic keys used to encrypt files. Users may download any policy/file data from the cloud, but are only allowed to decrypt and read files according to their access permissions. We do not consider data deduplication issue. If needed, secure data deduplication technique can be used here

3.5.1 Advantage of Proposed System

- This Project achieves efficient revocation, efficient file access and immediate revocation simultaneously.
- For revocation efficiency, it incurs lightweight communication overhead at the administrator side as it does not need to download and re-upload file data.
- For immediate revocation, the permissions of users are immediately revoked as the files are re-encrypted.
- For file access efficiency, the files are still encrypted by symmetric keys (symmetric key encryption)
- Threshold value is introduced to perform the encryption to avoid re encryption overhead of the data

3.6 HARDWARE AND SOFTWARE REQUIREMENTS

Table 3.1 HARDWARE AND SOFTWARE REQUIREMENTS

DEVELOPING KIT			
	PROCESSOR	RAM	DISK SPACE
Eclipse	Computer with a 2.6GHz processor or higher	2GB	Minimum 20GB
DATABASE			
MySQL 5.0	Intel Pentium processor at 2.6GHz	Minimum 512 MB Physical Memory; 1 GB	Minimum 20 GB
HeidiSQL 8.3	Intel Pentium processor at 2.6GHz	Minimum 512 MB Physical Memory; 1 GB	Minimum 20 GB

CHAPTER 4

MATERIALS AND METHODS

The purpose of this document is to present a detailed description of the Architecture and design of our project. It will explain how we implement our project. We use UML diagrams and E-R diagrams to represent our project.

4.1 SYSTEM ARCHITECTURE

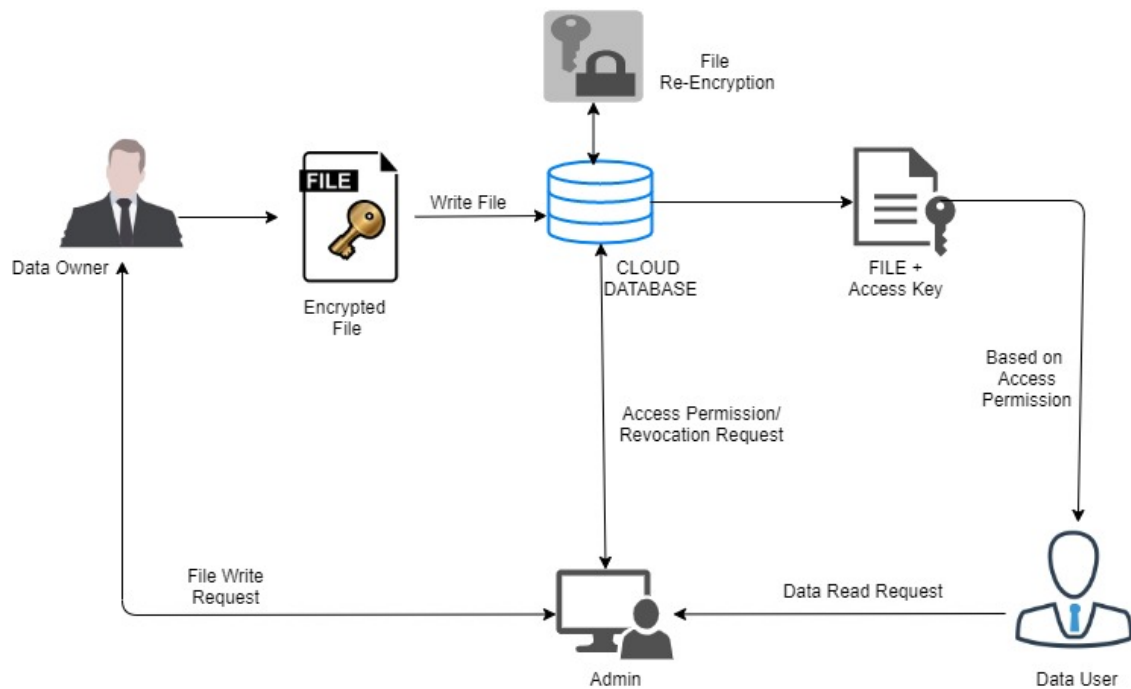


Figure 4.1 Architecture

Data Owner

The data owner uploads the data to the centralized database for sharing the data as shown in figures 4.2, 4.6, 4.10. The data stored in the database for sharing is in the encrypted format so that no need to worry about the data breach, while it is kept under secured. The user wants to post request to the data owners for accessing the data

Admin

The Admin module combines the database into a single unit i.e., it acts as an interface to access the data within the users in the company environment as shown in figures 4.4, 4.8, 4.12. It manages the write permissions and read permissions with in the shared data.

Data Users

The data is shared between the different users, for accessing the data the user searches and give request to the owner as shown in figures 4.3, 4.7, 4.11. Then it will transfer the request to the particular user with the required permission given to the data user.

Data Encryption

For data encryption we use attribute-based encryption which is one of the advanced encryption standard(AES) over the DES scheme. The data will be more complicated to decrypt without the decryption key, and it is the symmetric key encryption type. We use to encrypt and decrypt data in the data owner and the data user module.

Access Revocation

If the access is revoked to the users then the data must be secured from the unauthorized access. For that purpose, the data will be re-encrypted and an additional layer will be added to the data

4.2 IMPLEMENTATION OF AES ALGORITHM

The matrix of 4x4 consisting of 128 bytes input block is known as the state array. The process of encryption revolves around four stages namely mix, columns, sub bytes, add round key and shift rows.

Sub Bytes

It is defined as substitution step. It is nonlinear. Each byte is restored with another according to S-box. The operation gives an indirect proportion in cipher. The resultant matrix consists of four columns and four rows

Shift Rows

It is stage where each row is rotated repetitively a definite number of times. It is also known as permutation. The four rows in the matrix are rotated accordingly. The rows are shifted to the left. Shift is carried out as Row1 is not rotated. Row2 is shifted one byte place to the left. Row3 is shifted two places to

the left. Row4 is shifted three places to the left. The resultant matrix consists of the 16 bytes but rotated with respect to each other.

Mix Columns

In this step, each column is changed using matrix multiplication. Each column consists of four bytes. The resultant matrix consists of 16 bytes. The input is taken for each column. It takes four bytes. The output produces four bytes which is entirely different from the four bytes given as input.

Add Round Key

The round key is bounded to each byte of state. In this particular step, the matrix is XORed with the round key. A 4x4 matrix represents the original key. It contains 128bits. This 4-word key where each word is of 4 bytes, is converted to a 43 words key. The first four words represent W[0], W[1], W[2], and W[3].

4.3 E–R DIAGRAMS

- The relation upon the system is structured through a conceptual ER-Diagram, which not only specifies the existing entities, but also the standard relations through which the system exists and the cardinalities that are necessary for the system state to continue.
- The Entity Relationship Diagram (ERD) depicts the relationship between the data objects. The ERD is the notation that is used to conduct, the data modelling activity the attributes of each data object noted, is the ERD can be described design a data object description
- The set of primary components that are identified by the ERD are
 - Data Object
 - Relationships
 - Attributes
 - Various types of indicators

The primary purpose of the ERD is to represent data objects and their relationships

