A

Project Report

on

Dual Access Control For Cloud Based Data Storage And Sharing

Submitted in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology

by

G Keerteshwar Reddy
(20EG105410)

Mahesh Pawar
(20EG105424)

D Akhil
(20EG105716)

Under the guidance of

E. Radhakrishnaiah

Assistant Professor,

Department of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
ANURAG UNIVERSITY
VENTAKAPUR (V), GHATKESAR (M), MEDCHAL (D), T.S - 500088
TELANGANA
(2023-2024)

DECLARATION

We hereby declare that the report entitled **"Dual Access Control For Cloud Based Data Storage And Sharing"** submitted to the **Anurag University** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology (B. Tech)** in **Computer Science and Engineering** is a record of an original work done by us under the guidance of **E. Radhakrishnaiah, Assistant Professor** and this report has not been submitted to any other university for the award of any other degree or diploma.

Place: Anurag University, Hyderabad

G Keerteshwar Reddy
(20EG105410)

Mahesh Pawar
(20EG105424)

D Akhil
(20EG105716)

ANURAG UNIVERSITY | SCHOOL OF ENGINEERING

CERTIFICATE

This is to certify that the project report entitled **"Dual Access Control For Cloud Based Data Storage And Sharing"** being submitted by **G Keerteshwar Reddy, Mahesh Pawar, D Akhil** bearing the Hall Ticket numbers **20EG105410, 20EG105424, 20EG105716** respectively in partial fulfillment of the requirements for the award of the degree of the **Bachelor of Technology** in **Computer Science and Engineering** to **Anurag University** is a record of bonafide work carried out by them under my guidance and supervision from 2023 to 2024.

The results presented in this report have been verified and found to be satisfactory. The results embodied in this report have not been submitted to any other University for the award of any other degree or diploma.

Signature of The Supervisor
E. Radhakrishnaiah
Assistant Professor

Signature Dean,
Dr. G. Vishnu Murthy
Department of CSE

External Examiner

ACKNOWLEDGMENT

We would like to express our sincere thanks and deep sense of gratitude to project supervisor **E. RADHAKRISHNAIAH,** Assistant Professor, Department of Computer Science and Engineering, Anurag University for his constant encouragement and inspiring guidance without which this project could not have been completed. His critical reviews and constructive comments improved my grasp of the subject and steered to the fruitful completion of the work. His patience, guidance and encouragement made this project possible.

We would like to acknowledge our sincere gratitude for the support extended by **DR. G. VISHNU MURTHY**, Dean, Department of Computer Science and Engineering, Anurag University. We also express our deep sense of gratitude to **Dr. V.V. S. S. S. BALARAM**, Academic coordinator. **Dr. PALLAM RAVI**, Project Coordinator and project review committee members, whose research expertise and commitment to the highest standards continuously motivated us during the crucial stages of our project work.

We would like to express our special thanks to **Dr. V. VIJAYA KUMAR,** Dean School of Engineering, Anurag University, for their encouragement and timely support in my B. Tech program.

G Keerteshwar Reddy
(20EG105410)

Mahesh Pawar
(20EG105424)

D Akhil
(20EG105716)

ABSTRACT

Cloud-based data storage service has drawn increasing interests from both academic and industry in the recent years due to its efficient and low cost management. Since it provides services in an open network, it is urgent for service providers to make use of secure data storage and sharing mechanism to ensure data confidentiality and service user privacy. To protect sensitive data from being compromised, the most widely used method is encryption. However, simply encrypting data (e.g., via AES) cannot fully address the practical need of data management. Besides, an effective access control over download request also needs to be considered so that Economic Denial of Sustainability (EDoS) attacks cannot be launched to hinder users from enjoying service. In this paper, we consider the dual access control, in the context of cloud-based storage, in the sense that we design a control mechanism over both data access and download request without loss of security and efficiency. A dual access control system are designed in this paper, where each of them is for a distinct designed setting. The security and experimental analysis for the systems are also presented.

Keywords: Dual Access control,autentication,authorization,rule-based authentication,encryption.

TABLE OF CONTENT

List of Figures

List of Tables

List of Abbreviations

Abbreviations	Full Form
CP	Ciphertext Policy
CP-ABE	Ciphertext Policy - Attribute Based Encryption
CT	Cipher Text
Intel SGX	Intel Safe Guard Extensions
AES Algorithm	Advanced Encryption Standard

1. Introduction

Over the past few years, cloud storage services have garnered significant interest from both the academic community and the industrial sector. These services are increasingly being integrated into various online commercial platforms, such as Apple iCloud, because of their numerous advantages, including the ease of access and the elimination of the need for on-premise data management. As a result, a growing number of individuals and businesses are choosing to store their data on remote servers to decrease the expenses associated with the upkeep and enhancement of their own data handling systems. Nonetheless, concerns regarding the security of data stored on these platforms remain a significant barrier to their broader adoption.

Cloud-based data storage service has drawn increasing interests from both academic and industry in the recent years due to its efficient and low cost management. Since it provides services in an open network, it is urgent for service providers to make use of secure data storage and sharing mechanism to ensure data confidentiality and service user privacy.

In this project, we consider the dual access control, in the context of cloud-based storage, in the sense that we design a control mechanism over both data access and download request without loss of security and efficiency. Two dual access control systems are designed in this project, where each of them is for a distinct designed setting.

1.1. Two Access Control

In practical scenarios, it often becomes necessary to share stored data with third parties. For instance, a user of Dropbox, referred to here as Alice, might want to share pictures with her friends. To do this without employing encryption, Alice would need to create and distribute a shareable link. While this method does offer a degree of control over who can access the data, thereby preventing unauthorized access (e.g., by people outside Alice's circle of friends), it's possible for the link to be accessed by the cloud service's administrators. Given the inherent risks associated with storing data on a platform that is inherently insecure due to its connectivity to the internet, experts generally advise encrypting data before uploading it to the cloud. This ensures both

the privacy and security of the data. A common approach involves using specific encryption techniques, such as AES, to encrypt the data so that only authorized users with the appropriate decryption key can access it. When it comes to sharing sensitive materials like photographs, without prior knowledge of who the recipients might be, traditional encryption methods, which require the identity of the decryptor to be known beforehand, fall short. In such instances, a policy-based encryption strategy becomes invaluable. This approach allows the data owner to set access policies directly on the encrypted data, ensuring that only authorized individuals can access it. Cloud services are also vulnerable to specific attacks, such as resource-exhaustion attacks, which can disrupt service by overwhelming the server with excessive requests. This can lead to an Economic Denial of Sustainability (EDoS) attack, aimed at draining the financial resources of cloud users by exploiting the pay-as-you-go pricing model. Moreover, unlimited access to download requests can potentially expose encrypted data to network attacks, risking unauthorized information disclosure. To address these issues, this paper introduces a novel concept called dual access control. It incorporates Attribute-Based Encryption (ABE) as a key component, particularly focusing on Ciphertext-Policy ABE (CP-ABE), to secure outsourced data. CP-ABE allows for the establishment of detailed access policies over encrypted data, offering a sophisticated means to control both access and downloading activities. However, merely applying CP-ABE is insufficient for a comprehensive solution that effectively manages both data access and download requests.

1.2. Motivation

The motivation to develop Dual access control is crucial for ensuring the security and privacy of data in cloud-based systems. Developing a system for this purpose could have significant practical implications. Here are some potential motivations to consider:

Enhanced Security: Dual access control can provide an extra layer of security by requiring two levels of authentication or authorization before granting access to sensitive data. This can help prevent unauthorized access and data breaches.

Improved Privacy: By implementing dual access control, users can have more control over who can access their data, enhancing privacy and confidentiality.

Compliance: Many industries and jurisdictions have strict regulations regarding data privacy and security. Implementing dual access control can help organizations comply with these regulations.

Flexibility and Scalability: Dual access control systems can be designed to be flexible and scalable, allowing them to adapt to the changing needs of organizations as they grow.

Research and Innovation: Developing a project in this area can contribute to the research and innovation in the field of data security and access control, potentially leading to new insights and solutions.

1.3. Problem Definition

In the context of cloud-based data storage and sharing, the current challenge resides in establishing a robust and comprehensive access control system that ensures both security and flexibility.Existing single-layer access control mechanisms often fall short in adequately addressing the intricate demands of data security, leading to vulnerabilities and potential breaches.

To protect sensitive data from being compromised, the most widely used method is encryption. However, simply encrypting data (e.g., via AES) cannot fully address the practical need of data management. Besides, an effective access control over download request also needs to be considered so that Economic Denial of Sustainability (EDoS) attacks cannot be launched to hinder users from enjoying service.Implementation of a dual access control framework that effectively integrates multiple layers of authentication and authorisation protocols.

1.4. Problem Illustration

Implementing a dual access control framework in cloud-based data storage and sharing is a crucial step towards bolstering security and flexibility. One fundamental aspect of this framework involves the integration of multi-factor authentication (MFA) mechanisms. MFA enhances security by requiring users to provide two or more types of authentication factors before they are granted access. These factors typically include something they know (e.g., a password or PIN), something they have (e.g., a mobile device or smart card), and something they are (e.g., biometric data like fingerprints or facial recognition).

By demanding multiple forms of authentication, MFA significantly reduces the risk of unauthorised access, as even if one factor is compromised, an attacker would still need to bypass the additional layers of security. This makes it extremely challenging for malicious actors to gain access to sensitive data, even if they manage to obtain one piece of the authentication puzzle. So we used enhanced CP-ABE algorithm.

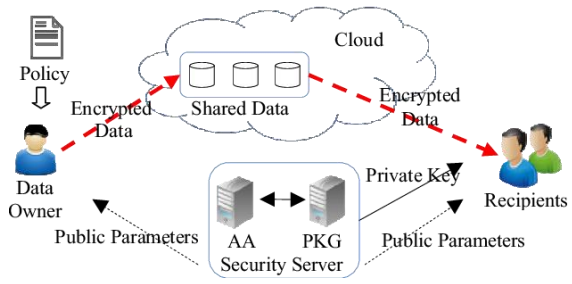


Figure 1.4.1. Architecture

1.5. Objective Of The Project

The objective of the project is to design, implement, and evaluate a dual access control system for cloud-based data storage and sharing. This system will include a robust authentication mechanism requiring two levels of authentication, such as password and biometric authentication, as well as an authorization logic allowing users to specify access control policies based on attributes like user roles and data sensitivity levels. The project will focus on ensuring secure communication between users and the cloud storage system, implementing a user-friendly interface for managing access control settings, and evaluating the system's performance in terms of authentication speed, access control enforcement, and scalability. Additionally, a security analysis will be conducted to identify and mitigate potential vulnerabilities, and a user study will assess the system's usability and user satisfaction, aiming to contribute to secure data storage and sharing practices in cloud environments.

2. Literature Survey

In recent years, the adoption of cloud-based storage solutions has seen exponential growth due to their significant advantages in cost efficiency and operational flexibility. This shift has been especially pronounced within both the academic community and various industries, where the demand for scalable, accessible, and cost-effective data storage solutions is ever-increasing. Such platforms, exemplified by services like Apple iCloud, offer compelling benefits, including enhanced accessibility to data and elimination of the need for extensive local data management infrastructure. As a result, a rising number of entities, ranging from individuals to large corporations, are moving towards outsourcing their data storage needs to cloud services. This transition not only facilitates significant cost savings by reducing the need for continuous investment in local data storage and management hardware but also enables users to access their data seamlessly from any location. Despite these advantages, the migration to cloud-based services is not without its challenges. Chief

among these is the concern regarding data security breaches, a critical issue that serves as a considerable deterrent to the wider adoption of cloud storage services. The fundamental issue stems from the inherent risk associated with storing sensitive information on remotely hosted platforms, which, by their very nature, are susceptible to various security threats. A particular area of concern is the sharing of outsourced data, a common requirement in many practical applications. For instance, a user of a service like Dropbox might wish to share photos with friends. Traditional sharing mechanisms, while providing a level of access control, often do not employ encryption, thereby exposing shared links to potential unauthorized access at the administrative level. To counteract this vulnerability, it's recommended to encrypt data before uploading it to the cloud. This encryption ensures that data privacy and security are maintained, as only users possessing the appropriate decryption keys can access the data. One popular encryption method is the use of Advanced Encryption Standard (AES) for encrypting data before its cloud storage. However, challenges remain, especially when the need arises to share encrypted data with users whose identities or specific attributes are not known in advance. Traditional public key encryption methods, such as Paillier Encryption, require prior knowledge of the data receiver's identity, making them unsuitable for scenarios where data needs to be shared based on attributes rather than identities. This gap necessitates the adoption of policy-based encryption mechanisms, allowing data owners to define access policies directly on encrypted data. This approach ensures that only authorized users, who meet the defined attributes or policies, can access the encrypted data. Furthermore, the open nature of cloud services exposes them to various types of attacks, notably resource-exhaustion attacks. These attacks, aimed at overwhelming the cloud server with excessive download requests, can lead to denial-of-service conditions, disrupting access for legitimate users. Such attacks not only impact service availability but also pose significant economic concerns in a pay-as-you-go model, as they can dramatically increase the costs for cloud service users. This phenomenon, known as the Economic Denial of Sustainability (EDoS) attack, highlights the necessity for effective control mechanisms over download requests to prevent unauthorized access and ensure the economic viability of cloud services. In response to these challenges, the paper proposes a novel dual access control mechanism that leverages the strengths of Ciphertext-Policy Attribute-Based Encryption (CP-ABE). This mechanism offers a robust framework for securing outsourced data in cloud storage, providing both

