Chapter 3

Secured Data Access Provision in Cloud Computing

The chapter describes the conceptual design that ensures the security of data over the cloud storage. The design supports efficient user verification and access provision facilities, which assure that only authorised users would be able to access the cloud data. There are some properties presented as thesis goals in chapter 1. This chapter achieves those properties through the use of the concepts presented in chapter 2 of thesis.

3.1 INTRODUCTION

The use of cloud storage over dynamic collaboration environment increases the security risks over the outsourced data. As the need for cloud computing increases rapidly, the security features that assists in protection of cloud data has also become a mandatory one. With intent to preserve the security over the outsourced data, the chapter defines a collective advancement on access control scheme for multi authority cloud storage system. This scheme consists of three major phases through which the property of security over the outsourced data is achieved. Assumptions made for this chapter are defined in section 3.2 and related works in section 3.3. Section 3.4 provides complete description to secured data access provision in cloud computing. Section 3.5 describes the experimental study, the results and discussions in section 3.6 and comparison
to thesis goals are given in section 3.7. Section 3.8 provides summary of the chapter.

3.2 ASSUMPTIONS

When using cloud services the Attribute Authority and Global Certificate Authority associated with the collective advancement scheme are assumed to be the trusted system entities. The Attribute Authority manages all the attribute related activities without the leakage of sensitive information to the unauthorised system users. It is also assumed that the Global Certificate Authority validates and provides appropriate user outputs in a trusted manner. Further, it is assumed that the user behaviour list database table associated with the collective advancement scheme maintains the list of user behaviours for future references in an effective manner.

3.3 RELATED WORKS

This section describes and analyses others approaches towards facing the challenge of security preservation over the cloud data storage. Numerous solutions have been examined, after which an overview is presented with a comparison to the solution properties over the thesis goals.

3.3.1 OVERVIEW

This section describes an overview of others works that are related to the collective advancement scheme described in this chapter. A key policy attribute based encryption scheme through the use of Attribute Based
Encryption technique was given by (Goyal et al., 2006). In KP-ABE the data owner encrypts the data by associating the set of descriptive attributes and access structures related to the decryption key. The access tree contains both leaf and non-leaf nodes, where the leaf node acts as the attributes and non-leaf nodes acts as a threshold gates. The process of cipher-text decryption takes place only when the attributes of the cipher-text matches with the access structure of the decryption key.

(Bethencourt et al., 2007) presented a scheme called Cipher-text policy based Encryption (CP-ABE), which is a converse of KP-ABE scheme. In CP-ABE, the access structure or access policy is related to the data while the set of attributes were related to the decryption key. The CP-ABE scheme defined in this work does not reveal the confidentiality of the data as the access policy associated with this scheme is related to the encrypted data content. The process of cipher-text decryption takes place only when the set of attributes associated with the decryption key fulfils the access policy. The collective advancement scheme defined in this chapter is safe against collusion and plaintext attacks. (Bethencourt et al., 2007), presents a concept, where the access policies were framed using the data attributes and the computational tasks associated with the user revocation process has been delegated to the third party cloud server, without the disclosure of the data content. This scheme achieves the system efficiency through the combination of proxy encryption and lazy-re-encryption techniques. The collective advancement scheme makes use of Attribute History Lists and User Lists through which the accountability of the system is maintained and the efficiency of the system is achieved.

A privacy preserving auditing protocol were given by (Wang et al.,
where the key objective is to overcome and achieve the problem of data integrity. The TPA performs the regular auditing process and verifies the correctness of the encrypted data content, without the need of residing local copy. It delegates the auditing tasks over to the TPA using batch processing concepts. This preserves the integrity of the outsourced data. Through this the communication and computation expenses were reduced. These schemes make use of the Homomorphic Linear Authenticator (HLA) and Random Masking to achieve its intended objectives. Third party auditing is done to find out irregularities of cloud service provider and to monitor the cloud operation according to the service level agreement (Balusamy et al., 2015) A framework of DAC-MACS was given by (Yang et al., 2013) with intent to achieve and increase the efficiency of the attribute revocation and decryption processes. This scheme processes different attributes together through the use of multi-authority cloud storage systems. It assigns the version number to each attribute of the system that assists in the process of solving user attribute revocation issues and assures forward and backward security features to the users of the system. The efficiency of decryption process is improved through the token based concepts and this work delegates the cipher-text update process to the cloud server itself.

