

12-2017

Data Access in Multiauthority Cloud Storage: Expressive and Revocable Data Control System

Rajender Bhavika Pooja Gandhi
St Cloud State University, bhavikapooja9@gmail.com

Follow this and additional works at: https://repository.stcloudstate.edu/msia_etds

Recommended Citation

Gandi, Rajender Bhavika Pooja, "Data Access in Multiauthority Cloud Storage: Expressive and Revocable Data Control System" (2017). *Culminating Projects in Information Assurance*. 40.
https://repository.stcloudstate.edu/msia_etds/40

This Starred Paper is brought to you for free and open access by the Department of Information Systems at theRepository at St. Cloud State. It has been accepted for inclusion in Culminating Projects in Information Assurance by an authorized administrator of theRepository at St. Cloud State. For more information, please contact rswexelbaum@stcloudstate.edu.

Data Access in Multiauthority Cloud Storage: Expressive and Revocable

Data Control System

by

Rajender Bhavika Pooja Gandi

A Starred Paper

Submitted to the Graduate Faculty of

St. Cloud State University

in Partial Fulfillment of the Requirements

for the Degree

Master of Science

in Information Assurance

December, 2017

Starred Paper Committee:
Susantha Herath, Chairperson
Phan Dien
Balasubramanian Kasi

Abstract

Cloud computing is rising enormously due to its advantages and the adaptable storage services being provided by it. Because of this, the number of users has reached the top level. The users will share the sensitive data through the cloud. Furthermore, the user can't trust the untrusted cloud server. Subsequently, the data access control has turned out to be extremely challenging in cloud storage framework. In existing work, revocable data access control scheme proposed for multi-authority cloud storage frameworks which supports the access control in light of the authority control. The authorized users who have desirable attributes given by various authorities can access the data. However, it couldn't control the attacks which can happen to the authorized user who is not having desirable attributes. In this work, they propose a new algorithm named Improved Security Data Access Control which beats the issue exists in the existing work. And furthermore, incorporates the efficient attribute revocation strategy for multi-authority cloud storage.

Keywords: Access control, multi-authority, attribute revocation, cloud storage.

Table of Contents

	Page
List of Tables	5
List of Figures	6
Chapter	
I. Introduction	7
Introduction	7
Problem Statement	14
Highlights	15
Existing System	15
Existing Methodologies	16
Disadvantage	17
Nature and Significance of the Problem	19
Objective of the Study	35
Definition of Terms	36
Summary	36
II. Background and Review of Literature	37
Introduction	37
Background Related to the Problem	42
Literature Related to the Problem	43
Summary	45
III. Methodology	46
Introduction	46
Design of the Study	46
Attribute Revocation	47
DAC-MACS Contain Five Algorithms	49

Chapter	Page
Design Data Access Control Scheme	51
System Architecture	53
System Model (Process Flow)	53
Performance Analysis	58
UML Diagrams	60
Security Model	63
Analysis and Discussion	67
Use Case Diagram	70
Class Diagram	70
Sequence Diagram	71
Activity Diagram	72
Output Design	75
Data Collection	76
System Requirements (Minimum)	76
IV. Conclusion	80
References	82
Appendix: Screenshots	87

List of Tables

Table	Page
1. Comparisons between Different Techniques	35
2. Definition of Terms Used in This Document	36
3. Comparison between Various Data Access Control Scheme with Attribute-Based Encryption	69

List of Figures

Figure	Page
1. Framework and basic protocol flow	16
2. Another sample of framework of basic protocol flow	29
3. General flow of AnonyControl and AnonyControl scheme	50
4. The figure demonstrates the system architecture and defines the structure	53
5. System model of data access control in multi-authority cloud storage	54
6. Architecture showcasing all the key elements	63
7. Use case diagram	70
8. Class diagram	71
9. Sequence diagram	72
10. Activity diagram	73

Chapter I: Introduction

Introduction

Cloud storage is a critical administration of cloud computing, which offers services for data owners to have their data in the cloud. This new worldview of data hosting and data access control services acquaint a significant challenge with data access control. Since data owners can't entirely trust the cloud server, they can no longer depend on servers to do access control. Ciphertext-Policy Attribute-based Encryption (CP-ABE), is viewed as a standout amongst the most appropriate technologies for data access control in cloud storage frameworks, since it gives the data owners more straightforward power on access policies (Jia & Yang, 2014). In CP-ABE scheme, there is an authority that oversees attribute management and key distribution. The administration can be the enrollment office in a college, the human resource office in an organization, and so forth. The data owner characterizes the access policies and encrypts data as indicated by the systems. Every user has issued a secret key reflecting its attributes. A user can decrypt the data just when its characteristics fulfill the access policies (Jia & Yang, 2014).

There are two types of CP-ABE systems:

- Single-authority CP-ABE where a single authority oversees all attributes, and multi-authority CP-ABE where characteristics are from various areas and managed by multiple authorities.
- Multi-authority CP-ABE is more suitable for data access control of cloud storage frameworks, as users may hold attributes issued by various authorities and data

owners may likewise share the data utilizing access strategy characterized over characteristics from multiple experts. For instance, in an E-health framework, data owners may share the data using access policy “Doctor AND Researcher”, where the attribute “Doctor” issued by a medicinal association and the attribute “Researcher” issued by the overseers of a clinical trial. However, it is hard to straightforwardly apply these multi-authority CP-ABE schemes to multiauthority cloud storage frameworks because of the attribute revocation issue. In multi-authority cloud storage frameworks, users' attributes changed dynamically. A user might be entitled some new attributes or revoked some present attributes. What is more, his authorization of data access changed accordingly. Although, existing attribute revocation strategies either depend on a trusted server or absence of proficiency, they are not appropriate for managing the attribute revocation issue in data access control in multi-authority cloud storage frameworks (Nia & Yang, 2014).

- As a rule, when a user encrypts sensitive data, it is basic that she set up a specific access control policy on who can decrypt this data. For illustration, assume that the FBI public corruption offices in Knoxville and San Francisco are researching an allegation of bribery including a San Francisco lobbyist also, a Tennessee member of Congress. The head FBI specialist might need to encode a sensitive memo so that just staff that have certain accreditations or attributes can access it (Sahai & Waters, 2005).

- For example, the head specialist may determine the accompanying access structure for accessing this data: `((("Public Corruption Office" AND ("Knoxville" OR "San Francisco")) OR (management-level > 5) OR "Name: Charlie Eppes")`. By this, the head specialist could imply that the memo ought to just see by specialists who work at public corruption offices at Knoxville or San Francisco, FBI authorities high up in the management chain, and a consultant named Charlie Eppes. As represented by this case, it can be vital that the individual possessing the secret data have the Storage to pick possession of the secret data given specific knowledge of the underlying data. Besides, this individual may not know the correct characters of all other individuals who ought to have the Storage to access the data, but instead, she may just have an approach to depict them as far as descriptive attributes or credentials.
- In the ciphertext-policy attribute-based encryption scheme, each user's private key (decryption key) tied to a set of attributes representing that user's permissions. When a ciphertext is encrypted, a set of attributes designated for the encryption, and only users tied to the relevant attributes can decrypt the ciphertext.
- The example presented on the website presents a ciphertext encrypted such that only employees with the attributes "Human Resources" UNION "Executive" can decrypt it. HR employees have the "Human Resources" attribute tied to their private keys, and Executive employees have the "Executive" attribute tied to their private keys. Both groups, therefore, can decrypt the encrypted message.

- Unlike other Role-Based Access Control (RBAC) systems, CPA does not require a trusted authority or any form of storage. The encryption serves as the RBAC mechanism.
- Cpabe-keygen: The program allows a user to produce private keys associated with a set of attributes. The example presented on the website was creating two new private keys for new employees Sara and Kevin; it should note that the Master Key is required to generate these keys! It is critical that the user keep this essential private t's clear from the code above that Sara is a system administrator in the IT department, has office room 1431, and hired today. Kevin is a business staff member of the Strategy Team, has executive level 7 permissions, works in place 2362, and was appointed today as well.
- The code above shows a security report encrypted with the user's public key and a set of attributes. Only (system administrators AND (hired before a specific date or on the security team)) OR (business staff AND 2 OF THE FOLLOWING (with an executive level 5 or higher, in the audit group, or on the strategy team) can decrypt the message. Looking back at Kevin and Sara, only one of the two has the necessary attributes. Kevin can decrypt this message with his private key; Sara cannot.
- For Kevin to decode, he must use his private key, the encrypter's public key, and the cpabe-dec function:

- cpabe-dec: This program decrypts an encrypted message using the encrypting user's public key, and the decrypting user's private key. The decrypted file will be sharing the name with the encoded data minus the cpabe.

Encryption is a technique for encoding information that shields its secrecy of its substance from unapproved aggressors. Encryption has been an apparatus to empower secure correspondence between a sender (encryptor) and a focused-on beneficiary of data. For instance, one might wish to store a message to such an extent that the client bob@yahoo.com must unencrypt it. While "point to point" encryption has many utilizations, this perspective of encoding is excessively inflexible, making it impossible to meet the majority of the data sharing requests of the present cloud situations (Jia & Yang, 2014).

Consider, for instance, if they had a database of encoded pictures that named with the date/time, area, source and watchwords identified with the image. Additionally, assume an authority later chose to give an examiner the capacity to inspect the ability to look at all pictures in the district of "Santa Clause Monica Pier" between 9 a.m. and 3 p.m. on December 1, 2015. On the off chance that the picture database was encrypted under a conventional open key encryption conspire, the authority would have two options. One is not to give the expert the private key and in this way not increase required access to the data. The second is to give the investigator the key and consequently give him the capacity to unscramble all pictures in the database—including those outsides of the extension.

Unmistakably, the two decisions result in an unfortunate result, and both are equal results of the "win or bust" energy of conventional encryption/decoding frameworks. Attribute-Based

Encryption (ABE) is another vision of encryption that moves past such customary limitations by considering flexible arrangement based access control that is cryptographically (or scientifically) implemented. How about they come back to their above illustration, yet this time accept that pictures in the database encoded in an ABE framework. In this situation when each image was encoded the information related with an arrangement of Attributes (chose by the encryptor); these could incorporate properties, for example, the time the picture took, GPS area alongside other picked meta information.

Later on, if such an examiner went to authority, the expert could make a private key for the investigator that was confined just to have the capacity to decode ciphertexts. The Attributes coordinated the arrangement of "Location: Santa Monica Pier" AND "Time: at 9 a.m.-3 p.m., December 1, 2015. This private key could decode any ciphertext whose properties coordinated this approach, however, would be useless in deciphering any that did not. Critically, the security of the framework based on numerically tricky issues, and the guard holds regardless of the possibility that an assailant figures out how to regenerate the capacity and get any ciphertext of his picking. While their spurring case concentrated on an encrypted database there are numerous cases of information that they wish to partake flexibly, for example, email, arrange bundles, sensor information. Besides, the settings where such information sharing is wanted can fluctuate from military/insight applications to friendly communities, business deals information.

The sorts of administrations are conveyed registering, parallel figuring and lattice computational development. In this distributed computing, all these users' information will be put away in these cloud assets Hubs. The outcome in the figuring disseminated to the client

through the system at whatever point the client needs. Even though distributed computing has turned into a powerful benefit show, and has an extensive request, distributed computing is still facing issues. Three noteworthy difficulties are:

1. Safety
2. Stability
3. Performance issue

It is difficult to specifically utilize the multi-specialist CP-ABE plans to the multi-specialist distributed storage frameworks because of property denial issue. The hugeness of multi-specialist in distributed storage frameworks is that users' qualities can change progressively. The client will be given new attributes or renounce current characteristics. Furthermore, authorization of information access ought to likewise appropriate change. All the current renouncement techniques, for the most part, depend on a confided in the server or might be the absence of proficiency; in this manner makes it not reasonable for managing the attributes disavowal issues in information get to control in the multi-specialist in distributed storage. This revocable multiauthority CP-ABE scheme is in which an efficient and furthermore secure denial technique is actualized to expel the attributes repudiation issue in the framework. The attributes repudiation strategy is sufficient that it takes just less correspondence cost and furthermore less calculation cost, and is secure as it demonstrates that it can accomplish both in reverse security and forward security. This plan does not expect the server to be trusted entirely because the vital refreshing is finished by each attribute expert and not by the server. If on the off chance that the server is not semi-confided in specific situations, this plan still shows ensure concerning the retrogressive security. This revocable

multi-expert CP-ABE scheme is applying as the hidden system to develop an efficient and secure attributes renouncement of information in multi-specialist cloud capacity (Hur & Noh, 2011).

Diverse authorities will provide distinctive attributes to the end users. Thus, here in the multi-authority framework, the data will likewise be of the various sort, yet all users will not have all the attributes. Henceforth the security issue emerges. In this paper, proposing a new algorithm called Improved Security Data Access Control. This algorithm is intended to enhance the security issue exists in the current framework. The data owner at the point when stores the data into the cloud server the first happens to encrypt the data then it will be stored on the cloud server. Then the key will be created by the authorities to various users. Also, given to the data owners. So, when the end user gets to any data, he ought to have desirable attributes as well as provide the keys to access the data (Jia & Yang, 2014).

The new algorithm likewise keeps up the integrity of the data stored. On the off chance that an attacker has changed the data the data owner will come to know about it when he verifies it. Also, when any of the users try to access the information which he cannot access then this sort of attack will likewise be notified by the authority and will be informed to the data owner. This framework does not require the server to be trusted entirely. And, regardless of the possibility that the server is semi-trusted then additionally this structure gives security (Basri & Rashmi, 2015).

Problem Statement

Chase's multi-authority CP-ABE protocol allows the data owner to decrypt all the ciphertexts since it holds the master key of the system. Chase's contract does not support

attribute revocation. The issue with the Chase multiauthority attribute-based encryption framework is that the CA can decrypt each ciphertext which decreases the user privacy and confidentiality of client data. Chase and Chow proposed a multi-authority-attribute-based encryption scheme without the central authority. Chase M and Chow S.S.M. proposed a multi-authority attribute based plan with user privacy (Rajkumar, George, & Batley, 2014).

Highlights

1. No trusted in central authority.
2. User privacy.
3. Distributed pseudorandom capacities utilize in the framework.
4. Collusion resistance for any number of colluding users.

Example. In a Medical Organization, if the Doctor is the having the master key he will be able to decrypt all the confidential files of all the patients, including the ones whom he is not authorized to view too. So, according to chase schema, the owner will hold the master key and will decrypt all the files. But with this new proposed scheme, this drawback is overcome by following an access structure which will decide the policy is defining who should be authorized to access that data or file. This way the user who is not possessing the secret key will not be able to view that file.

Existing System

This new model of data hosting and data access services acquaints a significant challenge with data access control. Since the cloud server cannot be trusted entirely by data owners, they can no longer depend on servers to access power. Ciphertext-Policy Attribute-based Encryption (CP-ABE) is viewed as a standout amongst the most appropriate

advancements for data access control in cloud storage frameworks since it gives the data owners more straightforward control over access policies. In CP-ABE scheme, there is an authority that oversees appropriate management and critical distribution (Yu, Wang, Ren, & Lou, 2010).

In a multi-authority cloud storage framework, attributes of users changed vigorously. A user might join some new qualities or revoked some present characteristics. In 2010, Yu, Wang, Ren, and Lou, took a shot at "Attribute-Based Data Sharing with Attribute Revocation". This paper utilizes semi trustable online proxy servers. This server empowers the authority to revoke user attributes with insignificant effort. This scheme was exceptionally incorporating the procedure of proxy re-encryption with CP-ABE and furthermore enables the authority to designate the clear majority of laborious tasks to proxy servers. The benefits of this scheme are More Secure against picked ciphertext attacks and give significance to attribute revocation which is troublesome for CP-ABE schemes.

Existing Methodologies

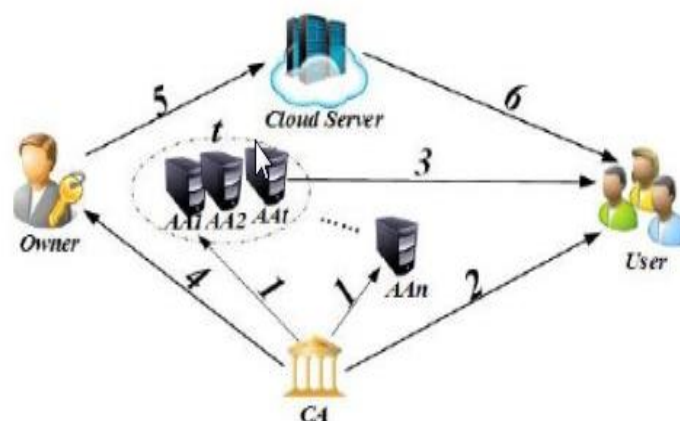


Figure 1. Framework and basic protocol flow. (National Conference on Recent Trends in Computer Science and Information Technology (NCRTCISIT), 2016)

The project structure of Threshold Multi-Access Control System (TMACS) shown in Figure 1. In TMACS, AA s should first register to CA to pick up the corresponding identity and authentication (help, aid, cert). At that point AA s will be engaged with the development of the framework, helping CA to complete the foundation of framework parameters. CA acknowledges users' registration and issues the declaration (uid, uid.cert) to each legitimate user. With the authentication, the user can contract with any t AA s one-by-one to pick up his/her secret key (SK). Owners who need share their data in the cloud can gain the public key (PK) from CA. At that point, the owner can encrypt his/her data under predefined access policy and transfer the ciphertext (CT) to the cloud server. Users can uninhibitedly download the ciphertexts (CT) that he/she occupied with from the cloud server. Nonetheless, he/she can't decrypt the ciphertext (CT) unless his/her attributes (National Conference on Recent Trends in Computer Science and Information Technology (NCRTCISIT), 2016).

- (1) AA registers to CA to gain (aid; aid:cert);
- (2) User registers to CA to gain (uid; uid:cert);
- (3) User gains his/her SK from any t out of n AAs;
- (4) Owners gain PK from CA;
- (5) Owners upload (CT) to the cloud server;
- (6) Users download (CT) from the cloud server [1].

Disadvantage

- The capacity overhead could be high if intermediary servers keep all the intermediary re-key. In 2011, S J. Hur and D.K. Noh worked on "Quality-Based Access Control with Proficient Revocation in Data Outsourcing Frameworks."

This paper proposes an entrance control system given figure content arrangement

attribute-based encryption to uphold get to control arrangements with productive property and client disavowal technique. The fine-grained get to can be accomplished by double encryption conspire. This double encryption instrument exploits the characteristic based encryption and specific gathering key appropriation in each attribute gathering. The benefit of this plan is safely dealing with the outsourced information. This scheme accomplishes effective and secure in the information outsourcing frameworks.

- The great issue in Enforcement of approval strategies and the help of strategy refreshes in 2011, Jahid, Mittal, and Borisov chipped away at "Less demanding: Encryption-Based Access Control in Social Networks with Efficient Disavowal." The proposed Easier engineering that underpins two methodologies are fine-grained get to control strategies and active gathering participation. Both plan accomplished by utilizing attributes based encryption, in any case, is that it is conceivable to expel access from a client without issuing new keys to different users or re-encrypting existing figure writings. They accomplish this by making an intermediary that takes an interest in the decoding procedure and upholds repudiation imperatives. The benefit of this conspire is the Easier engineering and development gives execution assessment, and model utilization of this approach on Facebook.
- Does not Achieve Stronger Security Guarantees.

Nature and Significance of the Problem

Proposed system. In this paper, they first offer a revocable multi-authority CP-ABE scheme, where an efficient and secure revocation strategy is advised to take care of the attribute revocation issue in the framework. This attribute revocation technique is proficient as it brings about less correspondence cost and calculation cost. Additionally, is secure as in it can accomplish both in backward security (The renounced user can't decrypt any new cipher text that requires the revoked attribute to decode) and forward protection (The recently joined user can likewise decrypt the beforehand distributed ciphertexts¹, on the off chance that it has adequate properties). Their scheme does not require the server to be trusted entirely because the vital upgrade authorized by each quality specialist, not the server. Regardless of the possibility that the server is not semi-trusted in a few situations, their scheme can even now ensure the backward security. At that point, they apply their proposed revocable multi-authority CP-ABE system as the fundamental methods to build the meaningful and secure data get access control scheme for multi-authority cloud storage frameworks (Basri & Rashmi, 2015).

The proposed system conquers the issue exists in the existing system. The author suggested another algorithm named as Improved Security data Access Control. This algorithm enhances the security of the framework. The data owner when stores the data into the cloud server he encrypts it and afterward stores it. The regarded authorities give keys to the authorized authorities. So, when the user tries to access the data to which he is not having the desirable attribute the demand gets rejected, and the user gets blocked by the authority. Also, the administration will likewise produce a message about the attack to the data owner so that

data owner can make additionally move. On the off chance that the user has done it by error the authorized user can contact the data owner to unblock him. On the off chance that the user has not done it then likewise the user can communicate the data owner and can guarantee greater security by requesting the data owner to change the login credentials. This new algorithm similarly gives data integrity (Basri & Rashmi, 2015). It educates about the attack by the unauthorized user to data owner when data owner verifies it. That is the point at which the data owner needs to check the files stored in the cloud often. Assuming any adjustments are found in the file on the server by any unauthorized access then this algorithm notifies the data owner that the document not protected, it changed.

There are five entities in the system as, a certificate authority (CA), attribute authorities (AAs), data owners (owners), the cloud (server) and data consumers (users). A global trusted certificate authority in the framework is CA. CA sets up the system and furthermore acknowledges the registration of the considerable number of users and also AAs in the structure. For each legitimate user in the structure, the CA assigns a unique user identity to it and furthermore creates a single public key for that user. Nonetheless, the CA do not engage with attribute administration and formation of secret keys that are related to attributes. For instance, the CA might be the Social Security Administration, a free office of the United States government. Each user is issued unique Social Security Number (SSN) as its standard attribute. Each AA is an independent attribute authority that oversees entitling and renouncing client's characteristics agreeing to their part or identity in its area. In this proposed scheme, each property associated with a single AA. However, every AA can deal with a personal number of attributes. What is more, every AA has added up to control over the structure and

semantics of its characteristics. Each AA is in charge of producing a public attribute key for each quality it oversees and, a secret key for every client mirroring their attributes.

To start with, let us consider what trait based encryption is and why this might be valuable. In standard crucial open cryptography, a document encrypted under a client's public key. The relating mystery key (and that key alone) would then be able to be utilized to decrypt the ciphertext. Presently, accept users each have different credits related to them. For instance, Alice might be in a gathering called "internal affairs," she is female and situated in the USA office of her association. Along these lines, they relegate her the traits "interior issues," "female" and "USA." On the off chance that Bob needs to encode a file, so it decrypted by everybody who is an individual from the "inner issues" gathering, he could make an encryption of the archive for each client in this group utilizing their open key. Notwithstanding, consider the possibility that Bob does not know who is in the gathering. Imagine a scenario in which users are added to this collection later. In this circumstance, they cannot utilize standard public key cryptography, in this manner, they swing to quality based encryption (ABE).

In ABE, a key expert is thought to be a trusted gathering who produces keys for users inside a framework. The critical specialist has an ace mystery key (MSK) and open key (PK). For every client in the structure, the essential specialist creates keys in light of the user's traits, utilizing the MSK. Every client then is given their relating mystery key, SK. Presently, when a client needs to scramble a report, they build a strategy for this file. The policy determines which ascribes are required to decrypt this story, for instance ("internal affairs" OR ("female" AND "Canada")). Given the built approach and the PK (of the key expert for a

framework), reports would then be able to be encoded and dispersed to everybody—except just decrypted by users who coordinate the arrangement allocated to the ciphertext (Ishii, Tempo, & Bai, 2013).

Note that if they have the approach ("internal affairs" AND "female" AND "Canada"), neither one of the bobs (given the qualities "male" and "Canada") nor Alice ought to have the capacity to decrypt archives with this strategy. They can see that together they meet the criteria - so significantly, they don't need conspiracy between users to enable them to decrypt files—just a client who meets the requirements alone ought to have the capacity to decode the archive.

Waters presents another ciphertext-strategy based (instead of key-approach based) ABE. When one develops another plan, the security must consider—formally one creates a proof diminishing to some hard-cryptographic issue. One thought while developing new projects is the thing that difficult problem they wish the plan to be lessened too. Be that as it may, at last security may rely upon whether the complicated issue to which will diminish is genuinely hard. In this way, Waters gives a few distinct developments, each lessening to the other point. One plan has a ciphertext size of $O(n)$, crucial private size of $O(A)$ and an encryption time of $O(n)$, where n is the span of an entrance equation. And the quantities of properties in a client's critical. Alternate plans have more regrettable complexities, yet diminish to various (harder) presumptions. This paper exhibits well the exchange off between suspicions required and the effectiveness of the subsequent plan (Li, 2015).

Types of attribute-based encryption. To comprehend the abilities of Attribute-Based Encryption, it sorts out consistently into three variations.

Content-based access control. In an ABE framework for content-based access control attributes will be related to a ciphertext while encoding delicate information. On the other side, a private key will connect to a strategy over these characteristics; ordinarily, the arrangement will communicate as a boolean equation. (In scholarly writing this variation is in some cases alluded to as "key policy" ABE.) For instance, in a framework that encodes messages they may remove the to: and from addresses alongside the time sent and subject as attributes, while scrambling the body of the email as mystery information. An authority produces a private key, that is utilized to express what sorts of ciphertexts the key can unscramble. For instance, a private key may take into consideration unscrambling of all messages that meet the strategy of to: engineering@corporation.com Or, on the other hand (subject: cascade-undertaking AND Date > Jan 1, 2015. A private key can decode a ciphertext if and just if its arrangement (boolean equation) fulfilled by the attributes of the ciphertexts. In an ABE framework, any string can fill in as quality. Moreover, properties can be numeric esteems and approaches can contain runs over these qualities. The arrangement of attributes utilized will rely upon the assigned application.

Role-based access control. An ABE framework for the part based access control "flips" the semantics of substance based access control. In such a framework, attributes will relate to a private key and an approach (or boolean recipe) related to the ciphertext. In such frameworks, the properties will frequently relate to the accreditations of a private key holder. (In scholarly writing this variation is some of the time alluded to as "Ciphertext-Policy")

ABE.) For occurrence, in an ABE framework for a partnership, a client may have a private key related with the properties Legal Department, Start: February 2013, SECRET Clearance or a programming engineer could have attributes for each venture she has chipped away. While scrambling a ciphertext, one will relate a strategy to the ciphertext. For instance, one could limit a ciphertext just to representatives who have been with the organization since 2012 and took a shot at the "Sound" programming venture. As in all ABE frameworks, get to control is numerically implemented is as yet secure regardless of the possibility that the aggressor approaches the information in scrambled shape.