confidentiality and fine-grained control over access to encrypted data. By employing CP-ABE, data owners can define explicit access policies that are enforced directly on the encrypted data, ensuring that only authorized users with matching attributes can access the data. This solution addresses the core security concerns associated with cloud-based storage services, offering a promising approach to enhancing data privacy, security, and overall system resilience against potential attacks. We positively address the query by introducing two secure and effective systems for dual access control within cloud environments, tailored for various situations. The essence of our approach is to provide an optimized method for dual access control, beginning with the foundation of a Ciphertext-Policy Attribute-Based Encryption (CP-ABE) system, identified as a key component. Additionally, we innovate by implementing a strategic oversight on users' requests to download data, refining the process beyond the conventional "testing" ciphertext approach. Specifically, we enable users to initiate download requests. Upon receipt of such requests, the cloud server, with support from either a designated authority or Intel SGX enclave, verifies the user's eligibility for data access without divulging any extra details aside from the user's authorization status. This underlying framework allows the cloud to effectively manage download requests. The introduced systems are distinguished by several notable attributes: 1. Data Privacy Assurance:Our systems ensure that data encrypted and uploaded to the cloud remains accessible exclusively to those with legitimate authorization. 2. Anonymity in Data Sharing: The cloud server is unable to identify the owner of the data being shared, preserving
the data owner's anonymity throughout the storage and sharing process. 3. Granular Access Control: The owner of the data maintains control over their encrypted information by setting specific access policies before cloud upload. This allows only users who meet these criteria to access the encrypted data. 4. Anonymous Download Request Management and Protection Against EDoS Attacks: The cloud server has the capability to manage download requests anonymously, thereby safeguarding the system against Economic Denial of Sustainability (EDoS) attacks. 5. Operational Efficiency: By leveraging the CP-ABE system as a foundation, our proposed solutions do not significantly increase computational or communication burdens, making them practical for real-world deployment when compared with existing systems.

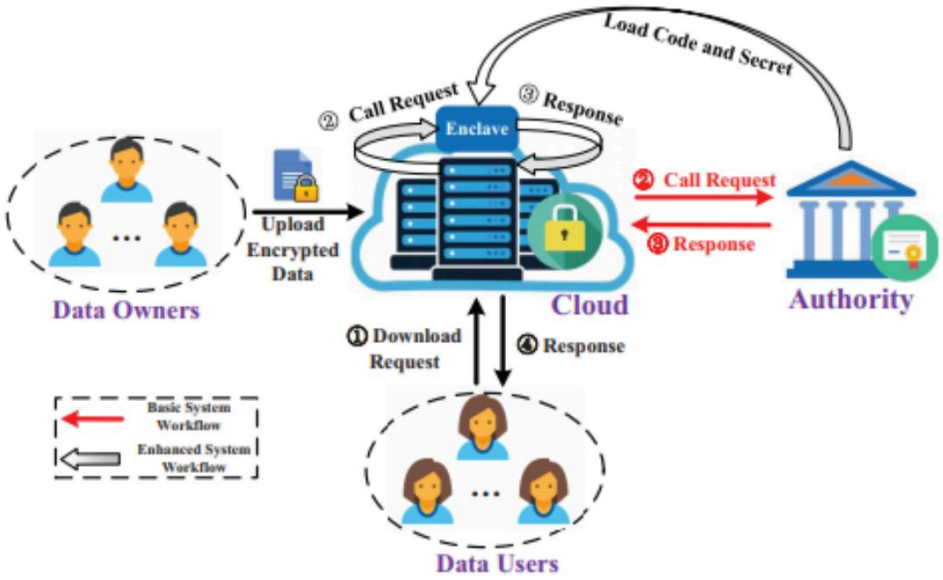


Figure 2.1. Access Flowchart

3. Dual Access Control For Cloud Based Data Storage And Sharing

The proposed method outlined dual access control in cloud-based data storage and sharing through the use of encryption and access control mechanisms. This approach focuses on ensuring data confidentiality and service user privacy in an open network environment. Key components and mechanisms involved:

3.1. Notation:

Let PPT be probabilistic polynomial-time. Define [k] = {1, 2, ..., k} for k $\in$ N. Let (a1, a2, ..., an) be a row vector and (a1, a2, ..., an)$\perp$ be a column vector. By vi we denote the i-th element in a vector ~v. Let G = (G, GT , p, e) be the groups and the bilinear mapping description, where G and GT are two multiplicative cyclic groups of prime order p and e : G × G → GT is a bilinear map.

3.2. Prime Order Bi-linear Groups:

Let G and GT be two multiplicative cyclic groups of prime order p. Let g be a generator of G and e : G × G → GT be a bilinear map. The bilinear map e has the following properties: (1) Bilinearity: ∀u, v ∈ G and x, y ∈ Zp, we have e(ux , vy) = e(u, v) xy; (2) Non-degeneracy: e(g, g)6= 1. Wesay that G is a bilinear group if the group operations in G and the bilinear map e : G×G → GT can both be computed efficiently.

3.3. Complexity Assumption:

Assumption 1. (Decisional q-parallel Bilinear Diffifie-Hellman Exponent assumption (decisional q-Parallel BDHE) [36]) The Decisional q-parallel Bilinear Diffie-Hellman Exponent problem as follows. Initially choose a group G of prime order p according to the security parameter, pick a random group element g ∈ G, and q + 2 random exponents c, s, b1, b2, ..., bq ∈ Zp. If an adversary is given the group description (p, G, GT , e) and ~z including the following terms: g, gs gc , ..., gcq , gcq+2 , ..., gc2q gs·bj , gc/bj , ..., g(cq/bj) , g(cq+2/bj) , ..., g(c2q/bj) ∀1≤j≤q gc·s·bk/bj , ..., g(cq·s·bk/bj) ∀1≤j,k≤q,k 6=j it is hard for the adversary to distinguish e(g, g) scq+1 ∈ GT from an element R which is randomly chosen from GT . An algorithm A that outputs β ∈ {0, 1} has advantage in solving the above assumption if |Pr[A(~z, e(g, g) scq+1) = 0] − Pr[A(~z, R) = 0]| ≥ .

Definition 1. We say that the decisional q-Parallel BDHE assumption holds if no P P T algorithm has a non-negligible advantage in solving the decisional q-Parallel BDHE problem.

3.4. Ciphertext-Policy Attribute-based-Encryption:

Ciphertext-Policy Attribute-based-Encryption (CP-ABE) is a versatile encryption supporting fifine-grained access control over encrypted data. In a CP-ABE system, each data user is issued with a secret key according to his attributes. A data owner can

choose an access structure A and encrypt his data under A. The encrypted file can be decrypted by any data user whose attribute set satisfies A. CP-ABE systems proposed in recent years usually make essential use of linear secret-sharing schemes. The definitions of access structure and linear secret-sharing schemes are shown as follows.

Access Structure:

Let S denote an attribute uni verse. A collection A ⊆ 2S is called monotone if ∀B, C ∈ A : if B ∈ A and B ⊆ C, then C ∈ A. A collection (respectively, monotone collection) A ⊆ 2S of non-empty subsets of S is an access structure (respectively, monotone access structure) on S. The sets in A are called authorized sets, and the sets not in A are called the unauthorized sets.

Linear Secret-Sharing Schemes (LSSS)):

Let S be an attribute universe and p be a prime. A secret-sharing scheme Q over S is called linear (over Zp) if (1) The shares of a secret s ∈ Zp for each attribute form a vector over Zp; (2) For each access structure A on S, there exists a matrix M with l rows and n columns called the share-generating matrix for Q . For i = 1, ..., l, we define a function ρ labels row i of M with attribute ρ(i) from S. When we consider the column vector ~v = (s, r2, ..., rn), where s ∈ Zp is the secret to be shared and r2, ..., rn ∈ Zp are randomly chosen. Then M~v ∈ Z l×1 p is the vector of l shares of the secret s according to Q . The share (M~v)j "belongs" to the attribute ρ(j) for j ∈ [l]. A CP-ABE system consists of four algorithms the follow
ing four algorithms:

• **Setup(λ, U).**

The setup algorithm takes as input a security parameter λ and attribute universe U, and outputs a master secret key MSK and the public parameters P P.

• **Encrypt(P P, A, M).**

The encryption algorithm takes as input the public parameters P P, an access structure A and a message M, and outputs a ciphertext CT.

• **KeyGen(MSK, S).**

The key generation algorithm takes as input the master secret key MSK and an attribute set S, and outputs a secret key SK.

• **Decrypt(P P, SK, CT).**

The decryption algorithm takes as input the public parameters P P, a secret key SK and a ciphertext CT. If the attribute set of SK satisfies the access structure of CT, it outputs a message M; otherwise, it outputs $\perp$. The definition of CP-ABE's security can be found in [36], which achieves indistinguishability under chosen-plaintext attacks (i.e., is IND-CPA secure).

3.5. Authenticated Encryption with Associated Data:

Authenticated encryption with associated data (AEAD) is a form of symmetric-key encryption which simultaneous ly provides confidentiality as well as integrity [28]. A symmetric-key encryption scheme SE mainly consists of the following two PPT algorithms:

• SE.Enc(m, sk) → ct: On input a message m and a symmetric key sk, it outputs a ciphertext ct.

• SE.Dec(ct, sk) → m: On input a symmetric key sk and a ciphertext ct, it outputs a message m. An symmetric-key encryption scheme SE should be se semantically secure under a chosen plaintext attack.

3.6. Intel SGX:

Intel Software Guard Extensions (SGX) is a set of new instructions available on recent-model Intel CPUs that allow for the creation of isolated execution environments called enclaves [19]. Our systems build on the notion of enclave, which is designed to run code and handle secrets in a trustworthy manner, even on a host where the system memory and OS are untrusted. The enclave provides three main security properties: isolation, sealing, and attestation. Isolation restricts access to a hardware guarded area of memory such that only that particular enclave can access it. Any other process on the same processor, even the OS, hypervisor, cannot access that memory. Sealing provides a way of encrypting enclave secrets for persistent storage to disk such that the secrets can be retrieved even if the enclave is torn down. Encryption is performed using a private seal key that is unique to that particular

enclave, no process other than the exact same enclave can decrypt (or modify) it. Attestation enables an entity to verify that the desired code is indeed running securely and unmodified within the enclave. In particular, there are two forms of attestation: local attestation and remote attestation [7]. Local attestation is used for attestation between two enclaves on the same platform. The two enclaves on the same machine can derive a shared key, called Report Key, using the Root Seal Key shared between them. Remote attestation enables an enclave to generate a report that can be verified by any remote entity. Specifically, in order to generate a quote, an enclave first attests to a special enclave called the Quoting Enclave locally and sends it a report. After verifying the received report, the Quoting Enclave converts it into a quote, which contains the same underlying data. Essentially, the quote is signed with a secret key for an anonymous group signature scheme called Intel Enhanced Privacy ID (EPID) [7], [13]. The signature generated from EPID can be essentially verified by using the group public key.

We utilize a hybrid security system that integrates the speed of symmetric-key encryption with the flexibility of public-key encryption. This approach involves two layers of access control in a Key/Data Encapsulation Mechanism (KEM/DEM) framework. The bulk of the message is encrypted using a fast symmetric-key algorithm, while a slower public-key method, specifically CP-ABE, is employed solely for encrypting and decrypting a small key.

For ensuring the privacy of shared data, maintaining its confidentiality, and controlling access, we rely on the CP-ABE method as the core component. We adopt a CP-ABE scheme known for its efficiency and sophisticated design to meet these security needs. Moreover, to cater to the anonymous requests for data downloads and manage access without compromising sensitive details (such as the user's identity or the data's unencrypted form), we have devised a system where the cloud can verify a user's authorization status with no need to access private information. Initially, this verification process requires assistance from an authority, necessitating its constant availability. Nonetheless, we acknowledge that this may not be feasible in all real-world scenarios, leading us to consider alternative approaches in certain cases.

In scenarios where maintaining constant online presence of the authority is not feasible, we introduce an improved model. This model permits the authority to go offline after setting up the necessary parameters. To facilitate this, we incorporate the SGX (Software Guard Extensions) technology, which effectively takes over the authority's role in managing access control during download requests.