Data Owner

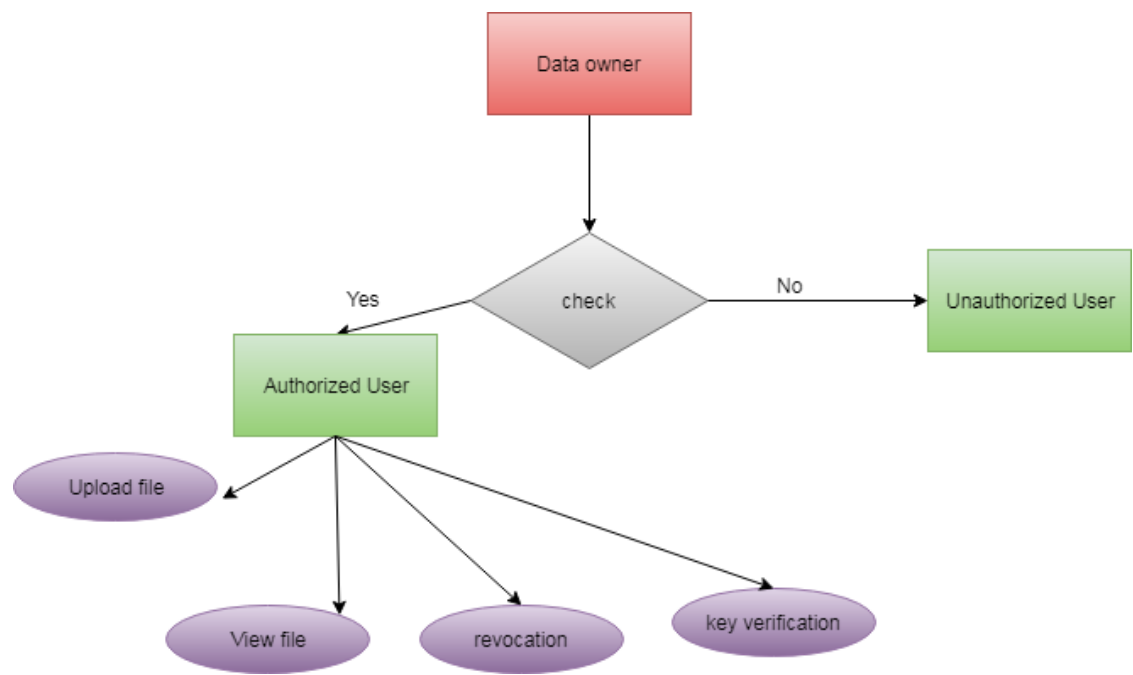


Figure 4.2 Data Owner E-R Diagram

Data User

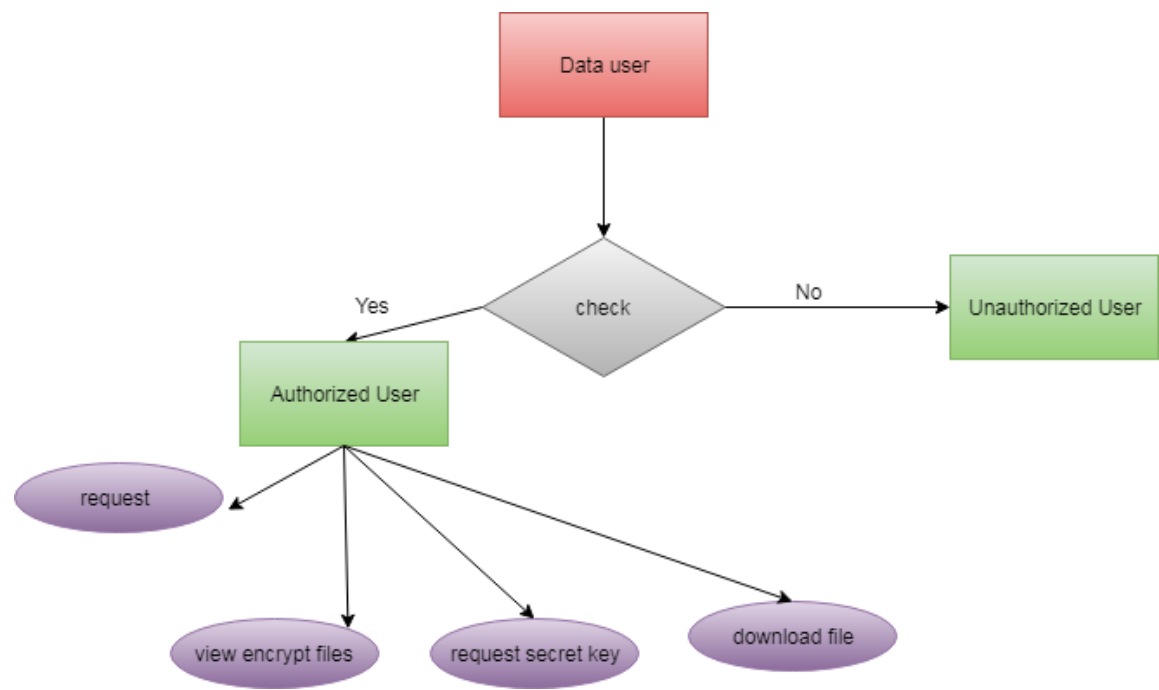


Figure 4.3 Data User E-R Diagram

Admin

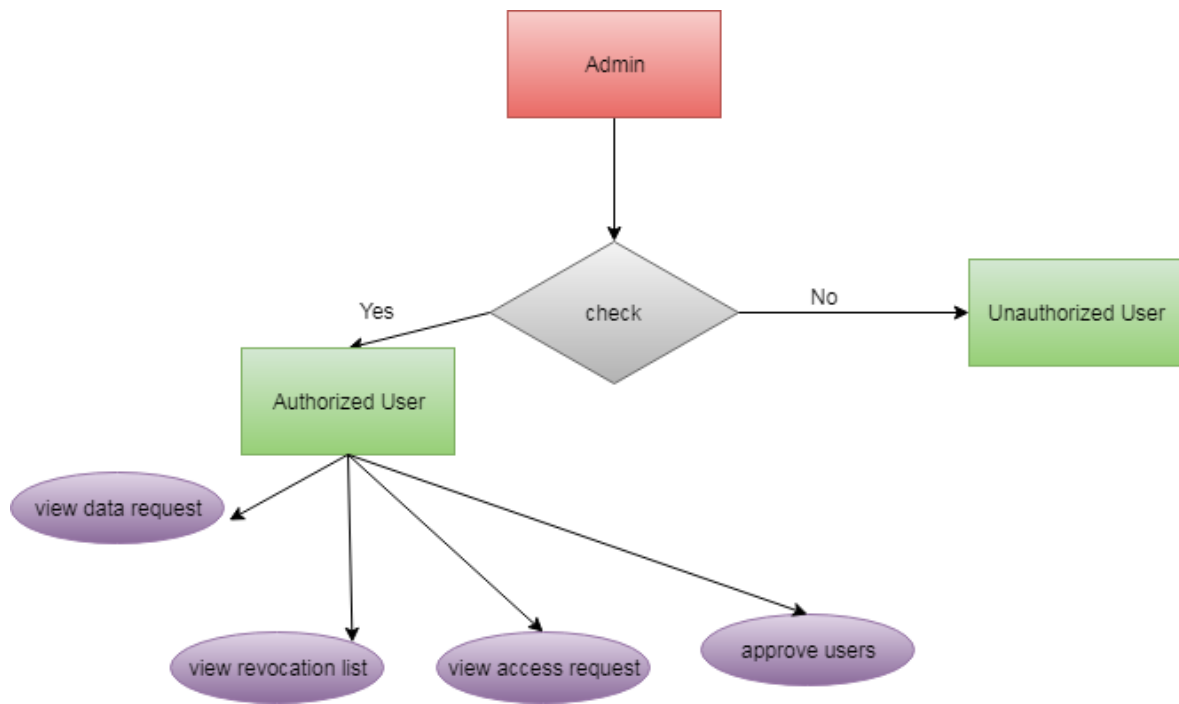


Figure 4.4 Admin E-R Diagram

Cloud

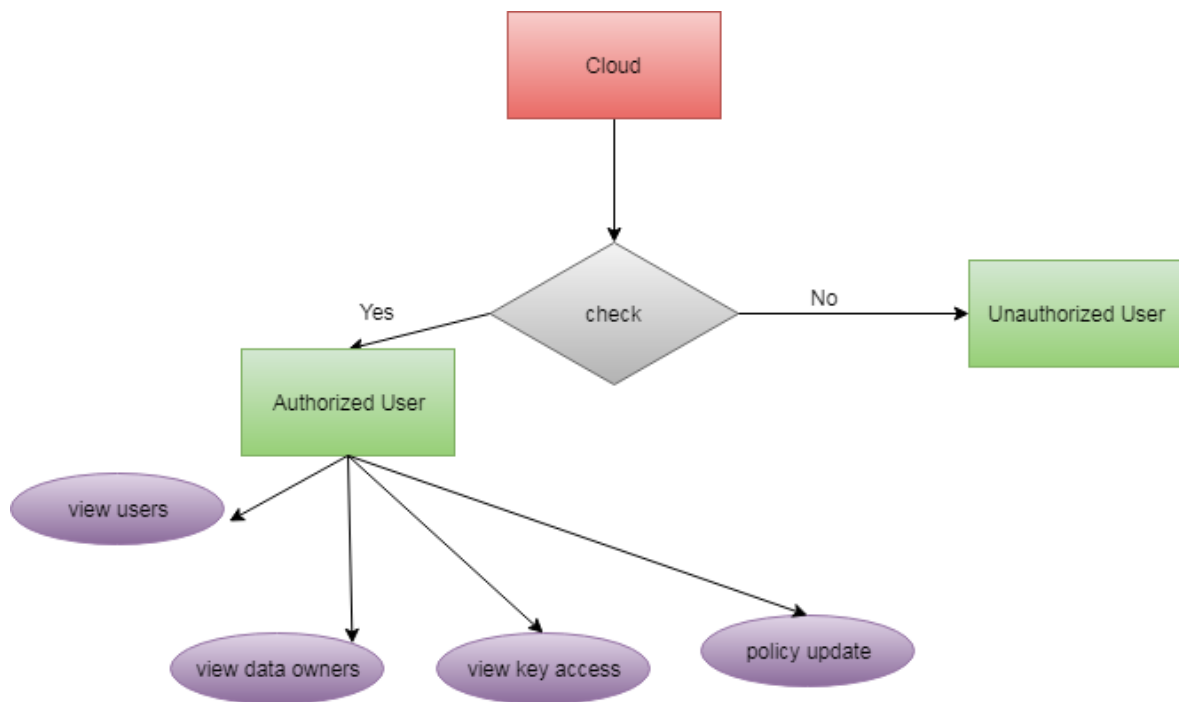


Figure 4.5 Cloud E-R Diagram

4.4 INTRODUCTION OF DFD'S

A data flow diagram is a graphical tool used to describe and analyze the movement of data through a system. These are the central tool and the basis from which the other components are developed. The transformation of data from input to output through processing, may be described logically and independently of physical components associated with the system. These are known as the logical data flow diagrams. The physical data flow diagrams show the actual implements and movement of data between people, departments and workstations. A full description of a system actually consists of a set of data flow diagrams. Using two familiar notations Yourdon, Gane and Sarson notation develops the data flow diagrams. Each component in a DFD is labelled with a descriptive name. The process is further identified with a number that will be used for identification purpose. The development of DFD'S is done on several levels. Each process in lower-level diagrams can be broken down into a more detailed DFD in the next level. The top-level diagram is often called context diagram. It consists a single process bit, which plays a vital role in studying the current system. The process in the context level diagram is exploded into another process at the first level DFD.