3.3.2 COMPARISON OF RELATED WORKS

This section provides comparison of two major works, which is related to collective scheme defined at this chapter. A clear description to comparison was provided in table 3.1.
WORK BY YANG, KAN, ET AL:

The solution to data access control issues were given by (Yang et al., 2013) in DACMAC. Their work states data access control as an effective way to ensure data security over cloud computing services. Due to data outsourcing and untrusted cloud servers, the data access control has become a challenging issue in cloud computing. The existing access control schemes were no longer applicable to cloud storage system, as they generate multiple encrypted copies of the same data or require fully trusted cloud server. Cipher-text Policy Attribute-based Encryption (CP-ABE) is a technique for access control over encrypted data and it requires a trusted authority to manage all the attributes and distributed keys in the system. In cloud storage systems, multiple authorities co-exist and each authority can issue attributes independently. But, the existing CP-ABE techniques could not be directly applied to the data access control processes over multi-authority cloud storage systems, due to inefficient decryption and revocation processes. DAC-MACS (Data Access Control for Multi-Authority Cloud Storage), is an effective access control scheme for multi-authority cloud storage systems with efficient decryption and revocation processes. It constructs a new multi-authority CP-ABE scheme with efficient decryption and revocation method, which achieves both the forward and backward security assurance over the outsourced data.

This work first describes the construction of a new multi-authority CP-ABE scheme with efficient decryption process and proposes an effective attribute revocation method for it. These designs were further applied to an access control scheme for multi-authority cloud storage systems. The
major contributions of this work are given as follows:

- An effective and secure data access control scheme called DAC-MAC (Data Access Control for Multi-Authority Cloud Storage) was given. It is comparatively secured than the random oracle models and has better performance with the existing schemes.

- A new multi-authority CP-ABE scheme with effective decryption process was given. The computations of the decryption process were outsourced using a token based decryption method.

- An efficient immediate attribute revocation method for multi-authority CP-ABE scheme, which achieves both the forward and backward security were given. It is efficient because it incurs less communication and computation costs associated with user revocation process.

The system model consists of five major entities such as global certificate authority (CA), Attribute Authority (AA), cloud server (server), data owners (owners) and data consumers (users). CA sets up the system and accepts the registration of all the users and AA in the system. Every legal user was assigned a user identity and given a pair of global secret and public key for the users. This process is done by CA associated with the system. The CA is not associated with the process of attribute management and secret key generation processes. AA is an independent attribute authority responsible for issuing, revoking and updating user attributes with respect to their role and identity in its domain. In DAC-MACS, each AA is responsible for a set of attributes and has full control over the structure and semantics of its attributes. Each AA is accountable of public attribute key generation and secret key generation process.
for every attribute it manages. The cloud server stores the data owners data and provides data access to the users of the system. Cloud server updates the cipher-text during attribute revocation processes. The data owner divides the data into several components and encrypts each component according to logic granularities and encrypts each data component with different content keys using symmetric encryption techniques. The data owner defines access policies over attributes from multiple authorities and performs the encryption process of their content keys under the policies. Then, the encrypted data were sent to the cloud server together with the cipher-texts. This states that any authorised user could access the cipher-text over the cloud server in an independent manner. The user could be able decrypt the cipher text only when their attributes satisfy the access policies. This enables the user with different attributes to decrypt number of content keys and obtain different granularities of information from the same data. In this manner the process of access provision were made in DACMAC. CASetup, AASetup, SKeyGen, TKGen, Encrypt, Decrypt, UKeyGen, SKUpdate and CTUpdate are the algorithms associated with the DACMAC. The analysis and simulation results depicts that DAC-MAC is highly efficient and secure under the security model. Though DACMAC provides security features over the outsourced data, it does not provide any preventive description to the security attacks that occurs over the system. This may lead to collision attacks during user registration process. In addition to this, DACMAC provides independent access to cloud data users, which may cause several security vulnerabilities over outsourced data. This is due to the unpredictable nature of the cloud data users. Sometimes hackers or malicious user can also access
the cloud server to perform unauthorised system activities.