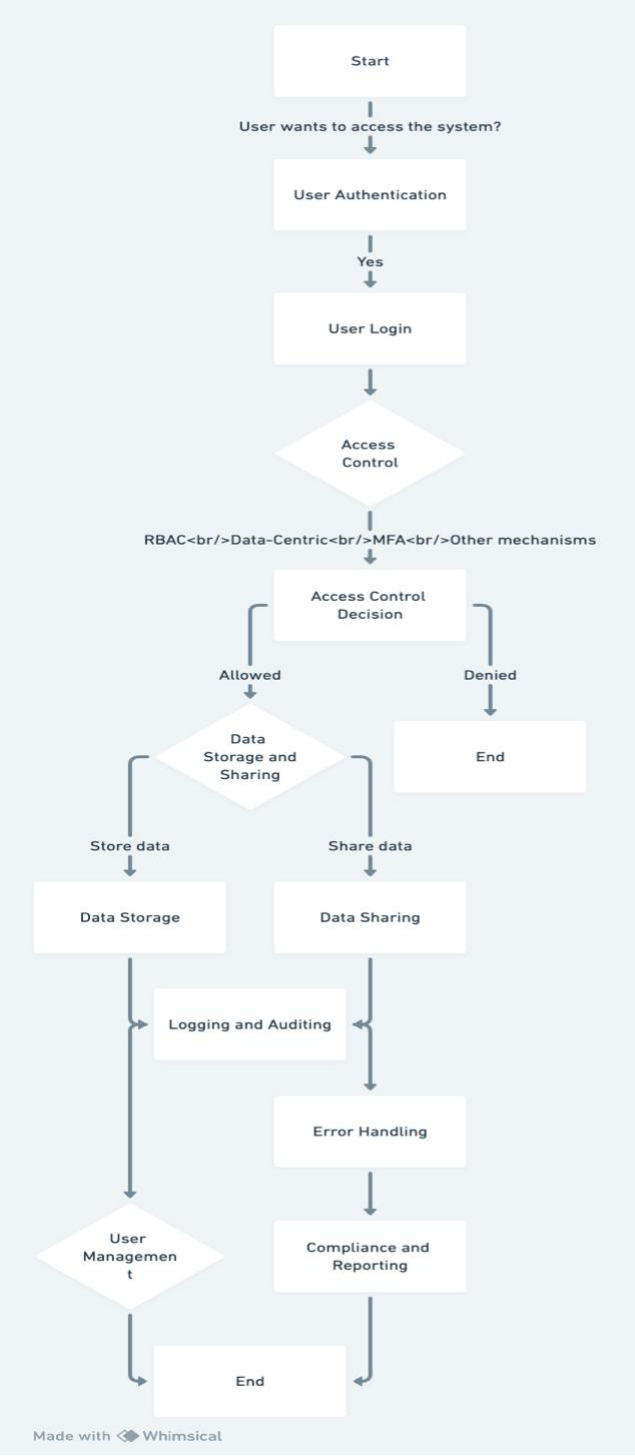


Figure 3.6.1. Overall Flowchart

The foundation of our systems is aimed at ensuring robust security and privacy for data shared on the cloud, protecting against threats such as EDoS attacks. Our strategy involves implementing a dual access control mechanism, as outlined earlier. We build upon the CP-ABE framework suggested in prior studies, adjusting it for use in a KEM/DEM configuration. However, integrating CP-ABE directly into KEM/DEM does not fully address the need for dual access control. Therefore, we introduce an innovative method that eliminates the need for the "testing" ciphertext used in preliminary models. This method involves the data owner creating a download request that incorporates a randomized version of their secret key. This key retains its ability to decrypt, allowing the cloud to verify if the requestor has the authority to access the encrypted data without revealing the data owner's identity. Hence, the download request serves as an anonymous means to ascertain the legitimacy of the data owner's access rights without compromising their anonymity. To ensure the cloud does not access sensitive information during this verification process, either an authority's intervention or the use of an Intel SGX enclave is necessary. Our first system relies on the authority's participation for this verification, whereas our second system utilizes Intel SGX for a more autonomous approach. This technique is versatile and can be adapted to most existing CP-ABE frameworks that employ bilinear maps. The architectures of our dual access control systems for Cloud data sharing is shown in Fig. Concretely, the systems consist of the following entities:

•Authority is responsible for initializing system parameters and data user registration. Also, it handles the call request from the cloud in the first proposed construction.

• Data owner holds the data and wants to outsource his data to the cloud. In particular, data owners (only) want to share their data with those who satisfy certain conditions (e.g., professors or associate professors). They will be offline once their data have been uploaded to the cloud.

• Data user wants to download and decrypt the encrypted data shared in the cloud. Those who are authorized can download the encrypted file and further decrypt it to access the plaintext.

• Cloud provides convenient storage service for data owners and data users. Specifically, it stores the outsourced data from data users and handles the download requests sent by data users.

• Enclave handles the call request from the cloud (used in the second system). The description of workflow is introduced as follows. Data owners encrypt their data under the access policies chosen by themselves and upload the encrypted data to the cloud. Authorized data users can download the shared data by sending a download request to the cloud. Upon receiving a download request from an authorized data user (see ① in Fig.), the cloud does as follows. (a) For our basic system, the cloud sends a call request to the authority (see red ② between the cloud and the authority in Fig.). After receiving a response from the authority (see red ③ between the cloud and the authority in Fig.), the cloud sends a response back to the data user (see ④ in Fig.). (b) For our enhanced system, the cloud sends a call request to the enclave (see black ② above the cloud in Fig.). After receiving a response from the enclave (see black ③ above the cloud in Fig.), the cloud sends a response back to the data user (see ④ in Fig.). In the basic construction, the authority must be always online. It is desirable that the cloud can check the download request by itself. In this subsection, to To address this issue, we present an enhanced system. The procedures Data User Registration, Shared File Generation and Outsourcing, Download Request Generation, Access Shared Data are the same as those of the basic system, the remaining algorithms are modified as follows.

• Parameter Initialization: This procedure is almost the same with that of the basic system, excepting for the following additional steps (that follows the last step of the basic system): – The cloud equipped with SGX processors creates an enclave 4. – The authority prepares a SGX program C for realizing the following functionality: Upon receiving an input h, compute E01 = (h)s0 and output E01 , where s0 is the internal secret inside an enclave. – The authority establishes a secure channel with the enclave, and securely loads the code of program C and the master secret parameter a to the enclave, using for instance AES-GCM for confidentiality and integrity protection [26] (In particular, the authority uses a randomly generated secret key to encrypt the code

and the data, and employs the secure channel to share the secret key with the enclave). – The enclave keeps a as its internal secret (i.e., sets a = s0). In order to verify the software running in the enclave on the cloud side, the authority uses remote attestation [2] to check the integrity of the code (i.e., the program C) and static data (i.e., the master secret parameter a) loaded into the enclave [26] (please refer to Subsection 2.6 for more details about remote attestation).

• Access Control on Download Request : The procedure is almost the same as that of the basic system, excepting for replacing the first step with the following steps: – The cloud sends a call request to the enclave with C2 (of CT) as input. – Upon receiving the call request with C2, the enclave runs program C with C2 as input (i.e., calculates E01 = (C2)a) and returns E01 to the cloud. – The cloud computes E1 = e(E01 , L02) = e(g, g)^savr. Side-channel resilience. Although the security of SGX is evolving, it is still susceptible to a number of side-channel attacks [6], [14], [30], [37]. One defense against these side channel attacks is to ensure that the enclave program is data-oblivious. That is, the program will not include control flow branches or memory access patterns that depend on the values of sensitive data [7], [13]. Another approach is to employ the technique of ORAM [27]. For the enhanced system, the only enclave operations that touch secret data are decryption operations (for loading the data via AESGCM) and the specific function (that compute
E01 = (h)s0 and output E01). In our implementation of AES-GCM, we utilize the SGX SDK cryptographic library, therefore, it is resilient to software-based side-channels (which is similar to [7]). For the function, we implemented it in a way that it achieves the property of data-oblivious (i.e., control flow branches or memory access patterns will not depend on the sensitive data). Therefore, the enhanced system is secure against side channel attack.

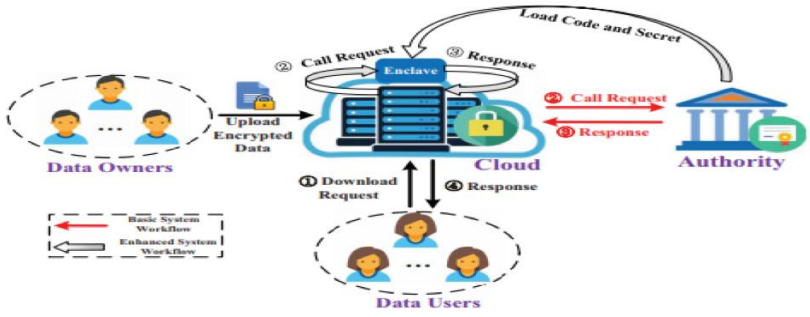


Figure 3.6.2. Request and accept

4. Implementation

We implement the two proposed systems within the Charm framework [1], where 224-bit MNT elliptic curves from Pairing-Based Cryptography library [18] are used. The experiments are performed in test beds of two PCs. The first PC plays the roles of data owner and data user, the second PC plays the role of authority and cloud. The hardware and software of the first PC are as follows: Intel Core i7-7700M CPU @3.6 GHz,Since the two proposed systems are built on the top of the CP-ABE system.

MODULES:

- Data Owner
- Data User
- Authority
- Cloud Server
- Enclave

MODULES DESCRIPTION:

Data owner:

Data owner holds the data and wants to outsource his data to the cloud. In particular, data owners only want to share their data with those who satisfy certain conditions (e.g., student, professors or principal). They will be offline once their data have been uploaded to the cloud.

Data User:

Data user wants to download and decrypt the encrypted data shared in the cloud. Those who are authorized can download the encrypted file and further decrypt it to access the plaintext.

Authority:

Authority is responsible for initializing system parameters and data user registration. Also, it handles the call request from the cloud in the first proposed construction.

Cloud Server:
Cloud provides convenient storage service for data owners and data users. Specifically, it stores the outsourced data from data users and handles the download requests sent by data users.

Enclave:
Enclave handles the call request from the cloud (used in the second system).

4.1. FUNCTIONALITY:

4.1.1. Authentication: Verify the identity of users and devices trying to access the data. This can include username/password, biometric authentication, or multi-factor authentication (MFA) to add an extra layer of security.

4.1.2. Authorization: Determine what actions users are allowed to perform on the data. This can include read, write, edit, delete, or share permissions. Access levels can be based on user roles, groups, or specific permissions assigned to individual users.

4.1.3. Encryption: Encrypt data both at rest (stored data) and in transit (data being transmitted). This ensures that even if data is intercepted or accessed without authorization, it cannot be read or understood without the decryption key.

4.1.4. Secure Sharing: Allow users to securely share data with others. This can involve generating secure links or tokens that grant temporary access to specific files or folders, ensuring that access is revoked after a specified period.

4.1.5. Scalability and Performance: Ensure that the access control mechanisms can scale to handle a large number of users and data volumes without compromising performance or security.

4.2. Attributes:

User Roles: Define different roles (e.g., owner, reader, writer) for users accessing the data.

Data Classification: Categorize data based on sensitivity (e.g., public, confidential, restricted).

Access Control Policies: Establish policies that determine who can access what data under what conditions.

Authentication Mechanisms: Use mechanisms like passwords, biometrics, or multi-factor authentication to verify user identities.

Authorization Mechanisms: Implement mechanisms (e.g., access control lists, role-based access control) to enforce access policies.

Data Encryption: Encrypt data at rest and in transit to protect it from unauthorized access.

Audit Logging: Maintain logs of access attempts and actions taken on the data for accountability and compliance purposes.

Data Masking: Mask sensitive data to protect it from unauthorized disclosure during processing or transmission.

Access Revocation: Enable the revocation of access rights when users no longer need them.

Compliance: Ensure compliance with relevant regulations and standards (e.g., GDPR, HIPAA) regarding data protection and privacy.

Confidentiality of Outsourced Data:

Existing System: The reliance on SGX for key revocation and hybrid encryption introduces potential security vulnerabilities. If the SGX enclave is compromised, the confidentiality of encrypted data is at risk.
Proposed System: By employing a dual access control mechanism with CP-ABE as a foundational layer, the proposed system enhances data confidentiality. It ensures that data is encrypted and only accessible by users who match the specified access policy, without solely relying on SGX for security.

Anonymity of Data Sharing:

Existing System: The existing system does not explicitly address the anonymity of data owners or users, which could potentially expose sensitive information about data transactions.
Proposed System: The proposed system guarantees the anonymity of the data owner and users during data storage and sharing. This is achieved by encrypting data under specific access policies and handling download requests in an anonymous manner, thereby protecting the identities of parties involved.

Fine-grained Access Control:

Existing System: The existing approach, while utilising ABE for access control, still relies heavily on SGX for revocation, which does not inherently provide fine-grained access control.

Proposed System: Introducing CP-ABE into the proposed system allows for more nuanced and fine-grained access control policies. This enables data owners to define precise criteria for who can access their data, enhancing security and flexibility.

Resistance to EDoS Attacks:

Existing System: The computational and bandwidth demands of the existing system, especially in generating and handling challenge cipher texts, could inadvertently facilitate EDoS (Economic Denial of Sustainability) attacks by overloading the cloud with expensive operations.

Proposed System: The new mechanism controls download requests more efficiently, mitigating the risk of EDoS attacks. By avoiding the generation of challenge cipher texts and instead validating download requests directly, it reduces the computational load and bandwidth usage, making the system less susceptible to exploitation.

High Efficiency:

Existing System: The need for generating challenge cipher texts and the requirement for data users to decrypt them impose significant computational and bandwidth burdens on both data owners and users.

Proposed System: By eliminating the need for challenge cipher texts and allowing for direct authorisation checks on download requests, the proposed system significantly reduces the computational and bandwidth requirements. This not only speeds up the data access process but also makes the system more scalable and user-friendly.