The idea behind the explosion of a process into more process is that understanding at one level of detail is exploded into greater detail at the next level. This is done until further explosion is necessary and an adequate amount of detail is described for analysts to understand the process.

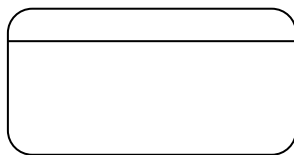
Larry Constantine first developed the DFD as a way of expressing system requirements in a graphical form, this leads to the modular design.

A DFD is also known as a "bubble Chart" has the purpose of clarifying system requirements and identifying major transformations that will become programmed in system design. So, it is the starting point of the design to the lowest level of detail. A DFD consists of a series of bubbles joined by data flows in the system.

4.4.1 DFD SYMBOLS

In the DFD, there are four symbols

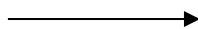
1. A square defines a source (originating) or destination of system data
2. An arrow identifies data flow. It is the pipeline through which the information flows
3. A circle or a bubble represents a process that transforms the incoming data flow into outgoing data flows.
4. An open rectangle is a data store, data at rest or a temporary repository of data



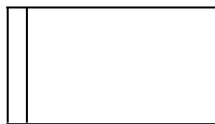
A process that transforms the data flow



Source or Destination of data



Data flow



Data Store

4.4.2 CONSTRUCTING A DFD:

Several rules of thumb are used in drawing DFD'S:

1. Process should be named and numbered for an easy reference. Each name should be representative of the process.
2. The direction of flow is from top to bottom and from left to right. Data traditionally flow from source to the destination although they may flow back to the source. One way to indicate this is to draw the long flow line back to a source. An alternative way is to repeat the source symbol as a destination. Since it is used more than once in the DFD it is marked with a short diagonal.
3. When a process is exploded into lower-level details, they are numbered.

4. The names of data stores and destinations are written in capital letters.
Process and dataflow names have the first letter of each word capitalized

A DFD typically shows the minimum contents of data store. Each data store should contain all the data elements that flow in and out.

Questionnaires should contain all the data elements that flow in and out. Missing interfaces redundancies and like is then accounted for often through interviews.

4.4.3 SAILENT FEATURES OF DFD'S

1. The DFD shows the flow of data, not of control loops and decision are controlled considerations do not appear on a DFD.
2. The DFD does not indicate the time factor involved in any process, whether the dataflow take place daily, weekly, monthly or yearly
3. The sequence of events is not brought out on the DFD.