**WORK DONE BY LAI, JUNZUO ET AL:**

A solution to the verifiability of the outsourced decryption process was given by (Lai et al., 2013). This work overcomes the disadvantages of the work done by (Akinyele et al., 2011). The work done by Green et al reduces the computation overhead of system users by delegating most of the outsourced decryption process activities to untrusted cloud servers. In such a system, the user provides the transformation key to the untrusted server and the cloud server performs most of the decryption process, where the user does only little amount of the decryption process. However this system ensures that even a malicious cloud server could not be able to learn about the encrypted content. But it does not provide any assurance to correctness of the outsourced decryption process. In order to overcome these issues, Junzuo Lai et al proposed a technique called Attribute Based Encryption with Verifiable Outsourced Decryption process, which guarantees user verifiability over the decryption process. This scheme provides both verifiability and security to system users without relying on random users. This work modifies the original model of ABE with outsourced decryption process that allows the data users to verify the correctness of the outsourced decryption process.

A general model of CP-ABE scheme with outsourced decryption process consists of seven algorithmic steps, which includes SetUp, Encrypt, Decrypt, KeyGen, $GenTk_{out}$, $Transform_{out}$ and $Decrypt_{out}$. During the setup phase master keys were given to system users and the KeyGen algorithm provides private key to the users of the system. The encryption process takes the user public key parameters, access structures
and message M and produces encrypted cipher-text to the users of the system. The decryption process takes set of public key parameters, secret keys and cipher-text as an input and produces message M to the users of the system. During $GenTk\text{\textsubscript{out}}$ users public key parameters and private keys were taken as input and returns their corresponding transformation key and retrieval keys, which is used during the process of outsourced decryption process. The $Transform\text{\textsubscript{out}}$ takes the set of public parameters, secret keys and cipher-text and generates partially decrypted cipher-text to system users. This step is done by untrusted cloud servers. The $Decrypt\text{\textsubscript{out}}$ process takes public key parameters, cipher-text, partially decrypted cipher-text and retrieval keys then it outputs the message M to the users of the system. This step is done by the authorized system users. In this manner the process of outsourced decryption process were done in normal CP-ABE scheme. But the ABE scheme with verifiable outsourced decryption process given by (Lai et al., 2013) performs the process of outsourced decryption in four major steps. Setup, Encrypt, Decrypt and KeyGen are the four primary algorithms that assist in the process of outsourced decryption process. Each algorithm has specific functionalities associated with it, through which the verifiability of the outsourced decryption was achieved. The security proofs to the proposed scheme were given through three different security proof theorems. The experimental implementation of ABE with verifiable outsourced decryption process provides better system performance measure, which comparatively reduces the computing resources imposed over the system users.
Table 3.1: Comparisons of Works Relating to Collective Scheme

<table>
<thead>
<tr>
<th>Category</th>
<th>Environment</th>
<th>Properties</th>
<th>Yang, Kan, et al</th>
<th>Lai, Junzuo et al</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security</td>
<td>Cloud and User</td>
<td>Data Confidentiality</td>
<td>++</td>
<td>++</td>
</tr>
<tr>
<td></td>
<td>Cloud</td>
<td>Data Integrity</td>
<td>++</td>
<td>++</td>
</tr>
<tr>
<td></td>
<td>Cloud and User</td>
<td>Data Availability</td>
<td>++</td>
<td>++</td>
</tr>
<tr>
<td></td>
<td>Cloud</td>
<td>Non-Repudiation</td>
<td>+++</td>
<td>+++</td>
</tr>
<tr>
<td></td>
<td>Cloud and User</td>
<td>Accessibility</td>
<td>+</td>
<td>++</td>
</tr>
<tr>
<td></td>
<td>Cloud</td>
<td>Expressiveness</td>
<td>++</td>
<td>++</td>
</tr>
<tr>
<td></td>
<td>Cloud and User</td>
<td>User Addition and Revocation</td>
<td>+++</td>
<td>++</td>
</tr>
<tr>
<td></td>
<td>Cloud</td>
<td>Fine-grained Access Control</td>
<td>+</td>
<td>++</td>
</tr>
<tr>
<td></td>
<td>Cloud and User</td>
<td>Access Control Delegation</td>
<td>+++</td>
<td>++</td>
</tr>
<tr>
<td></td>
<td>Cloud</td>
<td>Many-Many File Sharing</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Performance</td>
<td>Cloud</td>
<td>Scalability</td>
<td>+++</td>
<td>++</td>
</tr>
<tr>
<td></td>
<td>Cloud</td>
<td>Flexibility</td>
<td>+</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Cloud</td>
<td>Durability</td>
<td>++</td>
<td>++</td>
</tr>
</tbody>
</table>
3.4 SECURED DATA ACCESS PROVISION IN CLOUD COMPUTING

