International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 22 No. 2 (2026)

ijerst.editor@gmail.com

editor@ijerst.com

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 22, No. 2, 2026

Peer Reviewed, Referred & Indexed Journal

CRYPTCLOUD+ SECURE AND EXPRESSIVE DATA ACCESS CONTROL FOR CLOUD STORAGE

DINTAKURTHI SRI LAKSHMI SINDHU[1], K. RAJA RAJESWARI [2]

1.Student, Dept. of Computer Applications, B. V. Raju College, Bhimavaram. Garagaparru Road, Kovvada, Andhra Pradesh
2.Assistant Professor, Dept. of Computer Applications, B. V. Raju College, Bhimavaram. Garagaparru Road, Kovvada, Andhra Pradesh.

ABSTRACT

Cloud storage has become a widely adopted service for managing and sharing large volumes of data. However, ensuring secure and flexible access control over outsourced data remains a significant challenge because the data is stored outside the owner's physical control. Ciphertext-Policy Attribute-Based Encryption (CP-ABE) is considered a promising technique for enforcing fine-grained access control in cloud environments. Despite its advantages, CP-ABE suffers from a critical limitation known as the **"all-or-nothing" decryption problem**, which can lead to misuse of access credentials when decryption keys are leaked or shared with unauthorized users. To address this issue, this paper proposes **CryptCloud+**, a secure and expressive cloud storage framework that enhances CP-ABE with accountability, auditing, and revocation capabilities. The proposed system introduces an **accountable authority mechanism** to prevent misuse from semi-trusted authorities and integrates **white-box traceability** to identify malicious users who leak their credentials. Additionally, the system supports efficient **user revocation** without affecting legitimate users. Security analysis and experimental evaluations demonstrate that CryptCloud+ effectively enhances data confidentiality, ensures accountability, and provides secure and flexible access control in cloud storage systems.

Keywords

Cloud Storage Security, Ciphertext-Policy Attribute-Based Encryption (CP-ABE), Access Control, Credential Misuse, Data Confidentiality, White-box Traceability, Revocation Mechanism, Cloud Computing Security, Auditing.

I. INTRODUCTION

Cloud computing has revolutionized the way organizations store and manage data by providing scalable, cost-effective, and on-demand storage services. With the rapid growth of cloud platforms, an increasing number of individuals and organizations outsource their sensitive data to cloud storage providers. Although cloud services offer significant advantages such as flexibility, scalability, and reduced infrastructure cost, they also introduce serious security concerns because data is stored outside the owner's direct control. To ensure secure access to outsourced data, encryption techniques are widely used. Traditional encryption schemes provide basic confidentiality but lack flexibility in controlling access to multiple users with different privileges. **Ciphertext-Policy Attribute-Based Encryption (CP-ABE)** has emerged as an effective solution for fine-grained access control in cloud storage systems. In CP-ABE, data owners define access policies based on user attributes, and only users whose attributes satisfy the policy can decrypt the data. However, CP-ABE systems suffer from an inherent limitation known as the **"all-or-nothing" decryption property**, where users possessing valid decryption keys can fully access the ciphertext. This leads to potential **credential leakage**, where users or authorities may intentionally or unintentionally share their decryption keys with unauthorized entities. To address these challenges, this paper proposes **CryptCloud+**, an enhanced CP-ABE based cloud storage system that provides **accountable authority, auditing capability, white-box traceability, and efficient user revocation mechanisms**. The proposed system ensures secure and flexible access control while identifying and revoking malicious users who misuse access credentials.

II. LITERATURE REVIEW

Several research studies have addressed data security and access control issues in cloud storage systems.

Sahai and Waters (2005) introduced the concept of Attribute-Based Encryption (ABE), which enables fine-grained access control over encrypted data. Later, **Bethencourt et al. (2007)** proposed the CP-ABE scheme that allows data owners to define access policies directly in the ciphertext.

Researchers have also attempted to address the limitations of CP-ABE systems. **Chase (2007)** proposed a multi-authority ABE scheme to reduce reliance on a single authority. However, this approach increased system complexity.

Yu et al. (2010) proposed a secure data sharing framework with attribute revocation support, but it still suffered from high computational overhead. Similarly, **Li et al. (2013)** introduced a scalable revocation mechanism for CP-ABE systems, improving efficiency but lacking traceability features.

More recent research has focused on addressing **credential misuse and accountability** in ABE systems. **Liu et al. (2017)** proposed traceable CP-ABE schemes to detect malicious users, while **Zhang et al. (2019)** introduced auditing mechanisms for secure cloud storage.