4.4.4 DATA FLOW DIAGRAMS

Data Owner

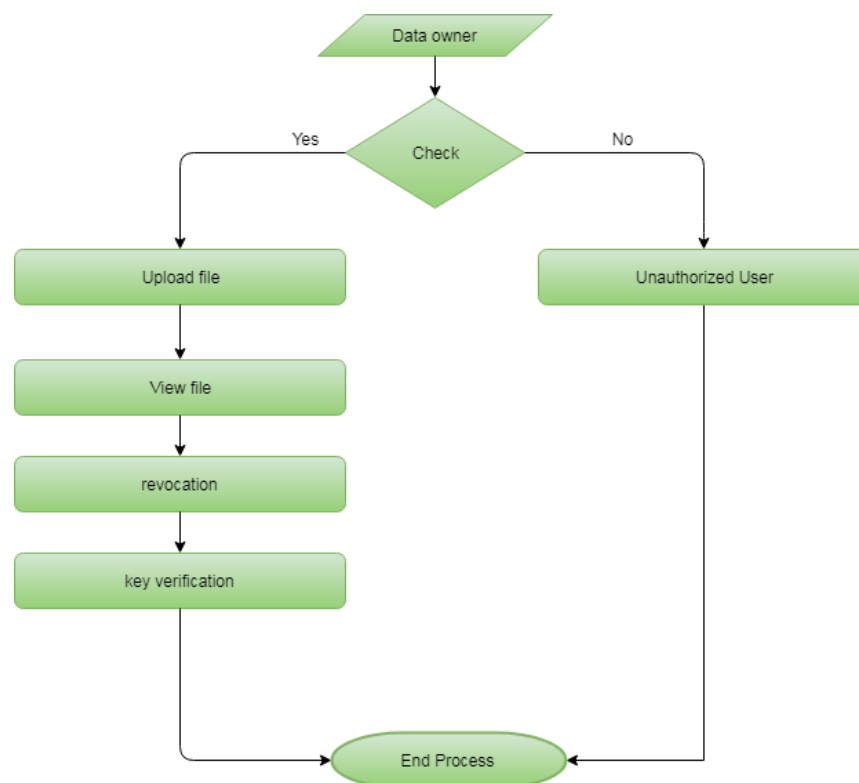


Figure 4.6 Data Owner DFD

Data User

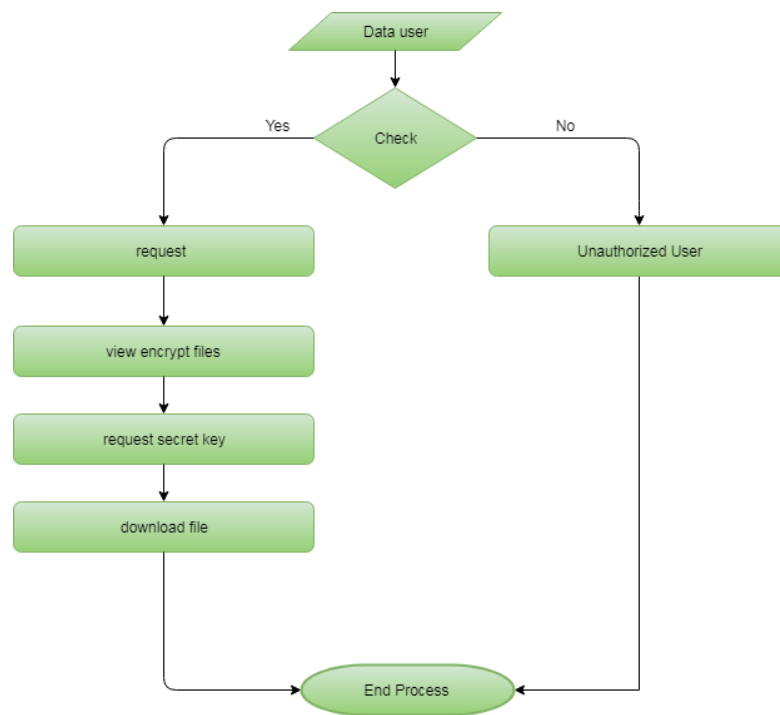


FIGURE 4.7 Data User DFD

Admin

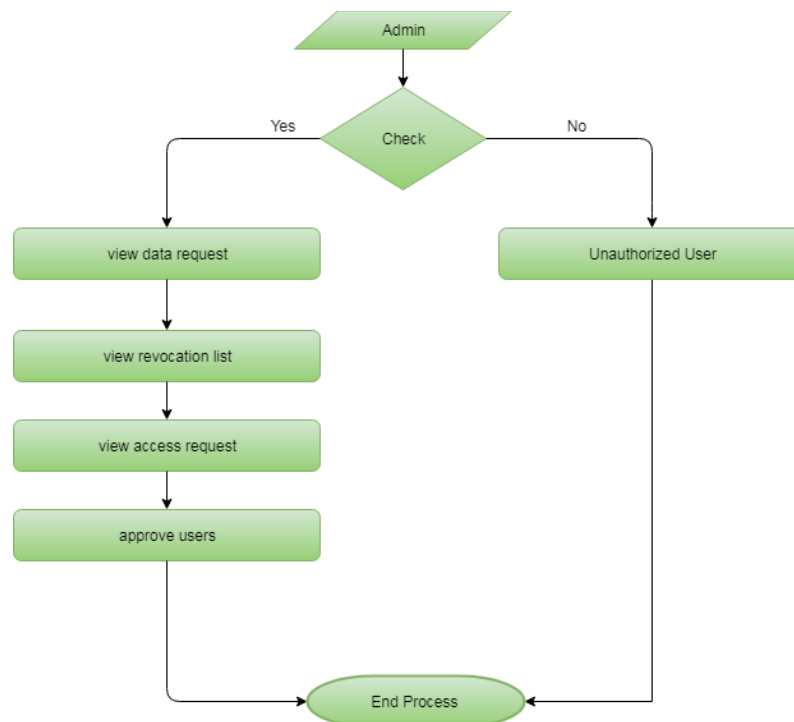


FIGURE 4.8 Admin DFD

Cloud

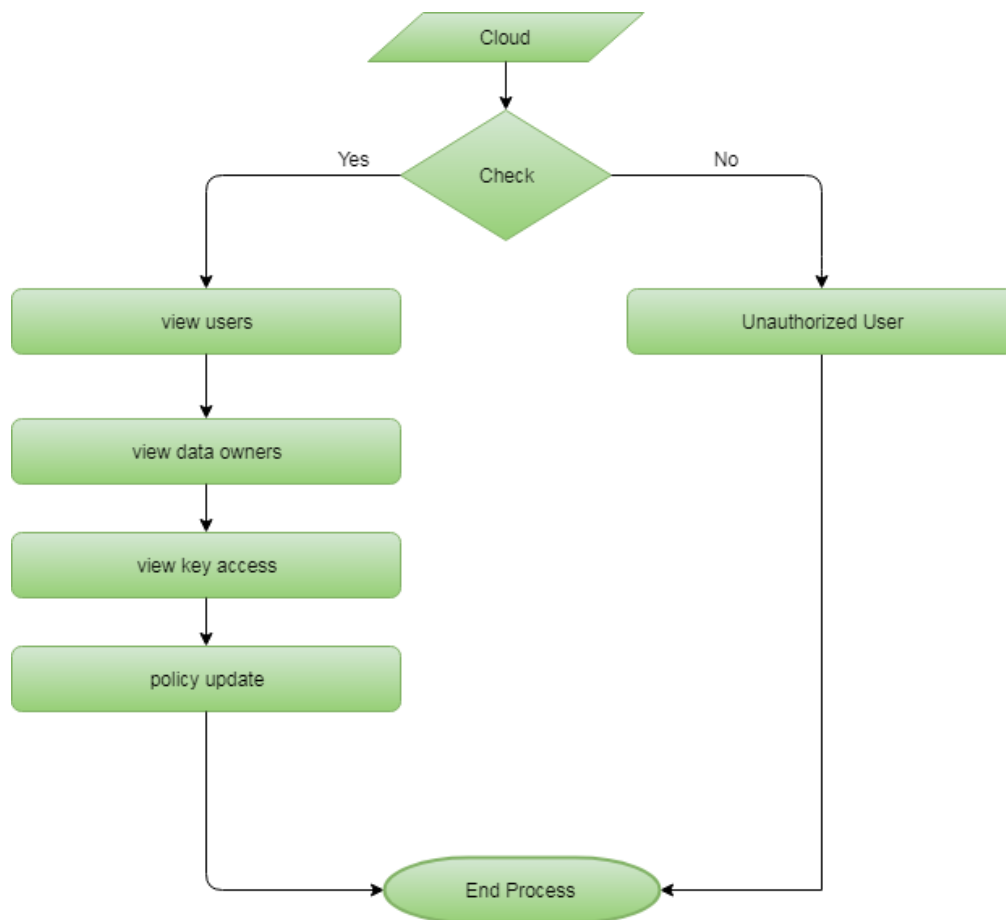


FIGURE 4.9 Cloud DFD

4.5 INTRODUCTION TO USE CASE DIAGRAM

The Use Case Document is a business document which provides a story of how a system, and its actors, will be utilized to achieve a specific goal. An effective Use Case should provide a detailed step-by-step description of how the system will be used by its actors to achieve the planned outcome. The purpose of the Use Case is to tie the business needs of the system to the design parameters of the system to ensure that the completed system achieves the goals established by the business requirements. The level of detail in Use Cases may vary greatly depending on the size and complexity of the system being designed. This Use Case template provides you with everything you need to develop your Use Case Document.