3.4.1 OVERVIEW

This section presents collective advancements on access control scheme for multi authority cloud storage systems, which is our conceptual design. It applies the elements from research and related works defined in section 3.3 of this chapter. An essential part of this work deals with the addition of security features to system framework establishment process (Khalil et al., 2014). It overcomes the limitations and extends the work done by DACMAC (Yang et al., 2013). Description to collective advancement scheme where given through several sub sections, where the section 3.4.2 describes the motivation and objective factors, system design in section 3.4.3 and system entities were defined in section 3.4.4. Format of database table in cloud computing and algorithmic steps associated with the system were described in section 3.4.5 and 3.4.6. Overall solution for collective advancement scheme is provided in section 3.4.7.

3.4.2 MOTIVATION AND OBJECTIVES

Cloud offers variety of services to data consumers, which increases the possibility of security breaches over the outsourced data. In addition to this, the multitenant nature of the cloud services in turn creates an essential need for secured cloud access provision methods. In order to solve these concerns, the chapter defines a scheme called collective advancement on access control scheme for multi-authority cloud storage systems. The overall objective of this work is to produce better security and im-
proved access provision to cloud data users over multi-authority cloud storage systems, which is achieved through three major steps accompanying it.

### 3.4.3 SYSTEM DESIGN

![System Design Diagram](image)

**Figure 3.1: System Design of Collective Scheme**

This section presents the conceptual design of a novel access control scheme called collective advancement scheme, which enforces secured access provision over the cloud data. The security establishment is made through three steps, which encompasses the addition of user digital signature during the user registration phase, three level verification processes during the validation of user file access request and the third step includes the creation and maintenance of User Behavior List Database Table (UBLDT) ([Tian et al., 2010](#)). The first part of the work comprises of
three level verification of the user file access requests. This includes the comparative validation of the each attributes with the user data stored at the server side. The next part of the work provides a brief summary about the algorithmic steps associated with secure cloud connection establishment and file transfer processes. The algorithmic step consists of the six phases. System initialization, generation of the secret key, encryption of the data content, decryption of the data content, efficient attribute revocation and back up process. Each phase consists of sub-steps associated with it like user registration, token generation, etc. Steps associated with each phase is clearly defined and instigated using the appropriate set of attributes or user credentials, which is essential for producing computation results accompanying each steps.

3.4.4 SYSTEM ENTITIES

The proposed system consists of four different entities which are described as follows: The Global Certificate Authority (GCA) sets up the system and accomplishes the registration of the users and the attribute authorities. GCA in turn confirms the digitally signed user credentials and assigns the global user ID to the users of the system. The GCA further performs attribute management process and issues a pair of global public and secret key (used in the process of decryption) to the users.

Attribute Authority (AA) is responsible for activities of like allocation, generation, revocation and updating of user attributes depending upon their identities. Further, the AA generates the public attribute key and secret key for all the attributes that he manages. The Cloud Server (CS) manages the data content stored over the cloud server and provides
response to each user file access request only after the completion of verification process. In addition to this, the cloud server is also responsible for maintenance of User Behavior List Database Table (UBLDT).

Data consumers are also called as users of the system, responsible for accessing the cloud services and performs file upload and download processes.

### 3.4.5 FORMAT OF DATABASE TABLES

1. **DATA.DB**: Database table for Encrypted data storage
   
   | UAA | Did | uid | AccR | TTexp | Upuk |
   
2. **UPL.DB**: User privilege list
   
   | UAA | Did | UID | AccR | TTexp | UPUK |
   
3. **UBL.DB**: Database for logging the user behaviour
   
   | U_ID | IP_Address | DID | No_of_Attempt | BL | AUL |
   
4. **BkUP.DB**: Database for storing the backup of data
   
   | BkUPID | LOC | Date | Time |

The above data tables provide the structure of the format of outsourced data in cloud storage. Due to lack of trust over cloud service providers the users store their data in cloud in an encrypted manner with the help of the content key. The user fixes the access policy for different attributes from multiple authorities before the encryption of the data. User of the system requests for the file access to CSP by sending the user access privilege list generated by global certificate authority. The CSP logs into the UPL database and verifies the validity of the user given details. The users would be provided with the token if and only if the entire user given UPL
is a valid one. Then the users could decrypt the data through the tokens provided to them. The User Behaviour List Database Table (UBL.DB) stores all the user performed access over the outsourced data. The maintenance of the UBL.DB is done for the purpose of future references.