Despite these advancements, existing solutions typically address only a subset of the following requirements:

- User accountability
- Credential traceability
- Efficient revocation
- Secure authority management

The proposed **CryptCloud+ system** aims to integrate all these features into a single secure framework.

III. PROPOSED METHODOLOGY

The proposed **CryptCloud+ framework** enhances the CP-ABE based cloud storage system by integrating accountability, auditing, and revocation mechanisms.

The system consists of four primary entities:

1. **Data Owner**
2. **Cloud Server**
3. **Attribute Authority**
4. **Cloud Users**

Key Features of the Proposed System

1. **Accountable Authority**
   - Prevents misuse of privileges by semi-trusted authorities.
   - Maintains traceable records of attribute key generation.
2. **White-Box Traceability**
   - Enables identification of malicious users who leak decryption credentials.
3. **Revocation Mechanism**
   - Allows administrators to revoke users whose credentials are compromised.
4. **Auditing Capability**
   - Provides monitoring mechanisms to detect unauthorized data access attempts.

Workflow of the Proposed System

1. Data owner encrypts the data using CP-ABE with defined access policies.
2. Attribute authority generates attribute keys for authorized users.
3. Encrypted data is uploaded to the cloud server.
4. Cloud users request access to encrypted data.
5. System verifies attribute conditions before allowing decryption.
6. If misuse is detected, the system traces and revokes the malicious user.

IV. SYSTEM ARCHITECTURE

The **CryptCloud+ architecture** consists of multiple interacting components designed to ensure secure data storage and controlled access.

Major Components

1. **Data Owner Module**
   - Encrypts files using CP-ABE policies.
   - Uploads encrypted files to cloud storage.
2. **Cloud Storage Server**
   - Stores encrypted files.
   - Handles access requests from users.
3. **Attribute Authority**
   - Generates attribute keys.
   - Maintains accountability records.
4. **Cloud User Module**
   - Requests access to encrypted files.
   - Decrypts files if attributes satisfy access policies.
5. **Auditing and Traceability Module**
   - Tracks credential usage.
   - Identifies malicious behavior.

System Process

1. User Registration
2. Attribute Assignment
3. Data Encryption
4. Data Upload
5. Access Request
6. Policy Verification
7. Data Decryption
8. Misuse Detection & Revocation

V. Algorithms Used

1. Ciphertext-Policy Attribute-Based Encryption (CP-ABE)

CP-ABE allows data owners to define access policies based on user attributes.

Steps:

1. **Setup**

   Generate public key and master key.

2. **Key Generation**

   Attribute authority generates private keys for users based on attributes.

3. **Encryption**

Data owner encrypts data using access policy.

4. **Decryption**

   User decrypts data only if attribute set satisfies policy.

. Traceability Algorithm

Used to detect and identify users who leak decryption keys.

Steps:

1. Monitor decryption attempts.
2. Identify key signature patterns.
3. Match leaked keys with registered users.
4. Flag malicious users.

3. Revocation Algorithm

Removes access privileges from compromised users.

Steps:

1. Identify malicious user.
2. Update revocation list.
3. Re-encrypt affected data.
4. Distribute updated keys to valid users.

Screens

This section presents the various screens of the CryptCloud system that demonstrate the working process of the application. Each screen represents a specific functionality such as user login, registration, file upload, file search, and file download. These screens help in understanding how the system interacts with users and manages cloud data. They also show the interface through which users perform different operations. The following screens illustrate the step-by-step execution of the project.

Home Page

This screen displays the home page of the CryptCloud system where users can select different options to access the application.

Data Owner Dashboard

Upload file to Cloud

Choose file to upload.

Attacker Module

- Simulates unauthorized access by entering file details and injecting malicious content into cloud data.
- It helps test system security, identify vulnerabilities, and demonstrate attack detection and prevention.

Attacker Screen 1 (Enter File Details):

Enter the file name and user details to initiate an attack.

Attacker Screen 2 (Add Malicious Content):

Allows the attacker to inject malicious content into the selected fie. Then, it shows successful attack execution.

Model Deployment

Deployment is the process of transitioning the developed application from a local development environment (Eclipse) to a production-ready web server (Apache Tomcat). This phase ensures that all hardware, software, and database components are synchronized to provide a seamless user experience.