4.5.1 USE CASE DIAGRAMS

Data Owner

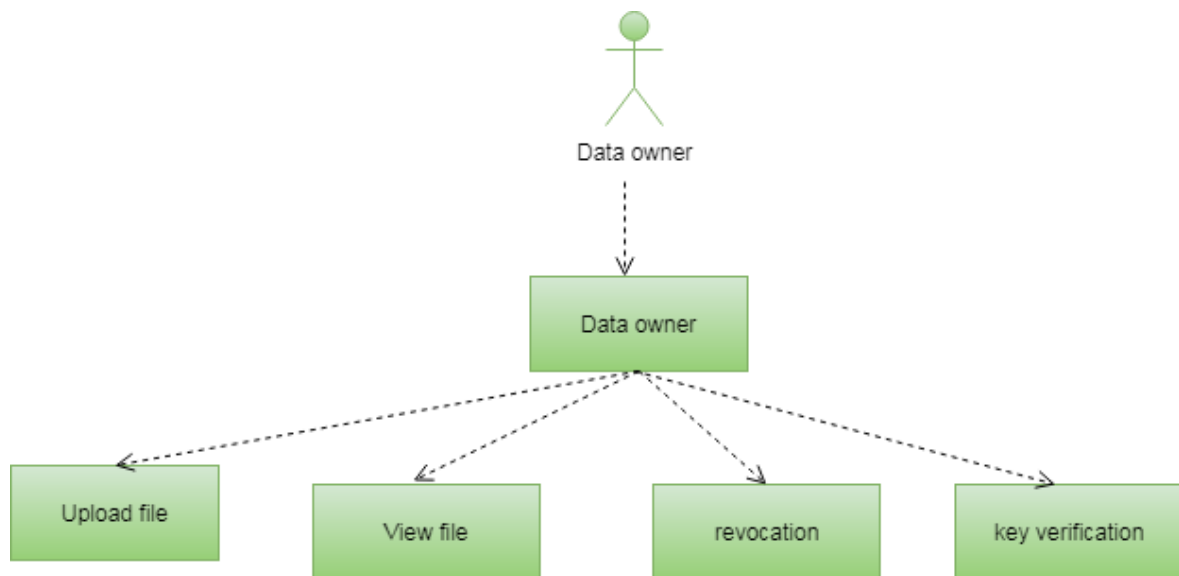


FIGURE 4.10 Data Owner Use Case Diagram

Data User

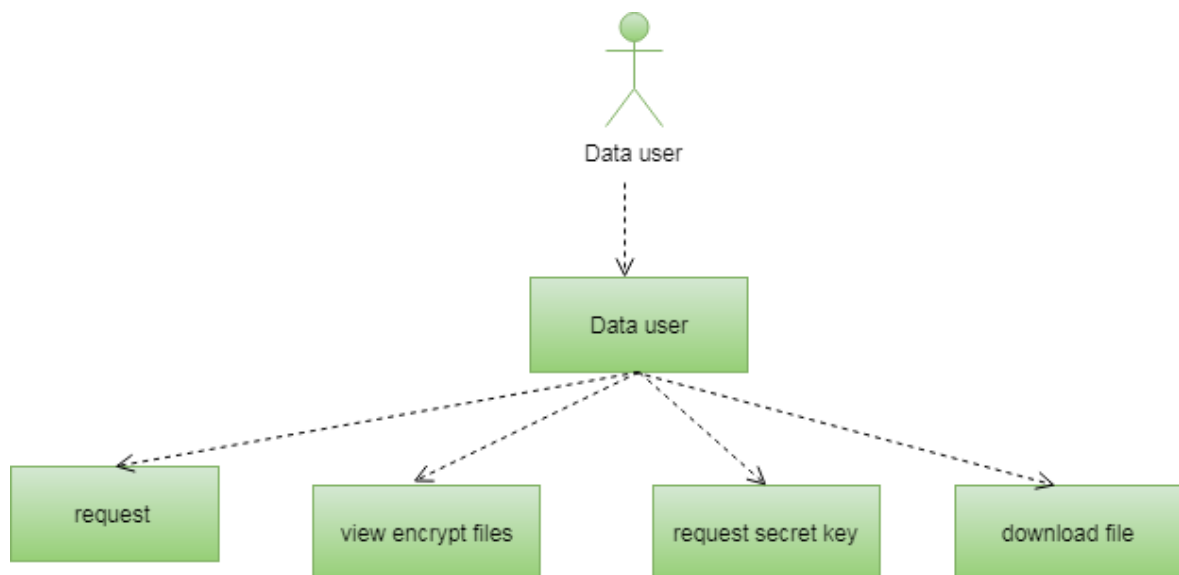


FIGURE 4.11 Data User Use Case Diagram

Admin

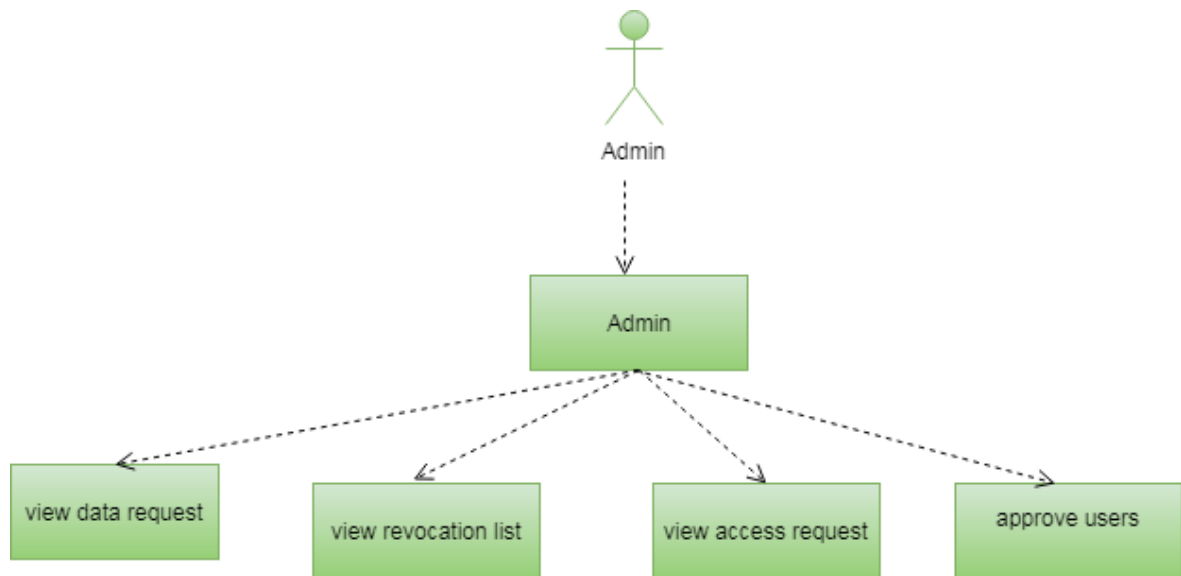


FIGURE 4.12 Admin Use Case Diagram

Cloud

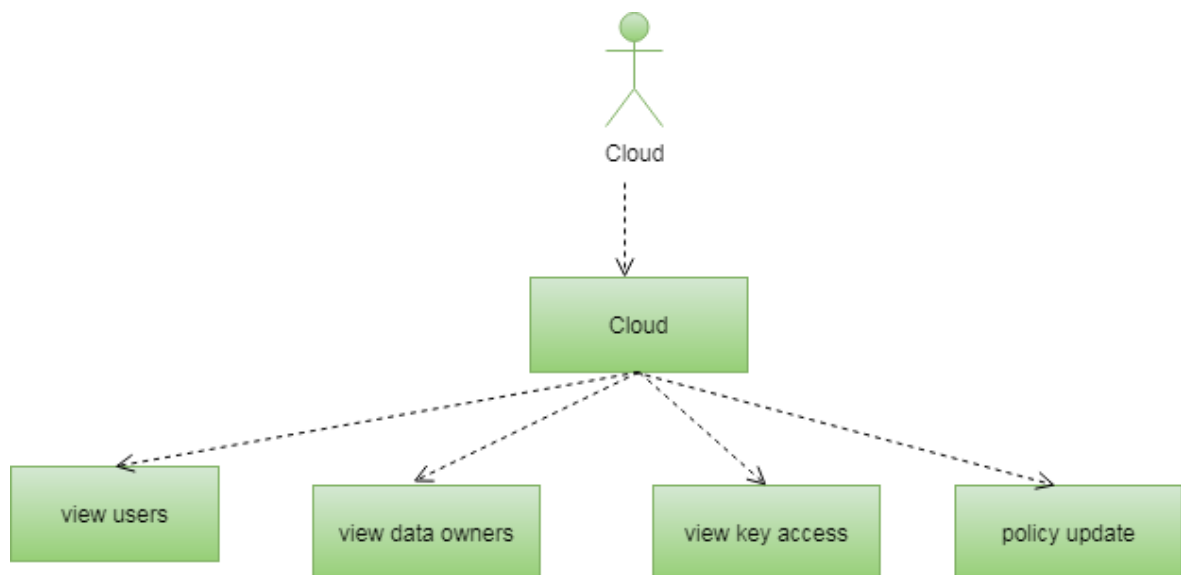


FIGURE 4.13 Cloud Use Case Diagram

4.6 INTRODUCTION OF CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modelling 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 objects.

4.6.1 PURPOSE OF CLASS DIAGRAM

1. Shows static structure of classifiers in a system
2. Diagram provides a basic notation for other structure diagrams prescribed by UML
3. Helpful for developers and other team members too
4. Business Analysts can use class diagrams to model systems from a business perspective

4.6.2 CLASS DIAGRAM

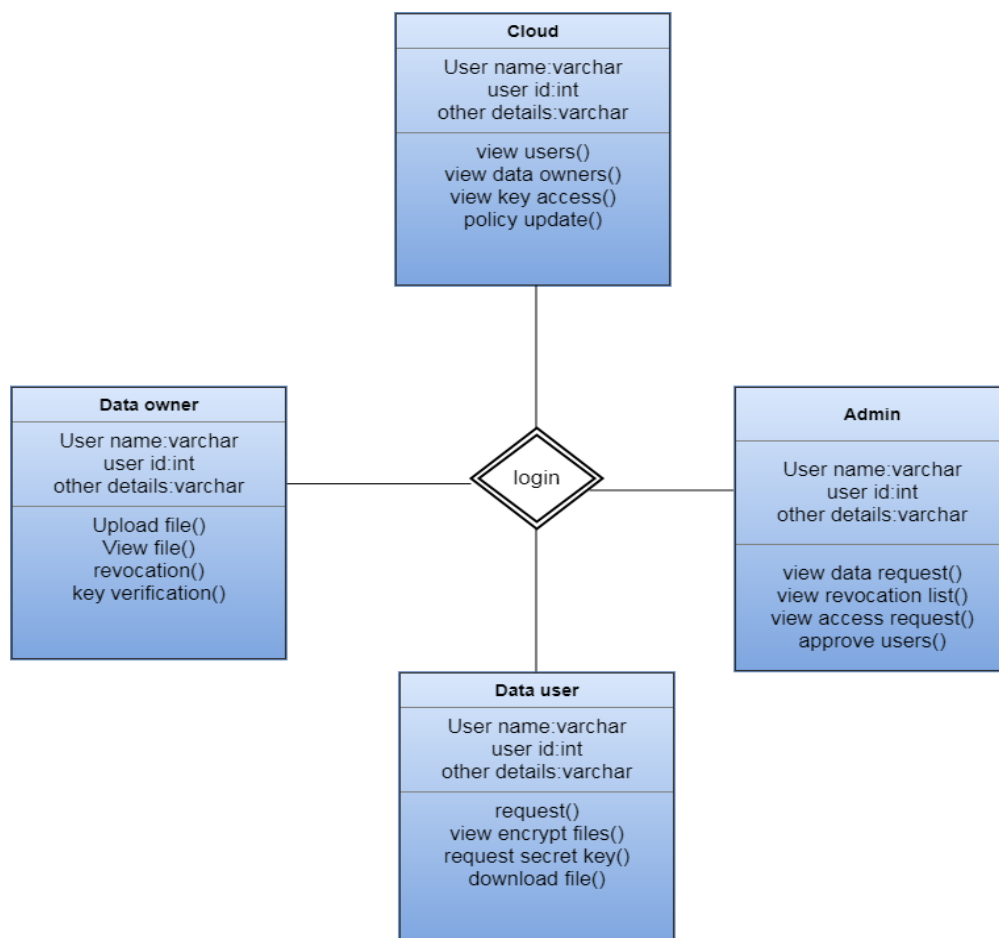


FIGURE 4.14 CLASS DIAGRAM

CHAPTER 5

RESULTS AND DISCUSSION

5.1 OUTPUT SCREENSHOTS

On the home page as shown in figure 5.1 the respective user will log in or sign up for a specific role available on this page. The data owner can only write the file to the cloud while the data user can read the file by decrypting the file