Multi-authority role-based access control. One issue with part based access control is that in numerous applications they might want to compose access control strategies that traverse crosswise over various managerial limits. One trouble with standard ABE is that it requires one expert to give out private keys. Notwithstanding, in numerous applications, it is normal for various authorities to oversee several traits. For example, an organization like Experian could appropriate attributes about a client's FICO rating, while an HSBC may vouch for the insurability of a person. A multi-authority ABE framework enables one to relate a ciphertext with an arrangement composed crosswise over traits issued by various authorities (Yu et al., 2010).

Example of CP-ABE scheme in a personal health file system. Personal health file (PHR) framework is a novel application that can acquire incredible comfort medicinal services. The privacy and security of PHR are the real concerns of the users, which could hinder further development and broad appropriation of the framework. PHR is an average utilization of distributed storage, taking advantages of elastic computing resources assets to

give adaptable, unavoidable, and on-request health cloud services. Patients store their PHRs in cloud storage servers and like this can impart this information to friends or doctors advantageously. Be that as it may, such encouraging cloud-based application addresses new security difficulties: () Since PHRs should share among doctors, scientists, patients, et cetera, the sharing situation is confused. Patients ought to have the capacity to control the entrance in a fine-grained way. () PHRs might move among various cloud storage servers which can't entirely trust. In this way, patients can't depend on servers to ensure their PHRs. Outsourced information typically encrypted with figure key and the capacity servers oversee circulating figure keys to legitimate accessors. Be that as it may, such component is a recently secure area, yet not reasonable for PHR framework which works over a few spaces.

It is noteworthy to discover a fine-grained get to control strategy for PHR framework. As of late, Attribute-based encryption (ABE) appeared to be a promising system for such one-document multiaccess cloud storage situation. In ABE calculation, the patient can control the security by explicitly indicating access approaches for their outsourced PHRs, while the outsider substances, named doctors, oversee quality administration and critical dispersion. Cloud storage just needs to store the encoded PHRs. Along these lines, PHR benefit situated to patients over a few spaces.

Frequently, ABE plans work in two models, key-strategy ABE (KP-ABE) and ciphertext-approach ABE (CP-ABE). KP-ABE applies the strategy to property keys of accessors. Hence, once a key is predefined and is utilized to encode PHRs, accessors which can decrypt them are restricted. The accessor can just decode the PHRs related with an arrangement of Attributes that fulfills the key. In other words, PHR owner should know all

properties that accessors possess before he scrambles one PHR, with the goal that he can relate a right arrangement of characteristics. It isn't characteristic and viable unless the properties of accessors are created and appropriated by PHR owner himself. CP-ABE plot works inversely, which is theoretically nearer to the conventional access control techniques, for example, Role-Based Access Control (RBAC). The entrance strategy set by PHR owner amid PHR encryption, where the arrangement is a Boolean recipe comprising of open Attributes and legitimate operations, as "AND" and "OR." PHR owner does not have to know who can get to his PHRs because it is the duty of expert. Just the accessors with properties that fulfill get to the arrangement can decrypt ciphertext of PHR. Apparently, it is more sensible to execute CP-ABE scheme out in the open characteristics situation, and it is additionally advantageous for PHR owner without keeping on the web regularly.

Given the application situations of KP-ABE and CP-ABE, Li et al. (2017) proposed a PHR framework structure that joins KP-ABE and CP-ABE together. In the construction, users partitioned into individual areas (PSDs) and public spaces (PUDs) as indicated by their parts. For the most part, PHR owners (patients) regularly know users who get to the framework using PSDs. It is smarter to apply revocable KP-ABE conspire for PSDs, with the goal that patients oversee characterizing properties and approving accessors. Proficient users get to the framework through PUDs. They ought to have known parts, for example, specialist and analyst. Along these lines, it is better for the credits in PUD to be characterized and approved by outsider Attribute authorities. Li et al. utilize Chase-Chow multiauthority ABE scheme (CC MA-ABE) with a credit disavowal strategy to control the attributes in PUDs.

Although there are a few favorable circumstances for the division of client domains, a few deficiencies still exist for Li's ABE scheme (truncated as Li's MA-ABE), which are filled as takes after. Since it works given CC MA-ABE which is precisely a variation KP-ABE conspire, it constrained on a strict "AND" arrangement over a foreordained arrangement of doctors. As remarked by Lewko and Waters (2011), such approach isn't adaptable and expressive. With a specific end goal to get a similar storage of CP-ABE, it utilizes an extra conjunctive ordinary frame (CNF) administer for the age of both approach and encryption. () PUDs and PSDs need to apply different ABE plans and work in parallel. Be that as it may, this paper uncovers a verifiable intrigue, named part based arrangement, between users from PUDs and PSDs. Users in PSDs may likewise have proficient elements, for example, doctors with public qualities in PUDs. In this circumstance, one PHR owner can keep accessor from PSD by partner his PHR with an arrangement of PSD ascribes, however, may neglect to follow this accessor from getting to using PUD. For instance, understanding A has a companion B who fills in as a doctor in healing facility C. Quiet A goes to healing center C for the conclusion. He indicates an entrance arrangement for his encoded PHR to permit every one of the doctors in healing facility C get. Notwithstanding, he all of a sudden recall that his companion B additionally works there, and he does not need him to know the determination. Albeit persistent A does not approve companion B to decrypt using PSD, he can't prevent companion B from getting to through PUD.

There exist a few MA-CP-ABE plans, yet they are not intended for PUD's situation. Remarked by paper, CC MA-ABE restricted by the strict "AND" arrangement. Muller et al. (cited in Lewko & Waters, 2011) proposed an ABE scheme that can understand any entrance

structure, however, needs a verification focus. The utilization of validation focus may confront security and execution bottleneck since every one of the authorities ought to be controlled by the attention. Li, Yu, Zheng, Ren, and Lou (2013) gave a plan without confirmation focus yet needs to settle the arrangement of powers early. It can oppose scheme of users not precisely, where is a picked parameter at setup stage. Lewko and Water's (2011) ABE arrangement is adaptable yet needs quality revocation system. Ruj, Nayak, and Stojmenovic (2011) proposed an answer considering Lewko's ABE to make quality revocable. Nonetheless, it requires PHR owner to remain online for revocation, and its productivity is very low.

More vitally, the part based scheme which is enormous for PHR formwork not tackled in these past MA CP-ABE plans. Keeping in mind the end goal to oppose the conspiracy the proposed MA CP-ABE scheme outlines a boycott for the owner. Every client (PHR owner) can indicate a boycott of accessor characters that can't decrypt his information from PUD. This boycott assigned to an outsider specialist that the owner trusts. The specialist labels each boycott with a one of a kind public quality in PUD, so the owner can utilize this interesting federal credit to indicate his entrance strategy. In any case, the measure of general attributes will increment directly with users of PUD, which brings about an overwhelming weight for doctors.

Subsequently, this paper intends to develop the CP-ABE conspire for PUD situation which has effective denial and backings numerous doctors without a validation focus. Contrasted and Li et al.'s (2013) ABE scheme in PUD, their proposed conspire acknowledges

access control with adaptable access strategy. Additionally, the proposed part based arrangement is likewise efficiently fathomed. Their commitments finished up as takes after.

They propose an adjusted multiauthority CP-ABE scheme given Lewko's plan. With it, PHR owner can determine adaptable and expressive access strategy to secure their outsourced PHRs. Then, authorities require not speak with each other or be controlled by a confirmation focus. The quantity of properties is practically unlimited since the expansion of qualities does not involve more assets. They proposed a useful Attribute repudiation component for their plan. Property can be denied proficiently through the intermediary encryption and lethargic repudiation, while the project does not require a confirmation focus and any other interchanges among doctors. To oppose the part based scheme, they propose a boycott answer for forestalling it. By supplanting the private ace key and public key with hash estimation of Attribute's spellbinding name, the stockpiles in authorities keep little notwithstanding when some properties increment.

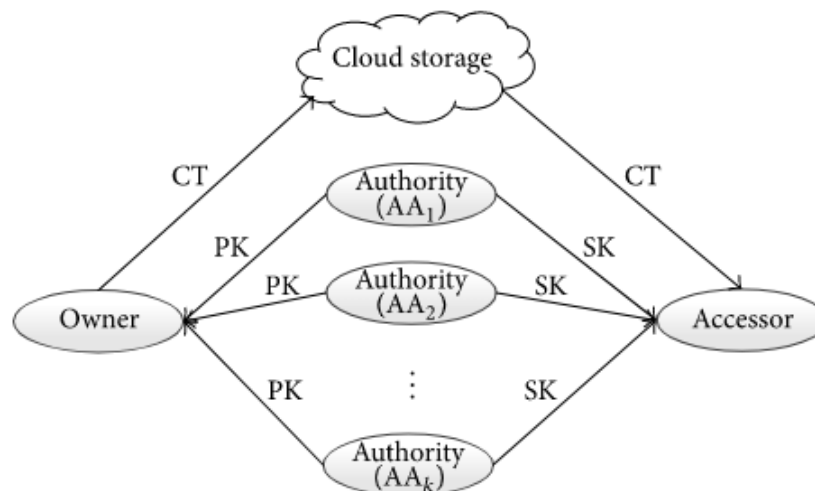


Figure 2. Another sample of framework of basic protocol flow.

The Multi-Authority Attribute-Based Encryption scheme for PUD gathers three kinds of participants, that is, cloud storage, authorities, and users (including data owner and accessors), as shown in Figure 2. The scheme includes five basic algorithms: System Setup, Authority Setup, Encrypt, KeyGen, and Decrypt. They described as follows.

System setup. The setup algorithm takes security parameter as input and outputs global parameters para.

Authority setup. Every attribute authority (AA) is running its authority setup process. The setup algorithm is taking global system parameters para and AA's descriptive attributes as input. Then, for each attribute that AA manages, AA generates a master key msk and the corresponding public key. The master keys are kept secret, while the general keys published (Wang et al., 2016).

Encrypt. Once the data owner gets public keys from authorities, he can execute encryption process in his terminal. The algorithm takes from several specialists, data for encryption, and an access policy specified by the data owner as inputs. Then, the algorithm encrypts to a ciphertext and generates a public attribute component (abbreviated as) for each leaf node. The whole data tuple of is the final ciphertext tuple and is uploaded to cloud storage.

KeyGen. Each authority manages its attributes set and is responsible for crucial distribution to legal users (accessors). Once an administration authenticates the identity of an accessor, it will process key generation which takes the master keys for a requested set of attributes as input and outputs user attribute components (abbreviated as) for each quality. All

the characteristics generated for the specific accessor are collected as the secret key of the accessor and sent back to the accessor secretly.

Decrypt. An accessor executes the decryption algorithm which takes the ciphertext tuple from cloud storage and the public keys and secret keys from authorities as inputs. If attributes set associated with satisfies access policy, the accessor can decrypt the plaintext data. Otherwise, it returns an error symbol.

PHR upload and access. Based on CP-ABE scheme they can quickly figure out the PHR upload and PHR access procedures. Peculiarly, once a data owner needs to upload his specific PHR file “pFile” to the cloud storage, he does the following steps: () Cut the data into contents segments s . () Pick random content key CK for each content segment. () Encrypt the section via symmetric cryptography and get the result. () Define an access policy over a set of attributes, encrypt content key CK as owner data via this proposed MA CP-ABE scheme, and get the ciphertext tuple. () Finally, upload and together as an integrated tuple to the cloud storage. The data owner can go offline, and authorities perform other key distribution workflows (Wang et al., 2016).

When an accessor needs to read the plaintext of one specific PHR on the cloud storage, he should process the following steps: () Get the whole ciphertext tuple and from the cloud storage. () Read the access policy from and know a minimal set of attributes required for decryption. () Get identity authenticated by several authorities, with which these authorities can return the keys associated with characteristics () to the accessor, respectively. () Collect enough keys to recover content key CK from () Decrypt to via symmetric cryptography by the content key and then construct the original PHR file “pFile.”

Efficient lazy revocation. There are two levels of revocation, that is, attribute revocation and accessor revocation. The attribute revocation is done by updating the attributes related to the PACs stored in the cloud storage so that the previously authenticated PACs are no longer used for decryption. The accessor revocation is done by the dismissal of all the attributes that an accessor owns.

The command of attribute revocation is started from authority when there are changes in the management of the accessors. Firstly, authority sends update parameter to the cloud storage and then the cloud storage updates via proxy encryption technique. In their revocation scheme, their corresponding will not update until someone requests them. Specifically, the cloud storage stores the updated parameters in an attribute history list (AHL) for each attribute revocation command. Once the ciphertext (related with a set of) requested, it can be updated only one time according to AHL, although the update parameters have been updated many times and filled in AHL. Such mechanism is called lazy revocation, which can accumulate update of parameters over time. Their revocation model has more efficiency than the DACC's solution when the delegates most computation workloads to the cloud storage and the revocation used.

For an accessor, once stored in the cloud storage is updated, their corresponding can no longer decrypt the ciphertext. Consequently, these accessors need to request authorities to update parameters. Instead of renewing the accessors', the authorities can just generate parameters, that is, updated keys (K), and let these accessors update there at their terminal. In previous papers, the revocation methods will produce the same update keys for all accessors. It is efficient but weak in security. Therefore, this proposed revocation scheme can support

two ways. The only one approach is to generate the same update parameters for all accessors, and the other one is to create different update parameters for different accessors. It is evident that the former method is efficient but has the potential risk in some circumstance. The latter approach is the opposite. PHR system can choose either process according to its strategy and environment (Li, 2015).

Collusion resistant. The same as most of the previous papers, their proposed MA CP-ABE scheme can resist both accessor collusion and authority collusion. Besides, the malicious but implicit role-based plot can also be opposed.

As discussed in Introduction, the role-based plot caused by the fact that PHR owner cannot predict the exact user identity who is an accessor from PUD because the attribute authentication controlled by the third authority party. To resist the conspiracy is essential for PHR owner to specify a blacklist which contains the access identities that are not allowed to access from PUD. And delegates the blacklist to a third authority party. The authority maps each blacklist to an attribute, such as attribute “Alice Blacklist1,” so that an owner can combine such characteristics in his access policy in the PUD is to restrict specific identity from access. The number of blacklisted attributes will grow linearly with the users in PHR system. Fortunately, this proposed ABE construction is efficient in managing characteristics because the algorithms replace attribute master keys with the hash values of attributes’ descriptive names. The storage for the attribute management will keep less at the authority even when the amount of the characteristics grows. It means that the blacklisted solution is highly efficient.

Accessor collusion denotes that different accessors will combine their attribute components (PACs) for decryption of a file despite the fact that they do not have enough attributes to decrypt it alone. This proposed MA CP-ABE scheme can resist the accessor collusion by embedding the accessor's hash value into their PACs. Accordingly, the temporary result in decryption phase differs among accessors. Therefore, the decryption process then resisted. Accessor collusion important security metric in the multiauthority plot. In this proposed scheme, since the authorities do not communicate with each other or have no predefined parameters among them, the authority collusion is impossible in their proposed system (Li, 2015).

Advantages of proposed system.

- They alter the structure of the scheme and make it more down to earth to cloud storage formworks, in which data owners not included in the key generation.
- They significantly enhance the effectiveness of the property revocation technique.
- Our system not just gives forward and backward security. However, it additionally gives increased security by providing access control to authorized users.
- The algorithm proposed by us improves the safety by notifying about the attack to the data owner (Srinath & Obulesh, 2016).
- They provide the data integrity. The data owner identifies the verification of the data storage when he checks the file.

Objective of the Study

- The fundamental goal of the project is to plan an Expressive, Efficient, and Revocable data access control scheme for multi-authority cloud storage formworks, where there are different authorities exist together, and every specialist can issue attribute freely.
- They adjust the structure of the scheme and make it more viable to cloud storage formworks, in which data owners not included in the key generation.
- They enormously enhance the proficiency of the attribute revocation technique.
- They additionally profoundly improve the expressiveness of their access control scheme, where they evacuate the restriction that each property can just show up at most once in a ciphertext (Srinath & Obulesh, 2016).

Table 1

Comparisons between Different Techniques

SR.no	Technique	Algorithm	Scalability	Efficiency	Security
1	ABE	DES	HIGH	LOW	LOW
2	CPABE	DES	LOW	HIGH	LOW
3	KPABE	DES	LOW	HIGH	LOW
4	IBE	AES	LOW	LOW	HIGH
5	MA-CPABE	AES	HIGH	HIGH	LOW
6	PROPOSED SYSTEM	RSA	HIGH	HIGH	HIGH

Definition of Terms

Table 2

Definition of Terms Used in This Document

ABE	Attribute - based encryption is a kind of public key encryption in which the secret key of a user and the cipher text are reliant upon attributes
CPABE	Cipher -text Attribute based Encryption
KP-ABE	Key Policy Attribute based Encryption
PK	a cryptographic key that can be acquired and utilized by anybody to encrypt messages expected for a specific recipient, with the end goal that the encrypted messages can be deciphered only by utilizing a second key that is known just to the recipient (the private key).
MK	A symmetric master key is utilized to determine other symmetric keys (e.g., data encryption keys, key wrapping keys, or authentication keys) utilizing symmetric cryptographic strategies.
PRE	Proxy re-encryption schemes are cryptosystems which permit third parties(proxyes) to modify a cipher text which has been encrypted for one party, with the goal that it might be decrypted by another.
CA	Certificate Authority
AA	Attribute Authority
CT	Cipher-Text
AES	Advanced Encryption Standard

Summary

In this chapter, studied the objectives of the proposed system and how it is overcoming the disadvantages exists in the existing system. The coming section will have details about the background and literature of the paper.

Chapter II: Background and Review of Literature

Introduction

Cipher Text-Policy Attribute Based encryption scheme represented a formwork for acknowledging complex access control on encrypted data. Utilizing the strategy will encrypt the data kept confidential regardless of the possibility that the storage server is untrusted. The proposed formwork takes into consideration another sort of encoded access control where user's private keys indicated by a set of attributes and a party encrypting data can determine a policy over their attributes meaning which users can decrypt it. It was demonstrated secure just under some general group heuristic, and, not in other circumstances (Waters, 2011).

Different computing needs accommodating for the users and organizations, who utilize cloud administrations. Reliability and accessibility ought to be kept up with the Cloud Service Provider as Data Centers; they are keeping up with any piece of the world. Aside from these, users who stressed over their data which contains sensitive data, for example, medical files or financial data and business-related data must be put away safely (Kumar & Lakshmi, 2015).

Security Risks in Single and Multi-authority cloud storage. While users outsource their confidential data to the cloud, the service provider checks the client data with the Third-Party Auditor without knowing the data; it checks the integrity and accuracy of data. In the single cloud, because of any byzantine disappointment or benefit inaccessibility, network issues with disaster or some different leads the client data in dangers. Indeed, even they had been ensuring utilizing Cryptosystems; Cloud Service Supplier cannot guarantee the hazard

associated with Single cloud or, on the other hand, Multi-authority cloud storage (Kumar & Lakshmi, 2015).

Sahai and Waters (2005) proposed the principal ABE scheme, in which ciphertext is encrypted and connected with an arrangement of attributes. An accessor can efficiently decrypt the ciphertext if and just if he gets an arrangement of characteristics parts where the set cover between the two qualities sets, that is, is past predefined limit. A short time later, Goyal, Pandey, Sahai, and Waters (2006) proposed KP-ABE scheme, in which an arrangement of conditions from an accessor built through a tree-like approach which taken as the key of the accessor. The leaf hubs of the tree related to terms and the nonleaf centers are coherent operations, for example, "or" "and." Data owner connects his ciphertext with an arrangement of attributes. Once the related properties fulfill a particular key-approach of the accessor, the accessor can decrypt the ciphertext. Be that as it may, the information owner should know all the keys of accessors before he encodes the information and afterward, he can appropriately relate the ciphertext with comparing properties. Such prerequisites of KP-ABE are not reasonable for community situation, where the information owner can't foresee which individual can get to his information (Li, 2015).

Subsequently, Bethencourt, Sahai, and Waters (2007) proposed CP-ABE which is reasonably nearer to the conventional access control techniques, for example, RBAC. CP-ABE plot connects to access the arrangement in ciphertext rather than attributes of accessors. It is more instinctive for the information owner to determine such approach during the time he encodes the information for accessors, they should possess enough qualities issued by the outsider, named specialists, to decrypt the ciphertext efficiently. Moreover, ordered binary

decision diagram (OBDD) is utilized to depict get to strategies in CP-ABE. The framework makes full utilization of both the total depiction capacity and the high computing proficiency of OBDD and enhances both execution and productivity. In any case, just a single expert may cause the bottleneck of implementation. Additionally, it is more regular and viable with numerous expert associations (authorities) to oversee arrangements of attributes. Security can be enhanced with the multiauthority because an aggressor should bargain a few authorities in the meantime to get the keys related to enough mechanisms of characteristics for decoding.

There are as of now a few endeavors to take care of multiauthority ABE issue with new cryptographic arrangements. Chase and Chow (2009) initially proposed a multiauthority ABE plot (CC MA-ABE) in which every client is approved considering a global identifier (GID, for example, a standardized savings number. The GID plays a linchpin to relate users' keys from various authorities together. Be that as it may, the arrangement still depends on a confirmation focus and the entrance strategy is not adaptable and expressive which restricted on "AND" door approach over the foreordained method of experts. Afterward, Li, Xue, Xue, and Hong (2014) proposed an ABE plot with trait renouncement system given CC MA-ABE, which restricts to a lead of CNF in the entrance arrangement. A limit multiauthority CP-ABE get to control conspire was proposed for open distributed storage in which both security and execution enhanced.

It is essential for MA CP-ABE to help a robust and adaptable access strategy. For instance, American Medical Association (AMA) approves attributes of therapeutic expert licenses. For example, junior attendant permits an experienced attendant permit, while American Hospital Association (AHA) recommends qualities of affiliations, for example,

doctor's facility A and healing facility B. If one patient feels that the analysis and treatment in healing facility A are superior to those in doctor's facility B. They may indicate an entrance approach that allows the medical caretakers with any level of the permit in clinic A to get to his PHR documents. Additionally, just permit the attendants with the junior level of the license from doctor's facility B get to. Such expressive strategy is exhibited as approach = $((\text{junior medical attendant level} \vee \text{experienced medical attendant level}) \wedge \text{doctor's facility A}) \vee (\text{junior medical attendant level} \wedge \text{clinic B})$. The arrangement can change to the "AND" strategy; for instance, approach =, where alludes to the specialist and alludes to the approach oversight by, and one expert has just a single statement.

There are some different plans which can set the entrance arrangement in any Boolean recipe over attributes from any number of authorities. Among them, Muller proposed another MA-ABE scheme which acknowledged on any entrance structure with a verification focus. Yang and Jia (2012) proposed a variation CP-ABE plan to help multiauthority, yet despite everything, it requires an extra validation focus to produce client mystery key and specialist mystery key. Additionally, it is powerless in renouncement security. Considering Yang's plan, a broad idea was proposed to withstand the weakness. For MA-ABE scheme with a verification focus to control different authorities, once the confirmation focus broke, the whole ABE framework bargained. Subsequently, it ought to be trusted entirely which is difficult to ensure. Also, the entire ABE framework challenging to be extended. Some considers endeavor to expel the confirmation focus from MA CP-ABE plans. Chase and Chow (2009) utilized pseudorandom capacities (PRFs) between various experts without the

middle. Notwithstanding, yet constrained on "AND" get to an arrangement over a decided mechanism of authorities. Li et al. proposed a limit based ABE conspire that is decentralized and authorizes a proficient characteristic denial plot. The framework is conspiracy safe for fewer users, where is picked statically amid the setup stage. Be that as it may, the experts set ought to be designed to the setup stage and settled in the running. The experts ought to associate with each other at the setup stage, and the entrance approach is unyielding. Afterward, Lewko and Waters (2011) proposed a plan for decentralized ABE situation, in which the authorities work autonomously without coordination among them. A significant downside is that the idea has no renouncement work. Although a further paper (DACC) tended to it, the calculations of key refresh and correspondence overhead for property renouncement are very overwhelming. Plus, DACC requires the information owner to partake in renouncement and transmit a refreshed ciphertext segment to each unrevoked client. It implies that the information owner should continue being on the web continuously, as is outlandish in practical application situation (Chase & Chow, 2009).

Attribute Revocation is a critical issue for an ABE framework and advantages security of the structure. Once an expert distinguishes a malignant client, every one of his attributes or one of his particular characteristics ought to be denied by the specialist, which implies the pernicious client can never again decrypt the ABE-produced ciphertext related with those qualities. In single specialist ABE conspire Yu et al. (2010) presented the idea of intermediary encryption into CP-ABE to acknowledge trait denial, in which means of encryption refreshes the influenced quality parts of ciphertext and the attributes segments put away in terminals of unrevoked users. Propelled by paper, Yang and Jia (2012) proposed the CP-ABE plot with a

more productive renouncement than that in, be that as it may, it requires a validation focus to control the different authorities.

Background Related to the Problem

- Token pre-computation
- Correctness verification and Error localization
- Error recovery

Token pre-computation. Before file allocation, the user pre-computes a certain number of unexpectedly substantiation updates on the characteristic vector. Users' needs to ensure the loading exactness for the data in the cloud, he experiences the cloud servers with a set of unintentionally created block indices. Each cloud server computes a short "signature" over the predetermined blocks and returns them to the user.

Correctness verification and error localization. Error localization is a crucial capability for excluding errors in loading formworks. Their formwork surpasses those by incorporating the accuracy substantiation and error localization (misbehaving server identification) in their challenge-response protocol.

Error recovery. The user can reconstruct the original document by changing the data vectors from the first m servers, accepting that they give back the suitable reaction values. That this substantiation formwork depends on random spot-checking, so the loading accuracy self-confidence is a probabilistic one. The data decline is detected, the association of pre-computed demonstrations and received reaction values can assure the identification of misbehaving server(s) (again with high probability).

Literature Related to the Problem

Waters introduced the attribute-based encryption (ABE) as the new means for the encrypted access control. In an attribute-based encryption system ciphertexts are not necessarily encrypted to one user as in traditional public key cryptography (Sahai & Waters, 2005). In fact, both users' private keys and ciphertexts will relate to a set of attributes or a policy over the attributes. A user can decrypt a ciphertext if there is a "match" between his private key and the ciphertext. Their central system depicted the Threshold ABE framework in which the ciphertexts marked with a set of attributes S and a user's private key which associated with both a threshold parameter k and another set of attributes S' . For the user to decrypt a ciphertext, there should be at least k attributes must overlap between the ciphertext and his private keys. One of the original primary motivations for this was to design an error-tolerant (or Fuzzy) identity-based encryption scheme that could use biometric identities.