4.3. Experimental Screenshot

Figure 4.3.1. Home Page

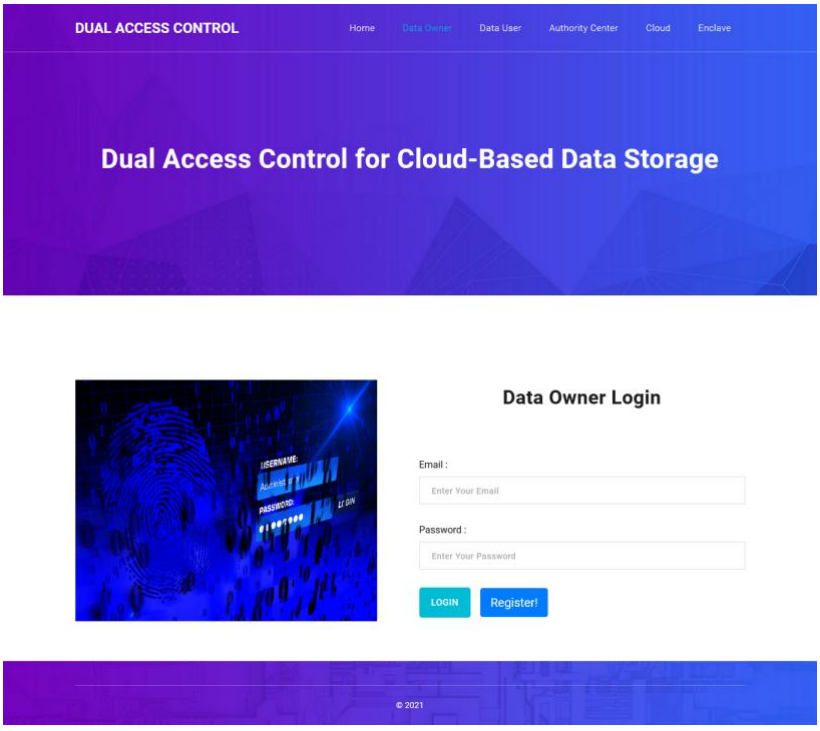
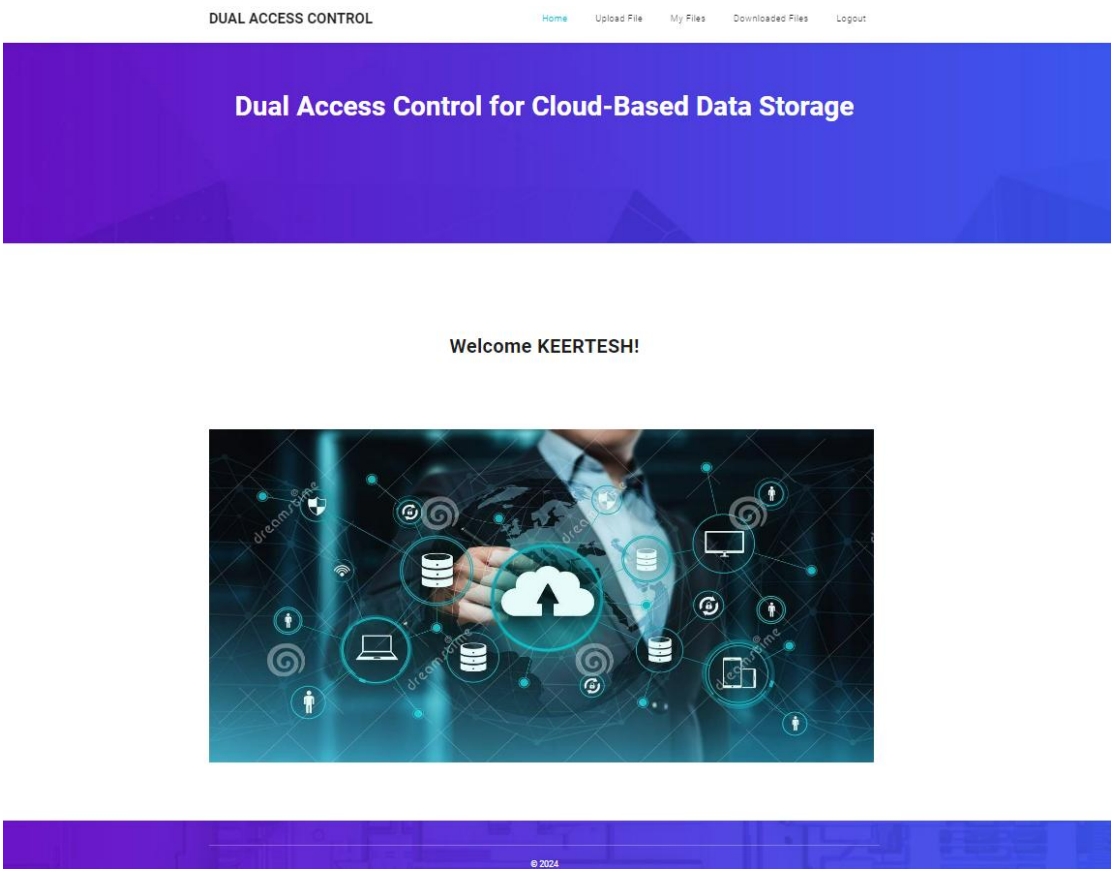


Figure 4.3.2. Owner Page



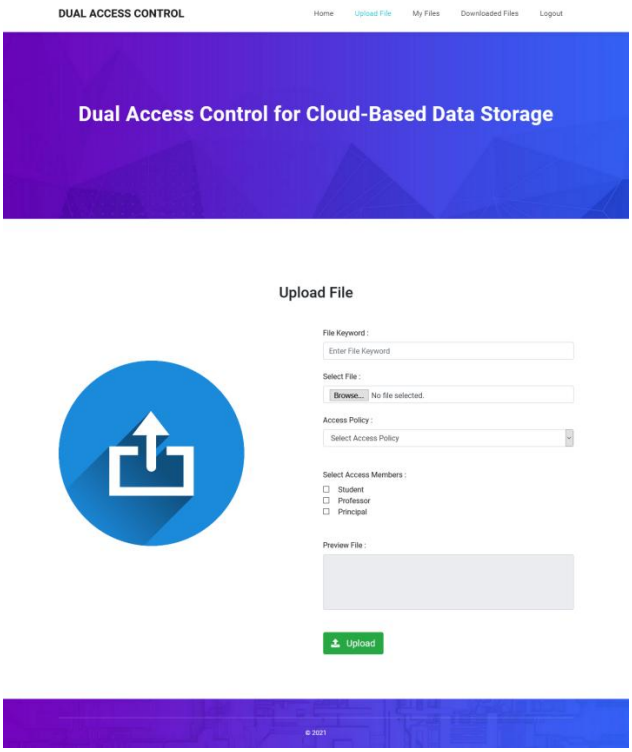


Figure 4.3.3. Owner Upload

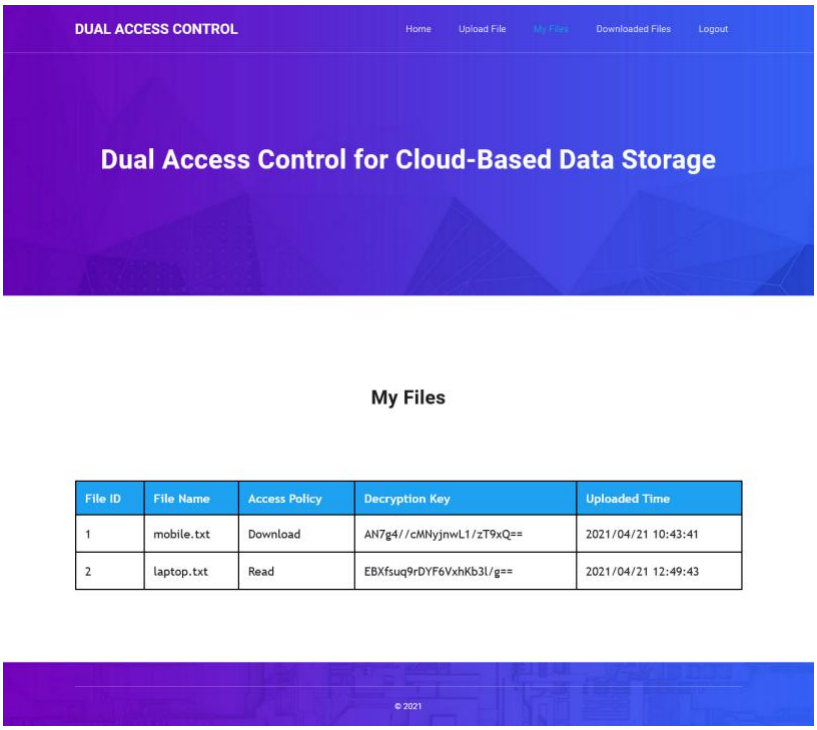


Figure 4.3.4. Access files and Requested files

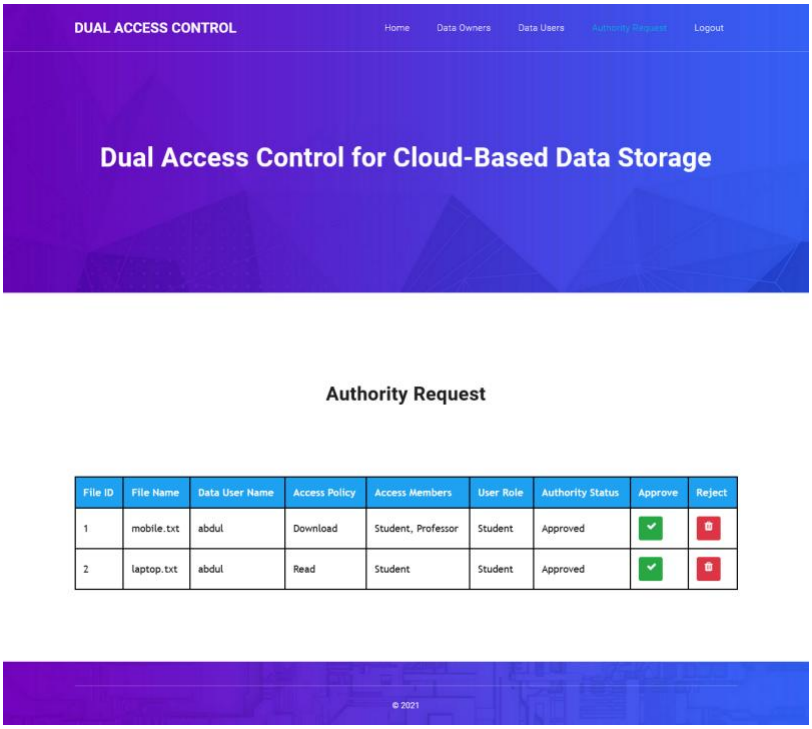


Figure 4.3.5. Authority and Analysis

5. Experimental Setup

Hardware Requirements:

System	:	Intel Core i5 Processor
RAM	:	8GB
Hard Disk	:	1 TB HDD
Input Devices	:	Keyboard, Mouse

Software Requirements:

Operating system	:	Windows 10.
Coding Language	:	JAVA.
Tool	:	Netbeans 8.2
Database	:	MYSQL
Cloud	:	Drive HQ

We use Netbeans 8.2 as the tool to access the website service, coded in javascripts and HTML,CSS and SQL for tables.Database used is MYSQL server and the cloud platform used is Drive HQ.