### 3.4.6 ALGORITHMIC STEPS ASSOCIATED WITH THE SYSTEM

The algorithmic steps associated with our system consist of six phases. These are described as follows:

**Phase 1: System Initialization**

The system initialization phase describes about setting up of the user and Certificate Authority. System initialization is done through the following steps:

1. The first step accompanying with system initialisation encompasses the setting up of the certificate authorities through the algorithm which could be demonstrated as

   \[
   GCA\text{Setup}(1) \rightarrow MSK, SP, (SigkCA, VerkCA) \tag{3.1}
   \]

   The GCA setup algorithm takes the security parameter \( \lambda \) as input and outputs the Master key, System Parameters and a pair of signature and Verification Key of the CA.

2. The next step encompasses the generation of user signature algorithm

   \[
   User\text{SigGen}(InfoU, Upk) \rightarrow \text{SignU}\{InfoU\}Upk \tag{3.2}
   \]

   This user information algorithm takes the user information and user

---

43
private key as an input and sends it to certificate authority for further registration process. This step helps in the verification of the authenticity and non-repudiation security properties associated with the system.

3. This step comprises the generation of attribute authority signature algorithm

\[
AASigGen(InfoAA,AAprk) \rightarrow \text{SignAA}\{\text{InfoAA}\}AAprk
\]  

(3.3)

This algorithm takes the users information and users private key as inputs and generates the digitally signed information (Okamoto and Takashima, 2014) that has to be sent to the certificate authority for further processing of the system.

4. This step includes the generation of User Registration algorithm which associates the progression of user registration.

\[
UserReg(\text{SignU},\text{Upuk},SP,SigkCA) \rightarrow \\
\{GPKuid,GSKuid,Cert(uid),UPL\{uid,AccR,TTExp,DID\}\}
\]

It takes the digital signature, Users Public Key, System Parameters and the Signature Key as an input and produces the global public/secret key, certificate ID and User Privilege List to the users only after the authentication of the digital signature associated with the user credentials. The user privilege list is created for every user accompanying the system based on their attributes which is used during the process of file access. GCA encrypts the user privilege list and sends it to both the cloud server and the users of the system. The encryption of UPL demonstrates the confidentiality associated with
5. The generation of Attribute Authority Registration algorithm verifies the digitally signed AAs information and generates a unique ID for each successfully verified Attribute Authority credentials.

$$AAReg(\text{SignAA}, \text{AApuk}) \rightarrow \text{aid}$$

(3.4)

6. The Attribute Authority set up algorithm takes attribute authoritys unique ID (aid) as an input and generates a pair of secret / public key and version/ public key as an output for each attribute associated with the system.

$$\text{system.AASetup}(\text{SP}, \text{aid}) \rightarrow \text{SKaid}, \text{PKaid}, \text{VrKxaid}, \text{PKxaid}$$

(3.5)

**Phase2: Generation of Secret Key**

This algorithm takes secret authority key, Public attribute Key, Set of attributes that describes the secret key and certificate associated with the user as an input and outputs the secret key for the user id associated with it.

$$SKeyGen(\text{Skaid}, \text{SP}, \{\text{PKxaid}\}, \text{Suid}, \text{aid}, \text{Cert(uid)}) \rightarrow \text{SKuid}, \text{aid}$$

(3.6)

**Phase3: Encryption of Data Content**

This step performs the encryption of data content. It takes security parameter, set of attributes of associated authorities, set of public attribute keys, data and access structure for all the preferred attributes as an input and produces cipher text as an output. This algorithm first encrypts the data content using the symmetric key encryption techniques with a con-
tent key. Then the encryption of content key under the access structure is made. This returns the cipher text.

\[ Encrypt(SP, \{PKk\}, \{PKxk\}, M, A) \rightarrow CT \] (3.7)

**Phase 4: Decryption of Data Content**

This phase is done by the users of the system with help of the cloud server. The data decryption phase consists of three phases which are described as follows:

A. Efficient Users Request Verification and Logging User Behaviours

1. User Request:

   The user access request includes the User Privilege List given by the Attribute Authority (AA). Here the encryption of User Privilege List, user private key and the public key is made. The user file access request would be sent to the cloud server for further processing.

   \[ UserReq(\{{{UPL}_{uprk}cspuk\}, GPKuid, \{SKuid, k\}}) \] (3.8)

2. Now the cloud server process the UPL (uid, AR, TTexp, DID) and decrypts the user given inputs.

3. Verification Process:

   i. If \((TTexp at [UPL from user] = = TTexp at [UPL from CS])\)

      Then the CS invokes ii.