The primary disadvantage of the Waters threshold ABE system is that their threshold semantics are not very expressive and therefore are restricting for designing more general systems. Goyal et al. introduced the idea of a more general key-policy attribute-based encryption system. In their establishment, a ciphertext related with a set of attributes and a user's key can associate with any monotonic tree access structure (Sahai & Waters, 2005). The construction of Goyal et al. can have viewed as an extension of the Waters techniques where instead of embedding a Shamir secret sharing scheme in the private key, the authority embeds a more general secret sharing scheme for monotonic access trees. Goyal et al. (2006) also suggested the possibility of a ciphertext-policy ABE scheme, but did not offer any constructions.

The storage overhead could be high if proxy servers keep all the proxy re-key. In 2011, Hur and Noh took a shot at "Attribute-Based Access Control with Efficient Revocation in Data Outsourcing Systems". This paper defines an access control instrument considering cipher text-policy attribute-based encryption to authorize access control strategies with proficient characteristic and user revocation strategy. Double encryption scheme can accomplish the fine-grained access control. This dual encryption system exploits the attribute-based encryption and group key distribution in each aggregate group. The benefit of this project is safely dealing with the outsourced data. This scheme accomplishes productive and secure in the data outsourcing formworks.

The most significant issue in Enforcement of authorization strategies and the support of policy updates in 2011, Jahid, Mittal, and Borisov, took a shot at "Easier: Encryption-Based Access Control in Social Networks with Efficient Revocation". The proposed Easier structure that supports two methodologies is fine-grained access control strategies and dynamic group membership. Both scheme accomplished by utilizing attribute-based encryption, in any case, is that it is conceivable to expel access from a user without issuing new keys to different users or re-encrypting existing ciphertexts. They accomplish this by making a proxy that participates in the decryption procedure and authorizes revocation constraints. The benefit of this scheme is the Easier design and development gives execution assessment, and prototype utilization of this approach on Facebook does not Achieve Stronger Security Guarantees (Jahid et al., 2011).

Summary

In this chapter, have studied about the Background and Literature review of the paper.

In the next section, will discuss the methodology of the article.

Chapter III: Methodology

Introduction

As the number of users in cloud computing is expanding, security issues are additionally expanding accordingly. The primary security issue can be on how to control the unauthorized data access in the cloud. In this paper, they proposed an efficient data access control scheme with enhanced security. Their plan restricts the unauthorized access as well as guarantees secure access by the approved users. Alongside that data, integrity likewise provided. This scheme proposed for multi-authority cloud storage formwork. This project can connect to social networks which are on the web and furthermore in the remote storage formworks. Java is the language used to implement the algorithms, and it is the most powerful language regarding security (Waters, 2011).

Design of the Study

A ciphertext-policy attribute-based encryption scheme comprises of four elemental algorithms: Setup, Encrypt, KeyGen, and Decrypt.

- **Setup.** The setup algorithm takes no input other than the implicit security parameter. It produces the output of public settings PK and a master key MK. The challenger runs the Setup algorithm and gives the public parameters, PK to the adversary (Sahai & Waters, 2005).
- **Encrypt (PK, M, A).** The encryption algorithm takes as input the public parameters PK, a message M, and access structure A over the universe of attributes. The algorithm will encrypt the message M and deliver a ciphertext CT with the aim

that only a user that has a set of attributes that fulfills the access structure will have the Storage to decrypt the message. They will assume that the ciphertext indeed contains A.

- **Key Generation (MK, S).** The key generation algorithm takes as input the master key MK and a set of attributes S that outline, the key. It yields a private key SK .
- **Decrypt (PK, CT, SK).** The decryption algorithm takes as input the public parameters PK , a ciphertext CT , which contains an access policy A , and a private key SK , which is a private key for a set S of attributes. On the off chance that the set S of characteristics fulfills the access structure A then the algorithm will decrypt the ciphertext and give back a message M (Sahai & Waters, 2005).

In CP-ABE the ciphertexts are related to access structures and the private keys with attributes. A party that wishes to encrypt a message will indicate through an access tree structure a policy that individual keys must fulfill to decrypt.

Attribute Revocation

As different authorities exist there will be various attributes to the user, and the attributes can change dynamically. That is the authority can give a user a few new characteristics or revoked some current qualities. This sort of attribute revocation ought to consider accordingly. The new scheme overcomes the issue of cancellation yet at the same time there exist security issues in the current formwork (Yu et al., 2010).

Attribute revocation has two requirements.

- The revoked user (whose attribute denied) can't decrypt new ciphertexts encrypted with new public attribute keys (Backward Security).
- The recently joined user who has adequate characteristics ought to likewise have the Storage to decrypt the already cloud ciphertexts, which encrypted with past public attribute keys (Forward Security). For instance, in a college, some file archives are encrypted under the policy "CS Dept. AND (Professor OR Ph.D. Student)", which implies that particular the teachers or Ph.D. students in CS department can decrypt these archives. Whenever a new teacher/Ph.D. student joins the CS department of the college, he/she ought to likewise have the Storage to decrypt these files. Their attribute denial strategies can accomplish both forward security and in backward safety (Yu et al., 2010).

Collusion resistance and attribute-based encryption. The defining property of Attribute-Based Encryption formworks is their resistance to collusion attacks. This property is fundamental for building cryptographic access formworks; else, it is unimaginable to ensure that a formwork will guarantee that a system will exhibit the desired security properties as there will exist devastating assaults from an attacker that figures out how to get it together a couple of private keys. While they should seriously think about ABE formworks with various kinds of expressibility, earlier work made it clear that collision resistance is a required property of any ABE formwork (Shamir, 1984).

Before property based encryption presented different frameworks attempted to address access control of encrypted data by utilizing secret sharing schemes consolidated with identity-based encryption; be that as it may, these plans did not deliver resistance to collusion attacks. As of late, Kapadia, Tsang, and Smith gave a cryptographic access control scheme that utilized proxy servers (Shamir, 1984).

Their work investigated new strategies for utilizing proxy servers to hide policies and use non-monotonic access control for small universes of attributes. They take note of that although they called this plan a type of CP-ABE, the idea does not have the property of collision resistance. Accordingly, they trust that their work ought not to consider in the class of attribute-based encryption formworks because of its absence of security against collusion assaults.

DAC-MACS Contain Five Algorithms

System Initialization, Secret Key Generation, Encryption, Decryption and Attribute Revocation to demonstrate the security, the authors propose an amusement between a challenger and an adversary, and decide that DAC-MACS are secure under the decisional q - parallel BDHE suspicion. Be that as it may, this amusement makes an indicative restriction that the adversary proved unable to get CUKxk of any denied attributes (Hong, Xue, & Li, 2015).

Multi-Authority Data Access Control For Cloud Storage System With Attribute-Based Encryption

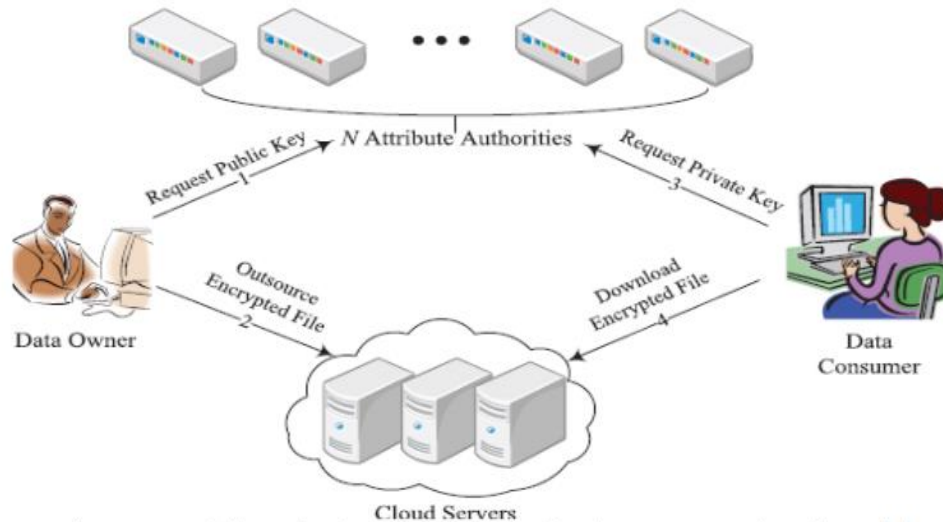


Figure 3. General flow of AnonyControl and AnonyControl scheme (Jung, Li, & Wan, 2015).

In this formwork, there are four sorts of substances. N Attribute Authorities, Cloud Server, Data Owners and, Data Consumers. A user can be a Data Owner and a Data consumer all the while.

Authorities are expected to have robust computation capacities, and they directed by government offices since a few attributes in part contain users' by and by attributes identifiable data. The entire attribute set is partitioned into N disjoint sets and controlled by every authority, in this manner every authority knows about just piece of attributes. A Data Owner is the element who wishes to outsource encoded data file to the Cloud Servers. The Cloud Server, who is expected to have a satisfactory storage limit, does only store them. Newly joined data Consumers ask for private keys from the majority of the authorities, and they don't know which attributes are controlled by which authorities. Whenever the Data Consumers ask for their private keys from the authorities, authorities together make

comparing private key and send it to them. All Data Consumers can download any of the encrypted data documents; however, just those whose private keys fulfill the privilege tree T_p can execute the operation related to benefit p . The server is designated to execute an operation p on the off chance that and just if the user's certifications confirmed through the privilege tree T_p (Jung et al., 2015).

Design Data Access Control Scheme

To plan the data, get to control conspire at first for the multi-specialist cloud storage formworks, the first Revocable Multi-specialist CP-ABE convention built. There are five stages: System Initialization, Key Generation, Data Encryption, Data Decryption and Attributes Revocation. In Chase has proposed a multi-specialist CP-ABE convention. Still, it can't explicitly connect to the primary strategies considering these two reasons:

- Security Issue
- Revocation Issue

The conference does not bolster property repudiation. This, new revocable multi-specialist CP-ABE convention which depends on single-specialist CP-ABE proposed by Lewko and Waters (2011) in is extended it to a multi-expert situation and make it revocable.

Additionally, the strategies in Chase and Chow's (2009) multi-expert CP-ABE convention connected with mystery keys produced by distinctive doctors for a similar client to counteract arrangement assault. The usefulness of the specialist partitioned into global declaration specialist (CA) and different attributes doctors (AAs). CA sets up the formwork and furthermore acknowledges enlistment of users and AAs in formwork. It allows a worldwide

client attributes to every client and an extension specialist character to each expert in the formwork (Chase & Chow, 2009).

The ID is all inclusive one of a kind in formwork, mystery keys issued by various AAs for the same ID together for encrypting. Likewise, each quality is recognized yet also a few AAs may issue same property. This plan requires authorities to create claim public keys and after that utilizations them to scramble data with the typical open parameters. It keeps the authentication expert in the idea from unencrypting the figure writings. At the point when a property revocation happens, figure writings are refreshed just for those parts related to disavowed attributes in mystery keys. At the end when a characteristic of a client renounced their produces another rendition key for the repudiated trait and furthermore provides a refresh key. The refreshed key, users, aside from disavowed client, who has renounced qualities can restore its mystery key. Utilizing updated key, segments related to the repudiated quality of the figure content likewise refreshed to most recent rendition. To move forward the proficiency re-encryption technique, that the newly joined client can decrypt the past cloud data, which was encrypted with previous public keys if they got enough qualities (Forward Security). When refreshing these figure writings, the users need just current mystery key, no need to keep files of past mystery keys (Chase & Chow, 2009).

System Architecture

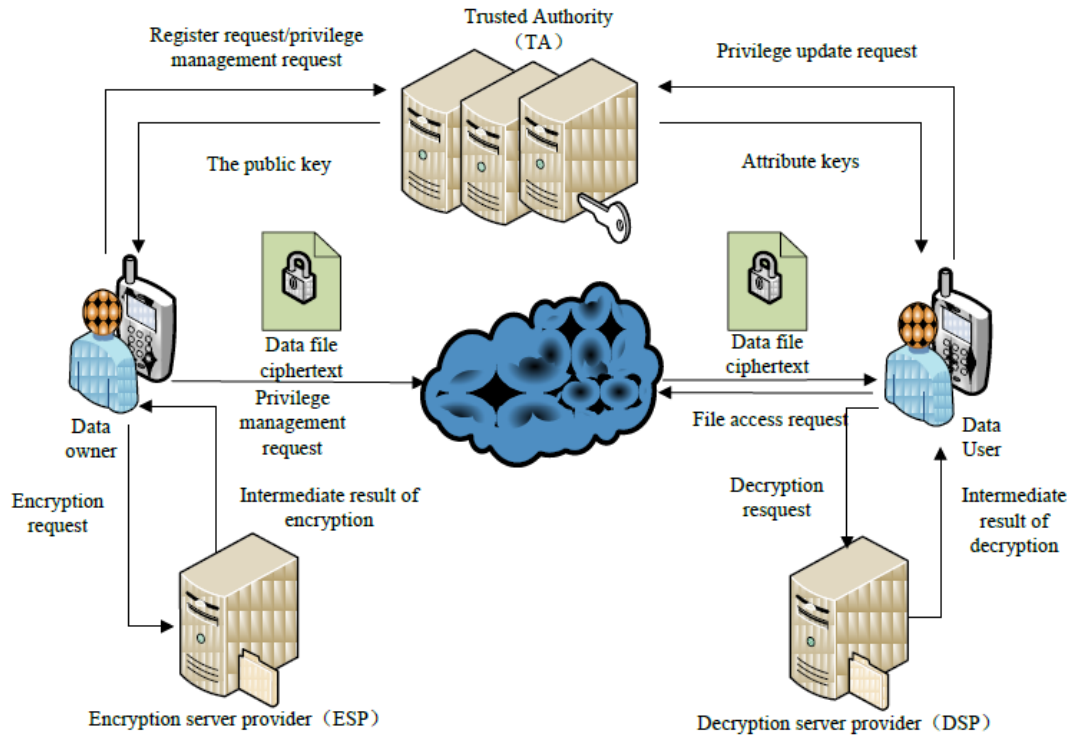


Figure 4. The figure demonstrates the system architecture and defines the structure, behavior, and more views of the system (Li, Shen, He, Xu, & Su, 217).

System Model (Process Flow)

The below figure demonstrates the system model, and it comprises of the modules: Data owner, Cloud Server, Data Encryption and Decryption, Authority, Data Consumer and Improved Security (Yang & Jia, 2012).

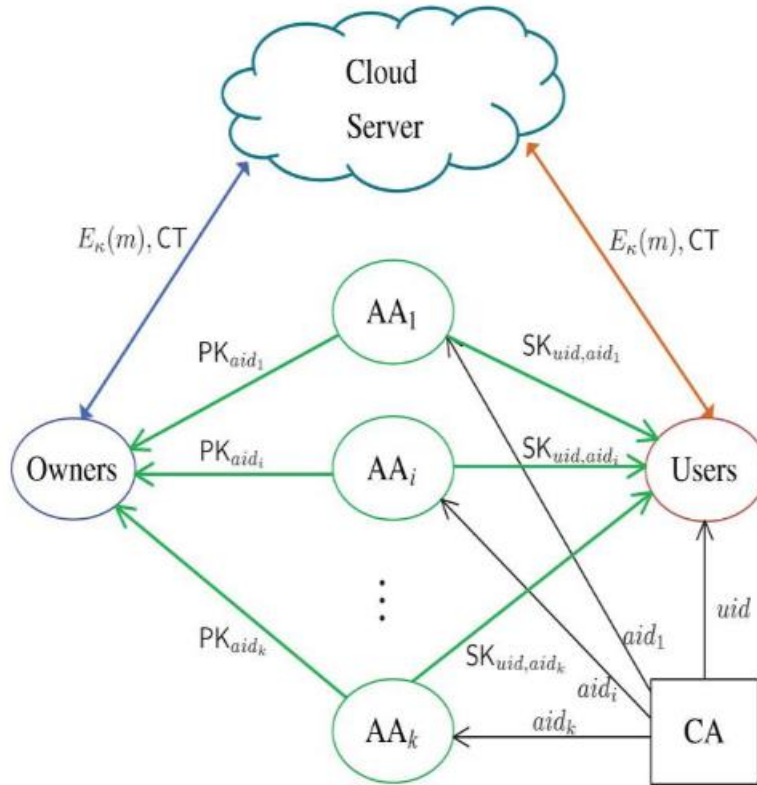


Figure 5. System model of data access control in multi-authority cloud storage (International Journal of Computer Science Trends and Technology (IJCTST), 2016).

The data access control scheme which they consider in multi-authority cloud storage portrayed in the fig. Five sorts of elements are there in the formwork: a certificate authority (CA), attribute authority (AA), data owner, data consumer, the cloud server. The trusted certificate authority in the formwork is the CA. The formwork is set up, and the enlistment of all client and AAs are acknowledged. The CA assigns out the particular worldwide id and furthermore creates a global public key for each authorized user. AA Oversees revoking client's attributes as per their part or character. Each attribute is related to single AA; however, some qualities are overseen by AA. The attributes "structure and semantics

controlled by each AA. Every AA creates people in general attribute key for each property it manages and a secret key for every client. This structure depicts that the owner outsources the information with the semi-trusted cloud servers with encoded cryptosystems. At that instance when the user needs to get to the data from cloud servers, users must be maintained by the Certificate Authority who gives the confirmation authentication to the client to access data. After acquiring the authentication user and owner share the data with the attributes confirmation for data access. In this framework, every user has a global identity. The user can have set of qualities which originate from different characteristic authorities. The relating characteristic doctors entitle its client related to a mystery key. The data isolated into a few components by the owner, and each data segment encrypted with various content keys utilizing symmetric encryption.

Attribute revocation algorithm. A Multi-authority Ciphertext-Policy Attribute-Based Encryption system with identity-based user revocation comprised of the following algorithms: $\text{Global Setup}(\lambda) \rightarrow \text{GP}$ The global setup algorithm takes in the security parameter λ and outputs global parameters GP for the system.

Central authority. $\text{setup}(\text{GP}) \rightarrow (\text{SK}^*, \text{PK}^*)$ The central authority (CA) runs this algorithm with GP as input to produce its secret key and public key pair, SK^*, PK^* (Chase & Chow, 2009).

Identity KeyGen $(\text{GP}, \text{RL}, \text{GID}, \text{SK}^*) \rightarrow \text{K}^*\text{GID}$ The central authority (CA) runs this algorithm upon a user request for secret identity key. It checks whether the offer is valid and if yes (i.e. $\text{GID} \in \text{RL}$), generates K^*GID using the global parameters and the secret key of the CA.

Authority Setup. $(GP) \rightarrow (PK, SK)$ Each attribute authority runs the authority setup algorithm with GP as input to produce its secret key and public key pair, SK, PK.

KeyGen. $(GP, SK, \text{GID}, i) \rightarrow K_i, \text{GID}$ The attribute key generation algorithm takes in an identity GID, the global parameters, an attribute (i) belonging to some authority, and the secret SK for this authority. It produces a key k_i , GID for this attribute, identity pair.

Encrypt. $(GP, CT, (A, \rho), \{PK\}, PK^*, RL) \rightarrow CT$. The encryption algorithm takes in a message M, An access matrix (A, ρ) , the set of public keys for relevant authorities, the public key of the central authority, the revoked user list and the global parameters. It outputs a ciphertext CT (Chase & Chow, 2009).

Decrypt $(GP, CT, (A, \rho), \{K_i, \text{GID}\}, K^*, \text{GID}, RL) \rightarrow M$. The decryption algorithm takes in the global parameters, the revoked user list, the ciphertext, identity key and a collection of keys corresponding to attribute, identity pairs all with the same fixed identity GID. It outputs either the message M. When The group of attributes (i) satisfies the access, matrix corresponding to the ciphertext. Otherwise, decryption fails (International Journal of Computer Science Trends and Technology (IJCSST), 2016).

Certificate authority. The CA is a worldwide trusted certificate authority in the formwork. It sets up the formwork and acknowledges the enlistment of the considerable number of users and AAs in the formwork. For each legitimate user in the formwork, the CA allocates a globally unique user identity to it and furthermore creates a global public key for this user. Although, the CA not included in any attribute management and the formation of secret keys that are related to attributes. For instance, the CA can be the Social Security

Administration, an autonomous organization of the United States government. Every user will issue a Social Security Number (SSN) as its global identity.

Data encryption and decryption. All the authorized users in the system can freely query any concerned encrypted and decrypted data. After accepting the data from the server, the user runs the decryption algorithm Decrypt to decrypt the ciphertext by utilizing its secret keys from various Attribute Authorities (AAs). Just the attributes the user has to fulfill the access structure defined in the ciphertext CT; the user can get the content key. Here AES Algorithm is being utilized to encrypt and decrypt each data (Jia & Yang, 2014).

Attribute authorities. Each AA is an independent attribute authority that is in charge of entitling and revoking user's attributes as indicated by their part or character in its domain. In their scheme, each attribute is related to a single AA. However, every AA can deal with a discretionary number of attributes. Each AA has full control over the structure and semantics of its attributes. Every AA was overseas creating a public attribute key for each attribute it monitors and a secret key for every user reflecting his/her attributes (Chase & Chow, 2009).

Authority. Authorities from the various area give the attributes to the end users. One end user can have the attributes offered by multiple authority, and indeed, even the authority can provide the characteristics too numerous end users. Just the end user who has the authorized features can access the specific files.

Data owners. This module, the data owner uploads their data to the cloud server. For the security reason, the data owner encrypts the data file and at that point stores it in the cloud. The data owner can change the policy over data files by updating the expiration time. The Data owner is prepared to manipulate the encrypted data file (Chase & Chow, 2009). The

data owner is additionally in charge of blocking and unblocking the malicious user when he gets the message of the attack by the authorized user. Data owner furthermore checks for the integrity by verifying the uploaded documents time to time

Data consumers. Here the user can just access the data file with the encrypted key if the user has the privilege to access the file that is if the user has enough attribute to access that file. For the user level, each one of the rights is given by the Domain authority as attributes, and the data users are controlled by the Domain Authority only. Users may attempt to access data files either inside or outside the extent of their access privileges. So malignant users may conspire with each other to get sensitive files past their rights. Also, these sorts of malicious users are caught by the improved security algorithm.

Cloud servers. The cloud service provider deals with a cloud to provide data storage service. Data owners encrypt their data files and store them in the cloud for sharing to data consumers. To access the entire shared data files, the data consumers should download encrypted data files of their attention from the cloud and after that decrypt them.

Improved security. This recently outlined algorithm oversees giving improved security to the data stored. It creates the email message to the data owner that some attack has happened to the malicious user (Chase & Chow, 2009). At that point, the data owner can take the additional activity by blocking that user. If any attacker adjusts some file, then it advises to the data owner about the changes when the data owner verifies that file.

Performance Analysis

In this segment, they dissect the execution of their plan by contrasting and the Ruj et al.'s (2011) DACC scheme and their past scheme in the conference adaptation, regarding

Storage overhead, correspondence cost and calculation productivity. They direct the correlation under a similar security level. Let $|p|$ be the component measure in the G, GT, Z_p . Assume there are n_A authorities in the formwork and each attribute authority AA_{aid} oversees n_{aid} qualities. Let n_U and n_O be the aggregate number of users and owners in the formwork individually. For a user uid , let $n_{uid, aid} = |S_{uid, aid}|$ indicate the number of attributes that the client uid acquired from AA_{aid} . Give n_{aid} be the aggregate number of attributes in the ciphertext.

Storage overhead. The storage overhead is a standout amongst the most outstanding issues of the entrance control scheme in cloud storage formworks. Let $a = \sum_{k=1}^{n_A} n_{aid}$ indicate the aggregate number attributes in the framework and $a_{uid} = \sum_{k=1}^{n_A} n_{uid, aid}$ report the total amount of attributes the client uid holds from every one of the AAs in the formwork.

Storage overhead on each AA. Each AA needs store the data of the considerable number of attributes in its area. Furthermore, in, each AA_{aid} likewise needs to store the secret keys from all the owners, where the storage overhead on every AA is moreover direct to the aggregate number of owners n_O in the formwork. In their scheme, other than the storage of attributes, each AA_{aid} additionally needs to store a public key and a secret key for each user in the formwork. Therefore, the storage overhead on every AA in their plan is likewise straight to the quantity of user's n_U in the formwork (International Journal of Computer Science Trends and Technology (IJCT), 2016).

Storage overhead on each owner. The public parameters contribute the first storage overhead on the owner. Other than general society parameters, in, owners are required to re-encode the ciphertexts and in owners are required to produce the refresh data amid the

renouncement, where the owner ought to likewise hold the encryption mystery for each ciphertext in the formwork. This about a substantial stockpiling overhead on the owner, particularly at the point when the quantity of ciphertext is extensive in cloud storage formworks.

Storage overhead on each user. The Storage overhead on every client in their plan originates from the mystery keys issued by every one of the AAs. In any case, in, the Storage overhead on every client comprises of both the mystery keys issued by all the AAs and the ciphertext segments that related with the renounced attributes x . Since when the ciphertext decrypted, some of its parts identified with the repudiated ascribes ought to be sent to each non-denied client who holds the disavowed attributes. In, the client needs to keep many mystery keys for various data owners, which implies that the storage overhead on every client is added directly to the number of owners nO in the framework (International Journal of Computer Science Trends and Technology (IJCST), 2016).

Storage overhead on server. The ciphertexts contribute the principle stockpiling overhead on the server (where they do not consider the encrypted data which are encoded by the symmetric content keys).

UML Diagrams

UML remains for Unified Modeling Language. UML is a standardized broadly useful modeling language in the field of object-oriented software engineering. The standard is managed and was made by, the Object Management Group. The objective is for UML to end up noticeably an ordinary language for creating models of object-oriented computer software. In its present shape, UML has involved two noteworthy segments: A Meta-model and

notation. Later, some strategy or process may likewise be added to; or connected with, UML. The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of programming formwork, as well concerning business modeling and other non-programming formworks (Ruj et al., 2011). The UML represents to a gathering of best designing practices that have demonstrated efficacy in the modeling of large and complex systems. The UML is a critical piece of creating objects oriented software and the software development process. The UML utilizes graphical documentations to express the outline of software projects.