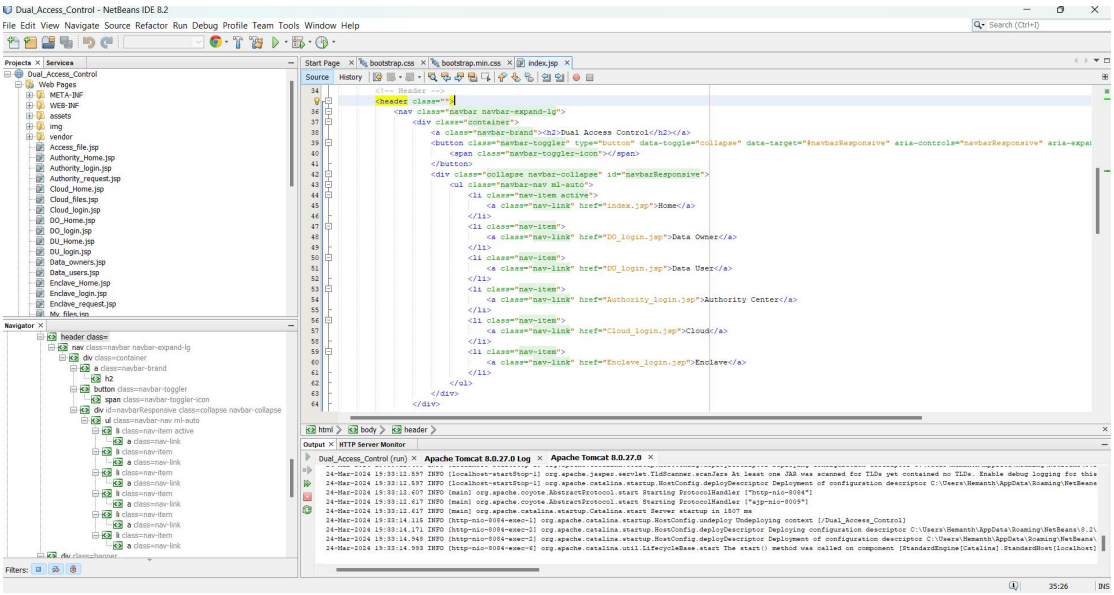


Figure 5.1. Netbeans 8.2

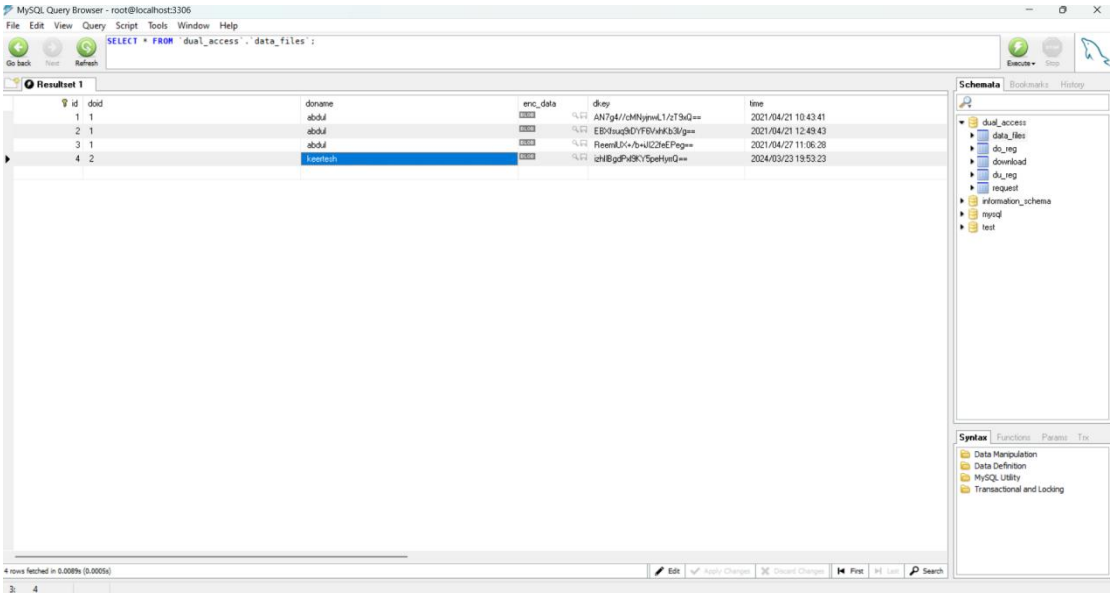


Figure 5.2. MySQL

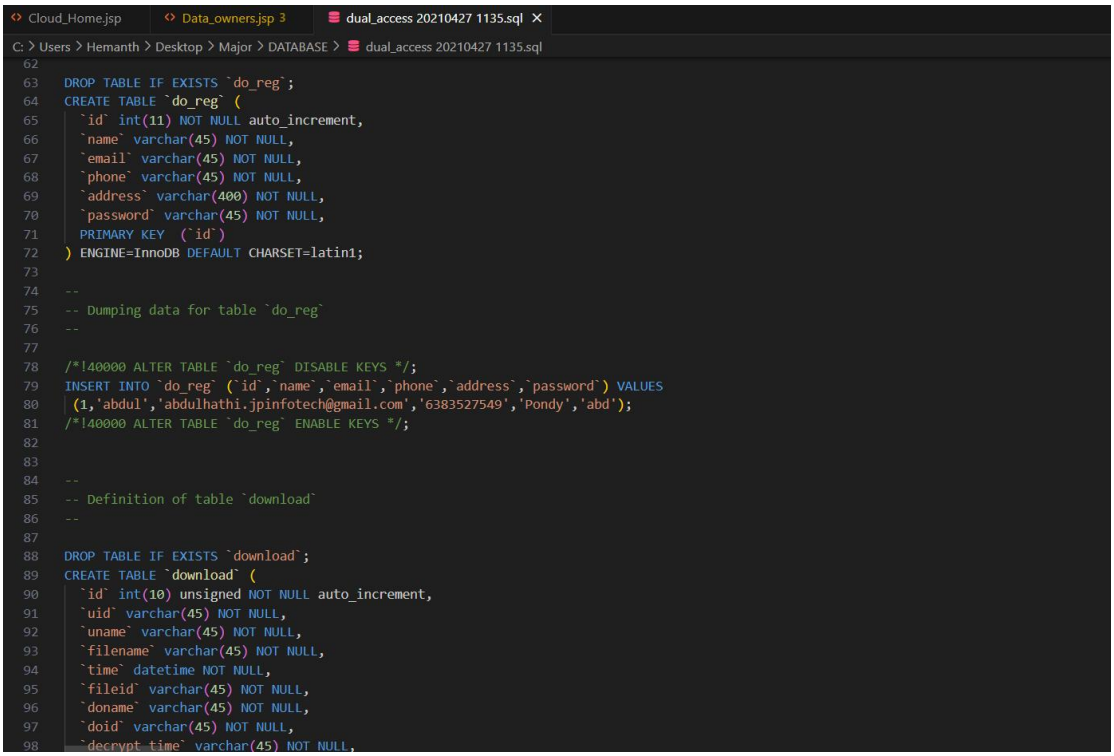


Figure 5.3. JavaScript

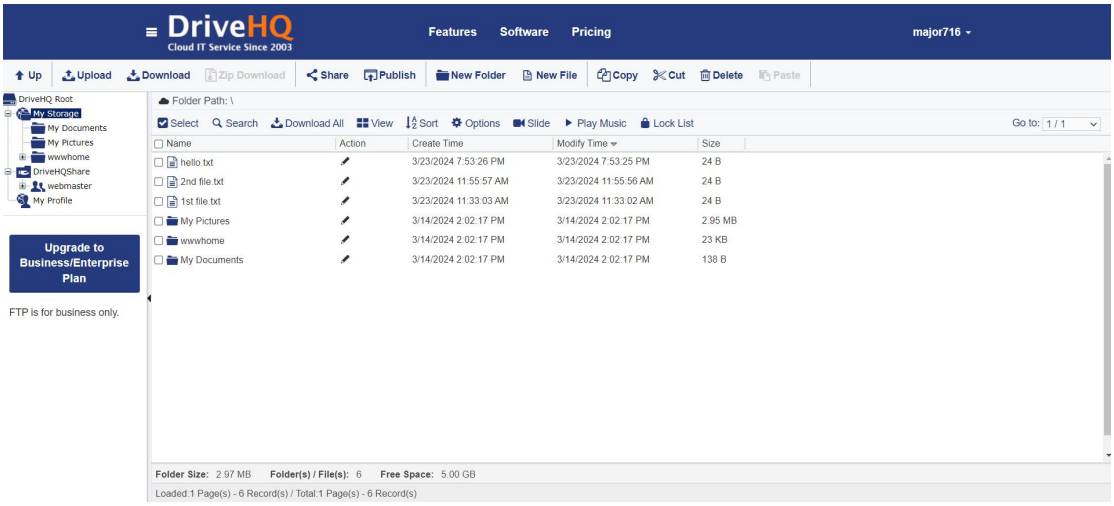


Figure 5.4. Drive HQ Cloud

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.

OBJECTIVES

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.
2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.
3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

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.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2. Select methods for presenting information.

3. Create document, report, or other formats that contain information produced by the system.

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.

Parameters:

1) Access Control Effectiveness (ACE): ACE measures how well the dual access control system prevents unauthorised access to sensitive data. It can be calculated as: ACE = (Number of Authorized Accesses) / (Total Access Attempts) * 100% This formula provides a percentage that reflects the success rate of the access control mechanism.

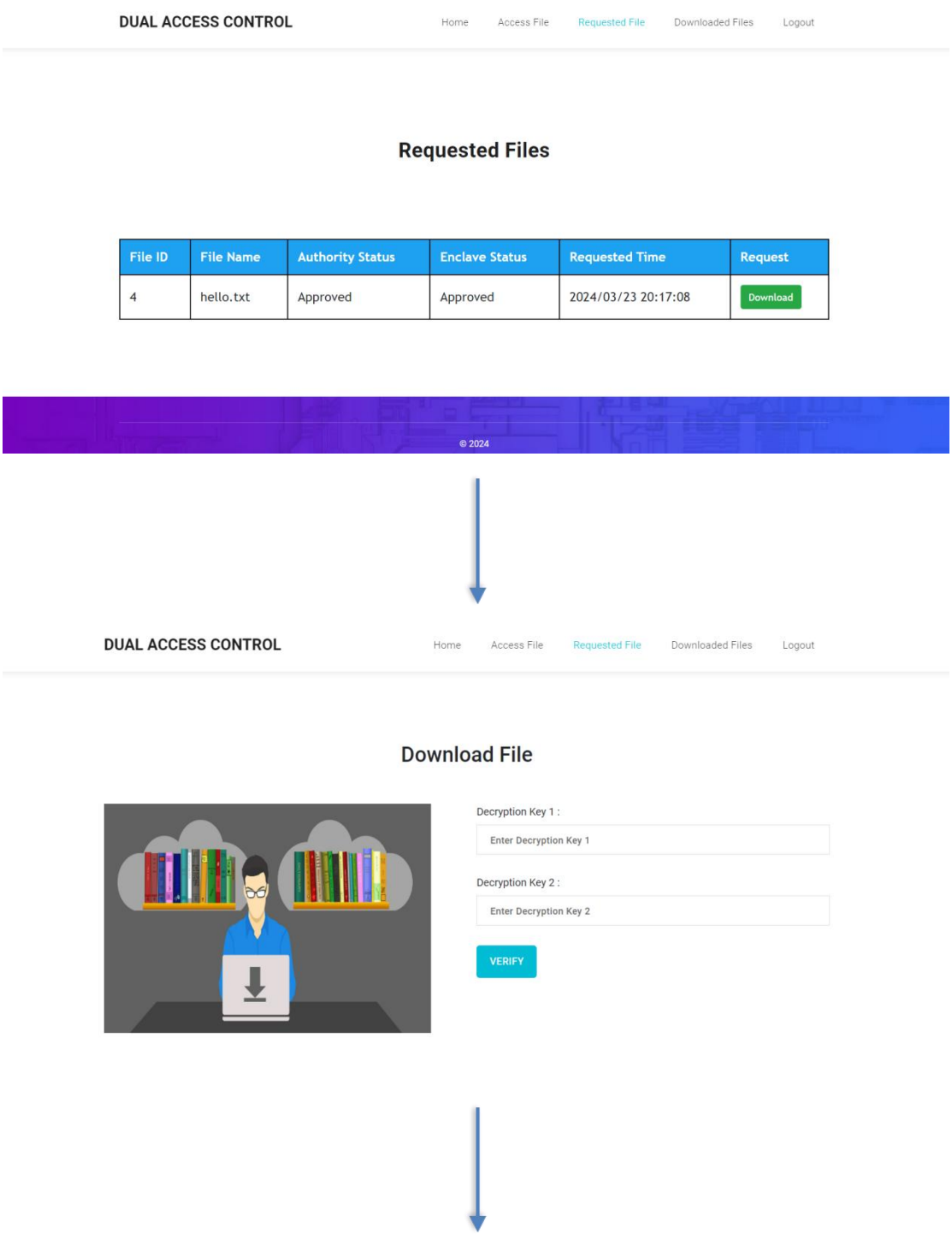
2) Data Availability (DA): Data availability is crucial to ensure users can access their data when needed. It can be calculated as: DA = (Total Uptime) / (Total Uptime + Total Downtime) * 100% This formula calculates the percentage of time when the data storage and sharing system is available.

3) Latency and Response Time (LRT): Latency and response time are important for user experience. You can calculate the average latency as: LRT = (Σ Latency measurements) / (Total number of measurements) This formula calculates the average time it takes for the system to respond to user requests.

6. Discussion of Results

Experimental Analysis To evaluate the practical performance, we implement the two proposed systems within the Charm framework [1], where 224-bit MNT elliptic curves from Pairing-Based Cryptography library [18] are used. The experiments are performed in test beds of two PCs. The first PC plays the roles of data owner and data user, the second PC plays the role of authority and cloud. The hardware and software of the first PC are as follows: Intel Core i7-7700M CPU @3.6 GHz,Since the two proposed systems are built on the top of the CP-ABE system in [36], in this subsection, we first give a theoretical analysis of the comparison between the two proposed systems and the (underlying) CP-ABE system in [36]. Let Σ0, Σ1, Σ2 be the CP-ABE system in [36], the basic system in subsection 4.2 and the enhanced system in subsection 4.3, respectively. Table 1 gives the comparison in terms of computational cost. In particular, the computational cost of Parameter Initialization of Σ1 (resp. Σ2) is the same (resp. almost the same) as the algorithm Setup(λ, U) of Σ0, excepting that it adds a into the master secret key MSK. Furthermore, the generation of secret keys of Σ1 (resp. Σ2) is the same as that of Σ0. In addition, the computational costs of encryption and decryption of Σ1 (resp. Σ2) are the same as that of Σ0 5 . That is, compared with Σ0, the two proposed systems do not impose any additional computational cost. Table 2 gives the comparison in terms of communication cost. In particular, the public parameters size, secret key size, ciphertext size of Σ1 (resp. Σ2) are all the same with that of Σ0. We note that the technique used to fulfill the feature of access control on download requests is “transplantable” to other CP-ABE. Table 3 gives the comparison among the strawman approach described in the Introduction, our proposed systems and the related work in terms of computational cost. For a fair comparison, for each computational cost in [38], we only count the computational cost which is used for access control on download requests. The computational costs for procedures Parameter Initialization, Shared File Generation and Outsourcing and access control on download request on the data user side of our proposed systems are less (or much less) than that of [38]. In contrast, the access control on download requests on the cloudside of our proposed systems require more computations.This exactly reflects the main design philosophy: to move expensive computations to the cloud as much as possible. comparison among the strawman approach described in

the Introduction, our proposed systems and the related work in terms of communication cost. For a fair comparison, we only count the communication cost in [38] which is used for access control on download requests. It shows that the communication cost for download requests of our proposed systems are less than that of [38]. In particular, the ciphertext size of our proposed systems are much less.