      Else CS ends by invoking (3.6)

      and logs the user behaviour in the UBL.DB.

This forms the first level of verification. The cloud service provider validates the expiration time from the user given UPL in the users request with the UPL stored in the UPL.DB at the cloud server side. As a re-
sult of the successful verification of the user given credentials the cloud server logs the users behaviour in the user behaviour database table and invokes the next level of verification. If the request fails the cloud servers logs the user behaviour and sends the error message to the users.

ii. If (uid at [UPL from user] == uid at [UPL fromCS])
Then the CS invokes iii.
Else CS ends by invoking (3.6)
and logs the user behaviour in the UBL DB.

During this step CSP compares the user unique ID from the user given UPL with the UPL database present at the cloud server. As a result of the successful completion of this step the cloud server logs the user behaviour and invokes the next level of verification. In case of failure of the verification process an error message would be given to the users (Taylor et al., 2010).

iii. If(AccR at [UPL from user] == AccR at [UPL from CS])
Then the CS invokes (B).
Else CS ends by invoking (3.6)
and logs the user behaviour in the UBL DB.

At this level the CSP checks the access right from the user given UPL with the UPL stored at the UPL database. As a result of the successful completion of this step cloud server logs the user activities in User Behaviour List table and the token generation process would be invoked. In case of the failure of this level an error message is generated to the users. Tokens would be generated to the users based on the validation of the user given access structure.
B. Token Generation

This algorithm takes Cipher text (CT), Global public key of user id, set of public keys of the user as inputs and generates the Token (TK) as output.

\[ TKGen(CT, GPKuid, \{Skuid, k\}) \rightarrow TK \] (3.9)

C. Data Decryption

This algorithm takes the Cipher text (CT), Token (TK), global secret of the user id (GSKuid) as input and transforms the CT to the original message M. This algorithm takes the Cipher text (CT), Token (TK), global secret of the user id (GSKuid) as input and transforms the CT to the original message M.

\[ Decrypt(CT, TK, GSKuid) \rightarrow M \] (3.10)

A counter is maintained as a part of the user verification process that keeps track of the number of times a user tries to perform the file access attempts. A counter threshold limit value of five was fixed for improper user file access requests. Each time the user performs an improper access the counter value is increased by one. A user is blocked when the threshold value of the particular user exceeds the fixed threshold. In this manner secured user access request verification process has been done in our proposed system.

Phase-V: Efficient Attribute Revocation

This phase consists of three processes and it is described as follows:
1. Update key Generation BY AAS

\[ U\text{KeyGen}(SKaid,\{uj\},VKxaid) \rightarrow (KUKj,xaid,CUKxaid) \]  
\hspace{5em} (3.11)

The update key generation algorithm takes secret authority key SKaid, a set of users secretuj and the previous version key of the revoked user as an input and generates the Key Update key(KUKj,xaid) and the cipher text update key(CUKxaid) as an output.

2. SECRET KEY UPDATE BY NON-REVOKED USERS

\[ SKU\text{pdate}(SKuid,aid,KUKuid,xaid) \rightarrow SKuid,aid \]  
\hspace{5em} (3.12)

The current secret key(Skuid,aid) and Key update Key(KUKuid,xaid) were given as inputs to the secret key update algorithm and it generates the new secret key(SKuid,aid) as an output.

3. CIPHER TEXT UPDATES BY CLOUD SERVERS

\[ CTU\text{pdate}(CT,CUKxaid) \rightarrow CT \]  
\hspace{5em} (3.13)

The cipher text update algorithm takes the current ciphertext and the cipher text update key as an input and produces the new cipher text (CT) as output.