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

- Offers users accessible, vivid Unified Modeling Language so that they can develop and interchange significant models.
- Provide extendibility and specialization mechanisms to extend the core concepts.
- Be independent of programming languages and development process.
- Provide a formal basis for understanding the modeling language.
- Encourage the growth of OO tools market.
- Support higher level development concepts such as collaborations, formworks, patterns, and components.
- Integrate best practices.

Proxy layer. This intermediary layer goes about an interface between the users and rest of the servers accessible in the cloud.

Cloud data server layer. Data server has two unique substances can be perceived as the cloud users and the cloud service provider. Different data servers proposed in this plan to maintain a strategic distance from the movement.

Cloud data storage server layer. All the data and the files are put away in these multiple storages which are set apart by both individual users and organizations. Comparative to data server there are numerous Storage servers are acquainted with massive handle volume of data.

Cloud key server layer. Multiple key servers proposed in this plan for efficient computation attribute revocation method. The key server is utilized to store a secret key that is encoded or divided by the key splitter.

Cloud consumer's layer. Cloud users are the one who has the data access in the cloud and relies upon cloud for data computation and change. Cloud consumers can be both users and individual organizations.

Cloud service provider (CSP). This layer claims, assembled furthermore, deals with the Storage servers in cloud way and functions as live cloud computing formworks. The access policies over the attributes characterized by the owner and encode the contents keys under the plans. The owner at that point sends the encrypted data together with the ciphertexts to the cloud server. The user can decrypt the ciphertext simply when his attributes fulfill the access policy characterized in the ciphertext. Users decode the distinctive number of secret keys with various characteristics, and from same data, extranormal data received (Kumar & Lakshmi, 2015).

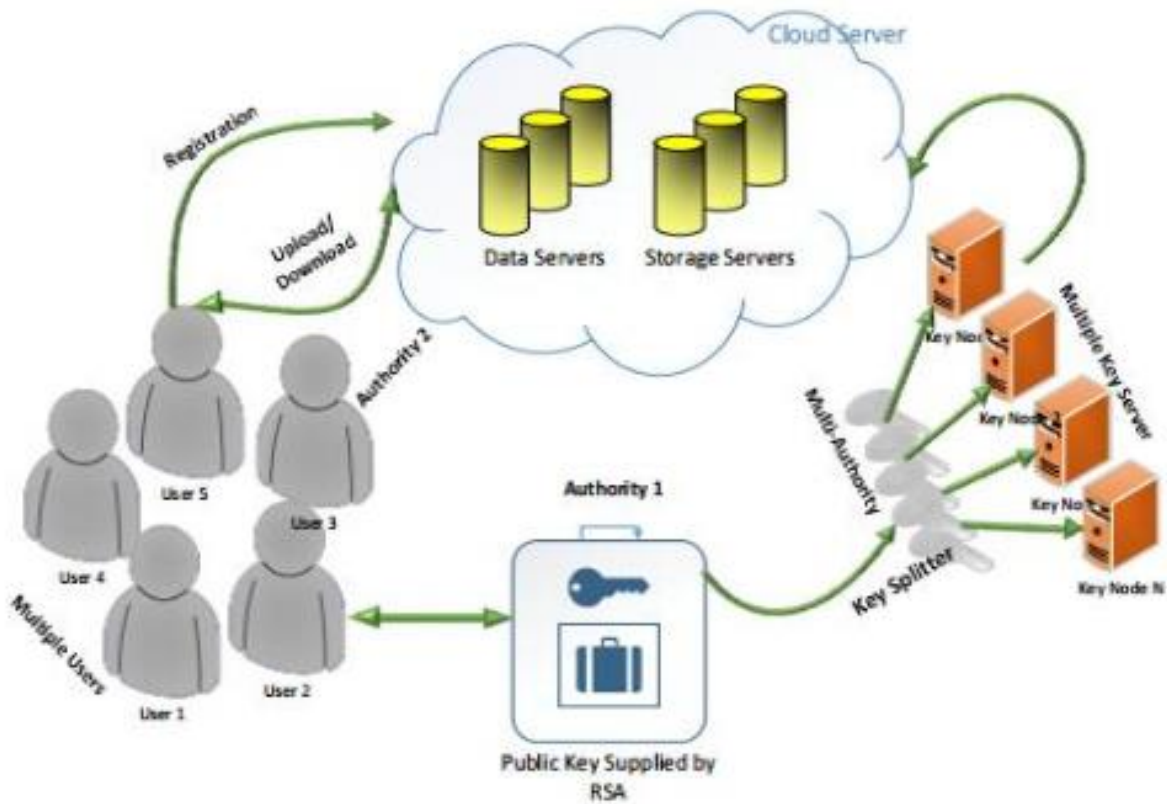


Figure 6. Architecture showcasing all the key elements (Kumar & Lakshmi, 2015).

Security Model

The following supposition made in multi-authority cloud storage formworks: In the formwork, the CA completely trusted. It won't coordinate secretly with any user and ought to keep from decrypting the ciphertext by itself. The trusted AA can defile by the adversary. The server is interested in the substance of data to be encrypted or to the message got. Be that as it may, the server is genuine and will execute the task assigned by each attribute authority accurately. The untrustworthy user may co-work covertly to get the unapproved access to data (Kumar & Lakshmi, 2015).

The author proposes another revocable multi-authority CP-ABE protocol in light of the single-authority CP-ABE introduced by Lewko and Waters (2011). That is author extend it to the multi-authority situation and make it revocable. Author apply the systems to Chase's multi-expert CP - ABE protocol to tie the secret keys created by a various authority for a similar user and keep the collusion attack. In particular, author isolates the usefulness of the authority into a worldwide global certificate authority (CA) and different attribute authorities (AAs). The CA sets up the formwork and acknowledges the enlistment of users and AAs in the formwork. It allocates a worldwide user identity uid to every user and global authority identity aid help to each quality expert in the formwork. Since the uid is internationally remarkable in the formwork, secret keys issued by various AAs for the same uid can tie for decoding. Moreover, since every AA is related with aid, each attribute is recognizable despite the fact that a few AAs may issue a similar characteristic to manage security issue in Multi-Authority Attribute-Based Encryption. Rather than utilizing the formwork extranormal public key to encrypt data, author's scheme requires all attribute authorities to create their particular public keys and uses them to encode data together with the global public parameters. It keeps the certificate authority in the scheme from decoding the ciphertexts. To take care of the attribute revocation issue, author appoints a version number for each attribute. To enhance the efficiency, author assigns the workload of ciphertext refresh to the server by utilizing the intermediary re-encryption strategy. With the end goal that the recently joined client is additionally ready to decrypt the beforehand published data, which encrypted with the past public keys, on the off chance that they have adequate attributes (Jia & Yang, 2014).

To accomplish secure data sharing for dynamic groups in the cloud, Authors hope to join the group signature and dynamic broadcast encryption techniques. This gathering mark scheme empowers users to utilize the cloud assets namelessly, and the dynamic broadcast encryption method enables data owners to safely share their data documents with others including new joining user.

Security instinct. As in former property based encryption conspires the principle challenge in outlining their scheme was to keep against assaults from plotting users. Like the plan of Sahai and Waters their arrangement randomizes user's private keys to such an extent that they can't join; in any case, in their answer, the mystery sharing must be installed into the ciphertext to the private keys. Keeping in mind the end goal to decrypt an aggressor plainly should recuperate $e(g, g)\alpha s$. Keeping in mind the end goal to do this the assailant must match C from the ciphertext with the D segment from some client's private key. It will come about in the coveted esteem $e(g, g)\alpha s$, however, blinded by a few esteem $e(g, g)r s$.

This esteem can be blinded out if and just if enough the client has the right key segments to fulfill the mystery sharing plan installed in the ciphertext. Arrangement assaults will not help since the blinding worth is randomized to the arbitrariness from a specific client's private key. While they portrayed their plan to be secure against picked plaintext assaults, the security of their project can productively be reached out to pulled ciphertext assaults by applying an arbitrary prophet system, for example, that of the Fujisaki-Okamoto change. On the other hand, they can use the designation instrument of their plan and apply the Canetti, Halevi, what's more, Katz technique for accomplishing CCA-security. Proficiency. The efficiencies of the critical age and encryption calculations are both genuinely evident.

The encryption calculation will require two exponentiations for each leaf in the ciphertext's entrance tree. The ciphertext size will incorporate two gathering components for each tree leaf. The key age calculation requires two exponentiations for each credit given to the client, what's more, the private key comprises of two gathering components for each characteristic. In its least complicated form, the unscrambling calculation could require two pairings for each leaf of the get to the tree that is coordinated by a private key characteristic also, (at most²) one exponentiation for every hub along away from such a leaf to the root (Waters, 2011).

Be that as it may, there may be a few approaches to fulfill an arrangement, so a more rational calculation may attempt to improve along these lines. In their execution portrayal in Section 5, they portrayed different execution improvements. Key-revocation and numerical Attributes. Key revocation usually is a troublesome issue in identity-based encryption and related plans. The central challenge is that since the gathering encoding the information does not acquire the collector's authentication online, he is not ready to check if the accepting party denied. In Attribute-based encryption, the issue is much more precarious since a few distinct users may coordinate the unscrambling strategy. The typical arrangement is to annex to each of the characters or to elucidate Attributes a date for when the quality terminates. For example, Pirretti et al. (2006) recommend expanding each condition with a lapse date. For instance, rather than utilizing the Attribute "Software engineering" they may utilize the Attribute "PC Science: Oct 17, 2006". This sort of strategy has a few weaknesses. Since the attributes fuse a correct date, there must be the concession to this between the gathering encoding the information and the key issuing expert. On the off chance that they wish for a conference to have the storage to indicate approach about renouncement dates on a fine-

grained scale, users will be compelled to regularly go to the expert and keep up a lot of private critical stockpiling, a key for each day and age. Preferably, they might want a property based encryption formwork to enable a pivotal expert to give out a private key with some termination date X as opposed to a different key for each day and age before X . At the point when a gathering encodes a message on some date Y , a client with a core terminating on date X ought to have the storage to decrypt if $X \geq Y$ and whatever left of the approach coordinates the client's attributes. In this way, extranormal termination dates can give to diverse users, and there does not should be any nearby the coordination between the parties encoding information and the expert (Sahai & Waters, 2005).

Analysis and Discussion

The author proposes another threshold multi-authority CP-ABE gets to control scheme TMACS, out in the public cloud storage, in which all AA s mutually deal with the entire attribute set and offer the master key α . Taking the favorable position of (t, n) threshold secret sharing, by collaborating with any t AA s, a legitimate user can create his/her secret key. In this way, TMACS maintains a strategic distance from anyone AA being a single-point bottleneck on both security and performance. The analysis comes about demonstrate that creator's entrance control conspire reliable and secure. It can without much of a stretch find proper estimations of (t, n) to influence TMACS to secure when not precisely authorities are traded off, additionally strong when no less than authorities are alive in the formwork. Further, given efficiently consolidating the traditional multi-authority conspire with TMACS, develop a hybrid scheme that is more appropriate for the whole situation. This plan addresses attributes coming from distinctive authorities, security and formwork-level power (Li, 2015).

The author analyzes the inadequacy of DAC-MACS in managing attribute revocation. Additionally, discovered that, if a revoked client needs to get to the unauthorized content whose access policy can fulfill his/her denied attributes. The primary activity is to utilize creator's proposed attack algorithm to change the new- version ciphertext to the old- version one on the off chance that he/she can scheme with the cloud provider organization to get enough ciphertext update keys. The security vulnerability exists because DAC-MACS wrongly utilize a bidirectional re-encryption scheme in the ciphertext updating method. This vulnerability enables any gathering to re-encrypt the ciphertext between old-version and new version, only if he/she can get the CUK s between these two versions (Hong et al., 2015).

Author's proposed schemes accomplished fine-grained privilege control and identity anonymity while directing privilege control relies upon client's attributes. More essential is, this formwork can endure up to N-two authority trade-off, which is for the most part incline toward especially in Internet-based cloud computing condition. Too conducted security and performance analysis which demonstrates that AnonyControl both secure and proficient for cloud storage system. The AnonyControl acquires the security from the AnonyControl furthermore, accordingly is proportionally safe as it. However, additional correspondence overhead is brought about amid the 1-out-of-n oblivious transfer (Jung et al., 2015).

The author proposed a revocable multi-authority CPABE scheme that could bolster proficient attribute revocation and efficient data access control scheme for multi-authority cloud storage systems. Author additionally demonstrated that this project was provable secure in the random oracle model. The revocable multi-authority CPABE is a reliable system,

which can be connected to any remote storage systems and on the online social networks, etc. (Jia & Yang, 2014).

Authors outlined a secure data sharing scheme Mona for dynamic groups in an untrusted cloud. In Mona, users can import data to others in the gathering without uncovering identity security to the cloud. Too, Mona is productive in client denial and new client joining. Even more exceptionally, productive client revocation can be accomplished by public repudiation list without refreshing the private keys of the other outstanding users, and new users can straightforwardly decrypt files put away in the cloud without their support. Additionally, the Storage overhead and the encryption computation cost are steady. By analysis, it demonstrated that proposed scheme was fulfill the security prerequisites and proficiency (Ishii et al., 2013).

Table 3

Comparison between Various Data Access Control Scheme with Attribute-Based Encryption

Data Access Control Techniques	Advantages	Disadvantages
Threshold multi-authority ciphertext-policy CP-ABE access control scheme (TMACS)	<ol style="list-style-type: none"> 1) It satisfies the scenario of attributes from different AAs 2) It can achieve security and system-level robustness 	Reusing of the master key shared among multiple attribute authorities (AAs)
Comments and corrections of CP-ABE	Analyze the shortcoming of DAC-MACS in dealing with attribute revocation, main construction proved it secure	Security vulnerability
Privilege control scheme AnonyControl AnonyControl-F	<ol style="list-style-type: none"> 1) Able to protect user's privacy against single authority 2) Tolerant against authority 	<ol style="list-style-type: none"> 1) Data confidentiality 2) Personal information defined by each user's attributes set is at risk 3) Resilient in security breach
Attribute revocable multi-authority CP-ABE scheme	<ol style="list-style-type: none"> 1) It incurs less communication and cost and computation cost, and is secure 2) It can achieve both backward and forward security 	Lack of efficiency
Secure multi-owner data sharing scheme MONA	<ol style="list-style-type: none"> 1) Reduced the computation overhead to encrypt files and cipher text size 2) The ciphertext size is constant and independent of revocation users 	<ol style="list-style-type: none"> 1) User compute revocation parameters to project the confidentiality 2) Computation overhead of the encryption

Use Case Diagram

A use case chart in the Unified Modeling Language (UML) is a sort of behavioral diagram characterized by and made from a Use-case analysis. Its reason for existing is to show a graphical description of the functionality provided by a system regarding actors, their objectives (represented as use cases), and any conditions between those use cases (Ruj et al., 2001). The principal reason for a use case outline is to show what formwork functions performed for which actor. Parts of the performing actor in the formwork can be delineated.

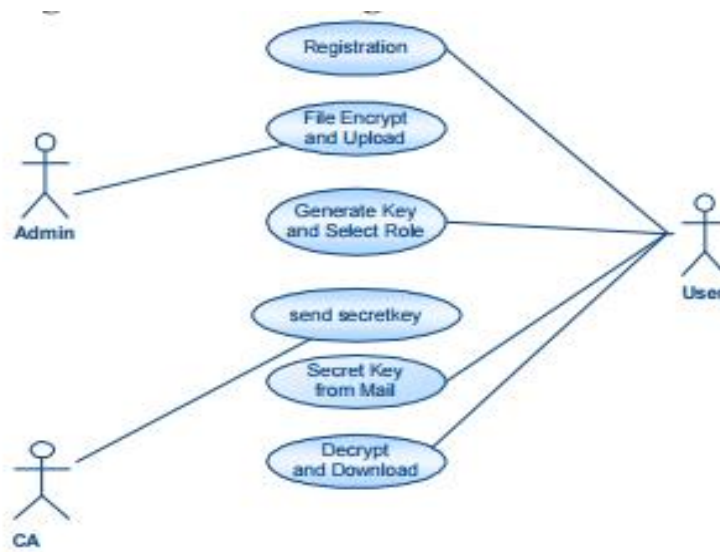


Figure 7. Use case diagram (Ruj et al., 2011).

Class Diagram

In application engineering, a class diagram in the Unified Modeling Language (UML) is a type of constant structure representation that depicts the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains data.

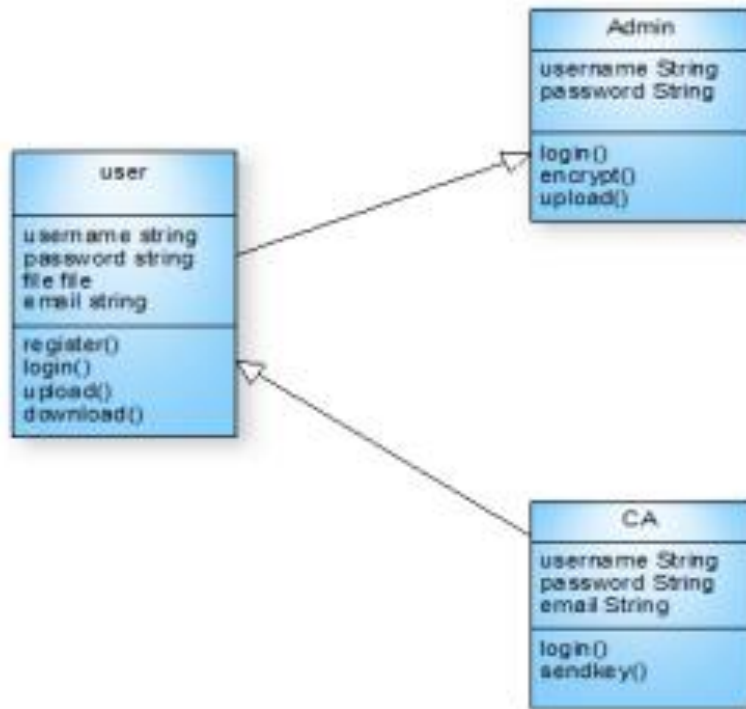


Figure 8. Class diagram (Ruj et al., 2011).

Sequence Diagram

A sequence diagram in Unified Modeling Language (UML) is a type of connection layout that depicts how the processes operate with each other and in what order it should flow. It is a systematization of the Message Sequence Chart. These in some cases called event diagrams, event scenarios, and timing diagrams.

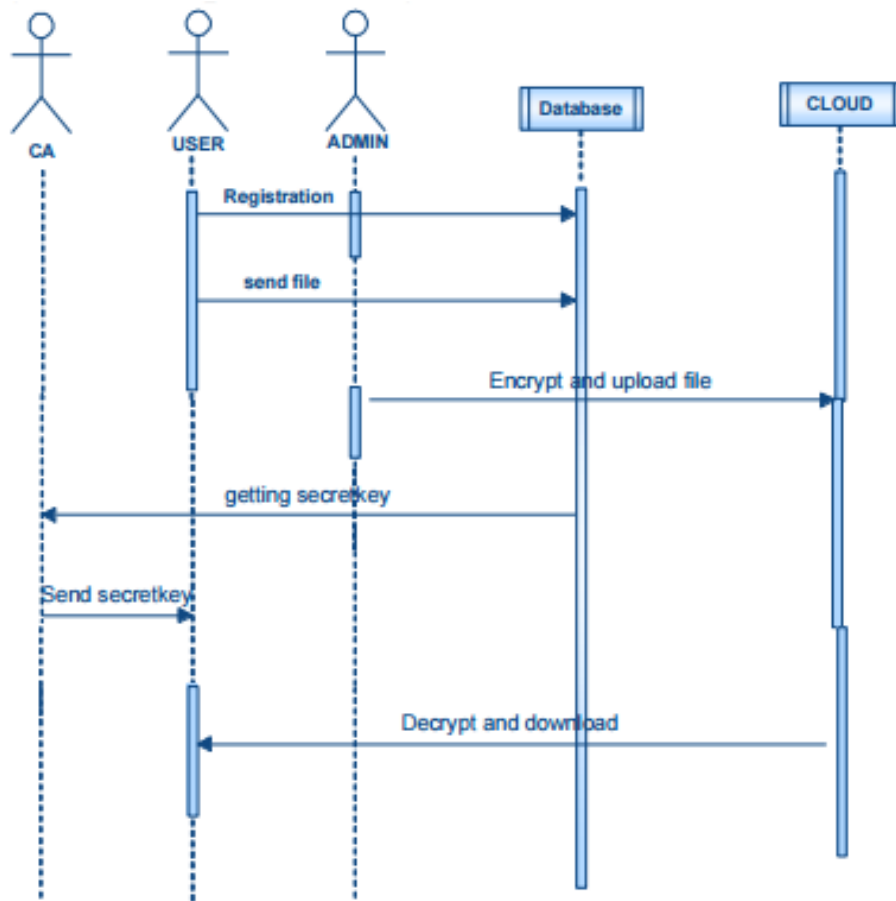


Figure 9. Sequence diagram (Ruj et al., 2011).

Activity Diagram

These diagrams are a graphical depiction of the workflows of stepwise activities and actions with support for the decision, iteration and concurrency. In the Unified Modeling Language, these activity diagrams can be used to show the business and operational in some step-by-step workflows of segments in a formwork. An activity diagram demonstrates the general stream of control.

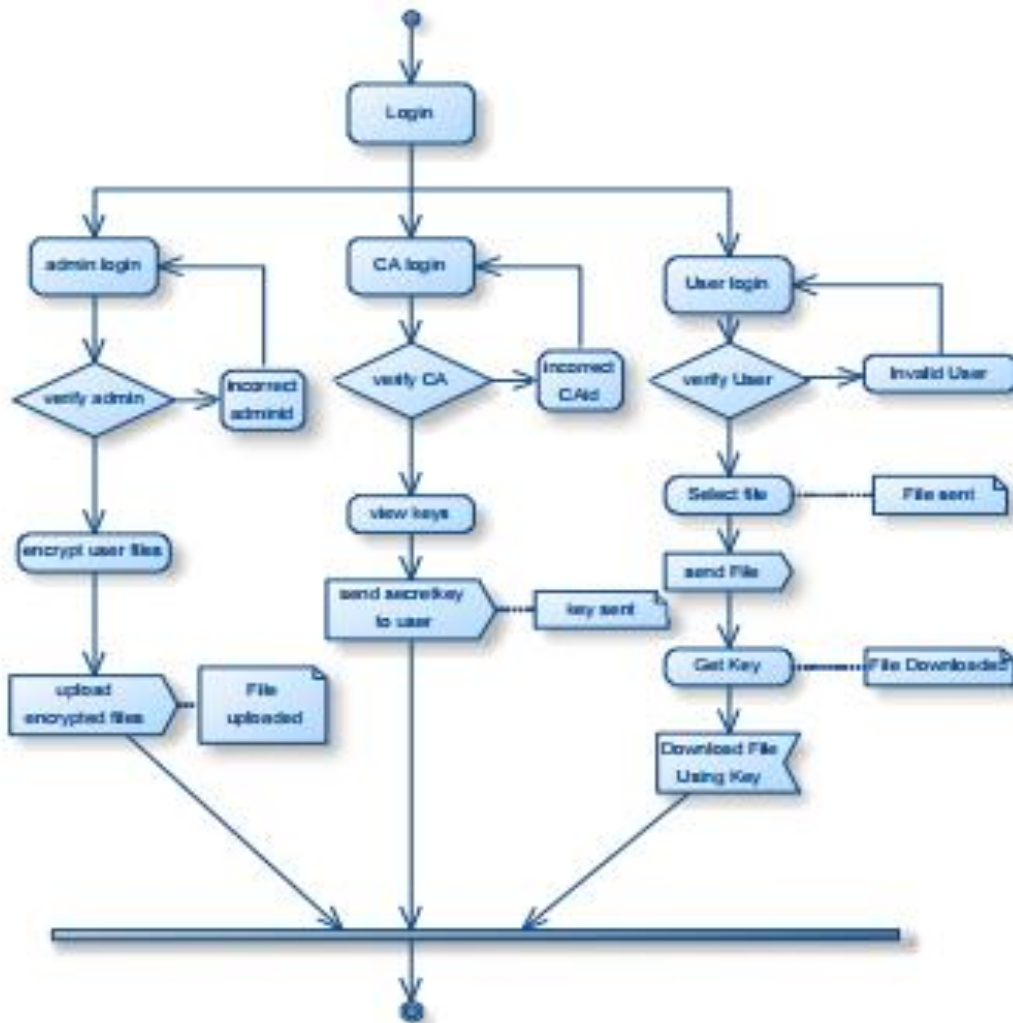


Figure 10. Activity diagram (Ruj et al., 2011).

Input design. The input design is the connection between the data formwork and the user. It includes the creating determination and procedures for data preparation, and those means essential to put transaction data into a usable shape for handling can accomplish by investigating the PC to read data from a written or printed document, or it can happen by having individuals entering the data straightforwardly into the formwork. The plan of input concentrates on controlling the measurement of info required, controlling the errors,

maintaining a strategic distance from deferral, staying away from additional means and keeping the procedure basic. The input outlined in such a route in this way, to the point that it gives security and usability withholding the protection (Rao & Pradeep, 2015).

Input Design considered the accompanying things:

- What data ought to given as input
- How the data ought to be organized or coded?
- The dialog to guide the operating personnel in providing information.
- Methods for preparing input validations and steps to follow when error occurs (Rao & Pradeep, 2015).

Objectives.

1. Input Design is known as the process of changing a user-oriented description of the input into a computer-based system. This design is essential to avoid errors in the data input process and show the correct direction to the management for getting accurate data from the computerized system.
2. It obtains by developing user-friendly screens for the data entry to handle the large volume of data. The goal of designing input is to make data entry more accessible and to be free from errors. The data entry screen designed in such a way that all the data performed. It also provides file viewing facilities (Rao & Pradeep, 2015).
3. When the data entered, it will check for its validity. Data can open with the help of screens. Exact messages provided for when needed so that the user will not be in maize for an 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 prerequisites of the end user and presents the data. In any formwork consequences of preparing are conveyed to the users and another formwork through outputs. In output design, it resolved how the data is to uproot for immediate need and furthermore the hard copy output. It is the most critical and direct source data to the client. Proficient and intelligent output design enhances the formwork's relationship to help output design. (Rao & Pradeep, 2015).

- Planning the output should proceed in an organized, well thought out manner; the right result must obtain while assuring that each result element depicted so that people will find the system can use smoothly and efficiently. When analyzing the design computer output, they should identify the significant production that is needed to meet the requirements.
- Select methods for presenting the data.
- Design a document, report, or other formats that contain data produced by the system. The output form of a data system should accomplish one or more of the following objectives.
- Convey data about past activities, status or projections of the Future.
- Signal notable events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.