Deployment Environment

For this project, the deployment is carried out on a Client-Server Architecture:

Web Server: Apache Tomcat 9.0 (Handles the JSP/Servlet execution). **Database Server:** MySQL 5.0 (Manages metadata and user authentication). **Development Platform:** Java Runtime Environment (JRE) 1.8.

Step-by-Step Deployment Process

Database Initialization

Before deploying the application code, the backend environment must be prepared:

International Journal of Engineering Research and Science & Technology

Schema Creation: Using SQLyog, the database named crypt cloud is created.

Table Deployment: All SQL scripts for tables like user, data owner, cloud server, request, search, transaction, attacker and auditor are executed to set up the relational structure.

Connectivity: The db_connection.java file in the project is configured with the correct database URL, username, and password to ensure the middleware can communicate with MySQL.

Web Archive (WAR) Generation

In Java J2EE projects, deployment is typically done using a WAR (Web Archive) file:

In Eclipse IDE, the project is right-clicked and the Export > WAR file option is selected.

All project resources—including Java classes, Libraries (.jar files), web content (HTML, CSS, JS), and the web.xml (Deployment Descriptor)—are bundled into a single file named CryptCloud.rar.

Server-Side Hosting

Tomcat Integration: The generated .war file is moved to the webapps directory of the Apache Tomcat installation folder.

Server Startup: The startup.bat file in the Tomcat bin directory is executed.

Auto-Deployment: Apache Tomcat automatically extracts the WAR file and creates a corresponding web directory, making the application live.

Application Access and URL Mapping

Once the server is running, the application is accessed through a web browser using the following URL structure:

http://localhost:8074/CryptCloud_Secure_and_Expressive_Data_Access_Control_for_Cloud_Storage/ index.jsp

VI. EXPERIMENTAL RESULTS

The proposed system was evaluated based on performance and security metrics.

Evaluation Parameters

- Encryption Time
- Decryption Time
- Key Generation Time
- Revocation Overhead
- Traceability Accuracy

Experimental Setup

- Dataset Size: 10MB – 100MB
- Users: 50 – 200
- Attributes: 5 – 20
- Cloud Platform Simulation Environment

Observations

Parameter	**Existing CP-ABE**	**CryptCloud+**
Encryption Time	Moderate	Slightly Higher
Decryption Time	Fast	Fast
Security Level	Medium	High
Credential Traceability	Not Supported	Supported
Revocation	Limited	Efficient

Result Analysis

Experimental results show that **CryptCloud+ improves security and accountability while maintaining acceptable computational overhead.**

CONCLUSION & FUTURE WORK

In this paper, we proposed **CryptCloud+**, an enhanced cloud storage framework designed to address credential misuse in CP-ABE based systems. The proposed system integrates **accountable authority, white-box traceability, auditing, and efficient revocation mechanisms** to strengthen cloud data security.

The system effectively detects malicious users who leak credentials and ensures that unauthorized access to encrypted data is prevented. Experimental results demonstrate that CryptCloud+ provides improved security and flexible access control compared to existing CP-ABE based systems.

Future Work

Future research can focus on:

- Reducing computational overhead in large-scale environments
- Integrating **blockchain-based auditing systems**
- Supporting **dynamic attribute updates**

- Enhancing scalability for **big data cloud platforms**

References

1. A. Sahai and B. Waters, "Fuzzy Identity-Based Encryption," *EUROCRYPT*, 2005.
2. J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-Policy Attribute-Based Encryption," *IEEE Symposium on Security and Privacy*, 2007.
3. M. Chase, "Multi-Authority Attribute Based Encryption," *TCC*, 2007.
4. S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving Secure Data Sharing in Cloud Computing," *IEEE INFOCOM*, 2010.
5. M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou, "Scalable and Secure Sharing of Personal Health Records," *IEEE TPDS*, 2013.
6. K. Yang and X. Jia, "Efficient Secure Data Sharing Scheme in Cloud Storage," *IEEE Transactions on Cloud Computing*, 2014.
7. H. Li et al., "Enabling Public Verifiability and Data Dynamics for Storage Security in Cloud Computing," *IEEE Transactions on Parallel and Distributed Systems*, 2015.
8. Z. Liu et al., "Traceable Attribute-Based Encryption for Cloud Storage," *IEEE Transactions on Information Forensics and Security*, 2017.
9. Q. Zhang et al., "Secure and Efficient Data Access Control in Cloud Computing," *IEEE Access*, 2019.
10. C. Wang et al., "Privacy-Preserving Public Auditing for Secure Cloud Storage," *IEEE Transactions on Computers*, 2013.