**PHASE VI: BACKUP PROCESS**

The backup database table contains the backup copy of the cipher text in a compressed storage manner. \cite{Ferris2008, Chen2012}.
3.4.7 OVERALL SOLUTION

This section presents an overall solution explanation, which clearly describes the benefits of the proposed solution over the existing methods. DAC-MAC (Yang et al., 2013) provides security over outsourced data and verifies authorised users for access provision in a secured manner. Though, DAC-MAC provides security features over outsourced data, there exists several attacks that creates problems to regular cloud users. This attack includes collusion attack called DOS attack, which denies the availability of cloud services to the users of the system. Even though DAC-MAC achieves scalability, fine-grained access and efficiency in decryption and revocation processes, it lags in the prevention of collision attacks that occurs during the user registration process. The collective advancement scheme defined in this chapter overcomes the problem of collision attack in DAC-MAC. This is done through the introduction of digital signature along with user credentials during user registration phase. The processes of embedding digital signature with user given credentials were implemented over existing DAC-MAC framework with the addition of user privilege list over it. DAC-MAC does not impose any constraints over the user registration phase (i.e. entry level) that creates the way for man-in-the-middle attack. It even does not provide any assurance to integrity of user credentials and identity to the personal from whom it is received. With a view to prevent the above mentioned issues, the concept of digital signature was used in the proposed collective advancement scheme that provides solution to the problem of data integrity and non-repudiation issues. In collective advancement scheme, the user sends their credentials with digitally signed signature during user reg-
istration phase. The certificate authority verifies the originality of the digital signature, and generates User Privilege List (UID, AccR, TTexp, DID), other security parameters to the users through the use of the user given inputs. This process is done only when the user given inputs contains a valid value. This imposes the entry level constraints over the user registration phase through which the fake users were identified. The CS sends the UPL to both the users and cloud server associated with the system. The cloud server stores the received UPL in a separate database table called UPL.DB.

Whenever the user sends the file access request, the cloud server would not easily allow the user to interact with it. This is due to the unpredictable nature of the cloud user, sometimes they can be hackers or malicious users. This creates the need for a mechanism to protect the client-server communication process. The collective advancement scheme defined in this chapter provides secure cloud server communication process through three level verification of UPL. This is done through the process of comparison between the UPL at client side to each element of the UPL stored at the server side database. At each level user behaviours were stored in to the User Behaviour Database table UBL.DB. The User behaviour logs database table traces the users ID, file access request given by a particular user, IP address from where the request has come from, Black list( the malicious and untrusted users will be logged in this list) and the Authorized User list (the legitimate or trusted users ID will be logged here). The entries of the UBL.DB log possess the capability to allow or deny the particular user file access request in future. This helps in the detection of DDOS attack from the client side. Further a single
backup database was maintained that stores the backup copy of the encrypted data in a compressed format. This assists in the process of easy recovery of user files during system failures. The remaining process associated with collective advancement scheme takes place in a same manner as it is in DAC-MAC. Further, this scheme provides efficient decryption functions than DAC-MAC, as the data is passed through three level verification process, followed by the process of token generation. As per the extensive data access control model (EDAC-MACS) \cite{Yang2013} with an intent to remove the assumption, that the revoked user were not allowed to send any updated keys to the revoked user. The usage of three level verification over the proposed scheme provides solution to this issue. In this manner the proposed collective advancement scheme provides better security features even at lesser security assumptions.

### 3.5 EXPERIMENTAL STUDY

A collective access advancement scheme was done to overcome the disadvantages of DAC-MAC \cite{Yang2013}. Each object is encrypted with a unique key where this unique key in turn is encrypted using the master key. The process of decryption is made automatically on reclaiming the data. We assume that there are 1000 cloud requestor nodes and per minute there can be 100 nodes to request for the cloud data service. Out of which there are chances that 4% out of 100% of users may be the malicious attackers who are trying to exploit the cloud resource by continuously sending the malicious requests and hence the legitimate users may not be able to request the cloud properly. It is found that as the
time increases, the chances for attackers are increased as the attacker attempts to enter inside the cloud. Hence we have added three levels of the verification process to filter that 4% of malicious users. Also a control measure is put forth in order to block the un-authorized users whose access fails after a number of attempts. Here in order to provide security a threshold value of 5 is kept as the number of attempts that could be made by the users. A counter is also set so that the each attempt made by the user is kept in account and after the counter value has crossed the threshold value the particular user is blocked thereby preventing unauthorized user from accessing. We are satisfied that our additional verification and logging process provides better results to the existing schemes. Our result shows that this method can be adopted in real time environment like Eucalyptus (Salmela-Aro et al., 2009). For the purpose of simulating the scenario KDD Dataset is used.