Data Collection

Resources are collected from articles, journals. The secondary resources will be gathered and analyzed from internet source and books.

System Requirements (Minimum)

Hardware Requirements

- System: Pentium IV 2.4 GHz and above
- Hard Disk: 40 GB(required)
- Monitor: 15 VGA Color.
- Mouse: Logitech/Any preferred
- Ram: 512 Mb or More

Software Requirements

- Operating system: Windows XP/7.
- Coding Language: JAVA/J2EE
- IDE: NetBeans 7.4 or Eclipse
- Database: MYSQL

GlassFish server. It is an open-source tool server started by Sun Microsystems to use on Java EE platform, and now it is supported by The Oracle Corporation. The recommended version is known as Oracle GlassFish Server. It is a free programming, dual-licensed under two free programming licenses: The Common Development and Distribution License (CDDL) and the GNU General Public License (GPL) with the classpath exemption.

GlassFish is the reference execution of Java EE and in that Storage, underpins Enterprise JavaBeans, JPA, JavaServer Faces, JMS, RMI, JavaServer Pages, servlets, and so forth. It enables developers to make enterprise applications that are convenient and versatile, and that incorporate with legacy innovations. Optional components can likewise introduce for other administrations.

Depending on a modular kernel powered by OSGi, this server runs straight over the Apache Felix execution. It additionally keeps running with Equinox OSGi or Knopflerfish OSGi runtimes. HK2 abstracts the OSGi module formwork to give segments, which can likewise be administrations. Such administrations can be found and infused at runtime. GlassFish depends on source code released by Sun and Oracle Corporation's TopLink diligence formwork. It utilizes a subsidiary of Apache Tomcat as the servlet compartment for serving Web content, with an additional segment called Grizzly which employs Java New I/O (NIO) for adaptability and speed (Wikipedia, n.d.)

These are the minimum system requirements used to set up the software and implement the project so that it can function without any interruptions and errors. Also, they are using the Windows OS as it is user-friendly and used by the majority of the population. The code is written in java language as it is designed for flexibility, allowing developers to write the code that would run on any machine regardless of a platform.

NetBeans IDE-the smarter and faster way for coding. NetBeans IDE is efficiently used to develop Java desktop, mobile, and web applications, and in addition to the HTML5 applications with HTML, JavaScript, and CSS. The IDE likewise gives an incredible set of

tools for PHP and C/C++ developers. It is a free and public source and has a vast group of users and developers around the globe.

Best support for latest java technologies. NetBeans IDE is the authority IDE for Java 8. With its editors, code analyzers, and converters, they can rapidly and efficiently redesign the applications to utilize new Java 8 language develops, for example, lambdas, functional operations, and technique references. Group analyzers and converters are given to look through different applications in the meantime, coordinating cases for transformation to new Java 8 language builds. With its continually enhancing Java Editor, numerous rich highlights and a broad scope of devices, formats, and tests, NetBeans IDE sets the standard for creating with front-line advancements out of the container (Netbeans.org, n.d.).

Quick and smart code editing. An IDE is more than a text editor. The NetBeans Editor indents lines, relates words and brackets, and then features source code syntactically and semantically. It lets us effortlessly refactor code, with a scope of convenient and capable devices, while it additionally gives code layouts, coding tips, and code generators. The editor supports numerous languages from Java, C/C++, XML, and HTML, to PHP, Groovy, Javadoc, JavaScript, and JSP. Since the proofreader is extensible, they can connect to help for some different dialects.

Efficient and straightforward project management. Keeping an unmistakable review of large applications, with many folders and files, and a great many lines of code is an overwhelming task. NetBeans IDE gives diverse perspectives of the data, from various project windows to supportive devices for setting up their applications and overseeing them productively, giving them a chance to penetrate down into their data rapidly and effortlessly,

while giving them versioning devices using Subversion, Mercurial, and Git joining out of the case. At the point when new developers join their undertaking, they can comprehend the structure of the application because their code is efficient (Netbeans.org, n.d.).

MySQL query browser. It is an independent authority GUI for the favorite MySQL database server. It fills the gap in MySQL Administrator by enabling them to perform queries straightforwardly onto any pattern that they pick. They can either compose the questions by hand or utilize the limited query generation that is a piece of MySQL Query Browser. Like MySQL Administrator, MySQL Query Browser requires the GTK+ runtime libraries to work (Linuxquestions, n.d.).

SQLyog. SQLyog is a GUI tool for the RDBMS MySQL. It produced by Webyog, Inc. situated in Bangalore, India and Santa Clara California. SQLyog v0.9 was first released to the general population in 2001 as following eight months of development. SQLyog was accessible for free however with closed source code, until v3.0 when it made an entire business programming. These days SQLyog is dispersed both as free programming for free and add a few paid, exclusive, versions (Wikipedia, n.d.).

Chapter IV: Conclusion

Proposed a revocable decentralized data access control system which can support effective characteristic revocation for multi-expert cloud storage formworks. It eliminates decryption overhead of users as indicated by attributes. This safe attribute-based encryption system for secure data security that shared in the cloud. This revocable multi-authority data gets to conspire with correct outsourced decryption, and it is protected and unquestionable. This scheme will be a promising system, which can connect to any remote storage systems and online social networks, etc. They made a framework for Ciphertext-Policy Attribute-Based Encryption. Their formwork takes into consideration another sort of encrypted access control where client's private keys determined by an arrangement of attributes and a gathering encrypting information can identify an approach over these Attributes indicating which users can decrypt. Their formwork permits approach to be communicated as any monotonic tree get to structure and is impervious to intrigue assaults in which an aggressor may acquire different private keys. At last, they gave a usage of their formwork, which incorporated a few enhancement systems. Later on, it is intriguing to consider property based encryption formworks with various sorts of expressibility. While Key-Policy ABE and Ciphertext-Policy ABE catch two exciting and complementary kinds of frameworks, there unquestionably exist different types of formworks. The essential test in this profession is to locate another formwork with fine structures of articulation that deliver more than a self-assertive blend of systems. One confinement of their formwork is that it is demonstrated secure under the non-exclusive gathering heuristic. They trust a vital undertaking is shown a formwork safe under a

more standard and non-intuitive assumption. This kind of work would intrigue regardless of the possibility that it brought about a direct loss of proficiency from their existing formwork.

References

- Basri, S. R., & Rashmi, K. (2015, May). Attribute based revocable data access control for multi authority cloud storage. *International Journal of Advanced Research in Computer Engineering & Technology*, 4(5), 1887-1890.
- Bethencourt, J., Sahai, A., & Waters, B. (2007). Ciphertext-policy attribute-based encryption. security and privacy, 2007. SP '07. *IEEE Symposium on Encryption, Security and Privacy*, pp. 321-334.
- Chase, M., & Chow, S. S. M. (2009). Improving privacy and security in multi-authority attribute-based encryption. In *16th ACM Conference on Computer and Communication Security (CCS'09)*, pp. 121-130.
- Goyal, V., Pandey, O., Sahai, A., & Waters, B. (2006). Attribute based encryption for fine-grained access control of encrypted data. In *ACM conference on Computer and Communications Security (ACM CCS)*.
- Hong, J., Xue, K., & Li, W. (2015). Comments on "DAC-MACS: Effective data access control for multi-authority cloud storage systems"/Security analysis of attribute revocation in multiauthority data access control for cloud storage systems. *IEEE Transactions on Information Forensics and Security*, 10(6).
- Hur, J., & Noh, D. K. (2011). Attribute-based access control with efficient revocation in data outsourcing systems. *IEEE Transactions on Parallel Cloud Systems*, 22(7), 1214-1221.
- International Journal of Computer Science Trends and Technology (IJCST)*, 4(3). (2016).

- Ishii, H., Tempo, R., & Bai, E.-W. (2013). Mona: Secure multi-owner data sharing for dynamic groups in the cloud, *IEEE Transactions on Parallel and Distributed Systems*, 24(6).
- Jahid, S., Mittal, P., & Borisov, N. (2011). Easier: Encryption-based access control in social networks with efficient revocation. In *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*, pp. 415-425, Hong Kong, China, March 22-24, 2011.
- Jia, X., & Yang, K. (2014). Expressive, efficient and revocable data access control for multi-authority cloud storage. *IEEE Transactions on Parallel and Distributed Systems*, 25(7), 1735-1744.
- Jung, T., Li, X.-Y., Wan, Z., & Wan, M. (2015). Control cloud data access privilege and anonymity with fully anonymous attribute-based encryption. *IEEE Transactions on Information Forensics and Security*, 10(1), 190-199.
- Kumar, P., & Lakshmi, S. N. (2015). Efficient data access control for multi-authority cloud storage using CP-ABE. *International Journal of Computer Engineering in Research Trends*, 2(12), 1182-1187.
- Lewko, A. B., & Waters, B. (2011). Decentralizing attribute-based encryption. In *Proceedings of the 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 568-588. Tallinn, Estonia, May 15-19, 2011.
- Li, J. (2015). Ensuring privacy in a personal health file system. *Computer*, 48(2), Article ID 7042698, 24-31.

- Li, M., Yu, S., Zheng, Y., Ren, K., & Lou, W. (2013). Scalable and secure sharing of attributes health files in cloud computing using attribute-based encryption. *IEEE Trans. Parallel Cloud Systems*, 24(1), 131-143.
- Li, R., Member, IEEE, Shen, C., He, H., Xu, Z., & Xu, C.-H. Member, IEEE. (2017). A lightweight secure data sharing scheme for mobile cloud computing. *IEEE Transactions on Cloud Computing*, PP(99).
- Li, W., Xue, K., Xue, Y., & Hong, J. (2014). TMACS: A robust and verifiable threshold multi-authority access control systems in public cloud storage. *IEEE Transactions on Parallel and Distributed Systems*, 27(5).
- National Conference on Recent Trends in Computer Science and Information Technology (NCRTCSIT)* (2016). Retrieved from <https://www.facebook.com/events/1678325239094584/>.
- Pirretti, M., Traynor, P., McDaniel, P., & Waters, B. (2006). Secure attribute-based systems. In *Proceedings of the 13th ACM Conference on Computing and Communication Security*, pp. 99-112. Alexandria, VA, October 30-Noember 3, 2006.
- Rajkumar, M. N., George, A., & Batley, B. (2014). An overview of multi-authority attribute based encryption techniques. *International Journal of Advanced Research in Computer and Communication Engineering*, 3(9), 8032-8036.
- Rao, T. V., & Pradeep, V. (2015). Expressive, efficient and revocable data access control for multi-authority cloud storage. *International Journal of Applied Sciences, Engineering and Management*, 4(6), 56-59.

- Ruj, S., Nayak, A., & Stojmenovic, I. (2011). DACC: Cloud access control in clouds. In *Proceedings of 10th IEEE International Conference on Trust, Security, and Privacy in Computing and Communications*, pp. 91-98, November 16-18-2011.
- Sahai, A., & Waters., A. S. (2005). Fuzzy identity based encryption. *Lecture Notes in Computer Science*, 3494, 457-473.
- Shamir, A. (1984). Identity based cryptosystems and signature schemes. In *Advances in Cryptology, Proceedings of CRYPTO'84*, Santa Barbara, CA., August 19-22, 1984.
- Srinath, M., & Obulesh, K. (2016). Expressive, efficient, and revocable data access control for multi-authority cloud storage. *International Journal of Advanced Technology and Innovative Research*, 8(8), 1564-1568.
- Wang, S., Zhou, J., Liu, J. K., Yu, J., Chen, J., & Xie, W. (2016). An efficient file hierarchy attribute-based encryption scheme in cloud computing. *IEEE Transactions on Information Forensics and Security*, 11(6), 1265-1277.
- Waters, A. L. (2011). Decentralizing attribute-based encryption. In *Proceedings of the 30th Annual International Conference on Theory and Applications of Cryptographic Techniques: Advances in Cryptology*, pp. 568-588. Tallinni, Estonia, May 15-19, 2011.
- Yang, K., & Jia, X. (2012). Attribute-based access control for multi-authority systems in cloud storage. In *Proceedings of the 32nd IEEE International Conference on Cloud Computing Systems (ICDCS'12)*, pp. 1-10, June 18-21, 2012.

Yu, S., Wang, C., Ren, K., & Lou, W. (2010). Attribute based data sharing with attribute revocation. In *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, pp. 261-270, Beijing, China, April 13-16, 2010.

Appendix: Screenshots

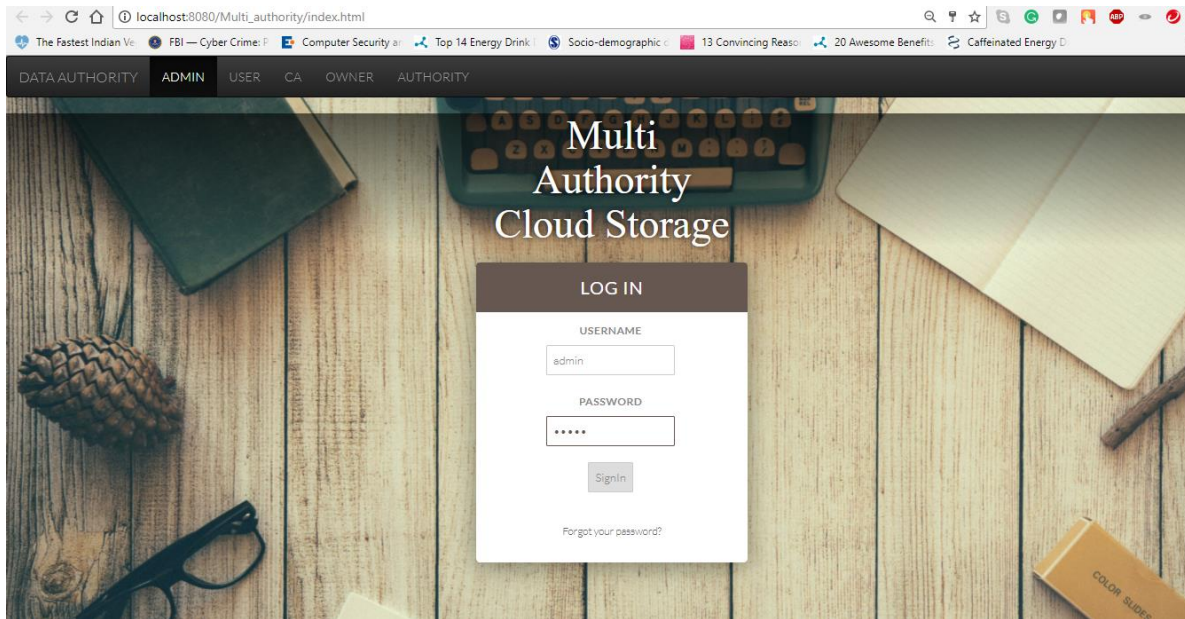


Figure 11. Login page for the Admin Module

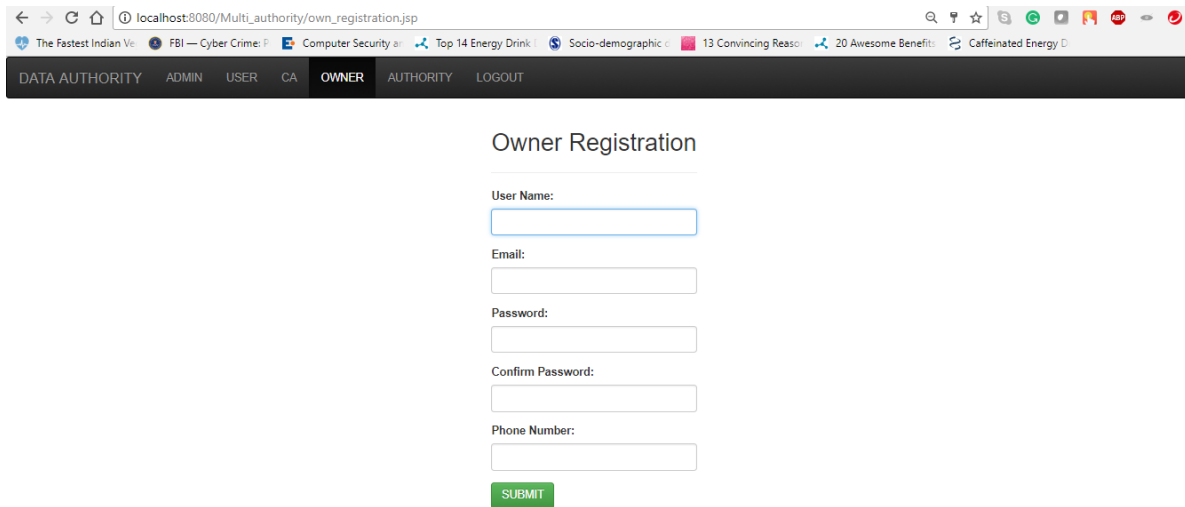


Figure 12. Registration page for the Owner Module

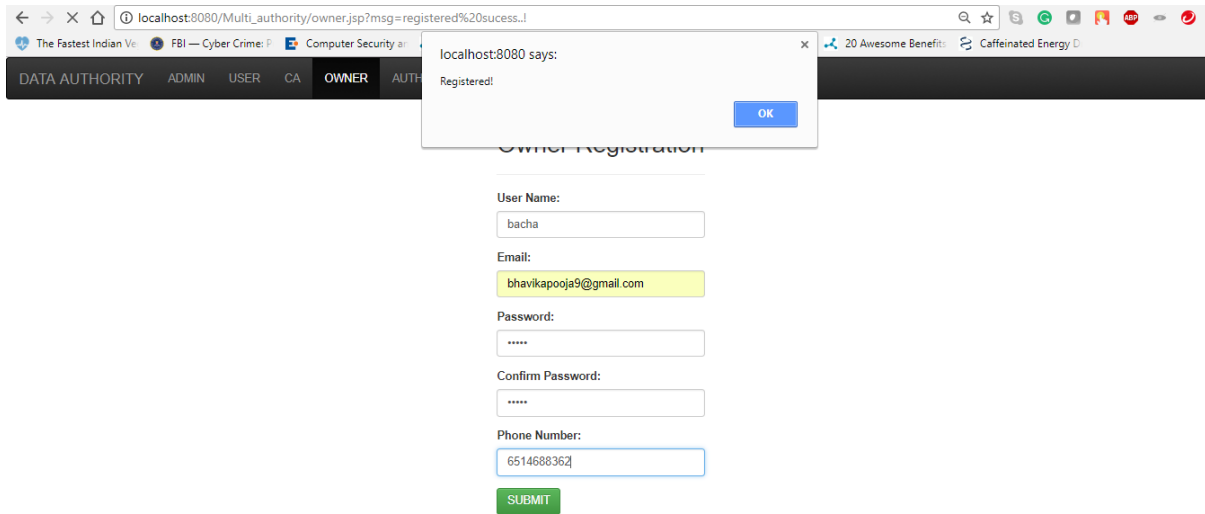


Figure 13. Confirmation on Owner's Registration

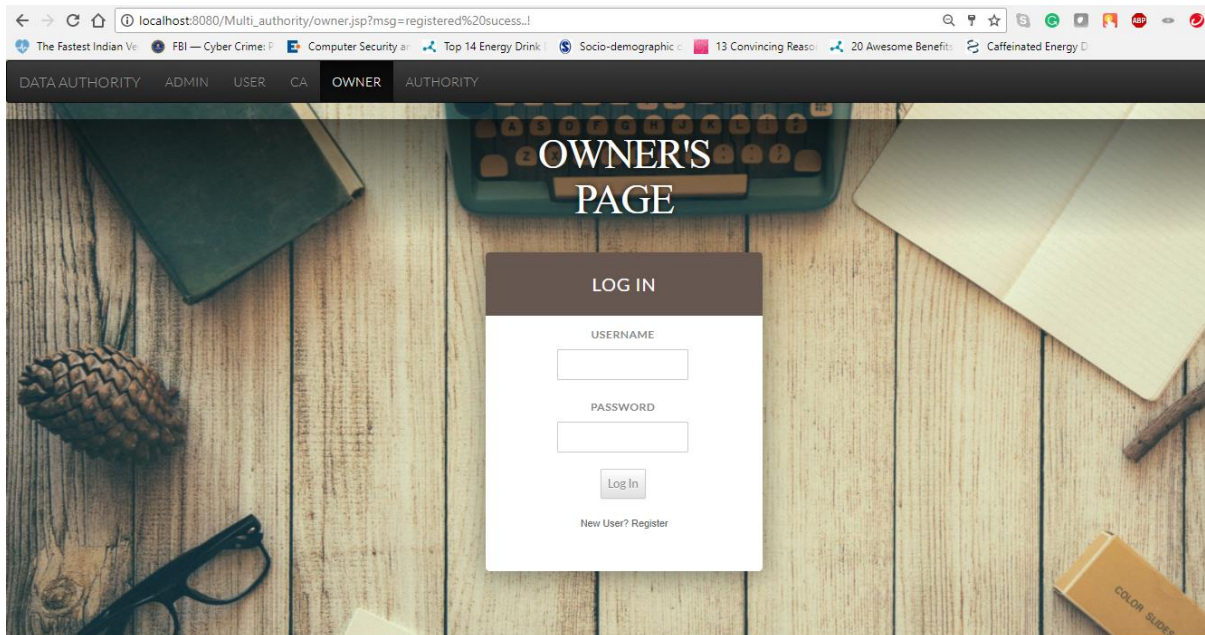


Figure 14. Login page for the Owner Module

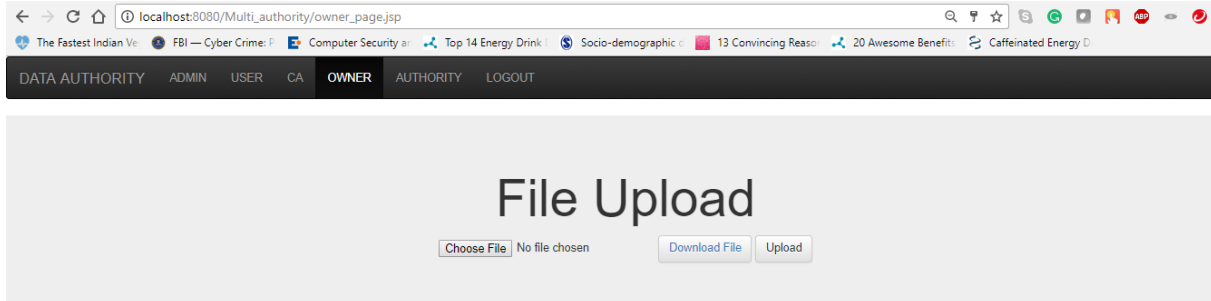


Figure 15. Owner Upload Page

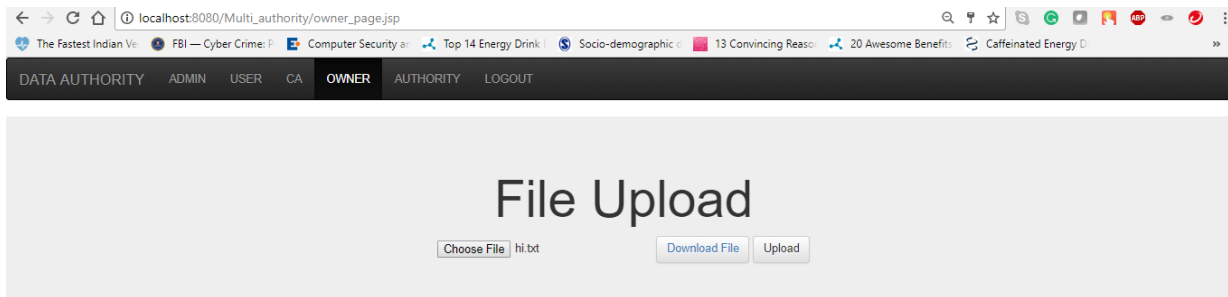


Figure 16. File has been selected for upload

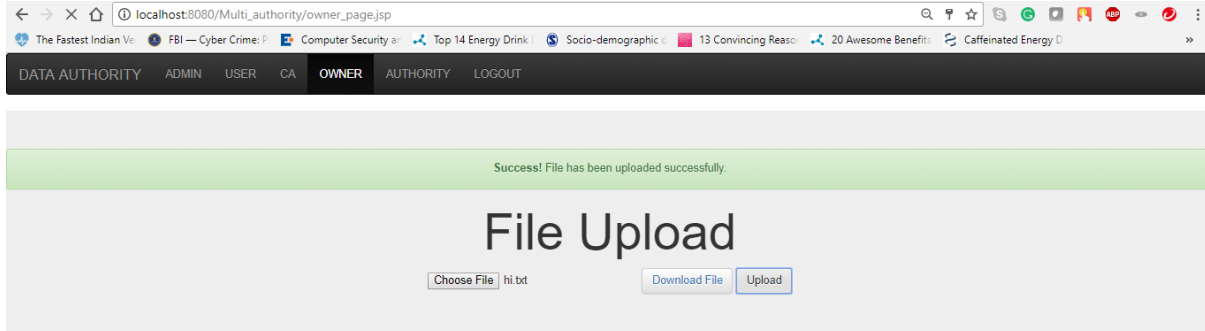


Figure 17. File has been uploaded successfully

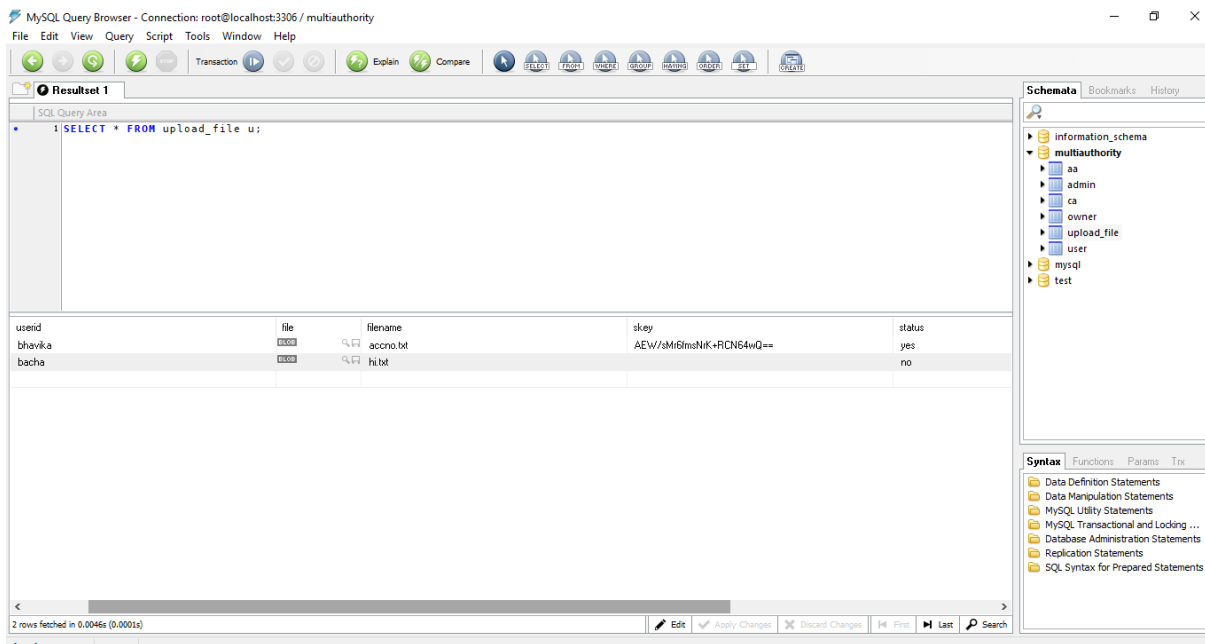


Figure 18. File has not been uploaded yet into the cloud with the status as 'no'