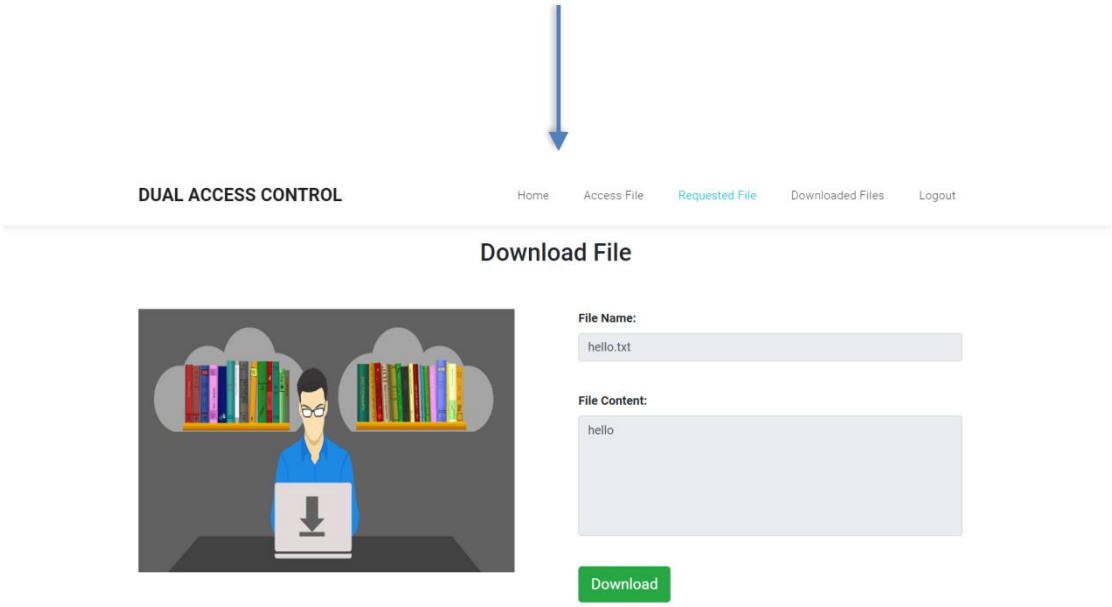


Figure 6.1. Flow of result

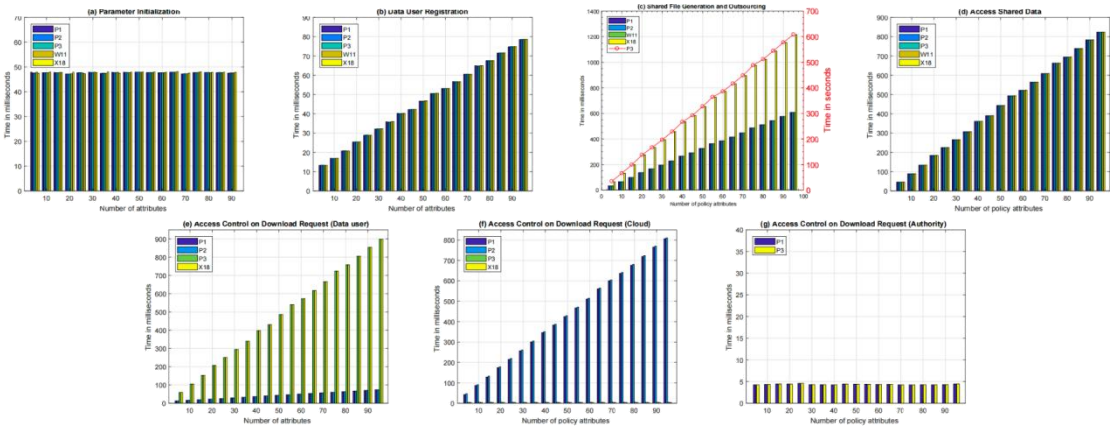


Figure 6.2. Graph Analysis

Parameter	**Previous Method (CP-ABE [36])**	**Proposed Basic System**	**Proposed Enhanced System**
Parameter Initialization (PI)	Same as CP-ABE	Same as CP-ABE	Same as CP-ABE + Intel SGX setup
Download Request Generation (DRG)	Not Applicable	Additional Cost	Additional Cost
Master Secret Key Size	Same as CP-ABE	Slightly Larger	Slightly Larger
Download Request Size	Not Applicable	Additional Size	Additional Size

Table 6.1. Comparison of previous and proposed system

EXISTING SYSTEM:

Antonis Michalas proposed a data sharing protocol that combines symmetric searchable encryption and ABE, which allows users to directly search over encrypted data. To implement the functionality of key revocation in ABE, the protocol utilizes SGX to host a revocation authority.

Bakas and Michalas later extended the protocol and proposed a hybrid encryption scheme that reduces the problem of multi-user data sharing to that of a single-user. In particular, the symmetric key used for data encryption is stored in an SGX enclave, which is encrypted with an ABE scheme. It deals with the revocation problem in the context of ABE by employing the SGX enclave.

DISADVANTAGES OF EXISTING SYSTEM:

The worry of security breach over outsourced data may be one of the main obstacles hindering Internet users from widely using cloud-based storage service.

Apart from economic loss, unlimited download itself could open a window for network attackers to observe the encrypted download data that may lead to some potential information leakage (e.g., file size).

In the existing system the data owner is required to generate a set of challenge ciphertexts in order to resist the attack, which enhances its computational burden. Second, a data user is required to decrypt one of the challenge ciphertexts as a test, which costs a plenty of expensive operations (e.g., pairing).

The computational complexity of both parties is inevitably increased and meanwhile, high network bandwidth is required for the delivery of ciphertexts.The considerable computational power of cloud is not fully
Considered.

PROPOSED SYSTEM:

In this project, we propose a new mechanism, dubbed dual access control, to tackle the existing system problem. To guarantee the confidentiality of outsourced data without loss of policy based access control, we start with a CP-ABE system, which is seen as one of the building blocks. We further employ an effective control over data users' download request on the top of the CP-ABE system. We design a new approach to avoid using the technique of "testing" ciphertext. Specifically, we allow data user to generate a download request. Upon receiving the download request, with help of the authority or the enclave of Intel SGX, a cloud server is able to check if the data user is authorized to gain access to the data. No other information is revealed to the cloud server except the knowledge of whether the user is authorized. Based on the above mechanism, the cloud maintains the control of the download request.

In our proposed systems, the outsourced data is encrypted prior to being uploaded to cloud. No one can access them without valid access rights. Given an outsourced data, cloud server cannot identify data owner, so that the anonymity of owner can be guaranteed in data storage and sharing. Data owner keeps controlling his encrypted data via access policy after uploading the data to cloud. In particular, a data owner can encrypt his outsourced data under a specified access policy such that only a group of authorized data users, matching the access policy, can access the data. A cloud server is able

to control the download request issued by any system user, where the download request can set to be anonymous. With the control over download request, we state that our systems are resistant to EDoS attacks.

ADVANTAGES OF PROPOSED SYSTEM:

Confidentiality of outsourced data

Anonymity of data sharing

Fine-grained access control over outsourced (encrypted) data

Control over anonymous download request and EDoS attacks resistance

High efficiency

Diagrams:

DATA FLOW DIAGRAM:

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.

Data Owner:

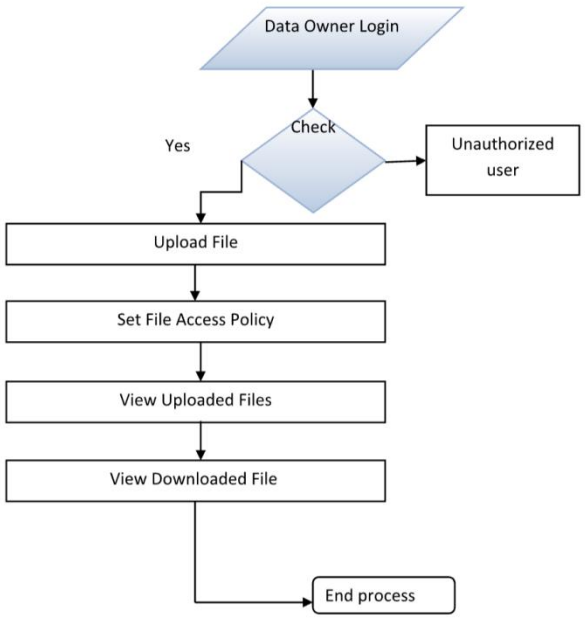


Figure 6.3. Data Owner

Data User:

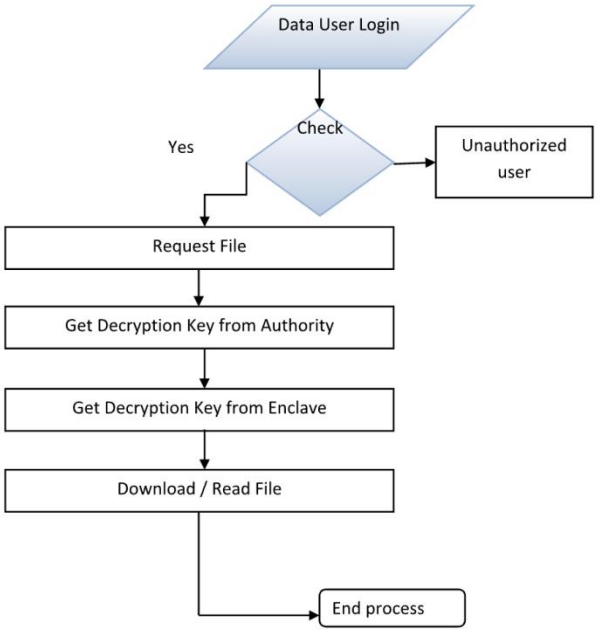


Figure 6.4. Data User

Authotity:

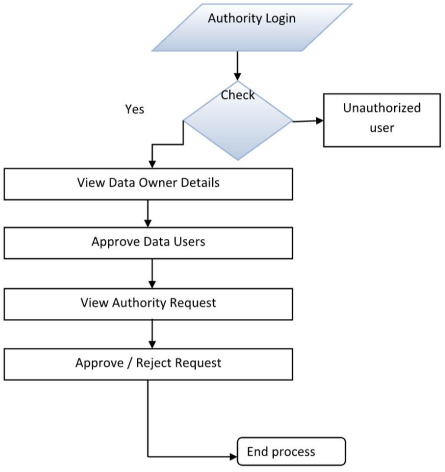


Figure 6.5. Authority

Cloud:

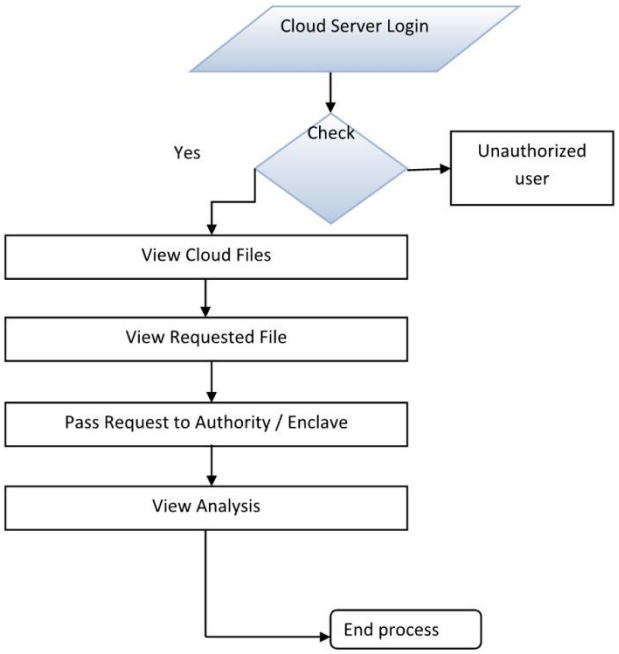


Figure 6.6. Cloud

Encalve:

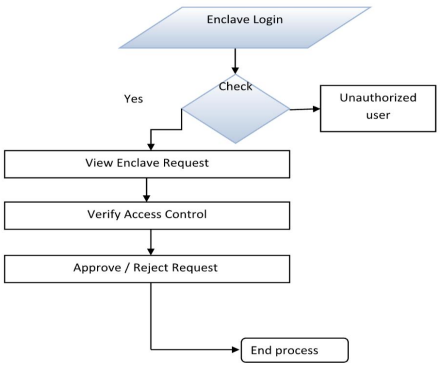


Figure 6.7. Encalve

Use Case Diagram:

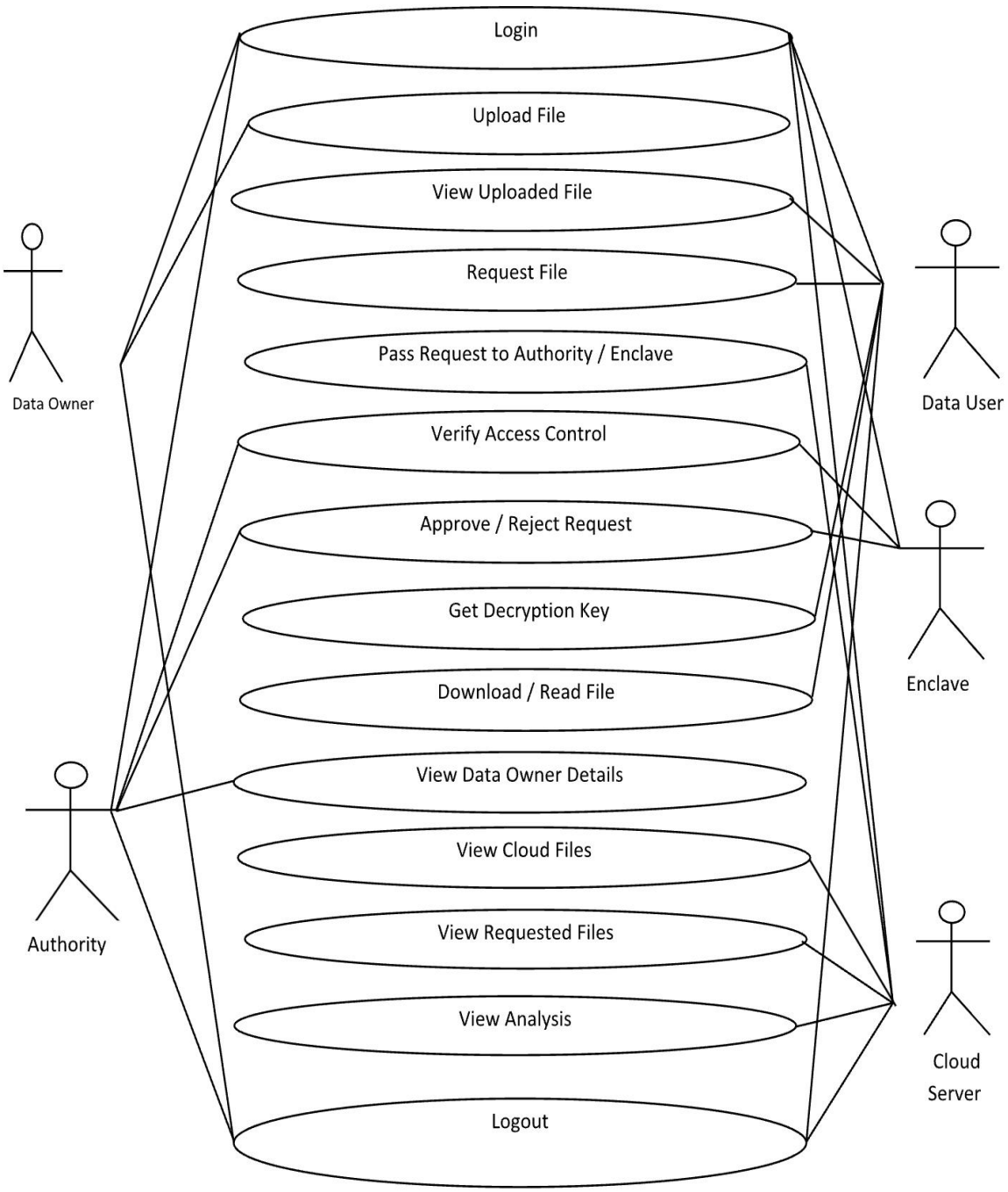


Figure 6.8. Use case diagram

Class Diagram:

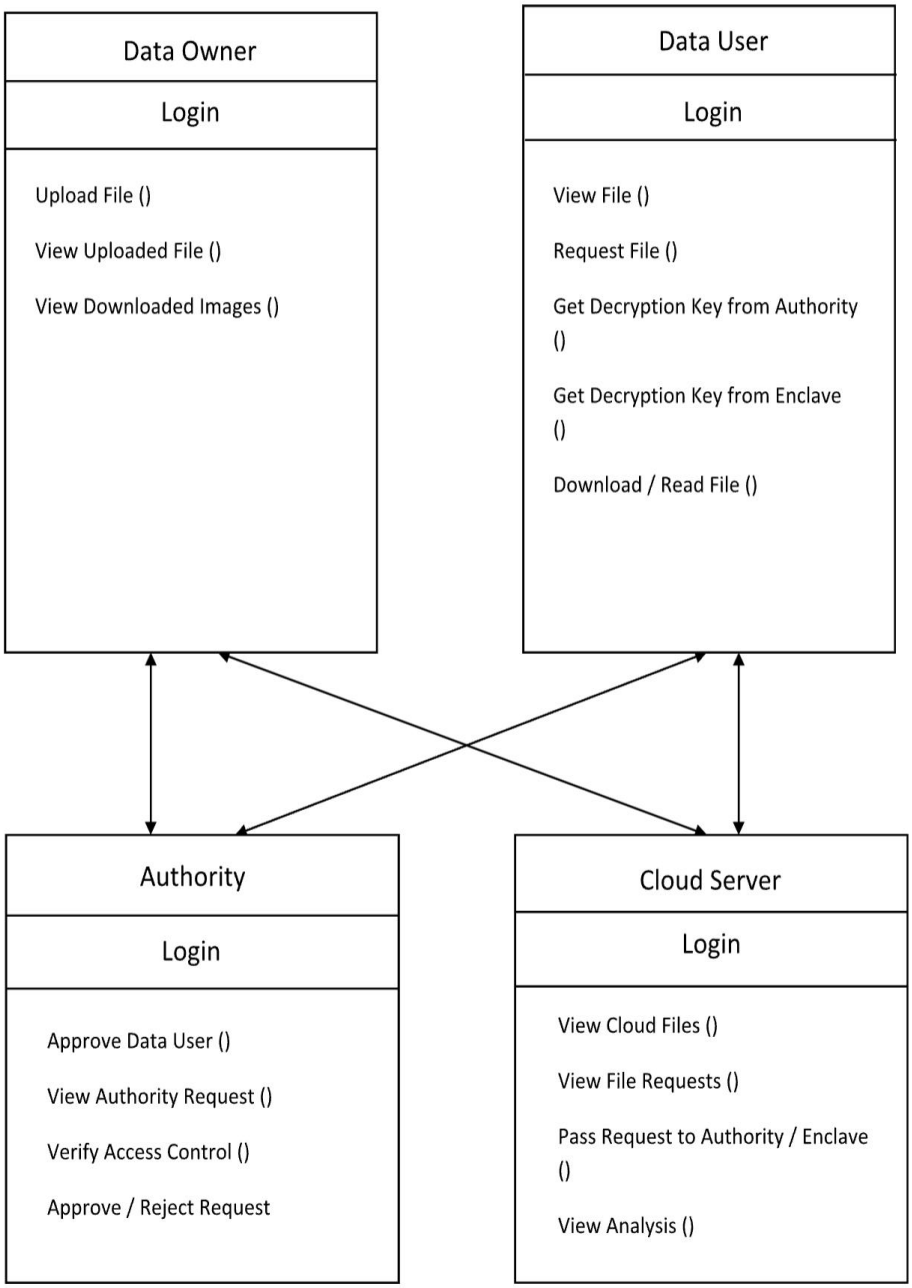


Figure 6.9. Class Diagram

Sequence Diagram:

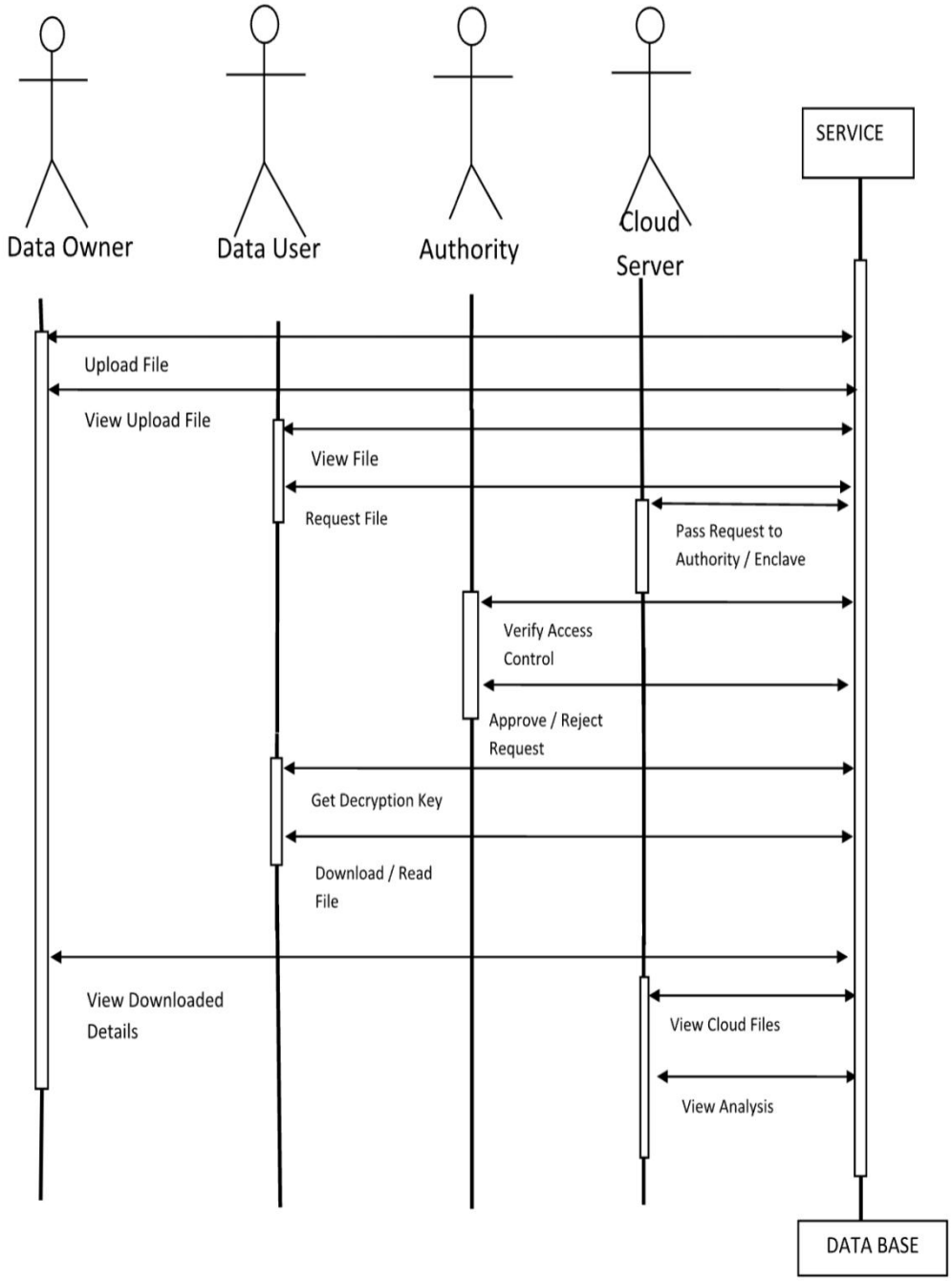


Figure 6.10. Sequence Diagram

Activity Diagram:

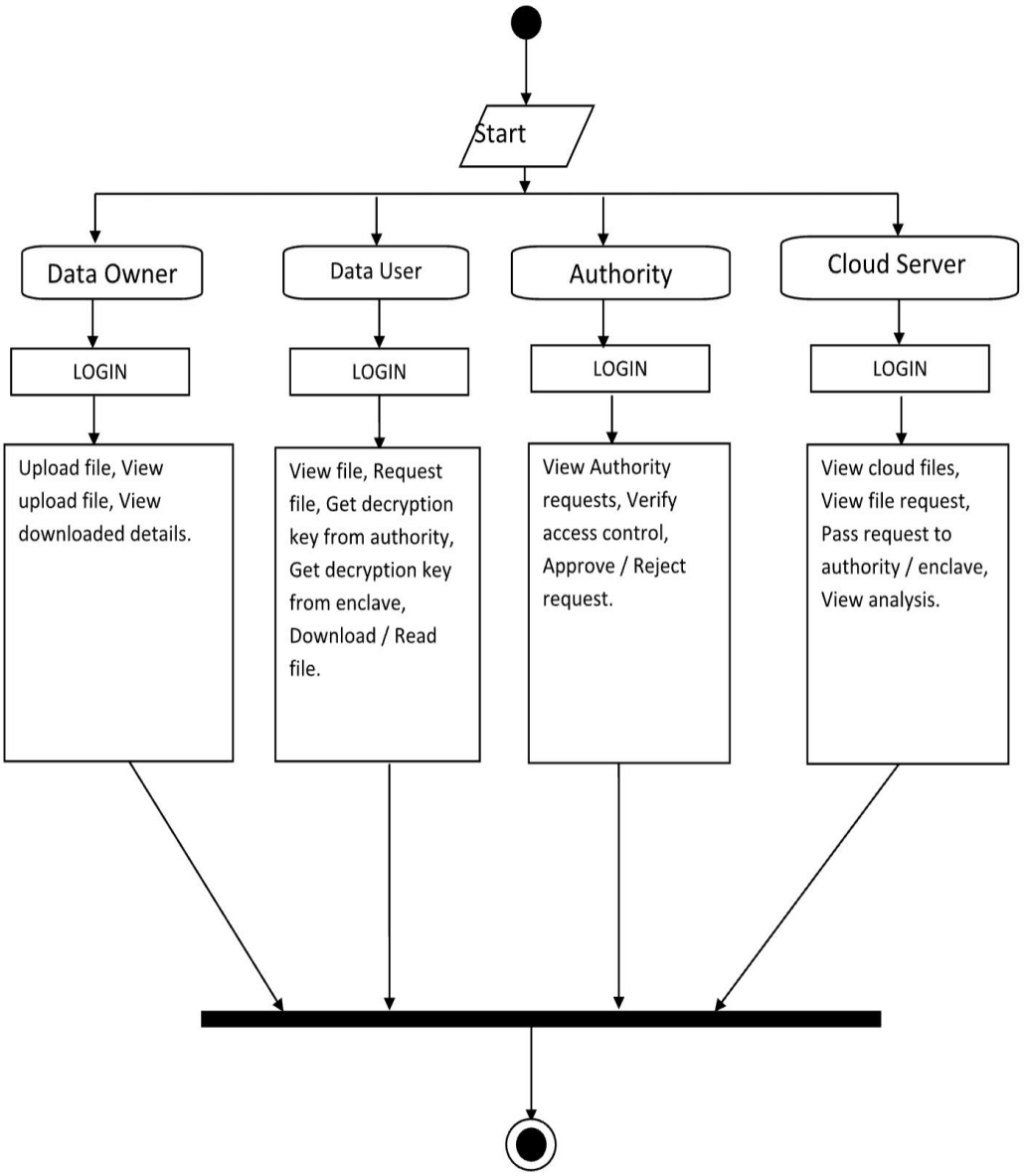


Figure 6.11. Activity Diagram

7. Summary, Conclusion And Recommendation

We addressed an interesting and long-lasting problem in cloud-based data sharing, and presented two dual access control systems. The proposed systems are resistant to DDoS/EDoS attacks. We state that the technique used to achieve the feature of control on download request is “transplantable” to other CP-ABE constructions.Our experimental results show that the proposed systems do not impose any significant computational and communication overhead (compared to its underlying CP-ABE building block).In our enhanced system, we employ the fact that the secret information loaded into the enclave cannot be extracted. However, recent work shows that enclave may leak some amounts of its secret(s) to a malicious host through the memory access patterns or other related side-channel attacks. The model of transparent enclave execution is hence introduced. Constructing a dual access control system for cloud data sharing from transparent enclave is an interesting problem. In our future work, we will consider the corresponding solution to the problem.

We tackled a persistent challenge in the realm of cloud data sharing by developing two dual access control systems designed to withstand DDoS/EDoS attacks. We highlight that the mechanism enabling control over download requests can be adapted and applied to other CP-ABE frameworks. Our tests confirm that these systems introduce minimal additional computational and communication burdens in comparison to the foundational CP-ABE elements they build upon. In our advanced system, we leverage the security feature that prevents the extraction of confidential information once it's within the enclave. However, recent studies indicate the possibility of enclaves revealing some secrets to a hostile host via memory access patterns or through other forms of side-channel attacks. This has led to the proposal of a model that ensures transparent execution within enclaves. Developing a dual access control system that operates on the principles of transparent enclave execution for cloud data sharing poses an intriguing challenge. Addressing this in our forthcoming research is a priority.

8. Future Enhancements

There are several potential future enhancements that could be considered for a project on dual access control for cloud-based data storage and sharing like :

Fine-Grained Access Control: Implement more granular access control policies to allow for more specific permissions on data and resources, such as restricting access based on time, location, or other contextual factors.

Dynamic Access Control: Enable dynamic adjustments to access permissions based on changing conditions, such as user behavior or the sensitivity of the data being accessed.

Integration with Threat Intelligence: Incorporate threat intelligence feeds to enhance access control decisions, allowing the system to dynamically respond to emerging threats.

Machine Learning for Access Control: Use machine learning algorithms to analyze access patterns and detect anomalies, enabling more proactive and adaptive access control mechanisms.

Data Masking and Redaction: Implement data masking or redaction techniques to obfuscate sensitive information when accessed by unauthorized users or in certain contexts.

Enhanced Auditing and Monitoring: Improve auditing and monitoring capabilities to provide more detailed logs and reports for better visibility into access patterns and potential security incidents.

Zero Trust Architecture: Adopt a zero-trust approach to access control, where access is never implicitly trusted and is continuously verified based on various factors.

Blockchain-Based Access Control: Explore the use of blockchain technology for maintaining access control policies and audit logs, ensuring the integrity and immutability of access control data.

Secure Multi-Party Computation: Use secure multi-party computation techniques to enable collaborative data sharing while maintaining data privacy and confidentiality.

User-Friendly Interfaces: Enhance the user interface to make it more intuitive and user-friendly, especially for managing complex access control policies and permissions.

9. Reference

[1] Joseph A Akinyele, Christina Garman, Ian Miers, Matthew W Pagano, Michael Rushanan, Matthew Green, and Aviel D Rubin. Charm: a framework for rapidly prototyping cryptosystems. Journal of Cryptographic Engineering, 3(2):111–128, 2013.

[2] Ittai Anati, Shay Gueron, Simon Johnson, and Vincent Scarlata.Innovative technology for cpu based attestation and sealing. In Workshop on hardware and architectural support for security and privacy (HASP), volume 13, page 7. ACM New York, NY, USA, 2013.

[3] Alexandros Bakas and Antonis Michalas. Modern family: A revocable hybrid encryption scheme based on attribute-based encryption, symmetric searchable encryption and SGX. In SecureComm 2019, pages 472–486, 2019.

[4] Amos Beimel. Secure schemes for secret sharing and key distribution. PhD thesis, PhD thesis, Israel Institute of Technology, Technion, Haifa, Israel, 1996.

[5] John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In S&P 2007, pages 321–334. IEEE, 2007.

[6] Victor Costan and Srinivas Devadas. Intel sgx explained. IACR Cryptology ePrint Archive, 2016(086):1–118, 2016.

[7] Ben Fisch, Dhinakaran Vinayagamurthy, Dan Boneh, and Sergey Gorbunov. IRON: functional encryption using intel SGX. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, pages 765–782, 2017.

[8] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In Advances in Cryptology-CRYPTO 1999, pages 537–554. Springer, 1999.

[9] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In ACM CCS 2006, pages 89–98. ACM, 2006.

[10] Jinguang Han, Willy Susilo, Yi Mu, Jianying Zhou, and Man Ho Allen Au. Improving privacy and security in decentralized ciphertext-policy attribute-based encryption. IEEE transactions on information forensics and security, 10(3):665–678, 2015.

[11] Christofer Hoff. Cloud computing security: From ddos (distributed denial of service) to edos (economic denial of sustainability).
http://www. rationalsurvivability.com/blog/?p=66.

[12] Joseph Idziorek, Mark Tannian, and Doug Jacobson. Attribution of fraudulent resource consumption in the cloud. In IEEE CLOUD 2012, pages 99–106. IEEE, 2012.

[13] Simon Johnson, Vinnie Scarlata, Carlos Rozas, Ernie Brickell, and Frank Mckeen. Intel R software guard extensions: Epid provisioning and attestation services. White Paper, 1:1–10, 2016.

[14] Sangho Lee, Ming-Wei Shih, Prasun Gera, Taesoo Kim, Hyesoon Kim, and Marcus Peinado. Inferring fine-grained control flow inside sgx enclaves with branch shadowing. In 26th USENIX Security Symposium, USENIX Security, pages 16–18, 2017.

[15] Jiguo Li, Xiaonan Lin, Yichen Zhang, and Jinguang Han. Ksfoabe: outsourced attribute-based encryption with keyword search function for cloud storage. IEEE Transactions on Services Computing, 10(5):715–725, 2017.

[16] Jiguo Li, Yao Wang, Yichen Zhang, and Jinguang Han. Full verifiability for outsourced decryption in attribute based encryption. IEEE Transactions on Services Computing, DOI: 10.1109/TSC.2017.2710190, 2017.

[17] Wei Li, Kaiping Xue, Yingjie Xue, and Jianan Hong. Tmacs: A robust and verifiable threshold multi-authority access control system in public cloud storage. IEEE Transactions on parallel and distributed systems, 27(5):1484–1496, 2016.

[18] Ben Lynn et al. The pairing-based cryptography library. Internet: crypto. stanford. edu/pbc/[Mar. 27, 2013], 2006.

[19] Frank McKeen, Ilya Alexandrovich, Alex Berenzon, Carlos V. Rozas, Hisham Shafi, Vedvyas Shanbhogue, and Uday R. Savagaonkar. Innovative instructions and software model for isolated execution. In HASP@ISCA 2013, page 10, 2013.

[20] Antonis Michalas. The lord of the shares: combining attributebased encryption and searchable encryption for flexible data sharing. In SAC 2019, pages 146–155, 2019.

[21] Jianting Ning, Zhenfu Cao, Xiaolei Dong, Kaitai Liang, Hui Ma, and Lifei Wei. Auditable σ-time outsourced attribute-based encryption for access control in cloud computing. IEEE Transactions on Information Forensics and Security, 13(1):94–105, 2018.

[22] Jianting Ning, Zhenfu Cao, Xiaolei Dong, and Lifei Wei. White-box traceable CP-ABE for cloud storage service: How to catch people leaking their access credentials effectively. IEEE Transactions on Dependable and Secure Computing, 15(5):883–897, 2018.

[23] Jianting Ning, Zhenfu Cao, Xiaolei Dong, Lifei Wei, and Xiaodong Lin. Large universe ciphertext-policy attribute-based encryption with white-box traceability. In Computer Security-ESORICS 2014, pages 55–72. Springer, 2014.

[24] Jianting Ning, Xiaolei Dong, Zhenfu Cao, and Lifei Wei. Accountable authority ciphertext-policy attribute-based encryption with white-box traceability and public auditing in the cloud. In Computer Security–ESORICS 2015, pages 270–289. Springer, 2015.

[25] Jianting Ning, Xiaolei Dong, Zhenfu Cao, Lifei Wei, and Xiaodong Lin. White-box traceable ciphertext-policy attribute-based encryption supporting flexible attributes. IEEE Transactions on Information Forensics and Security, 10(6):1274–1288, 2015.

[26] Olga Ohrimenko, Felix Schuster, C´edric Fournet, Aastha Mehta, Sebastian Nowozin, Kapil Vaswani, and Manuel Costa. Oblivious multi-party machine learning on trusted processors. In USENIX Security Symposium, pages 619–636, 2016.

[27] Ashay Rane, Calvin Lin, and Mohit Tiwari. Raccoon: Closing digital side-channels through obfuscated execution. In 24th USENIX Security Symposium, USENIX Security 2015, pages 431–446, 2015.

[28] Phillip Rogaway. Authenticated-encryption with associated-data. In Proceedings of the 9th ACM conference on Computer and communications security, pages 98–107. ACM, 2002.

[29] Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In Advances in Cryptology–EUROCRYPT 2005, pages 457–473. Springer, 2005.

[30] Ming-Wei Shih, Sangho Lee, Taesoo Kim, and Marcus Peinado. T-sgx: Eradicating controlled-channel attacks against enclave programs. In NDSS 2017, 2017.

[31] Victor Shoup. A proposal for an iso standard for public key encryption (version 2.1). IACR Eprint Archive, 112, 2001.

[32] Gaurav Somani, Manoj Singh Gaur, and Dheeraj Sanghi. Ddos/edos attack in cloud: affecting everyone out there! In SIN 2015, pages 169–176. ACM, 2015.

[33] Mohammed H Sqalli, Fahd Al-Haidari, and Khaled Salah. Edosshield-a two-steps mitigation technique against edos attacks in cloud computing. In UCC 2011, pages 49–56. IEEE, 2011.

[34] Willy Susilo, Peng Jiang, Fuchun Guo, Guomin Yang, Yong Yu, and Yi Mu. Eacsip: Extendable access control system with integrity protection for enhancing collaboration in the cloud. IEEE Transactions on Information Forensics and Security, 12(12):3110–3122, 2017.

[35] Florian Tramer, Fan Zhang, Huang Lin, Jean-Pierre Hubaux, Ari Juels, and Elaine Shi. Sealed-glass proofs: Using transparent enclaves to prove and sell knowledge. In EuroS&P 2017, pages 19–34. IEEE, 2017.

[36] Brent Waters. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In Public Key Cryptography–PKC 2011, pages 53–70. Springer, 2011.

[37] Yuanzhong Xu, Weidong Cui, and Marcus Peinado. Controlledchannel attacks: Deterministic side channels for untrusted operating systems. In S&P 2015, pages 640–656. IEEE, 2015.

[38] Kaiping Xue, Weikeng Chen, Wei Li, Jianan Hong, and Peilin Hong. Combining data owner-side and cloud-side access control for encrypted cloud storage. IEEE Transactions on Information Forensics and Security, 2018.

[39] Shui Yu, Yonghong Tian, Song Guo, and Dapeng Oliver Wu. Can we beat ddos attacks in clouds? IEEE Transactions on Parallel and Distributed Systems, 25(9):2245–2254, 2014.