3.6 RESULTS AND DISCUSSIONS

The process of simulation is done with KDD Dataset. Some of the properties such as the number of legitimate and malicious users per minute were extracted from the data set. The resultant graph proves that with the addition of the three level verification and logging process, the number of malicious requestors decreases and the number of malicious attempts made by the attacker increases. Figure 3.2 provides a clear description about the experimental implementation of collective advancement scheme. Since the user behaviours were logged in a separate database table UBL.DB, based on the entries and number attempts made by the
attackers, preventive measures, were taken that denies the Blacklisted requestors. The graph has been drawn such that time is plotted in x-axis, the number of users in y-axis and with the number of malicious users in z-axis. The conclusion derived from the resultant graph states that the number of malicious users attempted at the first time is found to be high, but in later parts the attempt by the malicious users is found to be comparatively low. This results in the lag of the malicious requestors i.e., the number of the malicious requestors is found to be low. Thus, the result shown by the graph provides a clear view of the collective advancement scheme defined in this chapter, and prevents the process of malicious user access through which the numbers of malicious users have been decreased gradually. The graphical results found from experimental analysis were clearly described in Figure 3.2 illustrating the number of cloud requestors and the number of malicious users during both the attempts.

![Security analysis results](image_url)
### 3.7 COMPARISON TO THESIS GOALS

Table 3.2: Comparisons of System Properties to Thesis Goals

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Security</strong></td>
<td>Cloud and User</td>
<td>Data Confidentiality</td>
<td>++</td>
<td>+++</td>
<td>+++</td>
</tr>
<tr>
<td></td>
<td>Cloud</td>
<td>Data Integrity</td>
<td>++</td>
<td>++</td>
<td>+ + +</td>
</tr>
<tr>
<td></td>
<td>Cloud and User</td>
<td>Data Availability</td>
<td>++</td>
<td>++</td>
<td>++</td>
</tr>
<tr>
<td></td>
<td>Cloud</td>
<td>Non-Repudiation</td>
<td>+ + +</td>
<td>+++</td>
<td>+++</td>
</tr>
<tr>
<td></td>
<td>Cloud and User</td>
<td>Accessibility</td>
<td>+</td>
<td>++</td>
<td>+</td>
</tr>
<tr>
<td></td>
<td>Cloud</td>
<td>Expressiveness</td>
<td>++</td>
<td>++</td>
<td>+ + +</td>
</tr>
<tr>
<td><strong>Access Control</strong></td>
<td>Cloud and User Cloud</td>
<td>User Addition and Revocation</td>
<td>+ + +</td>
<td>++</td>
<td>++</td>
</tr>
<tr>
<td></td>
<td>Cloud</td>
<td>Fine-grained Access Control</td>
<td>+</td>
<td>++</td>
<td>+</td>
</tr>
<tr>
<td></td>
<td>User and Cloud</td>
<td>Access Control Delegation</td>
<td>+ + +</td>
<td>++</td>
<td>+ + +</td>
</tr>
<tr>
<td></td>
<td>User and Cloud</td>
<td>Many-Many File Sharing</td>
<td>++</td>
<td>+</td>
<td>+ + +</td>
</tr>
<tr>
<td><strong>Performance</strong></td>
<td>Cloud</td>
<td>Scalability</td>
<td>+ + +</td>
<td>++</td>
<td>+ + +</td>
</tr>
<tr>
<td></td>
<td>Cloud</td>
<td>Flexibility</td>
<td>+</td>
<td>−</td>
<td>++</td>
</tr>
<tr>
<td></td>
<td>Cloud</td>
<td>Durability</td>
<td>++</td>
<td>++</td>
<td>+ + +</td>
</tr>
</tbody>
</table>
This section provides comparison to the thesis goals which is described in table 3.2. The conclusion derived from the comparison table states that the collective advancement scheme achieves all the properties under security and performance features in a best way. Thus it provides better security and performance features over the outsourced data. Only the property of accessibility under security and some of the access control features were found to be lagging in the collective scheme defined in this chapter. These limitations could be rectified and are achieved in the forthcoming chapters.

3.8 SUMMARY

A clear description to a novel access control scheme called collective advancements on access control scheme for multiauthority cloud storage systems were given in this chapter. This scheme provides effective and secured access provision techniques over multi authority cloud storage systems. It overcomes the limitations of DAC-MAC through effective user logging system, which assists in accountability purposes and further, the use of log database table provides user entry details through which the user access is denied or allowed in future. Finally the addition of digital signature over user given inputs improves the confidentiality of the outsourced data. It also reduces several attacks like manin themiddle, DOS and DDOS attacks. This work could be further extended to the process of auditing over the outsourced data.