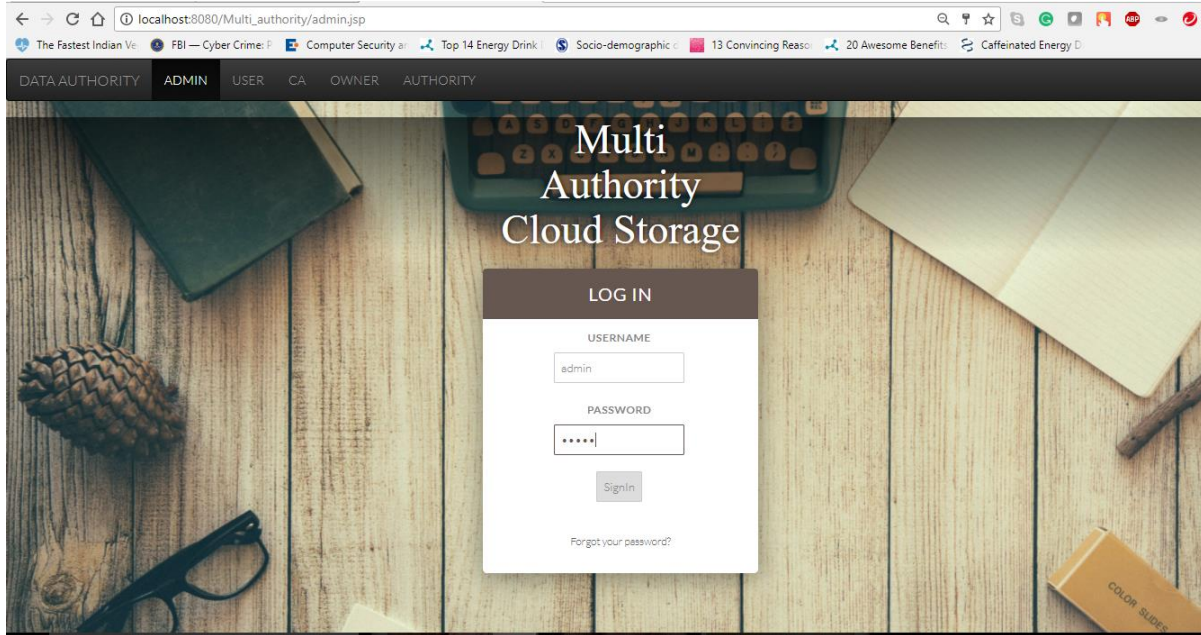


Figure 19. Admin Login page

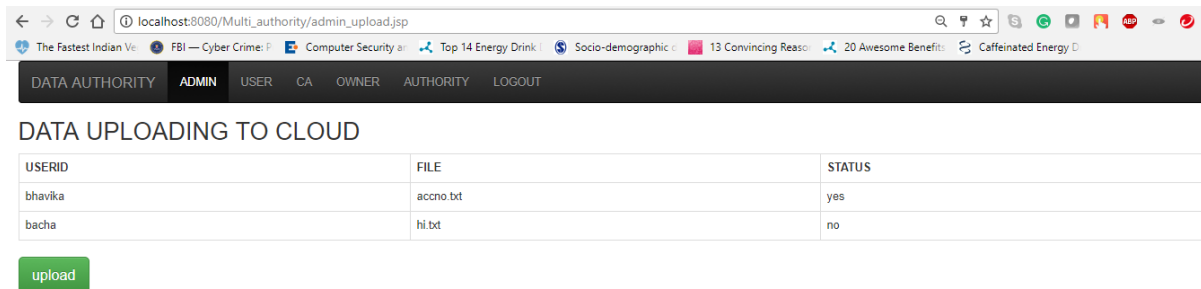


Figure 20. Files Admin will be uploading to the cloud

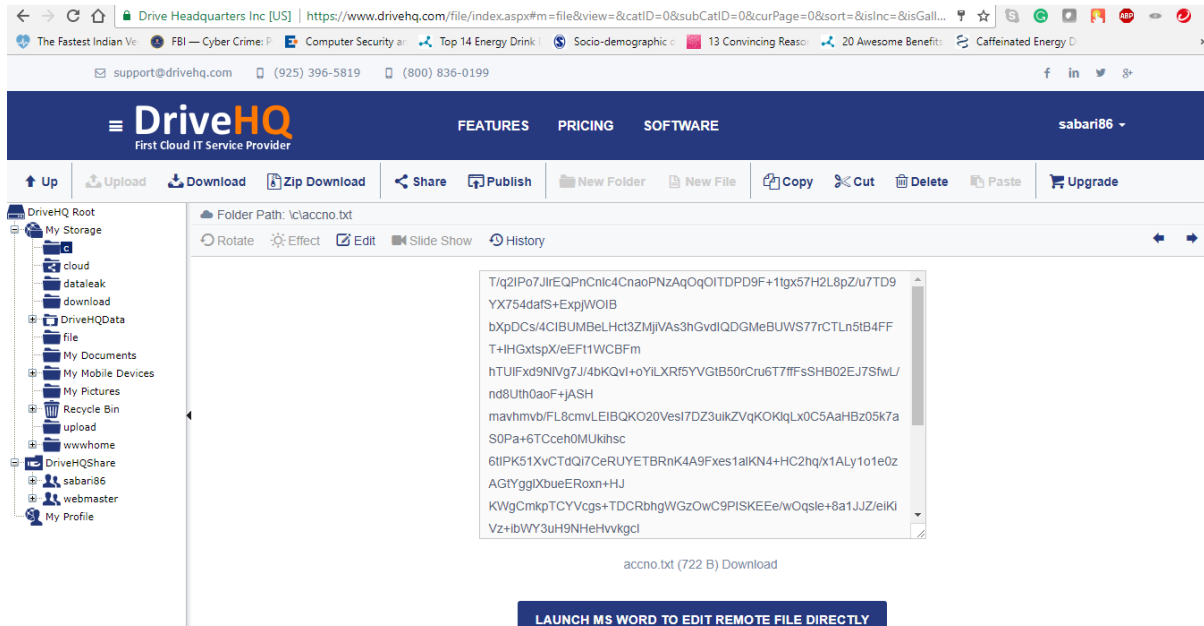


Figure 21. Files will be uploaded into the Cloud in an Encrypted format

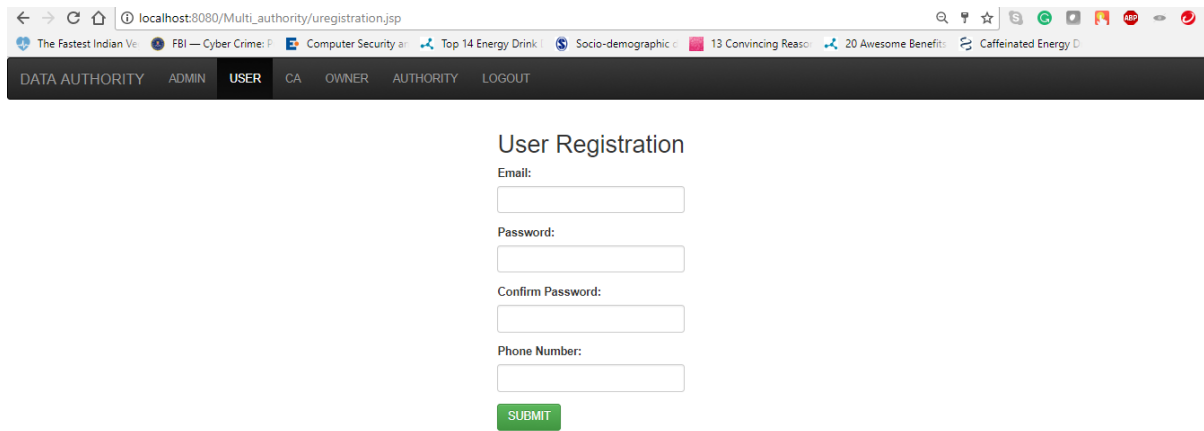


Figure 22. User Registration Page

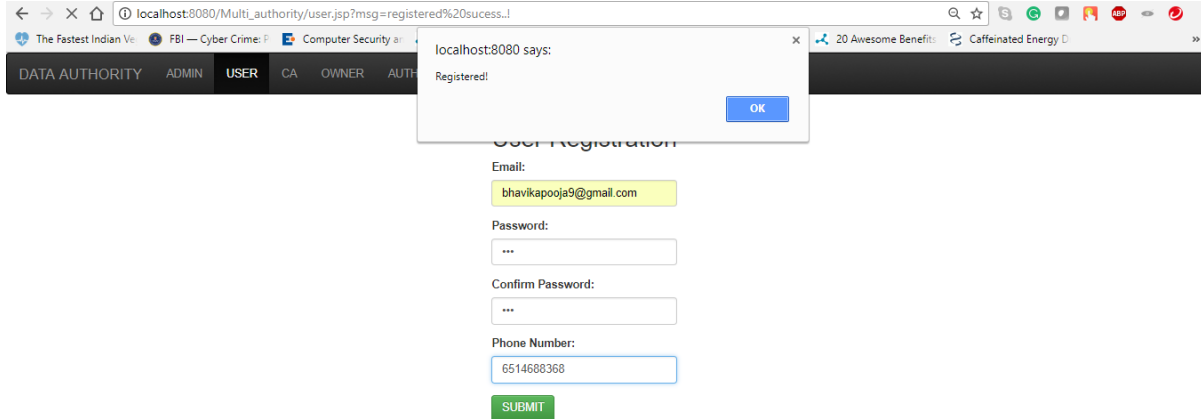


Figure 23. User Registration Page with a confirmation that he has been Registered

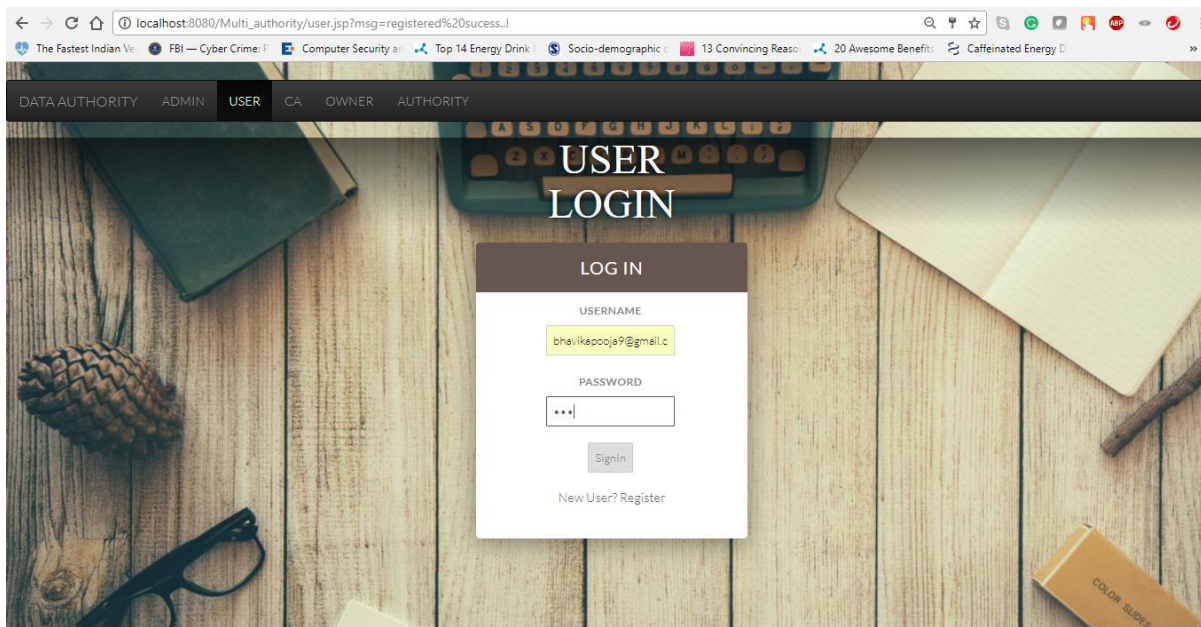


Figure 24. User Login Page

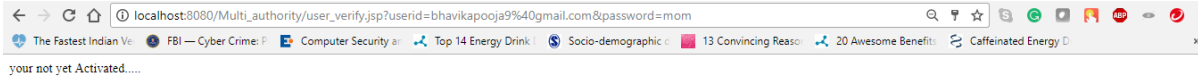


Figure 25. User has not been activated yet and the CA must activate his account

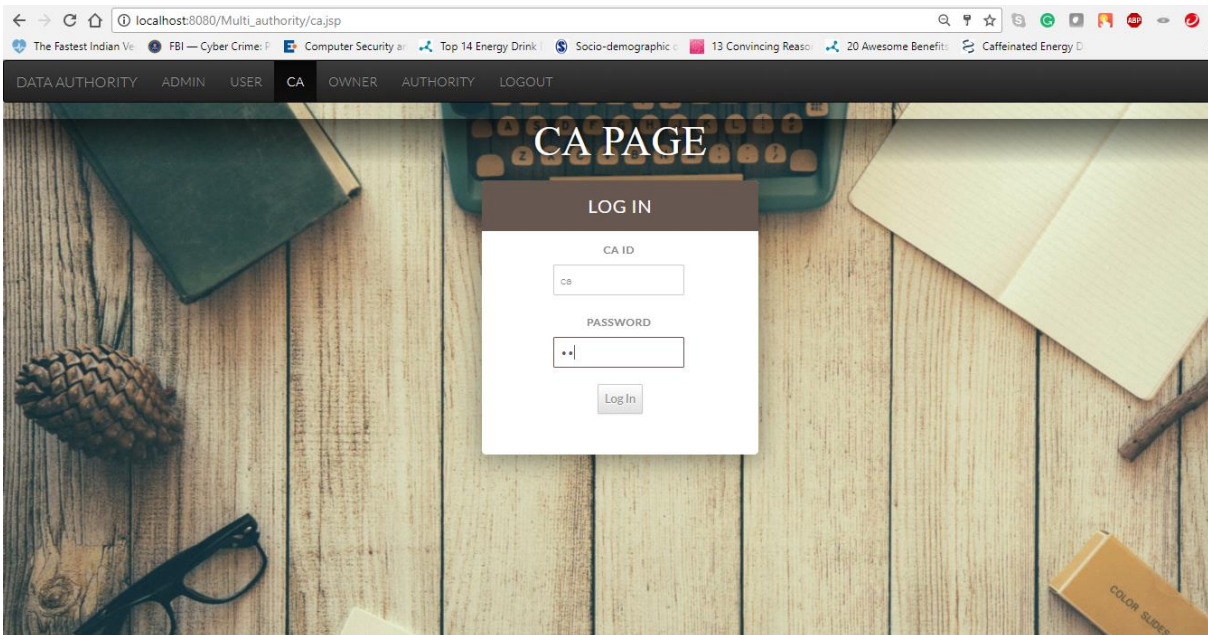


Figure 26. CA Login Page

USERID	PASSWORD	PHONENO	STATUS	ACTIVATE
nadanapathy.bluish@gmail.com	****	123647489	yes	Activate
nadanapathy.bluish@gmail.com	****	21321331187	no	Activate
bhavikapooja9@gmail.com	****	6514688368		Activate

Figure 27. CA Activating the Users and the status is still NO as its not activated yet.

USERID	PASSWORD	PHONENO	STATUS	ACTIVATE
nadanapathy.bluish@gmail.com	****	123647489	yes	Activate
nadanapathy.bluish@gmail.com	****	21321331187	no	Activate
bhavikapooja9@gmail.com	****	6514688368	yes	Activate

Figure 28. Status being changed to 'YES' after CA has activated the account

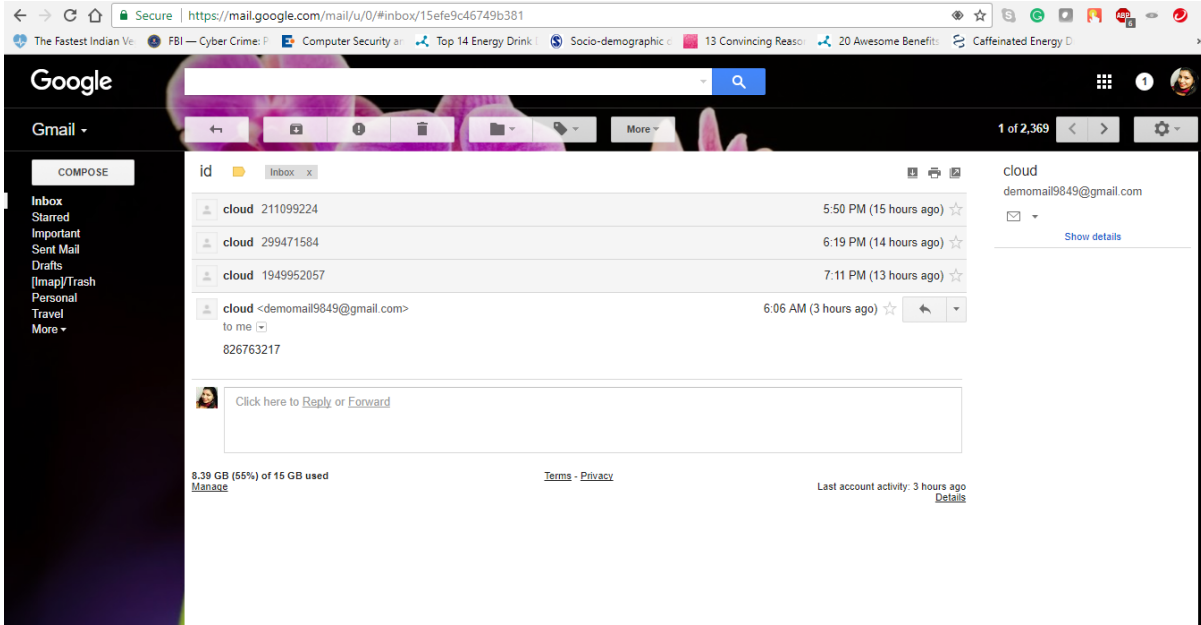


Figure 29. The product_id is being sent to the user's email

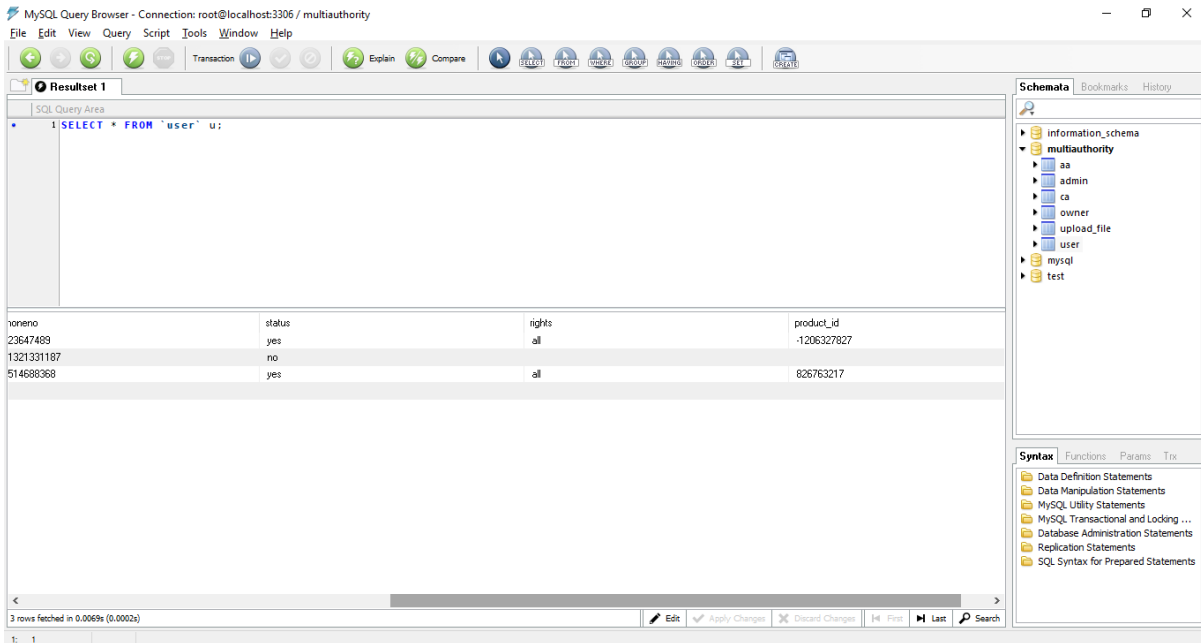


Figure 30. Database View

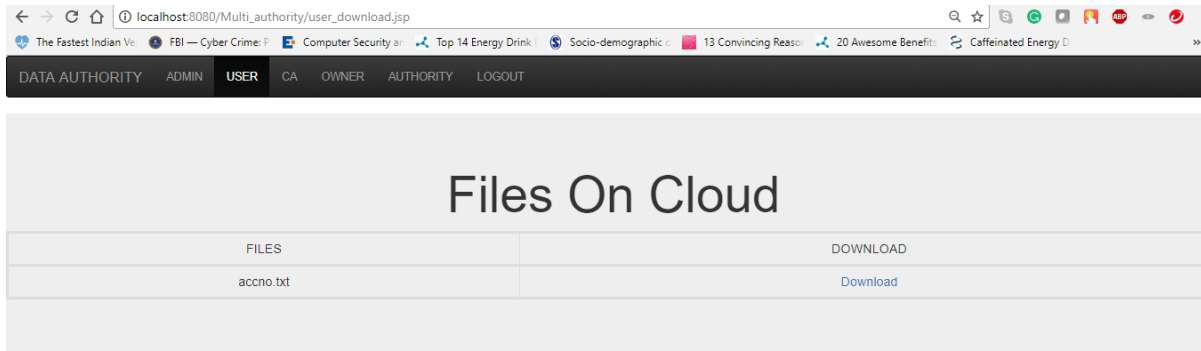
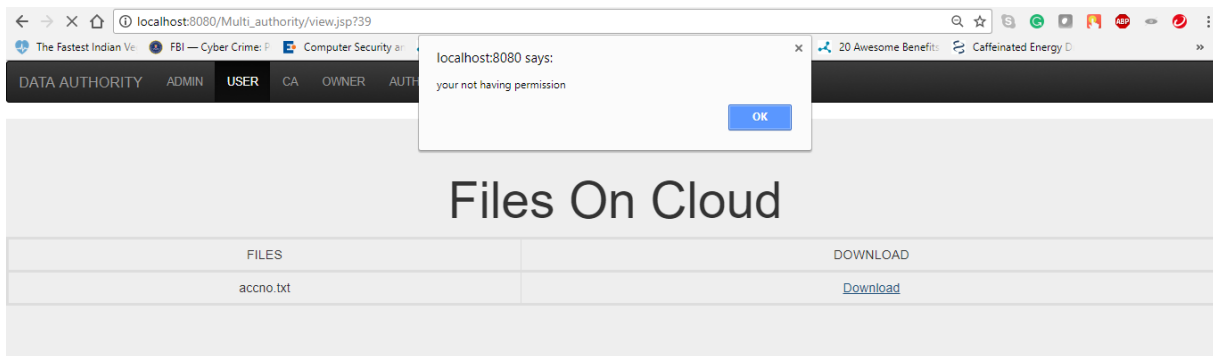


Figure 31. Files user can download from the Cloud



Waiting for localhost...

Figure 32. User do not have permission to download the file and needs to have the User Rights before downloading the file

DATA AUTHORITY ADMIN USER CA OWNER AUTHORITY

AA Registration

User Name:

Email:

Password:

Confirm Password:

Phone Number:

Figure 33. Authority Registration Page

localhost:8080 says:
Registered!