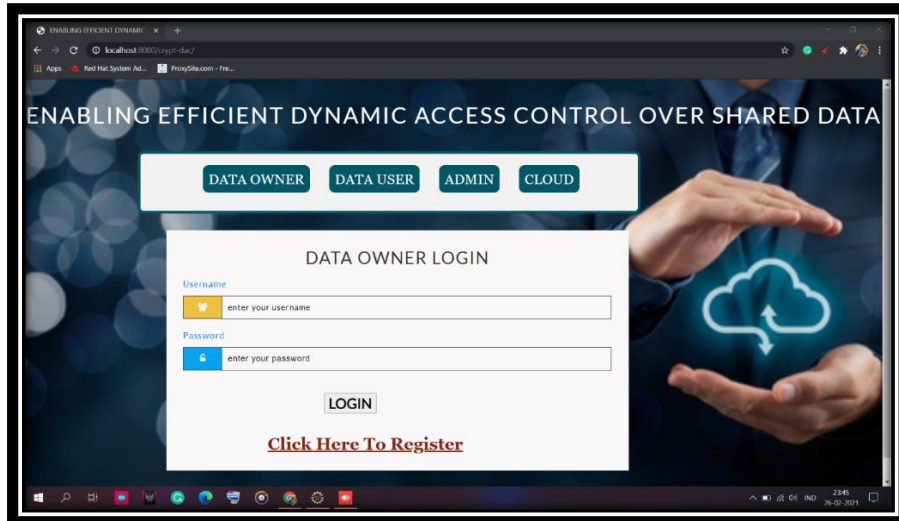


FIGURE 5.1 Home Page

If the data owner wants to write a file on the cloud then he/she has to get access from the admin. Once the request of the data owner has been accepted by admin, owner will receive a mail with an encryption key which he has to enter at the time of file upload as shown in figure 5.2

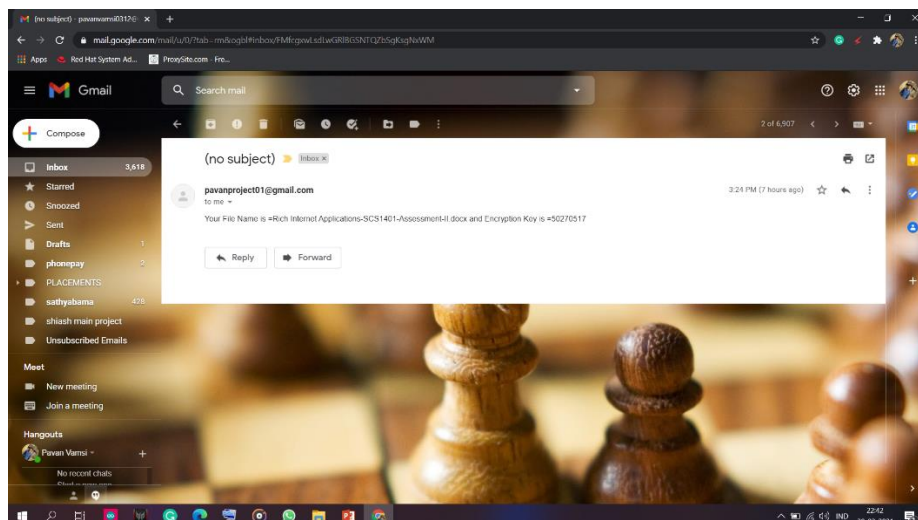


FIGURE 5.2 Encryption Key Mail

After entering the encryption key and other needed details, the data owner will receive the acknowledgement stating that the File encrypted and written to cloud successfully as shown in figure 5.3

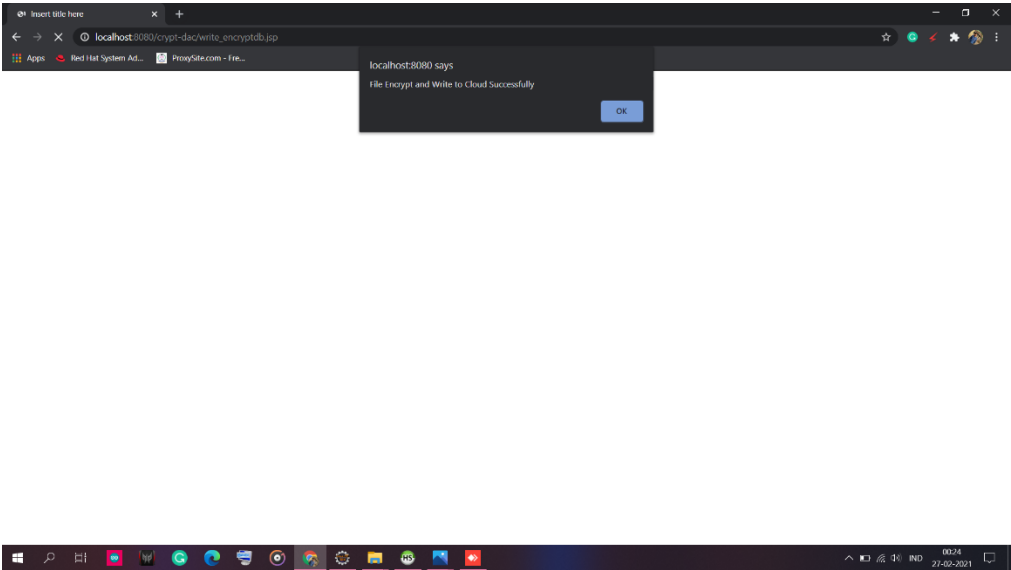


FIGURE 5.3 File Write Status

Once the files are uploaded into the cloud, the file is in an encrypted format as shown in figure 5.4 and no one can read it until and unless they had a decryption key

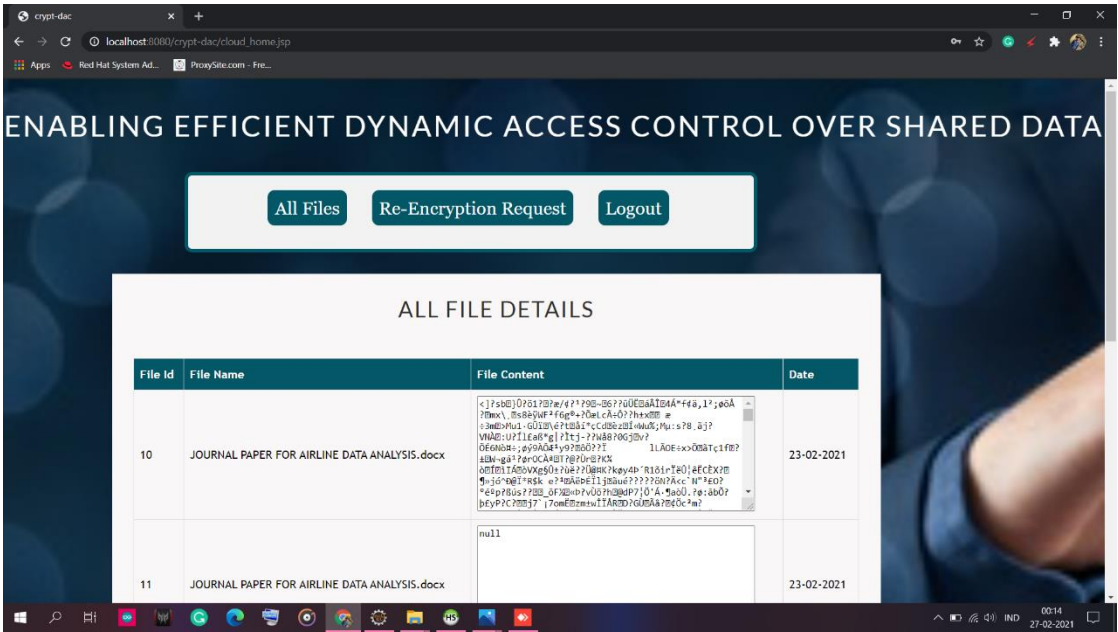


FIGURE 5.4 Encrypted Files In cloud

If the data user wants to read a file from the cloud then he/she has to get access from the admin. Once the request of the data user has been accepted by admin, data user will receive a mail with a decryption key which he has to enter at the time of file reading as shown in figure 5.5

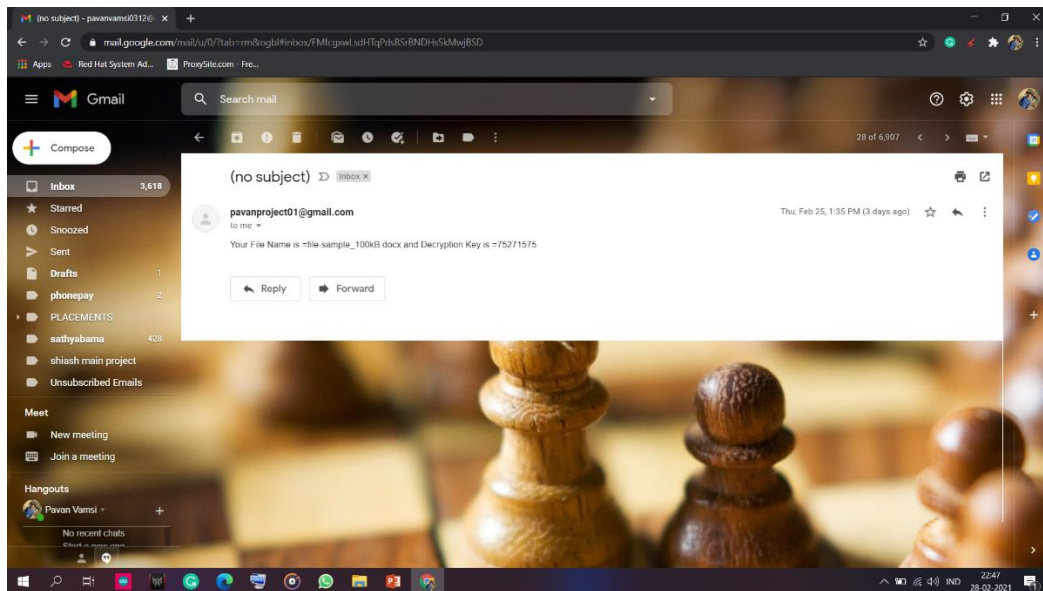


FIGURE 5.5 Decryption Key Mail

After entering the decryption key, User can able to read the file or they can even download the file to their local device as shown in figure 5.6