DATA AUTHORITY ADMIN USER CA OWNER AUTHORITY

AA Registration

User Name:

Email:

Password:

Confirm Password:

Phone Number:

Figure 34. Confirmation that the Authority has been Registered

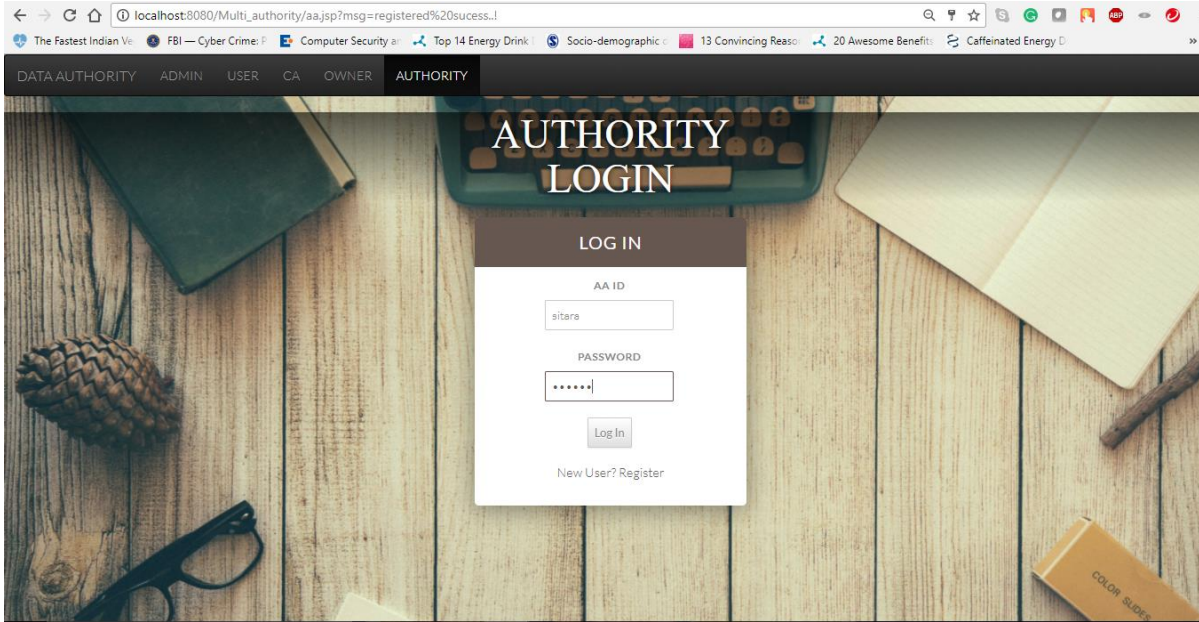


Figure 35. The Authority Login Page

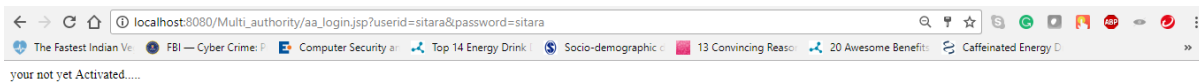


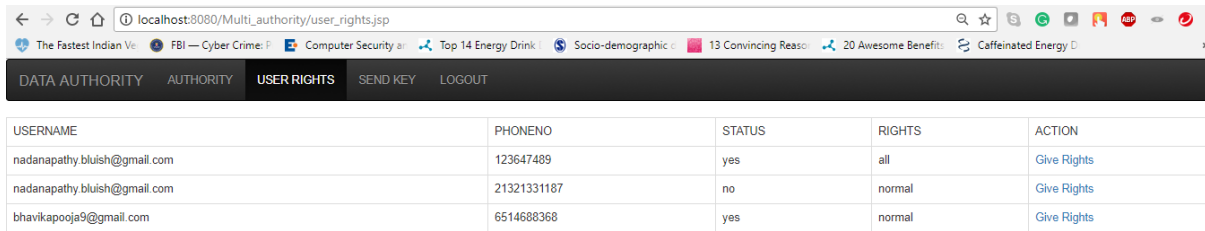
Figure36. The Authority is not activated yet and must be activated by the CA

USERID	PASSWORD	MAILID	PHONENO	STATUS	ACTIVATE
aa1	****	nadanapathy.bluish@gmail.com	312313123	no	Activate
santhuaa	****	bhavikapooja9@gmail.com	6513179471	yes	Activate
sitara	****	bhavikapooja9@gmail.com	6514688365	no	Activate

Figure 37. The CA activation Page for the Authority

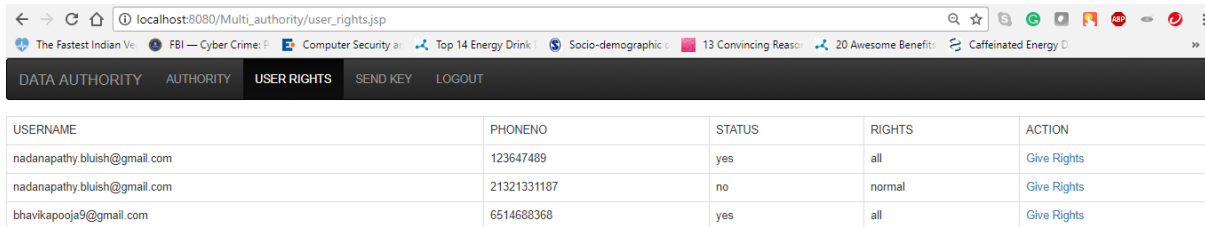
USERID	PASSWORD	MAILID	PHONENO	STATUS	ACTIVATE
aa1	****	nadanapathy.bluish@gmail.com	312313123	no	Activate
santhuaa	****	bhavikapooja9@gmail.com	6513179471	yes	Activate
sitara	****	bhavikapooja9@gmail.com	6514688365	yes	Activate

Figure 38. The status is changed to 'Yes' since the Authority is activated



USERNAME	PHONENO	STATUS	RIGHTS	ACTION
nadanapathy.bluish@gmail.com	123647489	yes	all	Give Rights
nadanapathy.bluish@gmail.com	21321331187	no	normal	Give Rights
bhavikapooja9@gmail.com	6514688368	yes	normal	Give Rights

Figure 39. The Authority Logins and provides the User Rights to the User



USERNAME	PHONENO	STATUS	RIGHTS	ACTION
nadanapathy.bluish@gmail.com	123647489	yes	all	Give Rights
nadanapathy.bluish@gmail.com	21321331187	no	normal	Give Rights
bhavikapooja9@gmail.com	6514688368	yes	all	Give Rights

Figure 40. The Rights has been changed to 'All' by the Authority

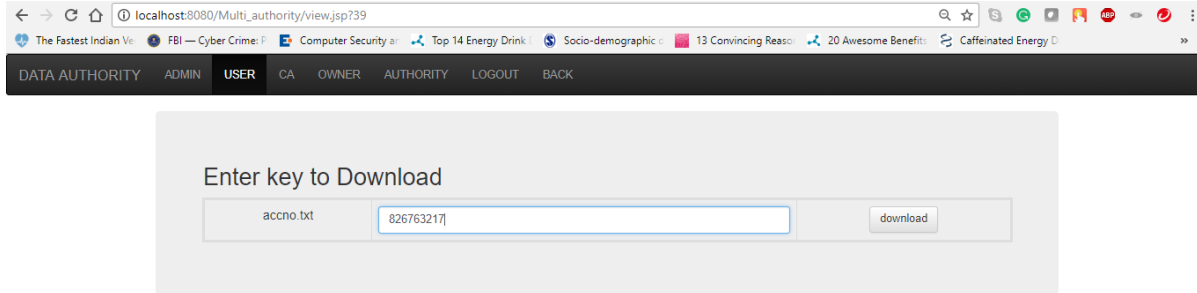


Figure 41. The user has to enter a product id to download the file

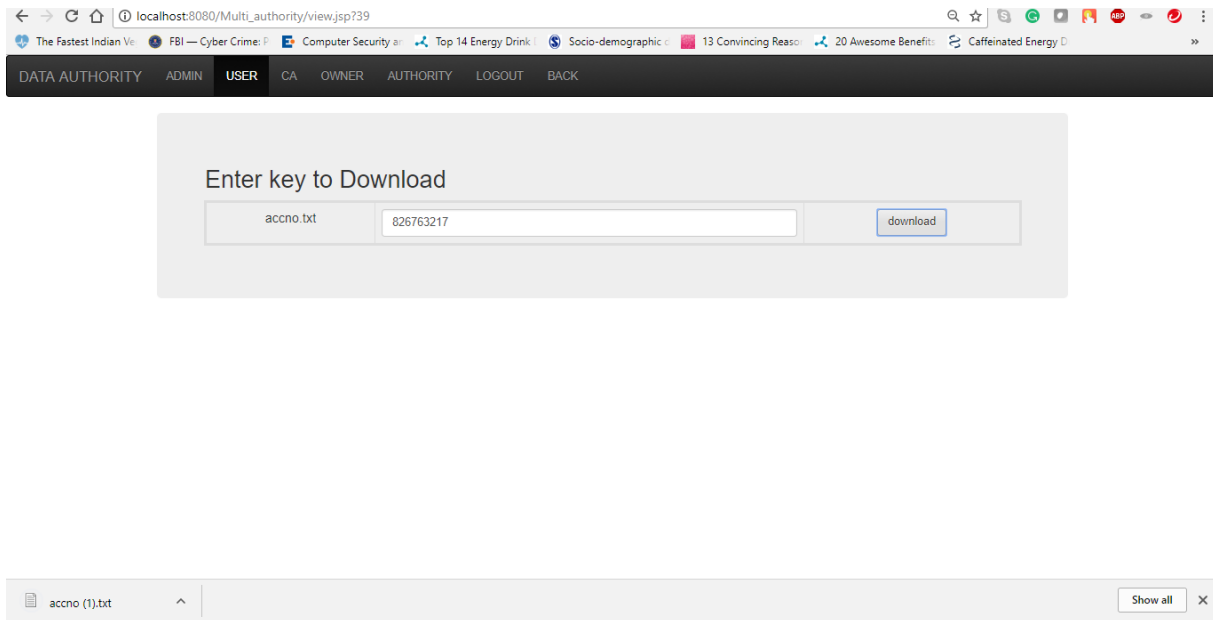
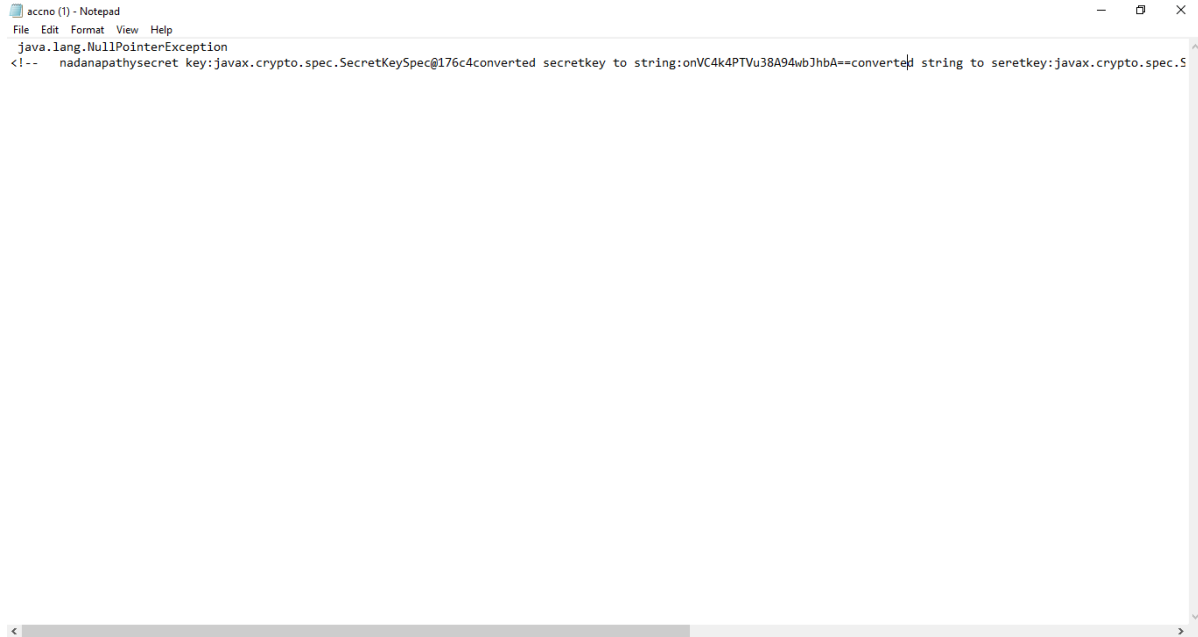


Figure 42. The file has been downloaded after entering the product id

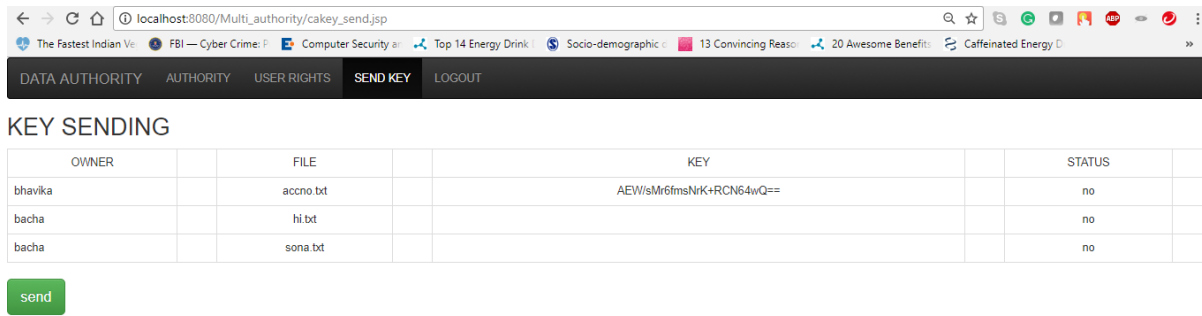


```

accno (1) - Notepad
File Edit Format View Help
java.lang.NullPointerException
<!-- nadanapathsecret key:javax.crypto.spec.SecretKeySpec@176c4converted secretkey to string:onVC4k4PTVu38A94wbJhbA==converted string to seretkey:javax.crypto.spec.S

```

Figure 43. File is Encrypted



localhost:8080/Multi_authority/akey_send.jsp

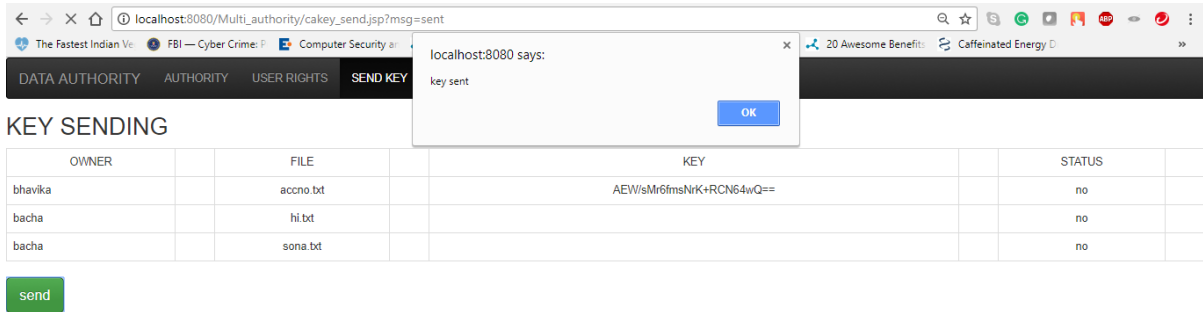
DATA AUTHORITY AUTHORITY USER RIGHTS **SEND KEY** LOGOUT

KEY SENDING

OWNER	FILE	KEY	STATUS
bhavika	accno.txt	AEW/sMr6fmsNrk+RCN64wQ==	no
bacha	hi.txt		no
bacha	sona.txt		no

send

Figure 44. The Authority sends the secret key to decrypt the file for the user



Waiting for extension Trend Micro Toolbar...

Figure 45. The key has been sent to user's email

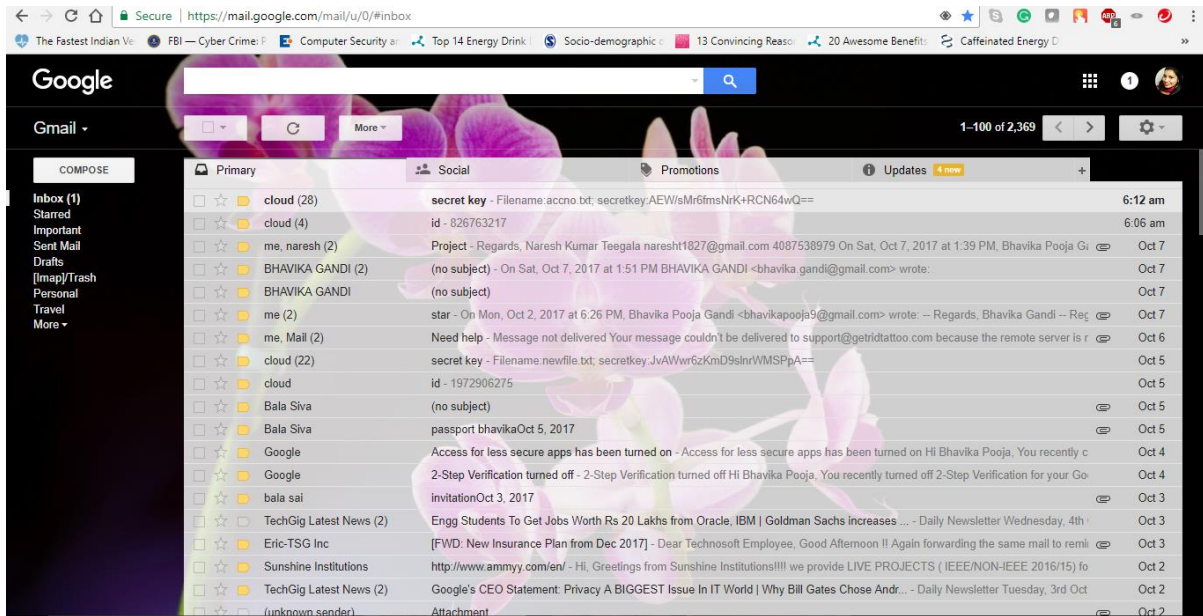


Figure 46. The Key is sent to the user's email

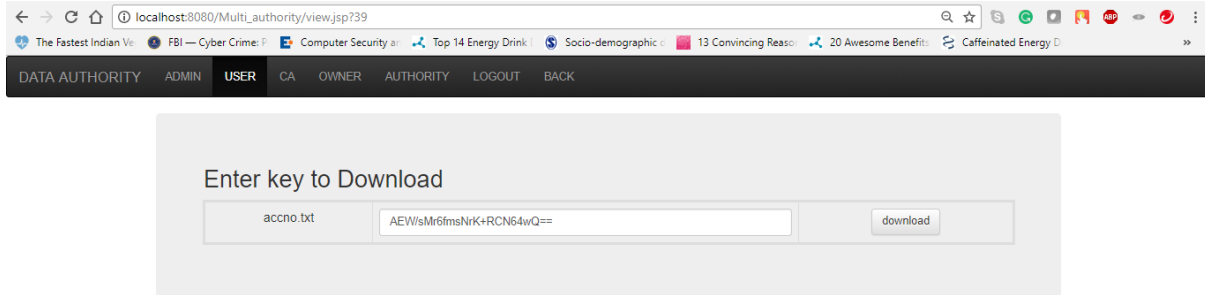
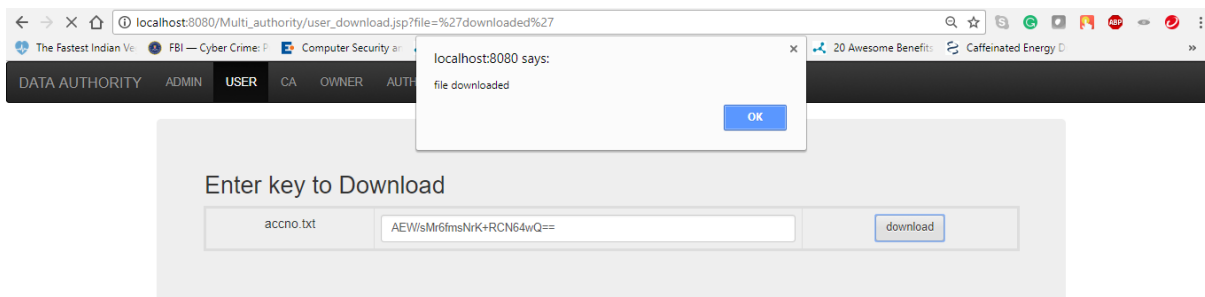


Figure 47. The key is being entered to download the file



Waiting for localhost...

Figure 48. The file is being downloaded successfully


```

rHyoN1iRsVXV4nD0JutlnGaslCJuC7uwjduW9SVrLvRYooPp2bWYgmgJQIXwl/Sp"
crossorigin="anonymous">
<script
  src="https://code.jquery.com/jquery-2.2.4.min.js"
  integrity="sha256-BbhdLvQf/xTY9gja0Dq3HiwQF8LaCRTXxZKRutelT44="
  crossorigin="anonymous"></script>
<!-- Latest compiled and minified JavaScript -->
<script      src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"
integrity="sha384-
Tc5IQib027qvyjSMfHjOMaLkfuWVxZxUPnCJA712mCWNlIpG9mGCD8wGNlcpD7Txa"
crossorigin="anonymous"></script>
<script>
  function validate (){
    var aid=document.name. adminid.value;
    var pass=document.name. password.value;
    if(aid==0) {
      alert ("Enter Adminid");
      document.name.adminid.focus();
      return false;
    }
    if(pass==0){
      alert("Enter password");
      document.name.password.focus();
      return false;
    }
  }
</script>
<style>
  body {
    background: url("./images/photo_bg.jpg") no-repeat center center fixed;
    background-size: cover;
    font-size: 16px;
    font-family: 'Lato', sans-serif;
    font-weight: 300;
    margin: 0;
    color: #666;
  }

  /* Typography */
  h1#title {
    font-family: 'Roboto Slab', serif;
    font-weight: 300;
    font-size: 3.2em;

```

```

    color: white;
    text-shadow: 0 0 10px rgba(0,0,0,0.8);
    margin: 0 auto;
    max-width: 300px;
    text-align: center;
    position: relative;
    top: 0px;
}
h1#title span span {
    font-weight: 400;}
h2 {
    text-transform: uppercase;
    color: white;
    font-weight: 400;
    letter-spacing: 1px;
    font-size: 1.4em;
    line-height: 2.8em;
}
a {
    text-decoration: none;
    color: #666;
}
a:hover {
    color: #aeaeae;
}
p.small {
    font-size: 0.8em;
    margin: 20px 0 0;
}/* Layout */
.container {
    margin: 0;
}
top {
    margin: 0;
    padding: 0;
    width: 100%;
    background: -moz-linear-gradient(top, rgba(0,0,0,0.6) 0%, rgba(0,0,0,0) 100%); /*
FF3.6-15 */
    background: -webkit-linear-gradient(top, rgba(0,0,0,0.6) 0%,rgba(0,0,0,0) 100%); /*
Chrome10-25,Safari5.1-6 */
    background: linear-gradient(to bottom, rgba(0,0,0,0.6) 0%,rgba(0,0,0,0) 100%); /*
W3C, IE10+, FF16+, Chrome26+, Opera12+, Safari7+ */

```

```
        filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#99000000',
endColorstr='#00000000',GradientType=0 ); /* IE6-9 */
    }
    .login-box {
        background-color: white;
        max-width: 340px;
        margin: 0 auto;
        position: relative;
        padding-bottom: 30px;
        border-radius: 5px;
        box-shadow: 0 5px 50px rgba(0,0,0,0.4);
        text-align: center;
    }
    .login-box .box-header {
        background-color: #665851;
        margin-top: 0;
        border-radius: 5px 5px 0 0;
    }

    .login-box label {
        font-weight: 700;
        font-size: .8em;
        color: #888;
        letter-spacing: 1px;
        text-transform: uppercase;
        line-height: 2em;
    }
    .login-box input {
        margin-bottom: 20px;
        padding: 8px;
        border: 1px solid #ccc;
        border-radius: 2px;
        font-size: .9em;
        color: #888;
    }
    .login-box input:focus {
        outline: none;
        border-color: #665851;
        transition: 0.5s;
        color: #665851;
    }
    .login-box button {
        margin-top: 0px;
```

```

border: 0;
border-radius: 2px;
color: white;
padding: 10px;
text-transform: uppercase;
font-weight: 400;
font-size: 0.7em;
letter-spacing: 1px;
background-color: #665851;
cursor:pointer;
outline: none;
}
.login-box button:hover {
  opacity: 0.7;
  transition: 0.5s;
}
}
.login-box button:hover {
  opacity: 0.7;
  transition: 0.5s;
}
}
.selected {
  color: #665851!important;
  transition: 0.5s;
}
}
/* Animation Delay */
#logo {
  -webkit-animation-duration: 1s;
  -webkit-animation-delay: 2s;
}
}
.login-box {
  -webkit-animation-duration: 1s;
  -webkit-animation-delay: 1s;
}
}
</style>
</head>
<body>
  <%
    if(request.getParameter("msg")!=null){
      out.println("<script>alert('incorrect password')</script>");
    }
    if(request.getParameter("msgg")!=null){
      out.println("<script>alert('username not exist')</script>");
    }
  <%

```

```

    %>
<div>
<nav class="navbar navbar-inverse">
  <div class="container-fluid">
    <!-- Brand and toggle get grouped for better mobile display -->
    <div class="navbar-header">
      <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-
target="#bs-example-navbar-collapse-1" aria-expanded="false">
        <span class="sr-only">Toggle navigation</span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </button>
      <a class="navbar-brand" href="#">DATA AUTHORITY</a>
    </div>
    <!-- Collect the nav links, forms, and other content for toggling -->
    <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
      <ul class="nav navbar-nav">
        <li class="active"><a href="admin.jsp">ADMIN <span class="sr-
only">(current)</span></a></li>
        <li><a href="user.jsp">USER</a></li>
        <li><a href="ca.jsp">CA</a></li>
        <li><a href="owner.jsp">OWNER</a></li>
        <li><a href="aa.jsp">AUTHORITY</a></li>
      </ul>
    </div><!-- /.navbar-collapse -->
  </div><!-- /.container-fluid -->
</nav>
</div>
<div class="top">
  <h1 id="title"><span id="logo">Multi Authority <span>Cloud Storage
</span></span></h1>
</div>
<div class="login-box animated fadeInUp">
  <div class="box-header">
    <h2>Log In</h2>
  </div>
  <form action="admin_verify.jsp" method="get" name="name" onsubmit="return
validate()">
    <label for="username">Username</label>
    <br/>
    <input type="text" name="adminid" id="username">
    <br/>

```



```

<label for="password">Password</label>
<br/>
<input type="password" name="password" id="password">
<br/>
<input type="submit" value="SignIn"></input>
<br/>
<a href="#"><p class="small">Forgot your password? </p></a>
</form>
</div>
<div>
<!-- <div style="position: absolute;left:210px;top: 150px">
  <h2>ADMIN LOGIN PAGE</h2>
  <form action="admin_verify.jsp" method="get" name="name" onsubmit="return
  validate()">
    <strong style="background-color: burlywood"> ADMIN ID:</strong><br>
    <input type="text" name="adminid" placeholder="enter admin id"></input><br>
    <strong style="background-color: burlywood">PASSWORD:</strong><br>
    <input type="password" name="password" placeholder="enter
  password"></input><br></br>
    <input type="submit" value="SignIn"></input>
  </form>
</div>-->
</div>
</body>
</html>

```

Admin Upload:

```

<% @page import="java.sql.ResultSet"%>
<% @page import="java.sql.Statement"%>
<% @page import="java.sql.DriverManager"%>
<% @page import="java.sql.Connection"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Multi Authority Cloud Storage</title>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<!--<link href="style.css" rel="stylesheet" type="text/css" />-->
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
integrity="sha384-
BVYiSiFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u"
crossorigin="anonymous">

```

```

<! -- Optional theme -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap-
theme.min.css" integrity="sha384-
rHyoN1iRsVXV4nD0JutlnGaslCJuC7uwjduW9SVrLvRYooPp2bWYgmgJQIXwl/Sp"
crossorigin="anonymous">
<script src="https://code.jquery.com/jquery-2.2.4.min.js" integrity="sha256-
BbhdLvQf/xTY9gja0Dq3HiwQF8LaCRTXxZKRutelT44="
crossorigin="anonymous"></script>
<! -- Latest compiled and minified JavaScript -->
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"
integrity="sha384-
Tc5IQib027qvyjSMfHjOMaLkfuWVxZxUPnCJA712mCWNIPg9mGCD8wGNiCpD7Txa"
crossorigin="anonymous"></script>
<script type="text/javascript">
    window.history.forward();
    function noBack()
    {
        window.history.forward();
    }
</script>
</head>
<body onLoad="noBack();" onpageshow="if (event.persisted) noBack();" onUnload="">
    <%
        String user,file,status;
        Class.forName("com.mysql.jdbc.Driver");
        Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/multiauthority","root","root
");

        try{
            if(request.getParameter("msg")!=null){
                out.println("<script>alert('File uploaded')</script>");
            }
            if(request.getParameter("msgg")!=null){
                out.println("<script>alert('failed')</script>");
            }
            if(request.getParameter("ms")!=null){
                out.println("<script>alert('File not found to send')</script>");
            }
        }
//        Class.forName("org.sqlite.JDBC");
//
//
//        Connection
con=DriverManager.getConnection("jdbc:sqlite:/home/ibn/Desktop/Nadanapathy/NetBeansPr
jects/multi-authority_cloud_storage/multiauthority");

```

```
//
Statement st=con.createStatement();
ResultSet rt=st.executeQuery("select * from upload_file");
%>
<div class="container-fluid">
  <nav class="navbar navbar-inverse">
    <div class="container-fluid">
      <!-- Brand and toggle get grouped for better mobile display -->
      <div class="navbar-header">
        <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-
target="#bs-example-navbar-collapse-1" aria-expanded="false">
          <span class="sr-only">Toggle navigation</span>
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
        </button>
        <a class="navbar-brand" href="#">DATA AUTHORITY</a>
      </div><!-- Collect the nav links, forms, and other content for toggling -->
      <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
        <ul class="nav navbar-nav">
          <li class="active"><a href="admin.jsp">ADMIN <span class="sr-
only">(current)</span></a></li>
          <li><a href="user.jsp">USER</a></li>
          <li><a href="ca.jsp">CA</a></li>
          <li><a href="owner.jsp">OWNER</a></li>
          <li><a href="aa.jsp">AUTHORITY</a></li>
          <li><a href="admin.jsp">LOGOUT</a></li>
        </ul>
      </div><!-- /.navbar-collapse -->
    </div><!-- /.container-fluid -->
  </nav>
  <div class="header">
    <div class="header_resize">
      <div>
        <h2>DATA UPLOADING TO CLOUD</h2>
        <form action="upload.jsp" method="post" name="name">
          <table class="table table-bordered">
            <!-- <tr> <td> CHOOSE FILE </td> <td> <input type="file" name="file"></td></tr>
          <tr style="height: 15px"></tr>
        -->
      </div>
    </div>
  </div>
  <thead>
    <tr>
      <th>USERID</th>

```

```

    <th >FILE</th>
    <th >STATUS</th>
<!-- <td align="center">UPLOAD</td><td></td>-->
</tr>
</thead>
<% while(rt.next()){
    user=rt.getString("userid");
    file=rt.getString("filename");
    status=rt.getString("status");
    %>
<tr>
    <td><%=user%></td>
    <td><%=file%></td>
    <td><%=status%></td>
</tr>
<%
}
    con.close();
    }
    catch(Exception e){
        out.println(e);
    }
    %>
    </table>
    <input type="submit" value="upload" class="btn btn-lg btn-success"></input>
</form>
</div>
</div>
</div>
</body>
</html>

```

Owner Registraton

```

<!DOCTYPE html>
<html>
<head>
<title>Multi Authority Cloud Storage</title>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<script type="text/javascript" src="js/cufon-yui.js"></script>
<script type="text/javascript" src="js/arial.js"></script>
<script type="text/javascript" src="js/cuf_run.js"></script>

```

```

<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
integrity="sha384-
BVYiSiFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u"
crossorigin="anonymous">
<!-- Optional theme -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap-
theme.min.css" integrity="sha384-
rHyoN1iRsVXV4nD0JutlnGaslCJuC7uwjduW9SVrLvRYooPp2bWYgmgJQIXwl/Sp"
crossorigin="anonymous">
<script src="https://code.jquery.com/jquery-2.2.4.min.js" integrity="sha256-
BbhdLvQf/xTY9gja0Dq3HiwQF8LaCRTXxZKRutelT44="
crossorigin="anonymous"></script>
<!-- Latest compiled and minified JavaScript -->
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"
integrity="sha384-
Tc5IQib027qvyjSMfHjOMaLkfuWVxZxUPnCJA712mCWNlPpG9mGCD8wGNlCPD7Txa"
crossorigin="anonymous"></script>
<script>
function validate(){
    var uname=document.name.uname.value;
    var mail=document.name.mail.value;
    var pass=document.name.password.value;
    var cpass=document.name.cpassword.value;
    var ph=document.name.phoneno.value;
    if(uname==0){
        alert("Enter username");
        document.name.uname.focus();
        return false;
    }
    if(mail==0){
        alert("Enter your Mailid");
        document.name.mail.focus();
        return false;
    }
    if(pass==0){
        alert("Enter your password");
        document.name.password.focus()
        return false;
    }
    if(cpass==0){
        alert("Enter your password cofirmation");
        document.name.cpassword.focus();
    }
}

```

```

    return false;
}
if(pass!=cpass){
    alert("Incorrect Confiorm password");
    document.name.password.focus();
    return false;
}
if(ph==0){
    alert("Enter your phone number");
    document.name.phoneno.focus();
    return false;
}
if(isNaN(ph)){
    alert("Incorrect phone numbers");
    document.name.phoneno.focus();
    return false;
}
}
}
</script>
</head>
<body>
<div class="">
    <nav class="navbar navbar-inverse">
    <div class="container-fluid">
    <!-- Brand and toggle get grouped for better mobile display -->
    <div class="navbar-header">
        <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-
target="#bs-example-navbar-collapse-1" aria-expanded="false">
            <span class="sr-only">Toggle navigation</span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
        </button>
        <a class="navbar-brand" href="#">DATA AUTHORITY</a>
    </div><!-- Collect the nav links, forms, and other content for toggling -->
    <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
    <ul class="nav navbar-nav">
        <li><a href="admin.jsp">ADMIN <span class="sr-only">(current)</span></a></li>
        <li><a href="user.jsp">USER</a></li>
        <li><a href="ca.jsp">CA</a></li>
        <li class="active"><a href="owner.jsp">OWNER</a></li>
        <li><a href="aa.jsp">AUTHORITY</a></li>
        <li><a href="owner.jsp">LOGOUT</a></li>

```

```

    </ul>
  </div><!-- /.navbar-collapse -->
</div><!-- /.container-fluid -->
</nav>
<div class="">
  <div class="header_resize">
    <div style="position: absolute;left:40%;">
      <h2>Owner Registration</h2>
      <hr />
      <form action="ownerreg_db.jsp" method="get" name="name" onsubmit="return
validate()">
        <div class="form-group">
          <label for="uname">User Name:</label>
          <input type="text" name="uname" class="form-control" id="uname" />
        </div>
        <div class="form-group">
          <label for="email">Email:</label>
          <input type="email" name="mail" class="form-control" id="email" />
        </div>
        <div class="form-group">
          <label for="password">Password:</label>
          <input type="password" name="password" class="form-control" id="uname" />
        </div>
        <div class="form-group">
          <label for="cpassword">Confirm Password:</label>
          <input type="password" name="cpassword" class="form-control" id="cpassword" />
        </div>
        <div class="form-group">
          <label for="phonenumber">Phone Number:</label>
          <input type="text" name="phoneno" class="form-control" id="phoneno" />
        </div>
        <input type="submit" value="SUBMIT" class="btn btn-success"></input>
      </form>
    </div>
  </div>
</div>
</div>
</body>
</html>

```

Owner Upload File

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

```

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Multi Authority Cloud Storage</title>
<meta http-equiv="content-type" content="text/html;charset=utf-8" />
<script type="text/javascript" src="js/cufon-yui.js"></script>
<script type="text/javascript" src="js/arial.js"></script>
<script type="text/javascript" src="js/cuf_run.js"></script>
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
integrity="sha384-
BVYiSiFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u"
crossorigin="anonymous">
<!-- Optional theme -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap-
theme.min.css" integrity="sha384-
rHyoN1iRsVXV4nD0JutlnGaslCJuC7uwjduW9SVrLvRYooPp2bWYgmgJQIXwl/Sp"
crossorigin="anonymous">
<script src="https://code.jquery.com/jquery-2.2.4.min.js" integrity="sha256-
BbhdLvQf/xTY9gja0Dq3HiwQF8LaCRTXxZKRutelT44="
crossorigin="anonymous"></script>
<!-- Latest compiled and minified JavaScript -->
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"
integrity="sha384-
Tc5IQib027qvyjSMfHjOMaLkfuWVxZxUPnCJA712mCWNIPg9mGCD8wGNicPD7Txa"
crossorigin="anonymous"></script>
<script type="text/javascript">
function validate(){
if($("#file").val() == "" || $("#file").val() == undefined ){
alert('choose file');
return false;
}
$(".banner").removeClass("hidden");
}

window.history.forward();
function noBack()
{
window.history.forward();
}

</script>
</head>
<body onLoad="noBack();" onpageshow="if (event.persisted) noBack();" onUnload="">

```



```

    <%    }
    if(request.getParameter("error")!=null){
        out.println("<script>alert('error when uploading')</script>");
    }
    %>
<div class="">
    <nav class="navbar navbar-inverse">
    <div class="container-fluid">
    <!-- Brand and toggle get grouped for better mobile display -->
    <div class="navbar-header">
        <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-
target="#bs-example-navbar-collapse-1" aria-expanded="false">
            <span class="sr-only">Toggle navigation</span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
        </button>
        <a class="navbar-brand" href="#">DATA AUTHORITY</a>
    </div> <!-- Collect the nav links, forms, and other content for toggling -->
    <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
    <ul class="nav navbar-nav">
        <li><a href="admin.jsp">ADMIN <span class="sr-only">(current)</span></a></li>
        <li><a href="user.jsp">USER</a></li>
        <li><a href="ca.jsp">CA</a></li>
        <li class="active"><a href="owner.jsp">OWNER</a></li>
        <li><a href="aa.jsp">AUTHORITY</a></li>
        <li><a href="owner.jsp">LOGOUT</a></li>
    </ul>
    </div><!-- /.navbar-collapse -->
    </div><!-- /.container-fluid -->
</nav>
<div class="conatiner">
    <div class="jumbotron" style="text-align: center">
    <div class="alert alert-success hidden banner">
        <strong>Success!</strong> File has been uploaded successfully.
    </div>
    <h1>File Upload</h1>
    <form      action="upload"      method="post"      enctype="multipart/form-data"
onsubmit="return validate()">
        <input type="file" name="file" id="file" style="display: inline"></input>
        <button class="btn btn-default"><a href="user_download.jsp">Download
File</a></button>

```

```

        <input type="submit" name="submit" value="Upload" class="btn btn-
default"></input>
    </form>
</div>
</div>
</div>
</body>
</html>

```

Aes Encryption

```

<% @page import="javax.swing.JOptionPane"%>
<% @page import="java.io.PrintWriter"%>
<% @page import="sun.misc.BASE64Encoder"%>
<% @page import="javax.crypto.Cipher"%>
<% @page import="javax.crypto.SecretKey"%>
<% @page import="javax.crypto.KeyGenerator"%>
<% void encrypt(String txt)
{
    try
    {
        String plainData=txt,cipherText,decryptedText;
        KeyGenerator keyGen = KeyGenerator.getInstance("AES");
        keyGen.init(256);
        SecretKey secretKey = keyGen.generateKey();
        System.out.println("secret key:"+secretKey);
        Cipher aesCipher = Cipher.getInstance("AES");
        aesCipher.init(Cipher.ENCRYPT_MODE,secretKey);
        byte[] byteDataToEncrypt = plainData.getBytes();
        byte[] byteCipherText = aesCipher.doFinal(byteDataToEncrypt);
        cipherText = new BASE64Encoder().encode(byteCipherText);
        aesCipher.init(Cipher.DECRYPT_MODE,secretKey,aesCipher.getParameters())
        byte[] byteDecryptedText = aesCipher.doFinal(byteCipherText);
        decryptedText = new String(byteDecryptedText);
        System.out.println("\n Given text : "+plainData+" \n Cipher Data : "+cipherText+" \n
Decrypted Data : "+decryptedText);

        FileWriter fw=new FileWriter("encrypted1.txt");
        fw.write(cipherText);

        FileWriter fw1=new FileWriter("decrypted1.txt");
        fw1.write(decryptedText);
        JOptionPane.showMessageDialog(null, "process finished");
        fw1.close();
        fw.close();
    }
}

```

```

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

%>
User
<!DOCTYPE html>
<html>
<head>
<title>Multi Authority Cloud Storage</title>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<script type="text/javascript" src="js/cufon-yui.js"></script>
<script type="text/javascript" src="js/arial.js"></script>
<script type="text/javascript" src="js/cuf_run.js"></script>
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
integrity="sha384-
BVYiSiFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u"
crossorigin="anonymous">
<!-- Optional theme -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap-
theme.min.css" integrity="sha384-
rHyOn1iRsVXV4nD0JutlnGaslCJuC7uwjduW9SVrLvRYooPp2bWYgmgJQIXwl/Sp"
crossorigin="anonymous">
<script src="https://code.jquery.com/jquery-2.2.4.min.js" integrity="sha256-
BbhdLvQf/xTY9gja0Dq3HiwQF8LaCRTXxZKRutelT44="
crossorigin="anonymous"></script>
<!-- Latest compiled and minified JavaScript -->
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"
integrity="sha384-
Tc5IQib027qvyjSMfHjOMaLkfuWVxZxUPnCJA712mCWNlP9mGCD8wGNlCpD7Txa"
crossorigin="anonymous"></script>
<script>
function validate(){
    var uname=document.name.userid.value;
    var pass=document.name.password.value;
    if(uname==0){
        alert("Enter your Userid");
        document.name.userid.focus();
        return false;
    }
}

```

```
    }
    if(pass==0){
        alert("Enter your password");
        document.name.password.focus();
        return false;
    }
}
</script>
<style>
    body {
        background: url("./images/photo_bg.jpg") no-repeat center center fixed;
        background-size: cover;
        font-size: 16px;
        font-family: 'Lato', sans-serif;
        font-weight: 300;
        margin: 0;
        color: #666;
    }

    /* Typography */
    h1#title {
        font-family: 'Roboto Slab', serif;
        font-weight: 300;
        font-size: 3.2em;
        color: white;
        text-shadow: 0 0 10px rgba(0,0,0,0.8);
        margin: 0 auto;
        max-width: 300px;
        text-align: center;
        position: relative;
        top: 0px;
    }h1#title span span {
        font-weight: 400;
    }h2 {
        text-transform: uppercase;
        color: white;
        font-weight: 400;
        letter-spacing: 1px;
        font-size: 1.4em;
        line-height: 2.8em;
    }a {
        text-decoration: none;
        color: #666;
```

```

}
a:hover {
    color: #aeaeae;
}p.small {
    font-size: 0.8em;
    margin: 20px 0 0;
}
/* Layout */
.container {
    margin: 0;
}
top {
    margin: 0;
    padding: 0;
    width: 100%;
    background: -moz-linear-gradient(top, rgba(0,0,0,0.6) 0%, rgba(0,0,0,0) 100%); /*
FF3.6-15 */
    background: -webkit-linear-gradient(top, rgba(0,0,0,0.6) 0%,rgba(0,0,0,0) 100%); /*
Chrome10-25,Safari5.1-6 */
    background: linear-gradient(to bottom, rgba(0,0,0,0.6) 0%,rgba(0,0,0,0) 100%); /*
W3C, IE10+, FF16+, Chrome26+, Opera12+, Safari7+ */
    filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#99000000',
endColorstr='#00000000',GradientType=0 ); /* IE6-9 */
}.login-box {
    background-color: white;
    max-width: 340px;
    margin: 0 auto;
    position: relative;
    padding-bottom: 30px;
    border-radius: 5px;
    box-shadow: 0 5px 50px rgba(0,0,0,0.4);
    text-align: center;
}
.login-box .box-header {
    background-color: #665851;
    margin-top: 0;
    border-radius: 5px 5px 0 0;
}
.login-box label {
    font-weight: 700;
    font-size: .8em;
    color: #888;
    letter-spacing: 1px;

```

```
        text-transform: uppercase;
        line-height: 2em;
    }
    .login-box input {
        margin-bottom: 20px;
        padding: 8px;
        border: 1px solid #ccc;
        border-radius: 2px;
        font-size: .9em;
        color: #888;
    }
    .login-box input:focus {
        outline: none;
        border-color: #665851;
        transition: 0.5s;
        color: #665851;
    }
    .login-box button {
        margin-top: 0px;
        border: 0;
        border-radius: 2px;
        color: white;
        padding: 10px;
        text-transform: uppercase;
        font-weight: 400;
        font-size: 0.7em;
        letter-spacing: 1px;
        background-color: #665851;
        cursor:pointer;
        outline: none;
    }
    .login-box button:hover {
        opacity: 0.7;
        transition: 0.5s;
    }
    .login-box button:active {
        opacity: 0.7;
        transition: 0.5s;
    }
    .selected {
        color: #665851!important;
        transition: 0.5s;
    }
}
```

```

/* Animation Delay */
#logo {
  -webkit-animation-duration: 1s;
  -webkit-animation-delay: 2s;
}
.login-box {
  -webkit-animation-duration: 1s;
  -webkit-animation-delay: 1s;
}
</style>
</head>
<body>
<%
if(request.getParameter("msg")!=null){
  out.println("<script>alert('Registered!')</script>");
}
if(request.getParameter("msgg")!=null){
  out.println("<script>alert('user not exist')</script>");
}
%>s
<div>
  <nav class="navbar navbar-inverse">
    <div class="container-fluid">
      <!-- Brand and toggle get grouped for better mobile display -->
      <div class="navbar-header">
        <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-
target="#bs-example-navbar-collapse-1" aria-expanded="false">
          <span class="sr-only">Toggle navigation</span>
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
        </button>
        <a class="navbar-brand" href="#">DATA AUTHORITY</a>
      </div>

      <!-- Collect the nav links, forms, and other content for toggling -->
      <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
        <ul class="nav navbar-nav">
          <li><a href="admin.jsp">ADMIN </a></li>
          <li class="active"><a href="user.jsp">USER <span class="sr-
only">(current)</span></a></li>
          <li><a href="ca.jsp">CA</a></li>

```

```

        <li><a href="owner.jsp">OWNER</a></li>
        <li><a href="aa.jsp">AUTHORITY</a></li>
    </ul>
</div><!-- /.navbar-collapse -->
</div><!-- /.container-fluid -->
</nav>
</div>
<div class="top">
<h1 id="title"><span id="logo">USER LOGIN</span></h1>
</div>
<div class="login-box animated fadeInUp">
<div class="box-header">
<h2>Log In</h2>
</div>
<form action="user_verify.jsp" method="get" name="name" onsubmit="return validate()">
<label for="username">Username</label>
<br/>
<input type="text" name="userid" id="username">
<br/>
<label for="password">Password</label>
<br/>
<input type="password" name="password" id="password">
<br/>
<input type="submit" value="SignIn"></input>
<br/>
<p>New User? <a href="uregistration.jsp">Register</a></p>
</form>
</div>
</body>
</html>

```

Authority

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Multi Authority Cloud Storage</title>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<script type="text/javascript" src="js/cufon-yui.js"></script>
<script type="text/javascript" src="js/arial.js"></script>
<script type="text/javascript" src="js/cuf_run.js"></script>
<link
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
rel="stylesheet"

```



```

integrity="sha384-
BVYiSiFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u"
crossorigin="anonymous">
<!-- Optional theme -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap-
theme.min.css" integrity="sha384-
rHyoN1iRsVXV4nD0JutlnGaslCJuC7uwjduW9SVrLvRYooPp2bWYgmgJQIXwl/Sp"
crossorigin="anonymous">
<script src="https://code.jquery.com/jquery-2.2.4.min.js" integrity="sha256-
BbhdlvQf/xTY9gja0Dq3HiwQF8LaCRTXxZKRutelT44="
crossorigin="anonymous"></script>
<!-- Latest compiled and minified JavaScript -->
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"
integrity="sha384-
Tc5IQib027qvyjSMfHjOMaLkfuWVxZxUPnCJA712mCWNIPg9mGCD8wGNiCpD7Txa"
crossorigin="anonymous"></script>
<script>
function validate(){
var uname=document.name.userid.value;
var pass=document.name.password.value;
if(uname==0){
alert("Enter your id");
document.name.userid.focus();
return false;
}
if(pass==0){
alert("Enter your password");
document.name.password.focus();
return false;
}
}
</script>
<style>
body {
background: url("./images/photo_bg.jpg") no-repeat center center fixed;
background-size: cover;
font-size: 16px;
font-family: 'Lato', sans-serif;
font-weight: 300;
margin: 0;
color: #666;
}

```

```
/* Typography */
h1#title {
    font-family: 'Roboto Slab', serif;
    font-weight: 300;
    font-size: 3.2em;
    color: white;
    text-shadow: 0 0 10px rgba(0,0,0,0.8);
    margin: 0 auto;
    max-width: 300px;
    text-align: center;
    position: relative;
    top: 0px;
}

h1#title span span {
    font-weight: 400;
}
h2 {
    text-transform: uppercase;
    color: white;
    font-weight: 400;
    letter-spacing: 1px;
    font-size: 1.4em;
    line-height: 2.8em;
}
a {
    text-decoration: none;
    color: #666;
}

a:hover {
    color: #aeaeae;
}p.small {
    font-size: 0.8em;
    margin: 20px 0 0;
}/* Layout */
.container {
    margin: 0;
}
.top {
    margin: 0;
    padding: 0;
    width: 100%;
```

```

        background: -moz-linear-gradient(top,  rgba(0,0,0,0.6) 0%, rgba(0,0,0,0) 100%); /*
FF3.6-15 */
        background: -webkit-linear-gradient(top,  rgba(0,0,0,0.6) 0%,rgba(0,0,0,0) 100%); /*
Chrome10-25,Safari5.1-6 */
        background: linear-gradient(to bottom,  rgba(0,0,0,0.6) 0%,rgba(0,0,0,0) 100%); /*
W3C, IE10+, FF16+, Chrome26+, Opera12+, Safari7+ */
        filter:  progid:DXImageTransform.Microsoft.gradient(  startColorstr='#99000000',
endColorstr='#00000000',GradientType=0 ); /* IE6-9 */
    }
    .login-box {
        background-color: white;
        max-width: 340px;
        margin: 0 auto;
        position: relative;
        padding-bottom: 30px;
        border-radius: 5px;
        box-shadow: 0 5px 50px rgba(0,0,0,0.4);
        text-align: center;
    }
    .login-box .box-header {
        background-color: #665851;
        margin-top: 0;
        border-radius: 5px 5px 0 0;
    }
    .login-box label {
        font-weight: 700;
        font-size: .8em;
        color: #888;
        letter-spacing: 1px;
        text-transform: uppercase;
        line-height: 2em;
    }
    .login-box input {
        margin-bottom: 20px;
        padding: 8px;
        border: 1px solid #ccc;
        border-radius: 2px;
        font-size: .9em;
        color: #888;
    }
    .login-box input:focus {
        outline: none;
        border-color: #665851;
    }

```

```

        transition: 0.5s;
        color: #665851;
    }
    .login-box button {
        margin-top: 0px;
        border: 0;
        border-radius: 2px;
        color: white;
        padding: 10px;
        text-transform: uppercase;
        font-weight: 400;
        font-size: 0.7em;
        letter-spacing: 1px;
        background-color: #665851;
        cursor:pointer;
        outline: none;
    }
    .login-box button:hover {
        opacity: 0.7;
        transition: 0.5s;
    }
    .login-box button:hover {
        opacity: 0.7;
        transition: 0.5s;
    }
    }
    .selected {
        color: #665851!important;
        transition: 0.5s;
    }/* Animation Delay */
    #logo {
        -webkit-animation-duration: 1s;
        -webkit-animation-delay: 2s;
    }

    .login-box {
        -webkit-animation-duration: 1s;
        -webkit-animation-delay: 1s;
    }
</style>
</head>
<body>
<%
    if(request.getParameter("msg")!=null){

```

```

        out.println("<script>alert('Registered!')</script>");
    }
    if(request.getParameter("msgg")!=null){
        out.println("<script>alert('user not exist')</script>");
    }
    %>
<div class="">
    <nav class="navbar navbar-inverse">
        <div class="container-fluid">
            <!-- Brand and toggle get grouped for better mobile display -->
            <div class="navbar-header">
                <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-
target="#bs-example-navbar-collapse-1" aria-expanded="false">
                    <span class="sr-only">Toggle navigation</span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                </button>
                <a class="navbar-brand" href="#">DATA AUTHORITY</a>
            </div>

            <!-- Collect the nav links, forms, and other content for toggling -->
            <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
                <ul class="nav navbar-nav">
                    <li><a href="admin.jsp">ADMIN</a></li>
                    <li><a href="user.jsp">USER</a></li>
                    <li><a href="ca.jsp">CA</a></li>
                    <li><a href="owner.jsp">OWNER</a></li>
                    <li class="active"><a href="aa.jsp">AUTHORITY</a></li>
                </ul>
            </div><!-- /.navbar-collapse -->
        </div><!-- /.container-fluid -->
    </nav>
    <div class="top">
        <h1 id="title"><span id="logo">AUTHORITY LOGIN</span></h1>
    </div>
    <div class="login-box animated fadeInUp">
        <div class="box-header">
            <h2>Log In</h2>
        </div>
        <form action="aa_login.jsp" method="get" name="name" onsubmit="return validate()">
            <label for="userid">AA ID</label>
            <br/>

```

```
<input type="text" name="userid" id="userid">
<br/>
<label for="password">Password</label>
<br/>
<input type="password" name="password" id="password">
<br/>
<input type="submit" value="Log In" class="btn btn-default">
<br/>
New User? <a href="aa_registration.jsp" style="cursor:pointer">Register</a>
</form>
</div>
</div>
</body>
</html>
```