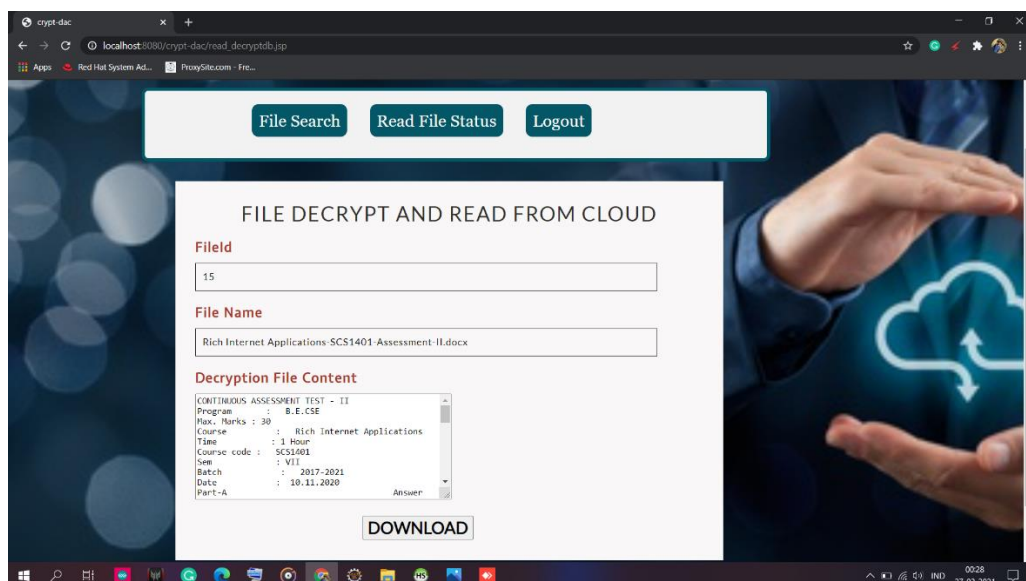
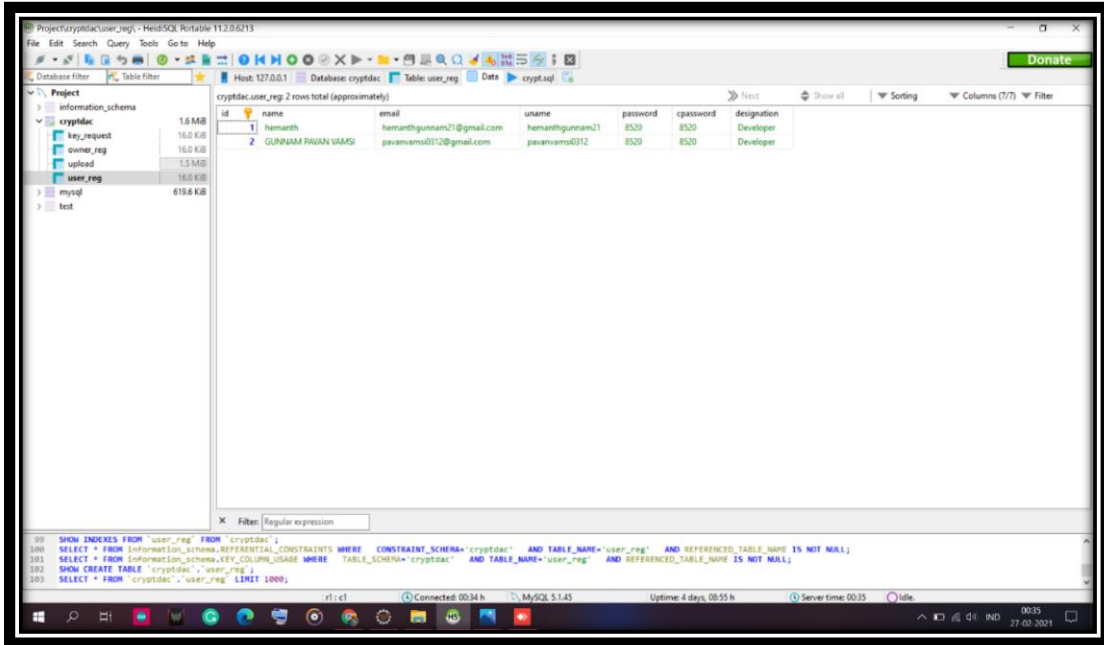


FIGURE 5.6 Decrypted File

5.2 Database Screenshots

The registered data user's data is stored in the database under user_reg table as shown in figure 5.7

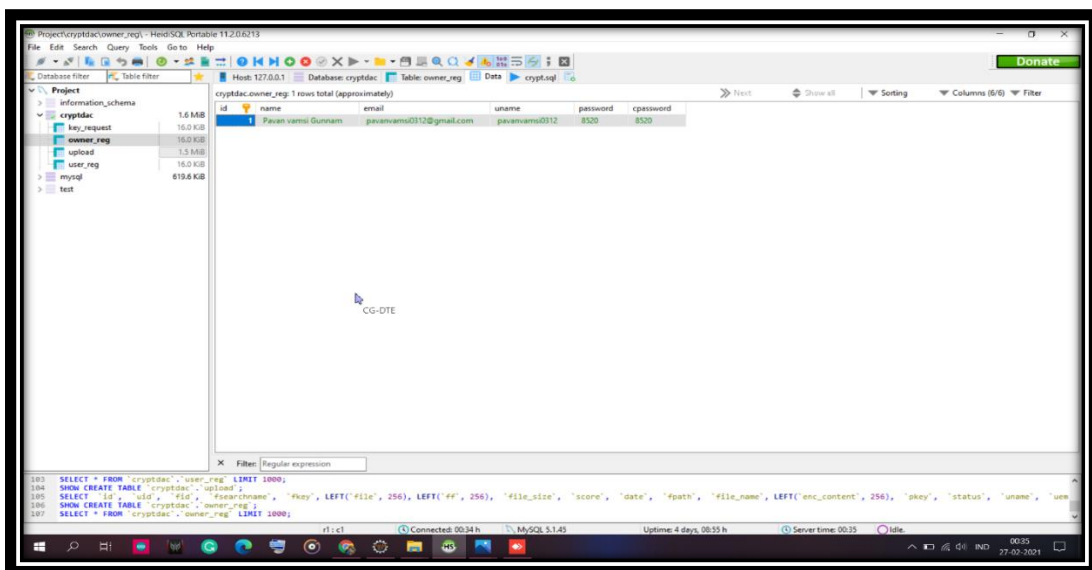


The screenshot shows the HeidiSQL interface with the 'cryptdac' database selected. The 'user_reg' table is displayed, showing two rows of data. The table structure is as follows:

id	name	email	uname	password	cpassword	designation
1	hemant	hemantgunnam21@gmail.com	hemantgunnam21	8520	8520	Developer
2	GUNNAM PRIAN VAMSI	pavanams0312@gmail.com	pavanams0312	8520	8520	Developer

Figure 5.7 Data Users Registration Database

The registered data owner's data is stored in the database under owner_reg table as shown in figure 5.8



The screenshot shows the HeidiSQL interface with the 'cryptdac' database selected. The 'owner_reg' table is displayed, showing one row of data. The table structure is as follows:

id	name	email	uname	password	cpassword
1	Pavan vamsi Gunnam	pavanams0312@gmail.com	pavanams0312	8520	8520

Figure 5.8 Data Owners Registration Database

Once data owners uploaded files they will store them in the upload table as shown in the figure 5.9

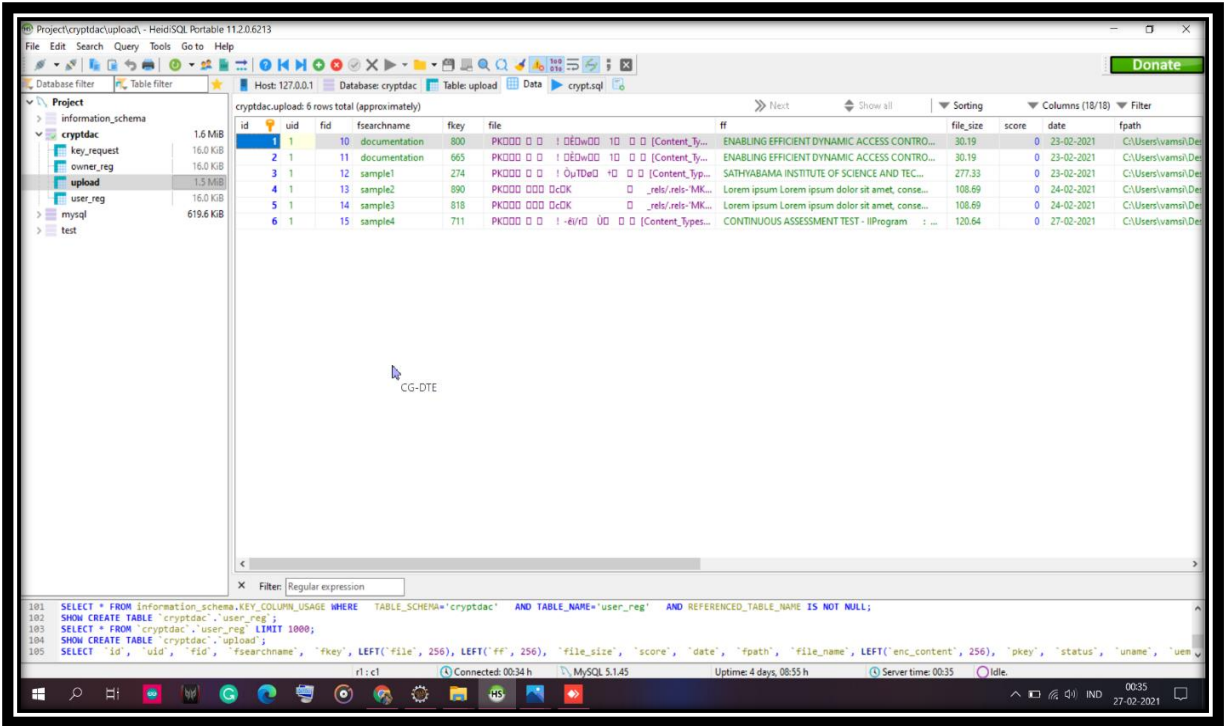


Figure 5.9 Uploaded Files Database

CHAPTER 6

CONCLUSION

In this project we enforce the Cryptographically enabled dynamic access control over the cloud stored data. We prevent the unauthorized access on the data after revocation of the access control. Their revocation operation relies on additional trusted infrastructure and an active entity to re-encrypt all affected files under the new policy. It achieves efficient revocation, efficient file access and immediate revocation simultaneously. For revocation efficiency, incurs lightweight communication overhead at the administrator side as it does not need to download and re-upload file data. For immediate revocation, the permissions of users are immediately revoked as the files are re-encrypted. The future enhancement includes increasing the efficiency of the access control by using other forms of access control methodology as it focuses on the role on which the entity has the permission to occur.

REFERENCES

- [1] Amrita Parashar, "Services & Security of Cloud Computing Using DES", International Journal of Computer & Mathematical Sciences, May 2016
- [2] Arijit Ukil, Debasish Jana and Ajanta De Sarkar, "A Security Framework In Cloud Computing Infrastructure", International Journal of Network Security and Its Applications, Vol.5, Issue No.5
- [3] F. Wang, J. Mickens, N. Zeldovich and V. Vaikuntanathan, "Cryptographically Enforced Access Control for User Data in Untrusted Clouds", in NSDI, 2016
- [4] Nasarul Islam.K.V, " Analysis of Various Encryption Algorithms in Cloud Computing", International Journal of Computer Science and Mobile Computing, July 2017
- [5] Prof. S. Delfin and Rachana Sai, " Cloud data security using AES algorithm", International Journal of Current Trends in Engineering and Research, Oct 2018
- [6] Roshani Raghatate , Sneha Humne and Roshna Wadhwe, "A Survey on Secure Cloud Computing using AES Algorithm", International Journal of Computer Science and Mobile Computing, December 2014
- [7] Saiyu Qi and Yuanqing Zheng, "Crypt-DAC: Cryptographically Enforced Dynamic Access Control in the Cloud", IEEE paper , March 2019
- [8] V.Surya, S.Ranichandra and R.Ranjani, "Secure Cloud Storage Using AES Encryption", International Journal of Innovative Research in Computer and Communication Engineering, Vol. 6, Issue 6, ISSN 2320-9801

APPENDIX

DATABASE:

```
package databaseconnection;

import java.sql.*;

public class Databasecon
{
    static Connection con;

    public static Connection getconnection()
    {
        try
        {
            Class.forName("com.mysql.jdbc.Driver");

            Con=
            DriverManager.getConnection("jdbc:mysql://localhost:3306/cryptdac","root","root");

        }
        catch(Exception e)
        {
            System.out.println(e);
        }
        return con;
    }
}
```

HOMEPAGE

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>ENABLING EFFICIENT DYNAMIC ACCESS CONTROL OVER SHARED DATA</title>
<!-- Meta tag Keywords -->
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta charset="UTF-8">
<meta name="keywords" content="effective register form Widget a Flat Responsive
Widget,Login form widgets, Sign up Web forms , Login signup Responsive web
form,Flat Pricing table,Flat Drop downs,Registration Forms,News Letter
Forms,Elements" />
<script type="application/x-javascript"> addEventListener("load", function() {
setTimeout(hideURLbar, 0); }, false);
function hideURLbar(){ window.scrollTo(0,1); } </script>
<!-- Meta tag Keywords -->
<link rel="stylesheet" href="css/style.css" type="text/css" media="all" /><!--
Style-CSS -->
```

```

<link rel="stylesheet" href="css/font-awesome.min.css" type="text/css"
media="all" />
<link href="//fonts.googleapis.com/css?family=Lato" rel="stylesheet">
<link href="//fonts.googleapis.com/css?family=Expletus+Sans" rel="stylesheet">
<script language="javascript" type="text/javascript">

```

```

function valid()
{

```

```

var aa=document.form1.username.value;
if(aa=="")
{
alert("Enter Username");
document.form1.username.focus();
return false;
}
var dd=document.form1.password.value;
if(dd=="")
{
alert("Enter Password");
document.form1.password.focus();
return false;
}

```

```

return true;
}
</script>
</head>
<body>
<section class="agile-main">
  <header><h1>Crypt-DAC</h1></header>
  <div class="left_column">
    <a href="index.html">DATA OWNER</a>
    <a href="user.jsp">DATA USER</a>
    <a href="admin.jsp">ADMIN</a>
    <a href="cloud.jsp">CLOUD</a>
  </div>
  <div class="agile-top">
    <h2>Data Owner Login</h2>
    <form action="owner_logindb.jsp" method="Post" name="form1"
onSubmit="return valid()">

```

```

      <div class="agile-username">
        <p>username</p>
        <span><i class="fa fa-users" aria-
hidden="true"></i></span>
        <input type="text" name="username" placeholder="enter
your username" >
        <div class="clear"></div>
      </div>

```

```

        <div class="agile-password">
            <p>password</p>
            <span><i class="fa fa-unlock-alt" aria-
hidden="true"></i></span>
            <input type="password" name="password"
placeholder="enter your password">
            <div class="clear"></div>
        </div>

        <input type="submit" value="LOGIN" name="submit"
style="font-size:25px;font-weight:bold;margin-left:250px;margin-
top:20px;"><br><br><br>
        <a href="owner_reg.jsp" style="font-family:georgia;font-
size:30px;font-weight:bold;margin-left:150px;color:#872510;">Click Here To
Register</a>
    </form>
    <div class="clear"></div>
</div>
<div class="clear"></div>
<footer>ENABLING EFFICIENT DYNAMIC ACCESS CONTROL OVER SHARED
DATA</footer>
</section>
</body>
</html>

```

REGISTRATION PAGE

```

<%@page import="databaseconnection.Databasecon"%>
<%@page import="java.sql.ResultSet"%>
<%@page import="java.sql.Statement"%>

<%@page import="java.sql.Connection"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<%
String age11=null;
String name=request.getParameter("name");
String uname=request.getParameter("username");
String email=request.getParameter("email");
String password=request.getParameter("password");
String cpassword=request.getParameter("cpassword");

Statement st2 = null;
ResultSet rs2=null;
try
{
    Connection    con1=Databasecon.getconnection();

```

```

        st2=con1.createStatement();
        String sql1="select * from owner_reg where uname='"+uname+"'";
        rs2=st2.executeQuery(sql1);
        if(rs2.next()){
            %>
            <script>
                alert("Username already exist");
                window.location="owner_reg.jsp";
            </script>
            <%
        }
        else
        {
            Connection con=Databasecon.getConnection();
            Statement st=con.createStatement();
            int x=st.executeUpdate("insert into
owner_reg(name,email,uname,password,cpassword)
values('"+name+"','"+email+"','"+uname+"','"+password+"','"+cpassword+"')");

            %>
            <script>
                alert("Register Successfully");
                window.location="index.html";
            </script>
            <% }

        }
    }
    catch(Exception e)
    {
        e.printStackTrace();
    }
    %>

</body>
</html>

```